

基于多线程的分段图像轮廓跟踪算法

苏成顺¹, 李贞培²

(1. 秦皇岛港股份有限公司 第一分公司, 河北 秦皇岛 066000;

2. 西北工业大学 自动化学院, 陕西 西安 710072)

摘要:针对将轮廓分段,并对每段单独进行或全部区段同时进行轮廓跟踪的需求,提出了基于多线程的分段图像轮廓跟踪算法,采用C#语言进行了程序实现。并以面向对象的观点,将轮廓跟踪操作封装成类。一段轮廓的跟踪对应一个线程和一个轮廓跟踪类的对象。应用表明,基于多线程的分段图像轮廓跟踪算法很好地解决了图像分段轮廓跟踪的问题,不仅可以灵活地对整个轮廓中所关心的某几段或所有段进行跟踪,而且各段轮廓的跟踪过程能够相互通信,跟踪速度快、效率高。

关键词:多线程;分段;轮廓跟踪;算法

中图分类号:TN911.73

文献标识码:A

文章编号:1673-629X(2009)10-0099-03

Algorithm for Multi-Segment Image Contour Following Based on Multithreading

SU Cheng-shun¹, LI Zhen-pei²

(1. The First Branch Company of Qinhuangdao Port Holdings Co., Ltd, Qinhuangdao 066000, China;

2. College of Automation, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract:Based on the requirements of dividing image contour to several segment and following one segment alone or following all segments simultaneously, an algorithm for multi-segment image contour following based on multithreading was presented, and was implemented by C# language. The operation of contour following was encapsulated to a class by the concepts of OOP. Contour following of one segment is executed by one thread and one instance of contour following class. Application results show that the proposed algorithm is a practicable solution to the multi-segment image contour following. With this solution the concerned segments in whole contour can be tracked separately, the processing of one segment's contour following can communicate with another's. The algorithm also has the advantage of fast speed and high efficiency.

Key words:multithreading; multi-segment; contour following; algorithm

0 引言

在基于栅格图像的地理信息系统(GIS)空间分析中,通常需要获得一个区域的轮廓以进行长度面积等分析,而这个区域又分属几个地区,需要对每个地区的轮廓进行单独提取,如图1所示,目标区域分属A、B区,需要对某个区单独进行或对全部区同时进行轮廓跟踪。目前,有关轮廓跟踪与特征提取的方法已有不少,但大都针对单区段的轮廓跟踪^[1~4]。文献[5]针对图像尺寸检测,提出了一种图像轮廓跟踪算法,该算法

检测出的图像轮廓定位准确、连续、光滑、边缘细节丰富、没有假边缘。同时,算法具有很好的抗噪能力,几乎可以完全消除噪声对图像的影响。文献[6]提出了均值漂移跟踪算法,该算法可以较准确地对目标物体进行跟踪,但是不能够确切得到目标物体的轮廓信息。由于多线程具有多任务、并发性、线程间易于通信等特点,能满足对多区段轮廓同时或单独进行轮廓跟踪的需求,因此,基于八邻域轮廓跟踪算法^[7],文中提出了基于多线程的分段图像轮廓跟踪算法,并进行了程序实现。

1 算法描述

文中采用的轮廓跟踪算法是八邻域轮廓跟踪算法,它的基本思想是:按某种严格的“探测准则”找到目

收稿日期:2009-02-05;修回日期:2009-05-24

基金项目:秦皇岛港务公司重点项目(2006J112002)

作者简介:苏成顺(1980-),男,辽宁辽阳人,助工,从事油气储运自动化、管道数字化等方面的工作;李贞培,硕士,从事数字图像处理、地理信息系统方面的研究工作。

标图像轮廓上的像素,再根据这些像素的某些特征用一定的“跟踪准则”找出目标图像上的其它像素。以黑白二值位图封闭区域轮廓跟踪为例,算法描述如下:

(1)按某种严格的“探测准则”找到目标图像上的第一个边界像素(设图像为二值图,且轮廓颜色为黑色),如按从左到右,从上到下的顺序,找到的第一个黑点一定是最左上方的边界点,记为 A 点;

(2)A 点的右、右下、下、左下四个邻点中至少有一个是边界点,记为 B 点;

(3)从 B 点起,按右、右下、下、左下、左、左上、上、右上的顺序找相邻点中的边界点 C;

(4)如果 C 就是 A 点,则表明已经转了一圈,程序结束;否则从 C 点继续找,直到找到 A 为止。

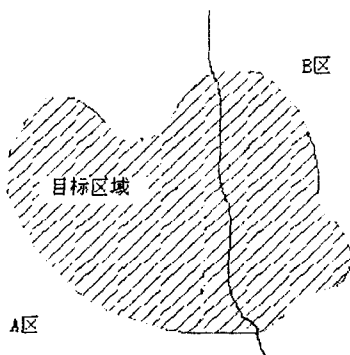


图 1 轮廓分区段示意图

判断一个点是不是边界点即“跟踪准则”是:如果它的上下左右四个邻接点都是黑点则不是边界点,否则是边界点。

在八邻域轮廓跟踪算法的基础上,基于多线程的分段图像轮廓跟踪算法描述如下:

1)查看目标区域被几个区所分割,每个区分割的轮廓为一段;

2)记录每段的首尾像素坐标;

3)为每一段启动一个线程,从首像素到尾像素用八邻域轮廓跟踪算法进行跟踪轮廓。

跟踪轮廓完毕后即可对各段轮廓进行其他处理。

2 程序实现

笔者采用 C# 语言对此算法进行了程序实现,并以面向对象的思想,将轮廓跟踪封装成类 Contour,便于代码的重用。类成员及意义见表 1。

由于轮廓跟踪类 Contour 包含轮廓区段的首尾坐标属性,且有轮廓跟踪行为(DoContour 函数),所以一个区段的轮廓跟踪过程,就是轮廓跟踪类 Contour 的实例化及调用 DoContour 方法的过程。多区段的轮廓跟踪过程,就是为每一区段实例化相应的轮廓跟踪类

Contour 实例,并启动线程进行轮廓跟踪的过程。整个过程用伪代码描述如下:

表 1 轮廓跟踪类 Contour

Contour	
变量	byte[] rgbValues; //原图数据字节数组
	byte[] rgbEdgeValues; //接收轮廓跟踪后新图数据的字节数组
	int stride; //图像每行字节数
	int x;
	int y; //起点坐标
	int x1;
	int y1; //终点坐标
	Color colorEdge; //轮廓像素颜色
方法	string segmentID; //区段标识
	Contour(byte[] rgbValues, byte[] rgbEdgeValues, int stride, int x, int y, int x1, int y1, Color colorEdge, string segmentID) //构造函数
	void DoContour() //轮廓跟踪函数;进行轮廓跟踪,并将轮廓跟踪后新图数据拷贝到 rgbEdgeValues 字节数组
	bool IsContourP(byte[] rgbValues, int x, int y, Color colorEdge) //判断一个点是否为边界点函数

将轮廓分段;

记录每段首尾坐标;

for(i=0; i<区段数; i++)

{ Contour contour = new Contour(...);

Thread thread = new Thread(new ThreadStart(contour.DoContour));

处理包含 i 段轮廓跟踪后数据的字节数组;

}

其中,轮廓跟踪函数 Docontour 的核心伪代码如下:

处理轮廓起点(置为黑点等);寻找轮廓起点右、右下、下、左下四个邻点中边界点,置为当前轮廓点;记录当前轮廓点坐标;

while(当前轮廓点不等于终点坐标)

{ 处理当前轮廓点;按右、右下、下、左下、左、左上、上、右上的顺序找相邻边界点

for(从当前坐标的右方开始,直到右上方,按顺时针方向依次旋转)

{ 计算下一点的坐标,置为当前点;

if((原图中的当前点为轮廓色点,且新图中的当前点还没搜索过,)或者(当前点是轮廓起点))

{

if(当前点是边界点)

{

记录当前轮廓点坐标;

退出 for 循环;

}

}

}

3 算法应用

中文提出的算法已应用于笔者实现的数字管道系统的 GIS 模块中。对于图 2 所示的栅格地图^[8], 可以看到, 东湖被明县和津县所分割, 需分别计算分属明县和津县的东湖湖岸线长度, 笔者采用提出的算法, 将湖轮廓分为两段, 每段首尾点坐标分别为两县县界与湖岸线的交点, 启动两个线程进行轮廓跟踪, 轮廓跟踪结果如图 3 及图 4 所示。

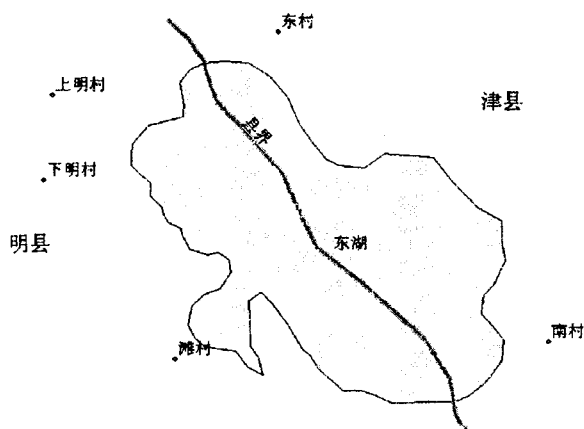


图 2 东湖栅格地图

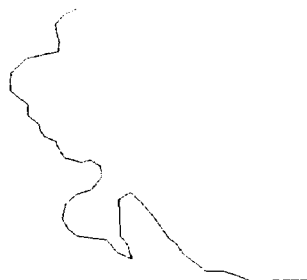


图 3 明县湖岸轮廓

应用结果表明, 基于多线程的分段图像轮廓跟踪算法能很好地满足多区段轮廓跟踪的需求, 有较高的工程应用价值。

4 结束语

文中基于八邻域轮廓跟踪算法, 提出基于多线程的分段图像轮廓跟踪算法, 很好地解决了多区段图像

轮廓跟踪的问题。算法采取了多线程的方法, 由于多线程使用同一地址空间, 可以共享进程的资源, 如全局变量、资源等, 因此线程间可以很好地进行通信, 当一个线程的轮廓跟踪进行完毕, 可以通知主线程或其他线程进行其他处理; 且轮廓跟踪线程均在后台运行, 并不影响主线程的工作。这些特点提高了程序性能和效率, 体现了算法的优越性。



图 4 津县湖岸轮廓

参考文献:

- [1] Paragios N, Deriche R. Geodesic Active Regions and Level Set Methods for Motion Estimation and Tracking[J]. Computer Vision and Image Understanding, 2005, 97(3): 259-282.
- [2] 余九华, 王敬东, 李 鹏. 基于 Camshift 的人脸跟踪算法[J]. 计算机技术与发展, 2008, 18(9): 12-15.
- [3] 周丰乐, 徐向民, 肖 跃. 一种新的二值图像目标轮廓跟踪算法[J]. 微计算机信息, 2007(2): 259-261.
- [4] Zheng L, Han C Z. A Multi-mode Image Tracking System Based on Distributed Fusion[C]//Proceedings of the 5th international conference on information fusion. USA: Omni press, 2002: 832-837.
- [5] 周 猛, 李 钢. 一种具有抗噪声干扰的图像轮廓跟踪算法的研究[J]. 计算机技术与发展, 2006, 16(9): 21-23.
- [6] Comaniciu D, Ramesh V, Meer P. Kernel-based Object Tracking[J]. IEEE Trans Pattern Analysis Machine Intelligence, 2003, 25(5): 564-575.
- [7] 何 斌, 马天予, 王运坚, 等. Visual C++ 数字图像处理[M]. 北京: 人民邮电出版社, 2002.
- [8] 马 瑾, 陈立潮, 张永梅. 轮廓跟踪与边沿检测的图像自动识别[J]. 中北大学学报: 自然科学版, 2006, 27(5): 431-435.

(上接第 98 页)

- [9] Lanckriet G, Cristianini N, Bartlett P, et al. Learning the kernel matrix with semidefinite programming[J]. Journal of Machine Learning Research, 2004, 5: 27-72.
- [10] Bezdek J, Hathaway R. Convergence of alternating optimiza-

tion[J]. Neural, Parallel & Scientific Computations, 2003, 11: 351-368.

- [11] Scholkopf B, Smola A. Learning with Kernels[M]. Cambridge, MIA: MIT Press, 2002.