



西安交通大学  
XI'AN JIAOTONG UNIVERSITY

# 数字图像处理实验报告

项目名称：图像配准

姓名：刘朔江

班级：自动化钱 91

学号：2196113328

## 摘 要

本文基于 Matlab 的计算机视觉相关组件，对两幅图像进行配准。通过选取图片中 7 对点的方法，计算出两幅图像间的转换矩阵，从而得出结果。

由此，我认识了图像配准的基本策略、特点及应用。通过查询相关资料，我还了解到，在实际应用中，有更为先进的、不局限于找点的配准方法。

# 一 题目 1

## 1.1 题目

要求根据已给的两幅图像，在各幅图像中随机找出 7 个点，计算出两幅图像之间的转换矩阵  $H$ ，并且输出转换之后的图像。

以下为固定图像：

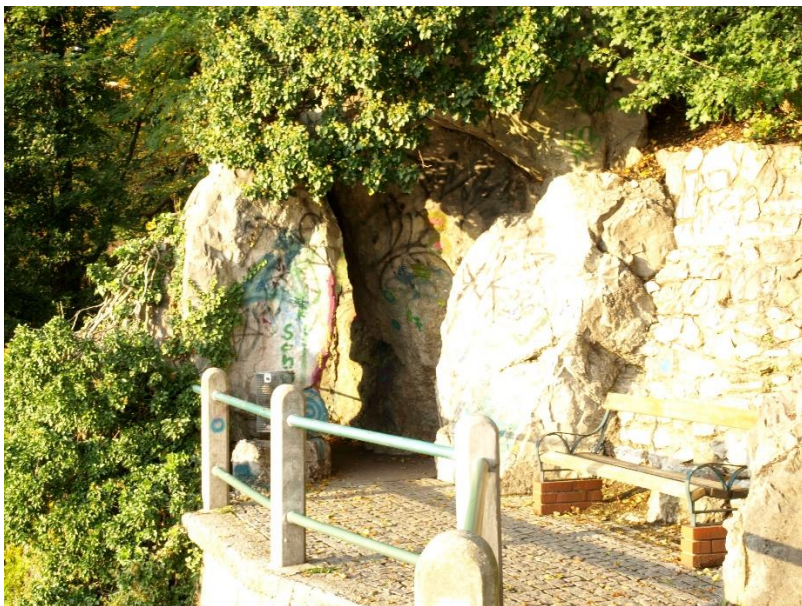


图1 固定图像

以下为移动图像：

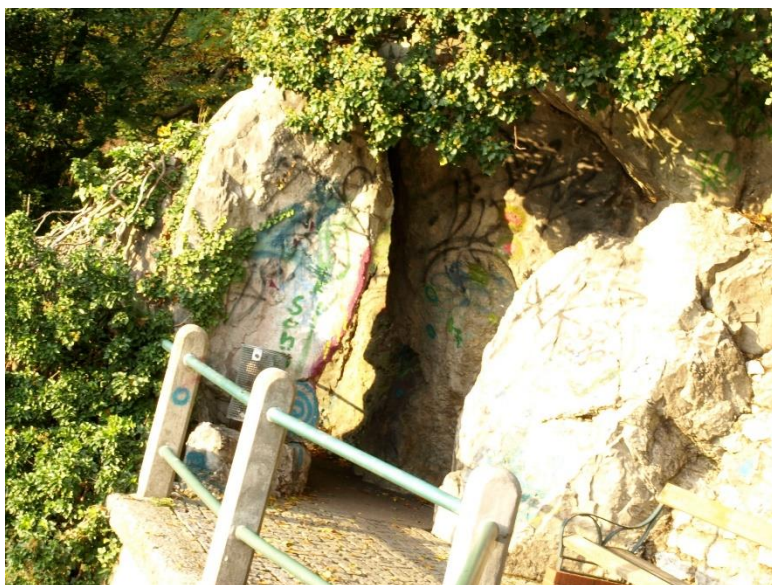


图2 移动图像



## 1.2 解答

图像配准是将不同的图像数据集转换为一个坐标系下的过程。数据可能是多张照片、来自不同传感器、时间、深度或视点的数据。它用于计算机视觉、医学成像、军事自动目标识别以及编译和分析来自卫星的图像和数据。为了能够比较或整合从这些不同测量中获得的数据，图像配准是必要的。

图像配准算法可以分为基于强度和基于特征的算法。其中一幅图像称为移动图像或源图像，而其他图像称为目标图像、固定图像或感测图像。图像配准涉及对源/运动图像进行空间变换，以与目标图像对齐。目标图像中的参考帧是静止的，而其他数据集被转换以匹配目标。基于强度的方法通过相关性度量比较图像中的强度模式，而基于特征的方法则找到图像特征（例如点、线和轮廓）之间的对应关系。基于强度的方法配准整个图像或子图像。如果配准了子图像，则将相应子图像的中心视为相应的特征点。基于特征的方法在图像中许多具有代表性的点之间建立对应关系。知道图像中多个点之间的对应关系，然后确定几何变换以将目标图像映射到参考图像，从而在参考图像和目标图像之间建立逐点对应关系。

在这里我们使用 Matlab 的图像配准相关函数及工具箱进行操作。

Matlab 的 Registration Estimator 提供了多种方法进行图像配准，将移动图像和目标图像输入到工具箱中，可根据具体需要选择配准策略，这里选择 Brisk Technique。

调整参数至 7 对点被匹配，如下图所示：

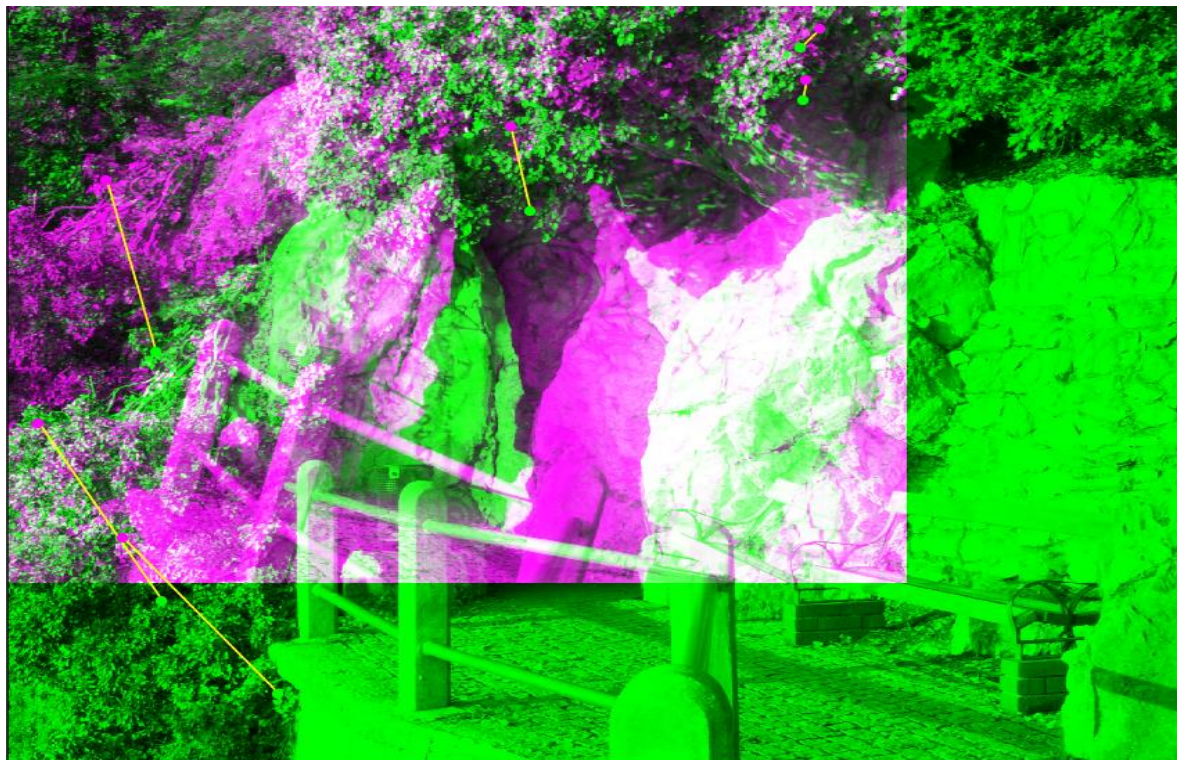


图3 选择七对点

接下来进行配准，配准后的转换图像输出如图所示：

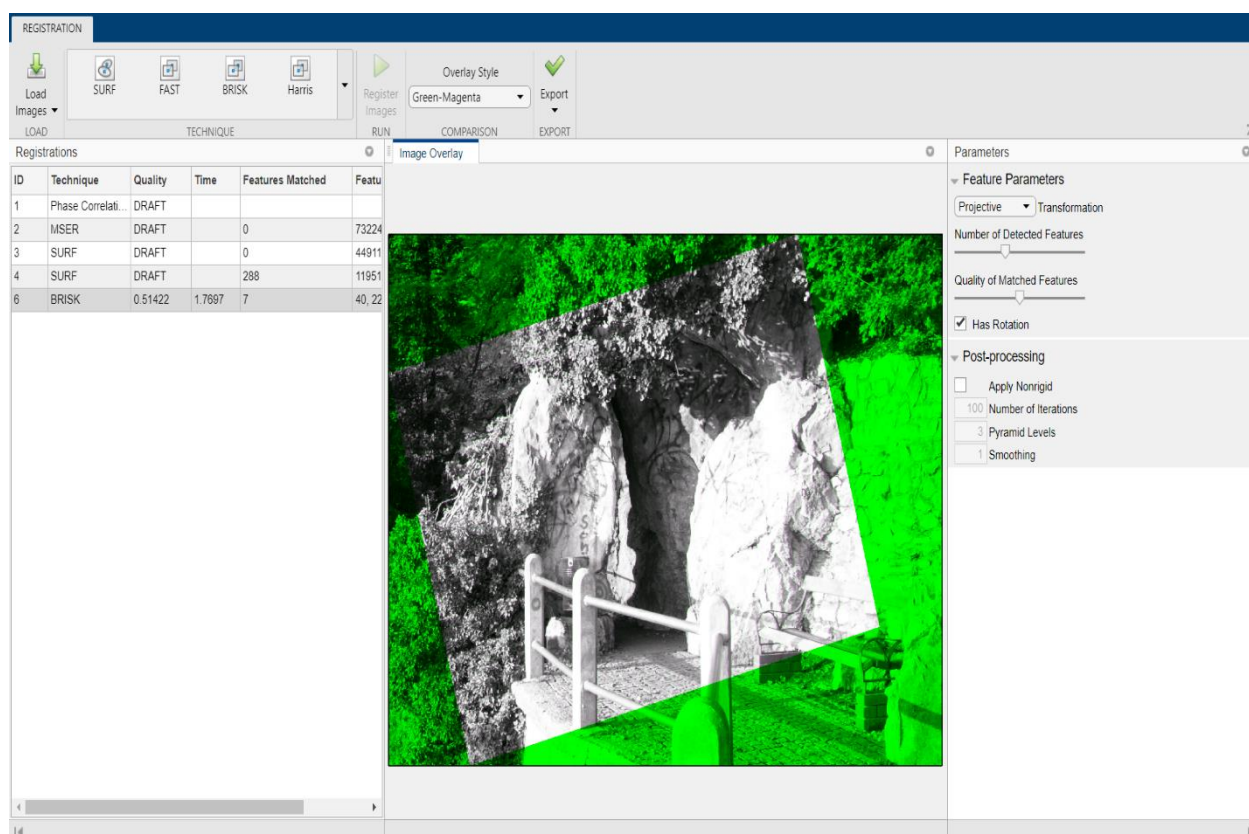


图4 配准结果

可见，配准的图像结果良好，其贴合度较高。

通过查询结果的结构体 `movingReg` 得出转移矩阵为：

$$\begin{bmatrix} 0.9652 & -0.2573 & -5.5423 \times 10^{-7} \\ 0.2557 & 0.9639 & -6.9392 \times 10^{-7} \\ 0.4339 & 718.6944 & 1 \end{bmatrix}$$

## 二 附录： Matlab 代码

```
function [MOVINGREG] = registerImages(MOVING, FIXED)

% Convert RGB images to grayscale
FIXED = im2gray(FIXED);
MOVINGRGB = MOVING;
MOVING = im2gray(MOVING);

% Default spatial referencing objects
fixedRefObj = imref2d(size(FIXED));
movingRefObj = imref2d(size(MOVING));

% Detect BRISK features
fixedPoints =
detectBRISKFeatures(FIXED, 'MinContrast', 0.688000, 'MinQuality', 0.649000, 'NumOctaves', 2);
movingPoints =
detectBRISKFeatures(MOVING, 'MinContrast', 0.688000, 'MinQuality', 0.649000, 'NumOctaves', 2);

% Extract features
[fixedFeatures, fixedValidPoints] =
extractFeatures(FIXED, fixedPoints, 'Upright', false);
[movingFeatures, movingValidPoints] =
extractFeatures(MOVING, movingPoints, 'Upright', false);

% Match features
indexPairs =
matchFeatures(fixedFeatures, movingFeatures, 'MatchThreshold', 50.000000, 'MaxRatio', 0.500000);
fixedMatchedPoints = fixedValidPoints(indexPairs(:, 1));
movingMatchedPoints = movingValidPoints(indexPairs(:, 2));
MOVINGREG.FixedMatchedFeatures = fixedMatchedPoints;
MOVINGREG.MovingMatchedFeatures = movingMatchedPoints;

% Apply transformation - Results may not be identical between runs
because of the randomized nature of the algorithm
tform =
estimateGeometricTransform2D(movingMatchedPoints, fixedMatchedPoints, 'projective');
MOVINGREG.Transformation = tform;
```

```
MOVINGREG.RegisteredImage = imwarp(MOVINGRGB, movingRefObj, tform,  
'OutputView', fixedRefObj, 'SmoothEdges', true);  
  
% Store spatial referencing object  
MOVINGREG.SpatialRefObj = fixedRefObj;  
  
end
```