

基于 FPGA 的二值图片连通区域标记算法

马毅超¹, 陶亚凡¹, 吴龙胜², 洪鑫扬¹, 李 寻¹, 何立风¹

(1 陕西科技大学 电气与信息工程学院, 陕西 西安 710021;

2 西安微电子技术研究所, 陕西 西安 710054)

摘 要: 提出一种基于 FPGA 的二值图片连通区域标记算法, 每次提取 9 个像素信息, 充分利用 FPGA 的并行处理能力, 在一个时钟周期内完成 9 个像素的处理, 在对图像进行遍历的同时进行快速标记. 最后搭成简单测试电路. 在 PC 上将二值图片通过 USB 传输给 FPGA, 然后 FPGA 将图片进行标记, 标记完成后将标记的图片和所用的时间返回 PC. 将实验结果与现有几种图像处理方式进行比较, 发现 FPGA 在处理图片能力与软件处理方式能力相当, 主频、功耗远低于软件处理设备, 且成本低, 可以作为专用图像处理设备.

关键词: 计算机视觉; FPGA; 二值图片; 连通区域; 区域标记

中图分类号: TP391.41

文献标识码: A

文章编号: 1000-7180(2017)01-0119-04

FPGA-Based Binary Connected Component Detection and Labeling Algorithm

MA Yi-chao¹, TAO Ya-fan¹, WU Long-sheng², HONG Xin-yang¹,
LI Xun¹, HE Li-feng¹

(1 College of Electric and Information Engineering, Shaanxi University of Science & Technology,
Xi'an 710021, China; 2 Xi'an Microelectronics Technology Institute, Xi'an 710054, China)

Abstract: A FPGA-based binary connected component detection and labeling algorithm is proposed, which Extracting 9 pixel information at one time, it makes full use of the parallel processing ability of FPGA, Complete 9 pixel processing in one clock cycle and quickly marks the image while scans. Finally, build a simple test circuit, transmit the two values image to FPGA from PC, and then FPGA will mark the image. When the mark is finished, FPGA will return the marked image and the time it is used to PC. Comparing the experimental results with the existing image processing methods, it is found that the FPGA can be used as a special image processing equipment for it's low cost and main frequency, power consumption is much lower than Software processing equipment when the speed that FPGA processes images is equivalent with Software processing method.

Key words: computer vision; FPGA; two value image; connected region; regional marker

1 引言

计算机视觉处理中的一大难点在于进行大量的数据采集和信息处理^[1], 常用的图像处理是计算能力有限且功耗高体积大的 CPU 和 GPU. 随着半导体工艺的高速发展, FPGA 的高集成度、低功耗成为其图像处理的重要选择^[2].

而在图像分析中, 二值图片连通区域的标记起着重要作用^[3]. 目前, 很多连通标记的算法采用简单直观的像素标记法^[4]. 传统的二值图像标记法有递归法、深度优先搜索法、广度优先搜索法^[5].

为实现在 FPGA 上快速标记连通区域, 在现有算法^[6-7]的基础上进行改进, 提出一种适用于 FPGA 的算法.

2 算法描述

2.1 算法概述

本算法首先需要在原图像外增加一圈背景像素(用 0 表示),然后整体记为 $I_{x \times y}$ (x 和 y 分别表示图片像素的行数和列数). 连通规则采用 8 邻域连通方式. 对图片的遍历共需两次,第一次遍历寻找图片中的连通区域并进行标记,标号 L 从 1 递增,同时创建新的等价标号链表. 当发现两个已经标记的区域是连通的时,合并两个对应的等价标号链表. 第二次遍历根据等价标号链表修正连通区域的标号.

2.2 算法流程

2.2.1 第一次遍历

图片的遍历顺序为从上到下,从左到右,每次提取三乘三的区域,记为 img , img 的每个元素如图 1 所示.

$img(1,1)$	$img(1,2)$	$img(1,3)$
$img(2,1)$	$img(2,2)$	$img(2,3)$
$img(3,1)$	$img(3,2)$	$img(3,3)$

图 1 img 示意图

行遍历时 img 中每个元素的列地址每次加一,行遍历结束后, img 中每个元素的行地址加二,列地址从左到右分别变为 1、2、3. 根据 img 中元素值的不同,将 img 共分为 5 种情况,记做 5 种状态,如图 2 所示,其中,0 表示对应像素为 0, L 表示对应像素已标记,每种情况第一列和第一行均已经遍历过, DC 表示不关心,即无论是 0 或 L 均为同一种状态情况,后两列的空白表示未知.

DC			0	L		0	0		L		DC		
0			L			L			L		0		
0			DC			DC			DC		L		
状态1	状态2	状态3	状态4	状态5									

图 2 五种状态示意图

每个状态任务相同,即搜索新的连通区域、与已标记连通区域连通的新像素以及合并不同标号的连通区域. 不同的状态只是搜索方式不同,搜索完成后根据现有情况决定下次进入的状态. 每次行遍历第一次提取数据时,由于提前添加了背景像素,状态为

状态 1,其处理过程如下:

判断 $img(2,2)$ 是否为 1,若 $img(2,2)$ 为 1,则判断该点是否与已经标记过的区域相连通,根据 $img(1,1)$ 、 $img(1,2)$ 、 $img(1,3)$ 的不同情况,列为如表 1 所示的 5 种可能,其中 DC 表示不关心,每种可能分别对应下述某个执行过程.

表 1 img 第一行 3 个像素的 5 种情况和对应执行过程

$img(1,1)$	$img(1,2)$	$img(1,3)$	执行过程
DC	1	DC	(1)
1	0	1	(2)
0	0	1	(3)
1	0	0	(4)
0	0	0	(5)

(1)将 $img(1,2)$ 标号赋予 $img(2,2)$,如果 $img(3,2)$ 为 1,同样赋值此标号,为 0 不作处理,读取下次所需数据后进入状态 4.

(2)将 $img(1,3)$ 标号赋予 $img(2,2)$,如果 $img(3,2)$ 为 1,同样赋值此标号,为 0 不作处理,然后开始合并 $img(1,1)$ 和 $img(1,3)$ 标号所对应的等价链表的处理(等价标号链表的合并详见下文),读取下次所需数据后进入状态 2.

(3)将 $img(1,3)$ 标号赋予 $img(2,2)$,如果 $img(3,2)$ 为 1,同样赋值此标号,为 0 不作处理,读取下次所需数据后进入状态 2.

(4)将 $img(1,1)$ 标号赋予 $img(2,2)$,如果 $img(3,2)$ 为 1,同样赋值此标号,为 0 不作处理,读取下次所需数据后进入状态 3.

(5) $img(2,2)$ 上边三个点均未标记,若 $img(2,2)$ 为 1,说明发现了一个新的区域,此时赋予此点一个新的标号,并创建一个新的等价标号链表(等价标号链表的创建详见下文),如果 $img(3,2)$ 为 1,同样赋值此新的标号,为 0 不作处理,然后读取下次所需数据后进入状态 3;若 $img(2,2)$ 为 0,则继续判断 $img(3,2)$ 的值,根据其值不同,共两种可能, $img(3,2)$ 为 1 时执行下述的过程(a),否则执行过程(b).

(a) $img(3,2)$ 与以遍历过的区域不连通,说明发现了一个新的区域,此时赋予此点一个新的标号,并创建一个新的等价标号链表,读取下次所需数据后进入状态 5.

(b)读取下次所需数据后继续进入状态 1.

其他状态的处理过程与状态 1 类似. 此处不再赘述.

2.2.2 等价标号链表的创建及合并

首先进行等价标号的解释:一个标号相同区域

的等价标号是与之连通的所有标号不同的区域中标号最小的区域的标号,即,假设有四个标号不同的区域 i, j, k, l 是连通的,若 i 最小,则这四个区域的等价标号均为 i 。显然,一个新标记的区域的等价标号等于其自身。

等价标号链表的每一个节点由 last、label、next 三个数组中相同下标的元素组成,为了表示方便,称此下标为该节点的下标,该节点的等价标号为该节点的 label,等价标号相同的节点构成一条等价标号链表,且称此等价标号为该链表的等价标号。等价标号链表的存储格式如图 3 所示。其中, label 中存储等价标号(也是链表头节点的下标),链表头结点的 last 存储尾节点的下标,next 存储链表中下一个节点的下标。

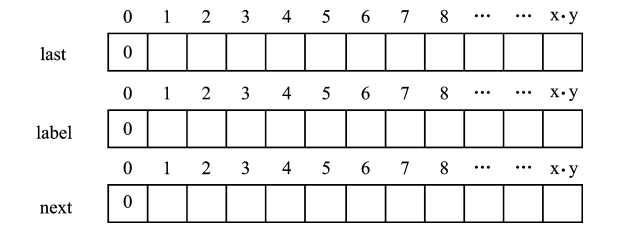


图 3 等价标号链表存储格式

等价标号链表的创建及合并:第一次遍历时,当发现新的连通区域时,给其标号为 L ,并创建新的等价标号链表,新链表只有一个节点,其 last、label、next 在三个数组中的下标均为 L ,且 $last(L) = L$, $label(L) = L$, $next(L) = 0$ 。当发现两个已标号的区域连通后,合并对应的等价标号链表,表示其连通,合并过程如下,设两个已标号的区域的等价标号分别为 i 和 j 。

(1) 判断 i 和 j 是否相等,若相等,说明两区域标号相同,退出合并等价标号链表过程;若不相等,则令 i 等于 i 和 j 的较小值, j 等于较大值,此时需要将等价标号为 j 的链表(下文简称为链表 j)的所有节点的等价标号更改为 i ,定义 $k = j$,进入步骤(2)。

(2) 令 $label(k) = i$,然后 $k = next(k)$,进入步骤(3)。

(3) 判断 k 是否等于 0,若等于零,则已经遍历完链表 j ,此时已将该链表的等价标号全部更改为 i ,进入步骤(4) 合并两个链表;否则,说明未全部遍历,此时遍历下一节点,即再次进入步骤(2)。

(4) 令 $next(last(i)) = j$, $last(i) = last(j)$ 。即将链表 i 的尾节点的 next 指向链表 j 的头节点,将链表 i 的头节点的 last 指向链表 j 的尾节点。然后合并

过程全部完成。

2.2.3 第二次遍历

第一次遍历完成后,每个标号的等价标号即为与之连通的最小标号区域的标号,所以只需要用等价标号将原有标号覆盖即可,所以第二次遍历时,对于标号为 i 的区域所做的操作便是令 $i = label(i)$,遍历完成后便完成了全部标记处理。

3 算法的 fpga 实现

3.1 概述

一般算法都在 CPU 上进行实现,但是由于结构所限,所有指令只能一个一个执行,但 FPGA 具有并行操作的先天优势,每次状态判断只需要一个时钟周期,即可以在一个时钟周期内标记多个像素点,下面介绍如何利用 FPGA 的并行处理能力实现上述算法。

3.2 实现过程

3.2.1 硬件结构

为了验证上述算法的功能及处理速度,设计一简单硬件结构,框图如图 4 所示,其中 PC 用来提供二值化图片,通过 USB 发送给 FPGA,同时发送指令,接收 FPGA 的应答信号和最终处理完成后的图片,SRAM 用来存储二值化图片信息,且作为 FPGA 处理图片时接收数据的源头和存储数据的目的的。

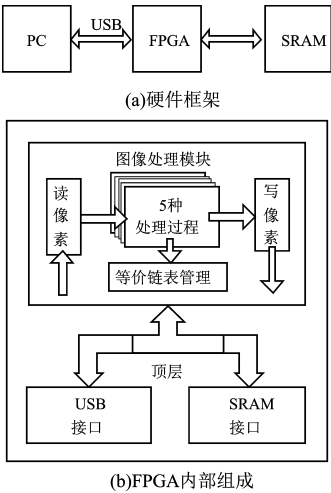


图 4 硬件结构框图

3.2.2 操作步骤

(1)PC 给 FPGA 发送二值图片,FPGA 将图片存储在 SRAM 中。

(2)FPGA 从 SRAM 中读取图片并开始处理,再将处理完成的图片存入 SRAM。

(3)FPGA 将 SRAM 中处理完成的图片和处理图片所用的时间发送给 PC.

3.3 验证结果

为了验证本文提出的算法在 FPGA 上运行的效果,选取 512×512 的随机噪声二值图像进行测试,共使用 21 张图片,其目标像素占总像素的比例(下文以目标像素的密度表示)从 0 到 1,以 0.05 为步长递增.选用 Altera 公司的 Cyclone IV E 系列的 EP4CE115F29C7 型 FPGA,分别以主频 50 MHz 和 500 MHz 处理图片,与目前较为先进的文献[8]提出的算法在 Intel Pentium D 930 3.0 GHz + 3.0 GHz CPUs, 2 GB 内存, Mandriva Linux OS 的 PC 环境下的运行时间进行对比.测试结果如图 5 所示,其中,为表示方便,以“本文 50 MHz”和“本文 500 MHz”表示主频分别为 50 MHz 和 500 MHz 时 FPGA 以硬件方式处理图片的结果,以“文献[8]中的算法”表示以软件方式处理图片的结果.

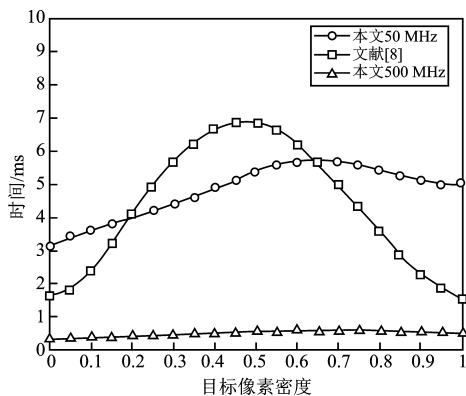


图 5 图片处理速度与目标像素密度的关系

由计算结果可见,在主频 50 MHz 时,在目标像素少于 20%和多于 65%时,图片中连通区域数量较少,对图片处理的大部分时间消耗在像素的提取和判断,本文提出的硬件处理方式由于主频慢于文献[8]的软件处理方式,所以在总的时间上也较长,但本文提出算法的处理速度受目标像素密度影响较小,在连通区域数量较多时,软件处理方式对于连通区域的处理耗时较长,所以在目标像素密度在 20%和 65%之间时,本文所用处理方式处于优势.而且当主频为 500 MHz 时,速度远快于软件处理方式,可见本文提出的处理方式非常实用与复杂的图像的

处理,可以作为专用的图像处理设备.

4 结束语

本文在部分现有算法的基础上进行改进,提出一种适用于 FPGA 的二值图片连通区域标记算法,充分利用其并行处理能力,实现二值图片连通区域的快速标记,根据标记后的区域可以提取出原物体的面积、周长、重心等参数,用于后续更高层次的操作.且由于本文提出的方式处理速度快,可用于实时性较强的应用中.

参考文献:

- [1] 戴华东,胡谋法,卢焕章,等.基于连通域标记的目标检测算法设计与实现[J].现代电子技术,2015,38(20): 71-74.
- [2] Shieh Jenyu, Liao Yilin, Huang Weicheng, et al. Applying FPGA to implement real-time infrared tracking of moving objects [J]. Journal of Selected Areas in Microelectronics, 2012 (11): 12-18.
- [3] Oleksandr Kalentev, Abha Rai. Connected component labeling on a 2D grid using CUDA[J]. Journal Parallel Distribution, Computer, 2001(71): 615-620.
- [4] 谢宜壮,谭许彬,陈禾.一种新的连通域标记算法[J].北京理工大学学报,2012,32(12): 1273-1278.
- [5] Burger W, Burger M J. Digital image processing[M]. 北京:清华大学出版社,2010.
- [6] 张恒,倪永婧.面向目标特征提取的连通域标记算法[J].计算机与网络,2015,41(7): 58-61.
- [7] 杨帆,张皓,马新文,姜勇.基于 FPGA 的图像处理系统[J].华中科技大学学报,2015,43(2): 119-123.
- [8] He Lifeng, Chao Yuyan, Kenji Suzuki. An Algorithm for connected-component labeling, hole labeling and euler number computing[J]. Journal of Computer Science and Technology, 2013, 28(3): 468-478.

作者简介:

马毅超 男,(1984-),博士,讲师.研究方向为高速数字电路设计与 ASIC 设计.

吴龙胜(通讯作者) 男,(1968-),研究员.研究方向为大规模集成电路设计. E-mail: wls771@163.com.