

## 基于硬件加速的实时二值图像连通域标记算法

赵菲\* 张路 张志勇 卢焕章

(国防科技大学 ATR 国防科技重点实验室 长沙 410073)

**摘要:** 针对光学成像制导武器系统对图像处理的实时性要求, 该文提出了一种基于硬件加速的 2 次扫描连通域标记算法。算法结合基于像素和基于游程扫描算法的优点, 以像素为基本的扫描单元, 以线段为基本的标号单元, 在第 1 次扫描过程中建立临时标号的树形拓扑结构, 并输出线段作为结果。第 2 次扫描对线段进行标号替换完成连通域标记。通过在 FPGA+DSP 平台中进行实验证明, 该文算法的硬件加速实现占用资源少, 能够达到较高的性能和执行效率, 保证了系统的实时性, 具有较高的实用价值。

**关键词:** 图像处理; 连通域; 实时性; 三层树; 硬件加速

中图分类号: TN911.73

文献标识码: A

文章编号: 1009-5896(2011)05-1069-07

DOI: 10.3724/SP.J.1146.2010.00793

## A Hardware Acceleration Based Algorithm for Real-time Binary Image Connected-component Labeling

Zhao Fei Zhang Lu Zhang Zhi-yong Lu Huan-zhang

(National Key Laboratory of Automatic Target Recognition (ATR),

National University of Defense Technology, Changsha 410073, China)

**Abstract:** Due to the requirement for real-time image processing in optical imaging homing weapon systems, this paper proposes a hardware acceleration based connected-component labeling algorithm, which is a real time and two-pass algorithm. The algorithm integrates the merit of pixel-based and the run-based algorithm, which sets pixel as scan unit and line as label unit. Tree-shape topology is constructed in the first scan process, and lines are exported as the results. Then the labels are replaced in the second scan process to complete the connected-component labeling. Experiments on DSP+FPGA platform demonstrate that the hardware acceleration implementation of algorithm reaches a higher performance and efficiency with less resources consumption, and meet the demand of real-time processing.

**Key words:** Image processing; Connected-component; Real-time; Three-layer tree; Hardware acceleration

### 1 引言

在光学成像制导武器系统中, 连通域标记作为底层最重要的处理任务之一为目标检测、跟踪和识别提取图像中的不同区域。由于系统的帧频较高(大于等于60 Hz), 从传感器采集并传输一帧图像给处理单元到处理单元完成对这帧图像所有的处理(目标检测、跟踪、识别等)一般只有几毫秒到十几毫秒, 能留给底层图像处理的时间很短。因此, 快速的连通域标记算法对提高系统性能有着重要的意义。

目前已有的快速连通域标记算法大体可分为两类: (1)基于等价标号的方法, 这类方法至少需要对图像进行两次扫描(按照从左到右, 从上到下的光栅扫描顺序), 该类算法需要记录并整理临时标号之间

的等价关系。根据扫描单元的不同, 该类方法又可分为基于游程扫描的方法<sup>[1,2]</sup>、基于像素扫描的方法<sup>[3-5]</sup>和基于块扫描<sup>[6]</sup>的方法。对此类算法的改进主要围绕提高标号等价关系处理及图像访问的效率方面。文献[7]提出了一次扫描的连通域分析方法, 其本质依然是基于图像等价标号的方法。该方法未标记图像中的连通区域, 而是提取出每个连通区域的特定信息(如像素灰度和、像素坐标和、最小包围框等), 在需要标识完整连通区域的场合中并不适用。(2)基于区域生长的方法。这类方法不记录和整理临时标号之间的等价关系, 只需一次扫描即可完成连通域标记。其中文献[8]提出一种基于轮廓追踪的方法, 能够达到较快的处理速度。基于区域生长的方法对于图像的扫描不具有规律性, 所以很难以并行或是硬件加速方式实现。

上述大部分连通域标记算法都是基于通用处理

2010-07-28收到, 2010-11-11改回

国家863计划重大项目(2009AA8100701)资助课题

\*通信作者: 赵菲 f\_z2010@126.com

器(GPP)平台(如PC机)进行设计或改进,其中文献[1]和文献[8]提出的算法在高性能PC机上对 $512 \times 512$ 大小的多幅二值图像进行连通域标记,分别只需 $0.8 \sim 1.4 \text{ ms}$ 和 $1.8 \sim 3.9 \text{ ms}$ <sup>[1]</sup>。但对光学成像精确制导武器系统而言,功耗和体积等因素的限制使得系统中处理器的能力与高性能PC相差甚远,算法无法达到上述处理速度,满足不了实时性的需求。因此,研究能够硬件加速并实时实现的算法成为解决实时性问题的有效途径。国外的学者对于连通域标记算法的FPGA硬件加速作了较多的研究,如文献[7,9-12]等都提出了对已有算法或是新算法的FPGA硬件加速设计,但都存在着一些不足或局限。如文献[9]中的算法及其FPGA设计对图像的扫描次数不固定,无法满足实时处理的需求。文献[10]对传统的二次扫描方法进行FPGA硬件加速,但其需要较大的存储资源缓冲中间图像而且需要两倍的像素时钟处理等价关系<sup>[7]</sup>。文献[7]和文献[12]中提出的基于FPGA硬件加速的一次扫描连通域分析算法必须利用图像扫描过程中的行消隐时间处理临时标号的等价关系,对于不存在行消隐的图像扫描方式将无法适用。文献[11]提出一种基于游程的连通域标记算法及其FPGA实现方式,但由于其算法本身结构串行性较强,因此并不适合在图像传输过程中实时处理,而且性能也有待提高。

总的来说,大多数基于GPP平台的算法不具有较好的并行性,步骤之间有较强的耦合,数据结构复杂,不适合硬件加速实现。本文提出了一种基于硬件加速的二值图像连通域标记算法,该算法是一种基于等价标号的二次扫描连通域标记算法。在第一次扫描中,建立树形拓扑结构整理临时标号的等价关系,并将结果以线段形式输出。第二次扫描对输出线段进行标号替换。在DSP+FPGA的实时处理平台中,算法的第一次扫描经过FPGA进行硬件加速,可在图像传输的过程中实时对图像进行处理;第二次扫描在图像传输完成后利用DSP进行标号替换,也可以结合其它复杂处理(如目标识别等)同时进行。

## 2 算法原理

传统基于等价标号的连通域标记算法以像素为基本的扫描和标号单元,对当前扫描像素 $P(y, x)$ ( $y$ 代表行,  $x$ 代表列)进行临时标号赋值时,一般以4连通(或8连通)的方式判断其相邻像素临时标号之间的关系。因此,在光栅扫描顺序的方式(扫描顺序如图1中虚线所示)下,将会产生冗余临时标号和等价关系,如图1所示,其中数字代表分配给该像素

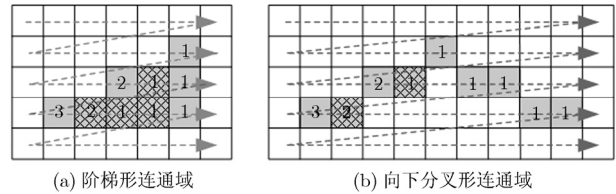


图1 产生冗余临时标号和等价关系示意图

的临时标号。从图中可以看出原本属于一个区域的像素被分配了几个临时标号,并且记录了等价关系,图中阴影部分代表扫描过程中出现等价临时标号的位置。

以游程(线段)作为扫描单元能够避免如上所述的问题,但需要预先提取当前行中的游程,并在游程的终点处通过判断当前游程与前一行中所有游程的重叠关系对当前游程赋以临时标号;同时对可能存在的临时标号等价关系(即当前游程可能与前一行中多个不同临时标号的游程重叠)进行整理<sup>[1,2,11]</sup>。这意味着在检测到游程终点之前,算法的计算量很小,大部分的运算集中在了游程的末尾处,这将成为算法硬件加速时的性能瓶颈。

本文提出的算法结合基于像素和基于游程扫描算法的优点,以像素为基本的扫描单元,以线段为基本的标号单元,发现等价的同时,构建树形拓扑结构整理临时标号的等价关系。这样做既避免图1中所出现的问题,相对基于游程的方法又降低了单点的运算复杂度。为了便于描述,下文以4连通方式为例,叙述本文算法的原理。

算法第1次扫描过程的实施可以分为两个并行的方面:(1)连通判断与结果输出:判断当前像素 $P(y, x)$ 是否与前一行中可能存在的线段有重叠关系,并在 $P(y, x)$ 所在线段 $\text{Line}_p$ 的终点处以线段的方式输出连通域标记的结果。(2)树形拓扑结构构建:建立并合并不同层次的树结构,在3层树的数据结构中记录临时标号到最终标号的索引关系。

由于对当前像素 $P(y, x)$ 的操作中需用到前一行的信息,算法中使用 $\text{info}(x) = \{\text{line\_start}, g, \text{label}\}$ ,  $x$ 代表列的结构保存前一行中每个像素的信息,其中 $\text{line\_start}$ 代表像素 $P(y-1, x)$ 是否是线段起点,  $g$ 代表 $P(y-1, x)$ 的像素值,  $\text{label}$ 代表 $P(y-1, x)$ 所在线段的临时标号。当前行中的线段 $\text{Line}_p$ 将作为连通域标记的结果输出,结构为 $\{y, \text{sp\_}x, \text{ep\_}x, p\_label\}$ ;其中 $y$ 代表当前行号,  $\text{sp\_}x$ 与 $\text{ep\_}x$ 是 $\text{Line}_p$ 的起点列号和终点列号,  $p\_label$ 是 $\text{Line}_p$ 的临时标号。

### 2.1 连通判断与结果输出

在图像扫描的过程中,若当前行与前一行有线

段重叠，则当前行中至少存在一个点与前一行中的线段有重叠关系，如图 2 中的黑色区域所示。

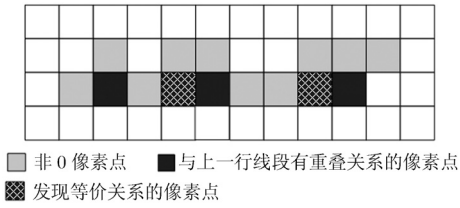


图 2 行间线段的重叠关系示意图

$P(y, x)$  与当前行中线段的关系可以根据扫描点  $P(y, x)$  和  $P(y, x - 1)$  的值确定。根据  $P(y, x)$  与线段的不同关系，连通判断和结果输出的流程如图 3 所示。其中 new\_label 是新分配的临时标号。

### 2.2 构建树形拓扑结构

当  $P(y, x) \neq 0$  并且  $\text{info}(x) \cdot \text{line\_start} = 1$  时，如果  $\text{overlap} = 1$ ，则说明在  $P(y, x)$  处发现了临时标号 last\_label 与  $\text{info}(x) \cdot \text{label}$  的等价关系，如图 2 中阴影部分所示位置。由于以线段作为基本的标号单元，因此只有在图像中存在向上分叉生长的连通区域时才可能出现临时标号的等价情况，并且每次等价情况只会发生两个临时标号之间。

算法通过构建树形拓扑结构整理临时标号之间的等价关系，树的节点为具有等价关系的临时标号，树根为对应的最终标号，该最终标号可以是所有等价临时标号中的最大值或最小值(下文采取最大值)。算法将临时标号对应的等价情况分为为 1 层树结构，2 层树结构和多层树结构，总体结构上采用 3

层树的数据结构存储图像中所有临时标号与最终标号的对应关系，固定了临时标号到最终标号的最大索引长度，如图 4 所示。从图中结构可以看出，从临时标号到最终标号最多需要 3 次索引，最少只需要 1 次。如图中  $L_n$  是一棵 1 层树结构，经第 1 层索引后索引表 1(index\_list1)中的值为 0，代表图像中临时标号为  $L_n$  的区域自身构成一个完整连通域，且根号为  $L_n$ ； $\{L5, L6, L7\}$  构成一棵 2 层树结构，经第 1 层索引后索引表 1(index\_list1)中的值为 index13，经第 2 层索引后索引表 2(index\_list2)中的值为  $L7$ ，这代表图像中临时标号为  $L5, L6, L7$  的区域相互连通并构成一个完整连通域，且其根号为  $L7$ ； $\{L8, L9, L10, L11\}$  构成一棵 3 层树结构，经第 1 层索引后索引表 1(index\_list1)中的值为 index14 和 index15，经第 2 层索引后索引表 2(index\_list2)中的值为 index22，经第 3 层索引后 3 层树树根(root\_list)中的值为  $L11$ ，这代表图像中临时标号为  $L8, L9, L10, L11$  的区域相互连通并构成一个完整连通域，且其根号为  $L11$ ； $\{L1, L2, L3, L4\}$  构成的 3 层树的索引情况类似。对于图像中可能存在的大于 3 层的结构，算法中直接对树根进行操作以保证索引结果的正确性。如图 4 所示，如果  $\{L1, L2, L3, L4\}$  和  $\{L8, L9, L10, L11\}$  分别已建立 3 层树后，假设出现临时标号  $L1$  与  $L8$  等价，标志着两棵树中所有的临时标号代表的区域都相互连通，根据树根的选取临时标号中最大值的原则，将  $\{L1, L2, L3, L4\}$  所对应的 3 层树树根由  $L4$  修改为  $L11$ 。

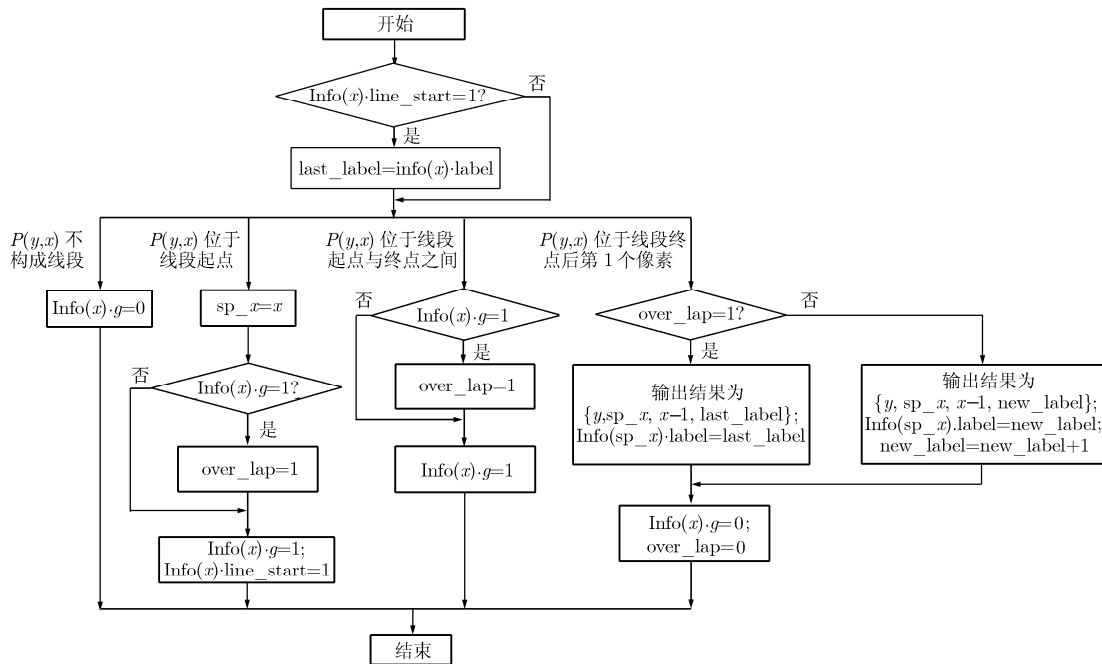


图 3 连通判断与结果输出流程图

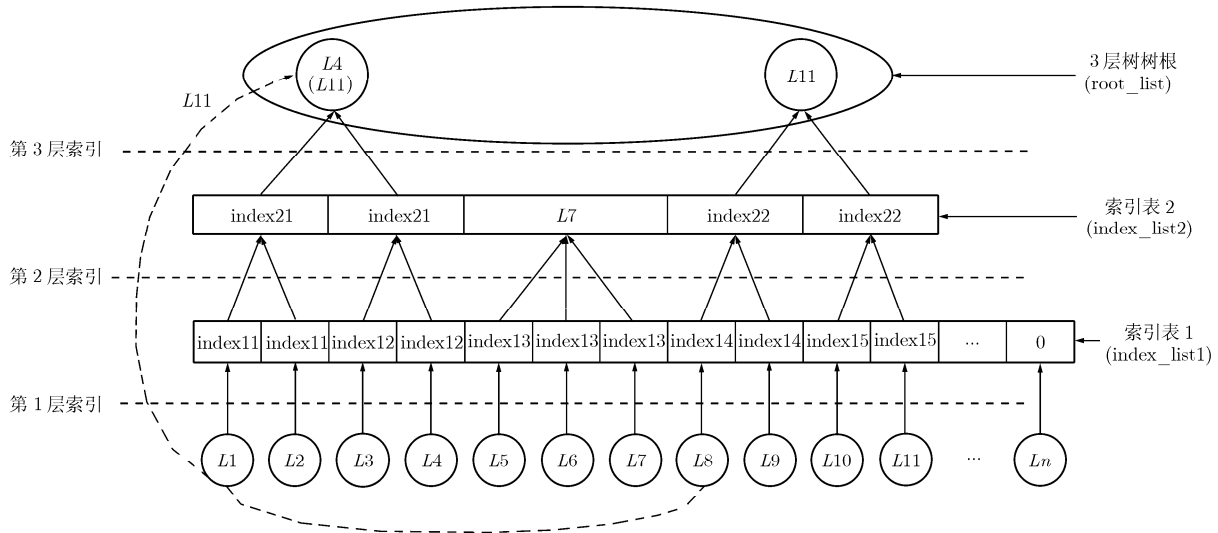


图 4 等价临时标号的树形结构图

从图 4 中可以看出，3 层树的结构主要由索引表 1(index\_list1)、索引表 2(index\_list2)和 3 层树树根(root\_list)3 个数组组成，初始值均为 0；index\_list1 的长度为扫描过程中分配的临时标号的个数，假设图像的大小为  $2^M \times 2^N$ ，那么以 4 连通方式进行连通判断最多能分配  $2^{M+N-1}$  个临时标号，以 8 连通方式进行连通判断最多能分配  $2^{M+N-2}$  个临时标号。极限情况下，以每两个临时标号等价构成一棵 2 层树，每 4 个临时标号等价构成一棵 3 层树，index\_list2 的长度最大能达到  $2^{M+N-2}$  (4 连通方式) 或  $2^{M+N-3}$  (8 连通方式)，root\_list 的长度最大能达到  $2^{M+N-3}$  (4 连通方式) 或  $2^{M+N-4}$  (8 连通方式)。在实际应用中，需要的长度将远小于以上各数组的极限长度。对于 index\_list2 中既可能存在 2 层树的根 (root\_l2)，又可能存在索引号(index2)，算法在存储的过程中将使用标志位加以区别。

两个临时标号等价关系的整理可归纳为是在图 4 所示的 3 层树结构中修改每一层的索引情况，最终合并成新的树结构。对等价关系的整理原则保证了 3 层树结构中树的总数目即是图像中完整连通域的个数，容易在等价关系的处理过程中统计得到。第 2 次扫描仅需对线段的临时标号在 3 层树结构中进行索引就可完成标号替换。

### 3 硬件加速实现

本文设计以 Xilinx 公司的 VII 系列 FPGA XC2V3000-6FG676 为平台,对第 1 次扫描硬件加速的结构可分为 3 部分:行缓冲、连通判断与结果输出、等价关系整理,如图 5 所示。

从对算法原理的叙述中可以看出,算法在扫描

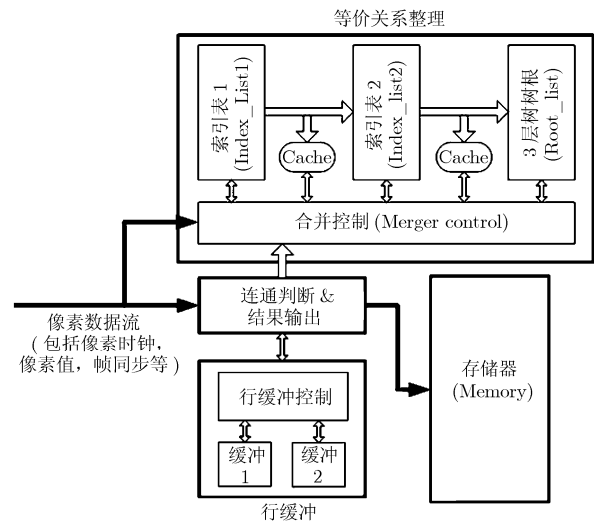


图 5 硬件加速设计结构图

像素的过程中进行连通判断与结果输出和进行等价关系整理之间不具有耦合关系，完全可以分为两个并行的进程同时执行。等价关系整理进程中具有很好的流水线结构，在每一层索引基础上对树的合并操作都可以构成 1 级流水操作，因此等价关系整理进程完全可以由 3 级流水线完成。3 层树结构中的索引表 1(index\_list1)、索引表 2(index\_list2)可以使用 FPGA 中的双口 BlockRAM 实现，考虑到 RAM 的最短存取间隔，设计一级 Cache 来保证对 RAM 同一地址重复操作的正确性。3 层树树根 (root\_list)采用寄存器实现。前一行中像素信息 info(x)使用双口 BlockRAM 进行缓冲，即行缓冲。考虑到对行缓冲内容的实时更新和线段终点处对线段起点位置像素信息的更新会产生冲突，设计使用“乒乓式”双 RAM 来实现行缓冲功能。

图 5 中的存储器资源用来保存连通域标记的实时输出结果, 由于不需要进行频繁的存取操作, 可以使用片外 RAM 资源。由于本文算法以线段的方式输出结果, 因此相对于保存图像中每个像素临时标号的输出方法可以减少对存储资源的需求。在极限情况下, 大小为  $2^M \times 2^N$  的图像会产生  $2^{M+N-2}$  (8 连通方式) 个临时标号, 每个标号需要用  $(M+N-2)$  bit 存储, 每个像素保存结果和以线段保存结果所需存储资源的比值为

$$\frac{(M+N-2) \times 2^{M+N}}{(M+N+N+M+N-2) \times 2^{M+N-2}} = \frac{4M+4N-8}{2M+3N-2}$$

而当二值图像中具有较大面积的连通区域时, 本文的结果保存方式将占用更少的存储资源。对索引表需要占用的最大存储资源已在前文作过分析, 对于具体应用中有实际意义的图像而言, 需要的存储量远远小于该值, 应根据具体情况合理定制索引表占用的 BlockRAM 资源。

假设针对 CCD 大小为  $256 \times 256$  的光学成像传感器的输出图像进行连通域标记, 设计容纳 256 个临时标号, 图像传输以常见的光栅扫描顺序逐个像素传输, 每个像素在传输过程中已由前端模块实时完成二值化, 传输信号有像素时钟, 二值图像数据 (1 bit, 0 或 1), 图像帧同步, 图像传输过程中没有行消隐周期。对算法第 1 次扫描过程的硬件加速只需使用单倍像素时钟就可以在每个像素传输的同时实时完成相应的处理, 在 XC2V3000-6FG676 FPGA 上完成布局布线后像素时钟可达 110 MHz 以上, 对 FPGA 内部资源的占用如表 1 所示。从表中可以看出, 算法第一次扫描的硬件加速实现占 FPGA 内部资源较少。

表 1 硬件加速的 FPGA 资源占用情况

资源名称	占用数目	占用百分比(%)
Flip Flops	385	1
Slices	482	3
BRAMs	4	4
4 input LUTs	888	3

需要说明的是, 算法的硬件加速设计了 256 个临时标号, 这对于相对简单和干净的二值图像而言已经足够, 但其处理能力并不局限于此。对于可能产生更多临时标号的复杂二值图像而言, 硬件加速设计时需要增加 BlockRAM 和寄存器资源用于 3 层树数据结构的存储。

如果仅考虑完成连通域标记, 本文算法的第 2

次扫描可以像前文提到的文献[9-11]一样, 在图像传输完成(即第一次扫描完成)后使用 FPGA 硬件加速完成标号替换, 由于标号替换过程较为简单, 对其硬件加速实现不再详述。除在前文中提到的不足外, 文献[9-11]中对于算法的硬件加速均针对传输完成后存储好的图像进行处理, 而本文算法第一次扫描的硬件加速可以在图像传输的过程中实时完成。其中文献[11]使用 Handel-C 将算法从高级语言转变为硬件描述语言实现硬件加速, 并且达到了相对较快的处理速度。其在 Xilinx 公司 XC4VLX160-10FF1148 型 FPGA 上布局布线后的处理时钟速率可达 49.73 MHz, 对于已存储好的  $2^M \times 2^N$  大小的图像最短处理时间为<sup>[11]</sup>  $(2^{M+N} + \text{游程数}) \times 2 \times (1/(49.73 \times 10^6))$ ; 如果与文献[11]相同对连通域中每个像素进行标号替换, 本文算法的第 2 次扫描在相同 FPGA 上的硬件加速设计在布局布线后可达到 102 MHz 的时钟速率, 完成图像标号替换的最短时间为  $2^{M+N} \times (1/(102 \times 10^6))$ , 小于文献[11]的四分之一。由于文献[11]中算法本身被分为较多具有耦合关系的串行步骤, 并且第一次扫描中的大部分处理集中在游程的末尾, 所以整体上并未达到十分理想的处理性能, 同时也不能利用图像传输的过程进行实时处理。相比之下, 文献[7,12]中对算法的硬件加速虽然可以在图像像素传输的同时实时处理, 并能够提取连通域的特定信息, 但并不能完成连通域标记, 而且必须依赖行消隐周期才能处理等价关系, 具有一定的限制。

考虑到弹载平台的实际应用中连通域标记并不是最终目的, 线段标号替换可以结合形状分析、目标识别等同时进行, 以获得更高的处理效率, 而这类复杂操作不适合硬件加速, 因此本文面向弹载图像处理的实际应用, 在下文的实验中将采用处理器平台完成第 2 次扫描的标号替换操作。

## 4 实验结果与分析

为验证并对比本文算法和基于处理器平台算法的处理效率, 分别设计在 PC 机(Pentium Dual-Core 2.5 GHz CPU, 2 G 内存, WinXp 操作系统)和 DSP (TI: TMS320C6713, 200 MHz)+FPGA(Xilinx: XC2V3000-6FG676)的实时处理平台上对本文算法和基于 PC 平台具有较快处理速度的文献[8]中的算法<sup>[13]</sup>进行了比较。实际弹载光学传感器对成像距离的要求在实验室环境内较难满足, 难以拍摄到较高复杂度的测试图像。为不失一般性, 实验图像部分来源于实验室条件下传感器对准平行光管所采集的靶标图像, 其它图像来源于东京大学开发的

Standard Image Database (SIDBA)中的 Misc 图像集<sup>[4]</sup>,所有实验图像大小均为  $256 \times 256$ 。图中所示为二值化的实验图像,图 6(a)–6(f)来源于 SIDBA 的 Misc 图像集,图 6(g),6(h)为实际采集的图像。其中图 6(a)–6(d)的连通性复杂程度较高,图 6(e)–6(h)的连通性复杂程度较低。在 DSP+FPGA 的实时处理平台进行实验时,图像预先存入专用的测试设备,实验时测试设备读取图像数据,然后通过与光学传感器相同的数据传输接口将图像像素数据以光栅扫描的顺序依次传输给实时处理平台,再由处理平台进行实时处理。

对于图 6 中的图像进行实验后的数据如表 2 所示。其中 PC 机耗时是指将本文算法与文献[8]算法使用 VC++6.0 编程实现后的处理用时,从表中可

以看出,在 PC 机上文献[8]算法的处理速度是本文算法处理速度的 2 倍以上。在实时处理平台上,由于本文算法的第 1 次扫描过程可在图像传输的过程中通过 FPGA 硬件加速实时完成,耗时仅在 DSP 实现第 2 次扫描过程中产生。而文献[8]的算法必须等待图像传输完成后,由 DSP 进行相应的算法处理,从表中可以看出,在实时处理平台上,本文算法的处理速度是文献[8]算法处理速度的 40 倍以上。

综合考虑第 1 次扫描的硬件加速后可达到的时钟频率和在 DSP 上的耗时,如果以 100 Hz 的帧频(10 ms/帧),30 MHz 的时钟速率传输复杂度如图 6 中的图像数据,本文算法图像传输时间( $1/(3 \times 10^7) \times 256 \times 256 \approx 2.2$  ms)加 DSP 处理耗时最多不到 3.5 ms,在下一帧图像来临之前能留给后续处理的时间

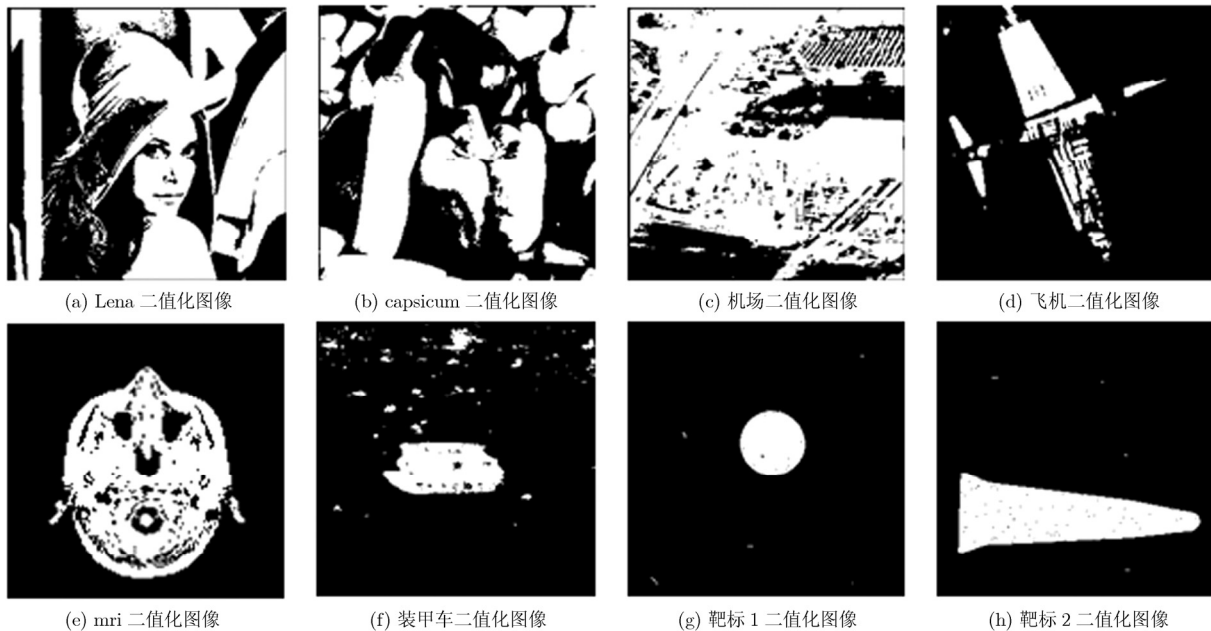


图 6 二值化后的实验图像

表 2 本文算法与文献[8]算法的处理结果

图像	总连通域 (树)数目	本文算法				文献[8]算法			
		1层树数目	2层树 数目	3层树 数目	线段数	PC 机 耗时(ms)	实时处理平 台耗时(ms)	PC 机耗 时(ms)	实时处理平 台耗时(ms)
图 6(a)	223	212	6	5	2180	1.1	0.83	0.45	34.5
图 6(b)	86	77	6	3	1465	1.08	0.6	0.4	30.8
图 6(c)	100	92	7	1	2472	1.15	1.0	0.55	40
图 6(d)	39	30	8	1	736	1.03	0.30	0.34	28.3
图 6(e)	32	31	1	0	1147	1.05	0.41	0.39	28.9
图 6(f)	108	102	6	0	411	1.03	0.13	0.33	23.1
图 6(g)	6	5	1	0	83	0.97	0.07	0.3	21.4
图 6(h)	5	4	1	0	154	1.05	0.09	0.32	22

有 6.5 ms, 完全可以满足实时图像处理的要求。如果考虑更高的时钟传输速率(只要在 110 MHz 以下), 本文算法经硬件加速后可以达到对  $512 \times 512$  等尺寸图像的实时处理能力。

## 5 总结

本文提出了一种基于硬件加速的实时二值图像连通域标记算法。算法的设计充分考虑了并行性和流水性, 可以在扫描图像像素的同时, 构建树形拓扑结构整理临时标号间的等价关系, 同时以线段方式输出标记结果。在图像传输完成后, 第 2 次扫描只需对第 1 次扫描的输出线段进行标号替换。算法中第 1 次扫描的硬件加速可以在图像传输过程中完成, 提高了执行效率, 可以满足光学成像精确制导武器系统对图像处理的实时性要求, 具有较高的实用价值。

由于算法的执行过程中具有较多的存储器访问等操作, 因此算法在 GPP 平台上的执行效率并不优于已有的优秀算法。在 PC 机等具有较高处理能力的 GPP 平台上, 文献[1,3,6,8]的算法相对占用资源少, 并且具有较高的处理效率, 是在 GPP 平台上进行实时连通域标记的较好选择。

## 参考文献

- [1] He Li-feng, Chao Yu-yan, and Susuki Ken-ji. A run-based two-scan labeling algorithm[J]. *IEEE Transaction on Image Processing*, 2008, 17(5): 749-756.
  - [2] Wang Hong-tao, Luo Chang-zhou, and Wang Yu, *et al.* New algorithm for binary connected-component labeling based on run-length encoding and union-find sets[J]. *Journal of Beijing Institute of Technology*, 2010, 19(1): 71-75.
  - [3] He Li-feng, Chao Yu-yan, and Susuki Ken-ji. An efficient first-scan method for label-equivalence-based labeling algorithms[J]. *Pattern Recognition Letters*, 2010, 31(9): 28-35.
  - [4] He Li-feng, Chao Yu-yan, and Susuki Ken-ji. Fast connected-component labeling[J]. *Pattern Recognition*, 2009, 9(42): 1977-1987.
  - [5] 张修军, 郭霞, 金心宇. 带标记矫正的二值图像连通域像素标记算法[J]. *中国图像图形学报*, 2003, 8(2): 198-202.  
Zhang Xiu-jun, Guo Xia, and Jin Xin-yu. The pixel labeled algorithm with label rectified of connecting area in binary pictures[J]. *Journal of Image and Graphics*, 2003, 8(2): 198-202.
  - [6] Costantino Grana, Daniele Borghesani, and Rita Cucchiara. Fast block based connected component labeling[C]. 2009 IEEE International Conference on Image Processing (ICIP2009), Cairo, Egypt, Nov. 2009: 4061-4064.
  - [7] Bailey D G and Johnston C T. Single pass connected components analysis[C]. *Proceedings of Image and Vision Computing*, Hamilton, New Zealand, Dec. 2007: 282-287.
  - [8] Chang Fu, Chen Chun-jen, and Lu Chi-jen. A linear-time component-labeling algorithm using contour tracing technique[J]. *Computer Visison and Image Understanding*, 2004, 93(2): 206-220.
  - [9] Crookes D and Benkrid K. An FPGA implementation of image component labelling[C]. *SPIE Configurable Computing Proceedings:Technology and Applications*, USA, Sep. 1999: 17-23.
  - [10] Jablonski M and Gorgon M. Handel-C implementation of classical component labelling algorithm[C]. *Euromicro Symposium on Digital System Design (DSD 2004)*, Rennes, France, Mar. 2004: 387-393.
  - [11] Appiah K, Hunter A, and Dickinson P, *et al.* A run-length based connected component algorithm for FPGA implementation[C]. *International conference on Field-Programmable Technology*, Taipei, China, Dec. 2008: 177-184.
  - [12] Johnston C T and Bailey D G. FPGA implementation of a single pass connected components algorithm[C]. *4th IEEE International Symposium on Electronic Design, Test and Applications (DELTA 2008)*, Hong Kong, China, Jan. 2008: 228-231.
  - [13] Fu Chang. <http://ocrlnx03.iis.sinica.edu.tw/dar/Download%20area/ccl.php3>, 2010.06.
  - [14] University of Tokyo. SIDBA:<http://sampl.ece.ohio-state.edu/data/stills/sidba/index.htm>, 2010.6.
- 赵菲: 男, 1983年生, 博士生, 研究方向为实时系统、图像处理技术。  
张路: 男, 1980年生, 博士生, 研究方向为实时系统、图像处理技术。  
张志勇: 男, 1976年生, 讲师, 研究方向为红外图像处理技术。  
卢焕章: 男, 1963年生, 教授, 研究方向为自动目标识别技术及实时系统设计。