

# 计算机视觉

## 第9章 多图拼接

chenfei314@fzu.edu.cn



## 本章内容



- ❑ 图像拼接概念
- ❑ 仿射变换
- ❑ 透视变换
- ❑ 图像拼接相关方法
- ❑ 图像拼接难点
- ❑ 图像拼接发展现状

## 9.1 图像拼接概念



图像拼接是将多张具有重叠区域（可能是不同时间，不同视角，不同传感器采集）的图片拼接成一张无缝的视野更广的全景图。常常被运用到虚拟现实VR，地质勘测，军事侦查，医学微创手术，航空航天以及视频会议等领域发挥着重要的作用。

在实际的科研以及工程中，经常会遇到超过人眼视角的场景，在近距离无法捕捉到这些广视野的场景，虽然通过增加相机和物体的距离可以捕捉到该场景，但是捕捉到的物体经过放大之后会出现马赛克。因此，进行图像拼接领域的研究是非常必要的。

人类视觉系统的视野大约为 $135 \times 200$ （垂直\*水平）度，而典型的照相机只有 $35 \times 50$ 度的视野。



3

## 9.1 图像拼接概念



在医学图像处理方面，图像拼接常常被用来**辅助诊断**，通过显微镜、超声波、CT等技术获得的图像一般视野都很小，在诊断时，医师需要通过多幅图像的结合来判断病因，图像拼接技术便可以将这些具有共同点的图像拼接到一起，方便医生的会诊工作。因此图像拼接也是医学领域的关键环节。



4

## 9.2 仿射变换



- 线性的变换（保持平行），仿射变换的基本内容



平移



放缩

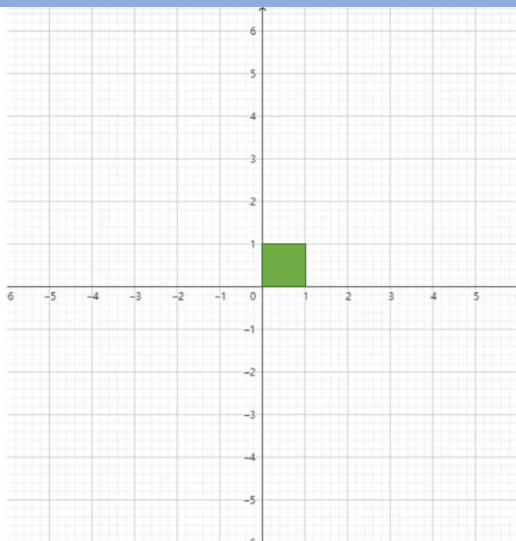


旋转



错切

## 仿射变换



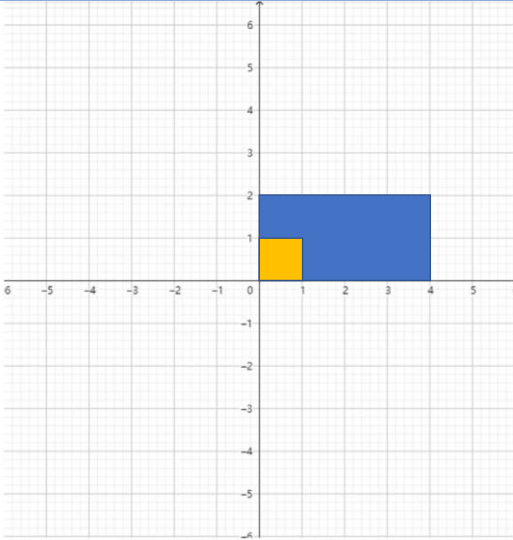
### • 平移 Translate

- 横坐标加上对应的横坐标差
- 纵坐标加上对应的纵坐标差

$$\begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + T_x \\ y + T_y \\ 1 \end{bmatrix}$$



# 仿射变换

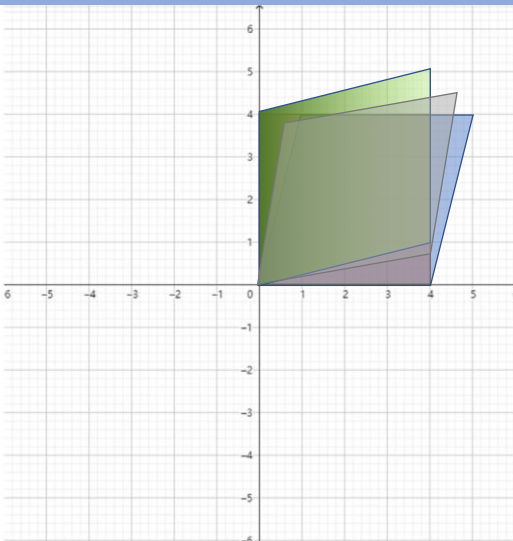


## • 放缩 Scale

- 横坐标乘上对应的横坐标倍数
- 纵坐标乘上对应的纵坐标倍数

$$\begin{bmatrix} T_x & 0 & 0 \\ 0 & T_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} xT_x \\ yT_y \\ 1 \end{bmatrix}$$

# 仿射变换



## • 错切 Shear

- x轴发生改变（取与y轴夹角为  $\varphi$ ）
- y轴发生改变（取与y轴夹角为  $\varphi$ ）

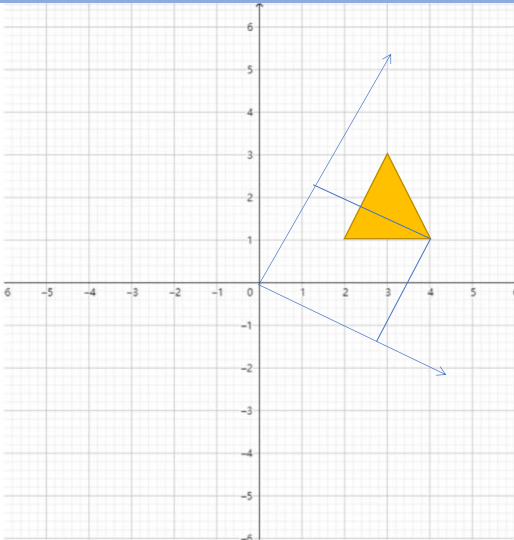
$$\begin{bmatrix} 1 & \tan\varphi & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + \tan\varphi y \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ \tan\varphi & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y + \tan\varphi x \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & \tan\varphi & 0 \\ \tan\varphi & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + \tan\varphi y \\ y + \tan\varphi x \\ 1 \end{bmatrix}$$



# 仿射变换



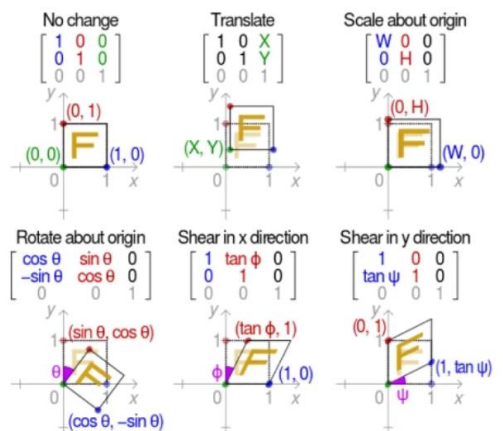
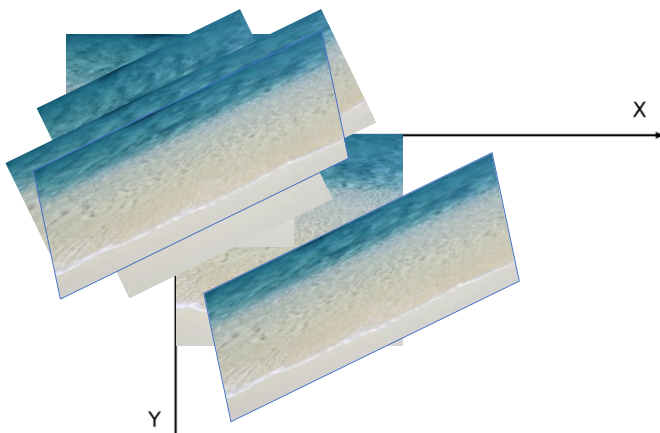
- 旋转 Rotate
- 逆时针旋转  $\varphi$

$$\begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

$$x_1 = x_0 \cos \varphi - y_0 \sin \varphi$$

$$y_1 = x_0 \sin \varphi + y_0 \cos \varphi$$

# 仿射变换



$$\begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \Theta & -\sin \Theta & 0 \\ \sin \Theta & \cos \Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} W & 0 & 0 \\ 0 & H & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \tan \phi & 0 \\ \tan \psi & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & b_1 \\ a_2 & a_3 & b_2 \\ 0 & 0 & 1 \end{bmatrix}$$

平移 Translate      旋转 Rotate      放缩      错切 Shear      平移

# Affine 矩阵



$$\begin{bmatrix} a_0 & a_1 & b_1 \\ a_2 & a_3 & b_2 \\ 0 & 0 & 1 \end{bmatrix}$$

错切  
旋转 放缩

$$\begin{bmatrix} a_0 & a_1 & b_1 \\ a_2 & a_3 & b_2 \\ 0 & 0 & 1 \end{bmatrix}$$

错切  
旋转 放缩 → 线性变换

$$\begin{bmatrix} a_0 & a_1 & b_1 \\ a_2 & a_3 & b_2 \\ 0 & 0 & 1 \end{bmatrix}$$

→ 平移

$$\begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} W & 0 & 0 \\ 0 & H & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \tan\phi & 0 \\ \tan\psi & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix}$$

平移 Translate      旋转 Rotate      放缩      错切 Shear      平移

## 9.3 透视变换



透视变换

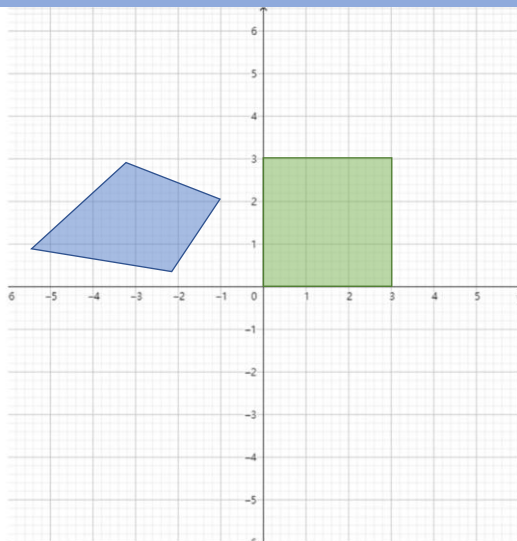
# 透视变换



Perspective



# 透视变换



## 透视变换

- 无法看出旋转了多少，放缩了多少（非线性）
- 暴力求解

线性变换      平移矩阵

$$\begin{bmatrix} a_0 & a_1 & b_0 \\ a_2 & a_3 & b_1 \\ c_0 & c_1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

透视变换

$$(X_0, Y_0) \longrightarrow (M_0, N_0)$$

$$(X_1, Y_1) \longrightarrow (M_1, N_1)$$

$$(X_2, Y_2) \longrightarrow (M_2, N_2)$$

$$(X_3, Y_3) \longrightarrow (M_3, N_3)$$

$$a_0 \quad a_1 \quad b_0$$

$$a_2 \quad a_3 \quad b_1$$

$$c_0 \quad c_1 \quad 1$$

# 透视变换



$$\begin{bmatrix} a_0 & a_1 & b_0 \\ a_2 & a_3 & b_1 \\ c_0 & c_1 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} M_0 \\ N_0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} a_0 & a_1 & b_0 \\ a_2 & a_3 & b_1 \\ c_0 & c_1 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} M_1 \\ N_1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} a_0 & a_1 & b_0 \\ a_2 & a_3 & b_1 \\ c_0 & c_1 & 1 \end{bmatrix} \begin{bmatrix} X_2 \\ Y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} M_2 \\ N_2 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} a_0 & a_1 & b_0 \\ a_2 & a_3 & b_1 \\ c_0 & c_1 & 1 \end{bmatrix} \begin{bmatrix} X_3 \\ Y_3 \\ 1 \end{bmatrix} = \begin{bmatrix} M_3 \\ N_3 \\ 1 \end{bmatrix}$$



$$\begin{cases} a_0X_0+a_1Y_0+b_0=M_0 \\ a_2X_0+a_3Y_0+b_1=N_0 \\ c_0X_0+c_1Y_0+1=1 \end{cases}$$

$$\begin{cases} a_0X_1+a_1Y_1+b_0=M_1 \\ a_2X_1+a_3Y_1+b_1=N_1 \\ c_0X_1+c_1Y_1+1=1 \end{cases}$$

$$\begin{cases} a_0X_2+a_1Y_2+b_0=M_2 \\ a_2X_2+a_3Y_2+b_1=N_2 \\ c_0X_2+c_1Y_2+1=1 \end{cases}$$

$$\begin{cases} a_0X_3+a_1Y_3+b_0=M_3 \\ a_2X_3+a_3Y_3+b_1=N_3 \\ c_0X_3+c_1Y_3+1=1 \end{cases}$$

$$\frac{a_0X_0+a_1Y_0+b_0}{c_0X_0+c_1Y_0+1}=M_0$$

$$\frac{a_2X_0+a_3Y_0+b_1}{c_0X_0+c_1Y_0+1}=N_0$$

$$\frac{a_0X_1+a_1Y_1+b_0}{c_0X_1+c_1Y_1+1}=M_1$$

$$\frac{a_2X_1+a_3Y_1+b_1}{c_0X_1+c_1Y_1+1}=N_1$$

$$\frac{a_0X_2+a_1Y_2+b_0}{c_0X_2+c_1Y_2+1}=M_2$$

$$\frac{a_2X_2+a_3Y_2+b_1}{c_0X_2+c_1Y_2+1}=N_2$$

$$\frac{a_0X_3+a_1Y_3+b_0}{c_0X_3+c_1Y_3+1}=M_3$$

$$\frac{a_2X_3+a_3Y_3+b_1}{c_0X_3+c_1Y_3+1}=N_3$$

$$a_0X_0+a_1Y_0+b_0-c_0X_0M_0-c_1Y_0N_0=M_0$$

$$a_2X_0+a_3Y_0+b_1-c_0X_0N_0-c_1Y_0N_0=N_0$$

$$a_2X_1+a_3Y_1+b_1-c_0X_1N_1-c_1Y_1N_1=N_1$$

$$a_0X_1+a_1Y_1+b_0-c_0X_1M_1-c_1Y_1M_1=M_1$$

$$a_0X_2+a_1Y_2+b_0-c_0X_2M_2-c_1Y_2M_2=M_2$$

$$a_2X_2+a_3Y_2+b_1-c_0X_2N_2-c_1Y_2N_2=N_2$$

$$a_0X_3+a_1Y_3+b_0-c_0X_3M_3-c_1Y_3M_3=M_3$$

$$a_2X_3+a_3Y_3+b_1-c_0X_3N_3-c_1Y_3N_3=N_3$$

$$\begin{bmatrix} X_0 & Y_0 & 1 & 0 & 0 & 0 & -X_0M_0 & -Y_0M_0 \\ 0 & 0 & 0 & X_0 & Y_0 & 1 & -X_0N_0 & -Y_0N_0 \\ X_1 & Y_1 & 1 & 0 & 0 & 0 & -X_1M_1 & -Y_1M_1 \\ 0 & 0 & 0 & X_1 & Y_1 & 1 & -X_1N_1 & -Y_1N_1 \\ X_2 & Y_2 & 1 & 0 & 0 & 0 & -X_2M_2 & -Y_2M_2 \\ 0 & 0 & 0 & X_2 & Y_2 & 1 & -X_2N_2 & -Y_2N_2 \\ X_3 & Y_3 & 1 & 0 & 0 & 0 & -X_3M_3 & -Y_3M_3 \\ 0 & 0 & 0 & X_3 & Y_3 & 1 & -X_3N_3 & -Y_3N_3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ b_0 \\ a_2 \\ a_3 \\ b_1 \\ c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} M_0 \\ N_0 \\ M_1 \\ N_1 \\ M_2 \\ N_2 \\ M_3 \\ N_3 \end{bmatrix}$$

$$\begin{bmatrix} a_0 & a_1 & b_0 \\ a_2 & a_3 & b_1 \\ c_0 & c_1 & 1 \end{bmatrix}$$

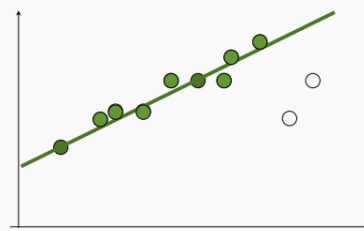
# 透视变换



- 至少四对点（即8个已知参数）
- 可得到8个齐次方程
- 求解8个未知参数
- 传统计算方法：RANSAC

## Simple example: fit a line

Use biggest set of inliers  
Do least-square fit



## RANSAC for estimating homography

RANSAC loop:

Select four feature pairs (at random)

Compute homography  $H$  (exact)

Compute inliers where  $\|p_i' - H p_i\| < \epsilon$

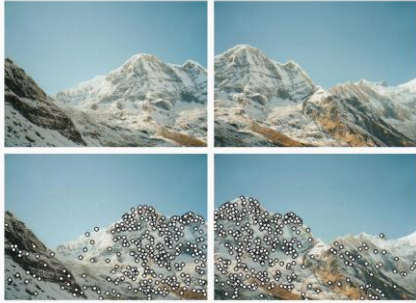
Keep largest set of inliers

Re-compute least-squares  $H$  estimate using all of the inliers

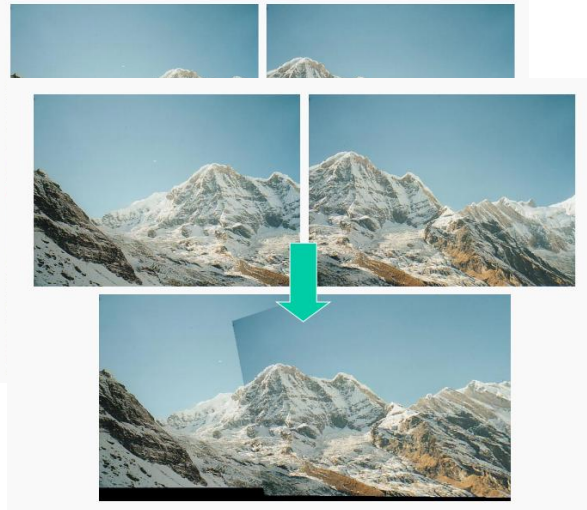
# 图像拼接中的透视变换



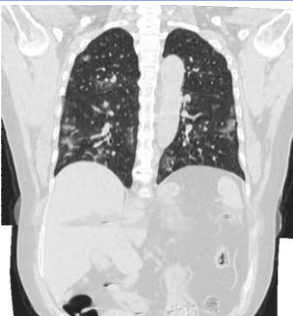
## RANSAC for Homography



1. 特征点检测
2. 特征点匹配
3. 根据特征点进行H估计以及内点的筛选（随机4对种子匹配点）



# 图像拼接中的透视变换



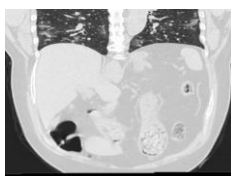
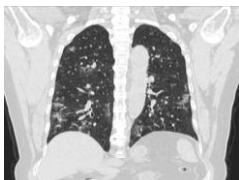
REW



+PERCEPTION SEAM



梯度融合方式替换成线性融合

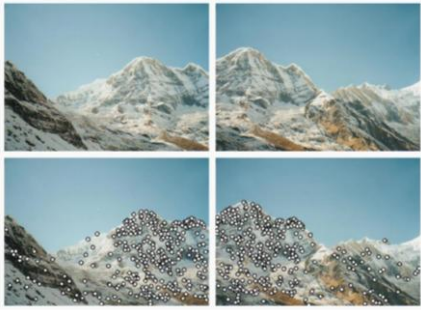


1. landmark based TPS
2. 加强局部显著性避免接缝经过或者尽可能进行相似变换

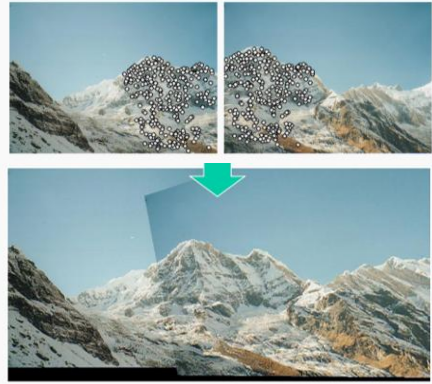
## 9.4 图像拼接相关方法



传统的图像拼接算法流程如下：



- 特征点检测
- 特征点匹配及筛选
- 计算特征点的变换关系（计算H）
- 将目标图像对齐到参考图像上
- 融合变换后的全景图



## 图像拼接



2003, Autostich: 采用一个(全局)单应性对齐, multi-band融合方法



在很长时间内满足大部分场景，  
但是新的问题出现了：视差

2011, DHW Constructing Image Panoramas using Dual-Homography Warping: 将场景划分为背景平面和前景平面，用两个单应性矩阵分别对齐背景和前景



过程：

- 提取两组特征点计算两个单应性矩阵
- 对两个矩阵进行加权

$$\mathbf{H}_{ij} = \omega_{ij}\mathbf{H}_g + (1 - \omega_{ij})\mathbf{H}_d, \quad (1)$$

$$\omega_{ij} = d_g / (d_g + d_d), \quad (3)$$

其中， $d_g$ 和 $d_d$ 是离两个平面中最近特征点的欧式距离

# 图像拼接

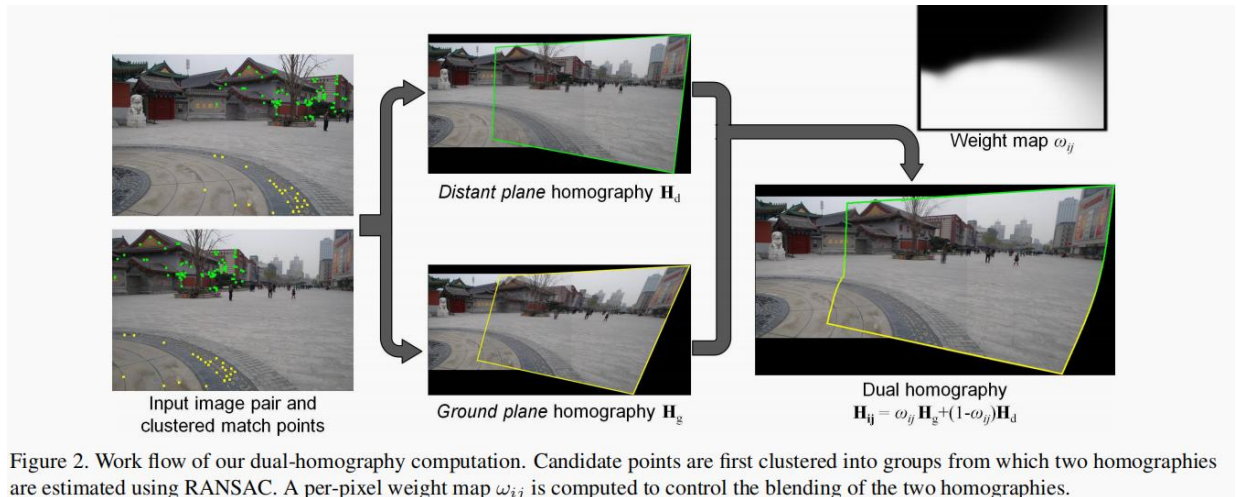


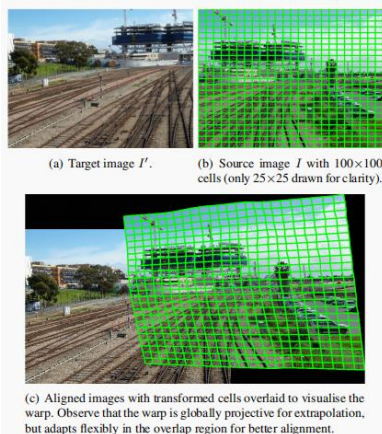
Figure 2. Work flow of our dual-homography computation. Candidate points are first clustered into groups from which two homographies are estimated using RANSAC. A per-pixel weight map  $\omega_{ij}$  is computed to control the blending of the two homographies.

21

# 图像拼接



2013, APAP As-Projective-As-Possible Image Stitching with Moving DLT: 将参考图像和目标图像网格化并进行网格的局部单应性对齐



过程:

- 提取特征点+消除外点
- 计算全局单应性
- 划分网格单元
- 根据权值计算每个网格的单应性

$$\mathbf{h}_* = \operatorname{argmin}_{\mathbf{h}} \sum_{i=1}^N \|w_*^i \mathbf{a}_i \mathbf{h}\|^2 \quad (7)$$

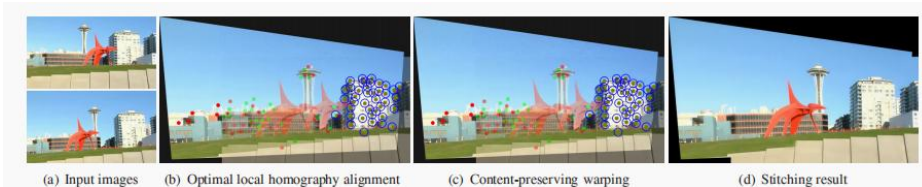
subject to  $\|\mathbf{h}\| = 1$ . The scalar weights  $\{w_*^i\}_{i=1}^N$  change according to  $\mathbf{x}_*$  and are calculated as

$$w_*^i = \exp(-\|\mathbf{x}_* - \mathbf{x}_i\|^2 / \sigma^2). \quad (8)$$

# 图像拼接



2014, Shape-Preserving Half-Projective Warps for Image Stitching



过程:

- 检测特征点消除外点计算最佳单应性
- 利用相似变换约束最佳单应性

$$\hat{H}_s = \arg \min_{H_s} \sum_{C_i} \|H_s C_i - \bar{C}_i\|^2, \text{ where } H_s = \begin{bmatrix} a & -b & c \\ b & a & d \end{bmatrix} \quad (3)$$



利用内容保持进行优化: **局部对齐项+全局对齐项+平滑项**

- **局部对齐项**: 划分网格, 衡量网格对齐:
- **全局对齐项**: 衡量预对齐结果与内容保持后的结果的对齐:
- **平滑项**: 相似变换约束:

$$E = E_p + \alpha E_g + \beta E_s, \quad (8)$$

$$E_p = \sum_{i=1}^n \left\| \sum_{j,k} \alpha_{j,k} \hat{V}_{j,k} - \hat{P}_j \right\|^2, \quad (4)$$

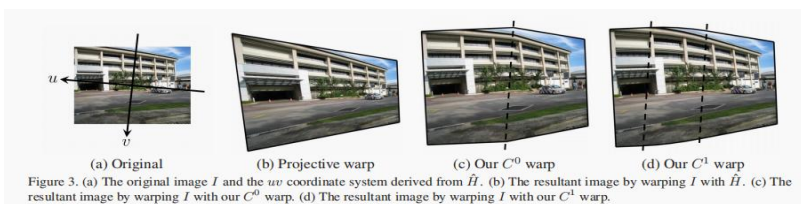
$$E_g = \sum \tau_i \|\hat{V}_i - \bar{V}_i\|^2, \quad (5)$$

$$E_s(\hat{V}_i) = w_s \|\hat{V}_1 - (\hat{V}_2 + u(\hat{V}_3 - \hat{V}_2) + vR(\hat{V}_3 - \hat{V}_2))\|^2, \quad (7)$$

# 图像拼接

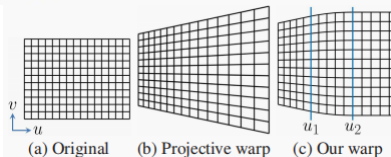


2014, Parallax-tolerant Image Stitching



过程:

- 检测特征点消除外点
- 计算全局单应性
- 计算 **u1和u2**, 将图像分为三个区域
- 对于u1左侧采用单应性变换
- 对于u1~u2采用**单应性至相似变换**的过度变换
- 对于u2右侧采用相似变换
- 利用**全局相似变换**约束, 来确定最佳的u1和u2



$$E_i(u_1, u_2) = \min_{a_i, b_i} \iint_{(x,y) \in \Omega_i} \left\| J_i(x, y; u_1, u_2) - \begin{bmatrix} a_i & -b_i \\ b_i & a_i \end{bmatrix} \right\|_F^2 dx dy \quad (27)$$

# 图像拼接



2015, AANAP Adaptive As-Natural-As-Possible Image Stitching: APAP基础上的优化

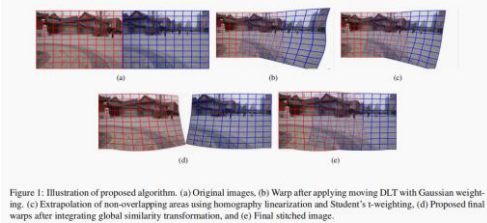
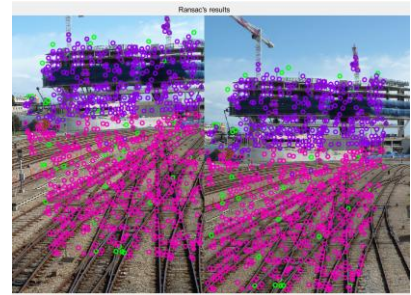


Figure 1: Illustration of proposed algorithm. (a) Original images, (b) Warp after applying moving DLT with Gaussian weighting, (c) Extrapolation of non-overlapping areas using homography linearization and Student's t-weighting, (d) Proposed final warps after integrating global similarity transformation, and (e) Final stitched image.



过程:

- 特征点提取, RANSAC筛选外点, 匹配
- 计算网格单应性
- 从重叠区域到非重叠区域:
  - 重叠区域**单应性线性化** (以此来减少非投影区域投影失真)
  - 计算**全局相似性** (分组匹配特征点, 独立计算多组相似变换, 选取旋转角度最小的作为相似变换)
  - 目标图像的变换约束: 使用**局部单应性**和**全局相似变换**约束:  $\hat{\mathbf{H}}^{(t)} = \mu_h \mathbf{H}_h^{(t)} + \mu_s \mathbf{S}$ , (10)
  - 参考图像的变换约束 (由于目标图像引入了全局相似变换, 会造成两张图像的不对齐):

$$\hat{\mathbf{H}}_r^{(r)} = \hat{\mathbf{H}}_r^{(l)} (\mathbf{H}_l^{(l)})^{-1}$$

# 图像拼接



2013, SEAM-CUT: 解决伪影问题 (定义一个接缝成本, 计算最小成本的标签集)



$$E = \sum_p E_d + \lambda \sum_{(p,q) \in \mathcal{N}} E_s, \quad (1)$$

原理: 找到标签使得上面式子最小化, 前面一项**数据项**代表分配一个标签 $l_p$ 给一个像素 $p$ 所需要的成本, 后面一项是**平滑项**, 分配一对标签 $p$ 和 $l_q$ 给一对像素 $p, q$ 所需要的成本。

# 国内外研究现状



方法	全局单应性	局部单应性 (数目)	网格化	全局相似性	局部相似性	seam-cut	Blend方式
Autostich	√						Multi-band
DHW		√ (2)			√	√	
APAP		√ (10000)	√				Linear
Parallax	√				√	√	Multi-band
SPHP	√			√			Linear
AANAP		√ (10000)	√	√			Linear

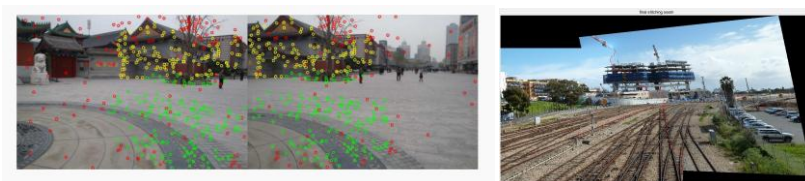
商用的拼接软件的诞生，如Autostitch，微软公司的Image Composite Editor

27

## 9.5 拼接难点



1. 特征点匹配不准确：传统的算法大都采用SIFT特征点及匹配，但是筛选的算法还不能完美的解决不好的匹配的问题，从而导致对齐变换的计算不精准。
2. 大视差环境下变换的选择：在大视差环境下，待拼接图像重叠的对齐效果和非重叠的自然效果是两个相互矛盾的指标。
3. 融合：由于待拼接图像来自不同时刻，亦或者由于拍摄角度的改变导致光线的改变，可能造成最后两张图片的重叠部分可以很明显的看见边缘。同时，由于被捕捉物体的轻微运动可能也会造成最后的伪影。



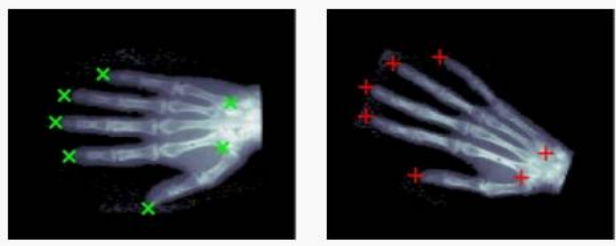
## 拼接难点



根据上述提到的难点，可以针对性的提出相应的解决方案：

- 特征点匹配不准确：**特征点的检测及匹配是图像拼接的第一步，如果该步骤出现大的误差，便会影响到后续的拼接操作。

方案：在医疗图像处理中，医院经常采用标记点进行一些配准操作，考虑使用类似该操作进行关键点的采集，或者对原始检测及筛选后的关键点集进行交互筛选。



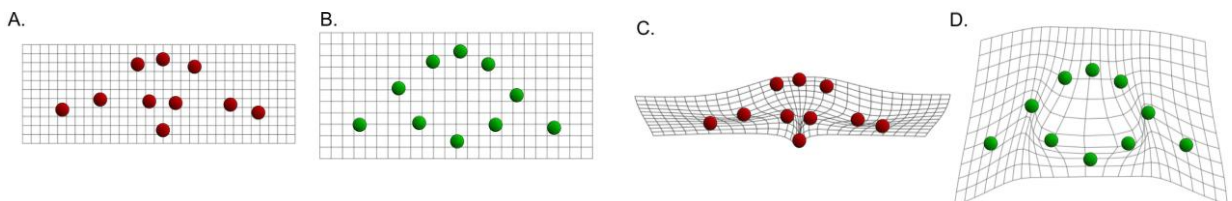
29

## 拼接难点



- 重叠区域对齐性能不佳：**对齐效果取决于特征点的匹配，然而即使特征点匹配十分完美，也会由于视差的原因，导致简单使用单应性变换无法完美的对齐。

方案：薄板样条插值（Thin Plate Spline, TPS）是插值方法的一种，是常用的2D插值方法。假如给定两张图片中一些相互对应的控制点，如何将其中一个图片进行特定的形变，使得其控制点可以与另一张图片的控制点重合。对比传统的单应性对齐方法，TPS可以提供更加自由的对齐方法，也为缺少特征点的区域提供性能更加的映射。



30

## 拼接难点



2. **非重叠区域**  
透视失真。  
相似最佳，  
方案：以多



不自然的  
代表局部

31

## 拼接难点



3. **融合问题**：上述提到由于光线或者物体运动以及视差导致的伪影问题。传统graph-cut只是简单计算全局的能量项，但是还是会造成接缝线经过物体，导致结果不好的问题。（如下图交通信号灯出现两盏）

方案：使用graph-cut算法进行后处理，对接缝能量项进行优化，采用类似文献Perception-based seam cutting for image stitching采用的基于视觉感知的接缝算法避免接缝经过高显著性目标物。



32

## 9.6基于标记点的薄板样条配准



受**医学图像配准任务**的启发，结合**人机交互**进行标记特征点，使用**薄板样条算法**进行配准任务。其优点一方面可以提供更加**准确的特征信息**，即使在特征点不足情况下依然可以完成配准任务。另一方面，采用非刚性变换，提供更加**高效和灵活的对齐性能**，使得拼接结果对齐性能更加强大。

定义薄板的总能量函数为：

$$\varepsilon = \varepsilon_{\Phi} + \lambda \varepsilon_s$$

其中  $\varepsilon$  为薄板的总能量， $\varepsilon_{\Phi}$  为对齐成本， $\varepsilon_s$  为平滑成本， $\lambda$  是平滑成本的权重系数，

$\varepsilon_{\Phi}$  用于衡量经过薄板变换前后所有标记点的距离的总误差：

$$\varepsilon_{\Phi} = \sum_{i=1}^n \mathbf{R}\Phi(p_i) - q_i \mathbf{P}^2$$

平滑成本  $\varepsilon_s$  用于测量薄板的扭曲程度，当薄板的扭曲程度最小时，变换后的目标图像越接近于一个平面，因此，该平滑成本可以表示为：

$$\varepsilon_s = \iint_{\Omega} \left[ \left( \frac{\partial^2 \Phi}{\partial x^2} \right)^2 + 2 \left( \frac{\partial^2 \Phi}{\partial x \partial y} \right) + \left( \frac{\partial^2 \Phi}{\partial y^2} \right)^2 \right] dx dy$$

33

## 引入全局相似性（对齐优化）



问题：两个待拼接视图的**视差**很大的情况下，为了使得人工标记点位置更好的重合且整个目标图像变换后更加趋于一个平面，造成了**非重叠区域的扭曲失真**的现象。

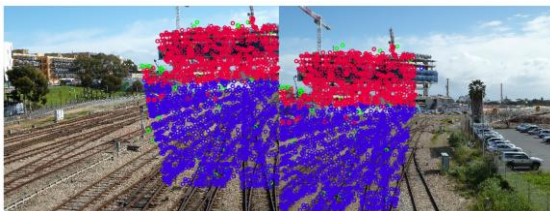
解决：使用**SIFT**算法搜索目标图像和参考图像之间的特征点，使用匹配算法进行配对，使用随机一致性采样**RANSAC(Random Sample Consensus)**算法对特征点进行大概的分类。其中，蓝色特征点和红色特征点为不同组的种子特征点组，绿色特征点为筛选掉的特征点。

计算不同组特征点对应的相似变换：

$$\varepsilon_s = \sum_{i=1}^n \| \mathbf{S}p_i - q_i \|^2$$

选择旋转角度最小的相似变换作为全局相似变换对上一节中薄板样条变换后的坐标进行加权：

$$\begin{cases} X_w = \mu_r X_r + \mu_s X_s \\ Y_w = \mu_r Y_r + \mu_s Y_s \end{cases} \quad \mu_s = 1 - \mu_r$$



34

# 交互切缝算法（伪影处理）针对重叠区



问题：在视差较大的环境下，由于目标图像和参考图像很可能是从不同**光照条件**，不同**时间**，不同**角度**捕捉的，在这种条件下极可能造成拼接结果中的**重叠区域无法完美的重合**，甚至会产生**糟糕的模糊**。

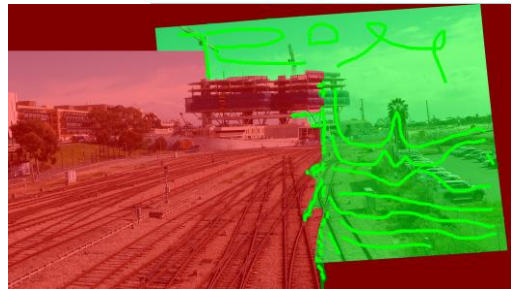
对于切缝的寻找问题通常表示为切缝能量的标签问题，通过最小化该能量函数得到最佳切缝，一条由像素集合组成的切缝的能量由组成该切缝的像素标签的数据成本以及像素之间的平滑成本组成：

$$E = \sum_{p \in \mathbf{P}} E_d(p, l_p) + \lambda \sum_{(p, q) \in \mathbf{N}} E_s(p, q, l_p, l_q)$$

数据成本：
$$E_d(p, l_p) = -\nabla I_{(l_p)}$$

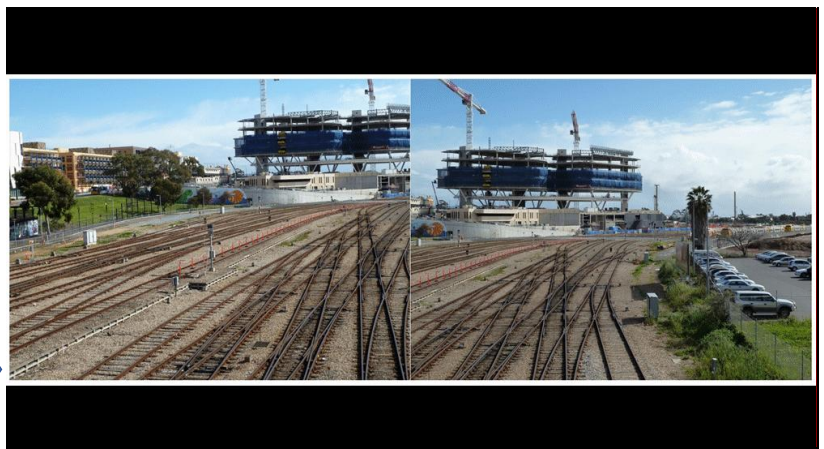
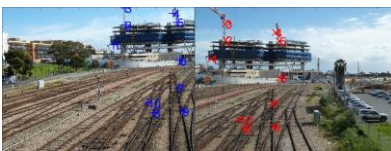
平滑成本：
$$E_s(p, q, l_p, l_q) = |l_p - l_q| (D(p) + D(q))$$

$$D = PI_0 - I_1 P^2 + 2PVI_0 - \nabla I_1 P_2^2$$



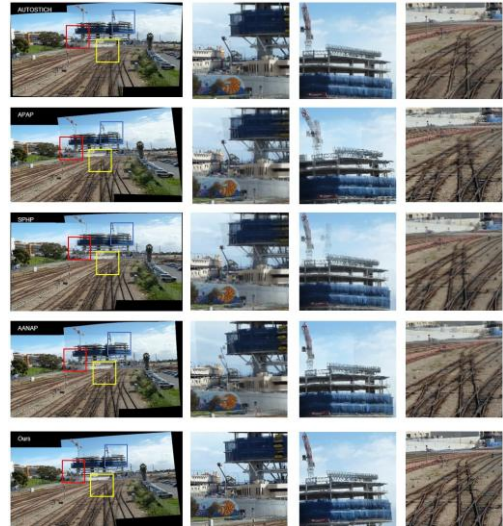
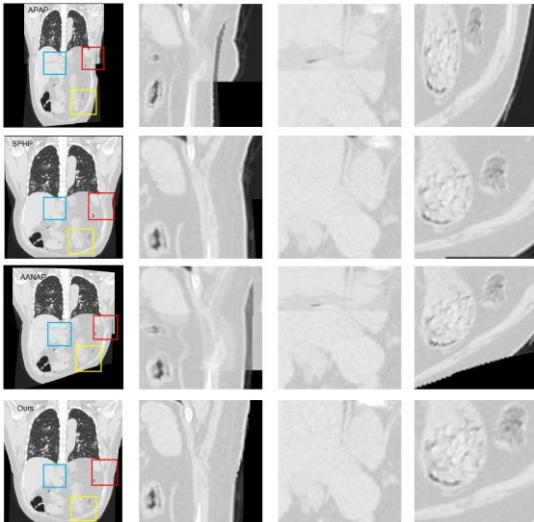
35

## 实验与结果



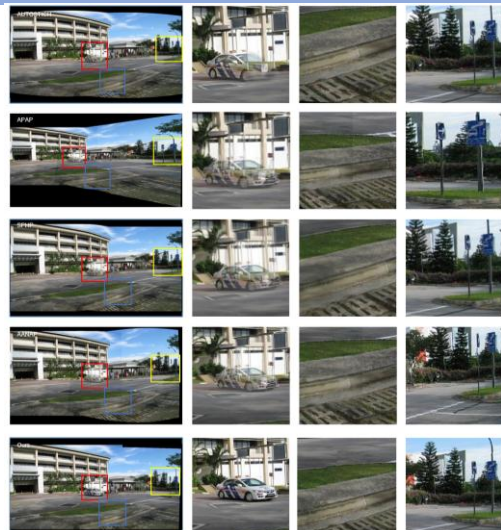
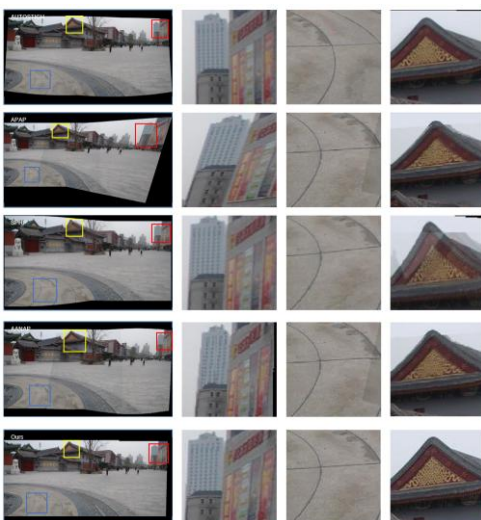
36

# 实验与结果



37

# 实验与结果



38

# 实验与结果



表3-1 不同方法生成结果的平均梯度(单位: %)

算法	Autostitch	APAP	SPP	AANAP	Ours
平均梯度	3.238	3.955	3.437	4.315	<b>5.190</b>



鲁棒性测试：第一行为精确标记点生成的结果，第二行为部分不精确标记点生成的结果



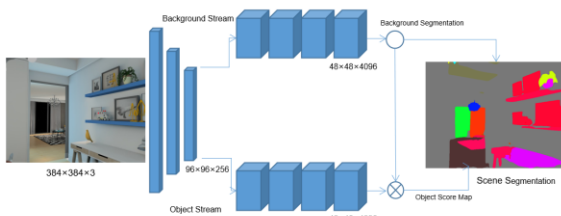
20%高斯噪声拼接结果

39

## 9.7 基于语义分割的特征点分类

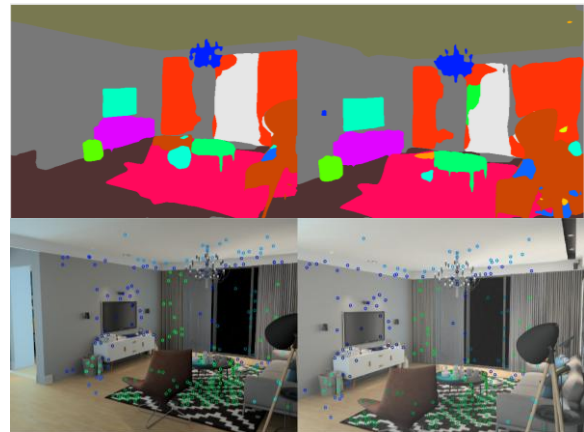


将级联分割模块集成到DilatedNet网络中，其使用的编码器为“resnet101dilated”，译码器为“ppm\_deepsup”，训练的数据集为“ADE20K”数据集。



$$Loss = L_{bg} + L_{ob}$$

将由SIFT检测出来的特征点根据语义分割的结果进行分类并进一步筛选，如右图。



40

# 图像拼接模型（对类别的单应性变换）



主要思想：使用这些不同类别的**特征点组**，得到每个组对应的**单应性变换**，并选取对齐性能最佳的作为全局单应性变换。

对于一对匹配的特征点对应的单应性变换需要满足： $\tilde{p} = H\tilde{q}$

因此，一组点对应的单应性变换可表示为： $\mathbf{h} = \arg \min_{\mathbf{h}} \sum_{i=1}^n \|\mathbf{A}_i \mathbf{h}\|_2^2 = \arg \min_{\mathbf{h}} \|\mathbf{A} \mathbf{h}\|_2^2 \quad \mathbf{A}^T \mathbf{A} = \sum_{i=1}^n \mathbf{A}_i \mathbf{A}_i$

则，不同类别的特征点对应的单应性变换可表示为： $\mathbf{h}_j = \arg \min_{\mathbf{h}} \sum_{i=1}^n \|\mathbf{A}_i \mathbf{h}\|_2^2 = \arg \min_{\mathbf{h}} \|\mathbf{A} \mathbf{h}\|_2^2$

计算不同类别对应的单应性变换的误差： $\varepsilon_{\mathbf{h}_j} = \sum_{i=1}^k \|\mathbf{h}_j \tilde{p}_i - \tilde{q}_i\|^2$

选择**误差最小**的对齐整张图像。

# 语义分割感知的切缝融合（针对切缝）



主要思想：为了进一步减少**重叠区域的伪影**，设置一个语义分割感知的切缝。该切缝基于语义分割结果，计算出符合观感的最佳切缝。

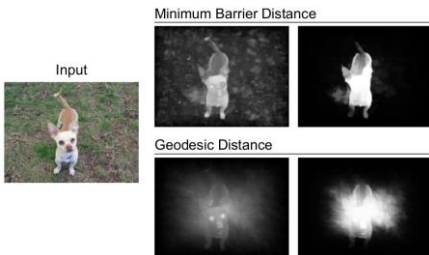
$$E = \sum_{p \in \mathcal{P}} E_d(p, l_p) + \sum_{(p,q) \in \mathcal{N}} \delta(p,q) E_s(p,q, l_p, l_q)$$

结合语义分割的结果，使用平均像素显著性定义显著权重：

$$\delta(p,q) = \begin{cases} 0 & \text{if } p|q \in \text{background} \\ 1 + \frac{\delta(p) + \delta(q)}{2} & \text{otherwise} \end{cases}$$



光照差异下的切缝效果：



# 语义分割感知的切缝融合（针对融合）



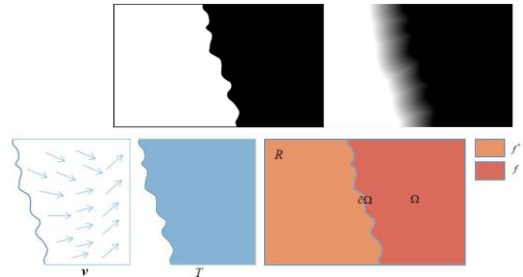
主要思想：为了进一步提高融合效果，将切缝的掩码上任意像素的坐标设为  $(x_i, y_i)$  在所有0像素值点中，设离  $(x_i, y_i)$  拟欧式距离最短的0像素值点的坐标为  $(x_j, y_j)$ 。根据拟欧几里得距离对切缝二值掩码进行羽化处理：

$$M(x_i, y_i) \begin{cases} |x_i - x_j| + (\sqrt{2} - 1)|y_i - y_j|, & \text{if } |x_i - x_j| > |y_i - y_j| \\ (\sqrt{2} - 1)|x_i - x_j| + |y_i - y_j|, & \text{otherwise} \end{cases}$$

$M(x_i, y_i)$  为切缝掩码上像素  $(x_i, y_i)$  的新值。因此，可以得到新的羽化切缝掩码。

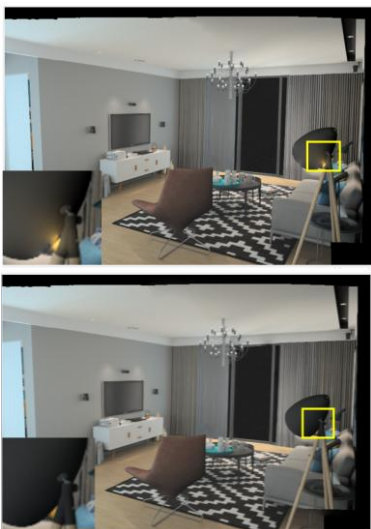
使用泊松融合算法对羽化后的切缝进行融合：  
(尽可能平滑，切缝上的像素值要一致)

$$\begin{cases} \min_f \iint_{\Omega} |\nabla f - v|^2 \\ f|_{\partial\Omega} = f^*|_{\partial\Omega} \end{cases}$$



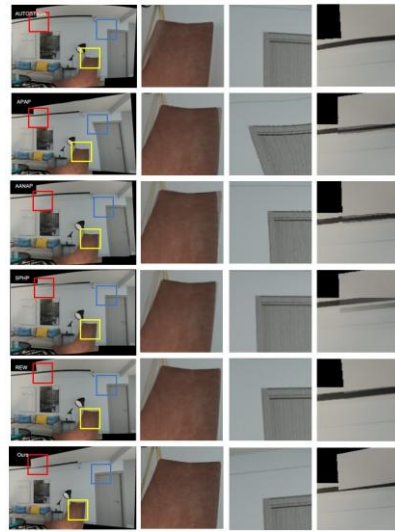
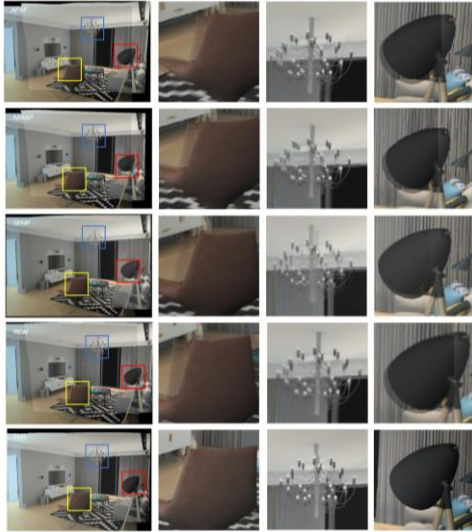
43

# 实验与结果



44

# 实验与结果



45

# 实验与结果

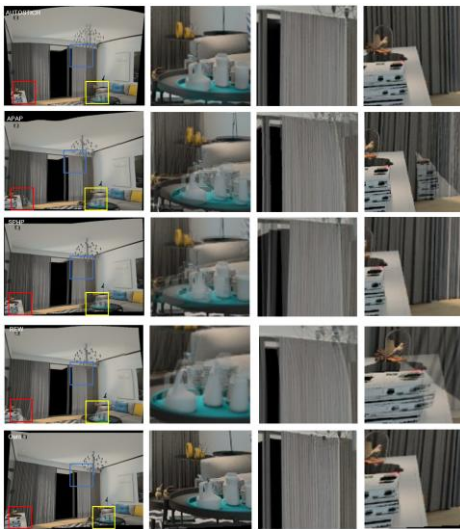


表4-1 不同方法生成结果的平均梯度(单位: %)

算法	Autostitch	APAP	SPHP	AANAP	REW	Ours
平均梯度	2.692	1.751	1.602	1.672	2.452	<b>2.882</b>



20%高斯噪声拼接结果

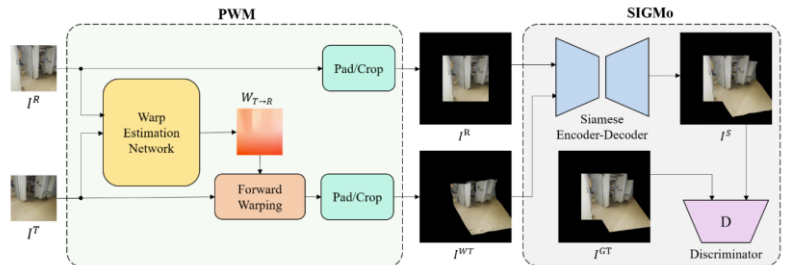
46

# Pixel-wise Deep Image Stitching



- 2021, 彻底抛弃了传统方法依赖的“单应性矩阵”, 转而利用深度学习模型为图像中的每一个像素都计算出一个独立的移动路径 (即像素级扭曲场)。

- 阶段一:
- 像素级精准定位 (PWM模块)
- 解决“怎么移动像素才对”的问题
- 阶段二:
- 智能图像融合 (SIGMo模块)
- 解决“如何融合才能天衣无缝”的问题。



47

# Pixel-wise Deep Image Stitching

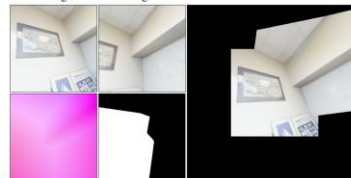


APAP

Pixel-wise Deep Image Stitching



(a) Reference image (b) Target image (c) Sample stitching result



(c) Ground truth pixel-wise warp (d) Ground truth overlap mask

48



李江川摄

Thank you!