

Canny 算法的 GPU 并行加速

张帆^{1,2*}, 韩树奎³, 张立国¹, 王文胜^{1,2}

(1. 中国科学院 长春光学精密机械与物理研究所, 吉林 长春 130033;

2. 中国科学院大学, 北京 100049;

3. 东北电力设计研究院, 吉林 长春 130021)

摘要: Canny 算法在 PC 机上的执行速度较慢, 这极大地限制了其实用性。本文在前人的研究基础上对算法进行更深的优化和改进。首先在 VS2012 开发环境下利用数字图像处理技术对原算法进行原理上的改进, 再利用 GPU 流处理器数量众多的优势以及强大的多线程并发执行能力对 Canny 算法进行并行加速。在 500 pixel × 500 pixel 的图片上, 对本文算法和原 Canny 算法进行了实验验证。实验结果表明, 在 4 096 pixel × 4 096 pixel 大小的图片上采用本文的 GPU 移植算法处理后, 执行速度从 80 ms 降到了 6 ms 以内。在不影响边缘检测效果的前提下极大地提高了算法的实用性。

关键词: 边缘检测; GPU; 并行处理; 连通域提取

中图分类号: TP751.1 文献标识码: A doi: 10.3788/CO.20171006.0737

Parallel acceleration of Canny algorithm based on GPU

ZHANG Fan^{1,2}, HAN Shu-kui³, ZHANG Li-guo^{1*}, WANG Wen-sheng^{1,2}

(1. *Changchun Institute of Optics, Fine Mechanics and Physic, Chinese Academy of Sciences, Changchun 130033, China;*

2. *University of Chinese Academy of Science, Beijing 100049, China;*

3. *Northeast Electrical Power Design Institute, Changchun 130021, China)*

* *Corresponding author, E-mail: zhangliguo@ciomp.ac.cn*

Abstract: Due to the slow execution speed of Canny algorithm in PC, the practicality of this algorithm is greatly restricted. Based on the previous studies, we further optimizes and improves the algorithm. First of all, we use the digital image processing technology to improve the original algorithm under the development environment of VS2012, and then accelerate the Canny algorithm by taking advantage of the large number of GPU stream processors and powerful multithreaded concurrent execution capability. Experiments were made on the improved algorithm and the original Canny algorithm. Experimental results show that in the 4 096 × 4 096 pixel-size images, the GPU migration algorithm presented in this paper can reduce the execution speed from 80 ms to less than 6 ms. Through this improvement, it can greatly improve the practicability of the algorithm without affecting the edge detection effect.

Key words: edge detection; GPU; parallel processing; connected component extraction

1 引言

边缘检测是图像处理领域的一种很重要算法,其能从原始图像中分离出目标的边缘信息,去除冗余信息。边缘检测效果比较好的算法为 Canny 算法。自 1986 年 John Canny 提出该算法以来,其高效性得到学者们的广泛认可^[1]。该算法在总结前人算法的基础上,提出了新的边缘检测原则:(1)能正确标识出图像的边缘;(2)标出的边缘要和图像的真实边缘尽可能接近;(3)图像中的边缘在结果图中只被标识出一次,因此,可有效去除虚假边缘信息。为了达到上述效果,Canny 算法采取了高斯滤波、非极大值抑制、边缘连接等方法进行处理。检测高效性是其重要优点,但是由于算法复杂导致其执行速度缓慢,从而极大制约了其实用性。已有的 C 版本,OpenCV 库函数版本的执行效率都比较低,不能用于快速场景检测以及工程项目开发中。

近年来,随着高性能计算设备的发展,以及 MPI、OpenGL 等一系列并行加速工具的出现使得算法的执行速度有了很大提高。特别是近年来 GPU 显卡编程为并行计算提供了新的途径。GPU 编程以高速的执行速度、良好的可编程性、较高的功耗比成为了并行编程的一个热门工具^[2]。目前,对于 GPU 版本的 Canny 算法,国内外许多文献的关注点不一样。文献[3]直接将原算法移植到了 GPU 上,使其执行效率受到很大的制约,有很大的提高空间;文献[4]利用 FPGA 系统对算法进行改进,而这种改进仅是将高斯滤波改成了中值滤波进行去噪。这种改进在某些程度上有可能降低算法的执行速率。文献[5]将改进重点放在了最后的边缘搜索部分,改进了搜索算法策略,但是单一的优化搜索策略对于算法性能的整体提升并不明显。

本文利用 GPU 加速工具对 Canny 算法的各个执行步骤都进行优化。其中,主要优化了高斯去噪求边缘步骤和边缘搜索链接步骤。利用 GPU 的大流量处理器和多线程并发执行的特点,对一幅 4 096 × 4 096 图像进行边缘检测,将边缘

算法的执行时间压缩到 6 ms 以内。

2 算法介绍和分析

John Canny 在前人的边缘检测标准上又加了一条判别依据,即图像中的边缘在结果图中应该只被标识一次,以去除虚假边缘信息。改进后的算法流程如下。

(1)高斯滤波:在检测之前,采用高斯滤波去噪以去除图像采集时的随机噪声;随后再分别对 x 和 y 方向做差分,求边缘信息。高斯卷积公式如式(1)所示, $*$ 表示卷积。

$$K = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x*x+y*y}{2\sigma*\sigma}} \quad (1)$$

由公式得出的离散化差分模板如式(2)所示。

$$S_x = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \quad S_y = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \quad (2)$$

利用式(2)的模板对图像逐像素卷积。

在实际图像处理中,高斯滤波经常采用的离散化卷积模板大小为 3 × 3 或 5 × 5,方差根据实际图像的噪声分布选取,由此可计算出模板中每个位置的值。

高斯滤波是一种线性平滑滤波,主要用于消除高斯噪声。在遥感图像数据采集过程中,空间相机的图像采集环境比较稳定,图像噪声主要是电子元器件造成的高斯噪声。因此可以采用高斯滤波进行初步的去噪处理。

(2)非极大值抑制:为了使标识的边缘和图像的真实边缘尽可能接近,算法引入了非极大值抑制方法。在粗线条的边缘信息中只保留梯度最大的边缘,从而使检测出的边缘尽可能地靠近真实边缘。非极大值抑制原理如图 1 所示,在所测像素 8 邻域内寻找差分值最大的可能点,再通过线性插值寻找梯度方向^[8]。

(3)双阈值边缘假设:Canny 在前人的基础上加入了双阈值边缘假设,算法设定了两个差分阈值 A、B(A > B)。边缘差分值大于阈值 A 的一定是所求的边缘,由此得出边缘图 A,阈值介于 A 和

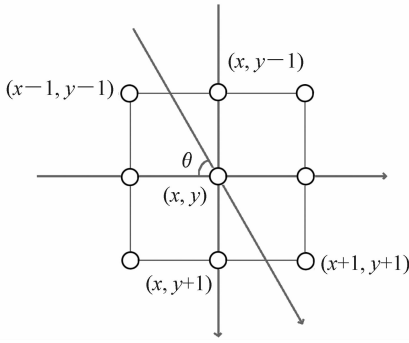


图1 非极大值抑制示意图

Fig. 1 Schematic diagram of non-maximum suppression

B之间的是可能有的边缘,由此得出边缘图B。最终在B图结果中寻找连接到A图的边缘区域,从而得到最终的边缘结果图。双阈值假设的原理如图2(彩色见期刊电子版)所示,其中黄色是图A中的边缘区域,绿色是图B中的边缘区域。算法以3为起点不断地在B图中搜索连接区

域,直到搜索完所有点为止。

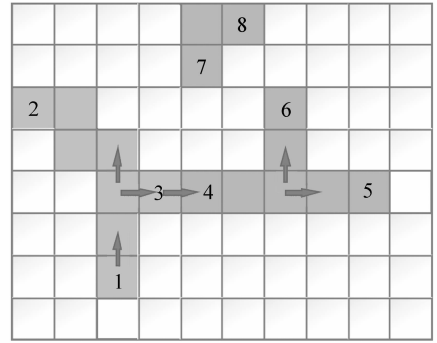


图2 双阈值假设搜索过程

Fig. 2 Searching process of dual threshold hypothesis

表1是对于一幅大小为4 096 × 4 096的8位灰度图执行Canny算法的耗时结果。基于Opencv2.4.8计算机视觉库在I5-2400,3.1 GHz主频,内存为16 GB的PC机上进行算法实现。

表1 C语言实现Canny算法各步骤执行时间

Tab. 1 Execution time of each step for Canny algorithm in C language

(ms)

	高斯滤波	非极大值抑制	边缘搜索	总计
4 096 × 4 096	57.4	5	17.6	82

从表1可以看出,在CPU的串行模式下,Canny算法耗时主要集中于第一步的高斯滤波求差分 and 最后一步的边缘搜索连接。这主要因为高斯模板对图像卷积计算在CPU单线程串行执行模式下的速度非常缓慢。对于在连通域搜索部分,算法的搜索策略是搜索已被标记点8邻域,判断是否存在有效点,这种单个点标记的搜索方法也是单线程逻辑搜索,耗时较多。而算法中的非极大值抑制部分耗时较少,优化起来相对简单。

3.1 算法优化

3.1.1 优化高斯滤波和差分

在图像求边缘差分算法中,基于Sobel算子的边缘检测方法既可以进行x和y方向的边缘检测,又可以对图像中的噪声起到一定的抑制作用^[12]。因此本文考虑利用Sobel算子代替Canny算法中的高斯滤波和差分求边缘两步,以降低计算复杂度。Sobel算子边缘检测的离散化模板和其卷积计算公式如下:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ 1 & -2 & 1 \end{bmatrix} \quad (3)$$

由式(3)可以看出,Sobel算子模板为3 × 3,其中有效计算的非零元素个数为6,在代码中单次循环的计算量为6;在原Canny算法流程中高斯滤波3 × 3模板的非零元素个数为9,差分的2 × 2模板的非零元素个数为4,因此在单个方向

3 算法改进与加速优化

依据以上分析,制约算法执行速度的两个主要原因是算法本身因素和串行执行模式^[10]。算法自身耗时主要集中在高斯卷积滤波和边缘搜索和连接上。串行执行模式主要是算法在PC机上基于CPU的单线程串行执行。因此本文将从这两个方面对Canny算法进行优化加速。

代码中的单次循环计算量总量是 13。由此可知, Sobel 算子的计算复杂度是原 Canny 算法的 1/2。对于 x 和 y 两个方向的图像求边缘差分计算, 可以通过 GPU 并行加速。图像处理的 GPU 并行加速原理如图 3 所示。其中, 每个像素点的计算过程可以映射为一个线程再赋予到一个流处理器上进行处理。这样可以利用 GPU 数量众多的流处理器, 实现大规模线程并行以压缩时间。

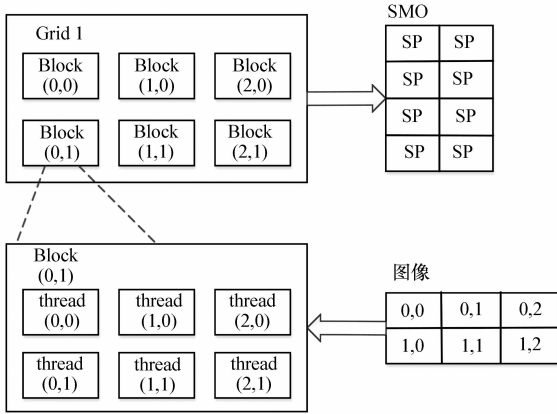


图 3 GPU 并行加速原理图

Fig. 3 Schematic diagram of parallel acceleration by GPU

3.1.2 GPU 加速非极大值抑制和双阈值图像分割

非极大值抑制和双阈值分割都是属于任务简单计算密集型的操作。它们可以采用如图 3 所示的 GPU 并行加速方法, 将每个像素点的计算映

射到一个线程, 并赋予一个流处理器 SP 进行处理。在非极大值抑制中, 为了简化计算, 不采用插值求 8 邻域最大值, 而直接采用 8 邻域求最大值的方法。

3.1.3 优化搜索和链接

本文将中间值边缘的搜索改成连通域搜索, 并且在搜索方式上采用 3 步进式搜索, 这样可以充分利用 3×3 卷积模板的信息。

原始 Canny 算法存在计算能力浪费的现象。每次进行连通区域扫描时都只是扫描了与边缘点连接的部分, 而对于可能的连通区域没有执行操作。算法中的边缘搜索与求连通域的操作步骤相似, 因此, 可以先将可能存在的连通域求出来, 再与真正的边缘点相连接。这样能够充分减少边缘搜索的时间。如图 4(a) 所示, 黄色是双阈值分割后 A 图中的连通区域, 绿色是 B 图中的连通区域。以 1、2、3 为边缘搜索 A 图中的连通域, 再以 4、5、6 和 7、8 搜索出 B 图中连通区域, 因为 B 图 4 点连接到 A 图中的连通域, 故将整个连通域判定为边缘。

在搜索策略上依据改进的 3 步进式搜索原理, 如图 4(b) 所示。1 点为起始搜索点传统方法是以 1 点为中心判断其 8 邻域的有效点, 如图 4(b) 右图。改进后以 1 点 8 邻域中的有效点 2 为中心, 判断其 8 邻域中除 1 点以外的有效点即 3 点, 如图 4(b) 左。3 步进式的搜索方法, 单次搜索可以标记两个以上的点, 效率较原算法中的 8 邻域单个点标记有了大大提高。

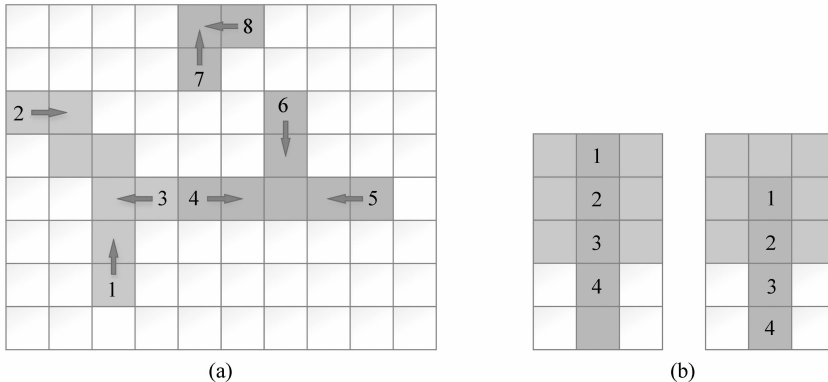


图 4 搜索方法优化示意图 4(a) (左), 4(b) (右)

Fig. 4 Schematic diagram of optimizing search and connection, 4(a) (left) and 4(b) (right)

3.2 双GPU优化

在 Canny 算法双阈值假设的两个阈值搜索连接过程中,图像 A 和图像 B 的连通域搜索连接过程是相互独立的。针对这种相互独立的分块任务可以利用双 GPU 进行加速,每个 GPU 分别负责一幅图像的搜索连接过程,原理如图 5 所示,即将两幅图像的边缘搜索连接任务分别放到两个 GPU 中进行。最终利用 GPU 的内部等待完成指令 *synthreads*,等两幅图像的搜索任务全部完成后,再进行有效区域边缘的连接与合并,这样可以隐藏计算用时较少的图像的处理时间。

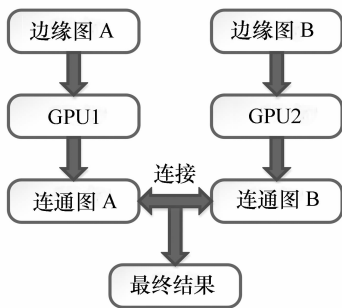


图 5 双 GPU 优化原理示意图

Fig. 5 Schematic diagram of dual-GPU optimization

此外,Canny 算法前面的从高斯滤波到非极大值抑制,也都可以利用双 GPU 进行优化加速。由于采用了两个相同的 GPU,则流处理器数量也将加倍,对于大规模的线程并行会带来线性的加速收益。类似步骤可以利用开发平台进行。

4 实验结果和讨论

本文采用 PC 平台,Opencv2. 4. 8 计算机视觉库以及 Nvidia 750Ti 显卡对本文算法进行了验证。对于 500 pixel × 500 pixel 大小的图片,边缘检测结果如图 6 所示,左图为 CPU 的边缘检测结果,右图为 GPU 的边缘检测结果。

可以看出,原始 Canny 算法和本文算法的差异很小,不影响目标的整体边缘轮廓^[11]。对于港口区域图像,能够看出图中的海上船只轮廓已经能够很好地与陆地分割开。在 GPU 的检测结果中(图 6(b))主要轮廓的边缘信息虽然有断点,但是能够反映出目标物的主要信息。上述实验说

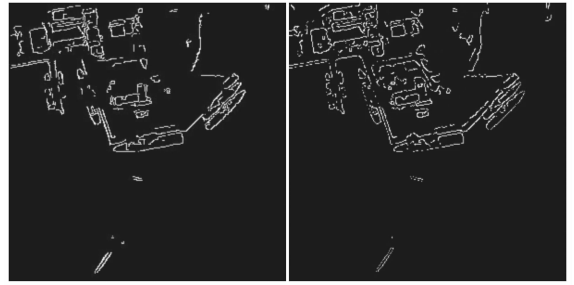


图 6 不同算法的比较结果(左图 CPU,右图 GPU)
Fig. 6 Comparison of two results by CPU (left) and GPU (right)

明本文算法的检测结果是可靠的。

针对不同版本的 Canny 检测算法的实验结果见图 7。图 7 是本文算法、OpenCV 库、GPU 库的实现检测结果,纵坐标是执行时间(单位:ms),横坐标是图像尺寸(单位:pixel²)。图 8 是本文算法、文献[3]算法和文献[7]用 FPGA 的执行结

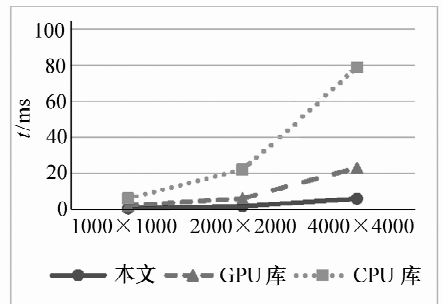


图 7 本文算法、OpenCV 库及 GPU 库算法的结果比较
Fig. 7 Results comparison of proposed method, OpenCV-lib and GPU-lib methods

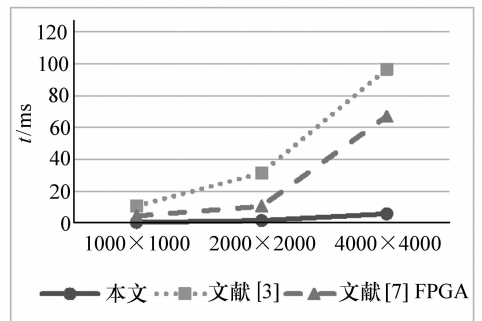


图 8 本文方法、文献[7]和文献[3]算法的结果比较
Fig. 8 Results comparison of proposed method, and methods from Ref. [7] and Ref. [3]

果。从图中可以看出,对于不同尺寸的图像,本文算法的执行速度较其他算法有明显提高^[13-14]。其中图 7 是本文算法的 GPU 并行移植与原算法 GPU 并行化移植的结果对比。图 8 为用 GPU 加速与其他硬件加速后的效果对比。

由图 7 和图 8 可以看出,在数据量较大的图像中,GPU 的加速效果比较明显。表 2 是本文使

用双 GPU 对于 Canny 算法的执行过程进行优化后的结果,实验采取 $4\ 096 \times 4\ 096$ 的大尺寸图像。实验数据表明,双 GPU 加速会带来线性的加速收益,这主要是因为更多的 GPU 芯片拥有更多的流处理器,可以同时并行处理更多的像素。因此,在实际工作以及硬件开发中,可以选择更多的 GPU 处理芯片群组进行更效率的程序开发。

表 2 对于 $4\ 096 \times 4\ 096$ 大小的图片双 GPU 优化本文改进算法执行时间

Tab. 2 Execution time by proposed method based on dual-GPU optimization on

	image with size of $4\ 096 \times 4\ 096$				(ms)
	高斯滤波	非极大值抑制	边缘搜索	总计	
单 980	1.52	2.15	2.3	5.9	
双 980	0.7	1.1	1.5	3.3	

从上面的实验结果可以看出,GPU 对于模板卷积这种任务简单,计算密集的操作可实现非常好的加速压缩,而对于边缘搜索连接这种任务复杂的操作加速效果不是很好。

前三步操作的速度随着 GPU 处理器能力的提高以及流处理器数量的增多而加速,当显卡从 750Ti 换成 980 以后,前面三步的计算速度有了很大的提高,说明前三步操作受流处理器数量的影响较大。而后半部分速度提升较小,主要因为其受限于算法本身。处理器换成双 GPU 以后,处理 $4\ 096 \times 4\ 096$ 大小的图像的执行时间从 19 ms 压缩到 5.9 ms,速度提升了 3.3 倍。

在遵从于原算法的 GPU 并行化移植后,Canny 边缘检测算法相对于 C 语言版本的算法效率提高数十倍,经过本文的算法改进后,算法效率又提高 3~5 倍。

5 结 论

本文从算法本身和执行模式两方面对 Canny 算法进行了优化和加速。

(1)通过对原算法各个执行步骤进行分析和优化,在不影响边缘检测效果的前提下,减少了 Canny 算法的计算复杂度,充分提高了算法的执行速度。

(2)利用 GPU 的多线程并行执行能力,对图像处理进行了串行转并行的移植。将 $4\ 096 \times 4\ 096$ 大小的图像的边缘检测时间从 80 ms 压缩到了 6 ms 以内。执行速度相比 C 版本以及 FPGA 和相关参考文献都有较大提升,大大提高了算法的实用性。

此外,GPU 在提供相同计算能力的基础上,功耗比硬件平台要小很多。此外,GPU 能够进行嵌入式硬件开发,有望将算法的执行速率进一步提高。

参考文献:

- [1] CANNY J. A computational approach to edge detection[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986,6:679-698.
- [2] 丁鹏,张叶,刘让,等.结合形态学和 Canny 算法的红外弱小目标检测[J].液晶与显示,2016,31(8):793-800. DING P,ZHANG Y,LIU R,*et al.*. Infrared small target detection based on adaptive Canny algorithm and morphology[J]. *Chinese Journal of Liquid Crystals and Displays*,2016,31(8):793-800. (in Chinese)
- [3] 朱玉斌,吴晓红,何小海.基于 GPU 图像边缘检测的实时性[J].电子测量技术,2009,32(2):140-142. ZHU Y E,WU X H,HE X H. Real-time edge detection based on GPU[J]. *Electronic Measurement Technology*,2009,32

- (2):140-142. (in Chinese)
- [4] 唐斌,龙文.基于GPU+CPU的CANNY算子快速实现[J].液晶与显示,2016,31(7):714-720.
TANG B, LONG W. Fast Canny algorithm based on GPU + CPU[J]. *Chinese Journal of Liquid Crystals and Displays*, 2016,31(7):714-720. (in Chinese)
- [5] 李大禹.基于多GPU的液晶自适应光学波前处理器[J].液晶与显示,2016,31(5):491-496.
LI D Y. Liquid crystal adaptive optics wavefront processor based on multi-GPU[J]. *Chinese Journal of Liquid Crystals and Displays*, 2016,31(5):491-496. (in Chinese)
- [6] 张宏薇,王仕洋,李宪龙,等.基于Hough变换的瞳孔识别方法研究与实现[J].液晶与显示,2016,31(6):621-625.
ZHANG H W, WANG SH Y, LI X L, et al. . Research and implementation of pupil recognition based on Hough transform [J]. *Chinese Journal of Liquid Crystals and Displays*, 2016,31(6):621-625. (in Chinese)
- [7] 张素文,陈志星,苏义鑫. Canny边缘检测算法的改进及FPGA实现[J].红外技术,2010,32(2):93-96.
ZHANG S W, CHEN ZH X, SU Y X. Improved Canny edge detection algorithm and implementation in FPGA[J]. *Infrared Technology*, 2010,32(2):93-96. (in Chinese)
- [8] Canny edge detection algorithm principle and its VC implement[EB/OL]. <http://blog.csdn.net/augusdi/article/details/12907151>.
- [9] 周海芳.遥感图像并行处理算法的研究与应用[D].长沙:国防科学技术大学,2003.
ZHOU H F. *Research and application of parallel acceleration in remote sensing image*[D]. Changsha: National University of Defense Technology, 2003. (in Chinese)
- [10] 徐亮,魏锐.基于Canny算子的图像边缘检测优化算法[J].科技通报,2013,29(7):127-131.
XU L, WEI R. An optimal algorithm of image edge detection based on Canny[J]. *Bulletin of Science and Technology*, 2013,29(7):127-131. (in Chinese)
- [11] 曾文静,万磊,张铁栋,等.复杂海空背景下弱小目标的快速自动检测[J].光学精密工程,2012,20(2):403-412.
ZENG W J, WAN L, ZHANG T D, et al. . Fast detection of weak targets in complex sea-sky background[J]. *Opt. Precision Eng.*, 2012,20(2):403-412. (in Chinese)
- [12] 陈娟,陈乾辉,师路欢,等.图像跟踪中的边缘检测技术[J].中国光学,2009,2(1):46-53.
CHEN J, CHEN Q H, SHI L H, et al. . Edge detection technology in imaging tracking[J]. *Chinese Optics*, 2009,2(1):46-53. (in Chinese)
- [13] XU Q, VARADARAJAN S, CHAKRABARTI C, et al. A distributed canny edge detector: algorithm and FPGA implementation[J]. *IEEE Transactions on Image Processing*, 2014,23(7):2944-2960.
- [14] 周克良,周利锋,刘太钢,等.基于改进的Canny算子实时视频边缘检测系统在FPGA上的设计与实现[J].计算机测量与控制,2016,24(1):219-222.
ZHOU K L, ZHOU L F, LIU T G, et al. . Design and implementation of real-time video edge detection system based on improvement of canny algorithm on FPGA[J]. *Computer Measurement & Control*, 2016,24(1):219-222. (in Chinese)
- [15] 王希远,成荣,朱煜,等.基于FPGA的BiSS-C协议编码器接口技术研究及解码实现[J].液晶与显示,2016,31(4):386-391.
WANG X Y, CHENG R, ZHU Y, et al. . Research and realization of BiSS-C protocol encoder interface based on FPGA [J]. *Chinese Journal of Liquid Crystals and Displays*, 2016,31(4):386-391. (in Chinese)

作者简介:



张帆(1992—),男,河南周口人,硕士研究生,主要从事数字图像处理、GPU并行加速、机器视觉等方面的研究。
E-mail: zhangfan_6284@126.com



张立国(1961—),男,吉林长春人,研究员,主要从事空间光学方面的研究。
E-mail: zhangliguo@ciomp.ac.cn