



中国科学院大学

University of Chinese Academy of Sciences

Automatic Panoramic Image Stitching using Invariant Features

基于不变性特征的自动全景拼接

MATTHEW BROWN * AND DAVID G. LOWE

汇报人：



目录 Contents

1

特征点匹配

2

图像匹配

3

捆绑调整

4

自动拉直全景图像

5

增益补偿

6

多波段融合

算法的整体流程

Step1：特征点的提取和匹配



(a) Image 1



(b) Image 2

使用SIFT对每一张图像都进行特征的提取，然后找到该特征点和其他图像的特征点之间的匹配关系，论文中使用的是k-d 树的方法，每次找到k个与之最相似的点，即为特征匹配点。



中国科学院大学

University of Chinese Academy of Sciences

算法的整体流程

Step2：图像匹配



(c) SIFT matches 1



(d) SIFT matches 2

根据上一步中特征点匹配的数量，给每一个图像找到 m 张候选图像，为这些匹配的图像对建立一个单应矩阵，最后用概率模型进行验证，剔除错误的图像匹配对，正确的图像匹配对加入到连通集



中国科学院大学

University of Chinese Academy of Sciences

算法的整体流程



(b) Connected components of image matches

对连通集中的图像进行拼接，在拼接时需要进行

- ① 相机参数调整：对于不同的方向和焦距的图像的调整
- ② 增益补偿：对于不同亮度的图像进行调整
- ③ 全景图像调整优化：对于拼接缝隙的调整



算法的整体流程

Step3：调整相机参数----捆绑调整（又叫光束法平差）

根据捆绑调整方法，找到图像对应的相机的各种参数，包括相机的旋转和焦距。



解决所有相机的参数，对于图像串联拼接是很重要的一步，把不同视角的图像调整为同一视角下的图像。



算法的整体流程

Step4：全景图像调整优化---图像拉直

图像拼接之后，可能存在弯曲的现象，可以通过全景图像拉直，进行调整。



(a) Without automatic straightening



(b) With automatic straightening

算法的整体流程

Step5：全景图像调整优化---增益补偿和图像融合

对于图像光照强度差异较大的，可以通过增益补偿，降低重叠区域的差异，通过多波段融合可以使拼接的地方更自然。



算法的整体流程

Step6：全景图像调整优化----增益补偿和图像融合



(c) With gain compensation



(d) With gain compensation and multi-band blending



Step1：特征点匹配

• 提取特征点-----SIFT

- 采用SIFT方法提取特征点的优势有：

- ① 仿射变换不变性
- ② 平移变换不变性
- ③ 旋转不变性
- ④ 放缩不变性
- ⑤ 光照不变性

参考董老师上课的ppt

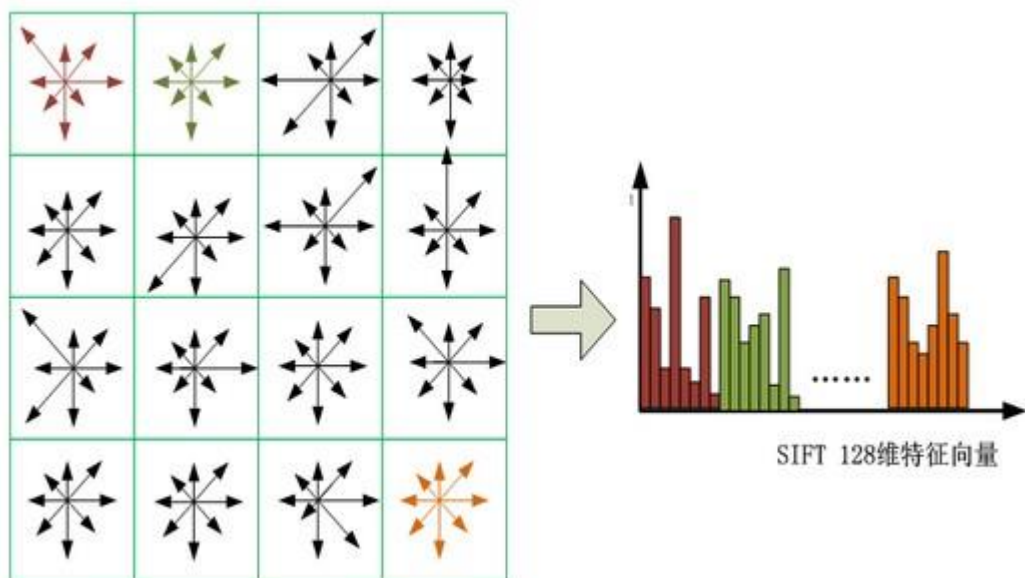
原因：SIFT特征点的提取的点是在尺度空间上，不同的高斯函数的极值点，每一个描述子都有一个确定的尺度和方向。不变性描述子的计算是基于局部梯度的，所以对仿射变换和平移变换具有鲁棒性，又因为消除了偏差和增益，所以对光照也有不变性。

对n张图像进行了特征点的提取



Step1：特征点匹配

- 提取特征点-----SIFT



SIFT将每一个检测到的特征点描述为128维的特征向量

实际操作并不一定是128维，128是SIFT原作者对一系列参数进行调整后得到的结果



Step1：特征点匹配

• 特征点匹配-----k-d树

K-d树实际上就是一个距离函数在高维矢量之间进行的相似性检索问题。

包括2个步骤：

① K-d树的构建：

K-d树的构建是一个递归的过程，每次根据样本点的维度（ x, y, z, \dots ），计算每个维度上，所有的点在该方向上的方差，取方差最大的维度作为切分的方向轴，取该维度上所有样本点的中值作为切分点进行切分，中值点左面的为左子树，右面的为右子树。依次递归。

② K-d树的搜索：

k-d树的搜索是一个回溯的过程，根据一个查询点，通过二叉搜索找到一个与之相似的点L，以L为起点进行回溯，判断相似点L的父节点F是否与之更相似，如果是，继续判断F的另一子树上的节点与之是否更相似，否则不用再验证其他子树。再把F当做起始点，回溯它的父节点及其子节点是否更相似，依次类推。

每两个图像之间，都找到了特征匹配点



中国科学院大学

University of Chinese Academy of Sciences

Step1：特征点匹配

- 特征点匹配-----k-d树

最直观的方法就是线性扫描的方法，时间复杂度为 $O(n^2)$ ，但是这样的没有利用数据呈现聚类形态的特点，导致计算的时间复杂度太高。

使用k-d数大大降低了特征点匹配的时间复杂度，时间复杂由原来的 $O(n^2)$ 降为 $O(n\log n)$ 。【n表示的是所有的图像提取的特征点的总和】

两两图像之间，都确定了特征匹配点的数量，为了简化计算，取数量最多的m张图像，作为后续图像匹配的候选图像（论文中 $m=6$ ）



Step2：图像匹配

- 图像匹配-----RANSAC估计单应阵

单应(Homography)：是射影几何中的概念，又称为射影变换。它把一个射影平面上的点(三维齐次矢量)映射到另一个射影平面上，并且把直线映射为直线，具有保线性质。总的来说，单应是关于三维齐次矢量的一种线性变换， $x_1 = Hx_2$

为什么要使用单应？

拍摄的各个图像，由于视角不同，所以无法进行直接的拼接，需要先使用单应，从一幅图像的视角转换到另一幅图像的视角，即把一张图像上的像素坐标转换为另一幅图像上的像素坐标。



Step2：图像匹配

- 图像匹配-----RANSAC估计单应阵

如何计算单应矩阵？

通过4对对应点的坐标，即8点法通过DLT直接线性求解，可以计算出两幅图像之间的单应矩阵。

公式为： $u=Hv$, H 是一个3*3的齐次矩阵。

$$\begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = H \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix}$$

具体计算过程参考申老师上课的PPT



Step2：图像匹配

• 图像匹配-----RANSAC估计单应阵

RANSAC：根据一组包含异常数据（外点）的样本数据集，计算出数据的模型参数，并得到有效样本数据（内点）的算法。

优势：对外点具有很好的鲁棒性

算法1：利用RANSAC方法估计两个图像的单应阵：

① 从两个图像中任意选取4对对应点，计算得到单应阵 H

② 对于图像中剩余的点

通过单应阵 H 的转换，得到一个转换点，计算转换点与另一图像对应特征点的距离，如果距离小于一定阈值，为内点，否则为外点。

③ 统计内点的数量 n

算法执行结果： H 和 n

重复执行500次上述算法，取内点数量最多的单应阵作为最终的单应阵，而内点数量小于某阈值，则认为不匹配



中国科学院大学

University of Chinese Academy of Sciences

Step2：图像匹配

• 图像匹配-----RANSAC评估单应阵

重复500次迭代，得到正确的单应阵的概率：

$$p(\text{H 是正确的概率}) = 1 - (1 - p^r)^n$$

P:选择的点对是内点的概率，取 $p=0.5$

r:选择的内点对的数量，取 $r=4$

n:n轮迭代之后，取 $n=500$

① p^r : r对都是内点的概率

② $1 - p^r$:有一次不是内点，即一次计算单应阵错误的概率为。

③ 重复了n次实验,n次计算单应阵全部错误的概率为 $(1 - p^r)^n$ ，
那么至少有一次计算正确的概率为 $1 - (1 - p^r)^n$

500轮之后找到单应阵的概率约等于 $1 - (1 - p^4)^{500}$ ，即 $1 - 10^{-14}$



Step2：图像匹配

- 图像匹配-----概率模型进行验证

在这一步骤之后，就可以确定图像之间的匹配情况，但是对于一些图相匹配对是否是正确的图像匹配对，我们还有使用概率模型，进行验证。



Step2：图像匹配

- 图像匹配-----概率模型对图像匹配验证

使用概率模型进行验证，剔除生成的错误图像匹配对，概率模型如下：

n_f ：是两张图像的重叠区域特征点的数量

n_i ：该图像上内点的数量

m ：正确或错误的图相匹配， $m=0$ （错误）或 1 （正确）

$f(i)$ ：第 i 次的特征匹配是内点还是外点， $f(i)=0$ 或 1

p_1 ：在图像匹配正确时，一次特征匹配为内点的概率

p_0 ：在图像匹配错误时，一次特征匹配为内点的概率



Step2：图像匹配

- 图像匹配-----概率模型对图像匹配验证

假设每一次特征匹配实验是独立的伯努利实验，那么

$P(f(1:nf)|m=1)=B(n_i;nf,p_1)$,nf次伯努利实验，表示的是在 $m=1$ 的情况下，一次实验中特征匹配的是内点的概率为 p_1 ，那么进行了nf次实验之后，有 n_i 次实验取得取得内点的概率。
同理：

$P(f(1:nf)|m=0)=B(n_i;nf,p_0)$,nf次伯努利实验，表示的是在 $m=0$ 的情况下，一次实验中特征匹配的是内点的概率为 p_0 ，那么进行了nf次实验之后，有 n_i 次实验取得取得内点的概率。

Tips :
$$B(x; n, p) = \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x}$$



Step2：图像匹配

- 图像匹配-----概率模型对图像匹配验证

那么在使用了 n_i 次内点进行特征匹配的情况下，图像匹配正确的概率为：（贝叶斯公式）

$$\begin{aligned} p(m=1|f^{(1:n_f)}) &= \frac{p(f^{(1:n_f)}|m=1)p(m=1)}{p(f^{(1:n_f)})} \\ &= \frac{1}{1 + \frac{p(f^{(1:n_f)}|m=0)p(m=0)}{p(f^{(1:n_f)}|m=1)p(m=1)}} \end{aligned}$$

设定一个阈值 P_{\min} ，当概率大于 P_{\min} 时，我们认为图像的匹配是正确的。



Step2：图像匹配

- 图像匹配-----概率模型对图像匹配验证

也就是说当重叠区域中的内点数量达到所有全部特征点数量的多少时，才算得上是正确的图相匹配结果，否则就认为图像匹配是错误的。

$$n_i > \alpha + \beta n_f$$



Step3：捆绑调整

- 相机参数调整

在前面的步骤中，我们只进行了每两张图像之间的视角的调整，而没有进行全局的图像调整，在这一步骤中，我们对全局的图像，进行了一个视角的调整，使得全局的重投影误差最小。



Step3：捆绑调整

- 相机参数调整

损失函数使用的是huber函数：

$$h(\mathbf{x}) = \begin{cases} |\mathbf{x}|^2, & \text{if } |\mathbf{x}| < \sigma \\ 2\sigma |\mathbf{x}| - \sigma^2, & \text{if } |\mathbf{x}| \geq \sigma \end{cases}.$$

因为huber损失函数相比于平方误差损失函数来说，对数据中的游离点不敏感，是一种对外点更为鲁棒的损失函数。并且，由于huber函数同时包含了L2范式优化对内点收敛速度快的优点和L1范式对于外点更鲁棒的优点，因此能更快的找到最优解。



Step3：捆绑调整

- 相机参数调整

对损失函数求导，找到最优解：（链式求导法则）

$$\frac{\partial e}{\partial \theta_{i1}} = \frac{\partial e}{\partial h} \times \frac{\partial h}{\partial \theta_{i1}} = \frac{\partial e}{\partial h} \times \frac{\partial h}{\partial r} \times \frac{\partial r}{\partial \theta_{i1}}$$

$$\frac{\partial r}{\partial \theta_{i1}} = - \frac{\partial p}{\partial \theta_{i1}}$$

因为 u_i^k 中没有 θ_i 项，所以求导等于0

$$\frac{\partial \mathbf{p}_{ij}^k}{\partial \theta_{i1}} = \frac{\partial \mathbf{p}_{ij}^k}{\partial \tilde{\mathbf{p}}_{ij}^k} \frac{\partial \tilde{\mathbf{p}}_{ij}^k}{\partial \theta_{i1}}$$



Step3：捆绑调整

• 相机参数调整

$\partial \tilde{\mathbf{p}}_{ij}^k$ 是 $\partial \mathbf{p}_{ij}^k$ 的齐次坐标形式，根据矩阵求导公式可得

$$\frac{\partial \mathbf{p}_{ij}^k}{\partial \tilde{\mathbf{p}}_{ij}^k} = \frac{\partial \begin{bmatrix} x/z & y/z \end{bmatrix}}{\partial \begin{bmatrix} x & y & z \end{bmatrix}} = \begin{bmatrix} 1/z & 0 & -x/z^2 \\ 0 & 1/z & -y/z^2 \end{bmatrix}$$

$$\frac{\partial \tilde{\mathbf{p}}_{ij}^k}{\partial \theta_{i1}} = \mathbf{K}_i \frac{\partial \mathbf{R}_i}{\partial \theta_{i1}} \mathbf{R}_j \mathbf{K}_j^{-1} \tilde{\mathbf{u}}_j^l$$

$$\frac{\partial \mathbf{R}_i}{\partial \theta_{i1}} = \frac{\partial}{\partial \theta_{i1}} e^{[\boldsymbol{\theta}_i]_{\times}} = e^{[\boldsymbol{\theta}_i]_{\times}} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

\mathbf{R} 是相机的旋转矩阵，
它的指数表示形式为：

$$\mathbf{R}_i = e^{[\boldsymbol{\theta}_i]_{\times}}, \quad [\boldsymbol{\theta}_i]_{\times} = \begin{bmatrix} 0 & -\theta_{i3} & \theta_{i2} \\ \theta_{i3} & 0 & -\theta_{i1} \\ -\theta_{i2} & \theta_{i1} & 0 \end{bmatrix}.$$



Step4：自动图像拉直

世界坐标系：

- 世界坐标系是为了描述相机的位置而被引入的，任何维的旋转可以表示为坐标向量与合适的方阵的乘积。平移向量是第一个坐标原点与第二个坐标原点的偏移量；
- 在世界坐标系下，有两个重要的参数：旋转矩阵 R 和平移向量 T

相机坐标：

以摄像机光心为原点（在针孔模型中也就是针孔为关心）， z 轴与光轴重合也就是 z 轴指向相机的前方（也就是与成像平面垂直）， x 轴与 y 轴的正方向与物体坐标系平行。



Step4：自动图像拉直

在前面以及找到了相机之间的旋转，但是相对于世界坐标系来说，我们还不确定，如果直接假定 $R=I$ （单位阵），那么就会造成拼接的图像呈现波浪形，所以要对图像进行拉直操作。

常识来说，相机相对于地平面都是平行的，没有相机是弯曲的相机，所以即使相机倾斜或者旋转，相机 X 向量，总是位于一个水平面上，所以求和它的协方差矩阵相乘为0的向量，这就是向上的方向向量 U ，即

$$\left(\sum_{i=1}^n \mathbf{X}_i \mathbf{X}_i^T \right) \mathbf{u} = \mathbf{0}.$$

再进行一个全局的旋转，使得弯曲的全景图像边直。



Step5：增益补偿

- 光亮调整----增益补偿

在前面的几个步骤中，处理的都是图像的几何参数，如不同的方向和不同的焦距的情况下，如何进行全景拼接，在接下来的步骤中，是对图像的不同光亮进行处理。



Step5 : 增益补偿

- 光亮调整----增益补偿

定义了一个损失函数：

$$e = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \sum_{\substack{\mathbf{u}_i \in \mathcal{R}(i, j) \\ \tilde{\mathbf{u}}_i = \mathbf{H}_{ij} \tilde{\mathbf{u}}_j}} (g_i I_i(\mathbf{u}_i) - g_j I_j(\mathbf{u}_j))^2$$

$\mathcal{R}(i, j)$ 表示的是图像i和图像j的重叠区域

\mathbf{u}_i 表示的是图像j上的点经过单应阵转换之后在图像i上的点

损失函数表示的是重叠区域内所有像素点，在投影转换之后和投影转换之前的增益损失和， e 越大，表示两个图像的光亮强度差异越大。



Step5：增益补偿

- 光亮调整----增益补偿

其中 $I_i(u_i)$ 可以近似为：

$$\bar{I}_{ij} = \frac{\sum_{\mathbf{u}_i \in \mathcal{R}(i,j)} I_i(\mathbf{u}_i)}{\sum_{\mathbf{u}_i \in \mathcal{R}(i,j)} 1}.$$

转化为正态分布：

$$e = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n N_{ij} \left((g_i \bar{I}_{ij} - g_j \bar{I}_{ji})^2 / \sigma_N^2 + (1 - g_i)^2 / \sigma_g^2 \right)$$

对e求导，找到损失函数最小时增益g的取值。



Step5：增益补偿

- 光亮调整----增益补偿

对于光照差异不大的，我们在这一步之后，得到的拼接图像，接缝处不太明显，但是对于光照差异大的，它的拼接处还是会比较明显的缝隙。



(b) Without gain compensation



(c) With gain compensation



Step6：多波段融合

- 优化调整----多波段融合

线性融合方法：

$$I^{linear}(\theta, \phi) = \frac{\sum_{i=1}^n I^i(\theta, \phi) W^i(\theta, \phi)}{\sum_{i=1}^n W^i(\theta, \phi)}$$

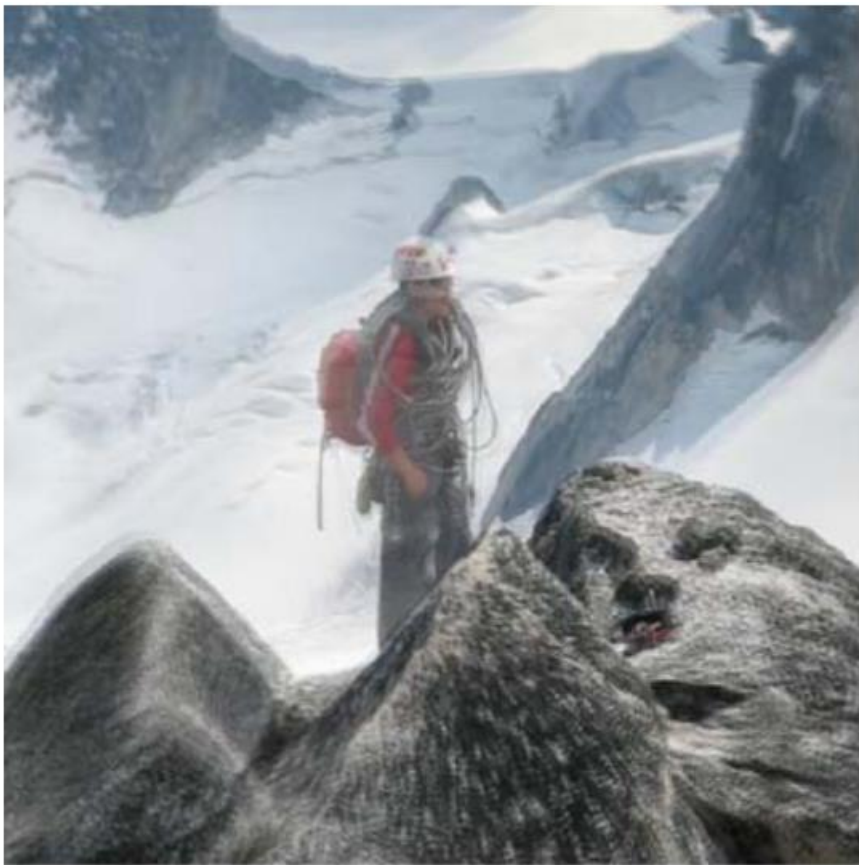
线性融合方法，就是基于加权平均的融合方法，该方法速度快，图像结构完整性保障，但是遇到融合图像像素差异较大的情况或定位有误差（如人在移动）会产生严重的拼接感。

而多波段融合的方法，对高频率区域融合采用较短的波长，低频率区域融合采用较大的波长，可以很好地解决拼接缝隙明显的问题。所以本论文中使用了多波段融合的方法。



Step6：多波段融合

- 优化调整----多波段融合



(a) Linear blending



(b) Multi-band blending



中国科学院大学

University of Chinese Academy of Sciences

Step6：多波段融合

• 优化调整----多波段融合

多波段融合具体实现流程：

- ① 找到最大权重图谱
- ② 利用高斯卷积生成多波段的图像——高斯金字塔
- ③ 从高斯金字塔生成拉普拉斯金字塔
- ④ 在拉普拉斯金字塔每一个波段上进行线性融合
- ⑤ 对融合之后的每个频段的图像，进行逆变换，得到融合之后的图像



Step6：多波段融合

- 优化调整----多波段融合

step1：找到最大权重图谱

$$W_{max}^i(\theta, \phi) = \begin{cases} 1 & \text{if } W^i(\theta, \phi) = \arg \max_j W^j(\theta, \phi) \\ 0 & \text{otherwise} \end{cases}$$

对于 (θ, ϕ) 这个点，如果它在图像i是的权重取值最大，那么对应的最大权重图谱上，该位置的值为1，否则为0



Step6：多波段融合

- 优化调整----多波段融合

step2：找到多波段图像

$$B_{\sigma}^i(\theta, \phi) = I^i(\theta, \phi) - I_{\sigma}^i(\theta, \phi)$$

$$I_{\sigma}^i(\theta, \phi) = I^i(\theta, \phi) * g_{\sigma}(\theta, \phi)$$

对 $I^i(\theta, \phi)$ 高斯卷积，得到高斯金字塔的第0级图像 $I_{\sigma}^i(\theta, \phi)$ ，
做差分得到拉普拉斯金字塔的第0级图像 $B_{\sigma}^i(\theta, \phi)$ ，依次类推，
得到整个高斯金字塔和拉普拉斯金字塔。



Step6：多波段融合

- 优化调整----多波段融合

step3：线性融合

$$I_{k\sigma}^{multi}(\theta, \phi) = \frac{\sum_{i=1}^n B_{k\sigma}^i(\theta, \phi) W_{k\sigma}^i(\theta, \phi)}{\sum_{i=1}^n W_{k\sigma}^i(\theta, \phi)}.$$

对拉普拉斯金字塔每一个波段，进行线性融合，其中权重是最大权重图谱和高斯卷积之后的结果

$$W_{\sigma}^i(\theta, \phi) = W_{max}^i(\theta, \phi) * g_{\sigma}(\theta, \phi)$$



Step6：多波段融合

- 优化调整----多波段融合

step4：图像融合

在融合时，先对低波段的图像进行上采样，得到高分辨率的图像，与底层的图像进行融合，这样保证了低波段的图像使用了较大的波长，高波段的图像使用了较小的波长。

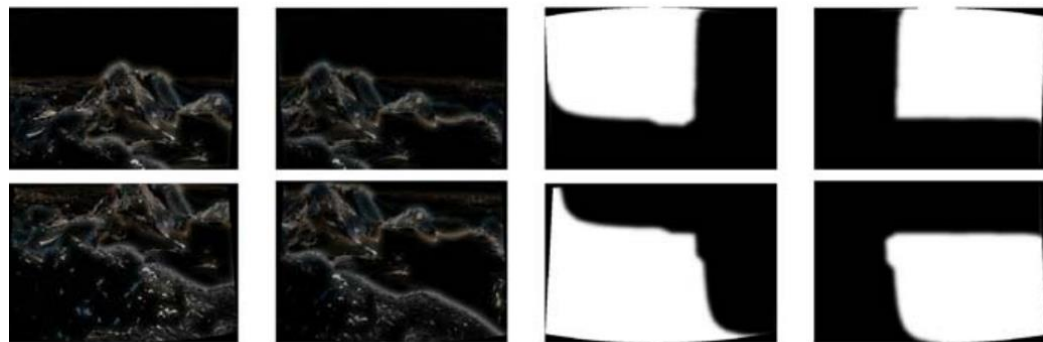


Step6 : 多波段融合

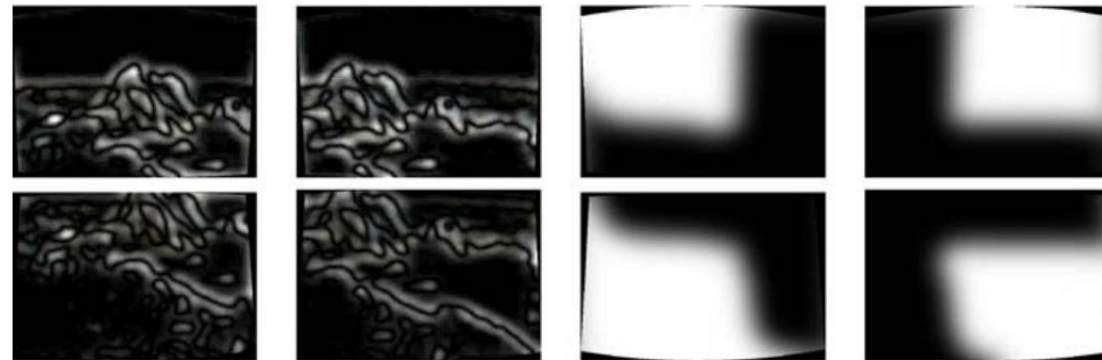
- 优化调整----多波段融合



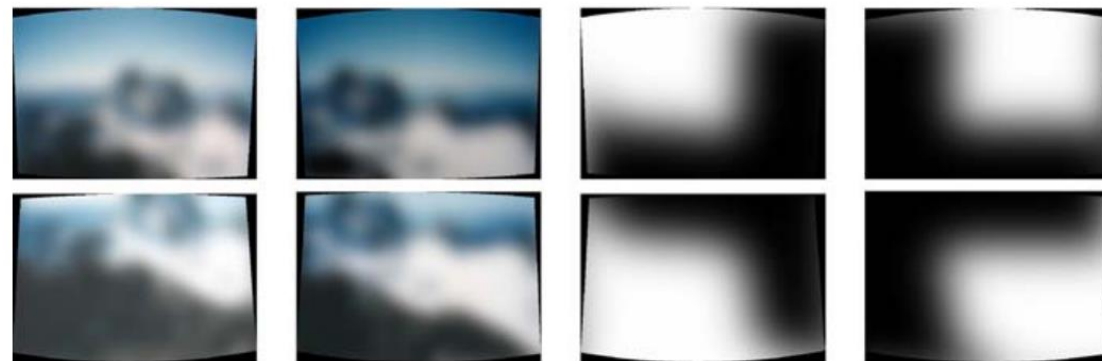
(a) Original images and blended result



(b) Band 1 (scale 0 to σ)



(c) Band 2 (scale σ to 2σ)



(d) Band 3 (scale lower than 2σ)



全景拼接全过程结束

接下来是结果展示



中国科学院大学

University of Chinese Academy of Sciences

结果展示

对于图像有放缩和旋转的情况下，也能进行良好的拼接



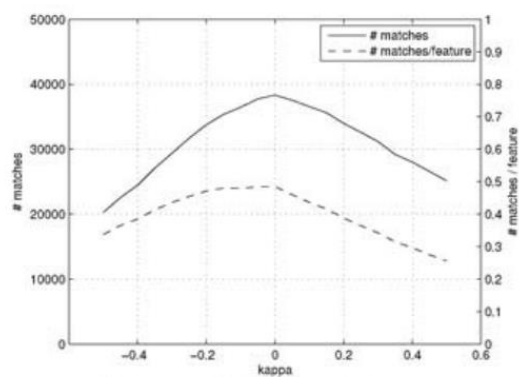
(a)



(b)

结果展示

对于图像有径向畸变情况下，也能进行良好的拼接



(a) Number of feature matches vs κ



(b) Example of stitching 2 images with $\kappa = -0.5$ (the full test set contains 44 images of this scene)



(c) Detail from the previous figure showing misregistration



(d) $\kappa = -0.5$



(e) $\kappa = -0.25$



(f) $\kappa = 0$



(g) $\kappa = 0.25$



(h) $\kappa = 0.5$



结果展示

对于图像运动物体和较大的光照变化的情况下，也能进行良好的拼接





中国科学院大学

University of Chinese Academy of Sciences

感谢!