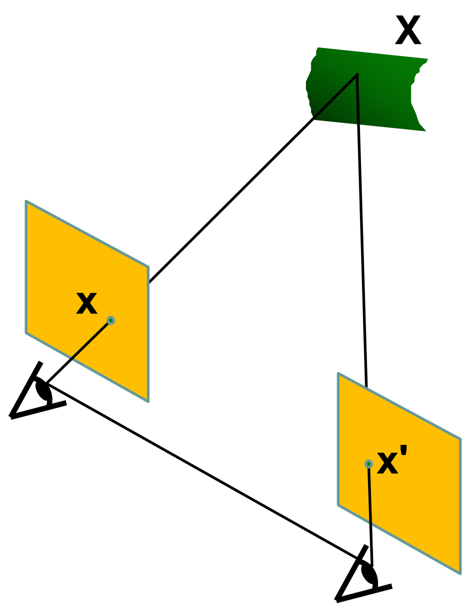
第十一课：对极几何

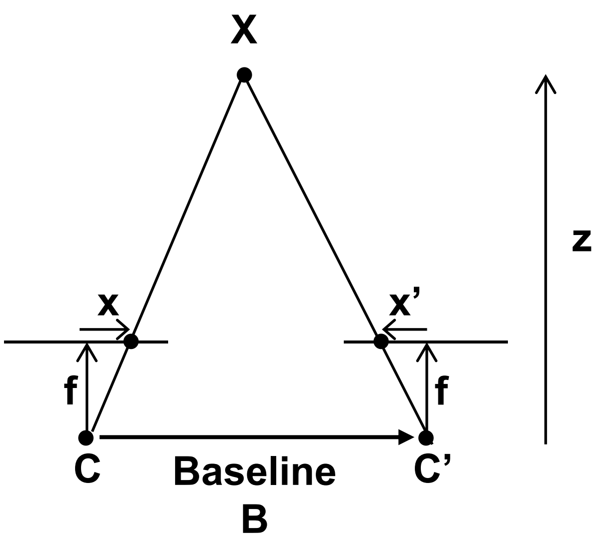
清华大学 自动化系

季向阳
xyji@tsinghua.edu.cn

基于立体视觉的深度估计

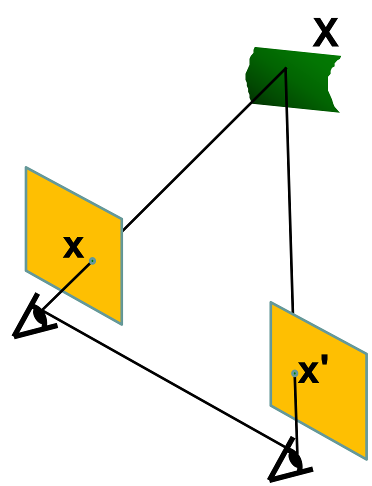
- 目标：通过找到与x对应的图像坐标x'来恢复深度。





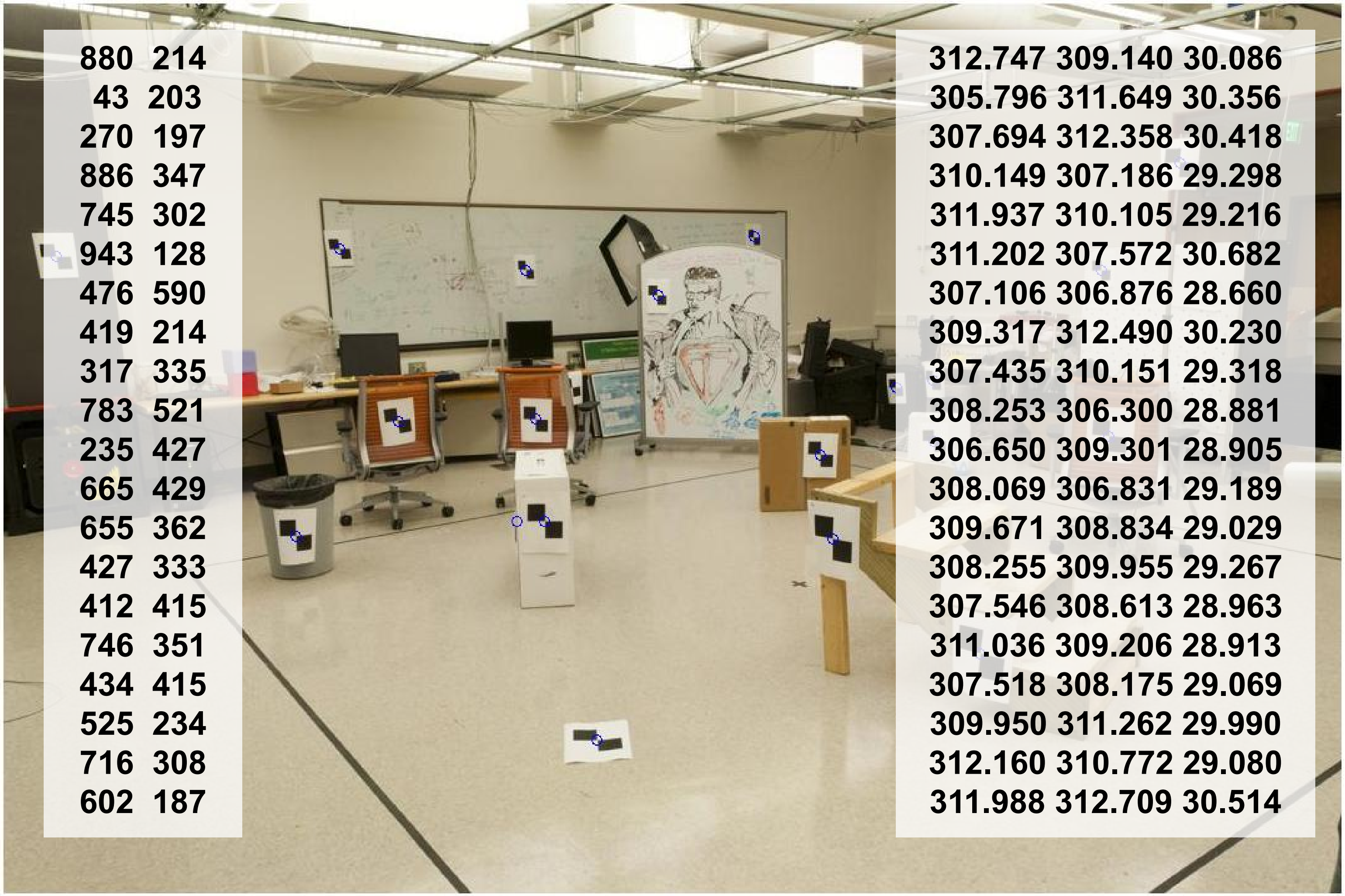
深度估计

- 目标：通过寻找与x相对应的图像坐标x'来恢复深度。
- 子问题
  - 校准： 如何恢复摄像机的关系（如果未知的话）？
  - 对应： 如何搜索匹配点x'？

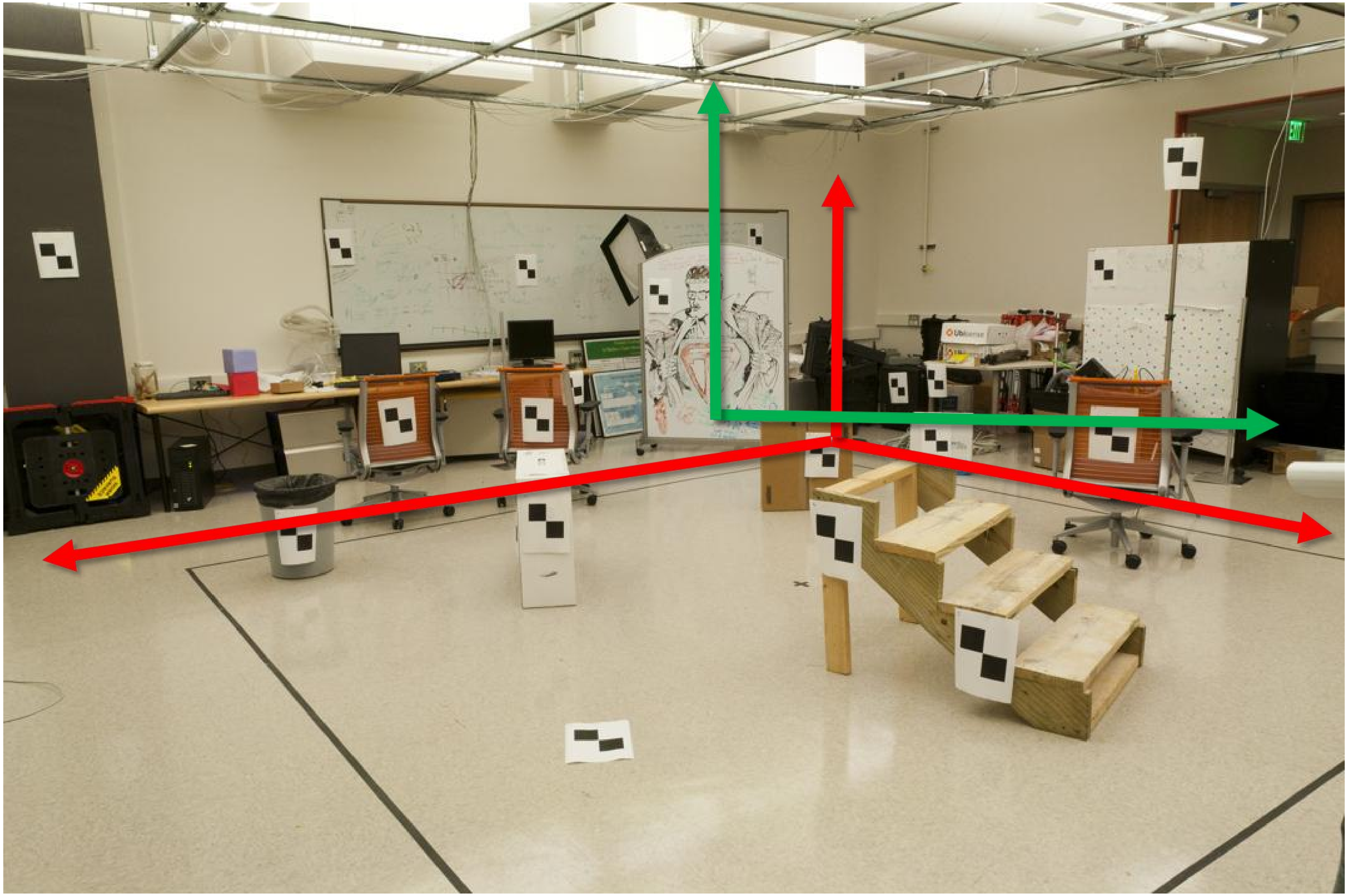


Outline

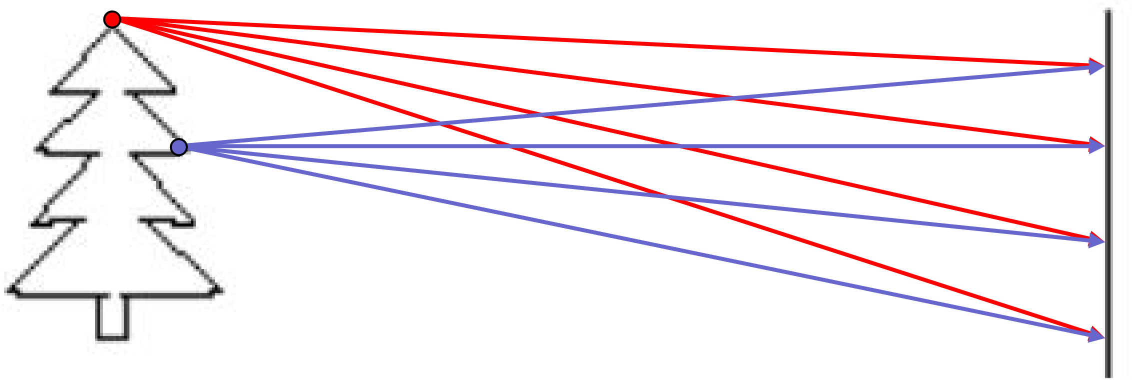
我们如何校准相机?


880 214
43 203
270 197
886 347
745 302
943 128
476 590
419 214
317 335
783 521
235 427
665 429
655 362
427 333
412 415
746 351
434 415
525 234
716 308
602 187
312.747 309.140 30.086
305.796 311.649 30.356
307.694 312.358 30.418
310.149 307.186 29.298
311.937 310.105 29.216
311.202 307.572 30.682
307.106 306.876 28.660
309.317 312.490 30.230
307.435 310.151 29.318
308.253 306.300 28.881
306.650 309.301 28.905
308.069 306.831 29.189
309.671 308.834 29.029
308.255 309.955 29.267
307.546 308.613 28.963
311.036 309.206 28.913
307.518 308.175 29.069
309.950 311.262 29.990
312.160 310.772 29.080
311.988 312.709 30.514

世界坐标与相机坐标



图像形成

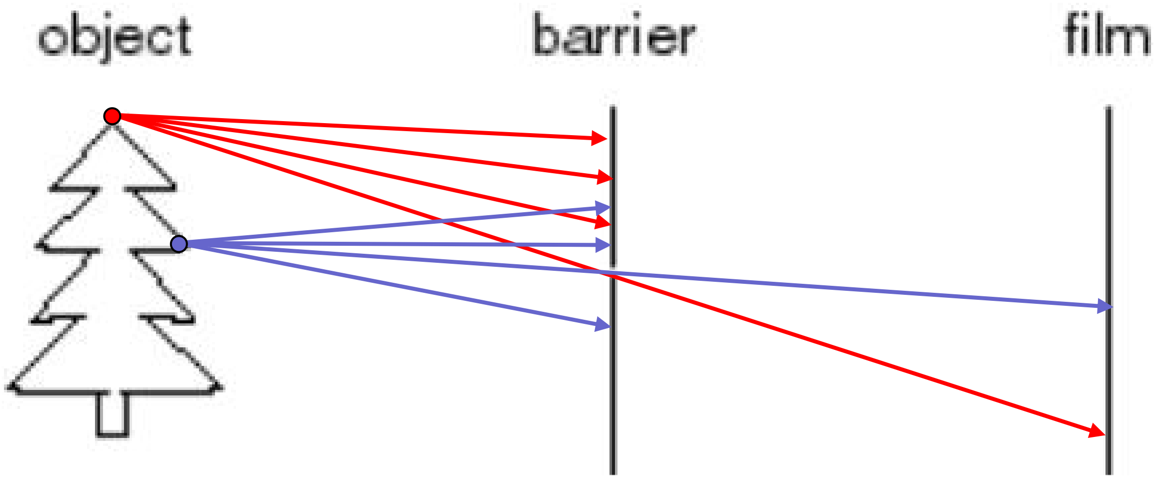


让我们设计一个相机

--想法1：在物体前面放一张胶片

--我们能得到一张显像合适的图像吗？

针孔摄像机

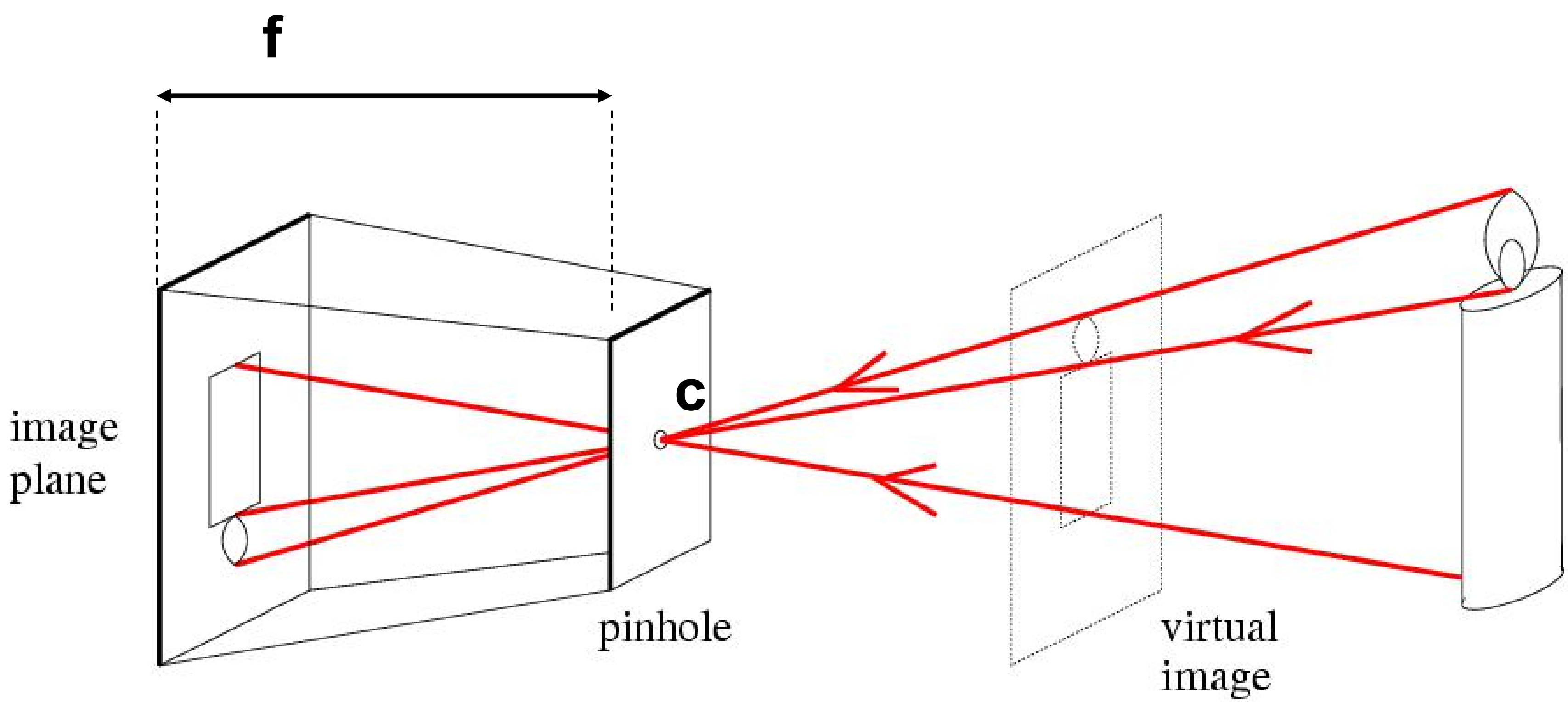


想法2：添加屏障以阻挡大部分光线

--这种方法减少了模糊

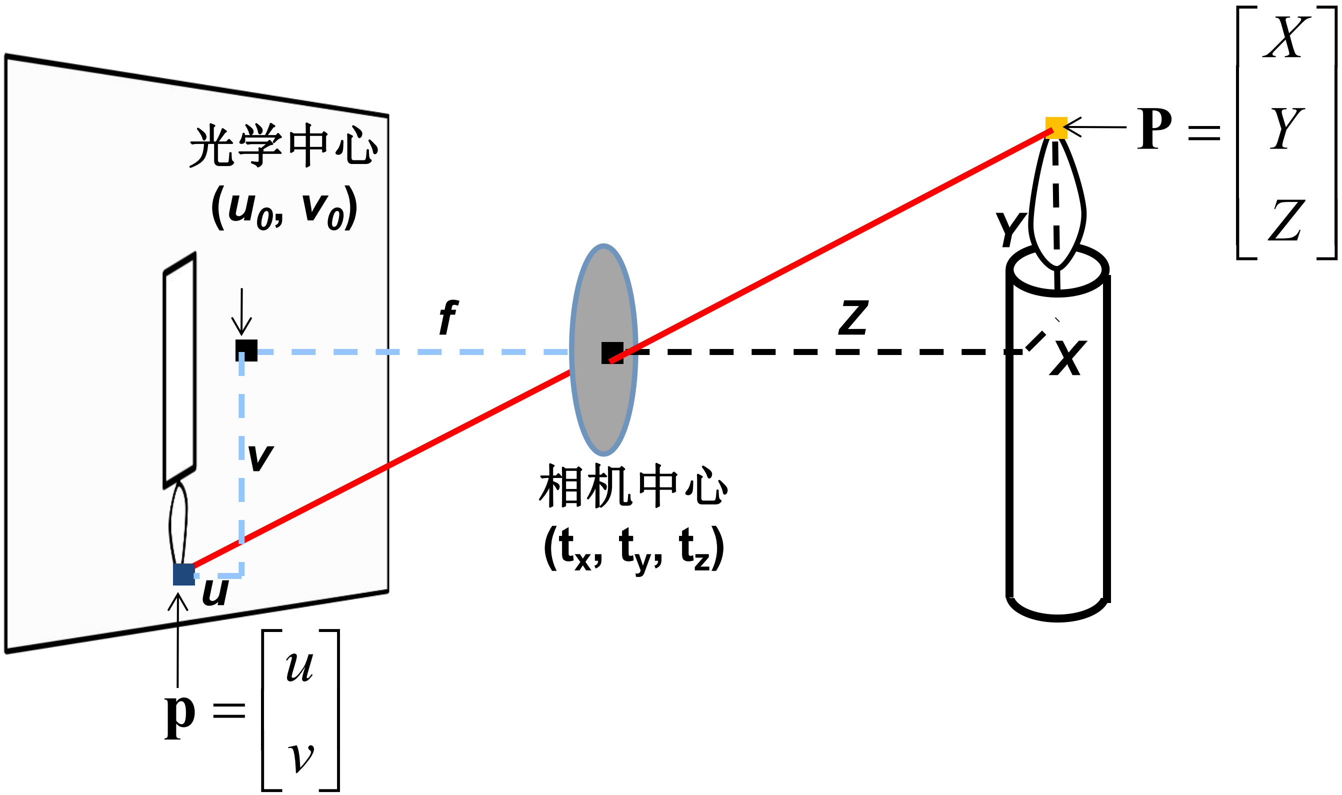
--这被称为光圈

针孔摄像机



f = 焦距
c = 相机中心

齐次坐标



光学中心
(u0, v0)
f
v
u
$\mathbf{p} = \begin{bmatrix} u \\ v \end{bmatrix}$
相机中心
(tx, ty, tz)
Z
Y
X
$\mathbf{P} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$

投影：世界坐标->图像坐标

转换过程

转换为齐次坐标

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

齐次图像坐标

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

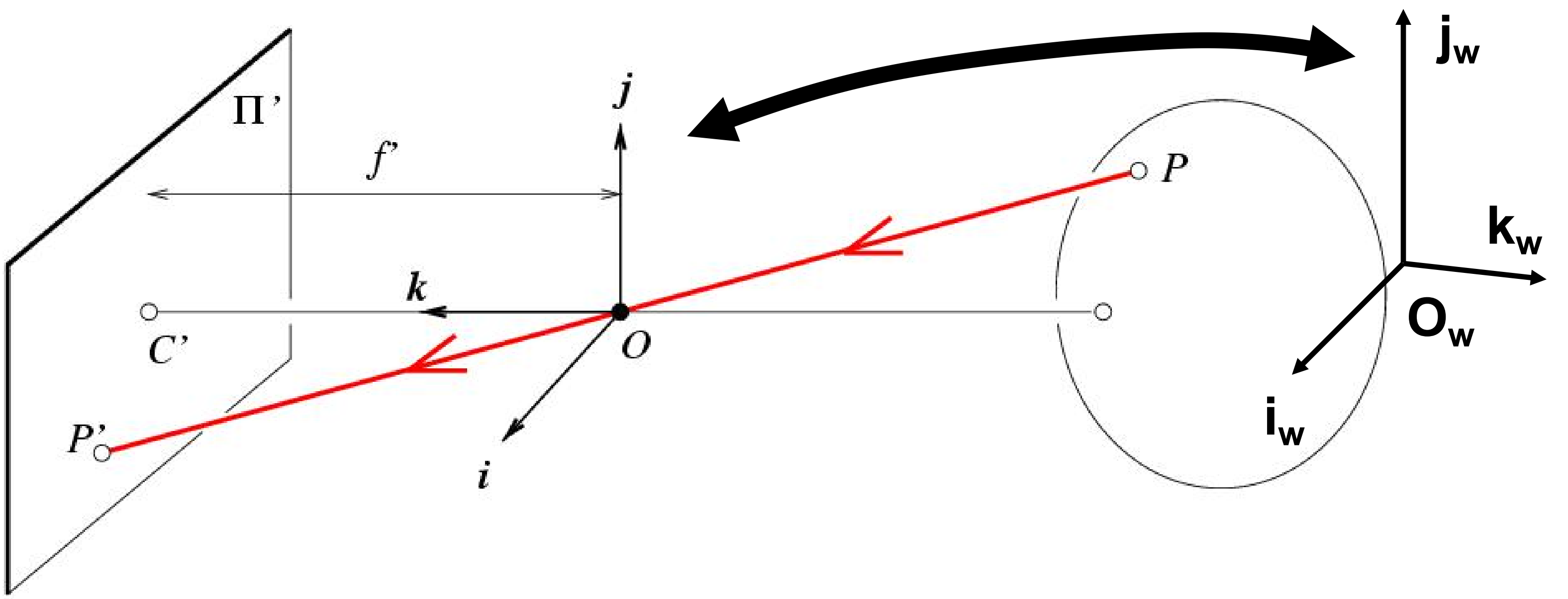
齐次场景坐标

由齐次坐标逆转换

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

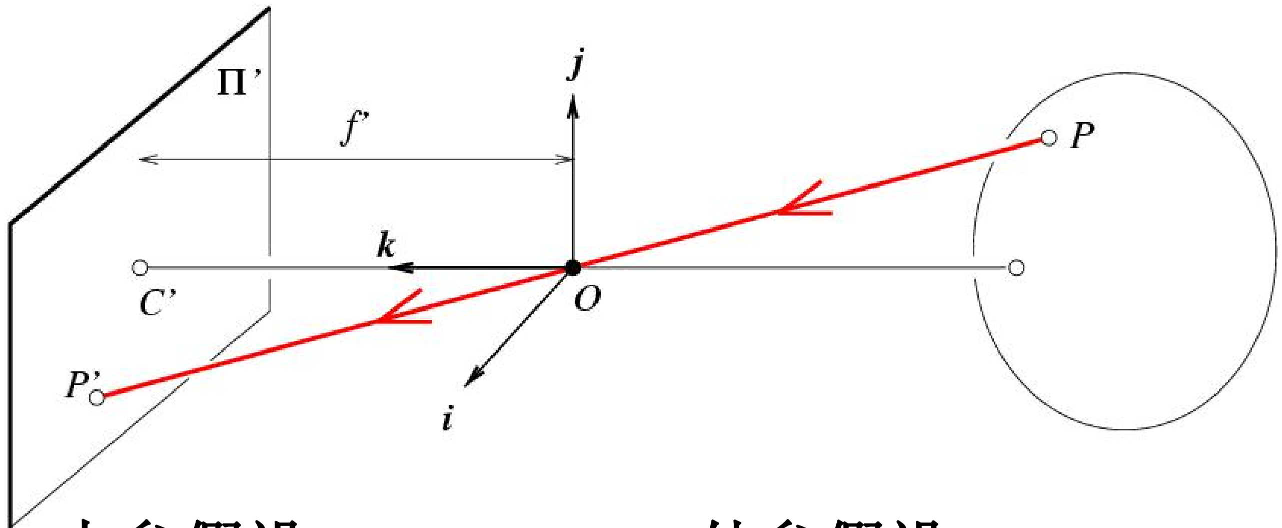
投影矩阵



$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

x:图像坐标: (u,v,1)
K:内参矩阵(3x3)
R:旋转矩阵(3x3)
t:平移矩阵(3x1)
X:世界坐标: (X,Y,Z,1)

投影矩阵



内参假设

- 单位长宽比
  光学中心在(0,0)处
- 无歪斜

外参假设

•无旋转
相机在(0，0，0)处

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}f & 0 & 0 & 0\\0 & f & 0 & 0\\0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

K

删除假设：已知光学中心

内参假设
- 单位长宽比

无歪斜

外参假设
- 无旋转

相机在（0，0，0）

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \quad \Rightarrow \quad w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \boxed{\begin{matrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{matrix}} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

删除假设：正方形像素

内参假设
•无歪斜

外参假设
•无旋转
相机在（**0**，**0**，**0**）

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}\alpha & 0 & u_0 & 0\\0 & \beta & v_0 & 0\\0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

删除假设：无歪斜像素

内参假设

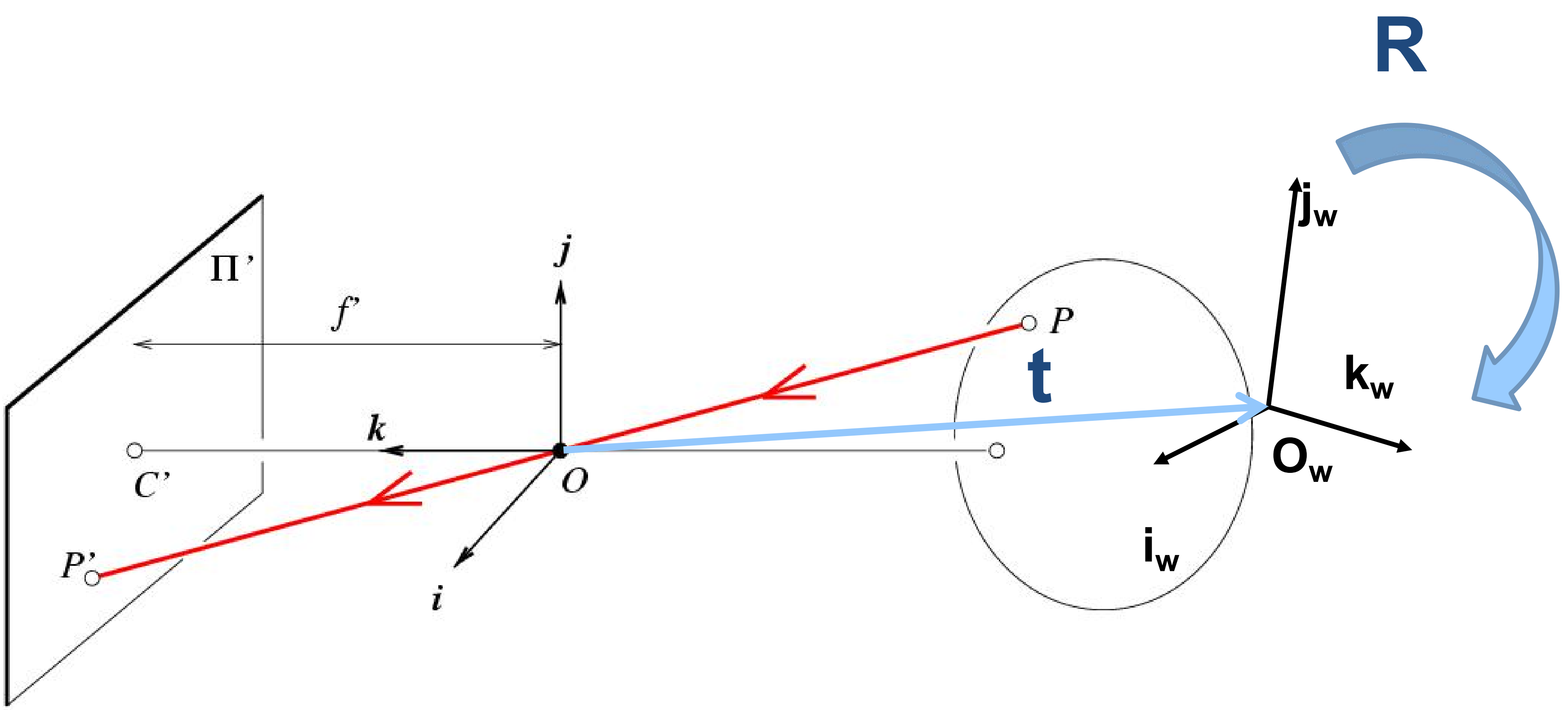
外参假设

•无旋转

相机在（**0，0，0**）

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}\alpha & s & u_0 & 0\\0 & \beta & v_0 & 0\\0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

定向和平移相机



允许相机平移

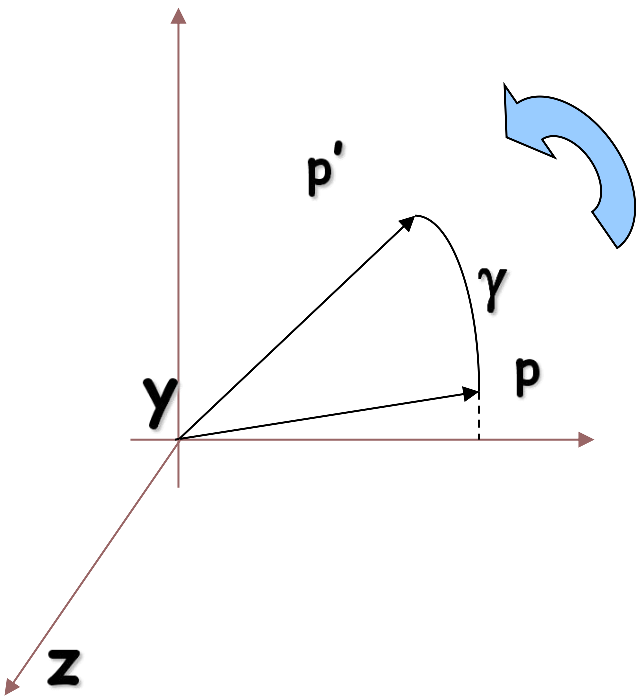
内参假设

外参假设
•无旋转

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{t}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix}u\\ v\\ 1\end{bmatrix} = \begin{bmatrix}\alpha & s & u_0\\ 0 & \beta & v_0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & t_x\\ 0 & 1 & 0 & t_y\\ 0 & 0 & 1 & t_z\end{bmatrix}\begin{bmatrix}x\\ y\\ z\\ 1\end{bmatrix}$$

点的 3D 旋转

绕坐标轴旋转，逆时针方向：



$$R_x(\alpha)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\alpha & -\sin\alpha\\ 0 & \sin\alpha & \cos\alpha\end{bmatrix}$$

$$R_y(\beta)=\begin{bmatrix}\cos\beta & 0 & \sin\beta\\ 0 & 1 & 0\\ -\sin\beta & 0 & \cos\beta\end{bmatrix}$$

$$R_z(\gamma)=\begin{bmatrix}\cos\gamma & -\sin\gamma & 0\\ \sin\gamma & \cos\gamma & 0\\ 0 & 0 & 1\end{bmatrix}$$

允许相机旋转

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & s & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

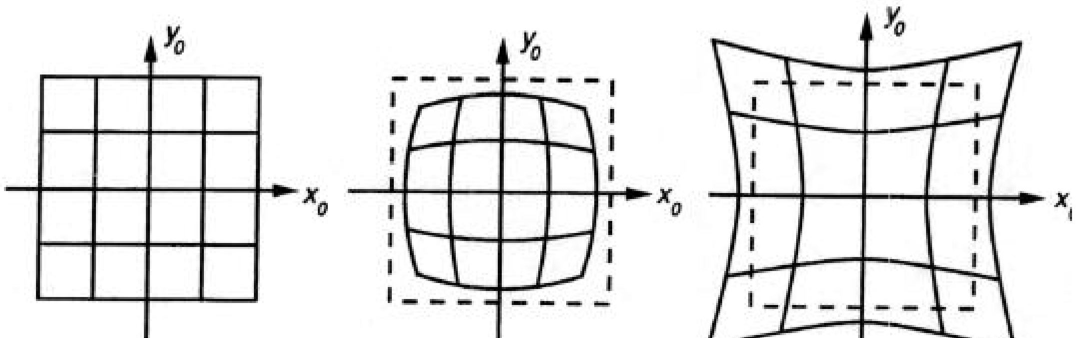
自由度

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \overset{\textbf{5}}{\begin{bmatrix} \alpha & s & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}} \overset{\textbf{6}}{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

针孔摄像以外：径向畸变

- 常见于广角镜头或特殊应用（例如安全）
  在投影中创建非线性项
  通常通过求解非线性项然后校正图像来处理



No Distortion　　Barrel Distortion　　Pincushion Distortion

修正桶形畸变

如何校准相机？
（也称为“相机切除术”）

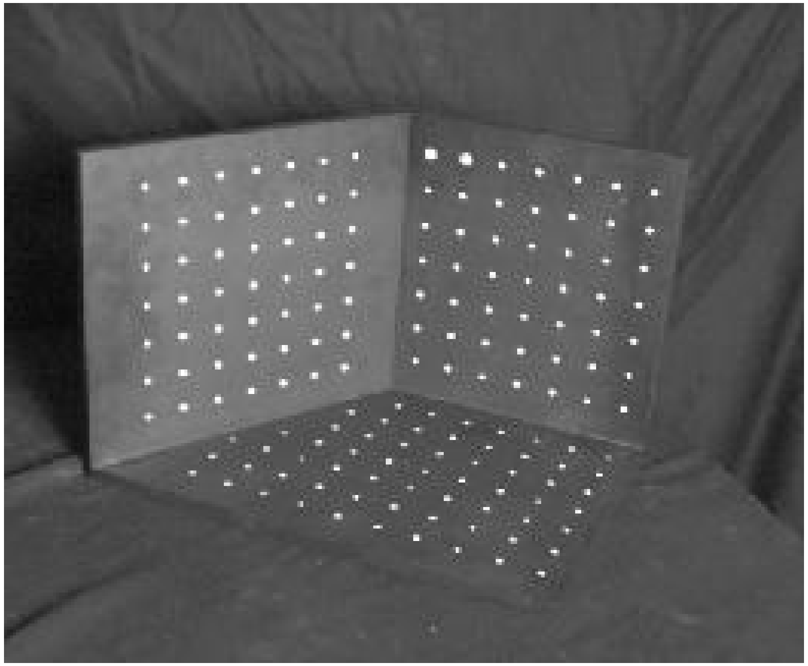
$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

校准相机

使用具有已知几何图形的场景

- 将图像点对应于 3D 点
  获取最小二乘解（或非线性解）



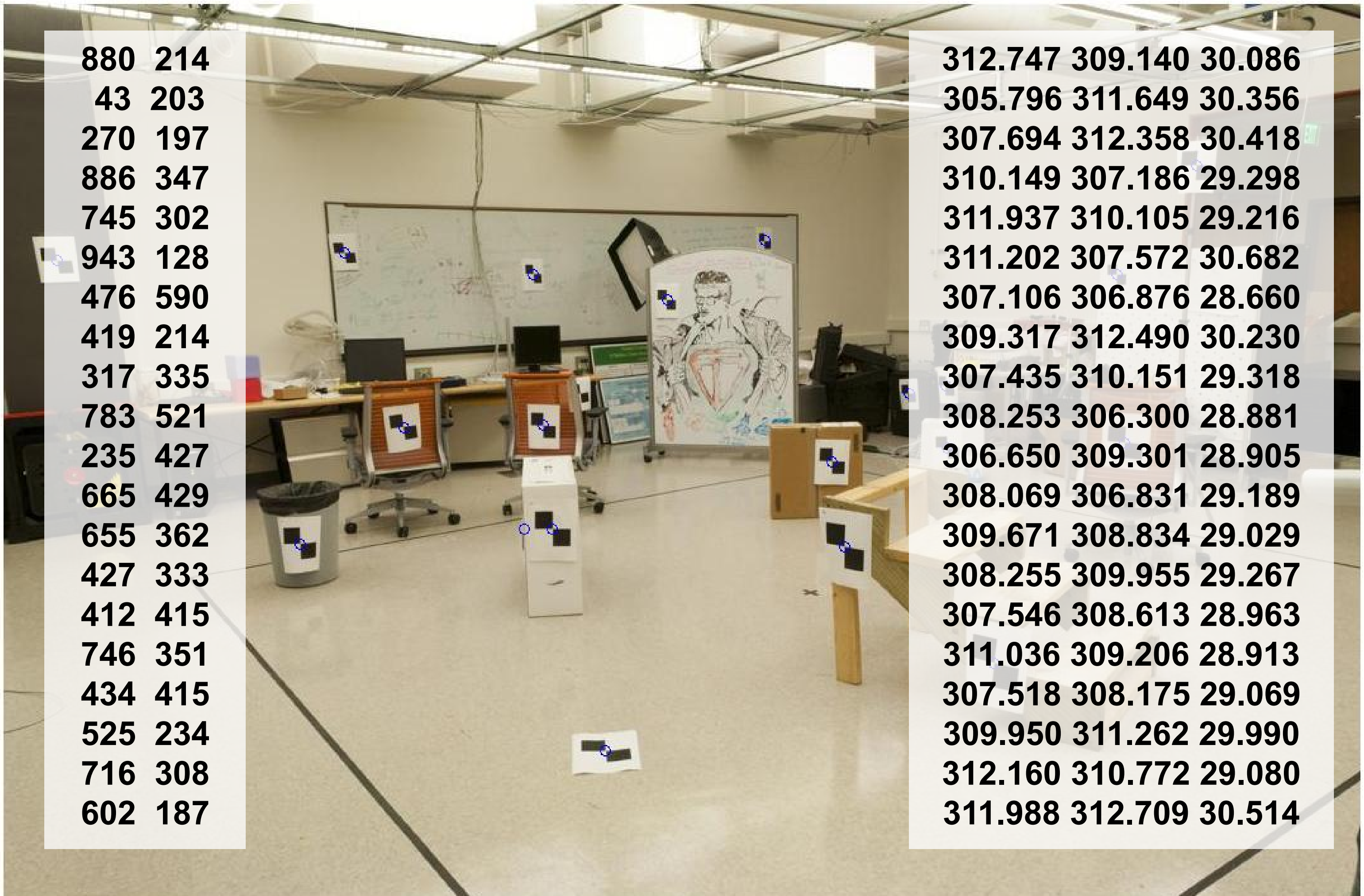
已知的 2D 图像坐标

已知的 3D 位置

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

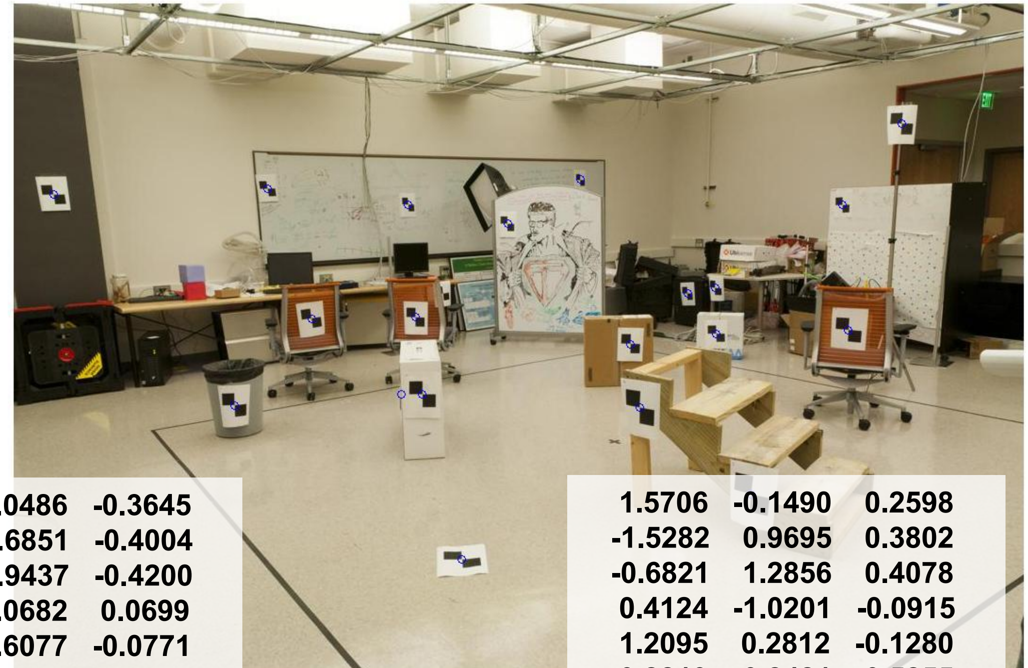
未知相机参数

我们如何校准相机？



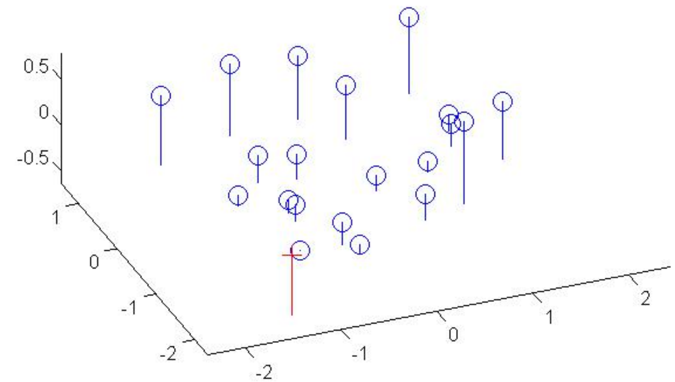
已知的 2D 图像坐标		已知的 3D 位置		
880	214	312.747	309.140	30.086
43	203	305.796	311.649	30.356
270	197	307.694	312.358	30.418
886	347	310.149	307.186	29.298
745	302	311.937	310.105	29.216
943	128	311.202	307.572	30.682
476	590	307.106	306.876	28.660
419	214	309.317	312.490	30.230
317	335	307.435	310.151	29.318
783	521	308.253	306.300	28.881
235	427	306.650	309.301	28.905
665	429	308.069	306.831	29.189
655	362	309.671	308.834	29.029
427	333	308.255	309.955	29.267
412	415	307.546	308.613	28.963
746	351	311.036	309.206	28.913
434	415	307.518	308.175	29.069
525	234	309.950	311.262	29.990
716	308	312.160	310.772	29.080
602	187	311.988	312.709	30.514

相机中心估算



1.0486	-0.3645
-1.6851	-0.4004
-0.9437	-0.4200
1.0682	0.0699
0.6077	-0.0771
1.2543	-0.6454
-0.2709	0.8635
-0.4571	-0.3645
-0.7902	0.0307
0.7318	0.6382
-1.0580	0.3312
0.3464	0.3377
0.3137	0.1189
-0.4310	0.0242
-0.4799	0.2920
0.6109	0.0830
-0.4081	0.2920
-0.1109	-0.2992
0.5129	-0.0575
0.1406	-0.4527

1.5706	-0.1490	0.2598
-1.5282	0.9695	0.3802
-0.6821	1.2856	0.4078
0.4124	-1.0201	-0.0915
1.2095	0.2812	-0.1280
0.8819	-0.8481	0.5255
-0.9442	-1.1583	-0.3759
0.0415	1.3445	0.3240
-0.7975	0.3017	-0.0826
-0.4329	-1.4151	-0.2774
-1.1475	-0.0772	-0.2667
-0.5149	-1.1784	-0.1401
0.1993	-0.2854	-0.2114
-0.4320	0.2143	-0.1053
-0.7481	-0.3840	-0.2408
0.8078	-0.1196	-0.2631
-0.7605	-0.5792	-0.1936
0.3237	0.7970	0.2170
1.3089	0.5786	-0.1887
1.2323	1.4421	0.4506



未知相机参数

已知的 2D 图像坐标

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

已知的 3D 位置

$$su = m_{11}X + m_{12}Y + m_{13}Z + m_{14}$$

$$sv = m_{21}X + m_{22}Y + m_{23}Z + m_{24}$$

$$s = m_{31}X + m_{32}Y + m_{33}Z + m_{34}$$

$$(m_{31}X + m_{32}Y + m_{33}Z + m_{34})u = m_{11}X + m_{12}Y + m_{13}Z + m_{14}$$

$$(m_{31}X + m_{32}Y + m_{33}Z + m_{34})v = m_{21}X + m_{22}Y + m_{23}Z + m_{24}$$

$$m_{31}uX + m_{32}uY + m_{33}uZ + m_{34}u = m_{11}X + m_{12}Y + m_{13}Z + m_{14}$$

$$m_{31}vX + m_{32}vY + m_{33}vZ + m_{34}v = m_{21}X + m_{22}Y + m_{23}Z + m_{24}$$

未知相机参数

已知的 2D 图像坐标

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

已知的 3D 位置

$$m_{31}uX + m_{32}uY + m_{33}uZ + m_{34}u = m_{11}X + m_{12}Y + m_{13}Z + m_{14}$$
$$m_{31}vX + m_{32}vY + m_{33}vZ + m_{34}v = m_{21}X + m_{22}Y + m_{23}Z + m_{24}$$

$$0 = m_{11}X + m_{12}Y + m_{13}Z + m_{14} - m_{31}uX - m_{32}uY - m_{33}uZ - m_{34}u$$
$$0 = m_{21}X + m_{22}Y + m_{23}Z + m_{24} - m_{31}vX - m_{32}vY - m_{33}vZ - m_{34}v$$

未知相机参数

已知的 2D 图像坐标

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

已知的 3D 位置

$$0 = m_{11}X + m_{12}Y + m_{13}Z + m_{14} - m_{31}uX - m_{32}uY - m_{33}uZ - m_{34}u$$

$$0 = m_{21}X + m_{22}Y + m_{23}Z + m_{24} - m_{31}vX - m_{32}vY - m_{33}vZ - m_{34}v$$

- 方法1 – 齐次线性方程组。使用线性最小二乘求解 m 的解

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\ & & & & & & \vdots & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

```
[U, S, V] = svd(A);
M = V(:,end);
M = reshape(M,[],3)'
```

对于 python，请参阅 **numpy.linalg.svd**

未知相机参数

已知的 2D 图像坐标

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

已知的 3D 位置

- 方法 2 – 非齐次线性方程组。使用线性最小二乘求解 m 的解

Ax=b 格式

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 \\ & & & & & & \vdots & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \\ \vdots \\ u_n \\ v_n \end{bmatrix}$$

```
M = A\Y;
M = [M;1];
M = reshape(M,[],3)';
```

对于 **python**，请参阅 **numpy.linalg.lstsq**

线性校准

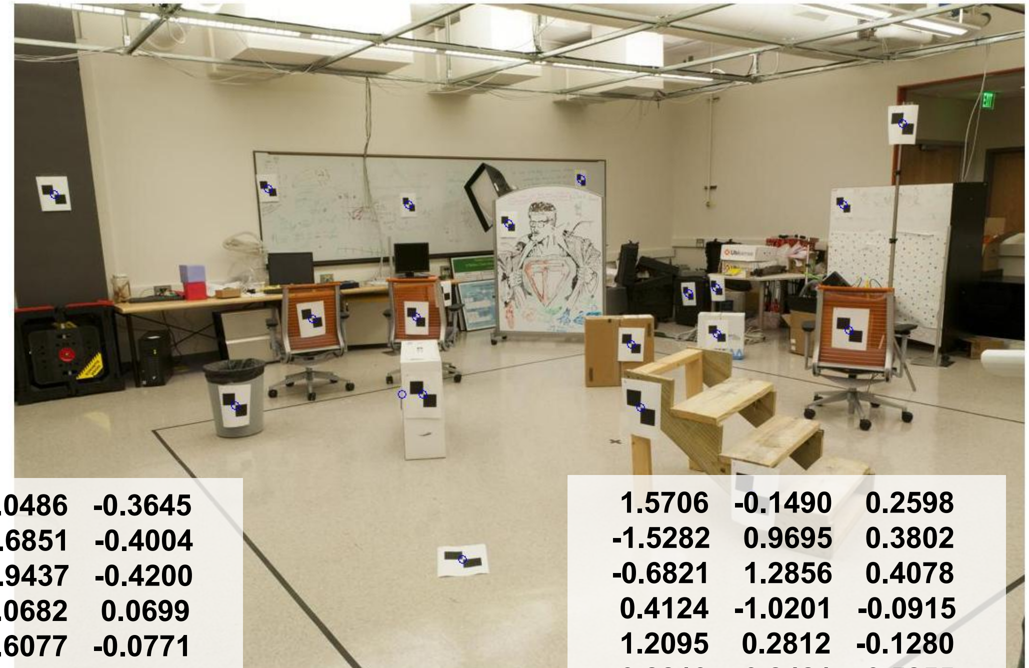
- 优势
  - 易于制定和解决
    为非线性方法提供初始化
- 劣势
  - 不直接提供相机参数
    不模拟径向畸变
    无法添加限制，例如已知焦距
- 非线性方法是首选
  - 将误差定义为投影点和测量点之间的差异
    使用牛顿法或其他非线性优化将误差降至最低

我们是否可以将M分解回 K[R|T]?

- 是的!
- 可以使用 RQ 分解（注意 - 不是更熟悉的 QR 分解）。R（右对角线）是 K，Q（正交基）是 R.T，[R |T]的最后一列，是 inv（K） * M 的最后一列。
  但是需要做一些后处理以确保矩阵有效。
  参考 http://ksimek.github.io/2012/08/14/decompose/

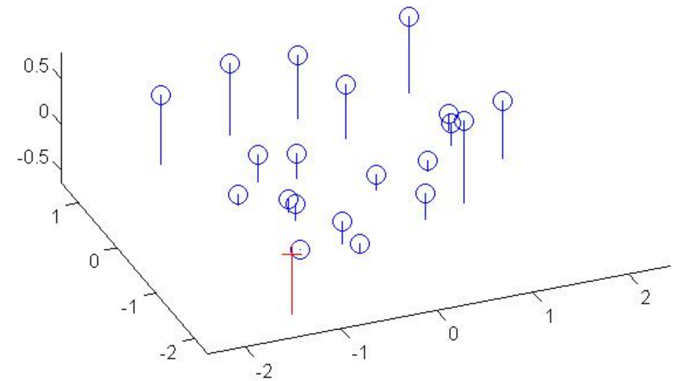
对于项目 3，我们需要相机中心

相机中心估算

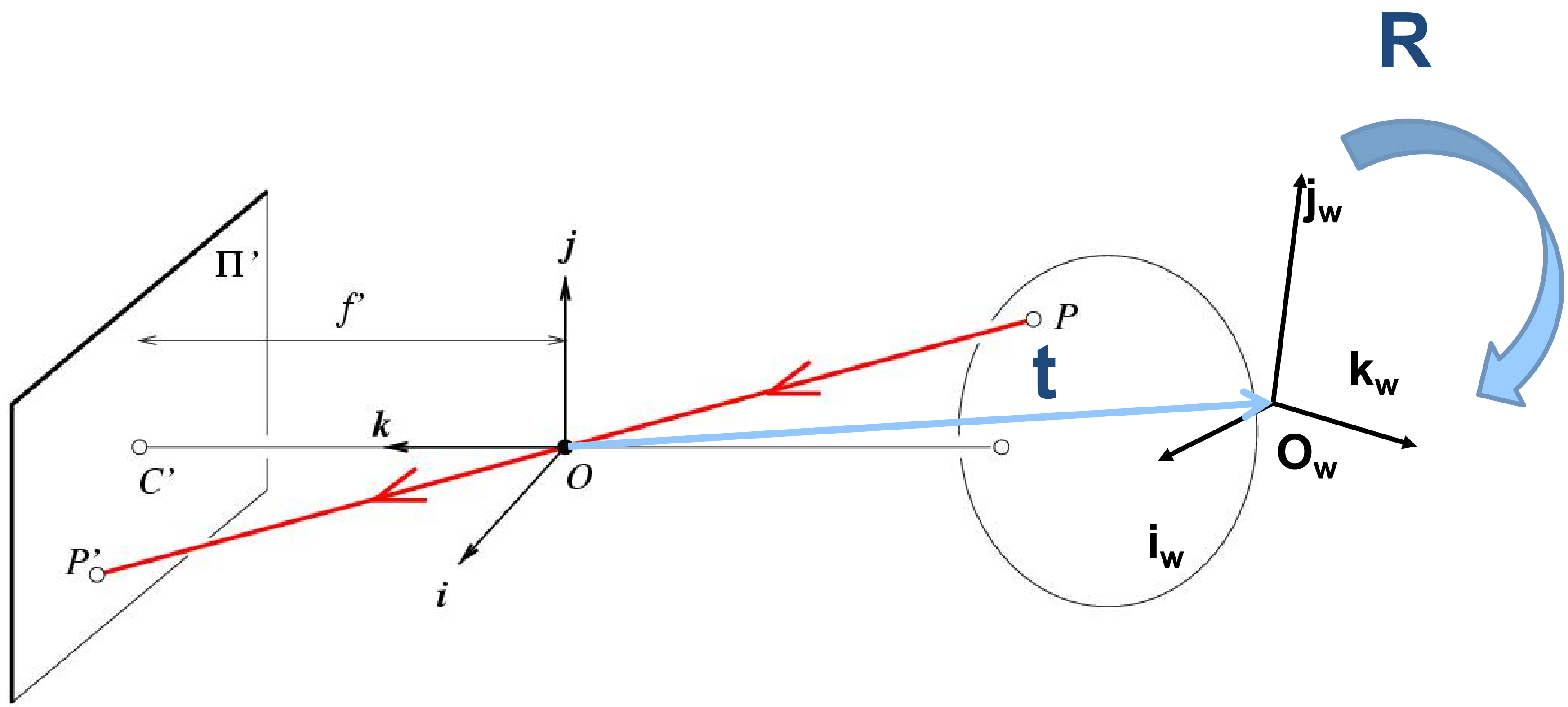


1.0486	-0.3645
-1.6851	-0.4004
-0.9437	-0.4200
1.0682	0.0699
0.6077	-0.0771
1.2543	-0.6454
-0.2709	0.8635
-0.4571	-0.3645
-0.7902	0.0307
0.7318	0.6382
-1.0580	0.3312
0.3464	0.3377
0.3137	0.1189
-0.4310	0.0242
-0.4799	0.2920
0.6109	0.0830
-0.4081	0.2920
-0.1109	-0.2992
0.5129	-0.0575
0.1406	-0.4527

1.5706	-0.1490	0.2598
-1.5282	0.9695	0.3802
-0.6821	1.2856	0.4078
0.4124	-1.0201	-0.0915
1.2095	0.2812	-0.1280
0.8819	-0.8481	0.5255
-0.9442	-1.1583	-0.3759
0.0415	1.3445	0.3240
-0.7975	0.3017	-0.0826
-0.4329	-1.4151	-0.2774
-1.1475	-0.0772	-0.2667
-0.5149	-1.1784	-0.1401
0.1993	-0.2854	-0.2114
-0.4320	0.2143	-0.1053
-0.7481	-0.3840	-0.2408
0.8078	-0.1196	-0.2631
-0.7605	-0.5792	-0.1936
0.3237	0.7970	0.2170
1.3089	0.5786	-0.1887
1.2323	1.4421	0.4506



定向和平移相机



恢复相机中心

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$w\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}\alpha & s & u_0\\0 & \beta & v_0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & r_{13} & \boxed{t_x}\\r_{21} & r_{22} & r_{23} & t_y\\r_{31} & r_{32} & r_{33} & t_z\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

这不是相机中心-C。
它是 –RC
（因为在添加 t_x, t_y, 和 t_z之前将对一个点进行旋转）

$$\begin{bmatrix}su\\sv\\s\end{bmatrix} = \begin{bmatrix}\boxed{\begin{matrix}* & * & *\\ * & * & *\\ * & * & *\end{matrix}} & \boxed{\begin{matrix}*\\ *\\ *\end{matrix}}\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

Q

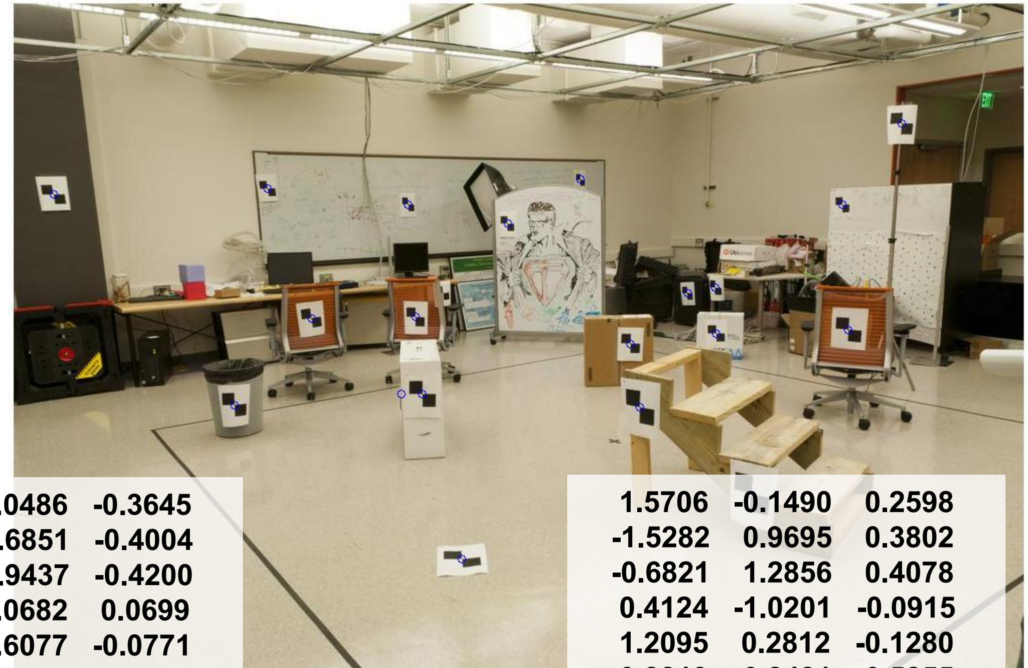
这里, m_4, 是 K * t

所以 $K^{-1} m_4$ 是 t

所以我们需要
$-R^{-1} K^{-1} m_4$ 来得到C

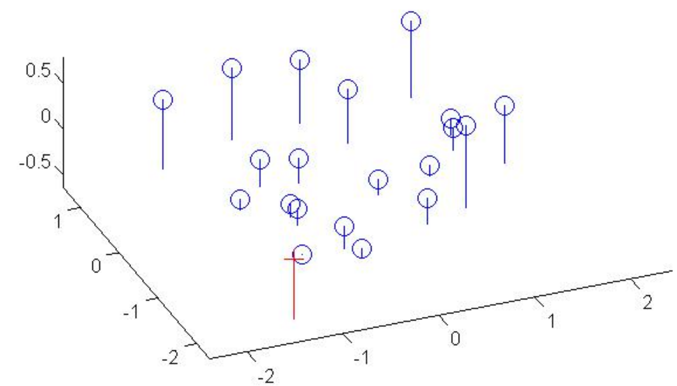
Q 是 K * R.所以我们只需要$-Q^{-1} m_4$

相机中心估算



1.0486	-0.3645
-1.6851	-0.4004
-0.9437	-0.4200
1.0682	0.0699
0.6077	-0.0771
1.2543	-0.6454
-0.2709	0.8635
-0.4571	-0.3645
-0.7902	0.0307
0.7318	0.6382
-1.0580	0.3312
0.3464	0.3377
0.3137	0.1189
-0.4310	0.0242
-0.4799	0.2920
0.6109	0.0830
-0.4081	0.2920
-0.1109	-0.2992
0.5129	-0.0575
0.1406	-0.4527

1.5706	-0.1490	0.2598
-1.5282	0.9695	0.3802
-0.6821	1.2856	0.4078
0.4124	-1.0201	-0.0915
1.2095	0.2812	-0.1280
0.8819	-0.8481	0.5255
-0.9442	-1.1583	-0.3759
0.0415	1.3445	0.3240
-0.7975	0.3017	-0.0826
-0.4329	-1.4151	-0.2774
-1.1475	-0.0772	-0.2667
-0.5149	-1.1784	-0.1401
0.1993	-0.2854	-0.2114
-0.4320	0.2143	-0.1053
-0.7481	-0.3840	-0.2408
0.8078	-0.1196	-0.2631
-0.7605	-0.5792	-0.1936
0.3237	0.7970	0.2170
1.3089	0.5786	-0.1887
1.2323	1.4421	0.4506

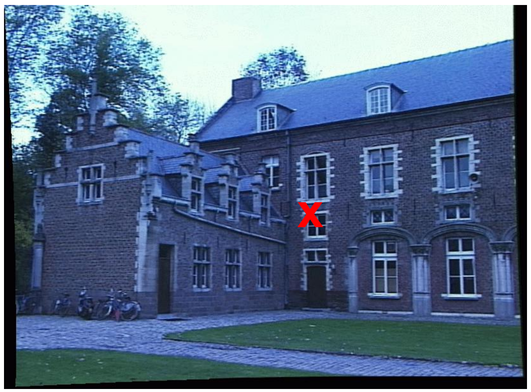


Outline

1 相机标定

2 对极几何

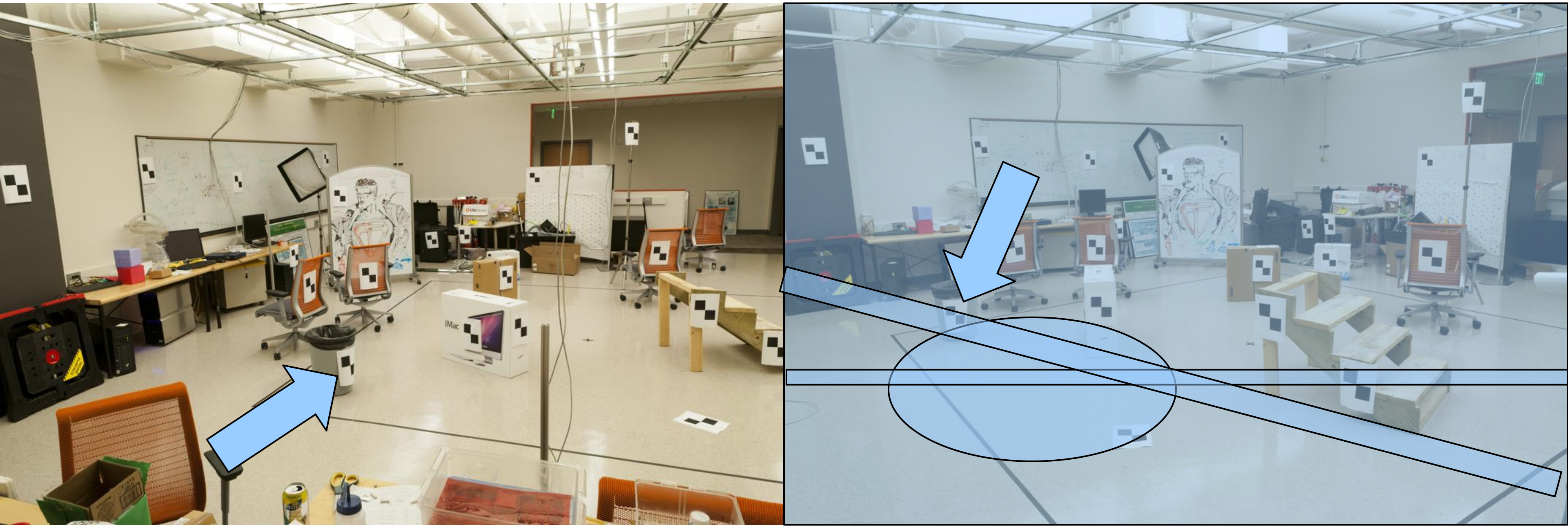
对应



- 我们有两张由具有不同内在和外在参数的相机拍摄的图像。

- 我们如何将第一幅图像中的一个点与第二幅图像中的一个点相匹配？ 我们怎样才能限制我们的搜索空间？

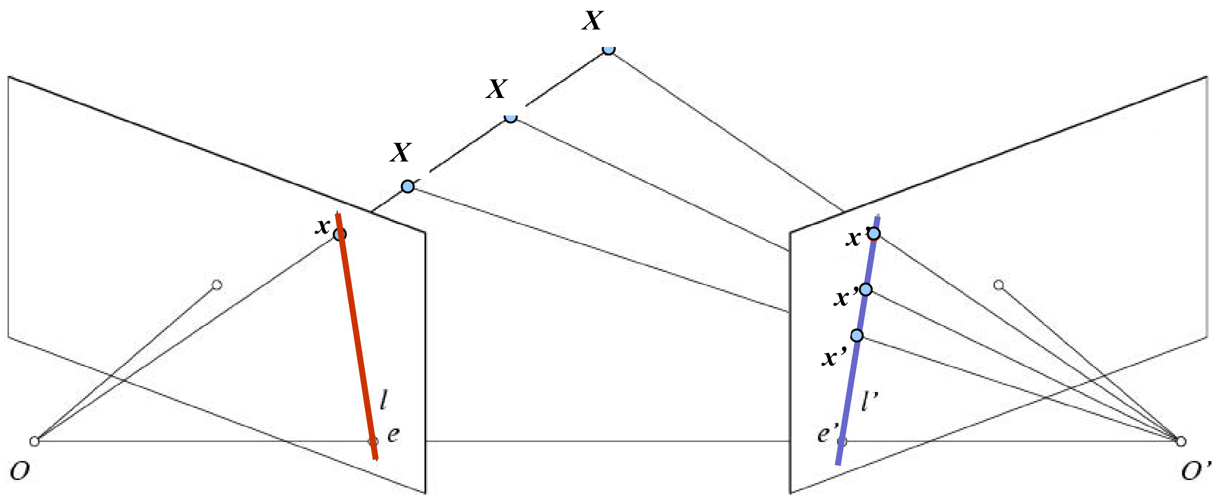
我们需要在哪里搜索？



Images from Lana Lazebnik

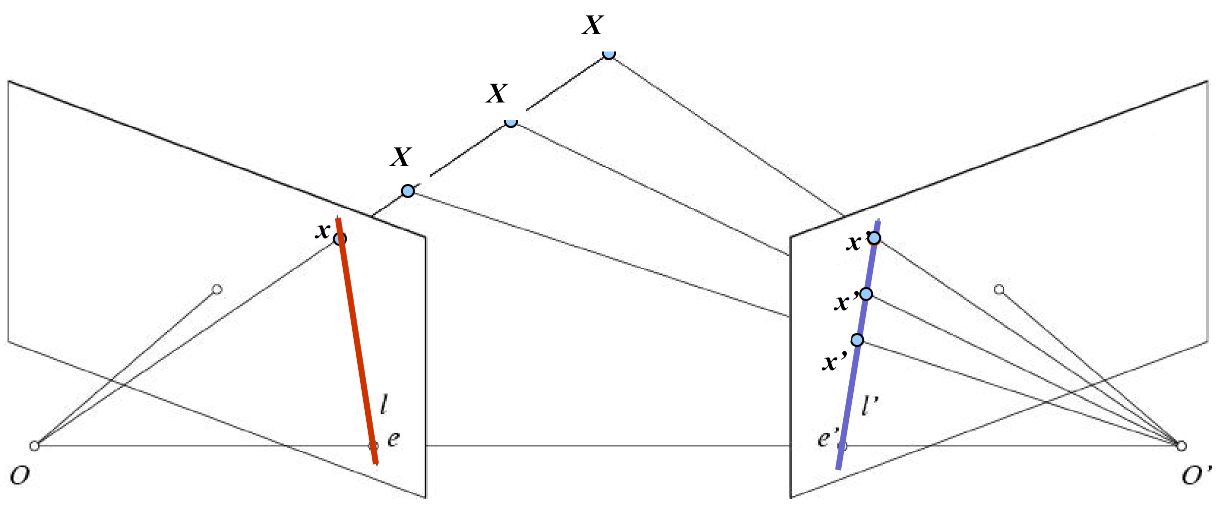
关键思想： 极线约束

极线约束



- x的潜在匹配点必须位于相应的直线l'上。
- x'的潜在匹配点必须位于相应的直线l上。

极线约束

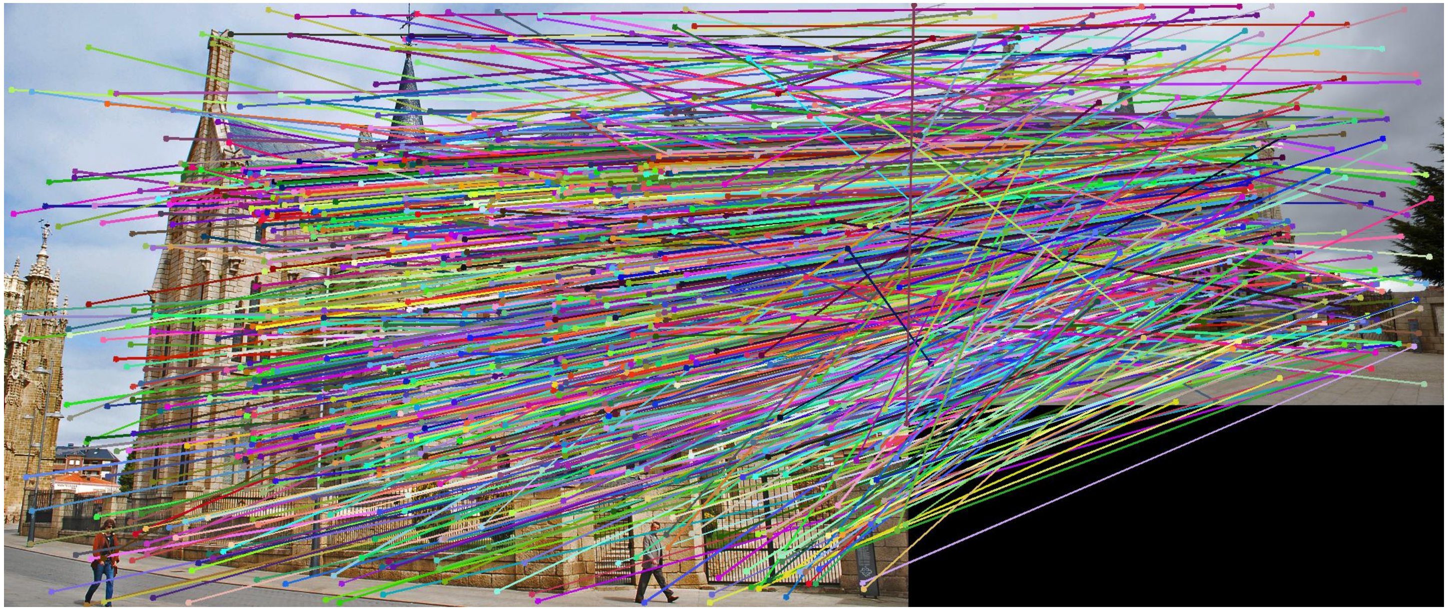


- x的潜在匹配点必须位于相应的直线l'上。
- x'的潜在匹配点必须位于相应的直线l上。

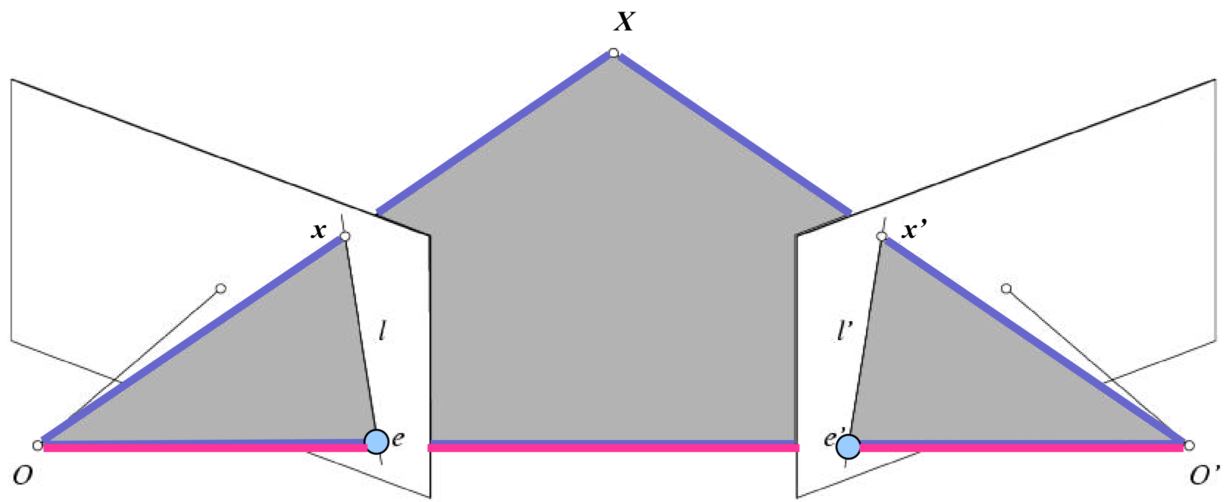
知道匹配点位于哪里，可以将二维空间的搜索限制在一维空间。

极线约束

- VLFeat在10,000多个本地特征点中选出800个置信度最高的匹配点。

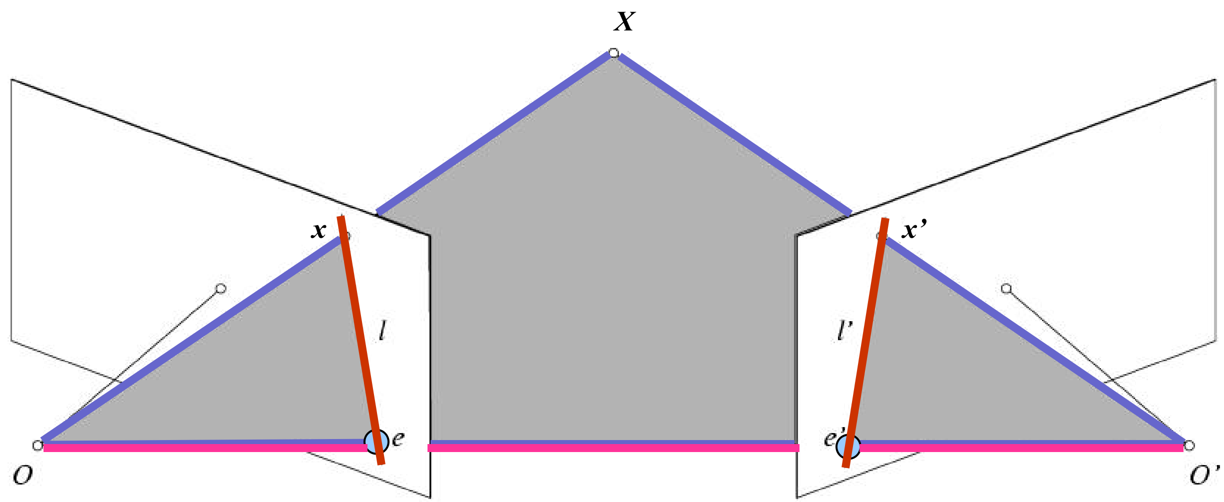


对极几何：符号



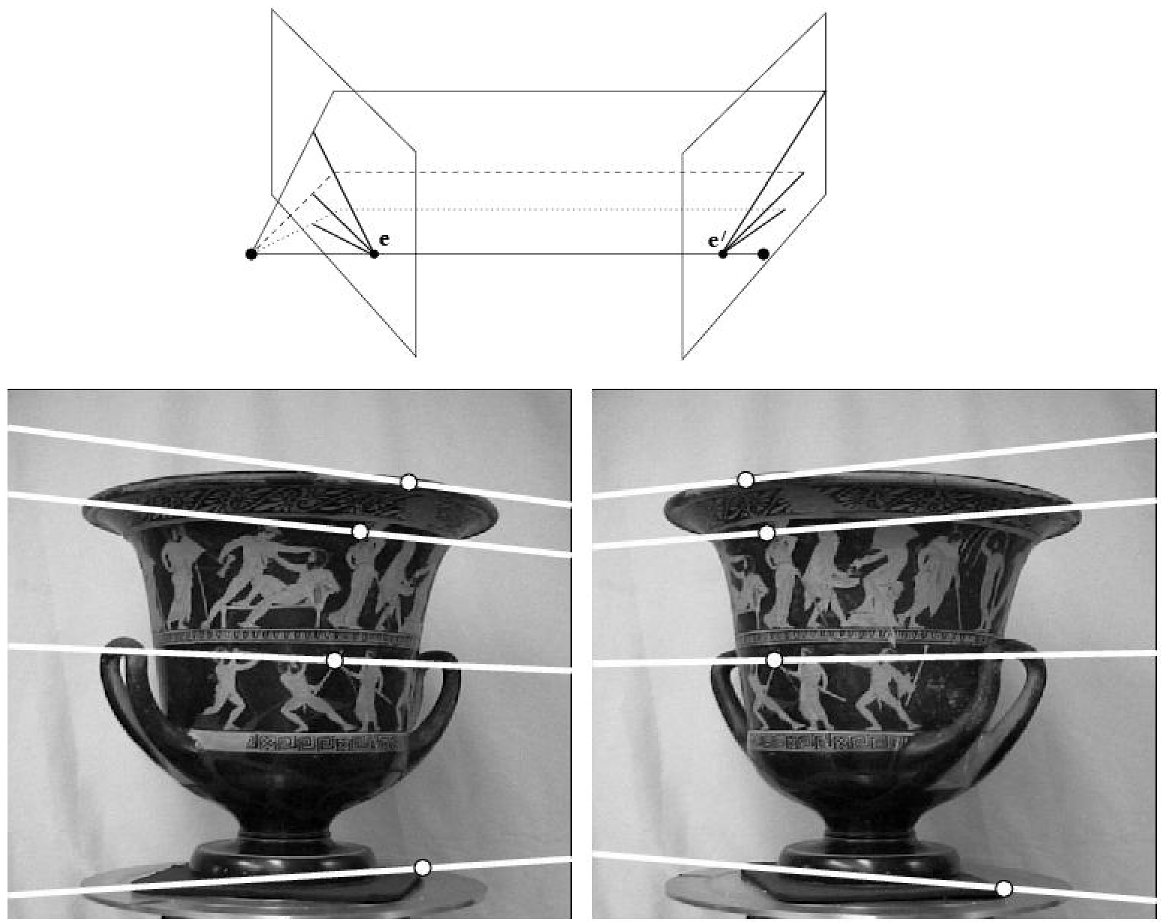
- **基线**：连接两个相机中心的直线
- **极点**：
  - = 基线与图像平面的交点
  - = 另一个相机中心的投影
- **极平面**：包含基线和X的平面。

对极几何：符号

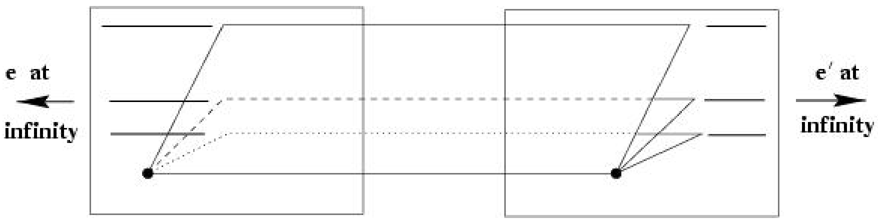


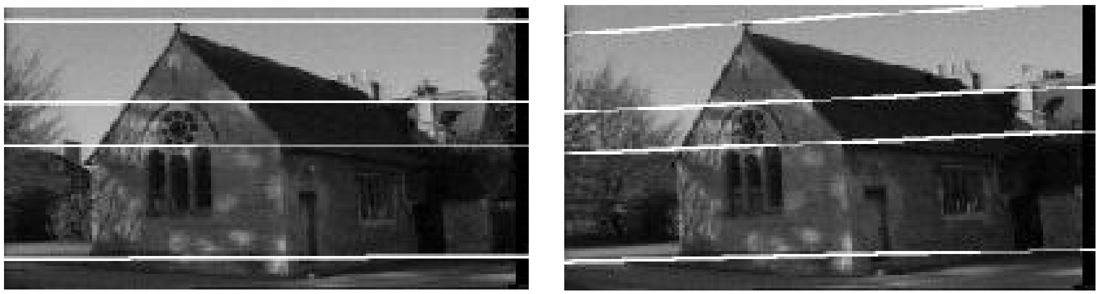
- **基线**：连接两个相机中心的直线
- **极点**：
  - = 基线与图像平面的交点
  - = 另一个相机中心的投影
- **极平面**：包含基线和X的平面。
- **极线**：极平面和图像平面的交线（成对出现）

示例：融合相机



示例：平行于图像平面的运动

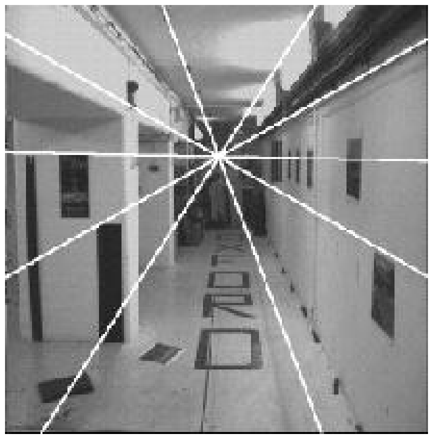


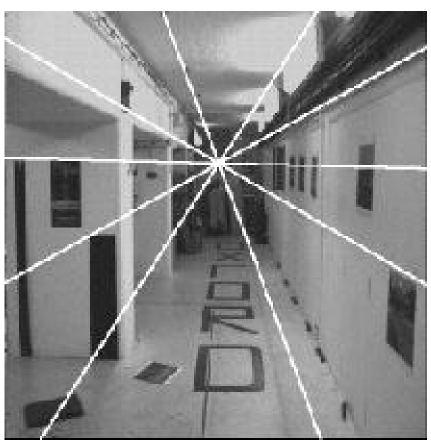


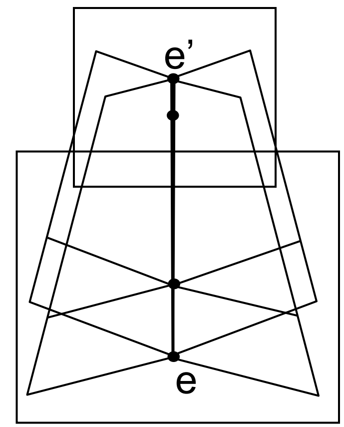
示例：前向运动

- 如果摄影机直接向前移动，极线会是什么样子？

示例：前向运动

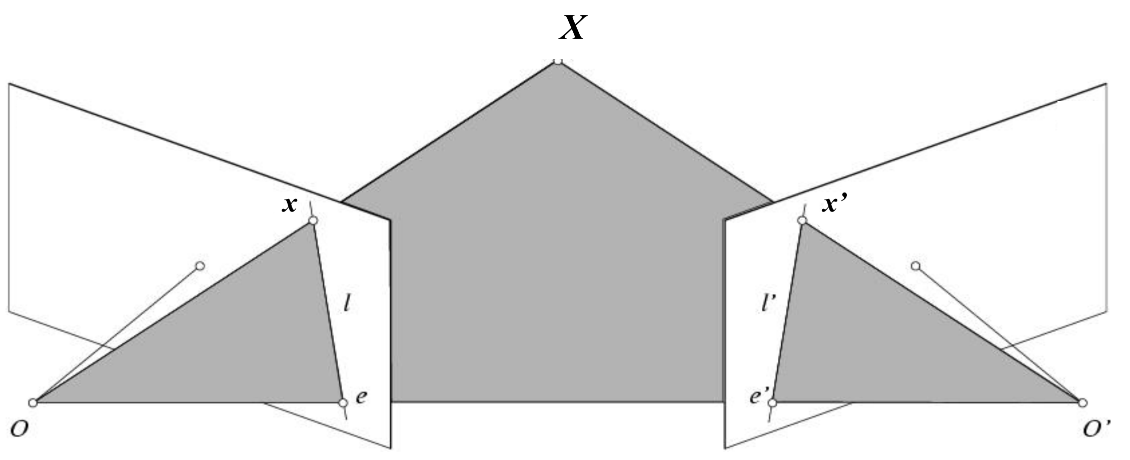






- 极点在两幅图像中具有相同的坐标。
- 其它点沿极点e的辐射线移动：“延伸焦点”

对极约束： 校准示例



给出摄像机的固有参数：

- 通过预先将所有的点与校准矩阵的倒数相乘来转换为归一化坐标；将第一台摄像机的坐标系设置为世界坐标

$$\hat{x} = K^{-1}x = X \qquad \hat{x}' = K'^{-1}x' = X'$$

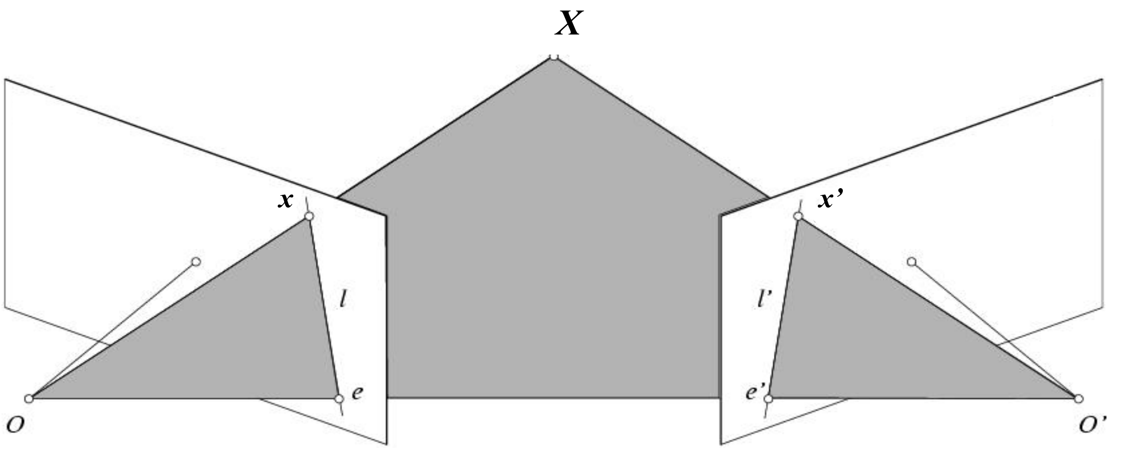
同质二维点（三维射线朝向X）。

二维像素坐标（同质性）。

三维场景点

第二台摄像机的三维坐标中的三维场景点

对极约束： 校准示例



给出摄像机的固有参数：

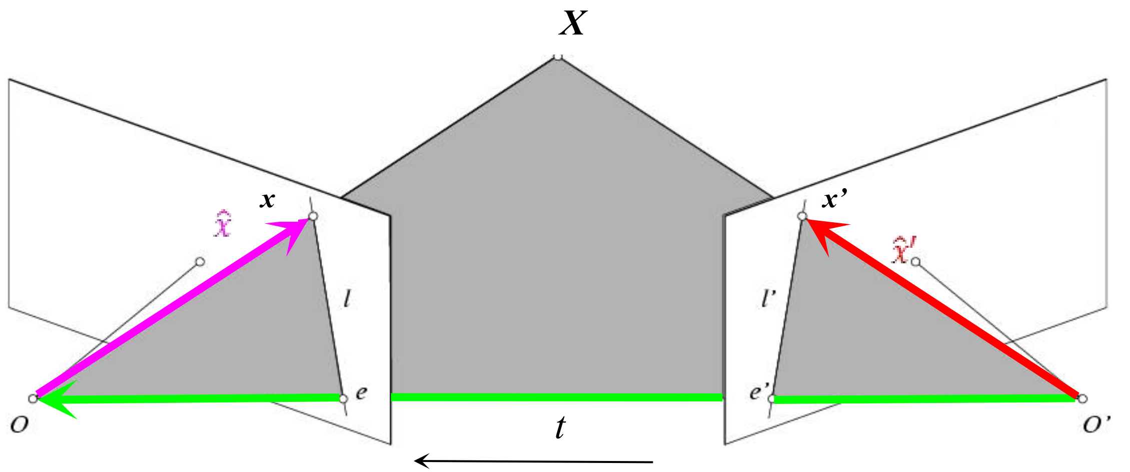
- 通过预先将所有的点与校准矩阵的倒数相乘来转换为归一化坐标；将第一台摄像机的坐标系设置为世界坐标
- 定义R和t，将X与X'联系起来，如下所示

某些比例因子

$$\hat{x} = K^{-1}x = X \qquad \hat{x}' = K'^{-1}x' = X'$$

$$\hat{x} = R\hat{x}' + t$$

对极约束：校准示例

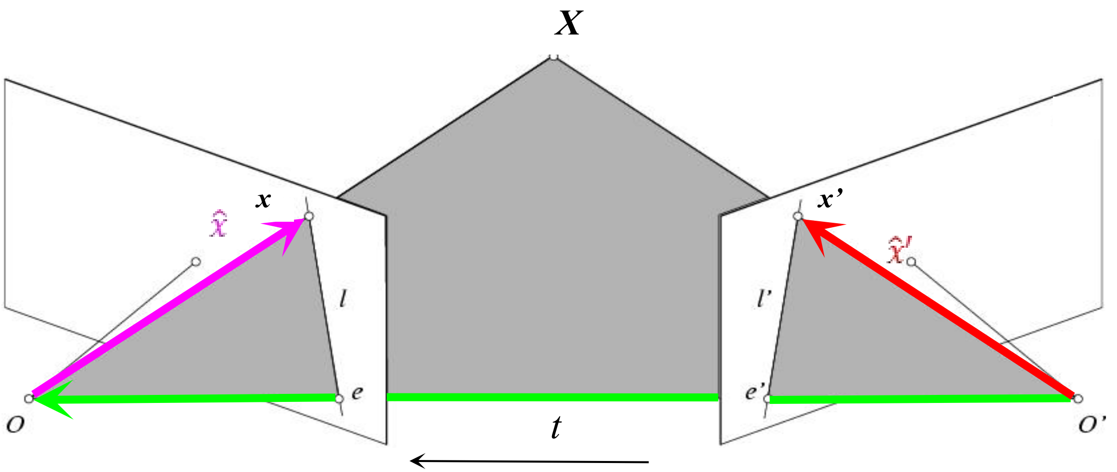


$$\hat{x} = K^{-1}x = X \qquad \hat{x}' = K'^{-1}x' = X'$$

$$\hat{x} = R\hat{x}' + t \quad \Rightarrow \quad \hat{x} \cdot [t \times (R\hat{x}')] = 0$$

(因为 $\hat{x}$, $R\hat{x}'$, 和 t 是共面的)

对极约束： 校准示例

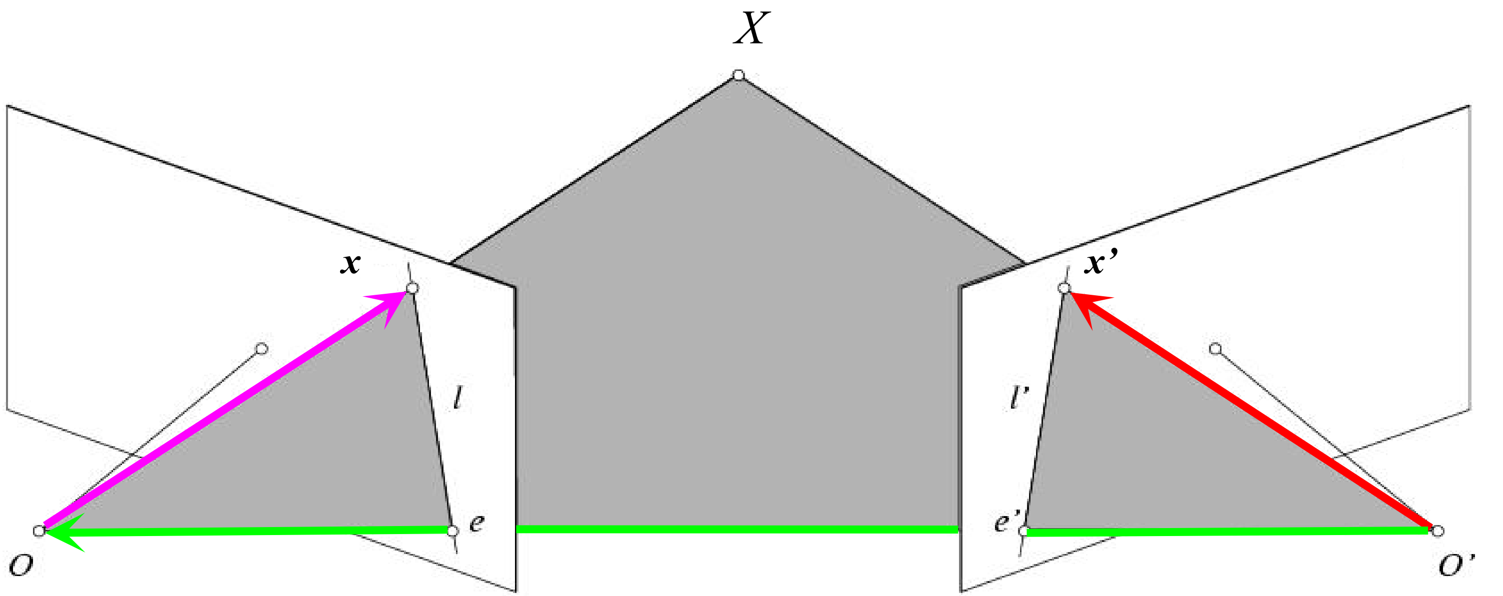


$$\hat{x} = K^{-1} x = X \qquad \hat{x}' = K'^{-1} x' = X'$$

$$\hat{x} \cdot [t \times (R\hat{x}')] = 0$$

(因为 $\hat{x}$, $R\hat{x}'$, 和 t 是共面的)

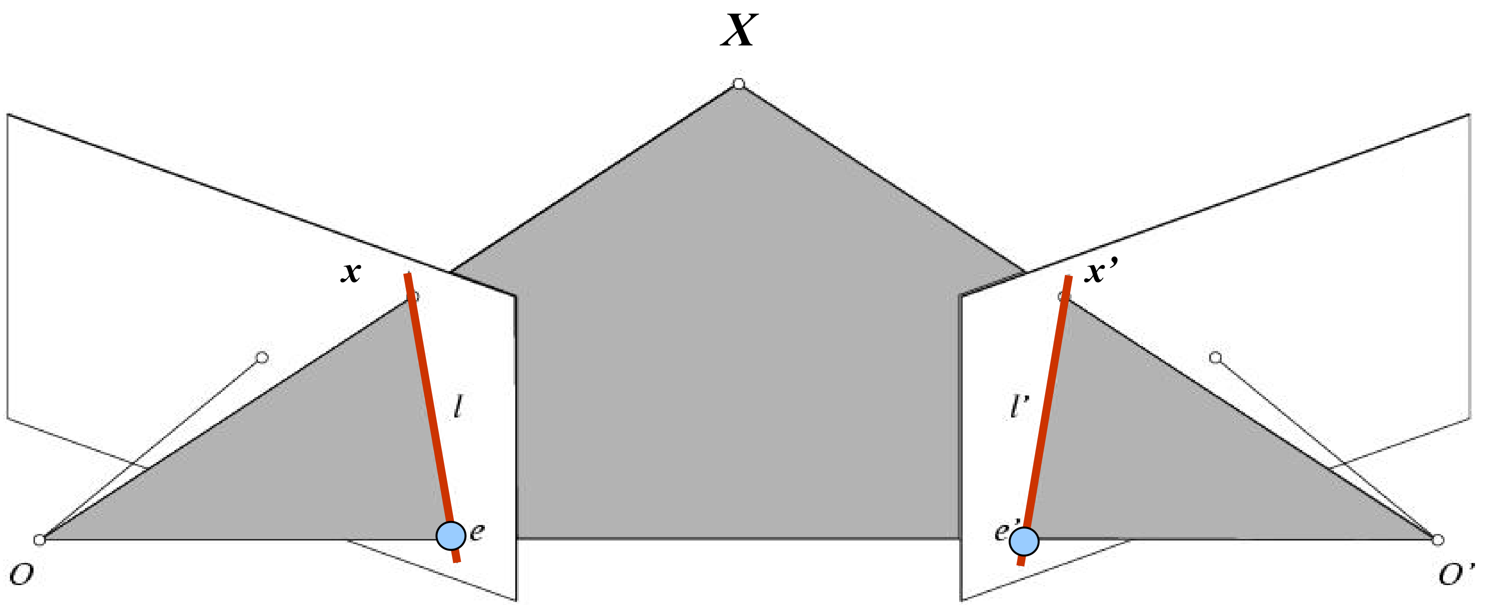
本征矩阵



$$\hat{x} \cdot [t \times (R\hat{x}')] = 0 \quad \Rightarrow \quad \hat{x}^T E \hat{x}' = 0 \quad \text{with} \quad E = [t]_\times R$$

本征矩阵
(Longuet-Higgins, 1981)

本征矩阵的性质



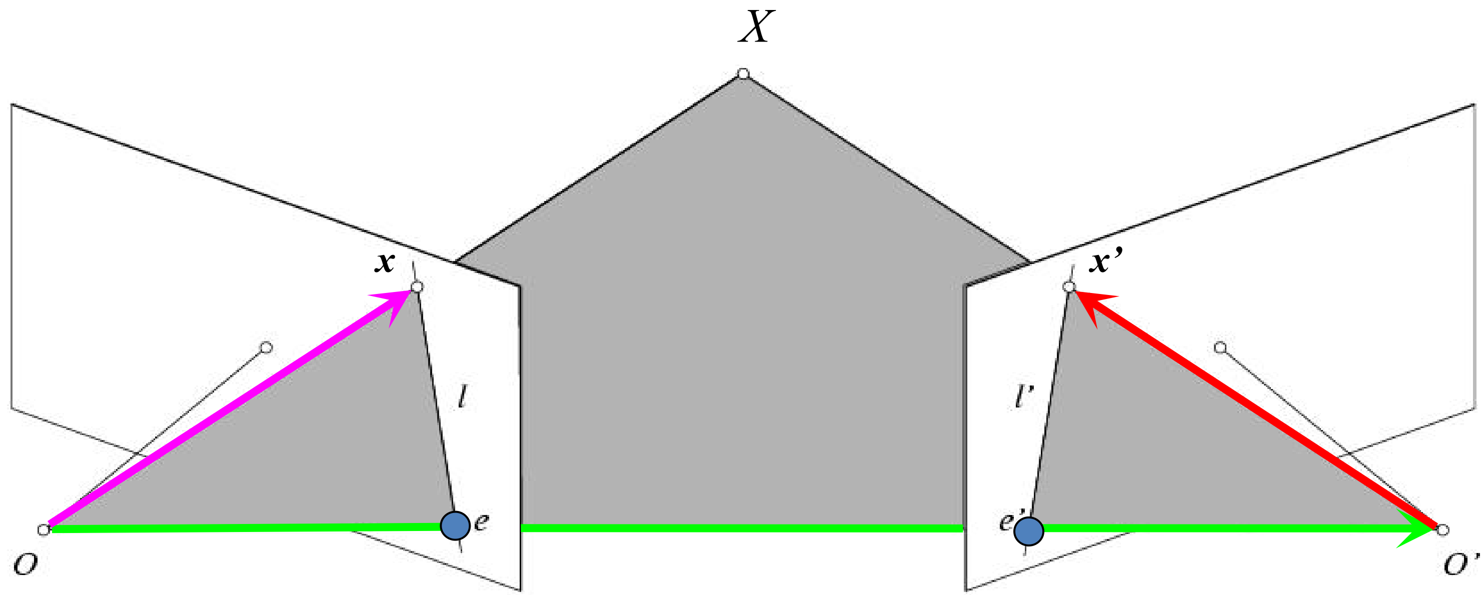
$$\hat{x} \cdot [t \times (R\hat{x}')] = 0 \Rightarrow \hat{x}^T E \hat{x}' = 0 \quad \text{with} \quad E = [t]_\times R$$

为了简化在下面去掉了 ^符号

反对称矩阵

- *E* x' 是与x' (*l* = *E* x') 相关的极线
- E^Tx是与x (*l'* = E^Tx) 相关的极线
- *E e'* = 0　和　*E^Te = 0*
- *E* 秩为2的单位矩阵
- *E* 有5个自由度
  - (R有3个, t有2个， 由于对极约束中的尺度等价性)

对极约束：非校准示例



- 如果我们不知道K和K '，那么我们可以用未知的归一化坐标来表示极坐标约束:

$$\hat{x}^T E \hat{x}' = 0 \qquad x = K\hat{x}, \quad x' = K'\hat{x}'$$

基本矩阵

在不知道K和K '的情况下，我们可以用未知的归一化坐标定义一个类似的关系

$$\hat{x}^T E \hat{x}' = 0$$

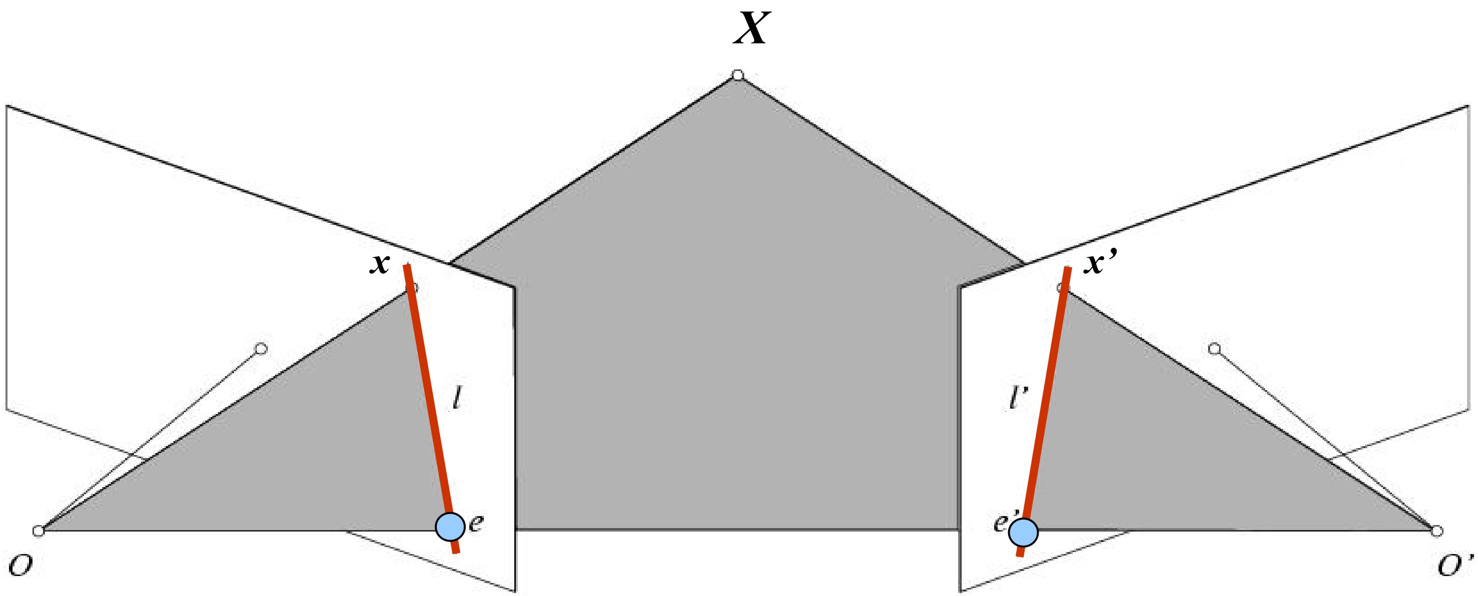
$$\hat{x} = K^{-1} x$$

$$\hat{x}' = K'^{-1} x'$$

$$x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

基础矩阵
(Faugeras and Luong, 1992)

基本矩阵的性质



$$x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

- ***F x'* = 0 是*x'* 的极线**
- ***F^T^x* = 0 是*x* 的极线**
- ***F e'* = 0　和　*F^T^e = 0***
- ***F*是秩为2的单位矩阵: det(F)=0**
- ***F* 有7个自由度: 9个参数，由于尺度等价性, det(F)=0**

基本矩阵的估计

- 8点算法
  - 用奇异值分解法求解8对对应方程的最小二乘
  - 在F上使用SVD强制det(F)=0约束
- 7点算法
  - 利用SVD和7对对应关系，用最小二乘法求解零空间（两个向量）
  - 求满足det(F)=0的零空间向量的线性组合
- 最小化重投影误差
  - 非线性最小平方法
  - 注:对于平面场景，F(或E)的估计是退化的

8点算法

- 解一个齐次线性方程组
  - 写出方程组

$$\mathbf{x}^T F \mathbf{x}' = 0$$

$$uu'f_{11} + uv'f_{12} + uf_{13} + vu'f_{21} + vv'f_{22} + vf_{23} + u'f_{31} + v'f_{32} + f_{33} = 0$$

$$\mathrm{A}\boldsymbol{f} = \begin{bmatrix} u_1 u_1{}' & u_1 v_1{}' & u_1 & v_1 u_1{}' & v_1 v_1{}' & v_1 & u_1{}' & v_1{}' & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_n u_v' & u_n v_n{}' & u_n & v_n u_n{}' & v_n v_n{}' & v_n & u_n{}' & v_n{}' & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ \vdots \\ f_{33} \end{bmatrix} = \mathbf{0}$$

8点算法

- 解一个齐次线性方程组
  - 写出方程组
  - 用奇异值分解从A//=0解出//

Matlab:

```
[U, S, V] = svd(A);
f = V(:, end);
F = reshape(f, [3 3])';
```

对于 **python,** 请看

numpy.linalg.svd

需要强制奇异性约束

- 基础矩阵的秩为2：det(F)=0
  - 左边：不正确的**F**—极线不是同时的
  - 右边：极线来自正确的F**F**

8点算法

- 解一个齐次线性方程组
  - 写出方程组
  - 用奇异值分解从A// = 0解出//

Matlab:

```
[U, S, V] = svd(A);
f = V(:, end);
F = reshape(f, [3 3])';
```

- 使用SVD求解det(F) = 0约束

Matlab:

```
[U, S, V] = svd(F);
S(3,3) = 0;
F = U*S*V';
```

对于 python, 请看
numpy.linalg.svd

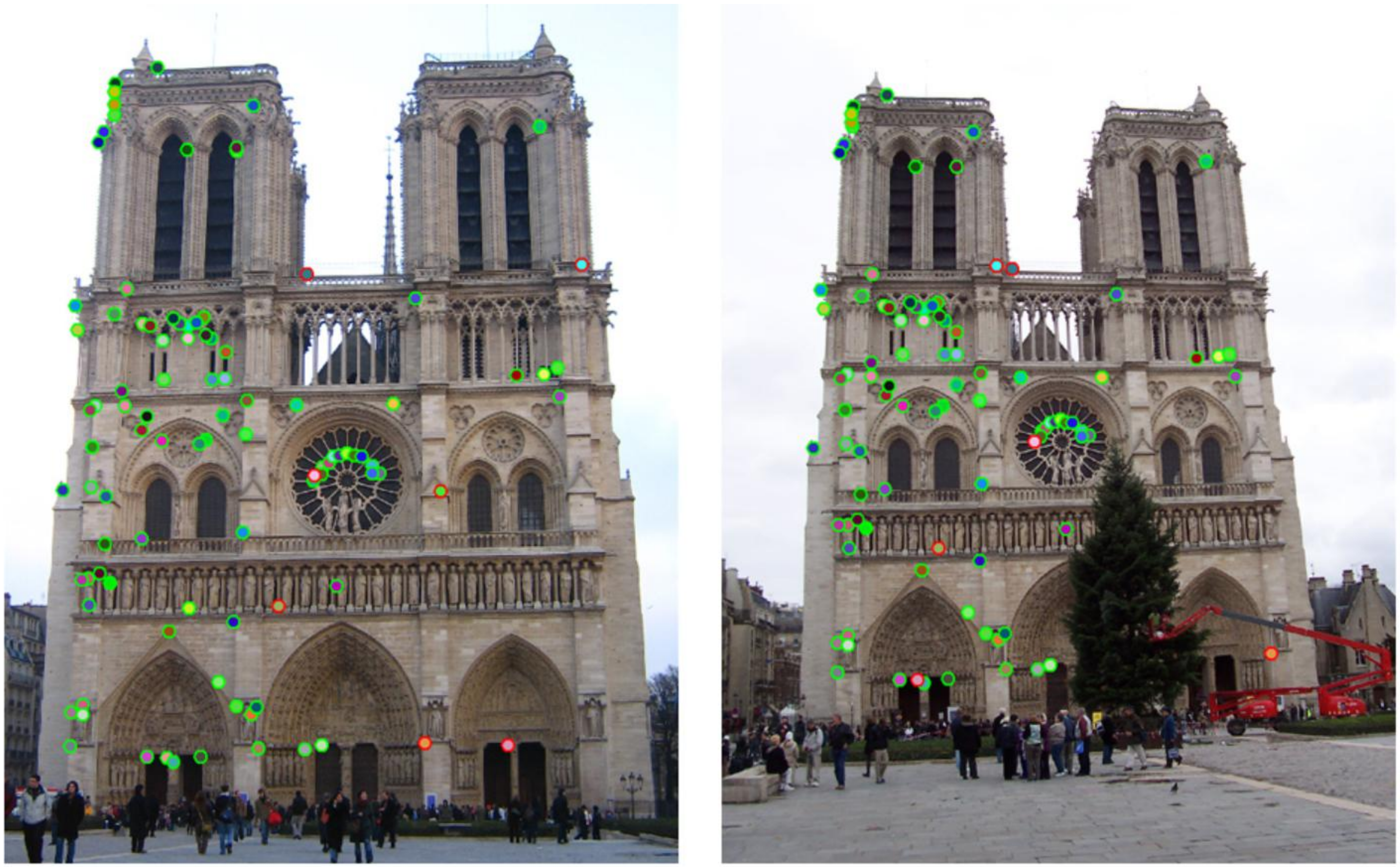
8点算法

- 解一个齐次线性方程组
  - 写出方程组
  - 用奇异值分解从A//=0解出//
- 使用SVD求解det(F) = 0约束
- 使用RANSAC处理异常值（采样8个点）

  如何检测异常值？

8点算法

- 如何检测异常值？



The top 100 most confident local feature matches from a baseline implementation of project 2. In this case, 93 were correct (highlighted in green) and 7 were incorrect (highlighted in red).

Project 2: Local Feature Matching

8点算法的问题

$$\begin{bmatrix} u'u & u'v & u' & v'u & v'v & v' & u & v \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \end{bmatrix} = -1$$

8点算法的问题

250906.36	183269.57	921.81	200931.10	146766.13	738.21	272.19	198.81
2692.28	131633.03	176.27	6196.73	302975.59	405.71	15.27	746.79
416374.23	871684.30	935.47	408110.89	854384.92	916.90	445.10	931.81
191183.60	171759.40	410.27	416435.62	374125.90	893.65	465.99	418.65
48988.86	30401.76	57.89	298604.57	185309.58	352.87	846.22	525.15
164786.04	546559.67	813.17	1998.37	6628.15	9.86	202.65	672.14
116407.01	2727.75	138.89	169941.27	3982.21	202.77	838.12	19.64
135384.58	75411.13	198.72	411350.03	229127.78	603.79	681.28	379.48

$$\begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \end{bmatrix} = -1$$

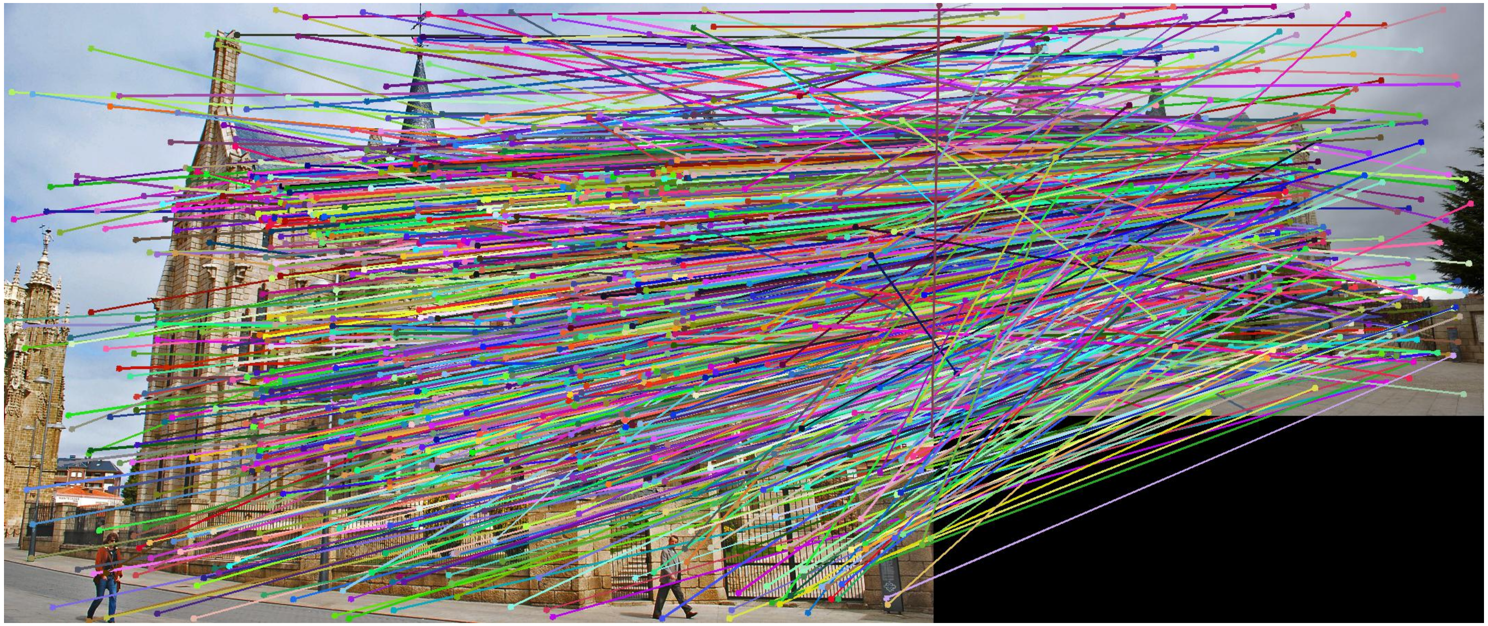
- 数值条件较差
- 通过缩放数据能解决吗？

归一化8点算法

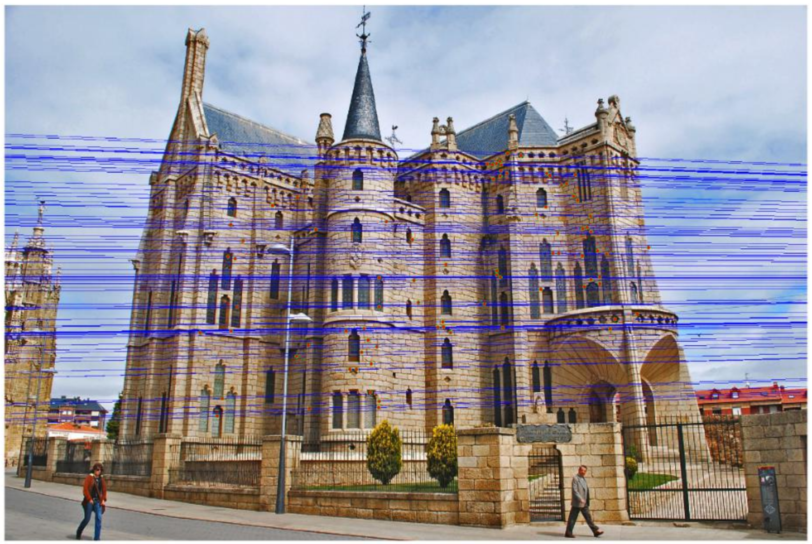
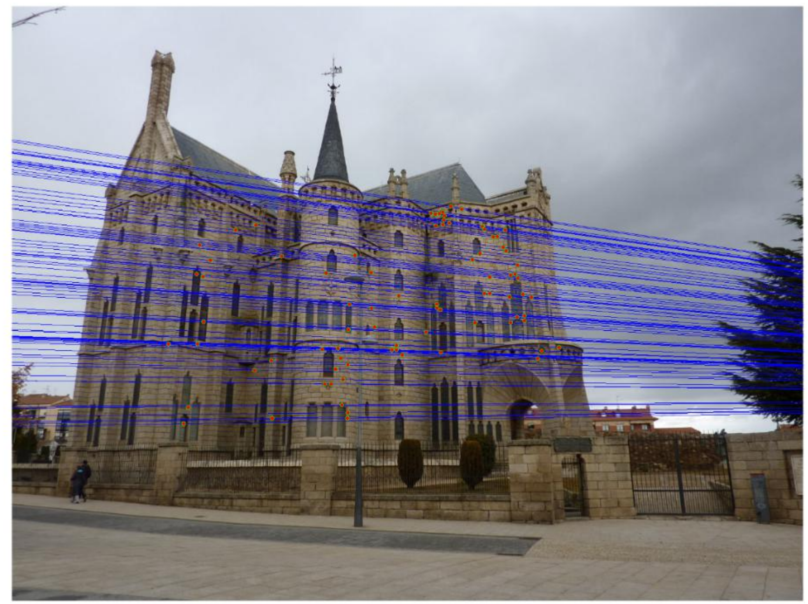
(Hartley, 1995)

- 将图像数据居中于原点，并对其进行缩放，使原点和数据点之间的平均平方距离为2像素
- 使用8点算法从归一化点计算F
- 强制rank-2约束(例如，取F的SVD并去掉最小的奇异值)
- 将基本矩阵变换回原始单位:如果T和T'是两个图像的归一化变换，则原始坐标下的基本矩阵为$T'^{T} F T$

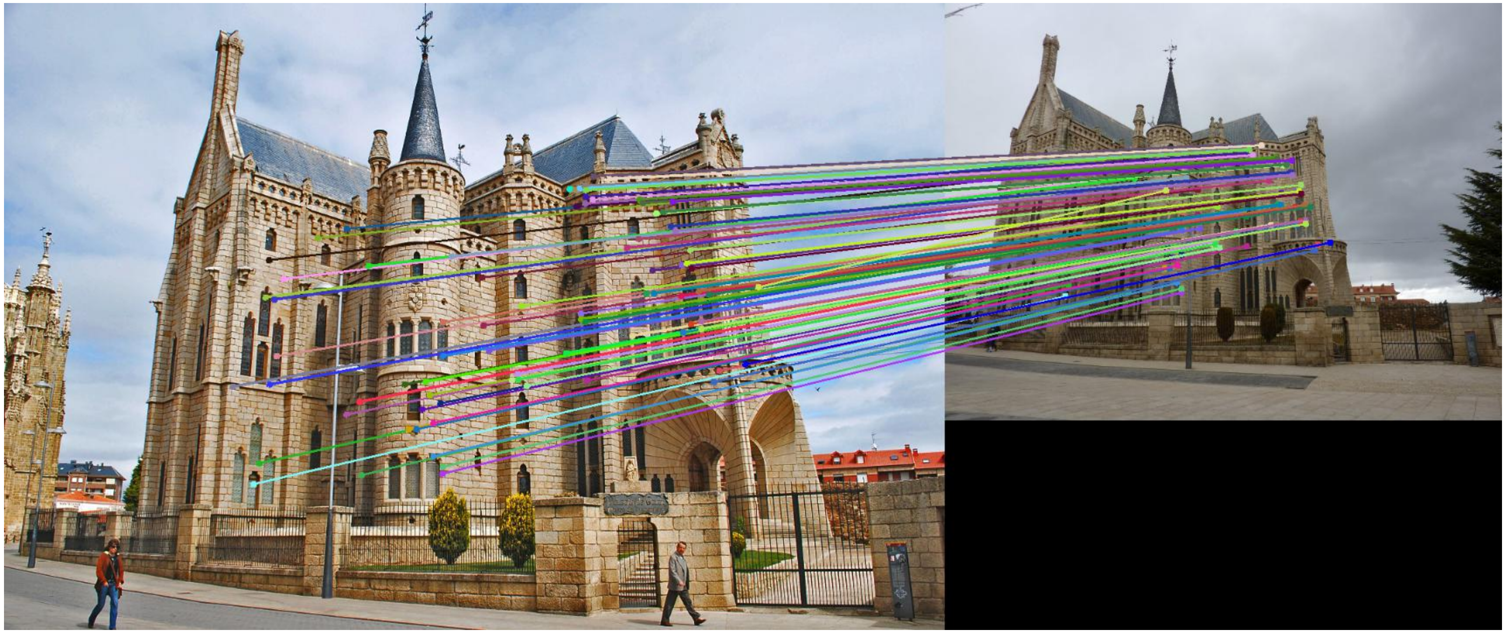
- 在10,000+个局部特征中，VLFeat的800个置信度最高匹配



极线

- 只保留相对于“最佳”基本矩阵的“内线”处的匹配



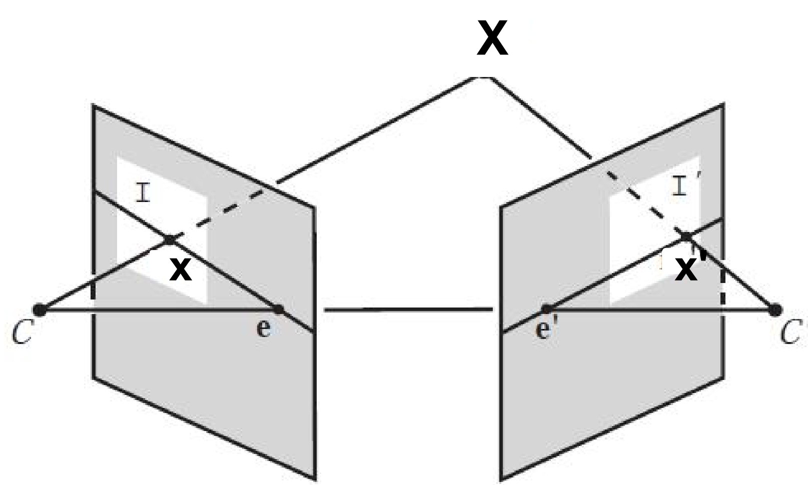
7点算法

- 从对应的7点中计算F
  - 形成7x9集方程$Af=0$
  - 系统有一个二维解集
  - 通用解（使用SVD）有形式 : $f= \lambda f_0+ \mu f_1$
  - 在矩阵术语中: $F= \lambda F_0+ \mu F_1$
  - 条件$\det F =0$ 给出三次方程中的λ 和μ
  - 对于比率λ ： μ ， 有1个或者3个解

更快(需要更少的点)并且可能更鲁棒(更少的点),
但也需要检查退化情况

金标准算法

- 使用8点算法得到F的初始值
- 用F求解P和P '(稍后讨论)
- 使重投影误差的平方最小，一起求解三维点X和F



参见算法11.2和算法11.3在HZ(页284–285)的详细信息

估计算法比较

	8点算法	归一化的8点算法	非线性最小二乘
Av. Dist. 1	2.33 pixels	0.92 pixel	0.86 pixel
Av. Dist. 2	2.18 pixels	0.85 pixel	0.80 pixel

- 我们可以得到投影矩阵P和P '直到一个投影歧义

$$\mathbf{P} = [\mathbf{I} \mid \mathbf{0}] \qquad \mathbf{P}' = [\underbrace{[\mathbf{e}']_{\times}}_{\text{K'*旋转}} \mathbf{F} \mid \underbrace{\mathbf{e}'}_{\text{K'*变换}}] \qquad \mathbf{e}'^{T}\mathbf{F} = 0$$

看 HZ p. 255-256

Code:

```
function P = vgg_P_from_F(F)
[U,S,V] = svd(F);
e = U(:,3);
P = [-vgg_contreps(e)*F e];
```

如果我们知道固有矩阵(K和K ')，我们可以解决歧义

从极几何到相机校准

- 估计基本矩阵被称为“弱校准”
- 如果我们知道两个相机的校准矩阵，我们可以估计基本矩阵:$E = K^T F K'$
- 基本矩阵给出了相机之间的相对旋转和平移，或者它们的外在参数

- Fundamental matrix song

谢谢！