一种基于错切原理的图像旋转方法

陈 芳1,2

(1.华东师范大学 信息科学与技术学院,上海 200062; 2.淮阴师范学院 计算机系,江苏 淮安 223001)

摘 要:通过对通用的图像旋转矩阵作变换处理,给出了基于错切原理实现图像旋转的推导过程.基于错切原理的图像旋转避免了通用旋转方法带来的缩放效应.

关键词:几何变换:图像旋转:错切

中图分类号: TP391.41 文献标识码: A 文章编号:1671-6876(2004)04-0319-04

0 引言

数字图像处理是计算机科学的一个重要研究领域,也是计算机图形处理软件的一项核心技术.在数字图像处理过程中,经常要用到旋转,例如:图像扫描输入时常发生位置倾斜,对其进行倾斜校正需要对图像进行旋转;在多幅图像比较和模式识别中,要对图像进行旋转操作;在图像的剪裁和拼接前,也要对图像作旋转处理.图像的旋转是数字图像处理的一个非常重要的环节,是图像的几何变换手法之一.

本文从通用的图像旋转矩阵着手,给出了基于错切原理的图像旋转变换的推导过程并通过了实验验证.

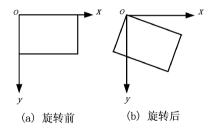


图 1 图像旋转前后比较

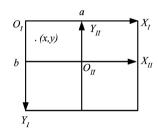


图 2 坐标系平移示意图

1 图像的几何旋转变换

图像旋转的通用公式^[1] 为:
$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$
 (1)

令旋转矩阵

$$\begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} = R.$$

式(1) 中: x_0 , y_0 为旋转前像素的坐标植; x_1 , y_1 为了旋转后映射图像对应像素的坐标植;旋转角 θ 绕基准点逆时针旋转为正值,顺时针旋转为负值.

式(1)是基于同一平面直角坐标系内绕原点旋转的像素坐标变换算法.由于旋转后图像长度和宽度往往会发生变化,而且超出平面直角角坐标系的第一限象(图1).为了保证图像旋转后,整体显示在计算机屏幕上(以屏幕左上角为坐标原点),一般以图像的中心为基准点旋转,这中间涉及到坐标系的平

移.假设图像未旋转时中心坐标为(a,b),旋转后中心坐标为(c,d)(在新的坐标系下,以旋转后新图像左上角为原点),则旋转变换公式为:

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & c \\ 0 & -1 & d \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -a \\ 0 & -1 & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$
(2)

式(2)中,a,b,c,d的值一般由图像旋转前后的顶点坐标确定.

式(2)是由原图像像素坐标值求解旋转后图像像素坐标值,即正向映射法^[2].由于数字图像中像素坐标值只能为正整数,旋转任意角度后由正向映射法求出的坐标值往往出现小数,这样就会有未被赋值的"空"像素.为避免这种情况,图像旋转中一般采用逆向映射法^[2],即由变换后的映射图像像素的坐标值,逆推出在原图像中对应的坐标值,如出现小数,根据精度要求采用不同的插值算法求得灰度值.由于正向映射与逆向映射互为反函数,因此式(2)的逆向映射的变换公式为:

$$\begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -a \\ 0 & -1 & b \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & c \\ 0 & -1 & d \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$
(3)

即

$$\begin{cases} x_0 = x_1 \cos(\theta) + y_1 \sin(\theta) - c\cos(\theta) - d\sin(\theta) + a \\ y_0 = -x_1 \sin(\theta) + y_1 \cos(\theta) + c\sin(\theta) - d\cos(\theta) + b \end{cases}$$

$$(4)$$

2 基于错切原理的旋转变换

2.1 公式推导

错切(shear) 是一种使对象形状发生变化的变换^[3].经过错切的对象好像是由已经相互滑动的内部夹层组成.常用的错切变换有两种:改变 x 坐标值和改变 y 坐标值.基于这一思想,对旋转矩阵 R 作如下变换,已知

$$\begin{bmatrix} 1 & 0 \\ \tan(\theta/2) & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 \\ \tan(\theta/2) & 1 \end{bmatrix} = I$$

I为单位阵.对式(1)中的旋转矩阵 R作如下的初等变换:

第一步:旋转阵 R 左乘 I 得式(5)

$$R = \begin{bmatrix} 1 & 0 \\ \tan(\theta/2) & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 \\ \tan(\theta/2) & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$$
 (5)

即

$$R = \begin{bmatrix} 1 & 0 \\ \tan(\theta/2) & 1 \end{bmatrix}^{-1} \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\tan(\theta/2) & 1 \end{bmatrix}$$
 (6)

第二步:由已知可得

$$I^{-1} = \begin{bmatrix} 1 & 0 \\ \tan(\theta/2) & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \tan(\theta/2) & 1 \end{bmatrix}^{-1}$$

式(6) 右 I^{-1} 乘得式(7)

$$R = \begin{bmatrix} 1 & 0 \\ \tan(\theta/2) & 1 \end{bmatrix}^{-1} \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\tan(\theta/2) & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \tan(\theta/2) & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \tan(\theta/2) & 1 \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} 1 & 0 \\ \tan(\theta/2) & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & \sin(\theta) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \tan(\theta/2) & 1 \end{bmatrix}^{-1}$$

$$(7)$$

经过上述变换,矩阵 R被分解成三个矩阵相乘,即式(7).由于数字图像处理在实现过程中多采用逆向映射法,因此对(7)式求逆得(8)式

$$R^{-1} = \begin{bmatrix} 1 & 0 \\ \tan(\theta/2) & 1 \end{bmatrix} \begin{bmatrix} 1 & -\sin(\theta) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \tan(\theta/2) & 1 \end{bmatrix}$$
(8)

2.2 算法实现

本算法是在 windows 环境下,采用 Visual C++6.0 为软件工具的条件下实现的.算法涉及的主要变量设定如下:

- i:目标图像像素所在行
- i:目标图像像素所在列
- i0:源图像像素所在行
- i0:源图像像素所在列

INewHeight:目标图像的高

lnewWidth:目标图像的宽

fRoateAngle:旋转角度

fTan - halfa:旋转角一半的正切函数值

fSina:旋转角的正弦值

offset:错切位移量

lpNewDIBBits:指向目标图像的指针

lpDIBBits:指向源图像的指针

lNewLineBytes:目标图像的每行字节数

lLineBytes:源图像每行的字节数

lpDst:目标图像的像素指针

lpSrc:源图像像素指针

沿 x 轴方向水平错切 $x = x + y \tan(\theta/2)$,其流程图如图 3 所示.

由错切原理可知,源图与目标图之间的映射关系属均匀映射,因此,对水平错切,只需对图像每一行的初始一列的像素作切移和插值处理,这样可以有效提高程序的时间效率.

沿 y 轴方向错切: $y = y - x\sin(\theta)$,其流程图与图 3 类似,本文不再赘述.

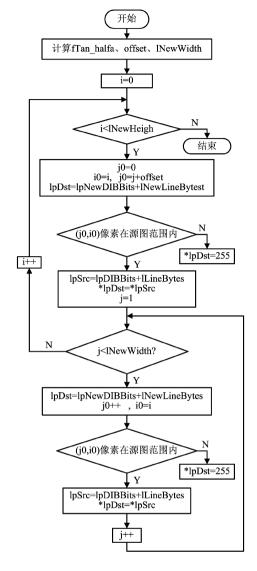


图 3 流程示意图

利用错切原理,在windows 环境下,以Visual C++6.0为工具对原图(图4(a))旋转30度角所得的实验结果见图4(d),图4(b)和图4(c)均为过渡图像.

测试

测试

测试

和证

(a) 原图 (256色灰度图)

(b) 30度水平切移

(c) 30度垂直切移

(d) 30度水平切移(目标图像)

图 4 原图(256色灰度图)与目标图(切移旋转 30度)

3 结束语

基于错切的旋转变换,由于将通用旋转矩阵分解成三个矩阵的乘积,其对线角线元素全为 1,有效地避免了缩放因子 $^{[4]}$,使所得的旋转图像不会"变形".对于 $(\theta/2)=(n\pi)/2$,n 为奇数情况,可以将 θ 分解,分步旋转得到.因此,基于错切原理的旋转算法仍适用于旋转任意角度.

参考文献:

- [1] 何斌, 马天予. Visual C++数字图像处理 [M]. 北京: 人民邮电出版社, 188-191.
- [2] 徐锦林,梁军.图像旋转的改进算法[J].西北建筑工程学院学报,1998,12(4):64-67.
- [3] Hearn D, Baler M P. 蔡士杰译. 计算机图形学 [M]. 北京: 电子工业出版社, 159-160.
- [4] Wolberg G. Digital Image Warping [M]. Los Alamitos California: IEEE Computer Society Press, 1990. 208 209.

The Method of Image Rotation Based on Shear Principle

CHEN Fang^{1, 2}

- (1. College of Information Science and Technology, East China Normal University, Shanghai, 200062, China) (2. Department of Computer Science, Huaiyin Teachers College, Jiangsu Huaian, 223001, China)
- **Abstract:** Through warping to the matrix of image rotation in common, provide the course of deriving of realizing image rotation based on shear principle. The image rotation on the basis of shear principle has avoided the scale effect that the rotatory method in common brings.

Key words: geometry warping; image rotation; shear

「责任编辑:蒋海龙]