

说明：

下载 pycharm, 在 windows 下运行 所有代码中均已经配好路径，并附带数据集，直接打开对应的.py 文件运行。

如果 X 是一个 m 行 n 列的数值矩阵， $\text{rank}(X)$ 是 X 的秩，假如 $\text{rank}(X)$ 远小于 m 和 n ，则我们称 X 是低秩矩阵。低秩矩阵每行或每列都可以用其他的行或列线性表出，可见它包含大量的冗余信息。利用这种冗余信息，可以对缺失数据进行恢复，也可以对数据进行特征提取。

PCA 方法鲁棒性不佳是由于矩阵的噪声并不完全是高斯噪声。对应到视频序列中就是，长时间的静止视频中每帧的图片相关性极高，而在有物体运动时，往往是部分像素有极大的变化，但是变化的像素较少。这也就是说，视频中图像可以分成相关性极高的背景以及少量像素的前景图像。即低秩部分以及稀疏部分。这是 RASL 优于 PCA 的部分，即前景图像对矩阵的分解影响较小，如下图对镜头中的动目标检测。（虽然论文中提到了，矩阵不是特别稀疏时也有很好的性能）



除此之外低秩矩阵还应用于鲁棒联合图像对齐，Peng 等人利用低秩矩阵分解的原理设计出了一种基于低秩稀疏分解的鲁棒联合图像对齐方法(Robust alignment by sparse and low rank decomposition, RASL)。用于对齐一组本质上线性相关的图像。因为只有一组线性

相关的图像在构成矩阵后，构造矩阵才具有低秩的特性。此方法的主要原理就是在低秩矩阵分解的过程中加入图像变换参数，使得经过变换后的图像构成的矩阵具有低秩结构，并通过低秩分解将其分解为低秩矩阵部分和稀疏矩阵部分，其中低秩矩阵部分代表对齐且去除噪声后的图像。这种鲁棒联合图像对齐的方法，不仅仅能实现图像的自动对齐，而且在对齐图像的同时能去除噪声等因素对图像的影响，如下图。



图 3-1 低秩矩阵分解对图像进行自动对齐的示意图
Figure 3-1 Low rank matrix decomposition and auto align

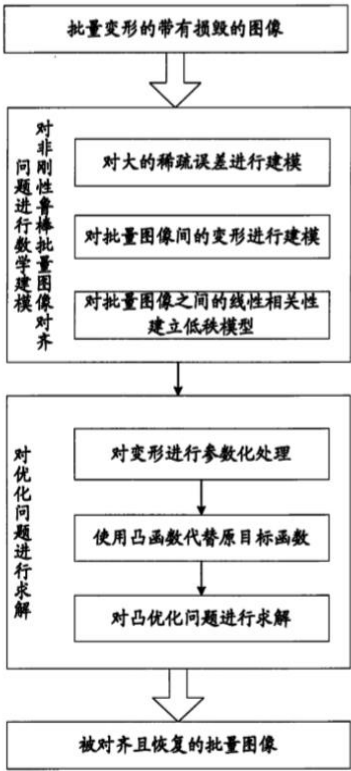
文章算法基本原理与流程

对图像进行对齐的基本思路是先寻找一变换因子。对于光场子图像，考虑子图像之间的差异较小，暂时考虑使用相似变换，即仅有旋转和平移变换。仿射变换包含了对图像进行的旋转、平移和缩放三种变换，每张自图像提取三个特征点即可使用仿射变换对子图像进行对齐。

寻找到变换因子后对图像实行此变换，经过变换后的图像构成的矩阵具有低秩结构，将变换后的矩阵通过分解，得到低秩矩阵部分和稀疏误差部分，这样不仅能对原图像实现姿态的纠偏，而且能去除图像的噪声污染。显然，稀疏误差部分不仅包含噪声，还包含原图像上的高频信息，对于光场子图像或者视频中相邻帧图像，误差矩阵包含了移动物体的轮廓信息。

用简单的流程图表示如图 3 . 1 所示：

用简单的流程图表示子图像批量对齐，如下图所示：



实现效果

光场图像压缩和编解码主要包括五个步骤：光场子图像提取、子图像批量对齐、低秩矩阵分解、编码传输、解码和恢复光场图像。

1. 光场子图像提取

使用 matlab 的工具包 LFToolbox0.4 对光场图像进行处理,得到一组光场图的子孔径图像，
设一共有 N 张。(为了简化运算，取 N=18)

每张子孔径图上取两个特征点,求解初始化的 N 个相似矩阵(仿射矩阵则需要使用三个点)。

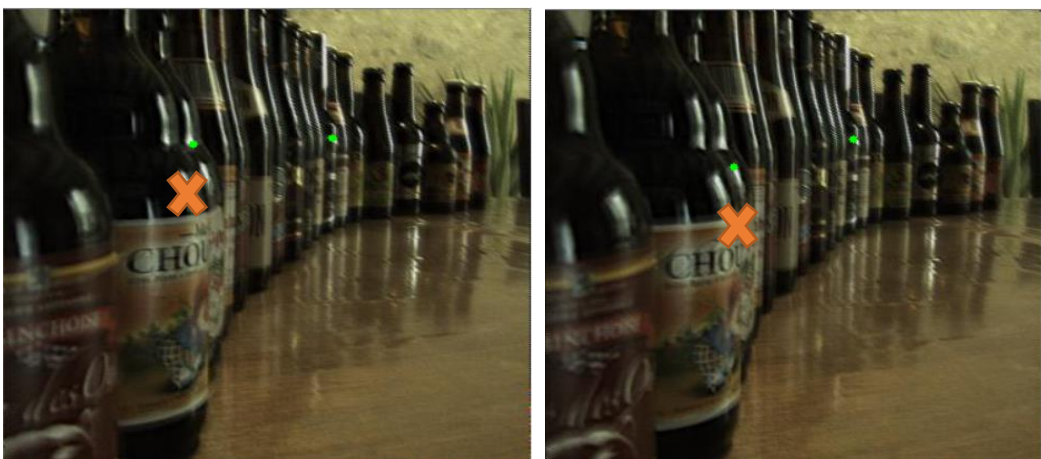
求解相似矩阵是为了对后续 ADMM (交叉方向乘子) 进行初始化,由于仅在初始化的操作中使用,因此对特征点的精度要求不是过分严格,这里使用**手动标定**的方法。

具体步骤

1. 将 N 张光场子孔径图像导入 opencv + python 3.6 中进行显示,使用 matplotlib 包读取图像上的特征点,手动确定两个特征在每张图中坐标位置。
2. 将得到数据导入 matlab,批量生成.mat 矩阵,存放每张图上的两个特征点的坐标位置。
3. 将 N 个 .mat 特征点坐标位置数据和对应的 N 张子孔径图像打包。放在

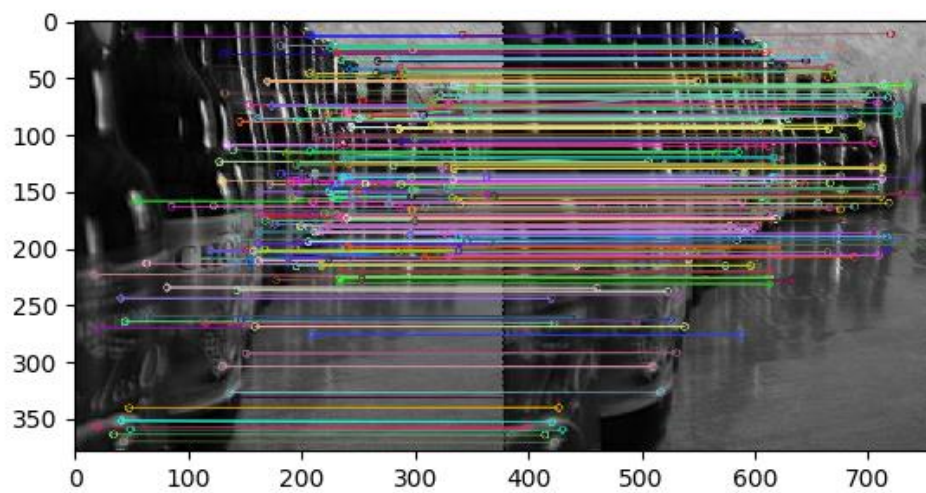
cvv\rasl_LF\data\data_test\Image

思考：此处特征点的匹配和标定也可以使用 opencv 包中的函数实现角点检测,提取两个特征点。但是,经过人工检查,效果并不好,两幅图中的特征点对应错误,如下图。



使用角点检测后,有一对特征点没有正确匹配

此处可以换用 sift 方法自动提取多个匹配的特征点。Sift 方法使用 `cvv\sift` 路径中的 `feature_points_main.py` 即可观察到下图的效果。



使用 sift 方法正确匹配，可以简化数据集的创建步骤，暂未使用该方法

2. RASL 用于子图像批量对齐

$$\operatorname{argmin}_{h, B, C} \|I \circ h - BC\|_F^2,$$

(1)

$$\min_{A, E, \Delta\tau} \|A\|_* + \lambda \|E\|_1 \quad \text{s.t.} \quad D \circ \tau + \sum_{i=1}^n J_i \Delta\tau \epsilon_i \epsilon_i^T = A + E.$$

(2)

Algorithm 1 (Outer loop of RASL)

INPUT: Images $I_1, \dots, I_n \in \mathbb{R}^{w \times h}$, initial transformations τ_1, \dots, τ_n in certain parametric group \mathbb{G} , weight $\lambda > 0$.

WHILE not converged **DO**

step 1: compute Jacobian matrices w.r.t. transformation:

$$J_i \leftarrow \frac{\partial}{\partial \zeta} \left(\frac{\text{vec}(I_i \circ \zeta)}{\|\text{vec}(I_i \circ \zeta)\|_2} \right) \Big|_{\zeta=\tau_i}, \quad i = 1, \dots, n;$$

step 2: warp and normalize the images:

$$D \circ \tau \leftarrow \left[\frac{\text{vec}(I_1 \circ \tau_1)}{\|\text{vec}(I_1 \circ \tau_1)\|_2} \dots \frac{\text{vec}(I_n \circ \tau_n)}{\|\text{vec}(I_n \circ \tau_n)\|_2} \right];$$

step 3: solve the linearized convex optimization:

$$(A^*, E^*, \Delta\tau^*) \leftarrow \arg \min_{A, E, \Delta\tau} \|A\|_* + \lambda \|E\|_1$$

$$\text{s.t.} \quad D \circ \tau + \sum_{i=1}^n J_i \Delta\tau_i \epsilon_i^T = A + E;$$

step 4: update transformations: $\tau \leftarrow \tau + \Delta\tau^*$;

END WHILE

OUTPUT: solution A^*, E^*, τ to problem (5).

Algorithm 2 (Linearized Inner Loop of RASL)

INPUT: $\mu^0 > 0$, $(A^0, E^0, \Delta\tau^0) \in \mathbb{R}^{m \times n} \times \mathbb{R}^{m \times n} \times \mathbb{R}^{p \times n}$.
Set $t^1 = t^0 = 1$, $(A^1, E^1, \Delta\tau^1) = (A^0, E^0, \Delta\tau^0)$, $k = 1$.

WHILE not converged **DO**

step 1: compute proximal points:

$$Y_A^k = A^k + \frac{t^{k-1}-1}{t^k} (A^k - A^{k-1}),$$

$$Y_E^k = E^k + \frac{t^{k-1}-1}{t^k} (E^k - E^{k-1}),$$

$$Y_{\Delta\tau}^k = \Delta\tau^k + \frac{t^{k-1}-1}{t^k} (\Delta\tau^k - \Delta\tau^{k-1});$$

step 2: gradient step:

$$G_A^k = Y_A^k + Y_E^k - D \circ \tau - \sum_{i=1}^n J_i Y_{\Delta\tau}^k \epsilon_i \epsilon_i^T,$$

$$G_A^k = Y_A^k - \rho^{-1} G_A^k, \quad G_E^k = Y_E^k - \rho^{-1} G_E^k,$$

$$G_{\Delta\tau}^k = Y_{\Delta\tau}^k + \rho^{-1} \sum_{j=1}^n J_j^T G_E^k \epsilon_j \epsilon_j^T;$$

step 3: soft-thresholding:

 compute the reduced SVD (U, S, V) of G_A^k ,

$$A^{k+1} = U T_{\mu^k/\rho}(S) V^T, \quad E^{k+1} = T_{\lambda\mu^k/\rho}(G_E^k),$$

$$\Delta\tau^{k+1} = G_{\Delta\tau}^k;$$

step 4: update:

$$t^{k+1} = \frac{1}{2} + \frac{1}{2} \sqrt{1 + 4(t^k)^2}, \quad \mu^{k+1} = \max\{0.9\mu^k, \bar{\mu}\};$$

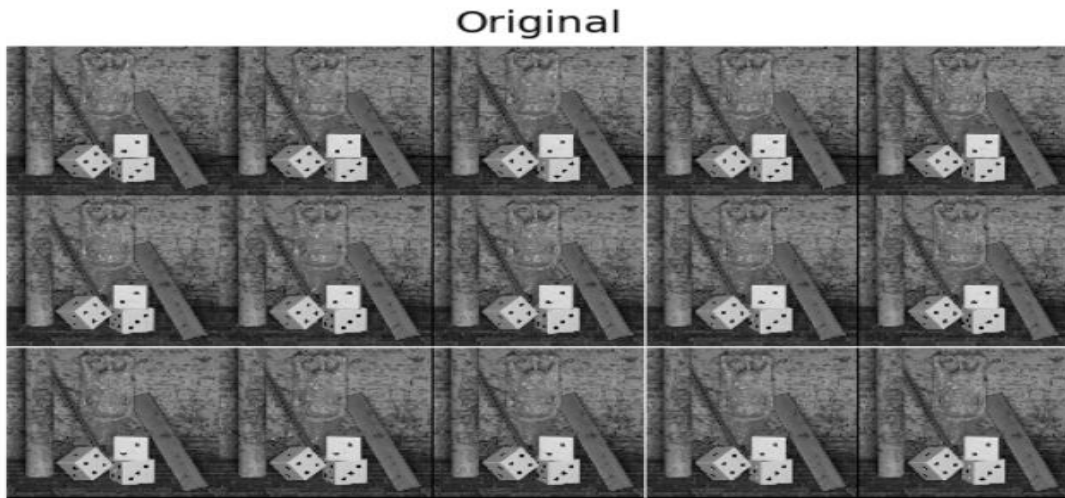
END WHILE

OUTPUT: solution $(A, E, \Delta\tau)$ to problem (7), and hence (6).

两层循环

使用 python3 运行 **cvv\rasl_LF\main 中的 budahan.py** 即可观察到下

图。经过 2 次迭代，得到对齐及矩阵分解的结果



1. *Chlorophyll *a** and *Chlorophyll *b** were determined by the method of Arar and Collins (1971).
 2. *Carotenoids* were determined by the method of Arar and Collins (1971).
 3. *Protein* was determined by the method of Lowry (1956).

使用 matlab 经过 9 外层迭代和 418 次内层循环，得到 15 张子图像的秩收敛到 $r=8$

1) 低秩矩阵分解

- Singular vectors are **dense**!

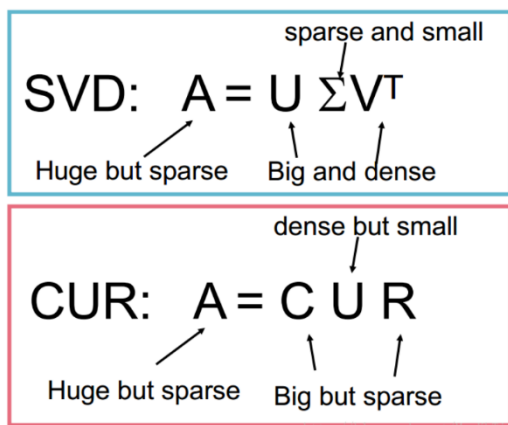
The diagram shows a 3x3 grid of dots labeled U on the left, followed by an equals sign, and then a 3x3 grid of stars labeled V on the right. Below the stars is a small 3x3 grid of squares labeled Σ , representing the transformation matrix.

Given h fixed, B and C are found by computing the singular value decomposition $I \circ h = U\Sigma V^\top$. Then B is set as the k first columns of $U\Sigma$ and C is set as the k first rows of V^\top , so that BC is the **closest k -rank** approximation of $I \circ h$.

思考：即使可以把一个很大的矩阵分解成两个长条，但是 SVD 不够好，因为得到的分解结

果矩阵不具有稀疏的特性。为进一步得到更为稀疏的分解形式，这里将对齐后的图片矩阵使用 CUR 分解。CUR 分解，适合大数据；有编码、计算优势，不方便呈现。

运行 **cvv\CUR 路径中的 CUR.py** ,可观察到分解效果 其中 **cvv\CUR 文件 1.xlsx** 是对齐后的图片拉成列向量后的矩阵。



```
The reconstruction error is
121866292.08603786
The time taken is
1.2254218893286428
```

抽取 500 行，10 列，平均每个像素点误差

$121866292 / (700 \times 700 \times 18) = 13.8$ 。

3. 编解码

尝试传统熵编码？

4. 恢复光场图像

效果分析

判断压缩的性能

码率和信噪比

总结

- 1) 对一张光场图的所有子图像降采样、改进特征点标定，测试算法步骤 123。
- 2) 完成编解码和图像恢复。
- 3) 安装加速包。

参考文献

- [1] Light Field Compression With Homography-Based Low-Rank Approximation
- [2] Homography-Based Low Rank Approximation of Light Fields for Compression
- [3] RASL: Robust Alignment by Sparse and Low-rank Decomposition for Linearly Correlated Images
- [4] Incremental low-rank and sparse decomposition for compressing videos captured by fixed cameras

