

# 双线性插值算法在荔枝树光合模拟中叶片图像旋转中的应用

李 震<sup>1</sup>, 洪添胜<sup>2\*</sup>, 吴伟斌<sup>1</sup>, 张文昭<sup>1</sup>, 郭 灼<sup>1</sup>

(1. 华南农业大学工程学院, 广州 510642; 2. 华南农业大学信息学院, 广州 510642)

**摘 要:** 为寻找荔枝冠层光合有效辐射模拟中合适的叶片旋转算法, 实验针对树叶灰度图像, 将双线性插值算法引入图像旋转中, 并与采用最邻近插值算法的结果进行对比分析, 着重比较了树叶与背景之间边缘的过渡情况和二值化后树叶所占面积的变化情况。结果表明, 采用双线性插值算法目标与背景之间的过渡平滑, 边界清晰, 旋转后的图像清晰度较高; 采用最邻近插值算法目标与背景之间的过渡不平滑, 有比较严重的方块效应, 图像整体清晰度较差, 二值化后图像的平均变形达到 20 个像素。

**关键词:** 图像旋转; 双线性插值; 最邻近插值; 果树喷雾; 光合有效辐射

**中图分类号:** TP391.41

**文献标识码:** A

**文章编号:** 1002-6819(2006)11-0179-04

李 震, 洪添胜, 吴伟斌, 等. 双线性插值算法在荔枝树光合模拟中叶片图像旋转中的应用[J]. 农业工程学报, 2006, 22(11): 179-182.

Li Zhen, Hong Tiansheng, Wu Weibin, et al. Application of bilinear interpolation algorithm in litchi leaf image rotation for photosynthetic simulation[J]. Transactions of the CSAE, 2006, 22(11): 179-182. (in Chinese with English abstract)

## 0 引 言

精确喷雾就是根据作物或果树的不同对象随时调整, 变量喷施农药。Tian 等利用机器视觉导航的喷雾器进行区域除草, 节省除草剂达到 48%<sup>[1]</sup>。以往对于果树精确喷雾的研究主要集中于根据果树位置、形态、与喷嘴距离等进行<sup>[2-4]</sup>, 根据光合有效辐射量确定喷雾策略是精细农业研究及精确喷雾研究的一个新课题。光合有效辐射量(Photosynthetically Active Radiation, PAR)是太阳辐射中能够被绿色植物吸收并用来进行光合作用的部分, 作物生物量的积累与冠层截获光合有效辐射量(Intercepted Photosynthetically Active Radiation, IPAR)有着密切的关系<sup>[5]</sup>。因此, 可建立果树生物量与光合有效辐射吸收之间关系的模型<sup>[6]</sup>, 依据实时检测的情况对药、肥等进行变量喷施, 避免药、肥的过量喷施对环境造成影响<sup>[7]</sup>。

对荔枝树叶冠层进行模拟时, 需要对采样叶片图像随机旋转一定的角度, 使其更加接近实际的树叶分布情况。传统的最邻近插值图像旋转方法虽然容易实现且速度快, 但是旋转后的图像有明显的方块效应, 旋转后的效果比较模糊, 只适用于对图像清晰度要求不高的场合<sup>[8]</sup>。本文研究的关键是寻找合适的树叶图像旋转算法从而改善旋转后图像的质量, 引入了双线性插值算法对树叶图像进行旋转并开发了基于双线性插值算法进行树叶图像旋转的程序。

## 1 图像的旋转

本文中使用的叶片样本彩色扫描图像经模型变换(RGB→HSI)后的灰度图像。图像的旋转是以中心为原点, 旋转一定的角度<sup>[9]</sup>, 如图 1。点  $(x_0, y_0)$  经过旋转  $\alpha$  度后, 坐标变成  $(x_1, y_1)$ 。

旋转前

$$\begin{cases} x_0 = r \cos \beta \\ y_0 = r \sin \beta \end{cases} \quad (1)$$

旋转后

$$\begin{cases} x_1 = r \cos(\beta - \alpha) = r \cos \beta \cos \alpha + r \sin \beta \sin \alpha \\ \quad = x_0 \cos \alpha + y_0 \sin \alpha \\ y_1 = r \sin(\beta - \alpha) = r \sin \beta \cos \alpha - r \cos \beta \sin \alpha \\ \quad = x_0 \sin \alpha + y_0 \cos \alpha \end{cases} \quad (2)$$

上述旋转是绕坐标轴原点  $(0, 0)$  进行的, 向右为  $x$  轴正方向, 向上为  $y$  轴正方向。而屏幕中的坐标一般是以左上角为原点, 向右为  $x$  轴正方向、向下为  $y$  轴正方向。如果是绕指定点  $(a, b)$  旋转, 则先要将坐标系平移到该点, 再进行旋转, 然后平移回新的坐标原点。

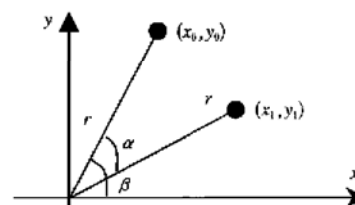


图 1 旋转图像示意图

Fig. 1 Sketch map of image rotation

假设图像在新的坐标系下, 以旋转后新图像左上角为原点, 未旋转时中心坐标为  $(a, b)$ , 旋转后中心坐标为  $(c, d)$ , 则旋转变换的矩阵表达式为

收稿日期: 2005-01-25 修订日期: 2006-06-27

作者简介: 李 震 (1981-), 男, 广州 华南农业大学工程学院, 510642

\*通讯作者: 洪添胜 (1955-), 男, 教授, 博士, 博士生导师, 广州 华南农业大学信息学院, 510642. Email: tshong@scau.edu.cn

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & c \\ 0 & -1 & d \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \quad (3)$$

因此

$$\begin{cases} x_1 = x_0 \cos\alpha - y_0 \sin\alpha + c \\ y_0 = -x_1 \sin\alpha + y_0 \cos\alpha - a \sin\alpha - b \sin\alpha + d \end{cases} \quad (4)$$

逆变换为

$$\begin{cases} x_0 = x_1 \cos\alpha + y_1 \sin\alpha - c \cos\alpha - d \sin\alpha + a \\ y_0 = -x_1 \sin\alpha + y_1 \cos\alpha + c \sin\alpha - d \cos\alpha + b \end{cases} \quad (5)$$

式(4)、(5)中,  $a = (H_0 - 1)/2$ ,  $b = (W_0 - 1)/2$ ,  $c = (H_1 - 1)/2$ ,  $d = (W_1 - 1)/2$ ;  $H_0, W_0, H_1, W_1$  分别为原图和新图的长与宽, 其中:

$$\begin{cases} H_1 = \sin(\arctan(H_0/W_0) + \alpha) \sqrt{H_0^2 + W_0^2} \\ W_1 = \sin(90^\circ - \alpha + \arctan(H_0/W_0)) \sqrt{H_0^2 + W_0^2} \end{cases} \quad (6)$$

这样, 对于新图中的每一点, 可以根据式(4) 求出对应原图中的点, 得到它的灰度, 如果超出原图范围, 则置成黑色或白色。在本文的算法中是设置为黑色。

## 2 计算旋转后图像的像素值

### 2.1 旋转后图像像素值计算中存在的问题

经过上述的计算分析, 由式(4) 可以确定新图中某一像素点在原图中所对应的位置。但是由于式(4) 计算所得的  $(x_1, y_1)$  一般不是整数, 即新图在原图中所对应的像素其坐标为非整数。计算机对图片中的像素只保存整数坐标, 这会造成即使旋转后的坐标处于原图内, 也无法获得准确的灰度值的情况<sup>[10]</sup>。其示意如图 2 所示。

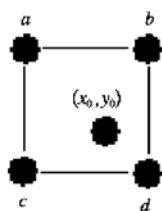


图 2 原图坐标示意图

Fig. 2 Sketch map of original coordinate

图 2 中, 点  $(x_0, y_0)$  为新图中  $(x_1, y_1)$  经过式(4) 计算得到的在原图中的位置。 $a, b, c, d$  为原图中  $(x_0, y_0)$  在一个单位范围内的 4 个整数坐标点。如果采用最邻近插值算法, 则令点  $(x_0, y_0)$  的灰度值等于与之最邻近的  $d$  点的灰度值, 再将此值送入新图的  $(x_1, y_1)$  处, 完成一个坐标点的像素赋值运算。可以看出, 在新图的某一个区域内, 都会用  $a, b, c, d$  中一个值作为其目标值。因此, 当  $a, b, c, d$  在原图中为边界点时, 在新图中会造成边界的模糊和方块效应。

### 2.2 基于双线性插值法确定旋转后图像的像素值

本文采用的是双线性插值的方法确定点  $(x_0, y_0)$  的灰度值<sup>[11]</sup>。令  $f(x, y)$  为两个变量的函数, 其在单位正方形顶点的位置已知。假设希望通过插值得到正方形内任意点  $f(x, y)$  的值, 令双线性方程

$$f(x, y) = ax + by + cxy + d \quad (7)$$

定义得一个双曲线与 4 个已知点的拟合。这个拟合的过程就是双线性插值。

在图 2 中, 分别令  $a, b, c, d$  点的灰度值为  $f(0, 0)$ 、 $f(0, 1)$ 、 $f(1, 0)$ 、 $f(1, 1)$ , 则以上 4 个已知像素点组成一个单位正方形, 点  $(x_0, y_0)$  的灰度值为  $f(x, y)$ , 得到新的示意图如图 3 所示。

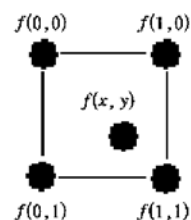


图 3 双线性插值算法示意图

Fig. 3 Sketch map of bilinear interpolation algorithm

用双线性插值法计算点  $(x_0, y_0)$  的灰度值的过程为:

首先, 对上端的两个顶点进行线性插值可得

$$f(x, 0) = f(0, 0) + x[f(1, 0) - f(0, 0)] \quad (8)$$

类似的, 对于底端两个顶点进行线性插值, 有

$$f(x, 1) = f(0, 1) + x[f(1, 1) - f(0, 1)] \quad (9)$$

最后, 做垂直方向的线性插值, 以确定

$$f(x, y) = f(x, 0) + y[f(x, 1) - f(x, 0)] \quad (10)$$

将式(8)和(9)代入式(10), 得

$$f(x, y) = [f(1, 0) - f(0, 0)]x + [f(0, 1) - f(0, 0)]y + [f(1, 1) + f(0, 0) - f(0, 1) - f(1, 0)]xy + f(0, 0) \quad (11)$$

该形式类似式(7), 因此是双线性的。利用双线性插值为新图像像素灰度赋值, 可以产生令人满意的效果。假设新图中某点的灰度值为  $G(x_1, y_1)$ , 其在原图中对应的坐标为  $(x_0, y_0)$ , 原图用灰度  $G'$  表示, 符号“ $\lceil$ ”表示取整运算, 则双线性插值的伪代码为

For  $i = 1$  to  $(W_1 - 1)$

For  $j = 1$  to  $(H_1 - 1)$

$x_1 = i, y_1 = j$ , 计算  $x_0, y_0$ ;

$f(0, 0) = G'(\lceil x_0 \rceil, \lceil y_0 \rceil); f(1, 0) = G'(\lceil x_0 \rceil + 1, \lceil y_0 \rceil); f(0, 1) = G'(\lceil x_0 \rceil, \lceil y_0 \rceil + 1); f(1, 1) = G'(\lceil x_0 \rceil + 1, \lceil y_0 \rceil + 1);$   
计算  $f(x, y); G(x_1, y_1) = \lceil f(x, y) \rceil$

End

利用 Matlab 7.0.4 编程工具开发了基于该算法的程序, 要进行旋转的叶片采样原图像、用最邻近插值算法旋转所得图像以及用双线性插值算法所得图像分别

如图 4 的 a、b、c, 旋转后所得图像的局部放大图如图 4d、4e 所示。

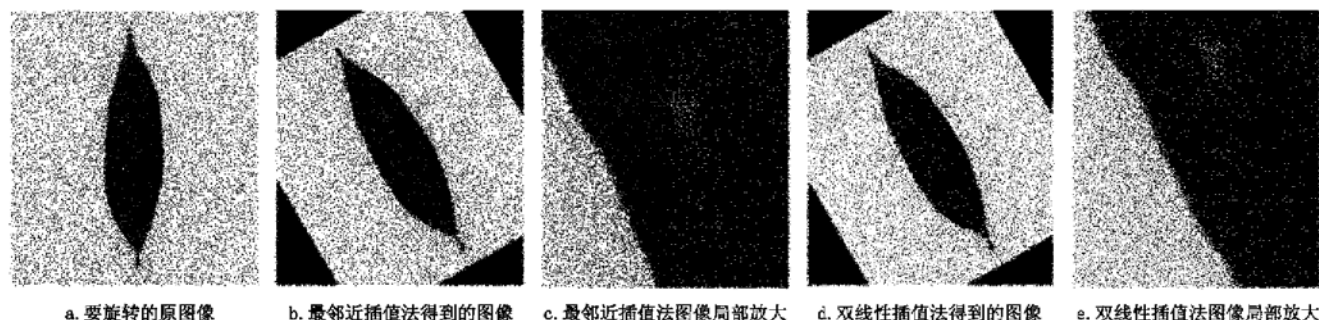


图 4 两种算法旋转所得图像的对比

Fig. 4 Comparison of rotated image using different algorithms

### 3 图像旋转结果的对比分析

分析的方法为: 分别利用最邻近插值法和双线性插值法对同一副图像进行不同角度的旋转, 将旋转前后的图像均进行二值化, 计算旋转后图像中目标元素(树叶)面积的变化。其中, 所使用图像中目标元素的初始面积为 23695 个像素, 共旋转 17 个角度, 对所得到的数据进行配对样本 T 检验, 结果如表 1 所示。

表 1 数据检验结果

Table 1 Main results of paired sample T Test

检验内容	平均值	标准差	标准误	显著值
最邻近插值	23715.24	14.948	3.625	0.000
双线性插值	23694.47	11.641	2.823	0.854
最邻近插值-双线性插值	20.765	10.437	2.531	0.000

表 1 中第 1、2 行的显著值为与原始面积 23695 进行二尾检验得出的显著值。由表 1 可以看出:

1) 采用最邻近插值法其显著值为  $0.000 < 0.05$ , 即采用该方法得出的图像与原图像之间的目标元素面积差异十分显著; 而采用双线性插值法其显著值为  $0.854 > 0.05$ , 即采用该方法得出的图像与原图像之间的目标面积无显著差异;

2) 采用双线性插值法图像之间的标准差、标准误均比采用最邻近插值法要小, 反应出采用双线性插值法所获得的图像的一致性比较好;

3) 采用最邻近插值方法得到的图像与原图间平均增加 20 个像素, 采用双线性插值方法得到的图像更接近原始值 23695。

### 4 结 论

1) 采用最邻近插值算法在边界处有比较严重的方块效应, 图像比较模糊; 采用双线性插值算法旋转后的图像边界比较清晰, 图像清楚。

2) 采用最邻近插值算法所得图像, 在二值化后目标元素的变形较大, 平均达到 20 个像素, 且其各个图像与原图像之间均有显著的差异; 采用双线性插值算法旋转后的图像在二值化后具有平均值近似等于原始值、标

准差/标准误较小(相对于最邻近插值算法所得结果)等优点, 基于该算法所得的图像进行进一步的模拟实验所得的结果更加准确。

3) 进行双线性插值图像旋转计算量较大, 所需的时间比较长, 对计算机的要求比较高。在仿真过程中应该根据实际要求(精度、次数等)选择合适的算法。

4) 经过大量实验发现旋转初始角度对于总的效率有一定的影响。进一步的研究可以针对旋转初始角度的选择(本文中的初始角度为  $0^\circ$ ), 找出最佳的旋转初始角度。

### [参 考 文 献]

- [1] Tian L, Reid J F, Hummel J W. Development of a precision sprayer for site-specific weed management[J]. Transactions of the ASAE, 1999, 42(4): 893- 900.
- [2] Molto E, Martin B, Gutierrez A. Design and testing of an automatic machine for spraying at a constant distance from the tree canopy[J]. Journal of Agricultural Engineering Research, 2000, 77(4): 379- 384.
- [3] Pergher G, Zucchiatti N, Gubiani R. Influence of spray application parameters on deposition in an asparagus crop[J]. J Agric Engng Res, 1999, (73): 19- 28.
- [4] 祈力钧, 傅泽田. 风助式喷雾器雾滴在果树上的分布[J]. 农业工程学报, 1998, 14(3): 135- 139.
- [5] 史泽艳, 高晓飞, 谢 云. 冠层底部光合有效辐射三种测量方法的比较[J]. 资源科学, 2005, 27(1): 104- 107.
- [6] Wilson T B, Norman J M, Bland W L, et al. Evaluation of the importance of lagrangian canopy turbulence formulations in a soil-plant-atmosphere model[J]. Agricultural and Forest Meteorology, 2003, 115: 51- 69.
- [7] 王万章, 洪添胜, 李 捷, 等. 果树农药精确喷雾技术[J]. 农业工程学报, 2004, 20(6): 98- 101.
- [8] 孙成叶, 桑 农, 张天序, 等. 图像双线性插值无级放大及其运算量分析[J]. 计算机工程, 2005, 31(9): 167- 169.
- [9] 杨淑莹. VC++ 图像处理程序设计[M]. 北京: 清华大学出版社/北京交通大学出版社, 2005: 50- 53.
- [10] 乐 宁, 翁世修. 基于单个字域的倾斜字符校正技术的研究[J]. 图像识别与自动化, 2002, (2): 1- 7.
- [11] 袁 泉, 李炳法. 基于频域双线性插值的数字水印算法[J]. 计算机应用, 2004, 24(1): 87- 89.

## Application of bilinear interpolation algorithm in litchi leaf image rotation for photosynthetic simulation

Li Zhen<sup>1</sup>, Hong Tiansheng<sup>2\*</sup>, Wu Weibin<sup>1</sup>, Zhang Wenzhao<sup>1</sup>, Guo Zhuo<sup>1</sup>

(1. College of Engineering, South China Agricultural University, Guangzhou 510642, China;

2. College of Information, South China Agricultural University, Guangzhou 510642, China)

**Abstract:** The purpose of this experiment was to find a suitable algorithm for leaf rotation in the Litchi canopy PAR (*Photosynthetically Active Radiation*) simulation. Bilinear interpolation algorithm was introduced to the rotation of leaf gray pictures in this study. Pictures were also rotated with the nearest interpolation algorithm for comparison. Results were analyzed emphasizing the transition of the borderline and ratio of target area in the binary image. It showed that in the rotated images, when using the bilinear interpolation algorithm, the transition between target and background was smooth, the borderline was clear and the definition of the whole image was satisfying. By using the nearest interpolation algorithm, the definition of the rotated image was not so satisfying that there were diamond effects on the borderline of target and background. It also showed that there was a distortion of 20 pixels in binary image for average when using nearest interpolation algorithm.

**Key words:** image rotation; bilinear interpolation; nearest interpolation; fruit tree spraying; photosynthetic active radiation