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

1. 填补backward\_geometry.m程序中缺失的部分，让文件能够运行

源代码

function outputIm = backward\_geometry(inputIm, A,type)

% inputIm = 输入的图像

% A = 仿射变换的系数

%获取输入图像的大小

inputSize = size(inputIm);

if(size(inputIm, 3) == 1)

inputSize(3) = 1;

end

%imshow(inputIm);

% 计算输出图像的画布大小¡

[outputSize, deltaShift] = calcOutputSize(inputSize, A,type);

%A\_inv = [(1/(A(1,1)\*A(2,2)/A(1,2) - A(2,1))) 0;0 (1/(A(2,2)\*A(1,1)/A(2,1) - A(1,2)))]\* [(A(2,2)/A(1,2)) -1;-1 (A(1,1)/A(2,1))] \* [1 0 -A(1,3);0 1 -A(2,3)];

A\_inv = A(1:2,1:2);

B = A(:,3);

outputIm = zeros(outputSize(1),outputSize(2),3);

% 根据输出画布大大小来遍历

for i = 1 : outputSize(1)

for j = 1 : outputSize(2)

y = j + deltaShift(2);

x = i + deltaShift(1);

% 逆向变换

z = A\_inv \ ([x;y] - B);

%z = round(z);

z\_floor = floor(z);

delta = z - z\_floor;

w00 = (1 - delta(1)) \* (1 - delta(2));

w01 = delta(1) \* (1 - delta(2));

w10 = (1 - delta(1)) \* delta(2);

w11 = delta(1) \* delta(2);

% 双线性插值

if z\_floor(1) >= 1 && z\_floor(1) + 1 <= inputSize(2) && z\_floor(2) >= 1 && z\_floor(2) + 1 <= inputSize(1)

outputIm(i,j,1) = w00 \* inputIm(z\_floor(2),z\_floor(1),1) + w01 \* inputIm(z\_floor(2) + 1,z\_floor(1),1) + w10 \* inputIm(z\_floor(2),z\_floor(1) + 1,1) + w11 \* inputIm(z\_floor(2) + 1,z\_floor(1) + 1,1);

outputIm(i,j,2) = w00 \* inputIm(z\_floor(2),z\_floor(1),2) + w01 \* inputIm(z\_floor(2) + 1,z\_floor(1),2) + w10 \* inputIm(z\_floor(2),z\_floor(1) + 1,2) + w11 \* inputIm(z\_floor(2) + 1,z\_floor(1) + 1,2);

outputIm(i,j,3) = w00 \* inputIm(z\_floor(2),z\_floor(1),3) + w01 \* inputIm(z\_floor(2) + 1,z\_floor(1),3) + w10 \* inputIm(z\_floor(2),z\_floor(1) + 1,3) + w11 \* inputIm(z\_floor(2) + 1,z\_floor(1) + 1,3);

%outputIm(i,j,2) = inputIm(z(2),z(1),2);

%outputIm(i,j,3) = inputIm(z(2),z(1),3);

end

end

end

outputIm = uint8(outputIm);

end

function [outputSize, deltaShift] = calcOutputSize(inputSize, A,type)

% 'crop'

% Make output image B the same size as the input image A, cropping the rotated image to fit

% {'loose'}

% Make output image B large enough to contain the entire rotated image. B is larger than A

ny = inputSize(1);

nx = inputSize(2);

inputBoundingBox = [ 1 1 1;...

nx 1 1;...

nx ny 1;...

1 ny 1];

inputBoundingBox = inputBoundingBox';

outputBoundingBox = A \* inputBoundingBox;

xlo = floor(min(outputBoundingBox(1,:)));

xhi = ceil(max(outputBoundingBox(1,:)));

ylo = floor(min(outputBoundingBox(2,:)));

yhi = ceil(max(outputBoundingBox(2,:)));

if strcmpi(type,'loose')

outputSize(1) = xhi - xlo;

outputSize(2) = yhi - ylo;

else

outputSize(1) = nx;

outputSize(2) = ny;

end

if strcmpi(type,'loose')

deltaShift(1) = xlo - 1;

deltaShift(2) = ylo - 1;

else

deltaShift(1) = 0;

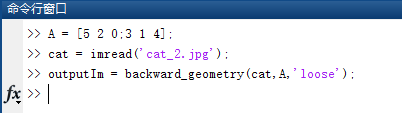
deltaShift(2) = 0;

end

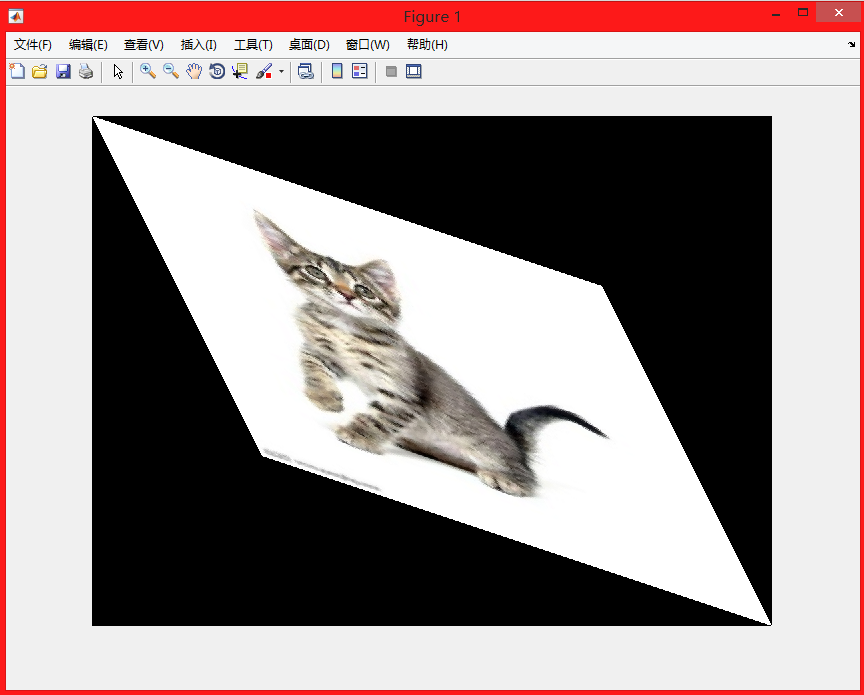
end

实验结果

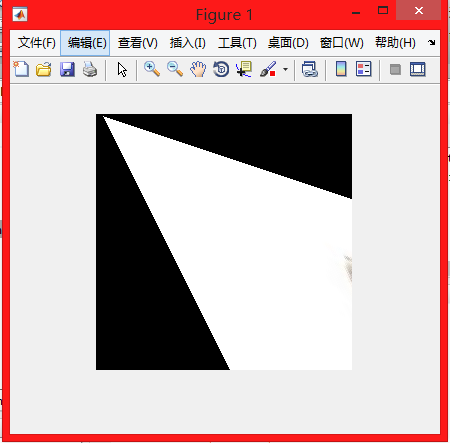
输入如下命令（使用loose模式）



输出结果如下

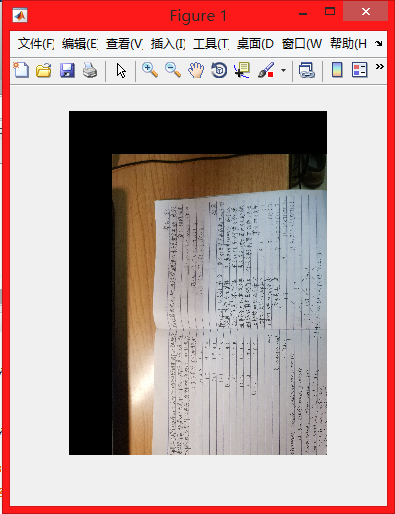


使用crop模式，结果如下

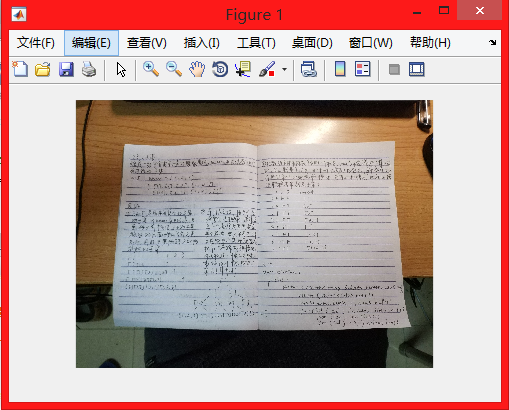


1. 自己选择3幅自己拍摄的照片，自己设计放射变换矩阵实现变换

平移变换，使用的系数矩阵A = [1 0 8;0 1 8]，为了方便展示效果使用crop模式，结果如下：



翻转变换，使用的系数矩阵A = [0 1 0;1 0 0]，结果如下：



错切变换，使用的系数矩阵A = [0.5 0.5 0;1.5 1 0]，结果如下：

