山东大学 计算机科学与技术 学院

计算机视觉 课程实验报告

学号：201600181073	姓名： 唐超	班级： 16 人工智能

实验题目：图像仿射变换及图像变形

实验内容：

1. 图像仿射变换
   - 设计一个函数 WarpAffine，可以对图像进行任意的二维仿射变换（用 2*3 矩阵表示）：
     - 采用双线性插值进行重采样；
     - 可以只考虑输入图像为 3 通道，8 位深度的情况；
     - 函数接口可以参考 OpenCV 的 warpAffine 函数
   - 调用 WarpAffine，实现绕任意中心的旋转函数 Rotate

2. 图像变形
   - 记[x’，y’]=f([x，y])为像素坐标的一个映射，实现 f 所表示的图像形变。f 的逆映射为：

$$[x,y]=f^{-1}([x',y'])=\begin{cases}[x',y'] & \text{if } r\geq 1\\ [\cos(\theta)x'-\sin(\theta)y',\sin(\theta)x'+\cos(\theta)y'] & \text{otherwise}\end{cases}$$

其中：$r=\sqrt{x'^2+y'^2}\quad \theta=(1-r)^2$

$[x,y],[x',y']$都是中心归一化坐标。

实验过程中遇到和解决的问题：

（记录实验过程中遇到的问题，以及解决过程和实验结果。可以适当配以关键代码辅助说明，但不要大段贴代码。）

- 图像的仿射变换

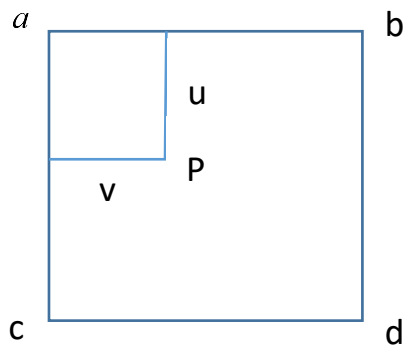
1. 求仿射变换矩阵的逆

为求得变换后图像的每个像素点在原图像中的位置，需要求仿射变换矩阵的逆，在仿射变换矩阵第三行添加{0，0，1}，经计算，

$$\begin{pmatrix}a_1 & b_1 & c_1\\ a_2 & b_2 & c_2\\ 0 & 0 & 1\end{pmatrix}$$ 的逆为 $$\frac{1}{a_1b_2-a_2b_1}\begin{pmatrix}b_2 & -b_1 & b_1c_2-c_1b_2\\ -a_2 & a_1 & a_2c_1-a_1c_2\\ 0 & 0 & a_1b_2-a_2b_1\end{pmatrix}$$

2. 双线性插值

由于变换后图像的像素点经逆变换对应的原图像像素点不全是整点，需要进行双线性插值求得该点的像素值。另外，如果对应的原图像像素点超出了原图像的范围，则不进行处理。



如图，P 点为变换后的图像中某一点像素对应于原图像的像素点的位置，其四周的像素点已经转换为了标准的 $(0,0),(0,1),(1,0),(1,1)$，分别对应某一通道的值为 a,b,c,d，P 点坐标为 (u,v)，则双线性插值的结果为

$$\begin{pmatrix}1-u & u\end{pmatrix}\begin{pmatrix}a & b\\ c & d\end{pmatrix}\begin{pmatrix}1-v\\ v\end{pmatrix}$$

3. 边界处理

由于双线性插值需要周围四个点的像素值，因此在图像边界处用双线性插值会出错。在图像的最右边一列和最下方一行采用最近邻插值进行处理

```
x = int(x1+0.5);
y = int(y1+0.5);
```

即把 $(x1,y1)$ 的像素值用 (x,y) 处的代替。

4. 复合仿射变换的矩阵

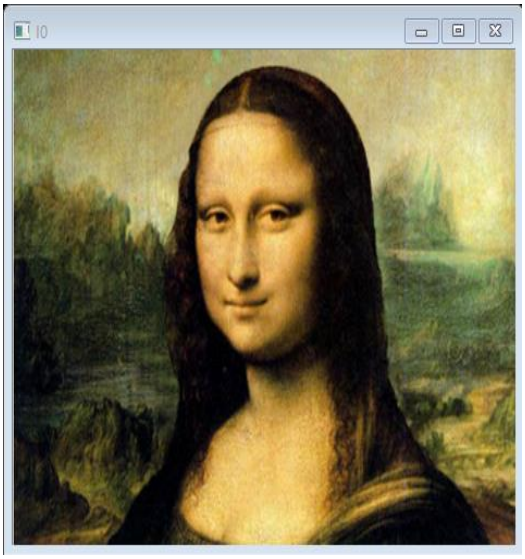
复合仿射变换的矩阵等于单独的变换矩阵按顺序左乘前一个变换的矩阵（每个变换的矩阵为三阶方阵），设置 3×3 矩阵乘法函数 mat33multiple，将复合仿射变换转化为只含一个矩阵的变换。由于在 WarpAffine 函数的参数中矩阵的大小为 2×3，在最终得到的矩阵中只取前两行（第三行一定为{0，0，1}）。

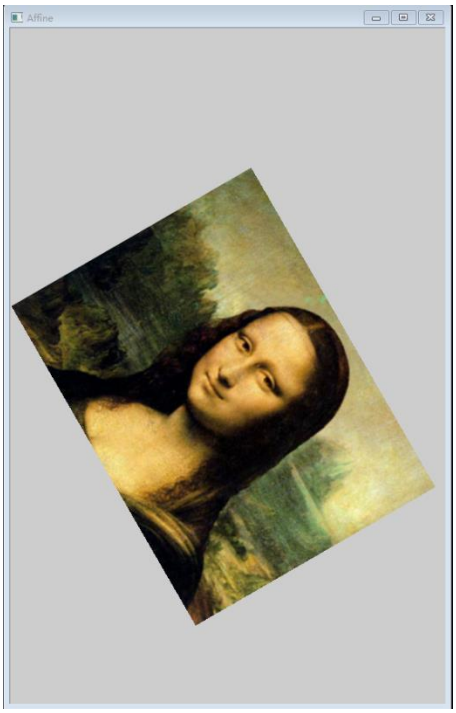
5. rotate 函数

用 rotate 函数调用 WarpAffine，实现绕任意中心的旋转，`void rotate(Mat src, Mat dst0,int x0, int y0, float angle)`，其中 src，dst0 分别为原图像和旋转后的图像，x0 和 y0 为旋转中心，angle 为旋转角度，经旋转中心平移至原点，旋转，旋转中心平移回原位置三个仿射变换实现绕 $(x0,y0)$ 旋转的目的，其中绕原点旋转 angle 的变换矩阵为

$$\begin{pmatrix}\cos(angle) & -\sin(angle) & 0\\ \sin(angle) & \cos(angle) & 0\end{pmatrix}$$

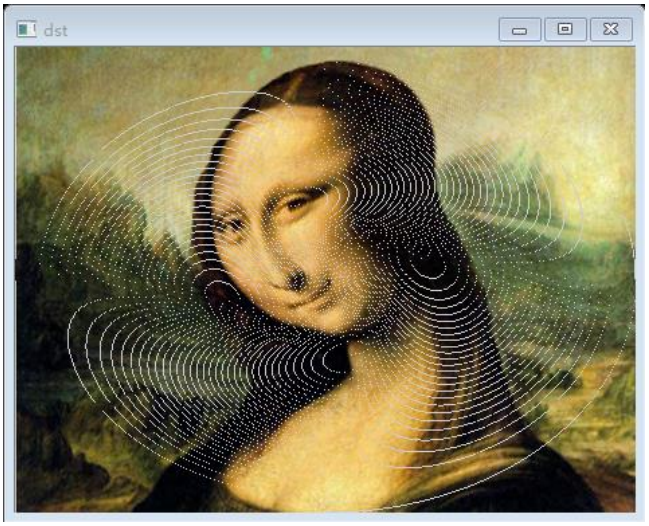
原图像绕其左下角顺时针旋转 $60°$ 的图像





- 图像变形

直接用逆映射公式会造成变形后图像部分点缺失，如下

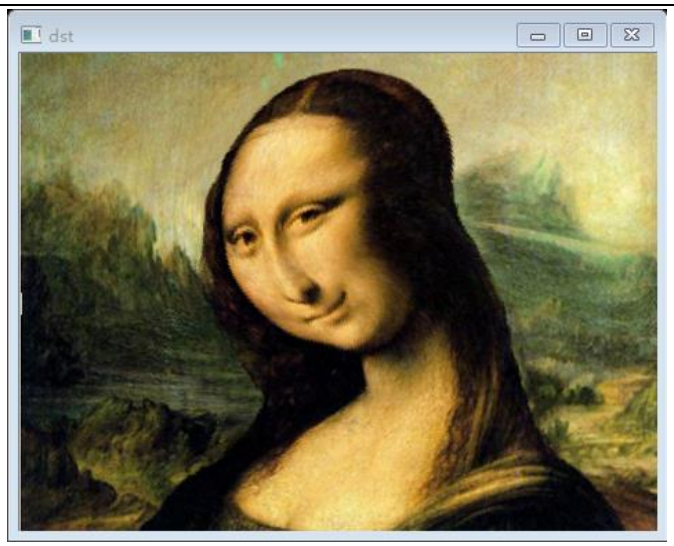


经坐标中心归一化和逆映射规则处理后，变形后图像(x_2, y_2)处像素点在原图像中的“行坐标”和“列坐标”分别为

$$x_1 = (x_c \cos\theta + y_c \sin\theta)x_0 + x_0, y_1 = (y_c \cos\theta - x_c \sin\theta)y_0 + y_0$$

其中 $x_c = \frac{x - x_0}{x_0}, y_c = \frac{y - y_0}{y_0}, x_0 = 0.5 \times dst.rows, y_0 = 0.5 \times dst.cols$

这样处理后生成的变形图像如下



结论分析与体会：

在图像的仿射变换中，由于原图像在窗口的左上角，很容易造成图像在窗口边界处截断，如果将图像事先平移至一个更大窗口的中心，会减少这种情况。但当旋转中心比较远或者旋转角度比较大时，仍然不可避免地会造成图像在窗口边界处截断，而这也是在特定旋转中心和特定旋转角度下旋转的体现，因此没有再调整。

通过这次实际实验，加深了我对图像仿射变换及图像变形的理解，也使我对 VS&OpenCV 使用得更加熟练了。