

基于像素分割的亚像素圆模板算法

陈良^{1,2}, 高成敏^{1,2}

CHEN Liang^{1,2}, GAO Cheng-min^{1,2}

1. 广东警官学院 计算机系, 广州 510232

2. 华南理工大学 计算机科学与工程学院, 广州 510640

1. Guangdong Police Officers College, Guangzhou 510232, China

2. South China University of Technology, Guangzhou 510640, China

E-mail: Chenl@scut.edu.cn

CHEN Liang, GAO Cheng-min. Sub-pixel circle template algorithm based on pixel segmentation. Computer Engineering and Applications, 2007, 43(16): 50-52.

Abstract: In order to improve the image quality and performance of rotation, a sub-pixel circle template algorithm is proposed. Suppose the image is continue, rotate it inversely; Choose the center of sub-pixel circle template by dividing the center pixel into mesh; Cover the image with discrete circle template which diameter equals the edge of pixel. Rotated pixel color equals the convolution R_s and C_s . R_s are ratios of the pixels areas covered by circle and the circle area, and C_s are the corresponding pixels colors. Experiments show that the image quality rotated by the proposed algorithm is better than that of nearest neighborhood interpolation and Gaussian template algorithm, its continuity near that of bilinear interpolation, its performance is higher than that of bilinear interpolation and Gaussian template algorithm.

Key words: circle template; pixel segmentation; sub-pixel; Gaussian template; bilinear interpolation; image rotation

摘要: 为提高旋转图像的质量和性能, 提出了一种亚像素圆模板算法。在假设图像是连续的前提下进行逆向旋转变换; 通过像素网格状分割选取亚像素圆模板的圆心; 用亚像素圆模板(直径等于像素边长)进行离散化。邻域像素被圆模板覆盖的面积与圆面积的比值 R_s , 相应像素的颜色值 C_s , 旋转后像素的颜色值等于 R_s 和 C_s 的卷积。实验表明该算法图像旋转效果好于最近邻点插值和高斯模板算法, 连续性接近双线性插值, 计算性能明显高于双线性插值和高斯模板。

关键词: 圆模板; 像素分割; 亚像素; 高斯模板; 双线性插值; 图像旋转

文章编号: 1002-8331(2007)16-0050-03 文献标识码: A 中图分类号: TP391.41

1 引言

数字图像处量是延续人类视觉的重要手段。我们在设计或使用数字图像处理算法或设备时, 应该考虑人的图像感知原理。如果一幅图像要由人来分析的话, 信息应该用人容易感知的变量来表达, 这些是心理物理参数包括对比度、边界、形状、纹理、色彩等等^[1]。图像旋转变换造成的失真给人的最直接视觉感受是图像变得模糊尤其边界处发生侵蚀。图像旋转的几何变换一般采用逆向旋转以防止出现空洞, 像素灰度值插值算法如最近邻点插值、双线性插值、双三次插值^[2]、高斯模板插值^[3]等对旋转后的图像质量起到决定性作用。最近邻插值的定位误差最大是半个像素。这种误差在物体具有直线边界时会显现出来, 在变换后可能会呈现阶梯状。双线性插值可能会引起小的分辨率降低和模糊, 原因在于其平均化的本性。它减轻了在最近邻插值中出现的阶梯状直线边界的问题。双三次插值免除了最近邻插值的阶梯状的问题, 也解决了线性插值的模糊问题。双三

次插值很好地保持了图像的细节^[4], 但双线性插值和双三次插值的计算速度不能满足实时性需要。

模板通常用于图像滤波、平滑、锐化、图像匹配、边界检测、放缩等^[4-6]。高斯模板插值也被用于图像旋转。用模板进行图像处理的一个优势在于其具有很高的性能, 因此本文设计了基于像素分割的亚像素圆模板算法并将其用于图像旋转。该算法计算性能较双线性插值明显提高, 旋转图像的质量接近双线性插值算法。

2 图像旋转与失真原因分析

图像旋转(如图1)一般采用逆向旋转获取像素值的方法, 可以避免目标图像产生空洞, 即先由源图像区域 Ω 及其旋转角度 θ 确定目标图像的图像区域 Ω' 。对目标图像的每一个像素点 $P'(x', y')$ 逆向几何旋转得到点 $P(x, y)$, 如果点 $P(x, y) \notin \Omega$, 像素点 $P'(x', y')$ 处的像素值 $C(x', y')$ 等于背景值; 否则, 即

基金项目: 广东省教育厅资助科研课题(the research Project of Department of Education of Guangdong Province, China under Grant No.2003C32401)。

作者简介: 陈良(1969-), 男, 博士生, 主要研究方向: 计算机图形学、计算机网络及计算机网络安全。高成敏(1974-), 女, 硕士生, 主要研究方向: 计算机图形学、信息系统开发。

点 $P(x,y) \in \Omega$, 像素点 $P'(x',y')$ 处的像素值可以选用各种灰度插值算法得到。因此灰度插值算法决定了旋转后图像的质量, 几可旋转算法和灰度插值算法的复杂度共同决定了算法的性能。

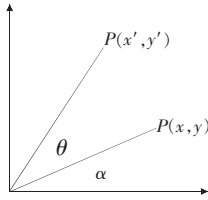


图1 图像几何旋转示意图

$$x' = r \times \cos(\theta + \alpha) = r \times (\cos\theta \cos\alpha - \sin\theta \sin\alpha) = x \cos\theta - y \sin\theta \quad (1)$$

$$y' = r \times \sin(\theta + \alpha) = r \times (\sin\theta \cos\alpha + \cos\theta \sin\alpha) = x \sin\theta + y \cos\theta \quad (2)$$

解式(1)、(2)构成的方程组得:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}$$

像素点 $P'(x',y')$ 和点 $P(x,y)$ 的坐标不可能同时取整, 也就是旋转前后的像素点的亮度不是一一对应的关系, 因此需要进行亮度插值。最近邻点插值算法图像连续性差, 但速度快; 双线性插值图像连续性好, 但速度慢; 高斯模板插值图像的模糊度高, 速度介于前两者之间。本文设计的算法连续性好、模糊度低, 速度接近最近邻点插值算法。

图像旋转变换造成图像失真原因是显示图像的像素点的离散性和像素点排列的各向异性。因此在虚拟的连续空间进行图像旋转, 并进行近似连续的离散化处理, 再映射到离散的显示空间, 这样可以降低离散化造成图像失真的效应。采用模板可以提高计算性能。

3 像素分割与亚像素圆模板

3.1 像素分割

图2是像素分割及圆模板灰度插值原理图。 $P_i (i=1, \dots, 9)$ 是 3×3 个像素, 其中像素 P_5 (正方形 $ABCD$, 边长为1), 被等分为 3×3 的子像素 $i (i=1, \dots, 9)$, 它们的中心分别是 $O_i (i=1, \dots, 9)$ 。设 A 点坐标为 (I_1, J_1) (I_1, J_1 为整数, 以图像的左下角为坐标原点), 像素 $P_5 (ABCD)$ 内任一点 P 的全局坐标为: $(I_1 + u, J_1 + v)$, $u, v \in [0, 1)$ 。以 A 点为坐标原点的子像素 $i (i=1, \dots, 9)$ 的中

心点 $O_i (i=1, \dots, 9)$ 的局部坐标分别为: $O_1 (\frac{1}{6}, \frac{1}{6}), O_2 (\frac{3}{6}, \frac{1}{6}), O_3 (\frac{5}{6}, \frac{1}{6}); O_4 (\frac{1}{6}, \frac{3}{6}), O_5 (\frac{3}{6}, \frac{3}{6}), O_6 (\frac{5}{6}, \frac{3}{6}); O_7 (\frac{1}{6}, \frac{5}{6}), O_8 (\frac{3}{6}, \frac{5}{6}), O_9 (\frac{5}{6}, \frac{5}{6})$ 。

3.2 亚像素圆模板

局部相关性假设: 像素点 $P'(x',y')$ 反向旋转后的点 $P(x,y)$ 。像素点 $P'(x',y')$ 的颜色取值是以点 $P(x,y)$ 为中心的一个小的邻域内 (本文取亚像素圆) 的颜色值的函数 F 。

近似代换: 当点 $P(x,y)$ 在某个子像素 $i (i=1, \dots, 9)$ 内, 则用子像素 i 的中心 O_i 的邻域来代替。

图2中的3个圆模板分别是以 O_2, O_5, O_9 为圆心以 $1/2$ 为半径的实线圆、长划线圆、点线圆。其面积 $S = \pi/4 = 0.7854 < 1$ (像素面积为1), 所以称为亚像素模板。点 E, F 是圆 O_2 和 AB 边的交点, 点 I, J, K, H 分别是圆 O_9 和 BC 边、像素 P_9 的两条边、 CD 边的交点。

函数 F_j 的计算方法: 设圆模板所覆盖的像素 P_i 的面积为 S_i , 相应的像素 P_i 的颜色值为 C_i , 则

$$F_j = \sum_{i=1}^9 \frac{S_i}{S} C_i \quad (3)$$

其中 i 是像素编号 $1, \dots, 9$; j 是子像素编号 $1, \dots, 9$ 。

对于圆模板 $O_2, EF = 2 \times \sqrt{(1/2)^2 - (1/6)^2} = 0.9428$, EF 所对的圆心角:

$$\begin{aligned} \angle EO_2F &= 2 \times \arccos\left(\frac{1/6}{R}\right) = 2 \times \arccos\left(\frac{1/6}{1/2}\right) = \\ &= 2 \times \arccos(1/3) = 2.4619 \end{aligned}$$

设扇形 O_2EF 面积为

$$S_{EO_2F} = S \times \frac{2.462}{2 \times \pi} = \frac{\pi}{4} \times \frac{2.462}{2 \times \pi} = 0.3077$$

三角形 EO_2F 的面积为

$$S_{\Delta EO_2F} = \frac{1}{2} EF \times \frac{1}{6} = \frac{0.9428}{12} = 0.0786$$

$$S_2 = S_{EO_2F} - S_{\Delta EO_2F} = 0.3077 - 0.0786 = 0.2291$$

$$S_5 = S - S_2 = 0.7854 - 0.2291 = 0.5563$$

对于 $i \in \{2, 5\} S_i = 0$ 。将 S_i, C_i 代入(3)可得:

$$\begin{aligned} F_2 &= \frac{1}{0.7854} (0.2291 C_2 + 0.5563 C_5) = \\ &= \frac{1}{7854} (2291 C_2 + 5563 C_5) \end{aligned} \quad (4)$$

同理, 对于圆模板 O_4, O_6, O_8 可得:

$$F_4 = \frac{1}{7854} (2291 C_4 + 5563 C_5) \quad (5)$$

$$F_6 = \frac{1}{7854} (2291 C_6 + 5563 C_5) \quad (6)$$

$$F_8 = \frac{1}{7854} (2291 C_8 + 5563 C_5) \quad (7)$$

对于圆模板 $O_5, S_5 = S, i \neq 5, S_i = 0$:

$$F_5 = \sum_{i=1}^9 \frac{S_i}{S} C_i = \frac{1}{S} (S_5 C_5) = C_5 \quad (8)$$

对于圆模板 $O_9, S_9 = 0.3877, S_8 = S_6 = 0.1686, S_7 = 0.0605, i \notin \{5, 6, 8, 9\}, S_i = 0$, 因此

$$F_9 = \frac{1}{7854} (605 C_9 + 1686 C_6 + 1686 C_8 + 3877 C_5) \quad (9)$$

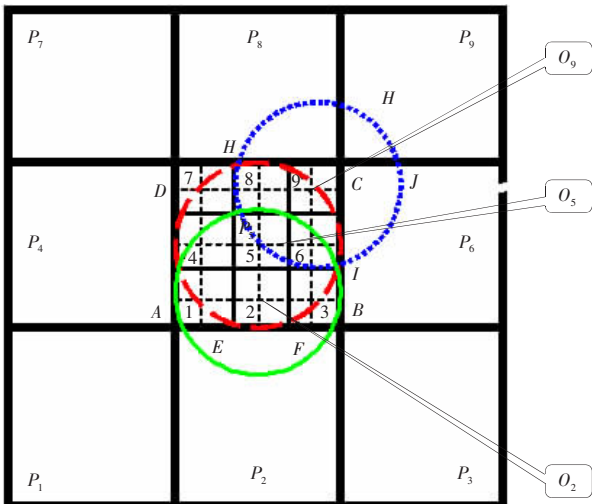


图2 分割中心像素 P_5 及圆模板示意图

同理,对于圆模板 O_1, O_3, O_7 可得:

$$F_1 = \frac{1}{7854} (605C_1 + 1686C_2 + 1686C_4 + 3877C_5) \quad (10)$$

$$F_3 = \frac{1}{7854} (605C_3 + 1686C_2 + 1686C_6 + 3877C_5) \quad (11)$$

$$F_7 = \frac{1}{7854} (605C_7 + 1686C_4 + 1686C_6 + 3877C_5) \quad (12)$$

从式(4)到式(12)得到对应每个子像素(近似取整)的模板:

$$T_7 = \frac{1}{4} \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix}; \quad T_8 = \frac{1}{4} \begin{bmatrix} 1 \\ 3 \end{bmatrix}; \quad T_9 = \frac{1}{4} \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}$$

$$T_4 = \frac{1}{4} [1 \ 3]; \quad T_5 = [1]; \quad T_6 = \frac{1}{4} [3 \ 1]$$

$$T_1 = \frac{1}{4} \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}; \quad T_2 = \frac{1}{4} \begin{bmatrix} 3 \\ 1 \end{bmatrix}; \quad T_3 = \frac{1}{4} \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}$$

4 实验结果与分析

4.1 实验环境与结果

对原始图像图3(a)用不同的算法进行旋转,旋转角度为 30° ,连续旋转12次。旋转的结果如图3(b)双线性插值旋转、图3(c)本文算法旋转、图3(d)最近邻插值算法、图3(e)高斯模板算法。从旋转后的图像质量来看,本文算法接近双线性插值,明显优于最近邻插值算法和高斯模板算法。



图3 用不同算法进行图像旋转($30^\circ \times 12$ 次)

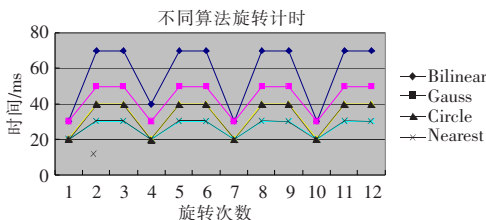


图4 对图3(a)用不同算法旋转计时($30^\circ \times 12$ 次)

表1 对图3(a)用不同算法旋转计时

累计转角/ $^\circ$	转角: $30^\circ \times 12$ 次												平均
	30	60	90	120	150	180	210	240	270	300	330	360	
Nearest/ms	20	30	30	20	30	30	20	30	30	20	30	30	26.7
Circle/ms	20	40	40	20	40	40	20	40	40	20	40	40	33.3
Gauss/ms	30	50	50	30	50	50	30	50	50	30	50	50	43.3
Bilinear/ms	30	70	70	40	70	70	30	70	70	30	70	70	57.5

从计算性能的比较看,最近邻插值算法计算最快,本文算法的计算速度接近最近邻插值算法,计算时间是最近邻插值算法的1.25倍(33.3/26.7),而双线性插值算法计算时间是最近邻

插值算法的2.15倍,是本文算法的1.73倍。Gauss模板是最近邻插值算法的1.63倍,是本文算法的1.3倍。

4.2 实验结果分析

最近邻插值算法只是对坐标值进行简单的取整,没有更复杂的亮度插值计算,因此其计算最快。双线性插值算法:

$$f(x,y) = [f(1,0) - f(0,0)]x + [f(0,1) - f(0,0)]y + [f(1,1) + f(0,0) - f(1,0) - f(0,1)]xy + f(0,0)$$

从上式可以看出双线性插值有4次浮点乘、3次浮点加和5次整数加。

高斯模板,当 $\tan(\theta) < 1, T_1 = \frac{1}{14} \begin{bmatrix} 1 & 1 & 1 \\ 2 & 4 & 2 \\ 1 & 1 & 1 \end{bmatrix}$, 当 $\tan(\theta) = 1, T_2 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$, 当 $\tan(\theta) > 1, T_3 = \frac{1}{14} \begin{bmatrix} 1 & 2 & 1 \\ 1 & 4 & 1 \\ 1 & 2 & 1 \end{bmatrix}$ 。设 T_1, T_2, T_3

使用的概率为 $P=1/3$,则整数加法的次数为: $1/3(8+8+8)=8$ 次;整数乘为: $1/3(3+5+3)=11/3$ 次;整数除为: $1/3(1+1+1)=1$ 次。

本文的算法,设 $T_i (i=1, \dots, 9)$ 的使用概率为 $P=1/9$ 。则整数加法: $1/9(1+2+1+1+0+1+1+2+1)=10/9$ 次。整数乘为: $1/9(1+1+1+1+0+1+1+1+1)=8/9$ 次。整数除为: $1/9(1+1+1+1+0+1+1+1+1)=8/9$ 次。其中的乘2可用 $<<i$ 代替,除2可用 $>>i$ 代替,可以提高计算速度。

表2 各种算法计算量统计

	整数加	整数乘	整数除	浮点加	浮点乘
Nearest	0	0	0	2	0
本文	10/9	8/9	8/9	0	0
Gauss	8	11/3	1	0	0
Bilinear	5	0	0	3	4

浮点乘占用处理器的周期最多,其次是浮点加,因此双线性插值计算复杂度最高。

5 结论

本文提出的算法可以用于旋转、放缩、失真校正等空域变换。提出的局部相关性假设,确保了设计的亚像素圆模板进行卷积运算时的连续性,因此其图像质量接近双线性插值。又因为像素分割使得选取的邻域是4像素(T_1, T_3, T_7, T_9)、2像素(T_2, T_4, T_6, T_8)甚至是1像素(T_5),因此减少了高斯模板9像素邻域加权平均造成的图像模糊。从统计的计算量和实测数据看,本文算法的计算速度明显优于双线性插值和高斯模板算法,能够应用于实时性要求较高的场合。(收稿日期:2007年3月)

参考文献:

- [1] Milan Sonka, Vaclav Hlavac, Roger Boyle. 图像处理、分析与机器视觉[M]. 艾海舟, 武勃, 译. 北京: 人民邮电出版社, 2003-09.
- [2] 张显全, 唐振军, 于金辉. 基于高斯模板的图像旋转算法[J]. 计算机工程与应用, 2005, 41(34): 79-81.
- [3] Li Xin. New edge-directed interpolation[J]. IEEE Transactions on Image Processing, 2001, 10(10): 521-1527.
- [4] 黄翔东, 王兆华, 李文元. 2维加窗全相位图像滤波模板的设计[J]. 中国图象图形学报, 2006, 11(6): 811-817.
- [5] 刘冬, 赵荣椿. 保存边界的图象平滑方法[J]. 西北工业大学学报, 2000, 18(1): 102-105.
- [6] Chen Chinghan, Hsieh Shenghsien. A novel image interpolator based on probabilistic neural network with shapeness/smoothness adaptation[C]//LNCS 3497: ISSN 2005, 2005: 698-706.