



# 全景拼接

陶煜波，鲍虎军，章国锋  
浙江大学CAD&CG国家重点实验室

# 全景拼接

## ■ 单视点全景拼接

- 特征匹配
- 剔除outlier
- 单应矩阵求解
- 无缝融合 (Multi-band Blend, Graphcut, Poisson Image Editing)

## ■ 多视点全景拼接

- Dual-Homography
- APAP
- Long scene

# 单视点全景拼接

- 增大视域
  - 相机的视域

$$\text{FOV} = 50 \times 35^\circ$$



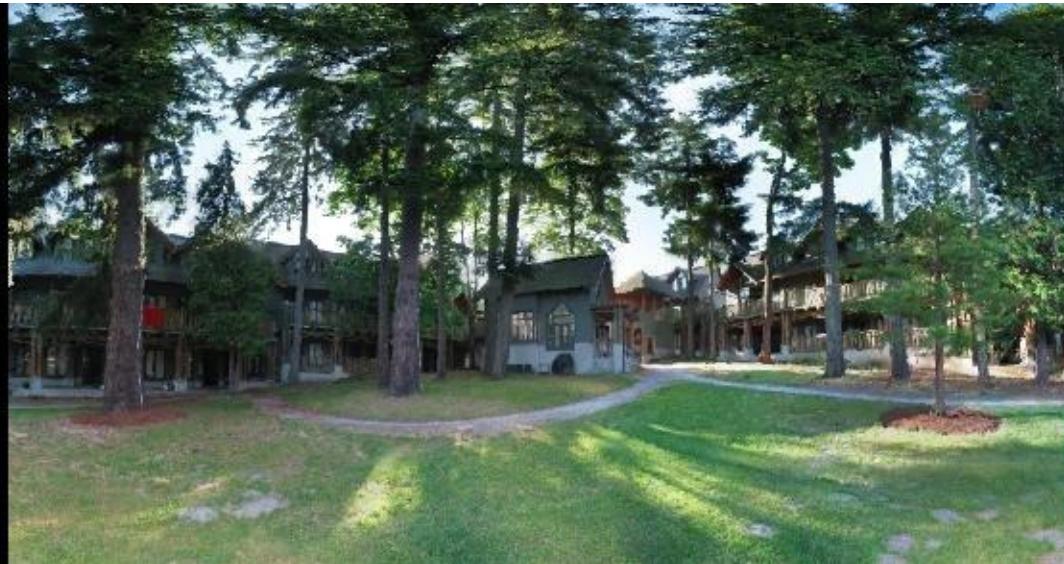
# 单视点全景拼接

## ■ 增大视域

- 相机的视域
- 人眼的视域

$\text{FOV} = 50 \times 35^\circ$

$\text{FOV} = 200 \times 135^\circ$



# 单视点全景拼接

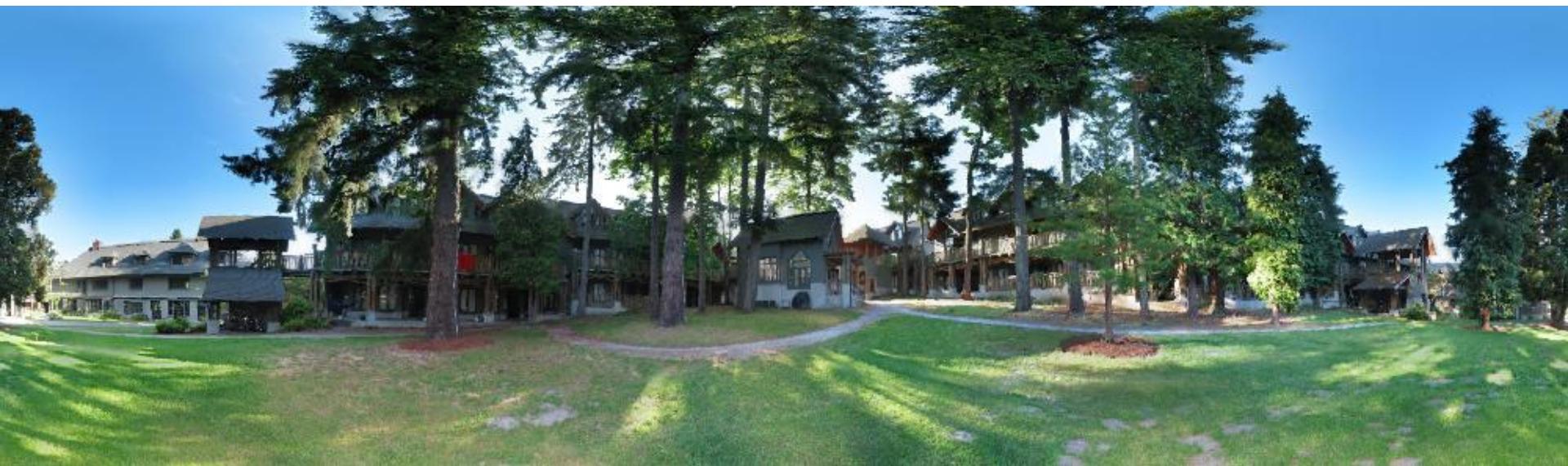
## ■ 增大视域

- 相机的视域
- 人眼的视域
- 全景图的视域

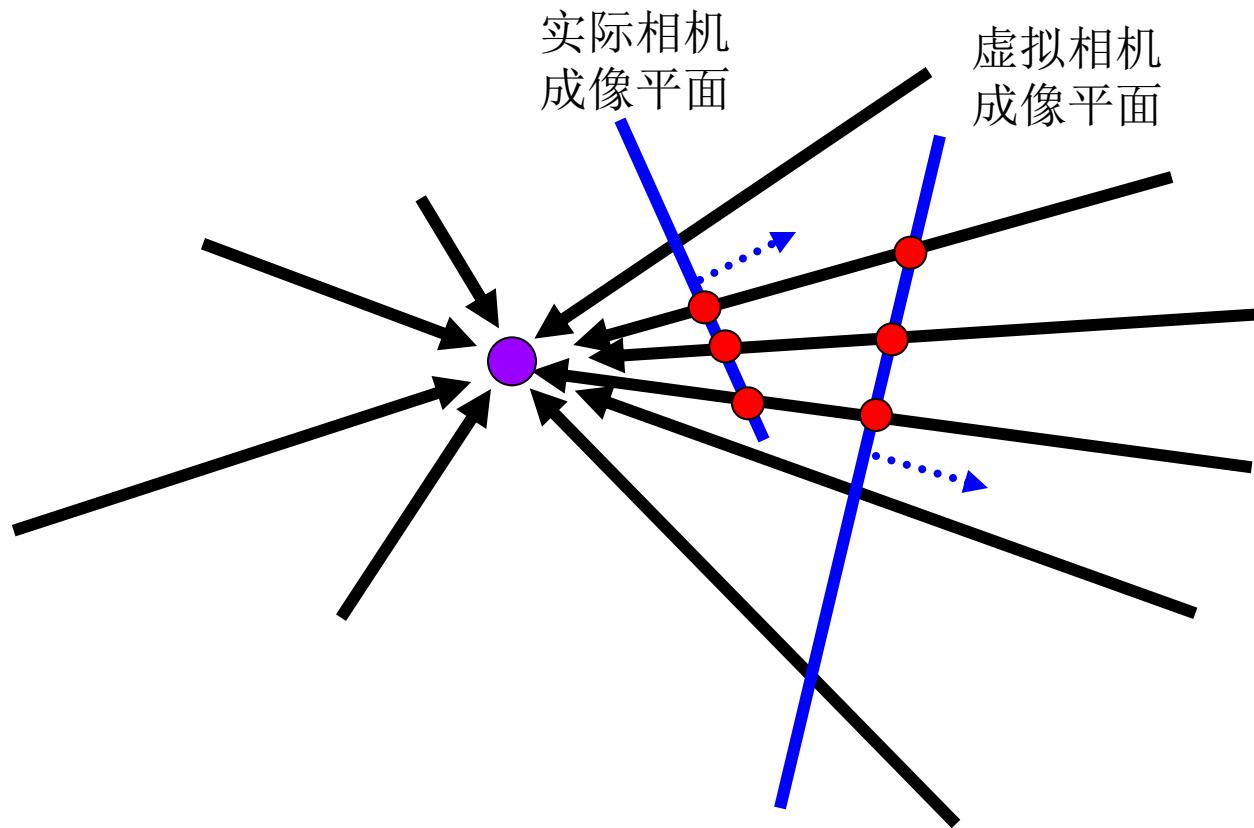
$\text{FOV} = 50 \times 35^\circ$

$\text{FOV} = 200 \times 135^\circ$

$\text{FOV} = 360 \times 180^\circ$



# 单视点全景拼接



# 单视点全景拼接



左图在上

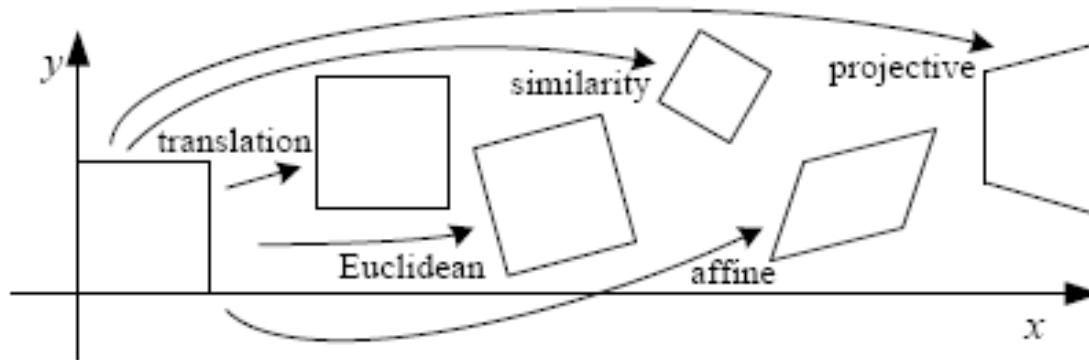
# 单视点全景拼接



右图在上

平移变换不足以解决图像的align问题

# 单视点全景拼接

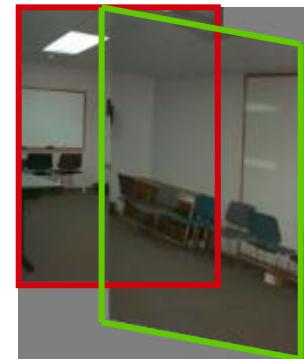


平移变换



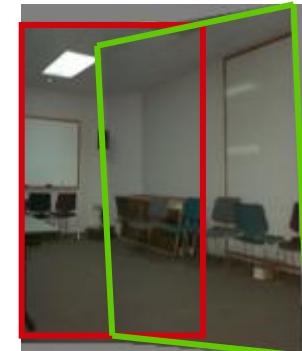
2自由度

仿射变换



6自由度

透视变换



8自由度

# 单视点全景拼接



虚拟的大视域相机视图

# 单视点全景拼接

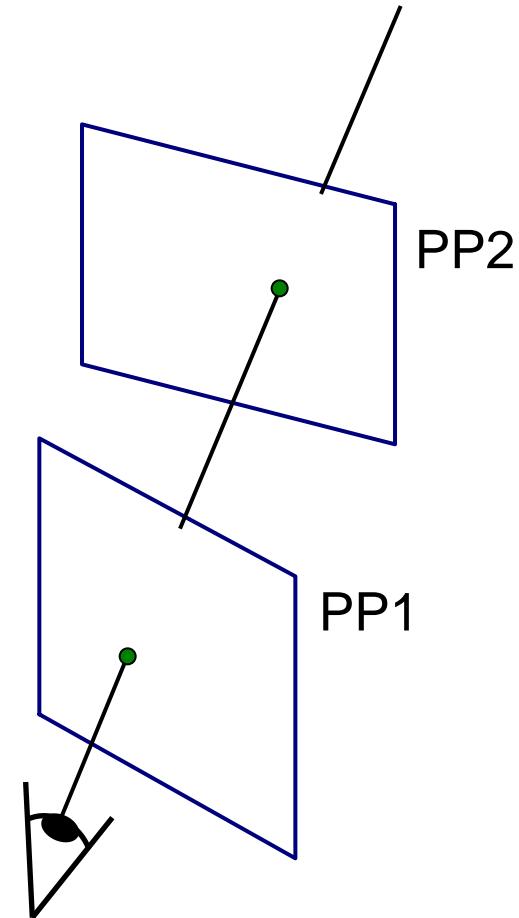
## ■ 透视变换

- 矩形可以被warp为任意四边形
- 保直线，不保平行

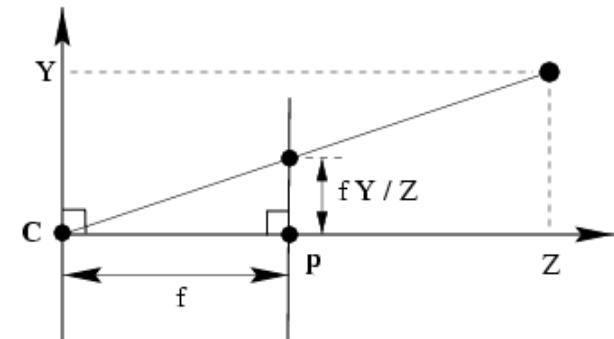
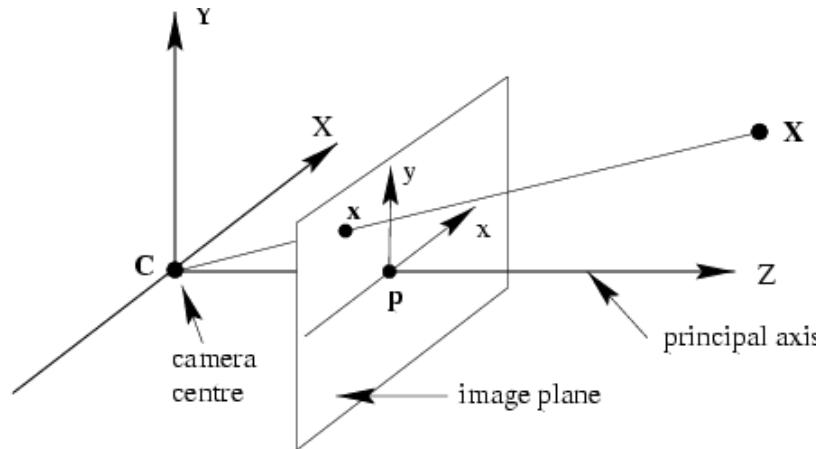
## ■ Homography

- 齐次坐标
- $3 \times 3, 8$ 个自由度

$$\begin{bmatrix} wx' \\ wy' \\ w \\ p' \end{bmatrix} = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \\ H \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \\ p \end{bmatrix}$$



# 单视点全景拼接



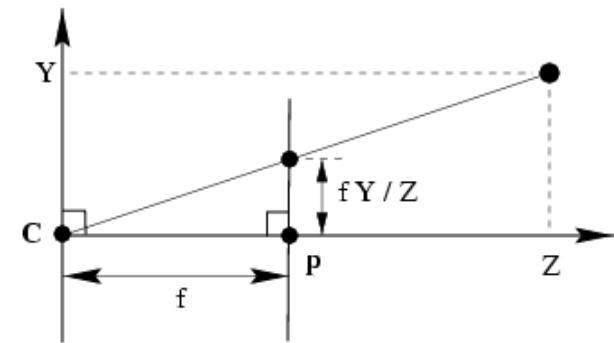
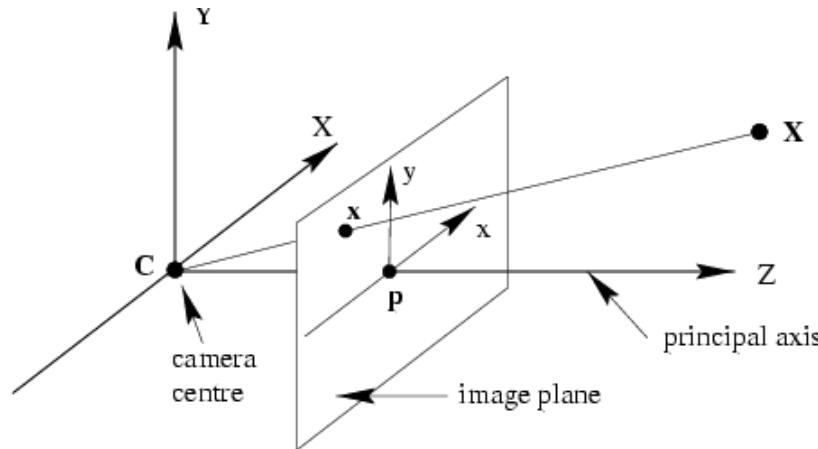
投影方程:

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

齐次坐标表示:

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

# 单视点全景拼接



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

$$K$$

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$[R|t]$$

# 单视点全景拼接

- $I_1, I_2$  两幅图上匹配的特征点  $p_1, p_2$
- 齐次坐标  $(x_1, y_1, 1), (x_2, y_2, 1)$
- 相差一个  $3 \times 3$  矩阵  $H$

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} \sim H_{10} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = \begin{pmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

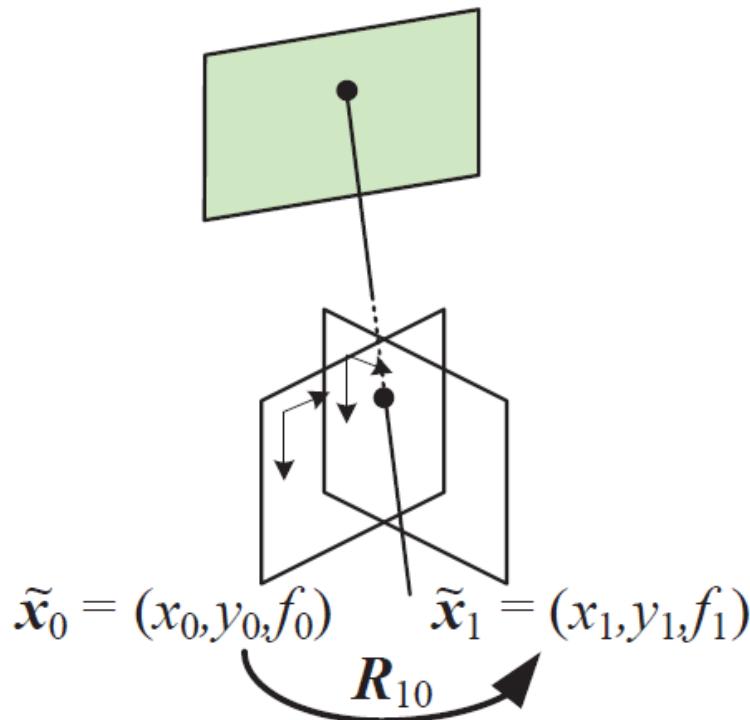
# 单视点全景拼接

## ■ 适用条件

- 相机纯旋转
- 平面性场景

# 单视点全景拼接

$$X = (X, Y, Z)^T$$



$$[R_0 \mid t_0] = [I \mid 0]$$

$$[R_1 \mid t_1] = [R_{10} \mid 0]$$

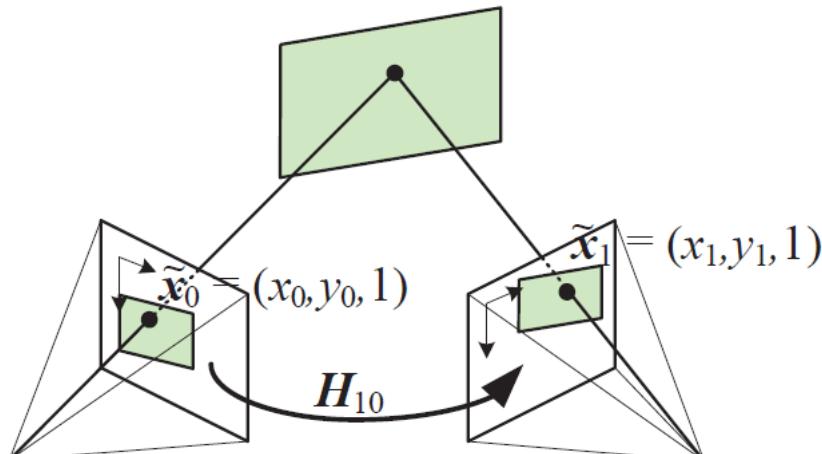
$$(x_0, y_0, 1)^T = K_0 (X, Y, Z)^T$$

$$(x_1, y_1, 1)^T = K_1 \cdot R_{10} \cdot (X, Y, Z)^T$$

$$\begin{aligned}(x_1, y_1, 1)^T &= (K_1 R_{10} K_0^{-1})(x_0, y_0, 1)^T \\ &= H_{10}(x_0, y_0, 1)^T\end{aligned}$$

# 单视点全景拼接

$$X = (X, Y, 0)^T$$



$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = K \begin{pmatrix} R \cdot \begin{pmatrix} X \\ Y \\ 0 \end{pmatrix} + t \end{pmatrix} = K \cdot (r_0, r_1, t) \cdot \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

$$\text{Homography : } H \equiv K \cdot (r_0, r_1, t)$$

$$(x_0, y_0, 1)^T = H_0(X, Y, 1)^T$$

$$(x_1, y_1, 1)^T = H_1(X, Y, 1)^T$$

$$(x_1, y_1, 1)^T = (H_1 H_0^{-1})(x_0, y_0, 1)^T$$

$$= H_{10}(x_0, y_0, 1)^T$$

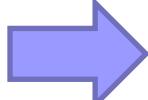
# 单视点全景拼接

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} \sim H_{10} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = \begin{pmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

# 单视点全景拼接

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} \sim H_{10} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = \begin{pmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} \quad \rightarrow \quad \begin{aligned} x_2 &= \frac{h_{00}x_1 + h_{01}y_1 + h_{02}}{h_{20}x_1 + h_{21}y_1 + h_{22}} \\ y_2 &= \frac{h_{10}x_1 + h_{11}y_1 + h_{12}}{h_{20}x_1 + h_{21}y_1 + h_{22}} \end{aligned}$$

# 单视点全景拼接

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} \sim H_{10} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = \begin{pmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$


$$x_2 = \frac{h_{00}x_1 + h_{01}y_1 + h_{02}}{h_{20}x_1 + h_{21}y_1 + h_{22}}$$
$$y_2 = \frac{h_{10}x_1 + h_{11}y_1 + h_{12}}{h_{20}x_1 + h_{21}y_1 + h_{22}}$$



$$x_1h_{00} + y_1h_{01} + h_{02} - x_1x_2h_{20} - y_1x_2h_{21} - x_2h_{22} = 0$$

$$x_1h_{10} + y_1h_{11} + h_{12} - x_1y_2h_{20} - y_1y_2h_{21} - y_2h_{22} = 0$$

# 单视点全景拼接

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} \sim H_{10} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = \begin{pmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} \quad \rightarrow \quad \begin{aligned} x_2 &= \frac{h_{00}x_1 + h_{01}y_1 + h_{02}}{h_{20}x_1 + h_{21}y_1 + h_{22}} \\ y_2 &= \frac{h_{10}x_1 + h_{11}y_1 + h_{12}}{h_{20}x_1 + h_{21}y_1 + h_{22}} \end{aligned}$$

$$x_1h_{00} + y_1h_{01} + h_{02} - x_1x_2h_{20} - y_1x_2h_{21} - x_2h_{22} = 0$$

$$x_1h_{10} + y_1h_{11} + h_{12} - x_1y_2h_{20} - y_1y_2h_{21} - y_2h_{22} = 0$$



$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x_2 & -y_1x_2 & -x_2 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y_2 & -y_1y_2 & -y_2 \end{pmatrix} \cdot \begin{pmatrix} h_{00} \\ h_{01} \\ \dots \\ h_{22} \end{pmatrix} p = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

# 单视点全景拼接

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} \sim H_{10} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = \begin{pmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} \rightarrow \begin{aligned} x_2 &= \frac{h_{00}x_1 + h_{01}y_1 + h_{02}}{h_{20}x_1 + h_{21}y_1 + h_{22}} \\ y_2 &= \frac{h_{10}x_1 + h_{11}y_1 + h_{12}}{h_{20}x_1 + h_{21}y_1 + h_{22}} \end{aligned}$$

$$x_1h_{00} + y_1h_{01} + h_{02} - x_1x_2h_{20} - y_1x_2h_{21} - x_2h_{22} = 0$$

$$x_1h_{10} + y_1h_{11} + h_{12} - x_1y_2h_{20} - y_1y_2h_{21} - y_2h_{22} = 0$$

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x_2 & -y_1x_2 & -x_2 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y_2 & -y_1y_2 & -y_2 \end{pmatrix} \cdot \begin{pmatrix} h_{00} \\ h_{01} \\ \dots \\ h_{22} \end{pmatrix} p = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

四对点决定一个Homography。SVD(奇异值分解), 求得最小特征值对应的特征向量, 就是解。

# 单视点全景拼接步骤

- 特征匹配
- 剔除outlier
- 求解单应矩阵
- 无缝融合

# 特征匹配

## ■ SIFT

- Scale Invariant Feature Transform
- 旋转，缩放不变性
- 尺度空间的极值点
- 128维的描述子
- Distinctive Image Features from Scale-Invariant Keypoints, IJCV'2004
- <http://cs.unc.edu/~ccwu/siftgpu/>

# 特征匹配

## ■ Affine SIFT

- <http://www.cmap.polytechnique.fr/~yu/research/ASIFT/demo.html>

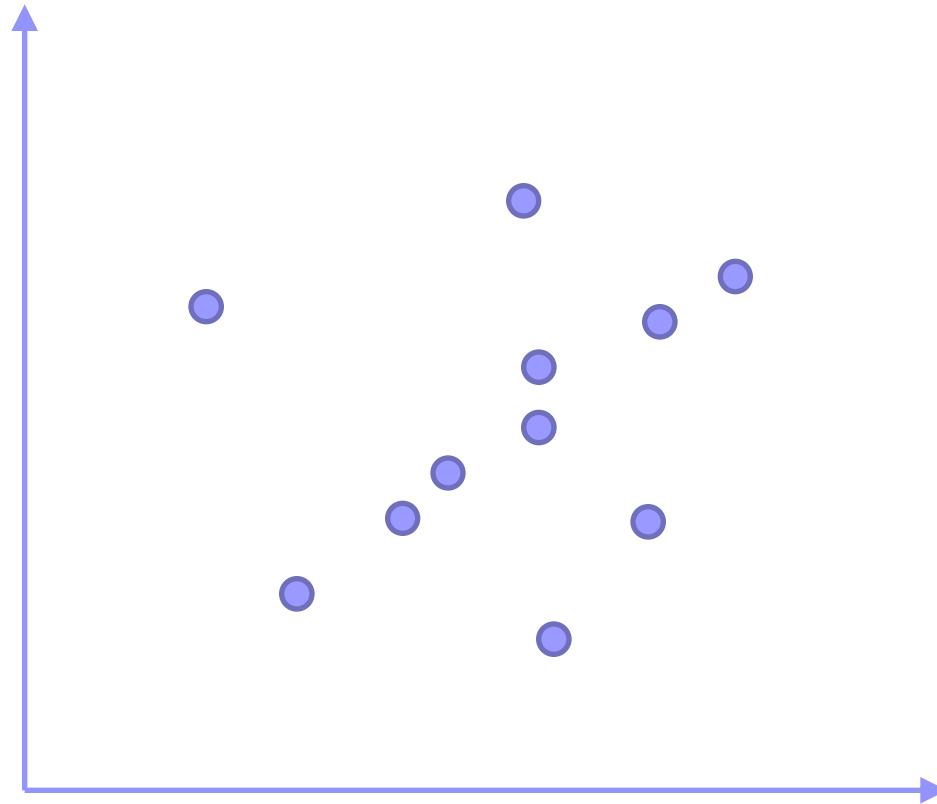
## ■ SURF

- OpenCV
- [http://docs.opencv.org/doc/tutorials/features2d/feature\\_detection/feature\\_detection.html](http://docs.opencv.org/doc/tutorials/features2d/feature_detection/feature_detection.html)

# RANSAC算法

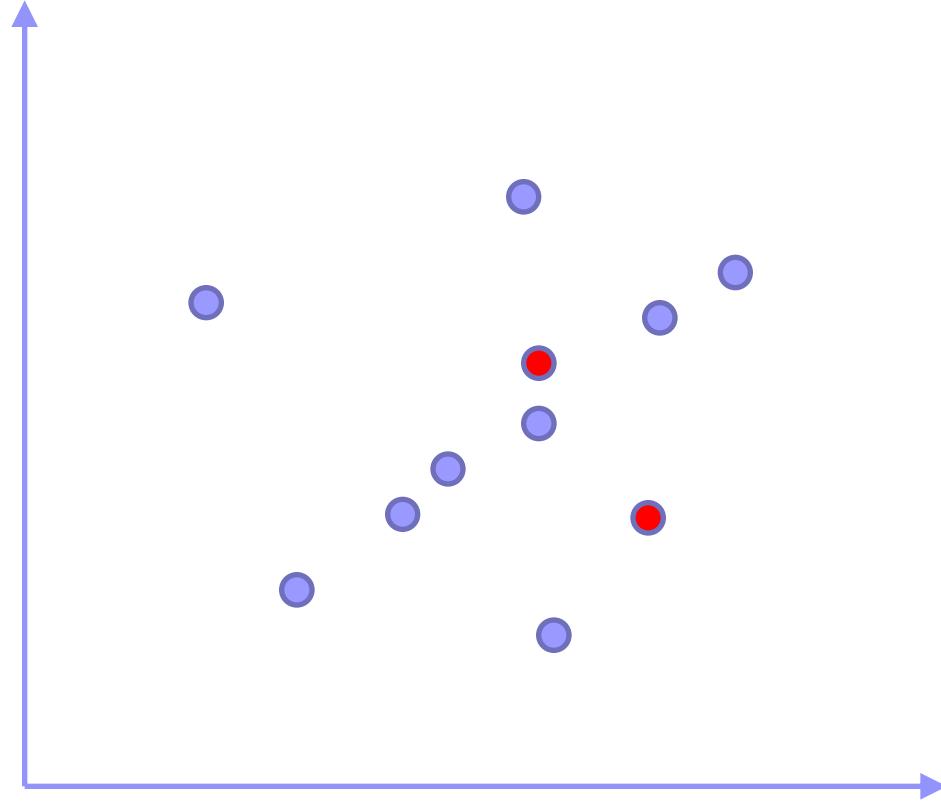
- 特征匹配结果可能存在outlier
- 需根据Homography的模型进行检验
- Random sample consensus
- 随机采样一致性算法

# RANSAC算法



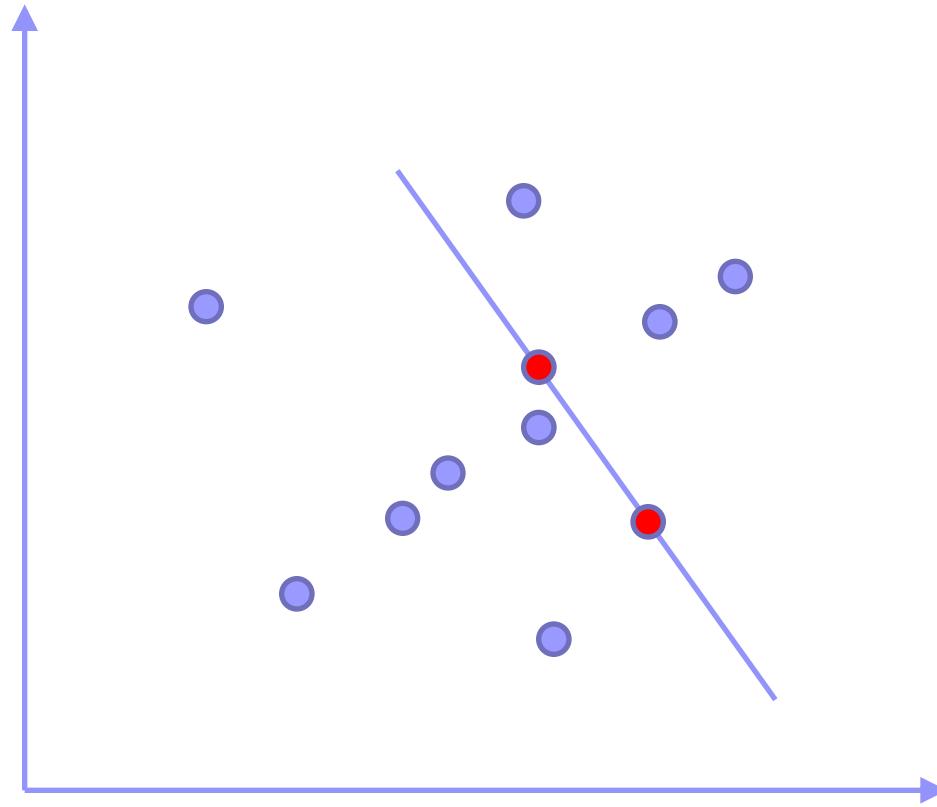
问题：找一条直线，拟合最多的点

# RANSAC算法



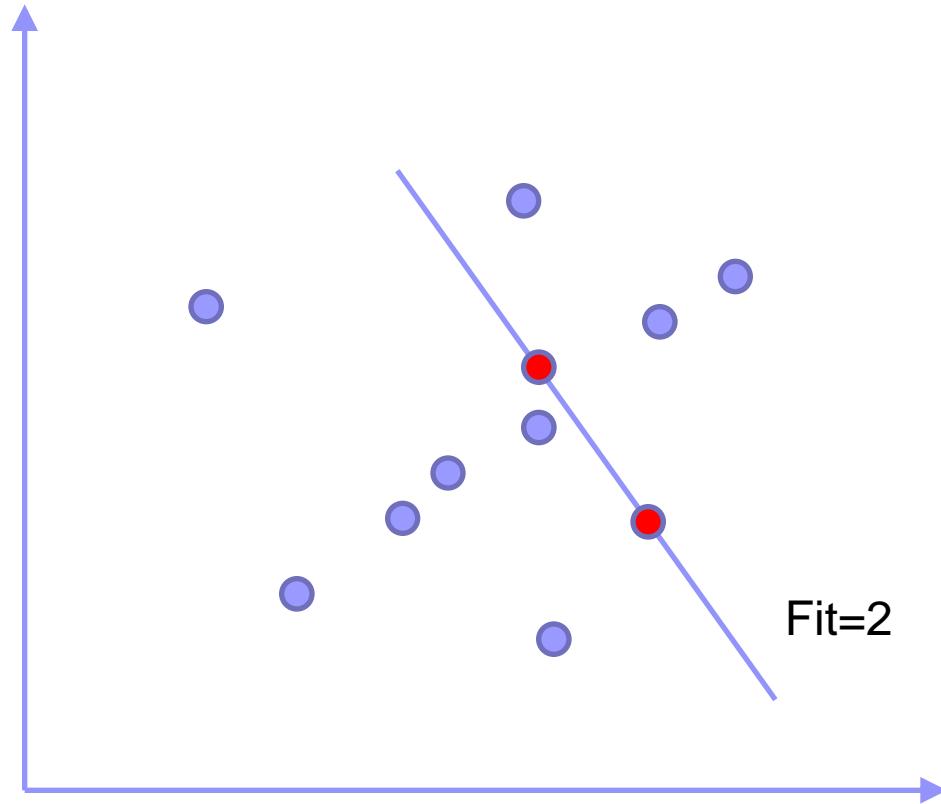
随机选2点

# RANSAC算法



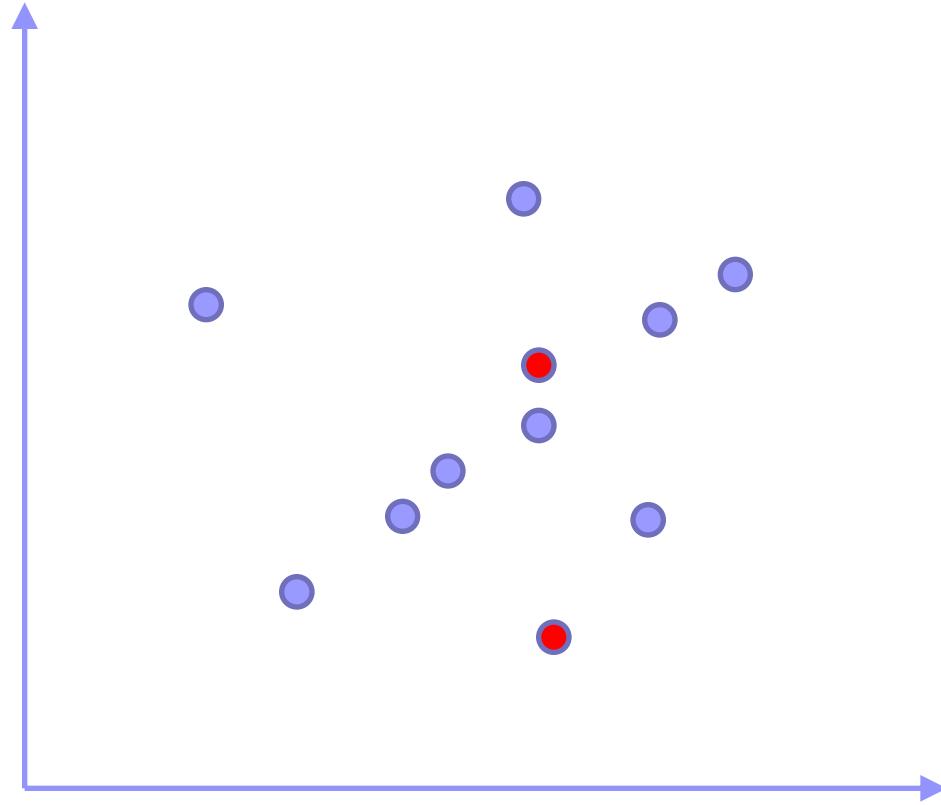
求出一条直线

# RANSAC算法



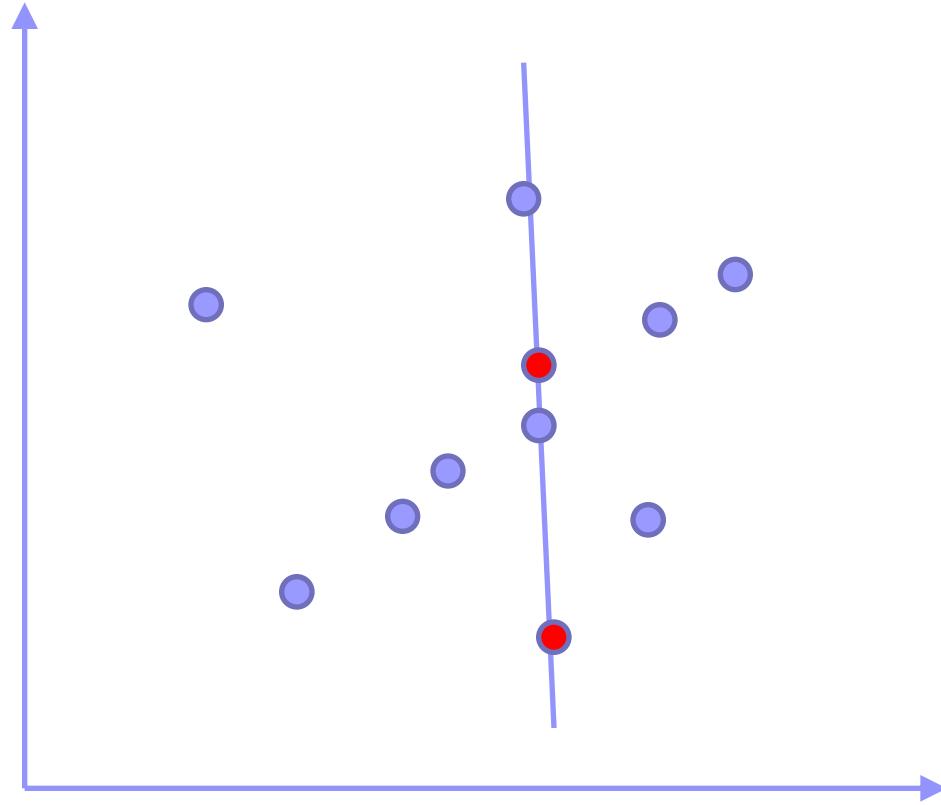
检验哪些点在直线上，有2个

# RANSAC算法



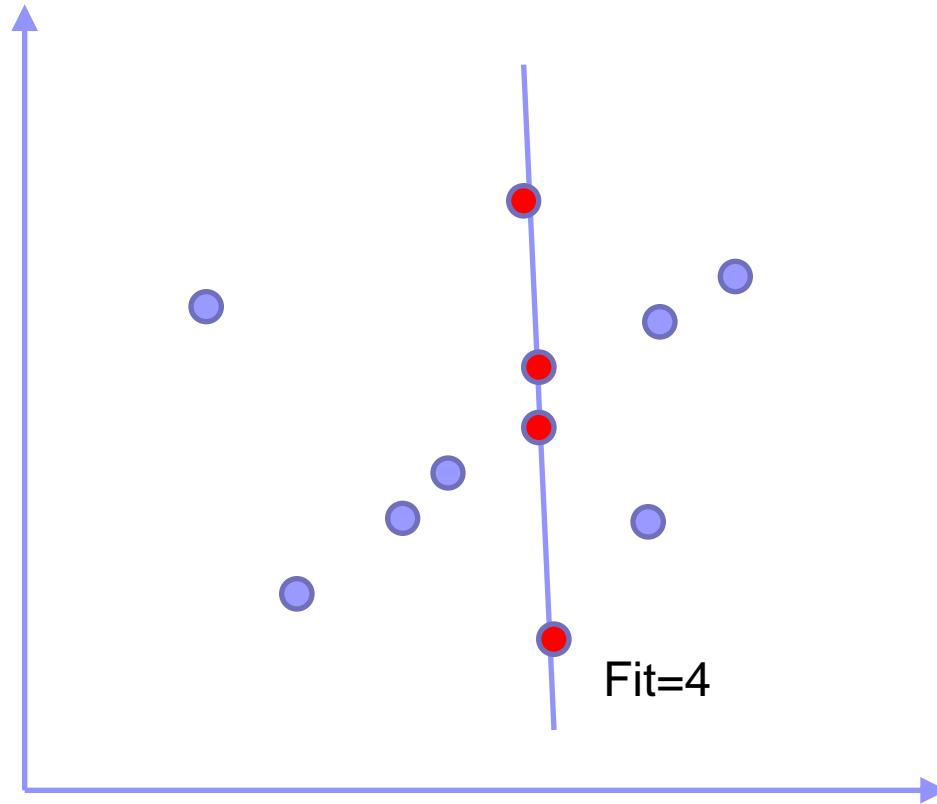
再随机选2点

# RANSAC算法



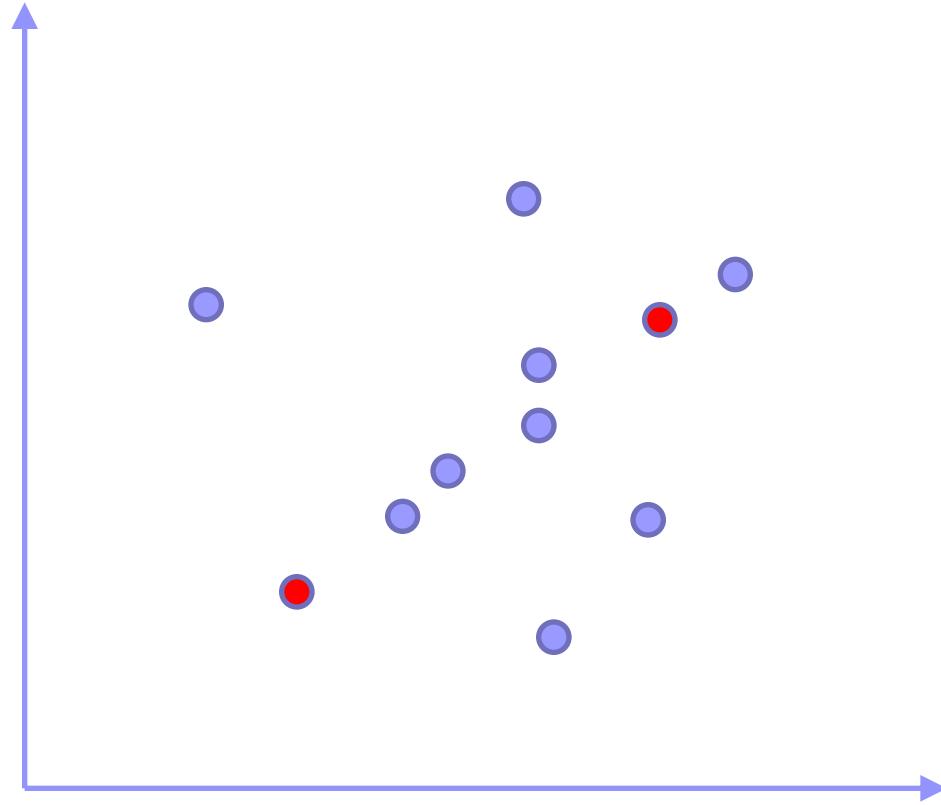
求出一条直线

# RANSAC算法



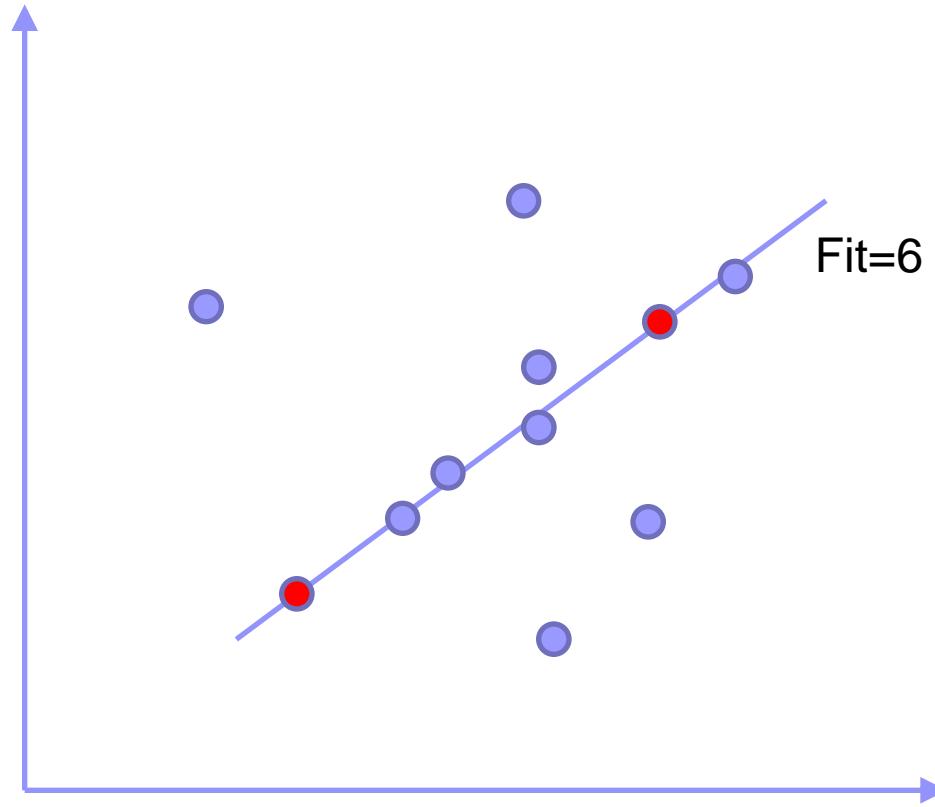
检验哪些点在直线上， $4 > 2$ ，记录下直线参数

# RANSAC算法



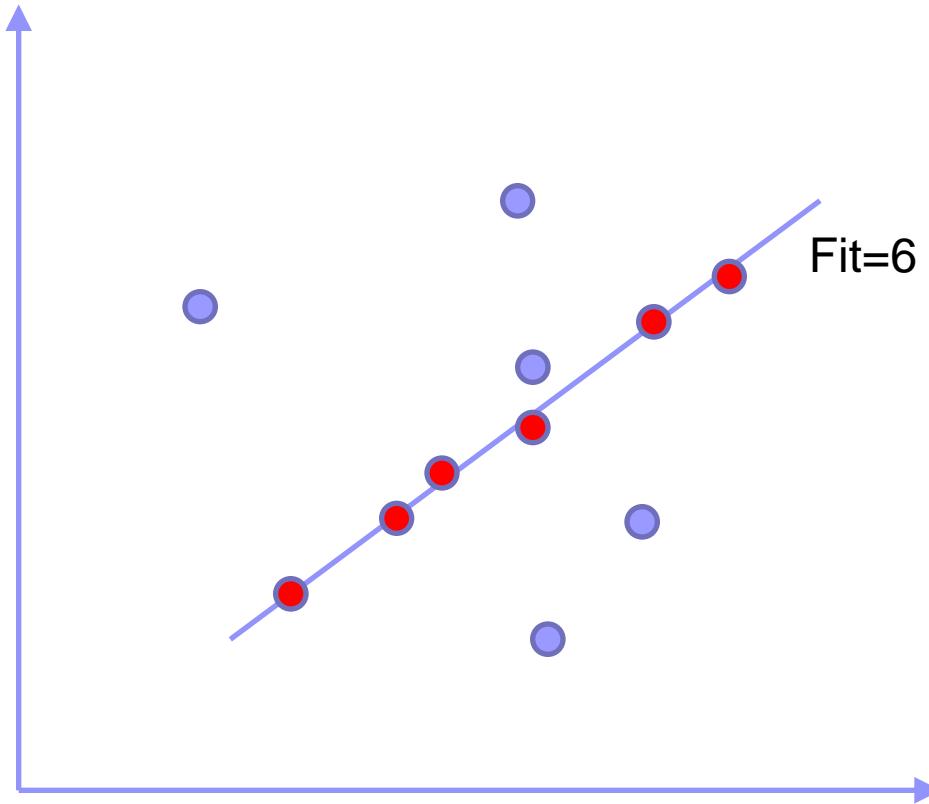
继续随机找2点

# RANSAC算法



求出一条直线

# RANSAC算法



检验哪些点在直线上， $6 > 4$ ，记录下直线参数

# RANSAC算法

- 假如有 $n$ 个点，其中 $m$ 个inlier， $n-m$ 个outlier，迭代 $t$ 次求得正确直线的概率是？

# RANSAC算法

- 假如有n个点，其中m个inlier，n-m个outlier，迭代t次求得正确直线的概率是？

$$1 - \left(1 - \frac{C_m^2}{C_n^2}\right)^t$$

# RANSAC算法

- 假如有n个点，其中m个inlier，n-m个outlier，迭代t次求得正确直线的概率是？

$$1 - \left(1 - \frac{C_m^2}{C_n^2}\right)^t$$

- 对于Homography模型呢？

# RANSAC算法

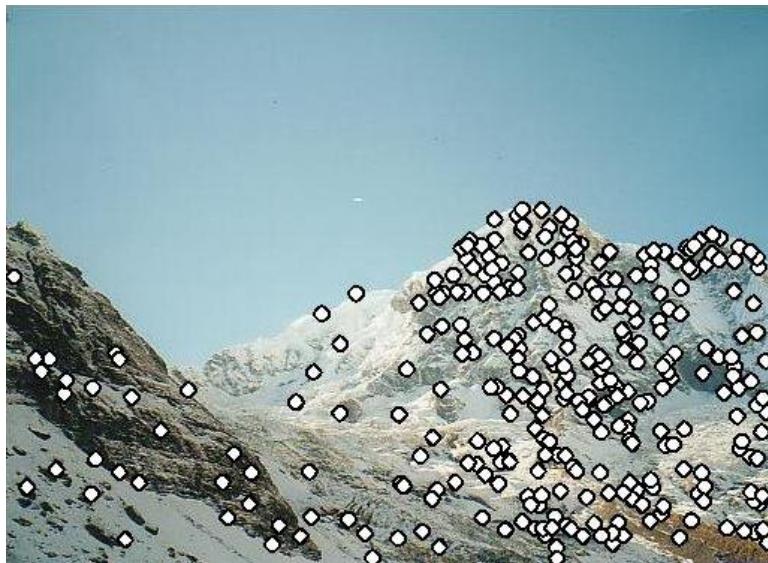
- RANSAC求出的模型参数（直线，Homography），仅用到部分数据（点）
- 剔除outlier之后，重新用所有的inlier再求解一次

# 单视点全景拼接



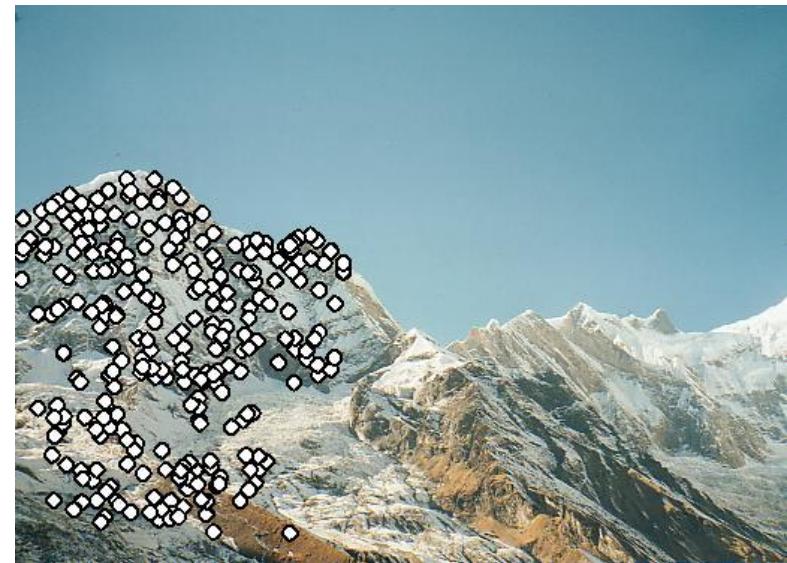
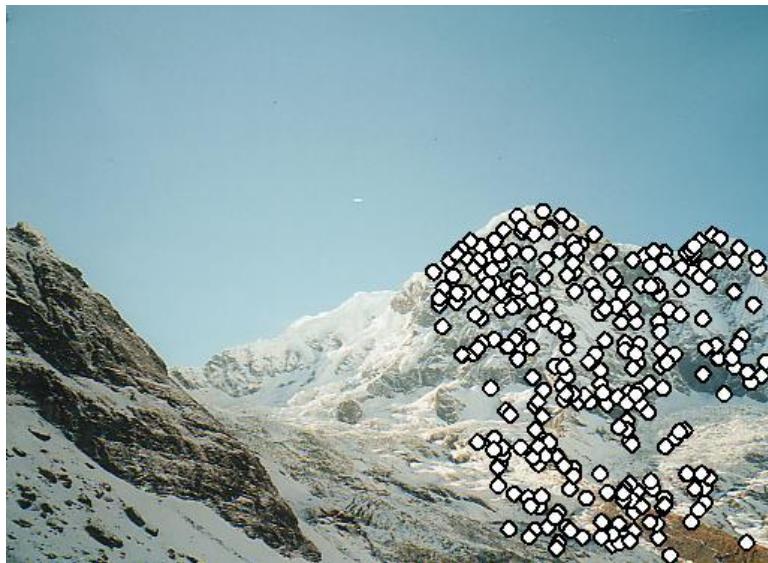
输入图像

# 单视点全景拼接



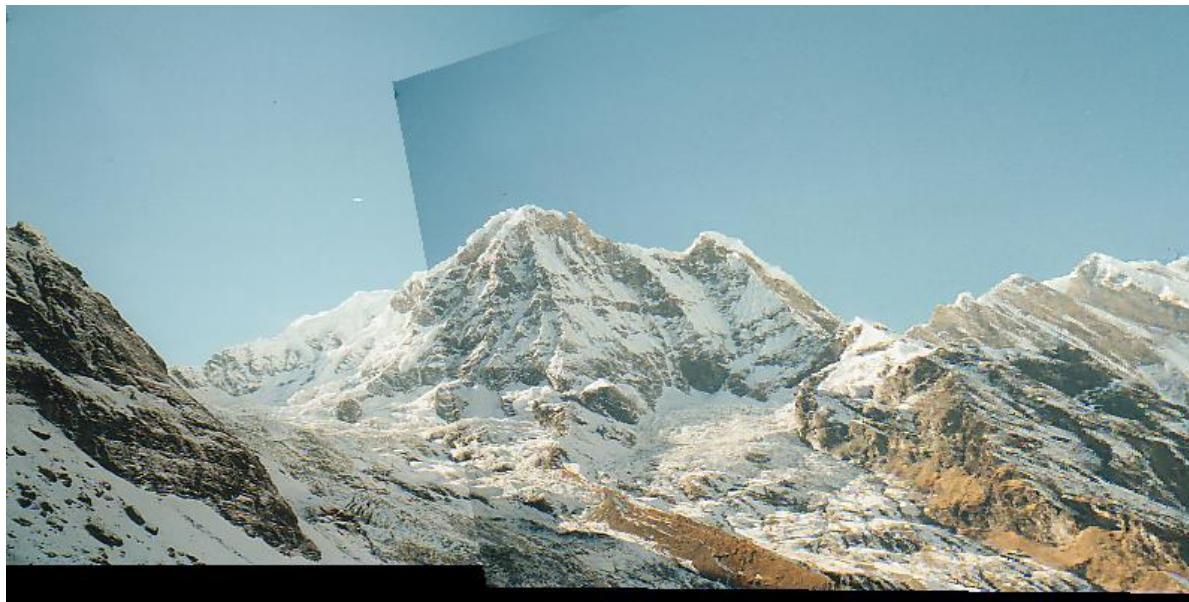
特征匹配

# 单视点全景拼接



RANSAC剔除outlier

# 单视点全景拼接



Homography进行变换，固定第一幅图

# 单视点全景拼接

## ■ 多幅图像全局优化

$$e = \sum_{i=1}^n \sum_{j \in \mathcal{I}(i)} \sum_{k \in \mathcal{F}(i,j)} f(\mathbf{r}_{ij}^k)^2$$

- $n$  = 图像数量
- $\mathcal{I}(i)$  = 与图像*i*重叠的图像集合
- $\mathcal{F}(i, j)$  = 图像*i, j*之间的特征匹配
- $\mathbf{r}_{ij}^k$  = 图像*i, j*之间第*k*个特征匹配的对齐误差（欧氏距离）

# 单视点全景拼接

- 多幅图像全局优化



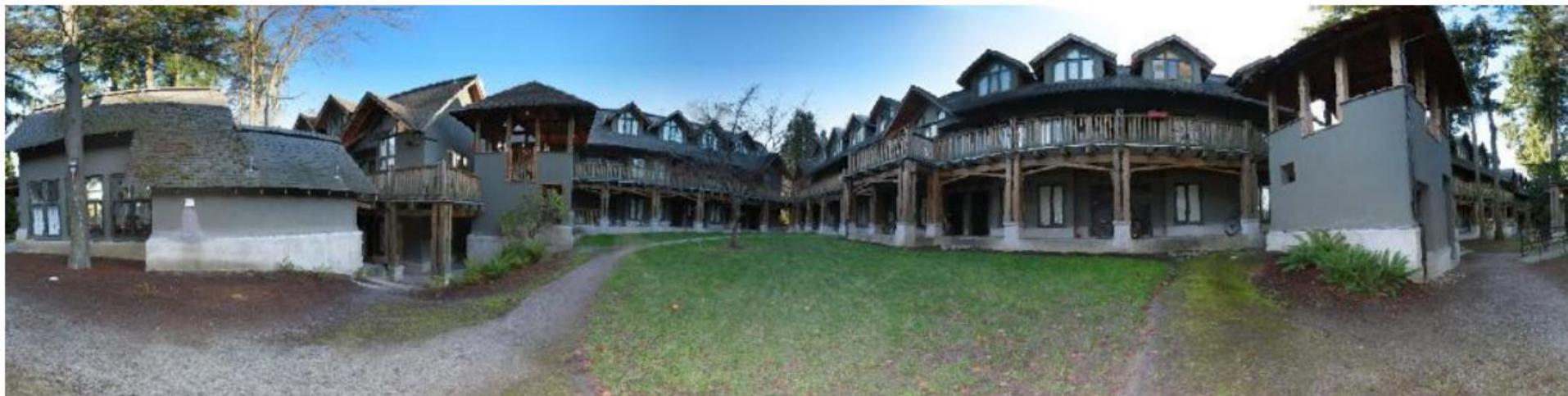
# 单视点全景拼接

- 多幅图像全局优化



# 单视点全景拼接

- 多幅图像全局优化



# 无缝融合

- Multi-band Blend
- Graphcut
- Poisson Image Editing

# 无缝融合

## ■ Multi-band Blend

- Linear Blend容易导致模糊
- 在多个尺度上进行Blend有助于减少模糊效果

## ■ Graphcut

## ■ Poisson Image Editing

# 无缝融合

## ■ Multi-band Blend

### □ 高斯金字塔

- 每一层对上一层压缩表达，因此有信息损失
- 计算残差量化损失，得到拉普拉斯金字塔

### □ 拉普拉斯金字塔

- 高斯金字塔相邻层之差
- 最末层同高斯金字塔

# 无缝融合

- Multi-band Blend
  - 高斯金字塔



level 0



level 1



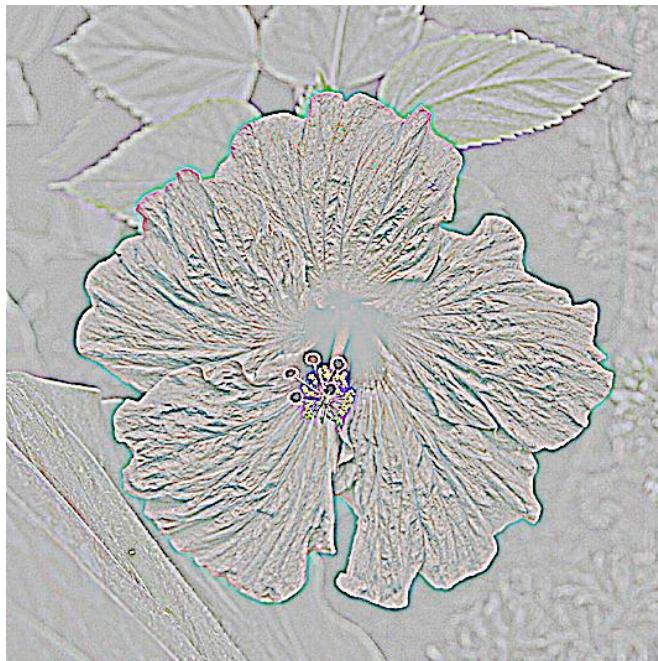
level 2



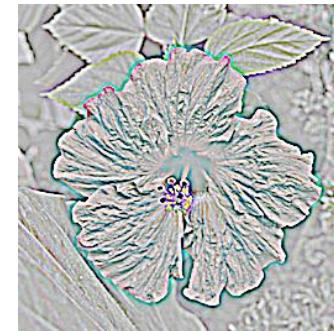
level 3  
(residual)

# 无缝融合

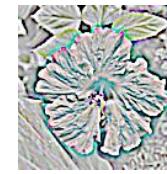
- Multi-band Blend
  - 拉普拉斯金字塔



level 0



level 1



level 2



level 3  
(residual)

# 无缝融合

- Multi-band Blend
- 金字塔的重建
  - 逐层放大
  - 叠加上一层拉普拉斯
  - 得到上一层高斯
  - 直到恢复出原图像

# 无缝融合

## ■ Multi-band Blend

- 对两幅图像分别建立拉普拉斯金字塔
- 逐层blend
- 再恢复原

# 无缝融合



# 2-band Blend



# Linear Blend

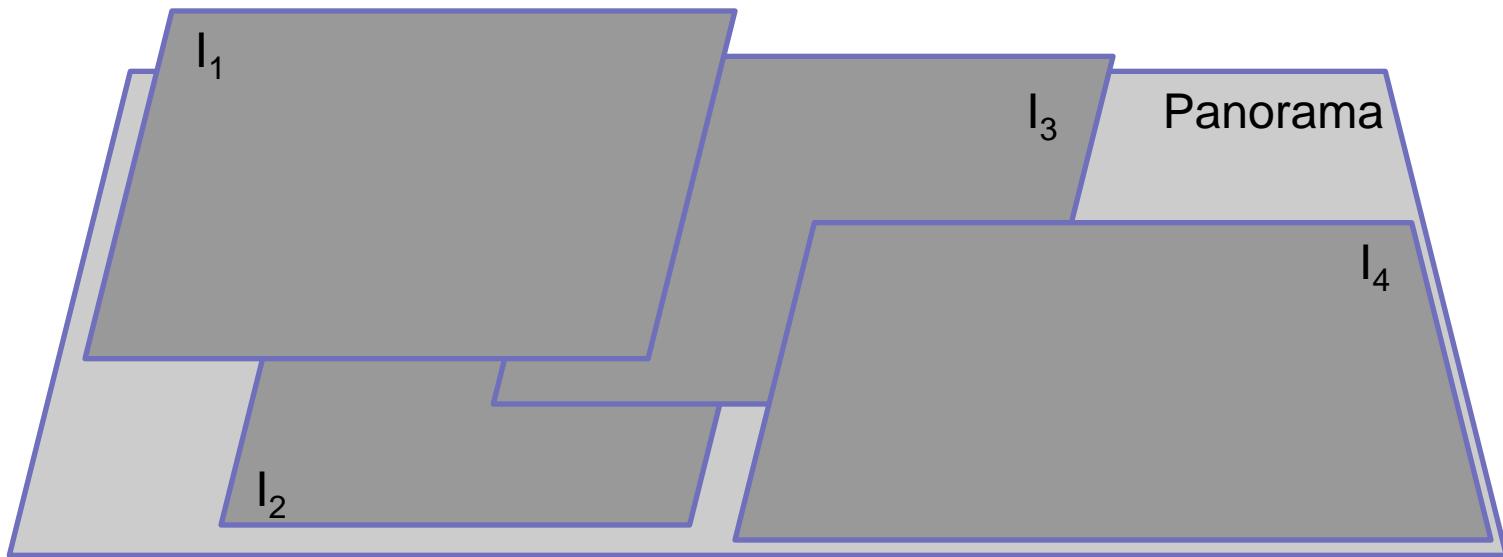


# 无缝融合

- Multi-band Blend
- Graphcut
  - 离散标记问题
  - 每个像素只选择一副图像的像素颜色
  - 没有模糊问题
- Poisson Image Editing

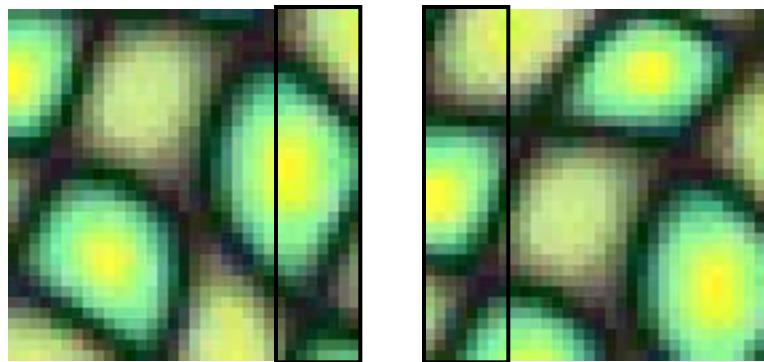
# 无缝融合

- 每个像素有多个来源
- 只选择其中一个

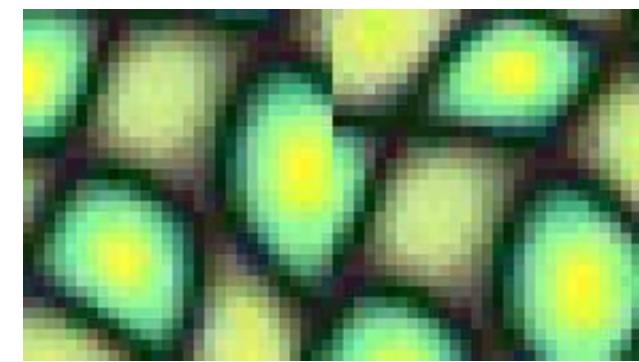


# 无缝融合

重叠的图像



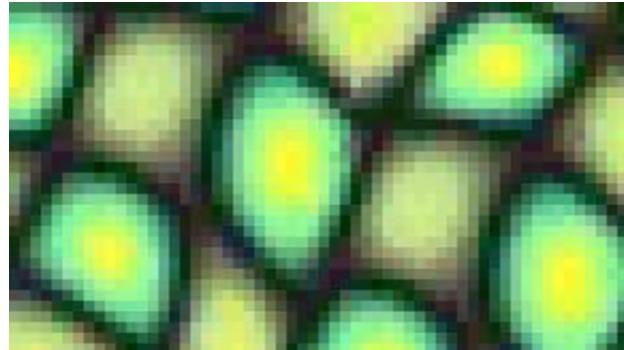
简单的接缝



$$\left[ \begin{array}{c} \text{image 1} \\ - \\ \text{image 2} \end{array} \right] ^2 = \text{color difference}$$

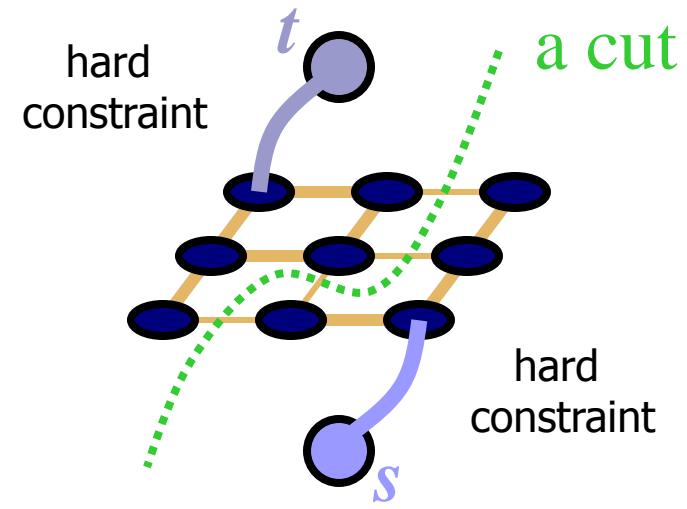
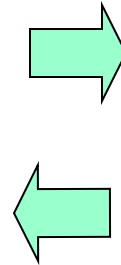
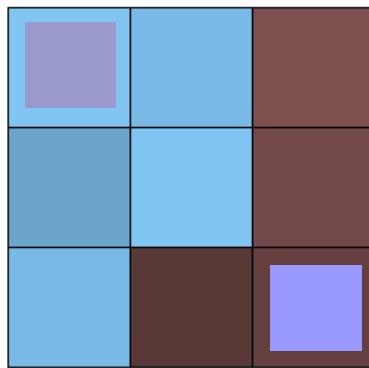
A diagram illustrating the calculation of color difference. It shows two small images, one above the other, with a minus sign between them. To the right of the minus sign is a bracket with a superscript 2, indicating a squared operation. An equals sign follows the bracket, leading to a small image showing a red and black diagonal band, representing the color difference.

颜色差



最小化颜色差的接缝

# 无缝融合



最大流最小割算法

多项式时间

# 无缝融合



<http://www.cc.gatech.edu/cpl/projects/graphcuttextures/>

# 无缝融合

- 离散标记问题
- 为每个像素确定一个Label， 表示来源图像的编号
- MRF(Markov Random Field)

$$E_{cut}(p, L) = \sum_p E_d(p, L_p) + \lambda \sum_{p,q \in N} E_s(p, q, L_p, L_q),$$

# 无缝融合

## ■ Data Term

- 全景图上一个像素p
- 一个变换后的图像编号L<sub>p</sub>
- 是否能覆盖到？

$$E_d(p, L_p) = \begin{cases} 0, & p \in \hat{I}_{L_p} \\ +\infty, & otherwise \end{cases}$$

# 无缝融合

## ■ Smoothness Term

- 相邻两像素p,q
- 如果取自同一副图像， 没有问题
- 如果取自不同图像， 会形成接缝
- 但如果两幅图上对应位置颜色一样， 也没有问题； 如果颜色不一样， 但比较接近， 接缝就不明显
- 因此优化接缝上的颜色差

# 无缝融合

## ■ Smoothness Term

- 相邻两像素p,q
- 假设分别取自图像L<sub>p</sub>,L<sub>q</sub>
- 优化两图像在该处的颜色差

$$E_s(p, q, L_p, L_q) = \|I_{L_p}^*(p) - I_{L_q}^*(p)\| + \|I_{L_p}^*(q) - I_{L_q}^*(q)\|$$

- 当L<sub>p</sub>,L<sub>q</sub>相等时能量为0， 无缝
- 当L<sub>p</sub>,L<sub>q</sub>在此处颜色越接近能量越小

# 无缝融合

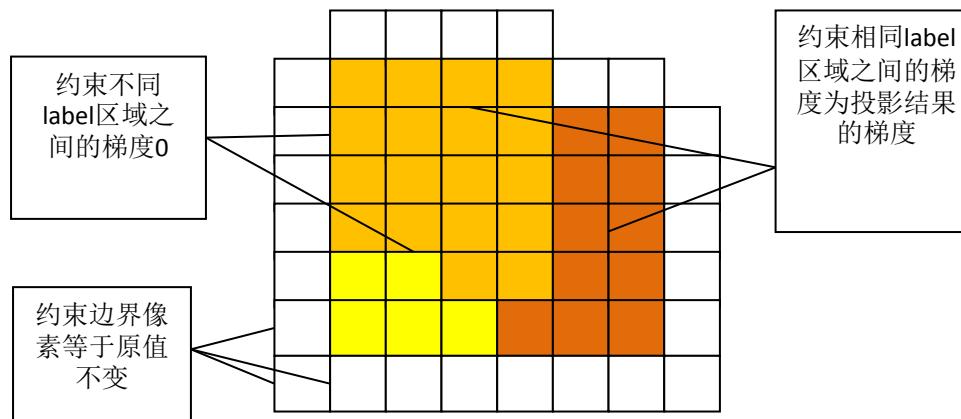
- 2-label的问题可转化为最大流最小割模型
- Multi-label转化为多次2-label问题迭代
- [http://www.csd.uwo.ca/faculty/olga/Papers/pami01\\_final.pdf](http://www.csd.uwo.ca/faculty/olga/Papers/pami01_final.pdf)
- <http://vision.csd.uwo.ca/code/>

# 无缝融合

- Multi-band Blend
- Graphcut
- Poisson Image Editing
  - 为每个像素求解颜色值
  - 全局颜色矫正

# 无缝融合

- 把接缝的色差分散到整幅图像。
- 绝对亮度：不敏感
- 相对的梯度：敏感
- 保持原有的梯度，但在接缝处梯度设为0。



# 无缝融合

## ■ 能量函数

$$E(I) = \sum_{p \in B} |I(p) - I_{L_p}(p)|^2 + \sum_{(p,q) \in S} |I(p) - I(q)|^2 + \sum_{(p,q) \in \bar{S}} |(I(p) - I(q)) - (I_{L_p}(p) - I_{L_q}(q))|^2$$

# 无缝融合

## ■ 能量函数

$$E(I) = \sum_{p \in B} |I(p) - I_{L_p}(p)|^2 + \sum_{(p,q) \in S} |I(p) - I(q)|^2 + \sum_{(p,q) \in \bar{S}} |(I(p) - I(q)) - (I_{L_p}(p) - I_{L_q}(q))|^2$$

边界条件

- **B**: 边界像素集合，一般不存在重叠
- **Lp**: Graphcut求解的结果

# 无缝融合

## ■ 能量函数

$$E(I) = \sum_{p \in B} |I(p) - I_{L_p}(p)|^2 + \sum_{(p,q) \in S} |I(p) - I(q)|^2 + \sum_{(p,q) \in \bar{S}} |(I(p) - I(q)) - (I_{L_p}(p) - I_{L_q}(q))|^2$$

边界条件              接缝处梯度为0

- **B**: 边界像素集合，一般不存在重叠
- **L<sub>p</sub>**: Graphcut求解的结果。
- $S = \{(p, q) | L_p \neq L_q\}$  接缝像素集合。

# 无缝融合

## ■ 能量函数

$$E(I) = \sum_{p \in B} |I(p) - I_{L_p}(p)|^2 + \sum_{(p,q) \in S} |I(p) - I(q)|^2 + \sum_{(p,q) \in \bar{S}} |(I(p) - I(q)) - (I_{L_p}(p) - I_{L_q}(q))|^2$$

边界条件

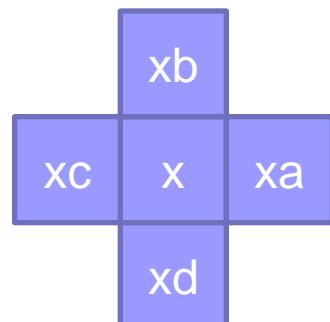
接缝处梯度为0

非接缝处梯度保持不变

- **B**: 边界像素集合，一般不存在重叠
- **L<sub>p</sub>**: Graphcut求解的结果。
- $S = \{(p, q) | L_p \neq L_q\}$  接缝像素集合。

# 无缝融合

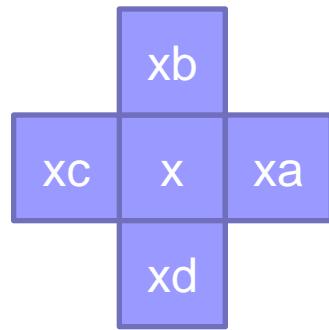
- 把图像所有像素的颜色作为未知量，形成一个长的列向量
- 方程形式如下



$$\begin{pmatrix} \dots & -1 & \dots & \dots & 1 \\ \dots & & -1 & 1 & \dots \\ \dots & & 1 & -1 & \dots \\ \dots & & 1 & \dots & -1 & \dots \\ \dots & & & & & \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{pmatrix} = \begin{pmatrix} \dots \\ y_0 \\ y_1 \\ \dots \\ y_3 \\ \dots \end{pmatrix}$$

# 无缝融合

- $x$ 与相邻4像素构成4个方程， 占矩阵的4行
- 可把4行叠加在一起



$$\begin{pmatrix} \dots & -1 & \dots & \dots & 1 & \dots \\ \dots & & & -1 & 1 & \dots \\ \dots & & & 1 & -1 & \dots \\ \dots & & & 1 & \dots & -1 & \dots \\ \dots & & & & & & \dots \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{pmatrix} = \begin{pmatrix} \dots \\ y_a \\ y_b \\ y_c \\ y_d \\ \dots \end{pmatrix}$$

# 无缝融合

- $x$ 与相邻4像素构成4个方程， 占矩阵的4行
- 可把4行叠加在一起
- 形成Laplacian矩阵

$$\begin{pmatrix} -1 & \dots & -1 & 4 & -1 & \dots & -1 \\ -1 & \dots & -1 & 4 & -1 & \dots & -1 \\ -1 & \dots & -1 & 4 & -1 & \dots & \\ -1 & \dots & -1 & 4 & -1 & \dots & \\ \dots & & & & & & \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \dots \\ \dots \\ \vdots \\ \dots \end{pmatrix}$$

# 无缝融合

- 稀疏&对称&方阵
- 对角线上元素为4
  - 部分元素为3（对应边界像素）
  - 部分元素为2（对应角上像素）
- 每行最多5个元素，并且和为0

# 无缝融合

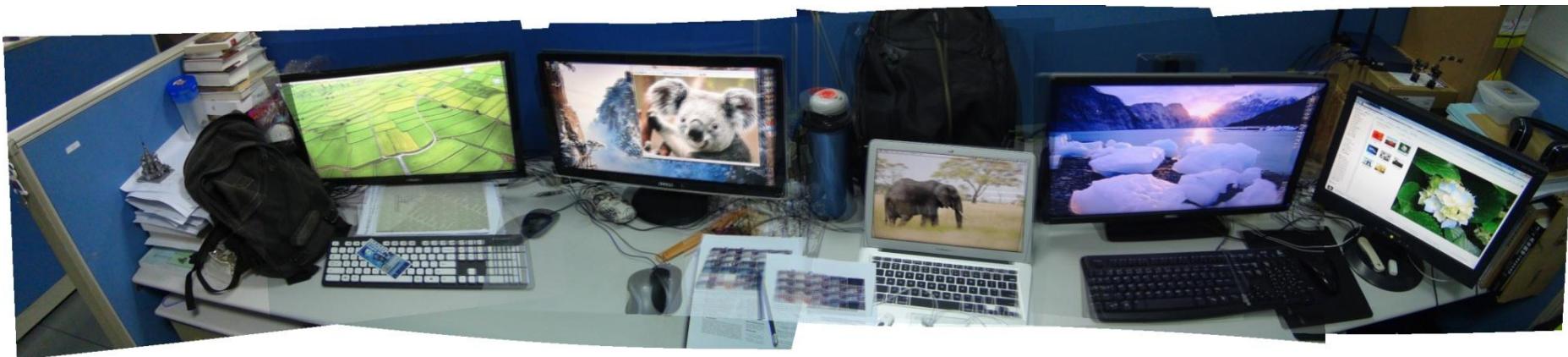
- 边界条件
  - 在对角线上加1
  - 右边加上相应的原图像颜色值
  - 使得矩阵正定
- Gaussian-sidel/Conjugate-Gradient/LLT

# 无缝融合

■ Laplacian的优化与直接用最小二乘法的区别何在？

- 利用方程组的特点，直接构造对称方阵
- 省去转置乘法，矩阵也更稀疏
- 求解效率更高

# 无缝融合



Blend

# 无缝融合



Graphcut

# 无缝融合



Graphcut+Poisson

# 全景拼接

## ■ 单视点全景拼接

- 特征匹配
- 剔除outlier
- 单应矩阵求解
- 无缝融合 (Multi-band Blend, Graphcut, Poisson Image Editing)

## ■ 多视点全景拼接

- Dual-Homography
- APAP
- Long scene

# 多视点全景拼接

## ■ Dual-Homography

- 单视点情况下，相机很难保持绝对静止
- 没有近景的情况下，分为地面，远景两部分
- 远景近似一个平面

## ■ APAP

## ■ Long scene

# 多视点全景拼接

## ■ Dual-Homography

□ Homography两个条件

- 相机纯旋转
- 场景是平面

□ 相机位置有不大的扰动

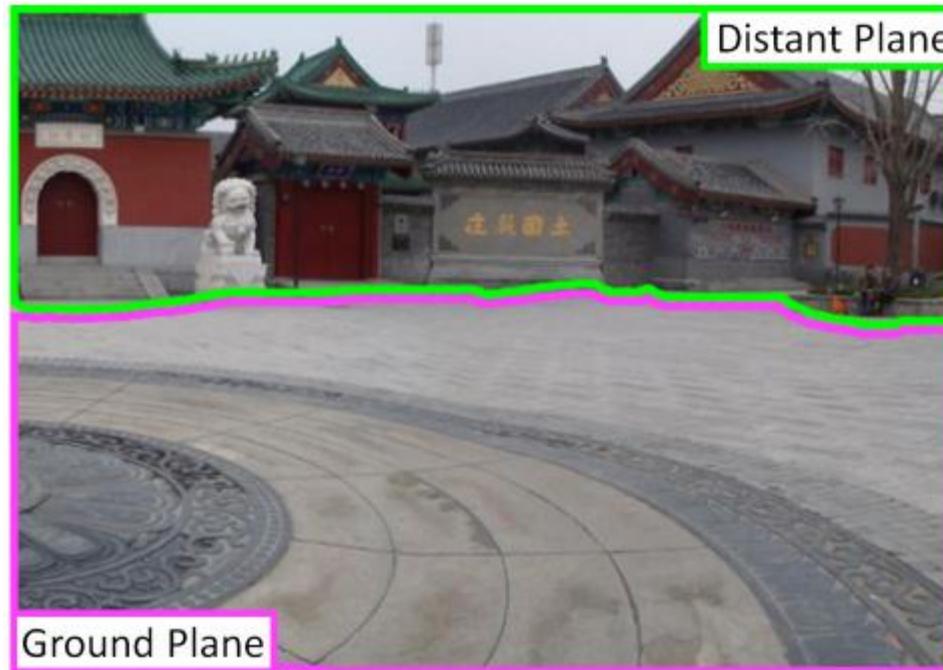
□ 使用两个Homography

■ [https://www.comp.nus.edu.sg/~brown/pdf/cvpr\\_dualhomography2011.pdf](https://www.comp.nus.edu.sg/~brown/pdf/cvpr_dualhomography2011.pdf)

# 多视点全景拼接

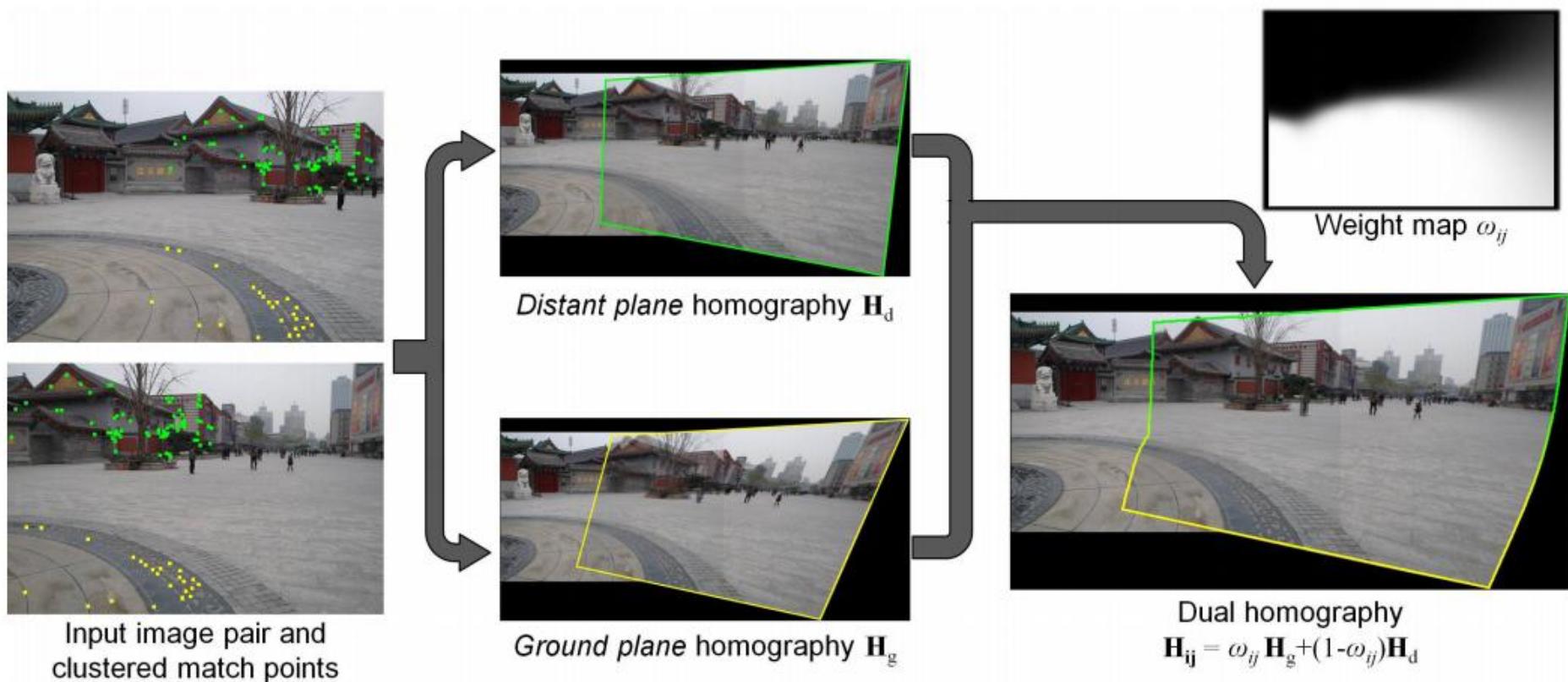
## ■ Dual-Homography

- 场景分为两个平面
- 远处物体可以近似为一个平面



# 多视点全景拼接

## ■ Dual-Homography



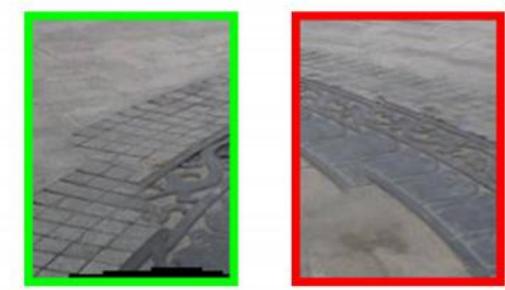
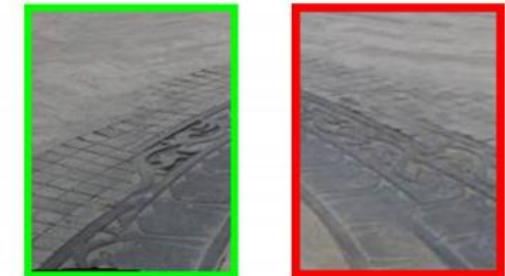
# 多视点全景拼接

## ■ Dual-Homography

- 提取特征点
- 特征点聚类
- 每类单独求Homography
- Cluster内像素直接使用Homography
- 其余像素根据位置计算权重，插值得到 Homography。

# 多视点全景拼接

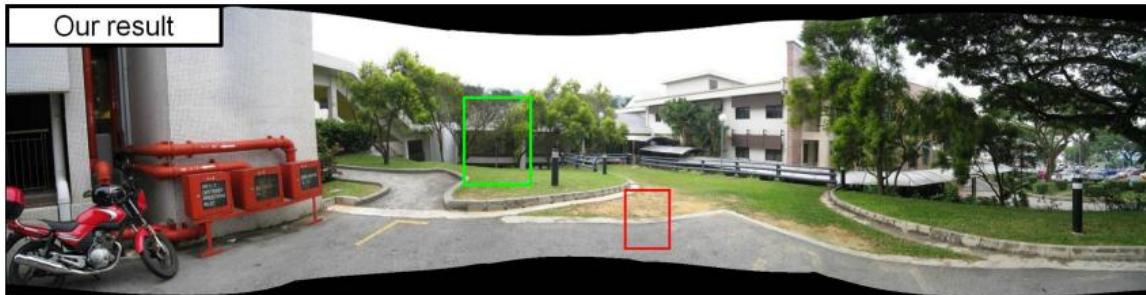
## ■ Dual-Homography



# 多视点全景拼接

## ■ Dual-Homography

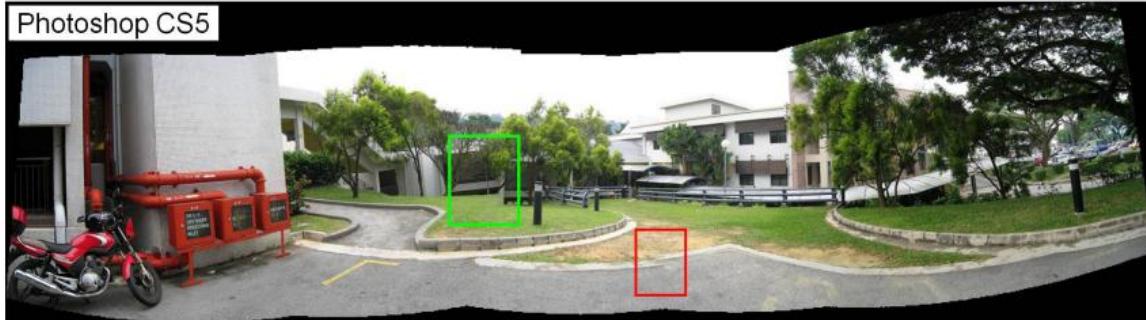
Our result



AutoStitch



Photoshop CS5



# 多视点全景拼接

- Dual-Homography

- APAP

- 使用更多的Homography

- 对图像不同区域使用不同Homography

- 按位置插值

- Long scene

# 多视点全景拼接

## ■ APAP

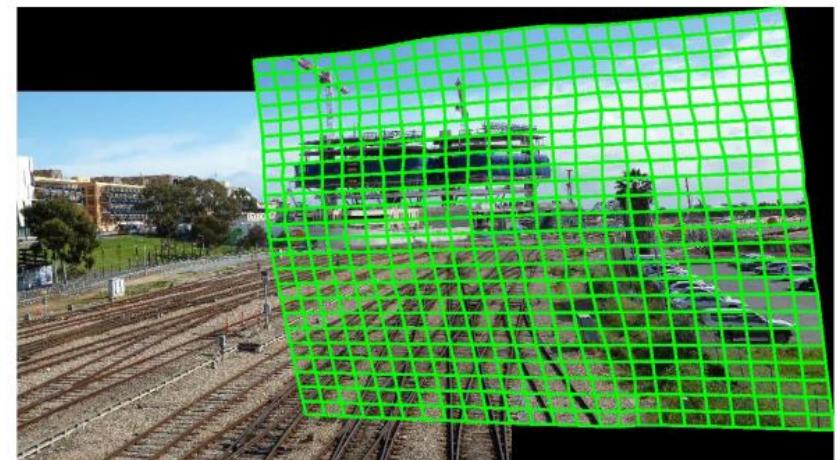
- 对图像划分均匀网格
- 每个格子一个 Homography
- 根据特征点距离插值



(a) Target image  $I'$ .



(b) Source image  $I$  with  $100 \times 100$  cells (only  $25 \times 25$  drawn for clarity).



(c) Aligned images with transformed cells overlaid to visualise the warp. Observe that the warp is globally projective for extrapolation, but adapts flexibly in the overlap region for better alignment.

# 多视点全景拼接

## ■ APAP

- 原能量函数

$$\arg \min_H \sum_{(p,q) \in C} |H(p) - q|^2$$

- H:全局的Homography
- (p,q)匹配的特征点对
- 优化拟合误差之和

# 多视点全景拼接

## ■ APAP

- 新能量函数

$$\arg \min_H \sum_{(p,q) \in C} w_p |H(p) - q|^2$$

- 加入权重  $w_p$

- 取决于特征点  $p$  到格子中心的距离

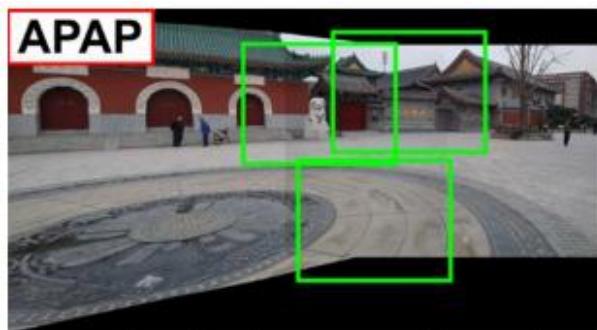
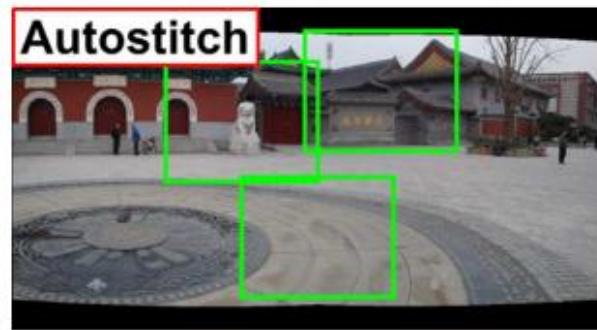
$$w_p = \exp\left(-\frac{|x-p|^2}{2\sigma^2}\right)$$

- 近的特征点贡献大，远的特征点贡献小

- Homography 具有一定的局部性

# 多视点全景拼接

## ■ APAP



# 多视点全景拼接

## ■ APAP



Autostitch

# 多视点全景拼接

## ■ APAP

APAP



# 多视点全景拼接

## ■ APAP

Autostitch



# 多视点全景拼接

## ■ APAP

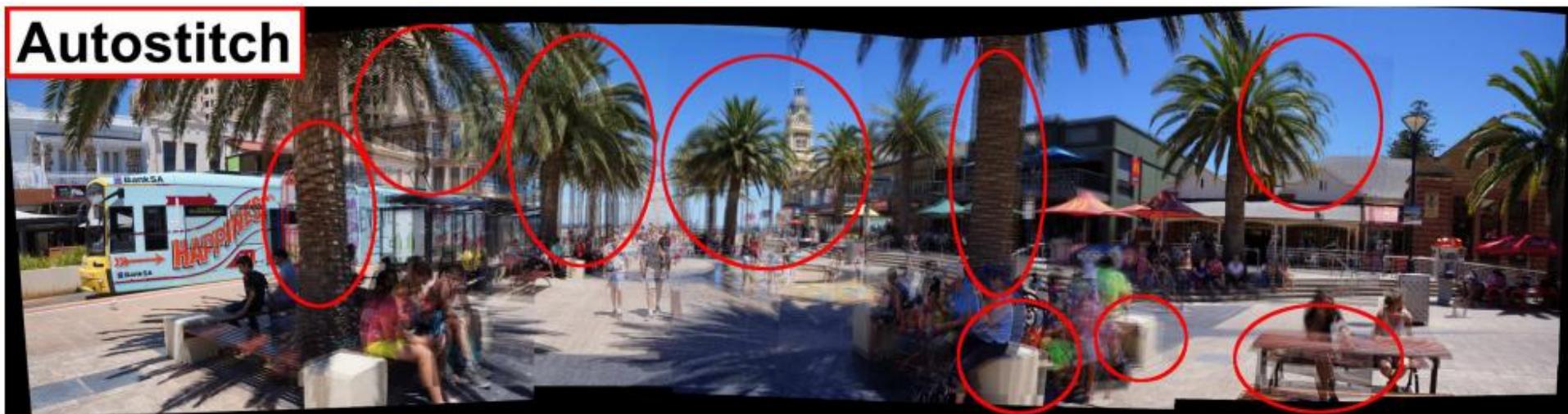
APAP



# 多视点全景拼接

## ■ APAP

**Autostitch**



# 多视点全景拼接

## ■ APAP

**APAP**

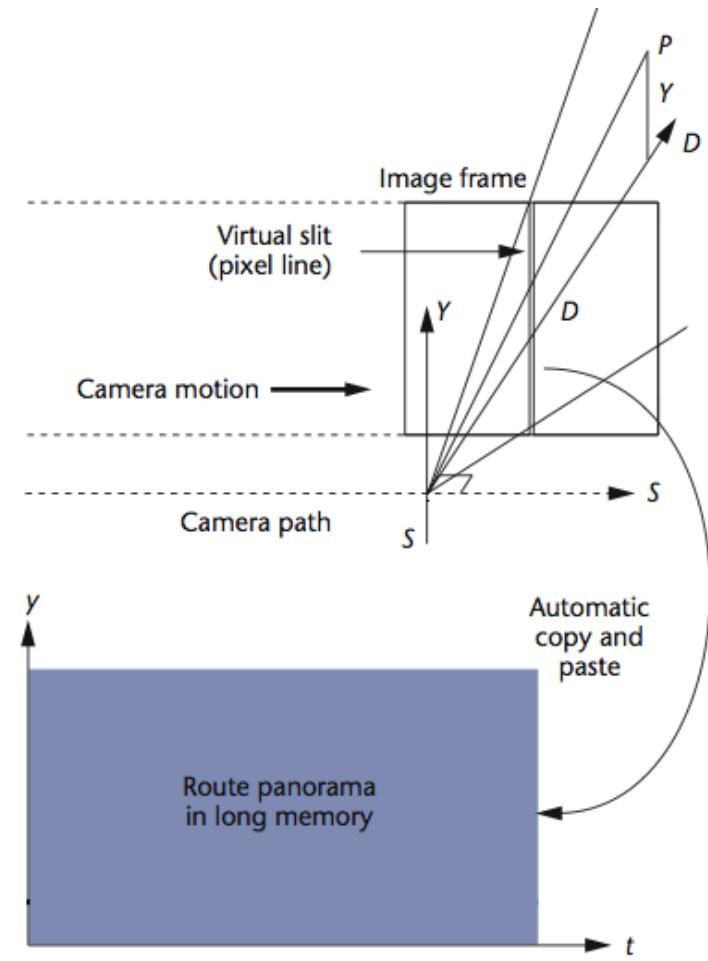
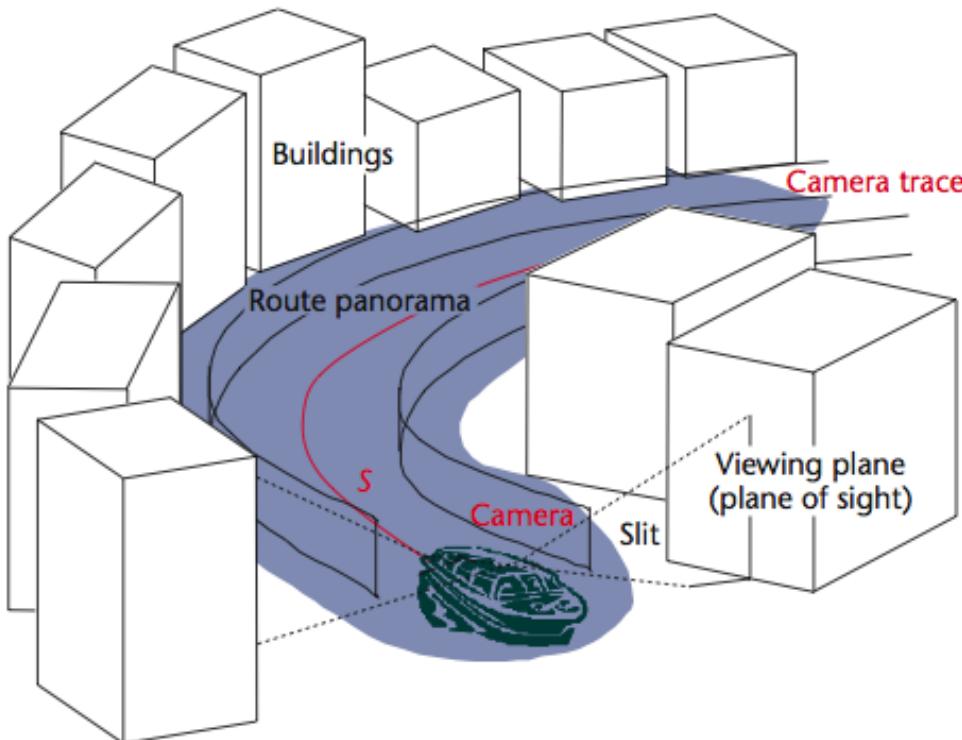


# 多视点全景拼接

- Dual-Homography
- APAP
- **Long scene**
  - 超长场景（街景）的正交投影
  - 散点透视，多个消失点分布在水平方向

# 多视点全景拼接

## ■ Strip Panoramas



# 多视点全景拼接

## ■ Strip Panoramas

- 水平方向正交投影
- 垂直方向透视投影



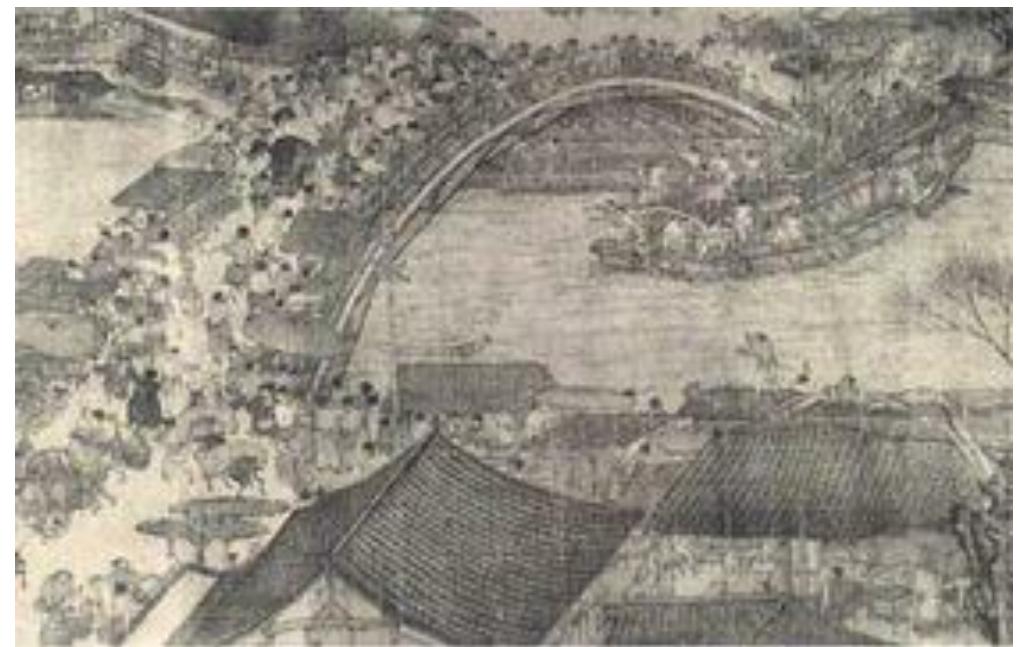
- 远处拉伸，近处挤压

# 多视点全景拼接

## ■ 清明上河图



- 散点透视
- 多个消失点
- 分布在一条线上



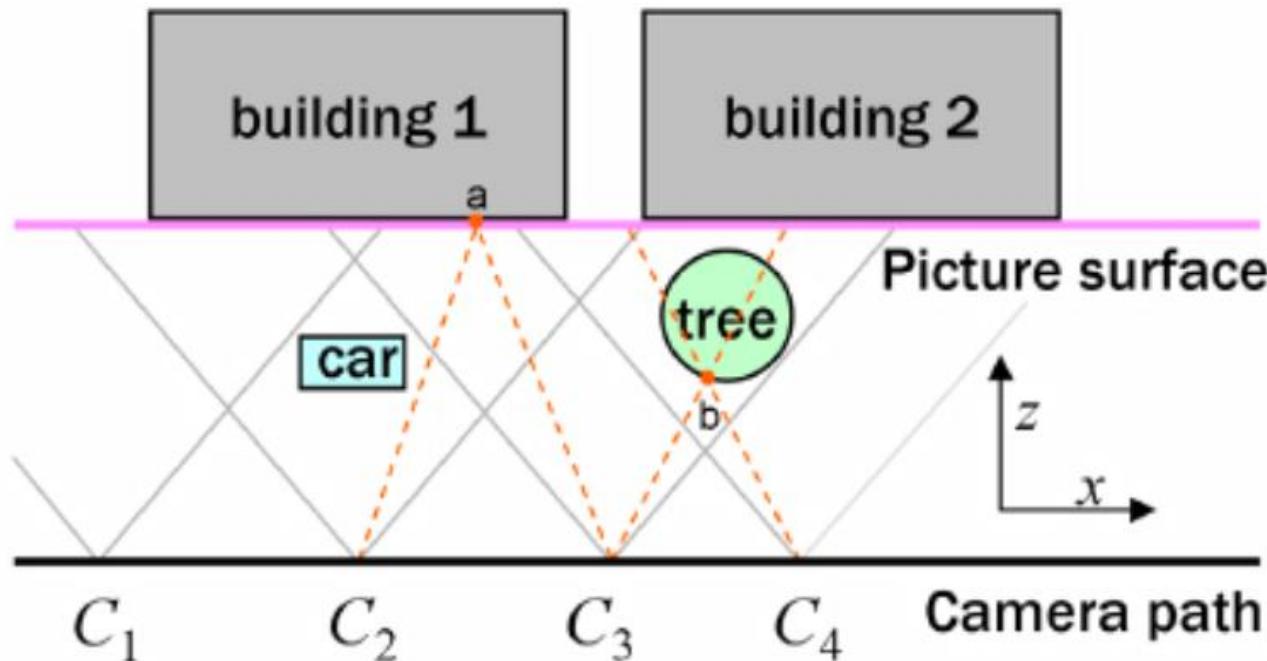
# 多视点全景拼接

- <http://grail.cs.washington.edu/projects/multipano/>



# 多视点全景拼接

- 不同深度的物体形成视差
- 不在主平面上的物体，投影位置不一致



# 多视点全景拼接

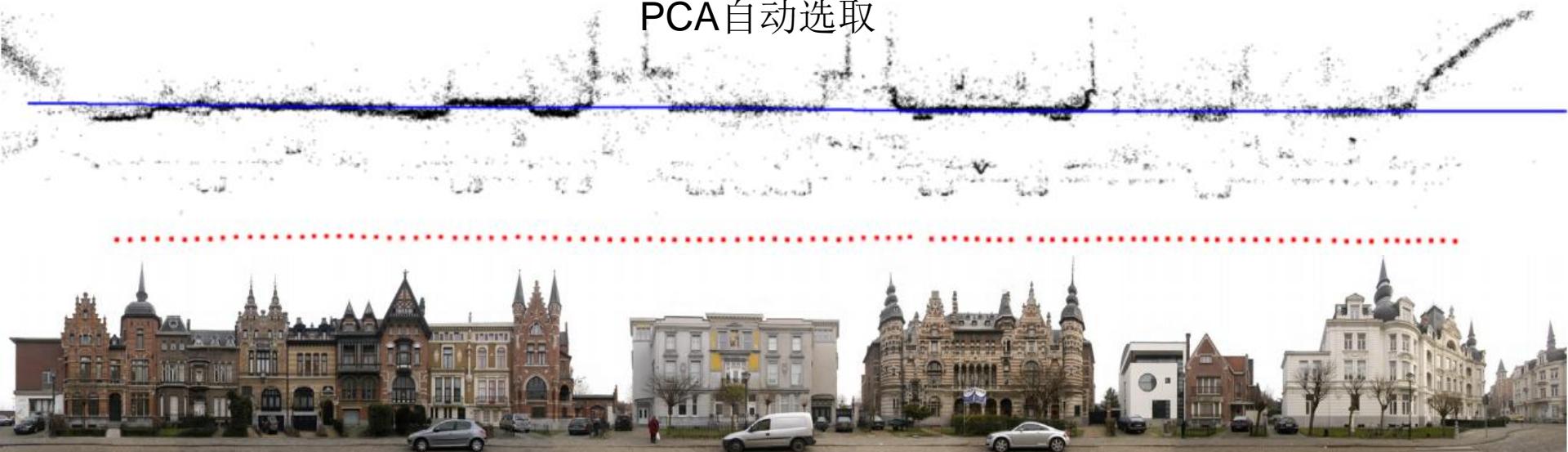
## ■ Structure from Motion

- 输入：图像序列
- 输出：每帧摄像机参数（RT）
- 输出：三维点云

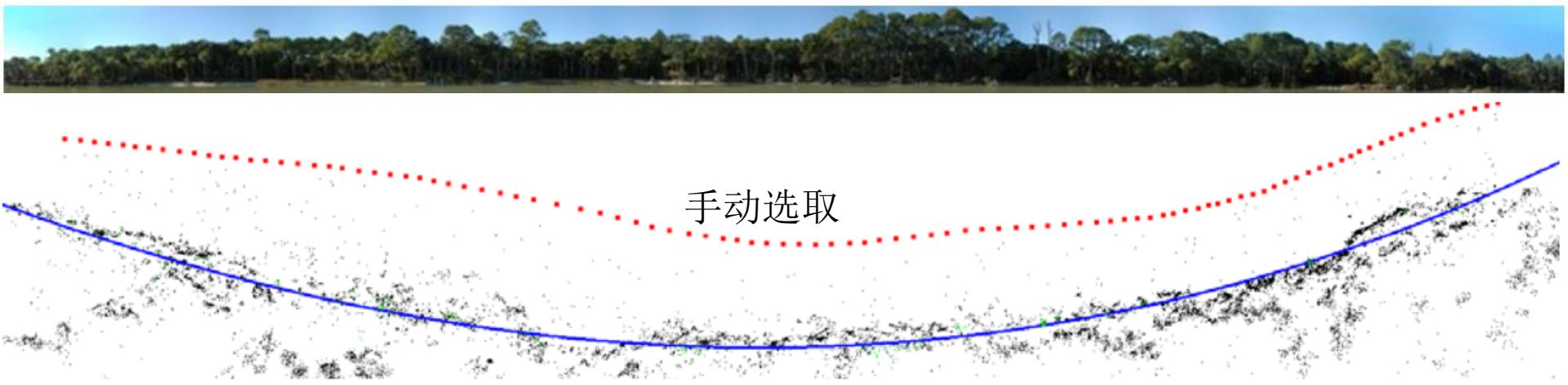


# 多视点全景拼接

PCA自动选取

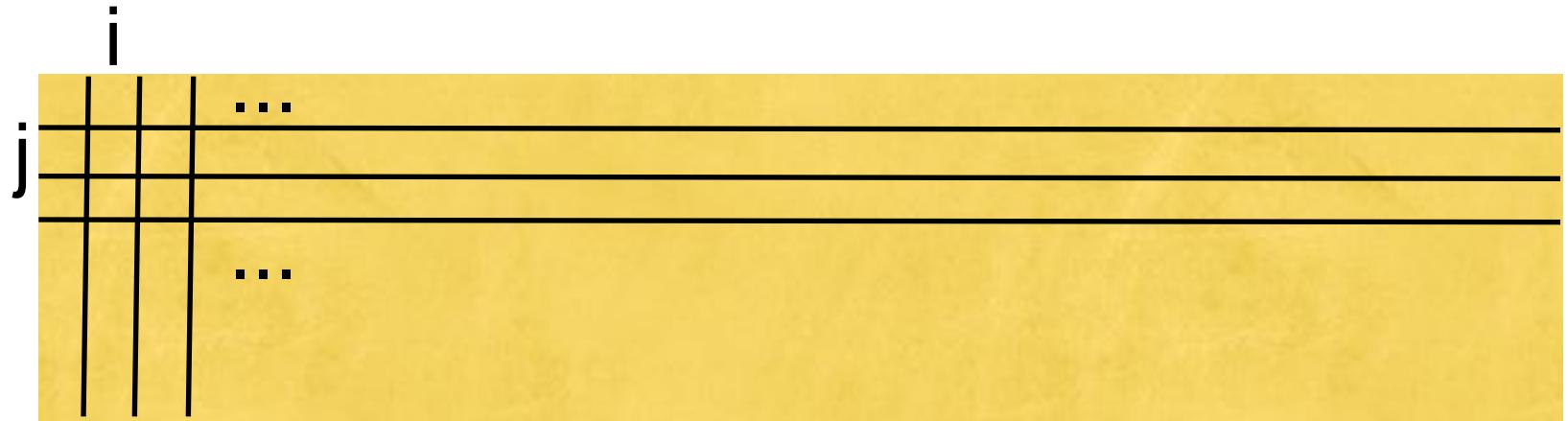


手动选取



# 多视点全景拼接

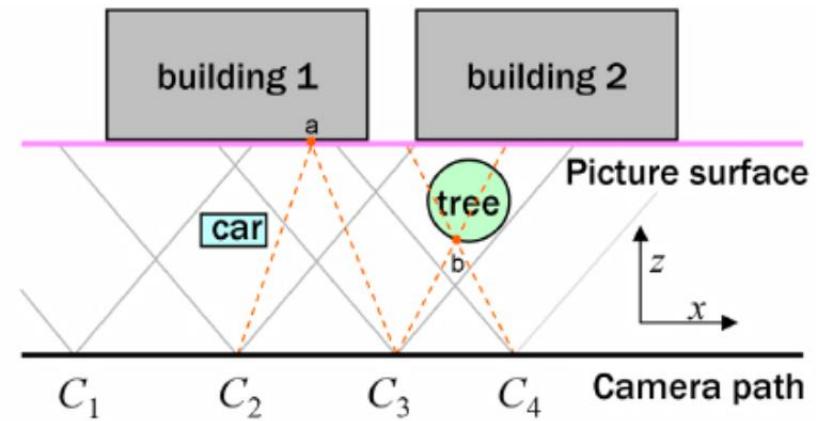
- 以选择的面  $S$ , 作为最终投影面



- 像素  $S(i, j)$ , 根据相机参数投影到图像
- 如投影在图像以内, 获得像素颜色
- 投影到多帧图像, 重叠问题

# 多视点全景拼接

- Blend
  - 投影面上
    - 各相机一致
  - 投影面以外
    - 视差
    - 模糊



# 多视点全景拼接

- Graphcut
- Data Term I
  - 希望相机正对像素
  - 像素 $p$ 对应曲面三维点 $S(p)$
  - 光线:  $S(p)-C_i$
  - 优化与 $S(p)$ 处法向的夹角
  - 近似为:

$$D(p, L(p)) = |p - p_{L(p)}|$$

# 多视点全景拼接

- Graphcut
- Data Term II
  - 对于主平面/曲面上的像素
  - 希望接近均值
$$H(p, L(p)) = \begin{cases} |M(p) - I_{L(p)}(p)| & \text{if } \sigma(p) < 10 \\ 0 & \text{otherwise} \end{cases}$$
  - M(p): 中值
  - IL(p)(p): 所选label对应的颜色
  - σ(p): 标准差, 区分主平面上的点

# 多视点全景拼接

- Graphcut
- Smoothness Term
  - 同传统方法
  - 相邻两像素在两个不同输入图像上的颜色差

$$V(p, L(p), q, L(q)) = |I_{L(p)}(p) - I_{L(q)}(p)|^2 + |I_{L(p)}(q) - I_{L(q)}(q)|^2$$

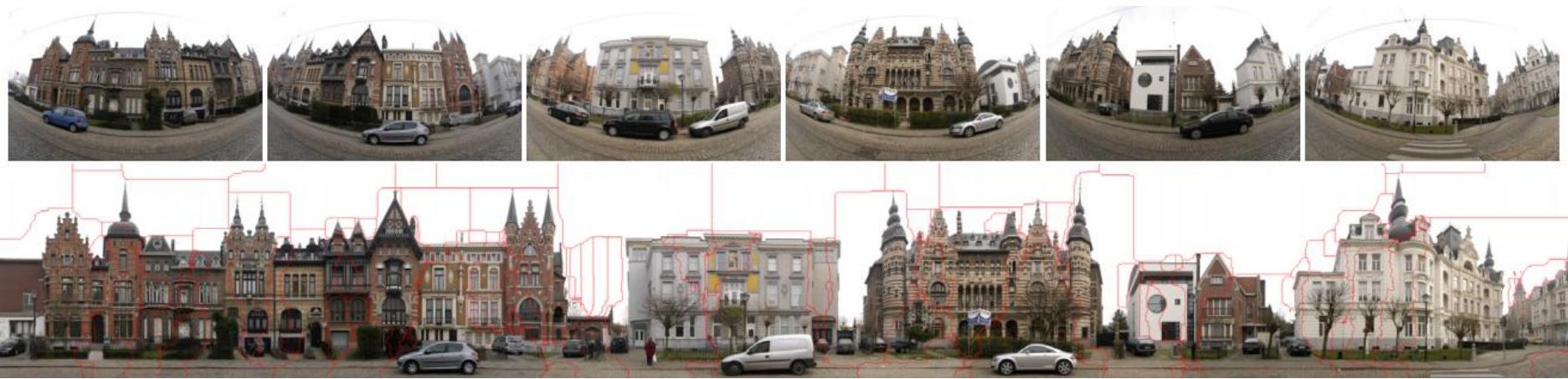
# 多视点全景拼接

- Graphcut
  - 总能量函数

$$\sum_p (\alpha D(p, L(p)) + \beta H(p, L(p))) + \sum_{p,q} V(p, L(p), q, L(q))$$

- $\alpha$ : 100
- $\beta$ : .25

# 多视点全景拼接



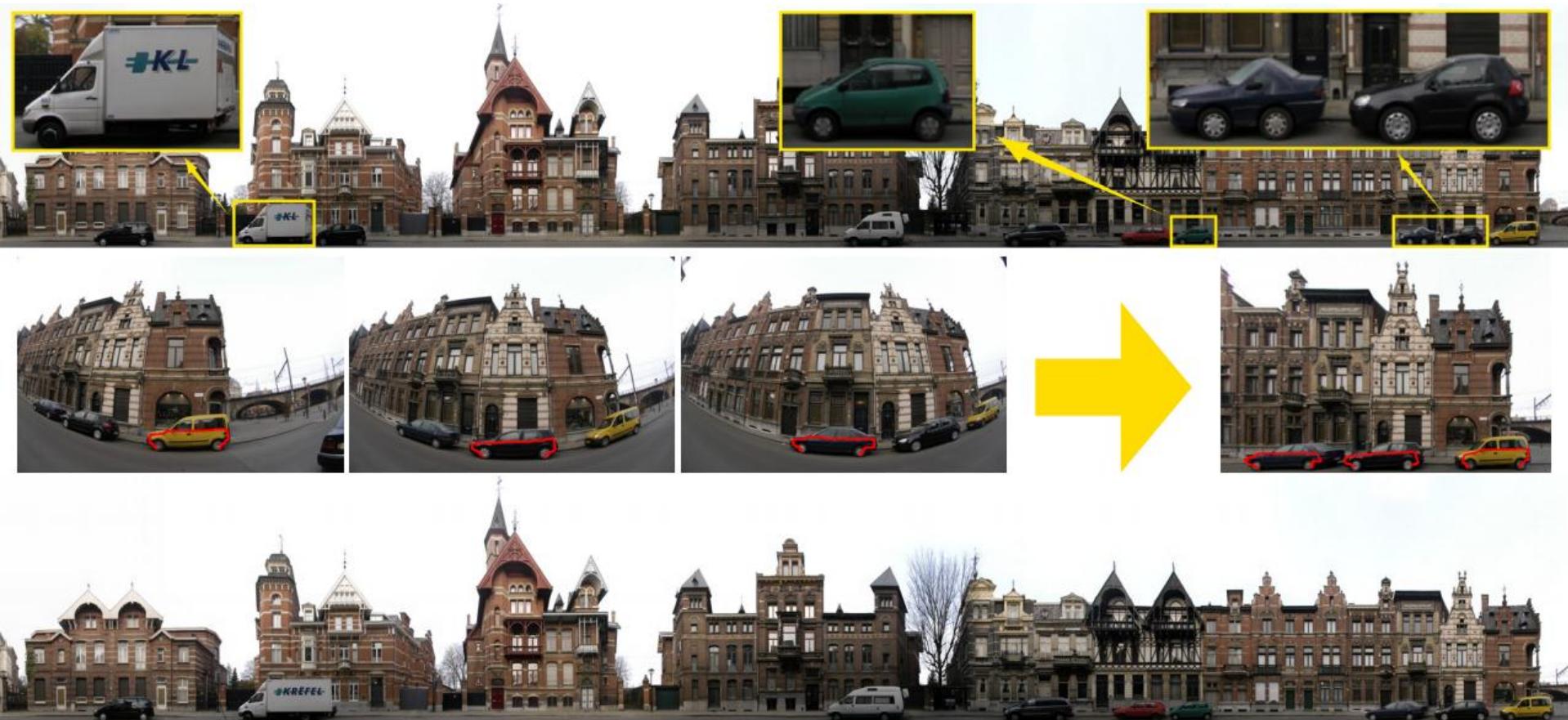
# 多视点全景拼接

## ■ 交互优化

- 车比主平面近，因此被挤压
- 交互指定连续区域

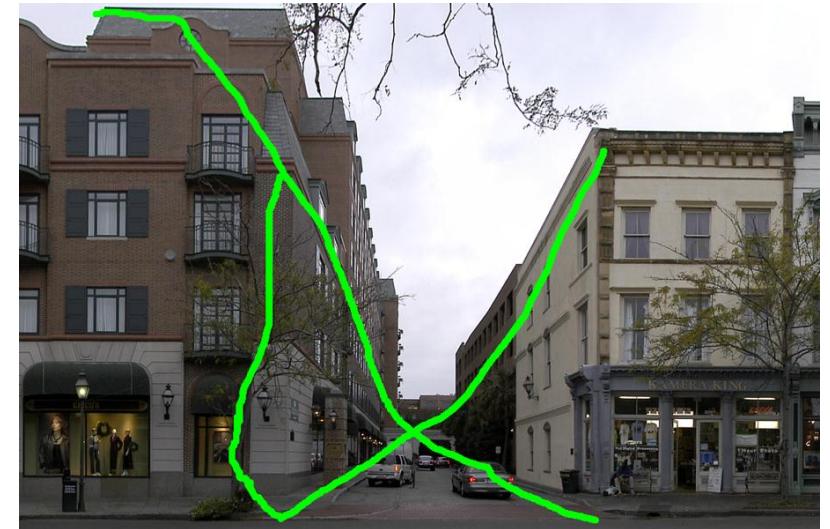
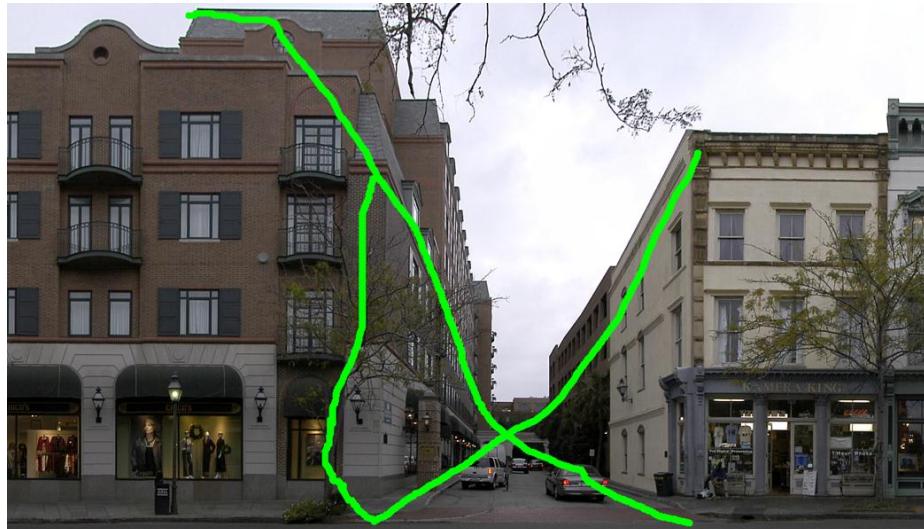


# 多视点全景拼接



# 多视点全景拼接

- 交互优化
  - 选择视图



# 多视点全景拼接

- 交互优化
  - 选择视图



# 多视点全景拼接



# 多视点全景拼接



# 多视点全景拼接



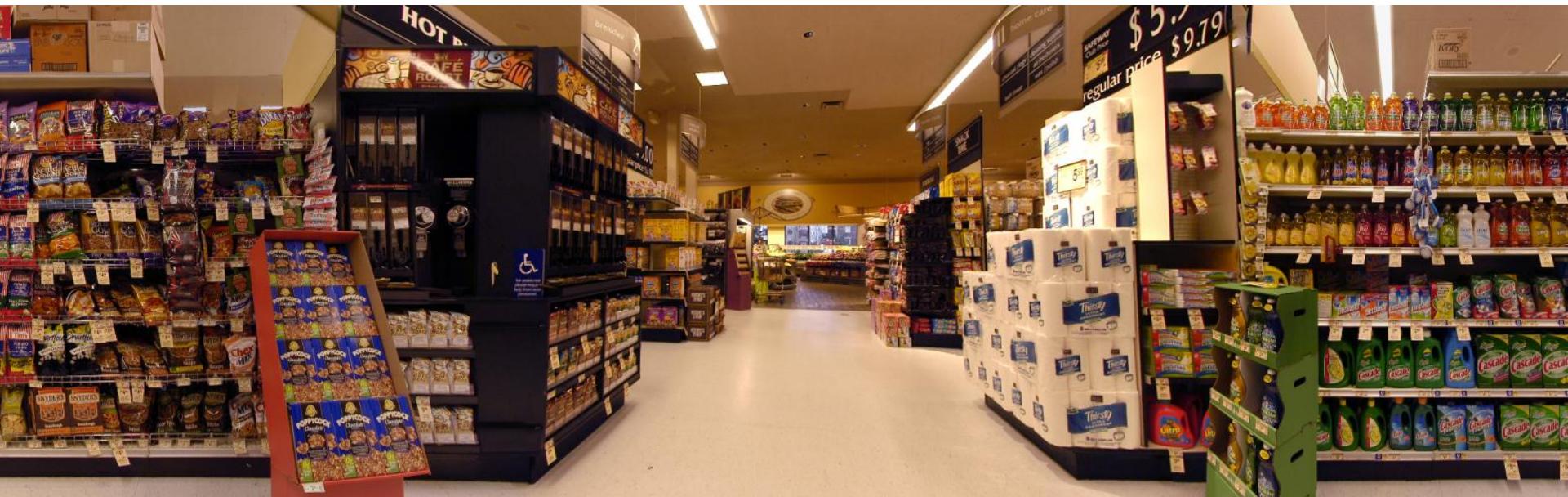
# 多视点全景拼接



# 多视点全景拼接



# 多视点全景拼接



# 多视点全景拼接



# 多视点全景拼接



# 多视点全景拼接



# 多视点全景拼接



# 多视点全景拼接



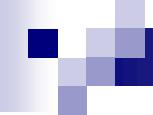
# 全景拼接总结

## ■ 单视点全景拼接

- 特征匹配
- 剔除outlier
- 单应矩阵求解
- 无缝融合 (Multi-band Blend, Graphcut, Poisson Image Editing)

## ■ 多视点全景拼接

- Dual-Homography
- APAP
- Long scene



*Thank you!*