

文章编号: 1003-501X(2012)03-0066-06

# 基于并行处理技术的谷物粒型快速测量算法

蒋 霓<sup>1,2</sup>, 段凌凤<sup>1,2</sup>, 杨万能<sup>1,2</sup>, 刘 谦<sup>1,2</sup>

(1. 武汉光电国家实验室(筹)—华中科技大学 Britton Chance 生物医学光子学研究中心, 武汉 430074;

2. 华中科技大学 生命科学与技术学院; 生物医学光子学教育部重点实验室, 武汉 430074)

**摘要:** 谷物粒型是决定谷粒品质和产量的重要参数之一。传统人工测量粒型的方法耗时、工作量大、主观性强。本文首先介绍一种基于线阵列采集技术和工业输送技术的谷物粒型自动测量系统。为提高系统测量效率, 文章中应用了图形处理器(GPU)并行处理技术, 在统一计算设备架构(CUDA)下对测量算法进行优化。实验结果表明, 基于 GPU 的并行加速算法, 能有效提高测量效率, 当图像中谷粒数近 2 000 颗时, 优化后的算法速度为中央处理器(CPU)下算法运行速度的 400 多倍, 且随着采集图像中谷粒数的增多, 优化测量算法的加速效果更显著。

**关键词:** 谷物粒型; 图形处理器; 并行处理技术; 加速算法

中图分类号: TP391

文献标志码: A

doi: 10.3969/j.issn.1003-501X.2012.03.012

## Fast Grain Shape Determination Algorithm Based on Parallel Processing

JIANG Ni<sup>1,2</sup>, DUAN Ling-feng<sup>1,2</sup>, YANG Wan-neng<sup>1,2</sup>, LIU Qian<sup>1,2</sup>

(1. Britton Chance Center for Biomedical Photonics, Wuhan National Laboratory for Optoelectronics-Huazhong

University of Science and Technology, Wuhan 430074, China;

2. Key Laboratory of Biomedical Photonics of Ministry of Education; College of Life Science and Technology,

Huazhong University of Science and Technology, Wuhan 430074, China)

**Abstract:** Grain shape is an important feature, which determines the grain quality and yield. Traditional manual determination of grain shape is time-consuming, laborious and subjective. A system for automatically determining the grain shape is presented based on a line-scan camera and a conveyor belt. In order to improve the grain shape determination efficiency, parallel processing technique based on Graphics Processing Unit (GPU) is used. On the basis of Compute Unified Device Architecture (CUDA), the grain shape determination algorithm is optimized. The experiment result shows that the GPU-based accelerated algorithm can achieve a good effect on the measurement efficiency. When the number of grain in an image is approximately 2 000, the optimized algorithm gets a speedup of more than 400 times. Moreover, as the quantity of kernels in the images gets larger, the performance of the grain shape determination algorithm is improved more significantly.

**Key words:** grain shape; graphics processing unit; parallel processing; acceleration algorithm

## 0 引 言

谷物粒型作为谷物外观的主要形状, 是评价谷物品质的一个重要指标之一<sup>[1-2]</sup>。此外, 粒型还与谷物重量密切相关, 在估计谷物产量中具有重要意义<sup>[3]</sup>。传统粒型(粒长与粒宽之比)测量方法为人工方法。按照中华人民共和国国家标准 GB1350-1999, 粒型人工测量方法是随机取 10 粒完整无损的糯米, 将糯米按头尾对接的方式排列, 用直尺测量总长度, 取其平均值为粒长。将糯米按肩靠肩的方向排列, 取宽度的平均值为粒宽。

收稿日期: 2011-10-09; 收到修改稿日期: 2011-11-09

基金项目: 湖北省国家自然科学基金重点项目资助(2008CDA087)

作者简介: 蒋霓(1988-), 女(瑶族), 广西贺州人。博士研究生, 主要研究工作是图像处理。E-mail: jiangni\_hust@126.com。

<http://www.gdgc.ac.cn>

计算粒长与粒宽的比值即为粒型。人工方法耗时, 工作量大, 且容易受测量人员的主观因素影响, 人为误差较大, 测量结果不准确。

机器视觉是一种快速、经济、一致和客观的测量技术, 在农业中的应用非常广泛<sup>[4]</sup>。利用机器视觉技术对谷物品质进行检测, 具有准确度高、重复性好等优点。目前, 根据图像采集方式, 用于检测谷物品质的机器视觉系统主要可分为基于平板扫描仪的系统<sup>[5-7]</sup>和基于数字摄像机的系统<sup>[7-9]</sup>。平板扫描仪虽然不受外界光照条件的影响, 但采集图像前需要手动将谷粒平铺放置于平板上, 且其扫描速度慢, 系统不能满足实时在线检测的要求。而基于数字摄像机的系统中, 可以借助传输系统将谷物运输到成像区域并对检测完的谷物进行收集, 实现谷物在线测量。为实时识别谷物中的饱满粒和空瘪粒, 测量实粒数、瘪粒数, 我们小组研发了一套结合可见光成像技术与 X 射线成像技术的机器视觉系统, 测量效率约每分钟 2 000 粒<sup>[10]</sup>。利用该系统进行粒型测量时, 需要对图像中每颗谷粒计算粒长、粒宽。当谷粒数较多时, 粒型计算量大, 系统效率的提高将会受限制。此外, 若系统中加入其它品质参数测量, 图像处理的运算复杂度增大也会影响实时测量系统的效率。为提供系统高速测量多个参数的可能, 粒型测量算法耗时的缩短非常必要。因此, 提高算法执行效率是保证系统高效率的重要手段。图形处理器(GPU)具有很强的浮点运算能力和存储带宽, 其强大的并行计算能力在图像处理算法诸如图像滤波<sup>[11]</sup>、直方图均衡化<sup>[12]</sup>、边缘检测<sup>[13]</sup>等中有广泛应用。本文采用线阵列采集技术和工业输送技术获取谷物图像, 并提出了一种基于 GPU 并行处理的谷物粒型快速测量算法。

### 1 自动测量系统和材料

系统的硬件部分主要包括给料机、线阵列摄像机、传输带、光源、收集装置和计算机。待测谷物通过给料机落到传输带上, 谷物随传输带运动的同时摄像机拍摄图像。图像经采集卡传输至计算机, 由计算机对图像进行处理获得粒型信息并控制整个系统运作。测量结束后, 谷物落入收集装置。系统组成结构细节介绍见文献[10]。系统检测总流程如图 1 所示。线阵列摄像机采集得到的图像经过校正、二值化、去噪、分割等图像预处理操作后, 得到谷粒二值图。然后基于并行处理技术, 在 GPU 中计算每颗谷粒的粒型。最后把计算结果拷回中央处理器(CPU)显示与存储。

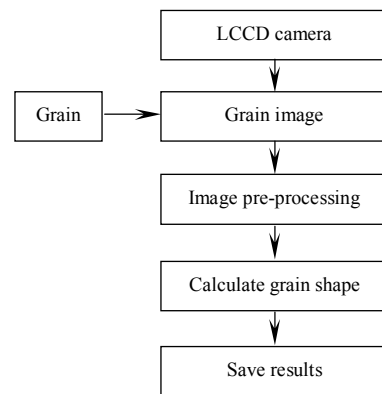


图 1 系统测量总流程

Fig.1 The overall process of the measurement

### 2 基于并行处理技术的粒型测量算法

本文利用谷粒轮廓, 求取单颗谷粒的粒长、粒宽, 计算粒长与粒宽的比值从而得到谷物粒型。粒长、粒宽示意图如图 2 所示。用  $f(x, y)$  表示谷粒轮廓, 点  $P_i(x_i, y_i)$  为轮廓上的任意点, 谷粒轮廓点个数为  $num$ 。设点  $P_1(x_1, y_1)$ 、 $P_2(x_2, y_2)$  为轮廓上距离最远的两点, 则以过点  $P_1$  和  $P_2$  的直线  $l$  为粒长所在直线,  $P_1, P_2$  两点间距离为粒长值。即粒长  $L$  的计算过程表达式为

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

$$L = \max(d_{ij}) \quad (2)$$

式中:  $d_{ij}$  为轮廓点  $P_i(x_i, y_i)$  和  $P_j(x_j, y_j)$  的距离,  $0 \leq i, j < num$  且  $i \neq j$ 。

按照人工测量粒型方法, 基于图像测量的粒宽可定义为谷粒在垂直粒长方向上的投影宽度, 即平行于

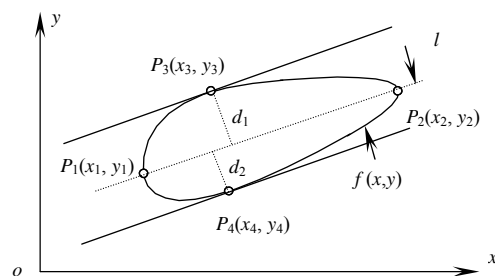


图 2 粒长、粒宽计算示意图

Fig.2 Computation of grain length and width

$l$  且与  $f(x, y)$  相切的两切线之间的距离。从图 2 中可以看到, 直线  $l$  将谷粒轮廓划分为直线上方和直线下方的两部分, 分别计算这两部分轮廓点到粒长直线的距离最大值, 即可找到切点  $P_3(x_3, y_3)$ 、 $P_4(x_4, y_4)$ 。记两切点到直线  $l$  的距离分别为  $d_{11}, d_{12}$ , 则粒宽  $w$  为  $d_{11}$  与  $d_{12}$  之和。若将距离值记为有符号值, 粒宽  $w$  的计算过程表达式为

$$l: y = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) + y_1 \quad (3)$$

$$d_{ii} = \frac{(y - y_1)(x_2 - x_1) - (x - x_1)(y_2 - y_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \quad (4)$$

$$w = \max(d_{ii}) - \min(d_{ii}) \quad (5)$$

式中:  $d_{ii}$  为轮廓点  $P_i(x_i, y_i)$  到直线  $l$  的距离,  $0 \leq i < \text{num}$ 。

由于每颗谷粒粒长、粒宽的计算互不相关, 因此可以利用 GPU 技术, 并行计算每颗谷粒粒型。基于并行处理技术的谷物粒型测量算法流程如图 3 所示。

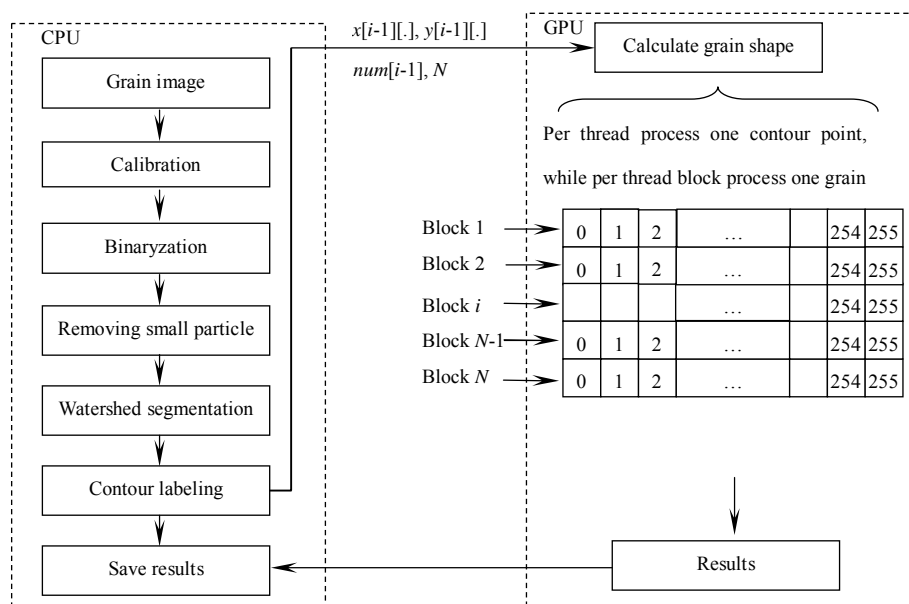


图 3 粒型测量算法流程

Fig.3 The flowchart of the fast grain shape determination algorithm.

步骤 1: 图像校正。系统传输带的运动速度与相机采集速度不匹配, 因此, 为保证粒长、粒宽测量的准确性, 采用模板校正的方法对原始图像进行校正。

步骤 2: 二值化。二值图像包含了完整的谷粒形状信息, 为减少后续处理的复杂度, 对图像进行二值化处理。系统中传输带为黑色, 即图像中背景颜色为黑色。因此, 可选用合适的固定阈值对图像进行处理, 得到二值图像。

步骤 3: 去噪。由于图像采集时光照不均匀等环境的干扰导致图像中存在一些噪声, 图像中面积小于阈值的连通区域被认为噪声点并去除。

步骤 4: 分水岭分割。考虑到谷粒粘连现象, 利用分水岭算法对图像分割。将图像中的粘连谷粒分割为单颗谷粒, 得到谷粒的二值图像。为避免分水岭过分割, 分水岭分割前对图像进行距离变换与灰度重建操作。

步骤 5: 轮廓标记。记步骤 4 得到的二值图像为  $I_b$ , 对  $I_b$  腐蚀得到图像  $I_c$ 。将图像  $I_b$  与图像  $I_c$  相减得到谷粒轮廓图像  $I_o$ 。从上至下, 从左至右扫描图像  $I_o$ , 为图像中每颗谷粒的轮廓标记。第一个扫描到轮廓上的像素点标记值为 1, 之后轮廓的标记值依次加 1。标记过程中, 同时保存每颗谷粒轮廓点的坐标值并计数轮廓点。对标记值为  $i$  的谷粒, 轮廓点的横纵坐标值分别存于数组  $x[i-1][:], y[i-1][:]$  中, 轮廓点个数保存

于数组  $num[i]$ 。标记完成后, 可得到所有谷粒的轮廓坐标及谷粒个数  $N$ 。

步骤 6: 粒型计算。将数组  $x[i-1][.]$ ,  $y[i-1][.]$ ,  $num[i]$  数据拷入 GPU 中, 并行计算每颗谷粒的粒型。最后把计算结果拷回中央处理器(CPU)显示与存储。具体粒型并行化测量算法描述如下。

### 2.1 算法并行化

算法并行执行性能的优化基于算法本身的可并行性, 即数据并行性。因此, 要实现算法的并行执行, 首先需要对算法结构化, 尽可能寻找算法计算过程中的数据无关性。经分析, 粒型计算过程中主要包含 4 个并行计算部分。

- a) 由于每颗谷粒粒型计算时只需利用其轮廓点的坐标信息, 因此每颗谷粒的粒型计算可以并发执行。
- b) 粒长计算式(2)可以写成:

$$d_i = \max(d_{i0}, d_{i1}, \dots, d_{i(num-1)}) \tag{6}$$

$$L = \max(d_i) \tag{7}$$

式中  $d_i$  为轮廓点  $i$  到其余轮廓点的最大距离。即对每个轮廓点可通过共享所有轮廓点的坐标, 并行计算  $d_{i0}$ 。

- c) 轮廓点到粒长直线的距离由该轮廓点坐标和粒长直线确定, 不同轮廓点的  $d_{ii}$  可以进行并行计算。
- d) 求取距离最大值时, 可使用削减操作<sup>[14]</sup>, 最大化算法的并行性。

如图 4 所示, 获得所有谷粒轮廓点坐标后, 可以并行计算每颗谷粒的粒型参数。在每颗谷粒的粒型参数计算过程中, 不同轮廓点间距离、不同轮廓点到直线的距离都可以并行计算。

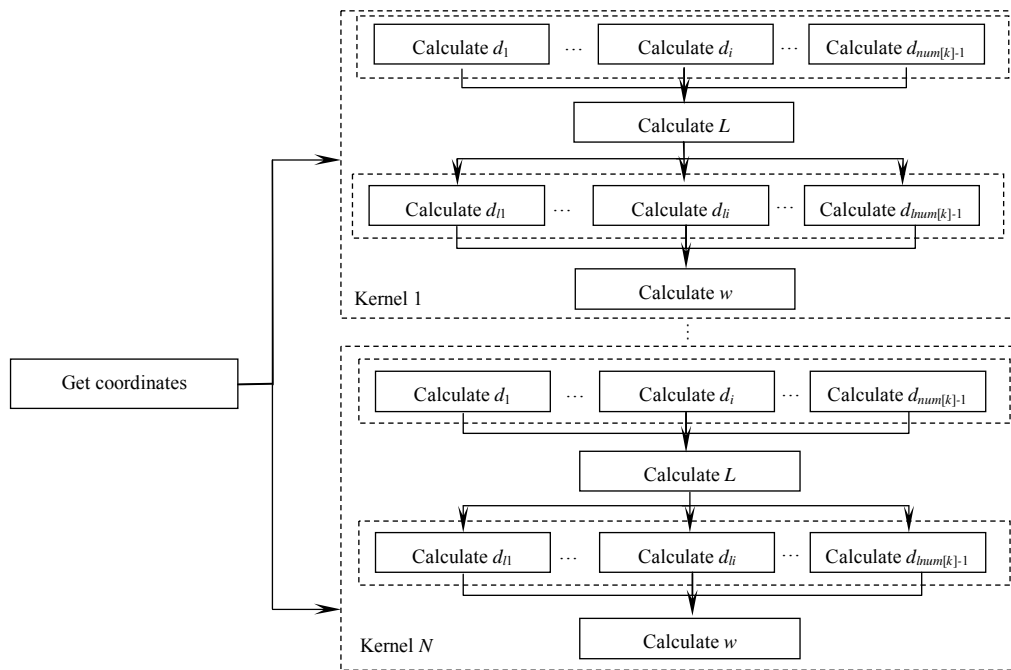


图 4 粒型测量算法流程

Fig.4 The flowchart of the fast grain shape determination algorithm

### 2.2 GPU 内核函数配置

统一计算设备架构(CUDA)编程模型中, GPU 上运行的程序部分以内核函数定义, 每个核函数由线程网格中的不同线程并行执行。每个线程可根据线程号对不同的数据执行相同的内核函数, 即单指令多数据。线程数量由所处理的数据大小决定, 而每个线程块内的线程数量有限, 同时线程块内可共享的存储空间也有限。因此, 为内核函数合理安排配置线程块, 最大化可用计算资源利用率, 是充分利用 GPU 并行性能的前提之一。本系统中利用的显卡为 GeForce 9800 GT, 每个线程块中最大线程数为 512, 线程网格中最大线程块数为  $512 \times 512 \times 64$ 。根据 2.1 节中分析, 假设图像中谷粒数为  $N$ , 则设置  $N$  个线程块, 每个线程块负责计算一颗谷粒的粒型。而每颗谷粒粒型计算中的并行部分, 则可以利用线程块中的每个线程来实现。具体处理方式线程  $i$  负责处理轮廓点  $i$ , 使各线程并行计算  $d_i$ , 然后同步线程块中的线程计算  $L$  和直线  $l$  方

程,再用每个线程并行计算  $d_{li}$ ,最后同步各线程计算  $w$ 。由于每个线程块中的线程数必须相同,考虑到轮廓点个数不多于 256,配置每块线程数为 256,程序运行中通过实际轮廓点个数控制各线程是否执行。

### 2.3 数据存储类型

CUDA 存储模型中,每个线程可直接访问寄存器和本地存储器,每个线程块内的所有线程可通过共享存储器共享数据,线程网格内所有线程均可访问全局存储器、常量存储器和纹理存储器。不同存储器的访问模式不同,空间大小不同,速度也不同。组织存储器访问,优化存储器利用率,使存储器带宽最大化,是程序性能的优化的重要步骤。按照内核函数配置,线程块中每个线程计算时要使用该线程块对应谷粒的所有轮廓点坐标,即  $f(x, y)$ 。为减少访问延迟,同时考虑数据大小,可使用共享存储器存储  $f(x, y)$ 。另外,计算  $L$  和  $w$  过程中,对求取距离最大值进行削减操作时,各线程之间需要数据共享,因此计算中间结果  $d_i$ ,  $d_{li}$  也采用共享存储器存储。

## 3 实验结果与分析

供试材料为中花 11 水稻。利用系统采集的 30 幅谷粒图像(图像分辨率为 2 715×3 350)对基于 GPU 并行处理优化后的算法进行性能测试。算法优化前后,即在 CPU 和 GPU 上运行耗对比如表 1 所示。其中算法执行时间包括轮廓提取、轮廓标记和粒型计算三个部分。系统使用 CPU 为 Intel (R) Xeon(R),主频为 2 GHz,内存为 3 GB,系统使用 GPU 为 GeForce 9800 GT。

表 1 算法优化前后的执行时间比较

Table 1 Time cost and speedup of the optimized method

No.	CPU time/ms	GPU time/ms	Speedup	No.	CPU time/ms	GPU time/ms	Speedup
1	4 081	114	36	16	8 563	114	75
2	2 399	105	23	17	2 120	104	20
3	10 293	112	92	18	1 836	109	17
4	1 792	109	16	19	5 619	106	53
5	2 729	105	26	20	1 696	108	16
6	3 478	106	33	21	1 555	103	15
7	7 394	108	68	22	82 631	207	399
8	26 928	133	202	23	24 172	130	186
9	3 290	107	31	24	1 799	105	17
10	2 534	105	24	25	1 931	103	19
11	2 399	104	24	26	29 826	141	211
12	2 257	110	21	27	1 791	105	17
13	51 601	167	309	28	2 026	103	20
14	2 356	104	23	29	14 406	122	118
15	2 255	110	21	30	58 194	170	342

从表 1 中可以看到,使用 GPU 并行处理技术优化算法后,算法执行速度明显提高。结果说明,粒型测量算法具有很好的并行性,可以充分利用 GPU 的并行处理能力,同时计算多颗谷粒的粒型,从而有效地提高算法速度。如表 1 所示,30 幅图像粒型计算耗时不等,算法优化后加速比也不同。这是因为不同图像中谷粒数目不同,谷粒数越多,粒型计算耗时就越长。测试 50 幅图像,比较不同谷粒数图像粒型计算所需要的时间,结果如图 5 所示。

如图 5 中曲线所示,加速比随谷粒数的变化而变化。图像中谷粒数目越多,GPU 并行加速效果越明显。当图像中谷粒数为 2 000 颗时,加速比接近 450 倍。出现这种现象的原因是随着谷粒数的增多,处理数据量增大,GPU 执行中线程切换、系统调度开销等占用的执行时间较少。

## 4 结 论

本文通过分析谷物粒型测量算法中的并行性,利用 GPU 并行处理技术对谷物粒型算法进行加速,实

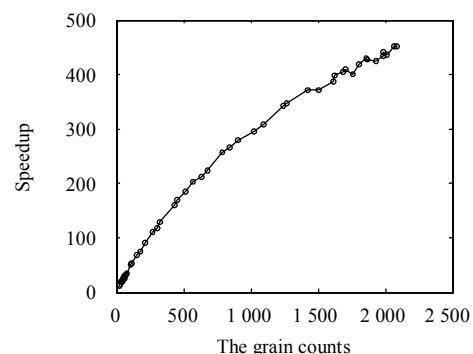


图 5 不同谷粒数图像粒型计算加速比

Fig.5 The speedup of images with different grain counts

现了粒型的快速测量。随着图像中谷粒的增多, GPU 的加速优势更明显, 当谷粒数为 2 000 颗时, 优化后的算法加速比达 400 多倍。基于并行处理技术的粒型测量算法满足实时、高通量的测量要求, 为系统性能提高、多参数高效测量提供了条件。此外, 该算法对其他颗粒状物体的尺寸测量也具有适用性。

#### 参考文献:

- [1] 罗玉坤, 朱智伟, 陈能, 等. 中国主要稻米的粒型及其品质特性 [J]. 中国水稻科学, 2004, **18**(2): 135-139.  
LUO Yu-kun, ZHU Zhi-wei, CHEN Neng, *et al.* Grain Types and Related Quality Characteristics of Rice in China [J]. **Chinese Journal of Rice Science**, 2004, **18**(2): 135-139.
- [2] LI Ji-ming, Thomson Michael, McCouch Susan R. Fine Mapping of a Grain-Weight Quantitative Trait Locus in the Pericentromeric Region of Rice Chromosome 3 [J]. **Genetics** (S0016-6731), 2004, **168**: 2187-2195.
- [3] JIANG Gong-hao, HONG Xue-yu, XU Cai-guo, *et al.* Identification of Quantitative Trait Loci for Grain Appearance and Milling Quality Using a Doubled-Haploid Rice Population [J]. **Journal of Integrative Plant Biology** (S1672-9072), 2005, **47**(11): 1391-1403.
- [4] Brosnan Tadhg, SUN Da-wen. Inspection and Grading of Agricultural and Food Products by Computer Vision Systems-a Review [J]. **Computers and Electronics in Agriculture** (S0168-1699), 2002, **36**: 193-213.
- [5] Dalen G van. Determination of the Size Distribution and Percentage of Broken Kernels of Rice Using Flatbed Scanning and Image Analysis [J]. **Food Research International** (S0963-9969), 2004, **37**: 51-58.
- [6] Emadzadeh B, Razavi S M A, Farahmandfar R. Monitoring Geometric Characteristics of Rice During Processing by Image Analysis System and Micrometer Measurement [J]. **International Agrophysics** (S0236-8722), 2010, **24**: 21-27.
- [7] Yadav B K, Jindal V K. Monitoring Milling Quality of Rice by Image Analysis [J]. **Computers and Electronics in Agriculture** (S0168-1699), 2001, **33**: 19-33.
- [8] Yadav B K, Jindal V K. Dimensional Changes in Milled Rice (*Oryza sativa* L.) Kernel During Cooking in Relation to Its Physicochemical Properties by Image Analysis [J]. **Journal of Food Engineering** (S0260-8774), 2007, **81**: 710-720.
- [9] WAN Y N, LIN C M, Chiou J F. Rice Quality Classification Using an Automatic Grain Quality Inspection System [J]. **Transaction of the ASAE** (S0001-2351), 2002, **45**(2): 379-387.
- [10] DUAN Ling-feng, YANG Wan-neng, BI Kun, *et al.* Fast Discrimination and Counting of Filled/Unfilled Rice Spikelets Based on Bi-modal Imaging [J]. **Computers and Electronics in Agriculture** (S0168-1699), 2011, **75**: 196-203.
- [11] Fialka O, Cadik M. FFT and Convolution Performance in Image Filtering on GPU [C]// **Information Visualization**, London, July 5-7, 2006. DC: IEEE Computer Society, 2006: 609-614.
- [12] 杨志义, 朱娅婷, 蒲勇. 基于统一计算设备架构技术的并行图像处理研究 [J]. 计算机测量与控制, 2009, **17**(4): 734-737.  
YANG Zhi-yi, ZHU Ya-ting, PU Yong. Study on Parallel Image Processing Based on Compute Unified Device Architecture [J]. **Computer Measurement and Control**, 2009, **17**(4): 734-737.
- [13] 左颢睿, 张启衡, 徐勇, 等. 基于 GPU 的快速 Sobel 边缘检测算法 [J]. 光电工程, 2009, **36**(1): 8-12.  
ZUO Hao-rui, ZHANG Qi-heng, XU Yong, *et al.* Fast Sobel Edge Detection Algorithm Based on GPU [J]. **Opto-Electronic Engineering**, 2009, **36**(1): 8-12.
- [14] Mark Harris. Optimizing Parallel Reduction in CUDA [EB/OL]. [http://developer.download.nvidia.com/compute/cuda/1\\_1/Website/projects/reduction/doc/reduction.pdf](http://developer.download.nvidia.com/compute/cuda/1_1/Website/projects/reduction/doc/reduction.pdf), 2007-11.