

# 基于图像边界跟踪的顶点矩阵算法

李雨田, 晋小莉

(西安工业大学机电工程学院, 西安 710032)

**摘要:** 分析传统图像边界跟踪中存在的问题, 根据 Freeman 链码跟踪方法提出一种改进的基于像素顶点矩阵的边界跟踪方法, 定义图像的像素顶点矩阵, 并利用像素顶点矩阵跟踪边界, 制定边界跟踪的搜索规则, 同时抽取边界的顶点链码, 生成围线树结构, 用像素顶点的链码值表示边界。实验结果表明, 该算法优于一般的链码跟踪方法, 具有广泛应用性。

**关键词:** 边界跟踪; 像素顶点矩阵; 链码

## Vertex Matrix Algorithm Based on Image Boundary Tracking

LI Yu-tian, JIN Xiao-li

(School of Electrical and Mechanical Engineering, Xi'an Technology University, Xi'an 710032)

**【Abstract】** The problems in traditional image border tracking are analyzed. According to Freeman chain code boundary tracking method, a pixel-based boundary tracking method of vertex is given. The image pixel vertex matrix is defined. The pixel matrix is used to track the boundary vertex. The boundaries track search rules is set up. The apex of the boundary chain code is abstracted, and the tree structure is generated. The pixel vertex chain code value is used to indicate the boundary. Experimental results show this algorithm is better than the Freeman chain code tracking methods, and has a wider versatility.

**【Key words】** boundary tracking; pixel vertex matrix; chain code

图像处理是计算机学科的一个重要研究领域, 也是计算机图形处理的一项核心技术。边界是图像最基本的特征, 包含了可用于识别的有用信息。边界存在于目标与背景、目标与目标、区域与区域之间, 给人们描述或识别目标以及解释图像提供了有价值的特征参数。

### 1 边界的表示方法

边界的表示方法有很多, 链码表示最早于 1961 年由 Freeman 提出, 如图 1 所示, 用 0,1,2,3 这 4 个代码表示四方链码, 用 0,1,2,3,4,5,6,7 这 8 个代码表示八方向链码。

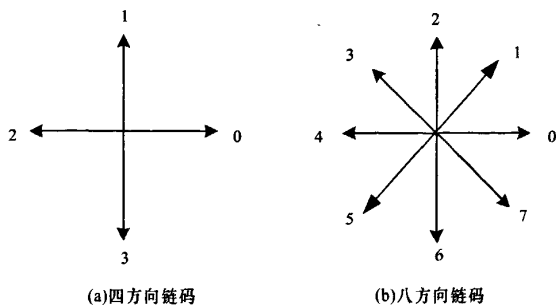


图 1 四方链码和八方向链码

从图像边界上的任一像素点开始, 按某一方向行走(逆时针或顺时针), 并用码记录相邻像素间的行走方向, 直到回到起始点为止, 就构成了图像的 Freeman 链码。图像边界可由 Freeman 链码唯一表示。

### 2 边界跟踪的研究现状与存在的问题

边界跟踪算法存在的问题主要有以下 2 个方面:

(1) 漏追踪和重复追踪, 即丢失外轮廓或内轮廓。文献[1]

为了避免丢失内轮廓, 采用边界填充的办法, 显然这很费时。

(2) 破坏连通性, 将一个轮廓跟踪成多个轮廓。为解决该问题, 文献[2]采用 2 种标记色分别标记边界像素和与边界近邻的背景像素, 以解决漏追踪和重复追踪问题。

顶点链码出现较晚, 基于顶点链码的跟踪算法也较少, 现有的边界跟踪算法大都是基于 Freeman 链码给出的, 文献[3]给出的边界标定自动机获得的是边界的顶点链码, 但该文也没有给出图像的围线树结构的生成算法。

本文给出一种能正确跟踪任意复杂图像边界的基于像素顶点矩阵边界跟踪算法, 通过跟踪边界获得边界的顶点链码。

### 3 边界跟踪与围线树结构的生成

边界跟踪包括 3 个步骤<sup>[4]</sup>:

(1) 确定作为搜索起点的边缘点, 根据算法不同, 可以是一个点或多个点, 起点的选择很重要, 整个算法对此点的依赖很大。

(2) 确定和采取一种合适的的数据结构和搜索机理, 在已发现的边界点基础上确定新的边界点, 这里要注意研究先前的结果对选择下一个检测点和下一个结果的影响。

(3) 确定搜索终结的准则或终止条件, 并在满足条件时停止进程, 结束搜索。

#### 3.1 定义像素顶点矩阵

文献[5]提出用边界像素的顶点来标记区域边界的方法。

**基金项目:** 西安工业大学校长基金资助项目“CNC 数控雕刻机技术开发与研制”(605-01000841)

**作者简介:** 李雨田(1981-), 男, 硕士研究生, 主研方向: 数控技术, 图像处理; 晋小莉, 教授

**收稿日期:** 2009-09-20 E-mail: liyutian721@163.com

对于正四边形点阵上的图像,可以有3种不同性质的顶点,如图2所示,分别用代码1,2,3作标记。

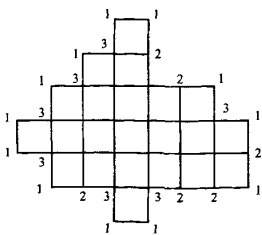


图2 图像边界的顶点链码

在图2所示的3种边界像素顶点的码值表示基础上,本文定义图像中的所有可能的像素顶点的链码值,如图3所示。其中,一个方格表示一个像素,箭头右端的数值为像素顶点对应的顶点链码值。

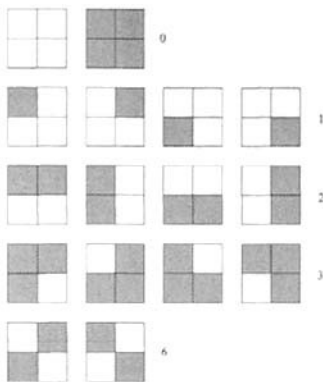


图3 像素顶点与其对应的链码值

### 3.2 边界跟踪与顶点链码的抽取

利用像素顶点矩阵给出边界的跟踪与顶点链码的抽取算法。按从上到下,从左到右的顺序遍历像素顶点矩阵 $M$ ,当 $M[u][v]=1,2,3$ 或6时, $(u,v)$ 点为图像边界上的点,这时追踪该区域边界,抽取该区域边界的顶点链码。

下面给出图像的边界跟踪与顶点链码抽取算法。

(1)确定起始点和初始行走方向。根据顶点链码的特性,按从上到下,从左到右的顺序遍历像素顶点矩阵,遇到的第1个边界像素顶点只有2种可能,其对应的顶点链码为1或3,所以,当遍历像素顶点矩阵,遇到链码1,3或6时,说明当前位置在图像区域的边界上,那么就开始跟踪该区域边界。当链码为1时,则该起始点在连通区域的外边界上,当链码为3或6时,则该起始点在图像区域的内边界上。因为按逆时针跟踪区域边界,当起始顶点的码值为1时,则设定初始行走方向为3,当起始顶点链码为3或6时,则设定初始行走方向为0。

(2)图像边界的跟踪搜索规则。图像的边界跟踪搜索规则包括边界跟踪的坐标迁移规则和行走方向变化规则。假设当前顶点坐标为 $P_i(u_i, v_i)$ ,在 $P_i$ 的行走方向为 $e_i$ ,像素顶点矩阵在 $P_i$ 的取值为 $c_i$ ,那么从当前点迁移到下一点的坐标 $P_{i+1}(u_{i+1}, v_{i+1})$ 和行走方向 $e_{i+1}$ 可由下面的坐标迁移规则和行走方向变化规则求出。

1)坐标迁移规则:当 $e_i=0$ 时,有 $u_{i+1}=u_i+1, v_{i+1}=v_i$ ;当 $e_i=1$ 时,有 $u_{i+1}=u_i, v_{i+1}=v_i-1$ ;当 $e_i=2$ 时,有 $u_{i+1}=u_i-1, v_{i+1}=v_i$ ;当

$e_i=3$ 时,有 $u_{i+1}=u_i, v_{i+1}=v_i+1$ 。

2)行走方向变化规则:当 $c_{i+1}=1$ 时,有 $e_{i+1}=(e_i+1)\%4$ ;当 $c_{i+1}=3$ 或 $c_{i+1}=6$ 时,有 $e_{i+1}=(e_i+3)\%4$ ;当 $c_{i+1}=2$ 时,有 $e_{i+1}=e_i$ 。

(3)边界的顶点链码的跟踪抽取及终止条件。根据(1)提供的起始点坐标、初始行走方向,按(2)给的坐标迁移规则和行走方向变化规则,可以沿图像区域边界顶点行走,当每走过像素顶点时,就记录该点的顶点链码值,这里当走过顶点的顶点链码值为6时,该顶点的顶点链码值作为3被记录下来,回到起始点时结束,这样就获得了该区域边界的顶点链码序列。加上起始点和初始行走方向,就获得了区域边界的顶点链码表示。

### 3.3 跟踪图像边界与定义围线树结构

上文给出了单条区域轮廓线的跟踪与顶点链码抽取算法,要跟踪图像所有的区域边界,还要解决发现新的区域边界、避免边界的漏跟踪、重复跟踪和生成边界的围线树结构的问题。

(1)发现新的区域边界。按从上到下,从左到右的顺序遍历像素顶点矩阵,当遇到值为1,3或6的顶点时,那么该点就是新边界的开始点,就可以开始追踪该边界,抽取该边界的顶点链码。

(2)避免边界的漏跟踪、重复跟踪和破坏连通性问题。当跟踪了一条边界后,通过更改像素顶点矩阵在该点的值,解决边界的漏跟踪、重复跟踪和破坏连通性的问题。在像素顶点矩阵中,当经过的点的值为1,2或3时,该值作为顶点链码值被记录下来,然后把该点的值改为0或者改为边界的跟踪顺序值;当经过的点的值为6时,该点的顶点链码值作为3被记录下来,然后把该点的值改为3。这样跟踪过的边界就不会再一次当作新的边界被发现。

(3)定义边界的围线树结构。定义围线节点类CLayer,成员变量 $m\_sChaincode$ 为该围线的顶点链码序列,成员变量 $m\_pParent$ 是个指向父节点指针,成员变量 $m\_Layerlist$ 是个围线类指针数组,它保存了该围线所包含的所有子围线的围线类指针。

定义围线类如下:

```

Class CLayer {
    int m_imgWidth, m_imgHeight;
    //图像的宽和高
    int m_direction; //顶点链码的初始方向
    int m_x; //顶点链码的起始点坐标的x分量
    int m_y; //顶点链码的起始点坐标的y分量
    CString m_sChaincode;
    //图像区域边界的顶点链码序列
    CArray<CLayer*, CLayer*> m_Layerlist;
    //该围线所包含的字围线的围线类指针数组
    CLayer* m_pParent;
    //包含该围线的父围线类指针
};
    
```

## 4 实验结果

实验表明,利用基于像素顶点矩阵的边界追踪算法能正确地跟踪图像区域边界,获得围线的树结构。图4为放大并显示网格的原始图像。图5为利用本文算法跟踪图所示的区域边界获得顶点链码及其链码树结构,其中,上箭头指向父围线,下箭头指向节点链码表示的边界所包含的子围线。

(下转第255页)

### 3.3 用户空间电源管理服务守护进程的建立

高级电源管理的用户空间守护进程为 apmd。当用户空间执行命令 `apm -s` 时，对应内核空间调用函数 `apm_suspend`，进而调用 `pm_suspend` 进入内核电源管理子系统。通过 `apmd` 可以显示系统的电池电量及充电状态，并将电池状态记录到系统日志，当电量状态变化时采取相应动作。

本文实现了一个用户空间电源管理服务守护进程，当系统空闲(没有外部事件触发或正在运行程序)时间超过由固定超时策略预知的时间间隔后执行命令 `apm -s` 使系统进入休眠，然后重新设置定时器。

### 4 实验结果与分析

实验硬件平台采用 YLP2440 开发板，S3C2440 处理器，64 MB sdram，128 MB nandflash，3.5 寸 sharp 240x320LCD 液晶屏，音频接口采用 IIS 接口芯片 `uda1341`，采用 DM9000AEP 的 10/100 Mb/s 网口，两通道串口，USB1.lhost，device 接口等。本文采取如下实验方案，将 YLP2440 开发板与稳压电源相连，以 12 V 直流电压作为系统输入。YLP2440 开发板内含有 LM2596 电源转换电压调整芯片。该芯片在 5 V 输出时具有 80% 的转换效率，同时可以读出系统工作电流。分别在 S3C2440 处理器工作频率为 405 MHz 和 271 MHz 情况下测量该系统在空闲状态和休眠状态下的耗电量，以电流表示，原始数据如表 1 所示。

表 1 系统空闲状态功耗与休眠状态功耗测量

S3C2440 工作频率/MHz	空闲状态 耗电量/mA	休眠状态 耗电量/mA	电源管理 节能百分比/(%)
405	300	35	11.67
271	270	35	12.96

实验数据表明，当处理器工作频率降低后，系统功耗降低。系统进入休眠状态以后功耗已不受处理器频率影响，这与前文分析的在休眠状态下处理器将切断内部供电，仅保留

唤醒逻辑(wake-up logic)模块的结论一致。实验数据表明，当 S3C2440 处理器工作在典型的工作频率(405 MHz 和 270 MHz)时，系统的电源管理功能可以使休眠后的功耗减小到系统空闲状态电量的 11.67% 和 12.96%，从而大大降低了嵌入式系统的功耗。

### 5 结束语

本文研究了嵌入式 Linux 系统中的高级电源管理机制，设计并实现了一套完整的电源管理软件系统。实验结果表明，该电源管理系统可以有效地减少系统功耗，从而延长便携式手持设备的电池使用时间。电源管理是一项硬件与软件相结合的系统工程，在今后的研究中将着重实现嵌入式系统的动态电源管理功能，以进一步降低嵌入式系统的功耗。

### 参考文献

- [1] Daisuke M, Yutaka I. Process Oriented Power Management[C]//Proc. of the 2nd International Symposium on Industrial Embedded Systems. Lisbon, Portugal: [s. n.], 2007.
- [2] 吴慧良, 史烈, 陈小平等. Linux 下休眠功能的软件实现[J]. 计算机工程, 2002, 28(4): 235-236.
- [3] 吴快快, 都思丹. 基于 Linux 的便携嵌入式设备电源管理解决方案[J]. 微处理机, 2005, 26(5): 66-68.
- [4] 阳富民, 梁晶, 张杰. 嵌入式 Linux 电源管理技术的研究实现[J]. 计算机工程与科学, 2004, 26(12): 91-93.
- [5] 周建政, 吕柏权. 基于 Linux 嵌入式的电源管理的研究[J]. 仪表技术, 2008, 1(1): 59-61.
- [6] Corbet J, Kroah-Hartman G, Rubini A. Linux Device Driver[M]. [S. l.]: O'Reilly & Associates, Inc., 2005.
- [7] Love R. Linux Kernel Development[M]. [S. l.]: Sams Publishing, 2005.

编辑 顾姣健

(上接第 232 页)



图 4 原始图像

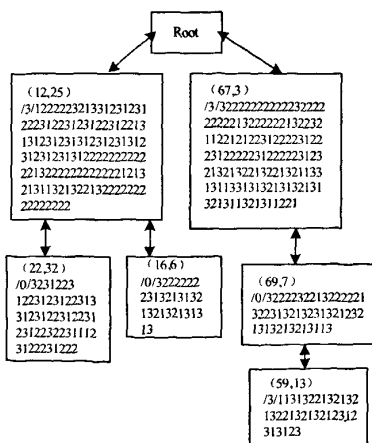


图 5 边界顶点链码表示及其树结构

### 5 结束语

对比现有的边界追踪算法，本文算法提供的边界追踪算法具备以下特点：(1)算法适用任意形状的图像区域；(2)算法给出了区域边界的顶点链码表示；(3)算法是线性的；(4)算法给出了图像区域边界的围线树结构。实验结果表明，该算法能正确跟踪复杂的图像，且具有容易实现、效率较高等优点。通过围线树结构，还可以得到区域边界的其他链码表示及其树结构，具有一定实用价值。

### 参考文献

- [1] 陆宗琪. 图像处理领域轮廓跟踪及应用[J]. 中国计算机用户, 1994, 10(5): 49-54.
- [2] Ren Mingwu, Yang Jinyu, Sun Han. Tracing Boundary Contours in a Binary Image[J]. Image and Vision Computing, 2002, 20(2): 125-131.
- [3] Rosenfeld A. Adjacency in Digital Picture[J]. Information and Control, 1974, 26(1): 24-33.
- [4] 章毓晋. 图像分割[M]. 北京: 科学出版社, 2001.
- [5] Bribiesca E. A New Chain Code[J]. Pattern Recognition, 1999, 32(2): 235-251.

编辑 陈文