

西安交通大学

计算机视觉与
模式识别

计算机 53 班

龙思宇

2150500103

一、 通过仿射变换将两幅图拼接起来，填充 64-65 两行程序，找到自己拍摄的两幅照片，将照片拼接起来

补充部分

```
A = [xsRight(1) ysRight(1) 1 0 0 0;  
      0 0 0 xsRight(1) ysRight(1) 1;  
      xsRight(2) ysRight(2) 1 0 0 0;  
      0 0 0 xsRight(2) ysRight(2) 1;  
      xsRight(3) ysRight(3) 1 0 0 0;  
      0 0 0 xsRight(3) ysRight(3) 1;];  
b =  
[xsLeft(1);ysLeft(1);xsLeft(2);ysLeft(2);xsLeft(3);ysLeft(3);]  
;  
  
x = A\b;
```

实验结果



二、 利用 Harris 角点进行仿射变换参数估计，进而进行图像拼接的

算法

detHarrisCorners 程序缺失的部分

%Filter for horizontal and vertical direction

$dx = [1 \ 0 \ -1];$

$dy = [1;0;-1];$

% Convolution of image with dx and dy

$I_x = \text{conv2}(\text{img}, dx, 'same');$

$I_y = \text{conv2}(\text{img}, dy, 'same');$

% Hessian Matrix I_{xx} , I_{yy} , I_{xy}

$I_{xx} = I_x .* I_x;$

$I_{yy} = I_y .* I_y;$

$I_{xy} = I_x .* I_y;$

$I_{xx} = \text{conv2}(I_{xx}, G, 'same');$

$I_{yy} = \text{conv2}(I_{yy}, G, 'same');$

$I_{xy} = \text{conv2}(I_{xy}, G, 'same');$

% Calculate the corner responseling'luan

$k = 0.04;$ % usually in the range[0.04 0.06]

$\text{delta} = (I_{xx} + I_{yy}).^2 - 4 * 1 * (I_{xx}.*I_{yy} - I_{xy} .* I_{xy});$

$\text{lambda1} = 0.5 * ((I_{xx} + I_{yy}) + \text{sqrt}(\text{delta}));$

```

lambda2 = 0.5 * ((Ixx + Iyy) - sqrt(delta));

R = lambda1 .* lambda2 - k * (lambda1 + lambda2).^2;

% corner = zeros(nr,nc);

corner = (R>corn_thresh*rmax).*R;

corner_peaks=imregionalmax(R);

for h=1:nr
    for w=1:nc
        if(corner_peaks(h,w)==1)
            if(corner(h,w)<=0)
                corner_peaks(h,w)=0;
            end
        end
    end
end

[iLoc,jLoc] = find(corner_peaks==1);

```

extractNccFeature 程序中缺失的部分

```

temp = img(ylo:yhi,xlo:xhi);

descps(i,1:numel(temp)) = reshape(temp,[1,numel(temp)]);

```

est_optimal_affine 函数

```
function F = est_optimal_affine(rightpoints,leftpoints)
```

```
    A = [rightpoints(1,1) rightpoints(1,2) 1 0 0 0;
```

```
          0 0 0 rightpoints(1,1) rightpoints(1,2) 1;
```

```
          rightpoints(2,1) rightpoints(2,2) 1 0 0 0;
```

```
          0 0 0 rightpoints(2,1) rightpoints(2,2) 1;
```

```
          rightpoints(3,1) rightpoints(3,2) 1 0 0 0;
```

```
          0 0 0 rightpoints(3,1) rightpoints(3,2) 1];
```

```
    b=[leftpoints(1,1);leftpoints(1,2);leftpoints(2,1);leftpoints(2,2);left  
points(3,1);leftpoints(3,2)];
```

```
    F = A \ b;
```

```
    F = [F(1:3)';F(4:6)'];
```

```
End
```

另外还在框架里找到了一个 **bug**

Demo_harris_affine.m 里

```
% 记录两个边框
```

```
bounds = cell(2,4);
```

```
    bounds{1,1} = [1 nx1 nx1 1;1 1 ny1 ny1] + repmat([-xlo+1;-  
ylo+1],[1 4]);
```

```
    bounds{2,1} = x2bound_transformed + repmat([-xlo+1;-ylo+1],[1  
4]);
```

```
bounds{1,2} = [1 0 -xlo+1; 0 1 -ylo+1];
```

```
bounds{2,2} = Aff; bounds{2,2}(:,3) = bounds{2,2}(:,3) + [-  
xlo+1;-ylo+1];
```

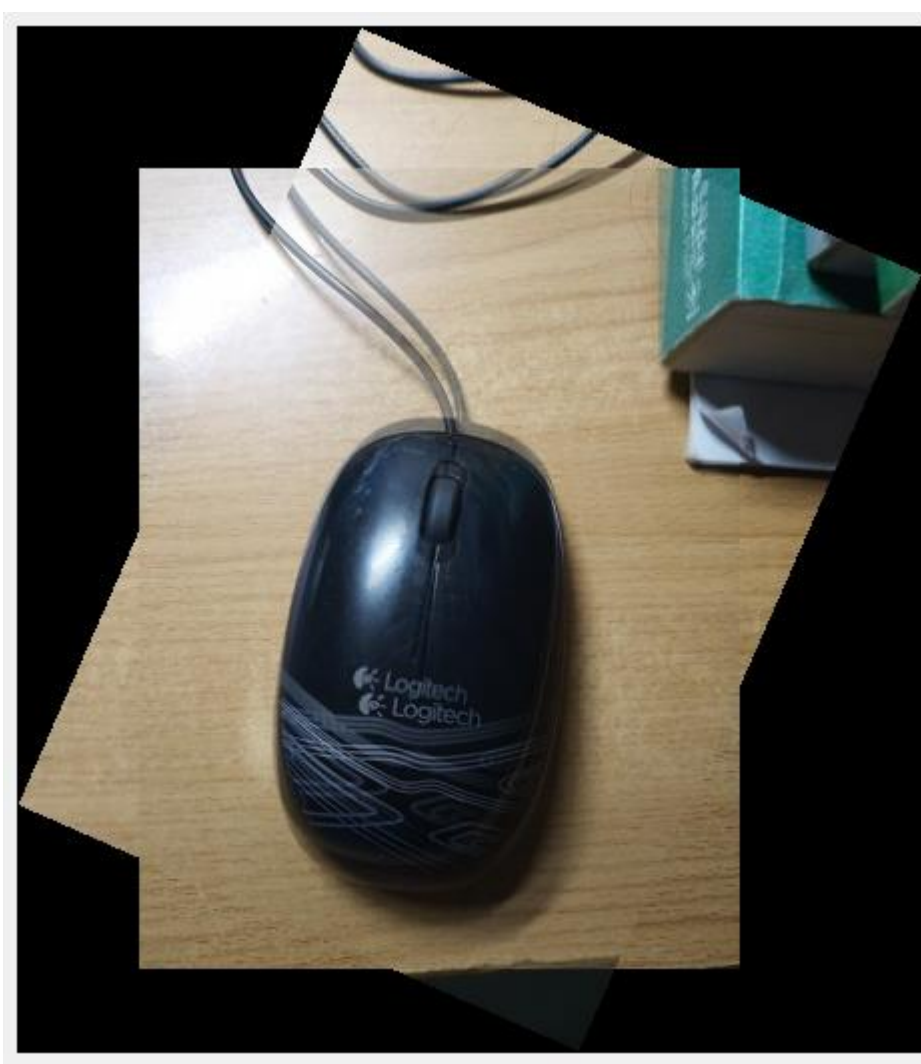
在下载下来的代码里[1 nx1 nx1 1;1 1 ny1 ny1]和 repmat([-xlo+1;-ylo+1],[1 4]);是用减号连接的，应该用加号。

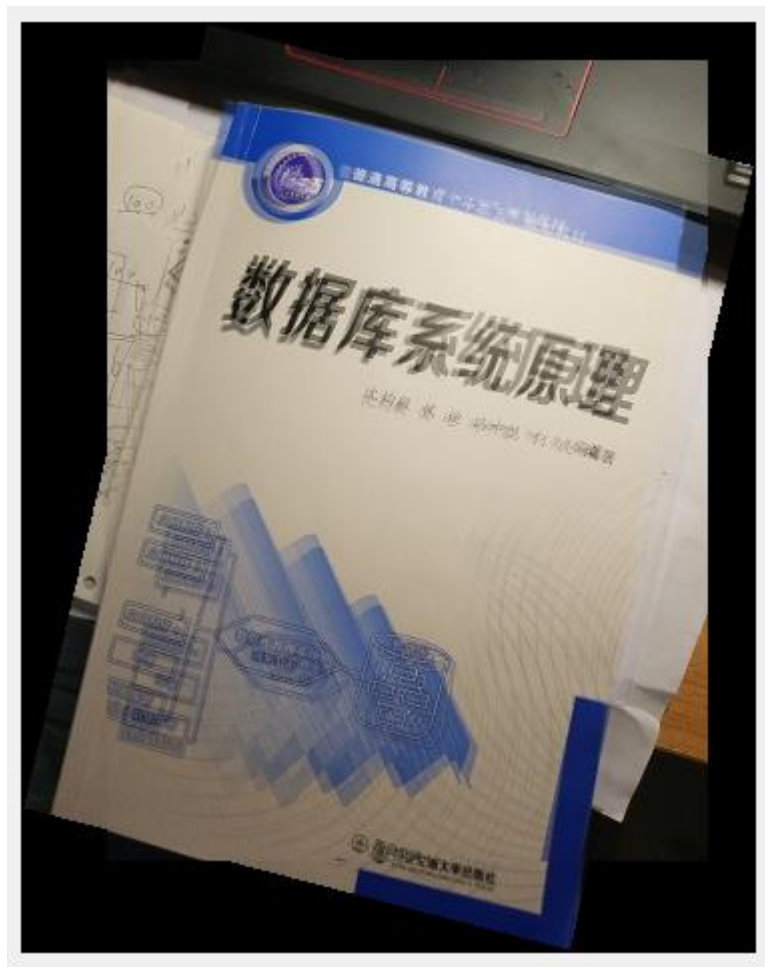
三、 阅读 est_optimal_Affine_ransac 程序，绘制流程图

见附录

四、 找到四组自己拍摄的照片，进行拼接，评价并分析拼接的结果
拼接结果







总体来说，拼接结果还是比较令人满意的，算法选取的点，以及计算出来的仿射矩阵还是比较合适的，但是还是可以看到拼接的边缘和细节处还是有比较大的差异的，这是由于现实中不是仿射变换，而是透视投影变换，我们使用仿射变换近似有误差，再有就是选取的点并不是没有误差，构造的仿射矩阵并不是完美的。

五、 附录

