

# CELL 处理器并行实现立体匹配算法

许岩岩<sup>1,2</sup>, 陈 辉<sup>1</sup>, 刘家驹<sup>1</sup>, 袁金钊<sup>1</sup>

(1. 山东大学 信息科学与工程学院, 济南 250100; 2. 麻省理工学院 土木与环境工程系, 剑桥 02139)

**摘要:**提出了适应于 IBM CELL BE 并行处理器的两种立体匹配算法——动态规划(DP)算法和置信传播(BP)算法的实现方案。改进了 BP 算法中的数据成本计算方法,采用重叠窗方法保持 BP 算法原有的匹配效果。实验证明,基于 CELL 处理器的并行实现极大地提高了生成视差图的运算速度。针对室外场景噪声较大的特点,提出采用 Sobel 边缘检测和 Residual 算子对图像进行预处理,实验结果表明,该方法改善了生成视差图的效果。本文算法在低成本硬件设备平台上提升了立体匹配技术的计算效率,在汽车自动和辅助驾驶系统、多媒体视觉技术的硬件一体化中具有重要的应用价值。

**关键词:**计算机应用;人工智能;CELL 芯片;视差图;计算机视觉;并行计算

**中图分类号:**TP391    **文献标志码:**A    **文章编号:**1671-5497(2017)03-0952-07

**DOI:**10.13229/j.cnki.jdxbgxb201703036

## Cell processor stereo matching parallel computation

XU Yan-yan<sup>1,2</sup>, CHEN Hui<sup>1</sup>, LIU Jia-ju<sup>1</sup>, YUAN Jin-zhao<sup>1</sup>

(1. School of Information Science and Engineering, Shandong University, Jinan 250100, China; 2. Department of Civil & Environmental Engineering, Massachusetts Institute of Technology, Cambridge 02139, USA)

**Abstract:** This paper presents an efficient parallel implementation of belief propagation and dynamic programming algorithms on an IBM CELL BE processor (Sony Playstation3) that speeds up stereo image matching. Overlapping windows and improved data cost with pyramid computation are adopted for implementation of belief propagation algorithms. Considering that real-world images are typically degraded by various types of noise, differing exposures and changes in light etc, Sobel edge and residual operator are used for image pre-processing to improve the stereo matching results. The proposed method has important application on hardware unification of automatic driving binocular computer vision together with multimedia visual technique.

**Key words:** computer application; artificial intelligence; CELL processor; disparity map; computer vision; parallel computation

## 0 引言

立体匹配是将现实场景中同一物体在不同视

点(摄像机)投影图像中的像素点对应起来。计算机双目技术采用两台平行摄像机模拟人类视觉系统,使用立体匹配算法在左、右图像中寻找 3D 环

**收稿日期:**2016-01-13.

**基金项目:**国家自然科学基金重点项目(61332015);山东省自然科学基金项目(ZR2013FM032).

**作者简介:**许岩岩(1987-),男,在站博士后.研究方向:计算机视觉,城市计算. E-mail:yanyanxu@mit.edu

**通信作者:**陈辉(1963-),女,教授,博士.研究方向:计算机视觉. E-mail:huichen@sdu.edu.cn

境中物体的对应点。基于计算机视觉技术的自动和辅助驾驶系统通过计算对应点在不同视点图像中的位置差异(视差)感知周围环境,探测前方障碍物的位置和距离<sup>[1]</sup>。然而基于全局信息的立体匹配算法运算量巨大,传统计算架构的串行计算方法无法满足实时应用需要。近年来随着多核处理器的普及,并行立体匹配算法成为国内外的研究热点。颜瑞<sup>[2]</sup>采用基于CUDA的GPU并行架构对立体匹配算法进行加速。李金奎<sup>[3]</sup>采用普通的Intel多核架构,基于OpenMP对置信传播算法进行改进,缩短了运算时间。

本文提出采用IBM公司开发的具备特殊计算架构的CELL BE处理器,实现对传统算法的并行改进。由于提供了SIMD和向量编程等高级计算功能,CELL BE非常适用于需要极高密度海量计算和数据吞吐的应用。通过与传统CPU

的比较,展示了该架构对置信传播(BP)<sup>[4]</sup>和动态规划(DP)<sup>[5]</sup>立体匹配算法的加速效果。本文的创新之处在于:首次将DP算法在CELL BE架构上并行实现;采用两种新方法对BP算法进行并行化改进,并与串行算法计算结果进行比较<sup>[6]</sup>;同时,本文采用Sobel和Residual算子对图像进行预处理<sup>[7,8]</sup>,有效地降低了室外场景中的不均匀光照对立体匹配算法的影响。

## 1 Cell broadband engine (CELL BE)

与普通PC编程不同,BP和DP算法在CELL BE上实现需要考虑CELL的特殊结构。由于SPE只有256 kB本地存储,这对BP算法是非常严峻的挑战。图1为CELL芯片的总体结构。

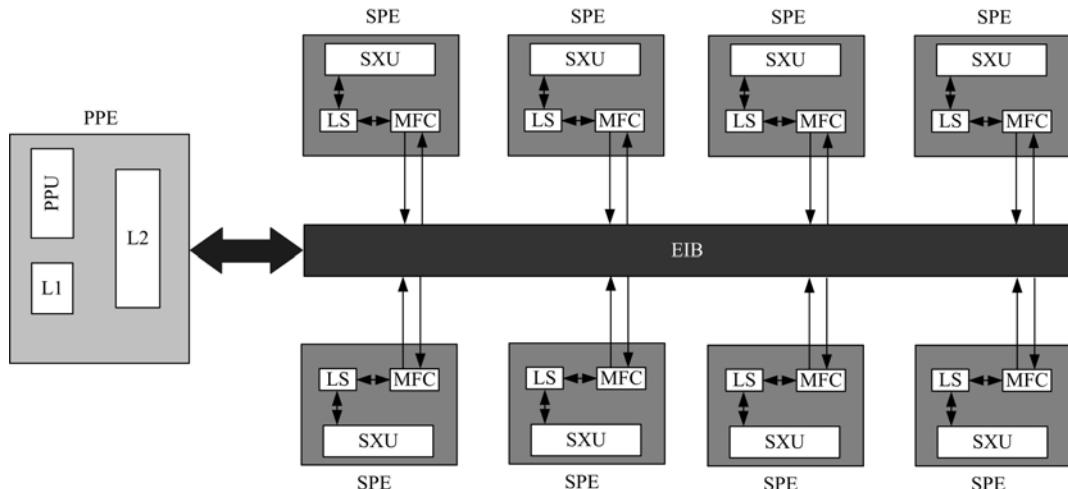


图1 CELL BE 总体结构

Fig. 1 Architecture of CELL BE

本文采用一台索尼公司出产的装有CELL BE芯片的游戏机PlayStation3作为测试用的主机。CELL架构芯片由一个Power架构的控制单元PPE以及8个能够执行SIMD操作的协处理器单元SPE(只有6个提供给用户做计算使用,另外2个PS保留),所有计算单元通过高速环形数据总线—EIB(Element interconnect bus)相连。

PPE是一个64 bit兼容PowerPC架构的两路并发多线程处理器,主频为3.2 GHz。包含一个32 kB的L1 Cache(32 kB数据cache和32 kB指令cache);512 kB的L2 Cache。如图2所示,它支持SIMD和浮点处理,也支持DMA和存储映像输入/输出(MMIO)。PPE在CELL中的

地位就像指挥部,一方面为操作系统的运行提供环境;另一方面指挥其他硬件工作,如为SPE分配任务,启停线程。

SPE是协处理器单元,主频为3.2 GHz,采用128 bit精简指令集,特别适用于大数据量和高计算密度的SIMD应用。SPE架构如图3所示,是Cell BE的真正优势所在,每颗Cell芯片有8个SPE,但在PlayStation中,只有6个开放给第三方开发使用。SPE只能访问本地存储(LS),它通过DMA方式访问主存(Main memory),由DMA负责将指令和数据搬到LS。这种方式能够有效地减小内存读写延迟对处理器利用率的影响。

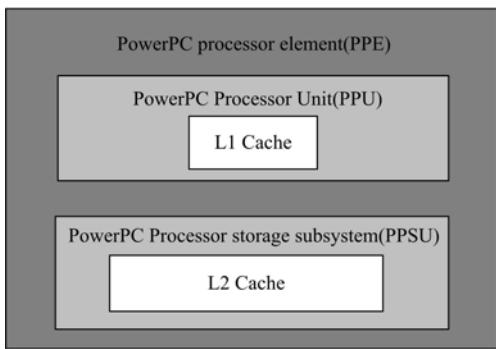


图 2 PPE 架构图

Fig. 2 PPE block diagram

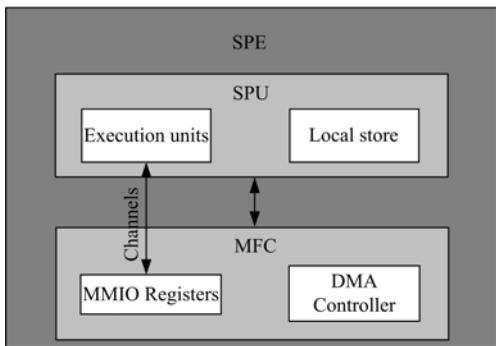


图 3 SPE 架构图

Fig. 3 SPE block diagram

为了协调多颗芯片之间的工作, CELL BE 提供几种消息机制供 SPE 与 PPE 之间通信使用。其中最常用的是 mailbox, 每个 SPE 单元都有一个四接口 32 bit 接入信箱和两个单接口 32 bit 输出信箱。信箱机制常用来协调不同 SPE 之间的工作,SPE 只需向 PPE 的输入信箱发送特定消息, 就可以使 PPE 中断操作, 使 PPE 短暂中止特定的 SPE 线程。此外, 信箱机制还能够“唤醒”或者“沉睡”空闲的 SPE。

## 2 测试数据

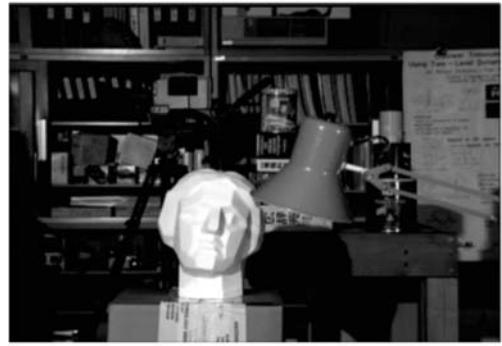
用于测试的图像 Tsukuba 选自 Middlebury 大学的立体图像数据库, 图像规格为 384 × 288<sup>[9]</sup>, 作为算法的综合测试数据; 同时, 用测试车辆(HAKA1)在新西兰拍摄的室外场景(图像规格为 640 × 480)作为测试图像, 如图 4 所示。这些图像中含有各种噪声, 包括光线变化、模糊、镜面和内部反射等。

## 3 基于 CELL 架构的并行算法

首先分析 DP 和 BP 算法的并行潜力, 然后提出两种算法在 CELL 多核芯片上并行实施方案。



(a)HAKA1实拍图像序列



(b)Tsukuba图像

图 4 测试图像

Fig. 4 Test image sequences

### 3.1 动态规划实施方法

由于 DP 算法逐行处理图像, 特别适合 CELL 架构处理方式。每行由一个线程处理, 每个 SPE 处理一个线程。如果图像有 N 行, 在整个图像处理过程中, 每个 SPE 要执行 N/6 行。虽然 DMA 传输数据的限制是每行数据必须少于 16 kB, 但足够用来处理本文中的测试图像。并行实现的算法步骤如下所示。

Step1 PPE 创建 SPE 上下文, 并启动 SPE 线程。

Step2 PPE 将数据从硬盘读入到 Main memory 中。

Step3 PPE 通过使用消息机制 Mailbox, 将要选取数据的 Main memory 地址传递给 SPE。

Step4 SPE 通过直接内存访问方式(DMA), 将尚未计算的数据从主存下载到自己的局部存储 (LS) 中。步骤如下:

Step4.1 六个 SPE 同时启动 6 线程, 计算图像对应点的差值, 即同时有 6 行图像在进行计算。每一行在计算时都采用矢量编程的方法进一步加强并行度。

Step4.2 将运算结果回写至主存中。

Step4.3 取下一行像素, 返回步骤 Step4.1

迭代运算。

Step5 PPE 负责将计算得到的视差图输出至外围存储设备。

在编程实现时,采用双缓存(Double cache)技术进一步对算法加速。双缓存类似诊所候诊室,病患大部分时间都是在等待叫号,为了提高效率会在等待时做一些其他工作。CELL 处理器同样适用这种方法,在编写代码时,可以一次申请两块内存,一块用来处理数据;另一块利用这段时间采用 DMA 方式传输数据。另外,为了避免上文 Step4 在抽取数据时遗漏或重复,使用 Mailbox 和 MFC 机制同步协调不同的 SPE。

DP 算法极易受噪声影响,沿行传播匹配误差,形成条纹瑕疵。为此,考虑采用一种空间域传递方法 DP-spatial(DPs)修正算法<sup>[10]</sup>。 $f^y$  和  $f^{y-1}$  分别为当前计算的行和先前计算的行的视差图。 $h^y$  修正后的视差图如下:

$$h^y = (1 - \lambda_1) f^y + \lambda_1 f^{y-1} \quad (1)$$

式中:  $\lambda_1$  为修正参数。

图 5 为两种算法的测试结果。在 Cell 处理器上图像会被 SPE 等分,6 个 SPE 各处理一部分,且各部分同时进行。因此修正效果仅限每一块内,并不是特别理想。

### 3.2 置信传播的并行实现

BP 算法是一种信息在网络中传播的迭代推理算法,能有效解决计算机视觉领域的早期视觉问题。本文将 Felzenszwalb 等<sup>[4]</sup>提出的基于普通 PC 的 BP 立体匹配算法改进并移植到 CELL 并行系统中。BP 算法是二维能量最小化问题<sup>[11]</sup>,其能量方程为:

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p, q) \in N} V(f_p - f_q) \quad (2)$$

式中:  $D_p(f_p)$  为像素  $p$  在视差为  $f_p$  时的代价函数,本文采用对应点像素灰度值之差的绝对值作为匹配代价; $V(\cdot)$  为指示函数,即当  $f_p = f_q$  时, $V = 0$ ,否则, $V = d$ ,其中  $d$  为常量,用来控制代价  $D$  的上限,本文设置  $d = 20$ ;  $P$  代表图像中所有像素点;  $N$  为像素图像中上、下、左、右 4 个方向像素集合;  $f_p$  为像素  $p$  的视差值;  $f$  表示一幅完整的视差图,整个计算的目标是为每一个像素点分配一个视差值,使式(2)代表的能量值最小<sup>[7]</sup>。

文献[6]提出采用红黑格法和金字塔法两种方法加速 BP 算法。红黑格法将像素点分割成黑点和红点,在第  $t$  个迭代阶段,信息从黑点传送到

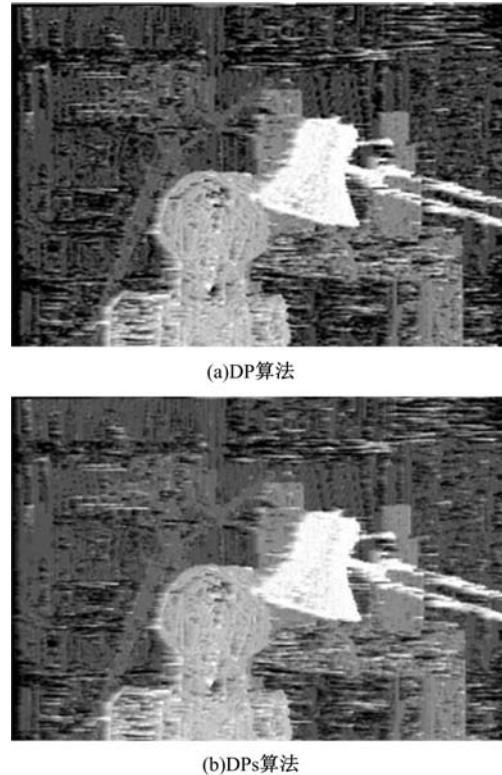


图 5 两种算法在 CELL 芯片上计算的视差图

Fig. 5 Disparity image by DP and DPs on CELL

红点,红点只接受信息不发送信息;在  $t+1$  个迭代阶段信息从红点传送到黑点,同样黑点只接受红点发送的信息不发送信息,所以在每个迭代阶段只需计算一半信息传输量。

金字塔算法兼顾运算速度和运算精度,在满足精度要求的同时极大地加快了运算速度。本文使用的是通用高斯金字塔,金字塔的上一层像素点数量是下一层像素点数量的  $1/4$ ,缩短了像素点之间的距离,使信息传播得更快速、高效,将上一层的迭代结果传给下一层作为迭代的初始值。这样可以尽快完成尽可能多的迭代次数,算法的全局性也越好。

BP 算法以高密度运算为代价获得高精度视差图。因此本文采用 CELL 芯片做并行化处理。

#### 3.2.1 方法一

由于每个 SPE 只有 256 kB 局部存储单元,而 DMA 每次只能传递 16 kB 的数据,所以 SPE 无法处理一幅完整图像。为此,对图像进行切割,每一块有  $\hat{M}$  行  $\hat{N}$  列,每个像素点采用 20 byte 进行信息传递,每一个视差值都需要窗口的空间存放数据,窗口大小由式(3)决定:

$$\text{float } \hat{M} \times \hat{N} < L_{\text{DMA}} \quad (3)$$

式中:  $L_{\text{DMA}}$  为 DMA 的运量, 限制为  $16 \times 1024$  bytes。

从式(3)得出窗口的大小取决于视差值的范围。以 Tsukuba 为例, 若设定视差范围值  $n = 16$ , 则  $\hat{M} = 32, \hat{N} = 16$  比较合适。

CELL BE 提供了 3 种方法(MailBox, DMA 和 notice)进行 SPE 和 PPE 之间的通信。MailBox 只有 32 bit, 用来为 DMA 传送数据块地址, 或者传递一些特殊的标号, 通知 PPE 唤醒或关闭哪个 SPE。将整幅图裁成不连续的图像对, 每对图像块之间有部分重叠区域, 重叠窗口如图 6 所示。

W1	W1 W2	W1 W2 W3	W2 W3 W4	W3 W4 W5	W4 W5	W6
W1	W1 W2	W1 W2 W3	W1 W2 W3	W3 W4 W5	W4 W5	W6
W1	W1 W2	W1 W2 W3	W1 W2 W3	W3 W4 W5	W4 W5	W6

图 6 重叠窗口示例

Fig. 6 Sample of overlapping windows

### 3.2.2 方法二

由图 7 可见, 将整个图像分解为小图块进行处理效果并不理想, 所以考虑采用其他并行处理方式。不同于方法一将图片分成若干块, 方法二将 BP 整个运行过程分成计算数据成本、生成金字塔、信息传递<sup>[6]</sup>和计算深度(视差)4 个阶段, 如图 8 所示。

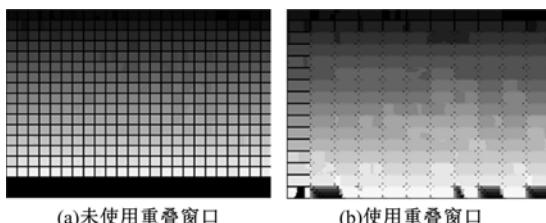


图 7 Tsukuba 的结果

Fig. 7 Results on Tsukuba

由图 8 可见, PPE 将数据发送到数个 SPE 之后, 再将中间阶段结果传回主存储单元。方法二可以最大程度地保证图像的完整性, 这对全局算法非常重要。

与文献[6]相比, 本文的第 1 点贡献是数据成本计算方法。将图像在 Main memory 中的地址

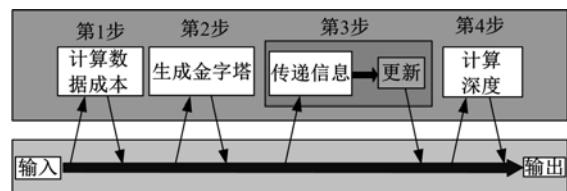


图 8 BP 算法流程

Fig. 8 Sketch of BP procedure on CELL BE

分配给多个 SPE, SPE 根据地址取的像素数量  $s$  由式(4)决定:

$$[\overbrace{2s+n}^{\text{image\_data}}, \overbrace{s}^{\text{data\_cost}}] \times \overbrace{4}^{\text{float}} < L_{\text{DMA}} \quad (4)$$

解得:

$$s < \frac{L_{\text{DMA}} - 4n}{4(n+2)} \quad (5)$$

通过式(5)可见, 当视差值  $n = 16$  时(Tsukuba 的最大视差),  $s$  取值 192 较为合适, 对于真实室外环境图像  $n=32$ (更远距离对于周围环境探测没有意义),  $s$  取值为 160。本文使用 Mailboxes 机制协调 6 个 SPE 确保不遗漏也不重复计算。

本文的第 2 点贡献是构建金字塔的方法。当使用金字塔算法加速时, 在完成第  $l$  层最后一次迭代计算时, 将上、下、左、右 4 个方向的值的地址通过 Mailboxes 发送到 4 个 SPE 中, SPE 从 Main memory 中取值而后生成  $l-1$  层的数据, 再将结果返回。SPE 一次取像素点的数量满足式(6):

$$s < \min \left( \frac{L_{\text{DMA}} - 4n}{4(n+2)}, \frac{L_{\text{width}}}{l} \right) \quad (6)$$

这样可以同时构建 4 个金字塔, 从而大大缩短了计算时间。

由图 9 可见, 并行算法与传统串行算法相比, 结果无明显差别。BP 算法与 DP 算法(图 5)相比, 生成的视差图更精确、细节反映更细腻。进一步实验, 采用方法二对 HAKA1 拍摄的图片进行处理。直接采用原始图像计算, 结果并不理想。原因是室外实拍图像受到各种噪声的影响, 尤其当两台摄像机光照条件不同时, 同一物体得到的图像灰度值不同, 从而造成误匹配。通过分析大量的实验数据发现, 尽管物体的灰度值不同, 但是物体的边缘轮廓并不随光照变化而改变。可见, 光噪声更多地影响图像的低频部分而较少影响高频部分, 故采用 Sobel 边缘检测和 Residual 算子对图像进行预处理, 保留对光照变化不敏感的高频成分。

图 10(a)为经 Sobel 算子处理后的图像, 作为

BP 算法的输入处理对象<sup>[7]</sup>。图 10(b)为残差图<sup>[8]</sup>,指从一幅原始图像中减去自身模糊图像而



(a)BP方法二的结果



(b)Tsukuba图像对的真实视差

图 9 BP 并行算法结果和实际视差图

Fig. 9 BP result when following strategy 2  
and ground truth

(a)Sobel边缘检测



(b)均值残差图

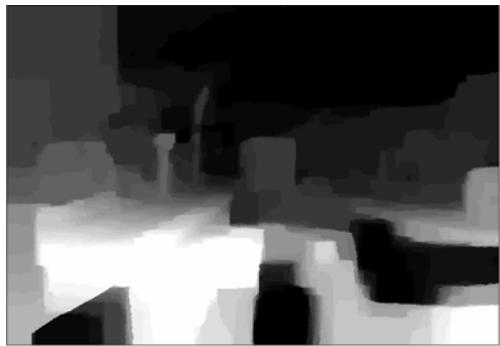
图 10 图像预处理

Fig. 10 Image preprocessing

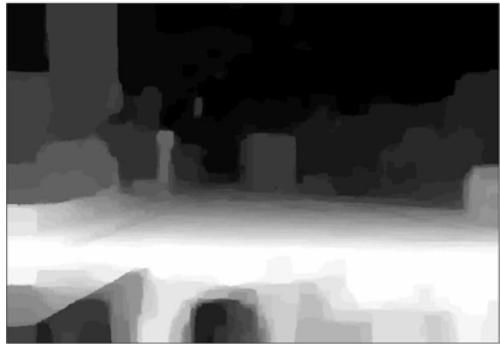
得到的锐化图像,自身模糊图像是由平滑原图像得到的。文献[8]的实验结果表明,在众多方法中,效果最好的是均值滤波法,故选择  $3 \times 3$  均值滤波器。均值滤波器近似一个优质低通滤波器,所得残差图像含有高频部分用来进行匹配运算(即纹理部分),可以通过迭代选择留下的信息量。

#### 4 实验结果

采用具备 CELL 处理器的 PlayStation3 (PS3)作为实验平台。该平台提供 1 个 3.2 GHz 的 Cell 处理器、6 个 SPEs(只用了其中 4 个)和 1 个 PPE,运行在 fedora 操作系统之上,同时还使用了 IBM 公司提供的 CellSDK 3.1<sup>[12]</sup> 工具箱。图 11 为原始图像、经 Sobel 和 Residual 预处理之



(a)原始图像



(b)Sobel



(c)Residual

图 11 BP 不同图像上运行的结果

Fig. 11 BP results when using different image data

后使用 BP 算法的结果,通过对比可见采用两种方法预处理之后,效果明显改善。

对于 DP 算法,采用主频 3 GHz 的传统 PC 处理一幅图像需要 4.3 s;采用 CELL 并行处理,在使用 4 个 SPE 时需要 0.09 s,6 个 SPE 时需要 0.06 s,即 1 s 可以处理超过 15 帧。在 CELL 处理器与传统 PC 主频大致相等的情况下,性能提升 70 倍以上。对于 BP 算法,用标准的 Tsukuba 图像( $384 \times 288$ ),整幅图像采用五层金字塔,每层迭代 5 次,最大视差为 16,输入图像的平滑常数  $\sigma$  为 0.7;truncation-of-discontinuity-costs 参数为 1.7;truncation-of-data-cost 参数为 15;数据成本权重  $\lambda$  为 0.07。按照以上设置在普通 PC 上 BP 算法的运行时间为 25.2 s,在 CELL 上采用方法一运行为 4.1 s,方法二运行为 2.1 s。与传统串行算法相比,性能提升了 12 倍。理论上在 6 个 SPE 上采用 SIMD 方式消耗的时间应该是实验值的两倍,这主要得益于本文在 PS 上采用了适量的编程技术和恰当的内存管理方式(比如 double buffer)。需要注意的是,与文献[6]相比,本文算法改进效果更明显(本文速度提升 12 倍,文献[6]提升 7 倍)。文献[6]并没有给出算法详细的参数设置,所以很难直接进行评价。

BP 算法中方法二的效果远优于方法一,如图 7 和图 9 所示。需要注意的是,在 PS 中 SPE 只有 256 kB 本地内存,DMA 一次只能传送 16 kB 内存,这制约了计算性能。

## 5 结束语

采用 SIMD 理念对 DP 和 BP 立体匹配算法进行并行化改进,采用重叠窗口、改进的数据成本计算和金字塔构建方法,在多核处理器 CELL 上对本文算法进行了实验验证。实验结果表明:采用 Sobel 边缘检测和 Residual 算子对输入图像进行预处理改善了输出结果;与传统架构的 CPU 相比,本文算法大幅提高了运算速度。本文算法对低成本视觉方法的实现有重要应用价值,也为在移动计算平台上实现自动和辅助驾驶系统以及多媒体视觉技术奠定了基础。本文所采用的并行化分析方法同样适用于其他计算机视觉算法。

## 参考文献:

[1] Khan W, Klette R. Stereo accuracy for collision

avoidance for varying collision trajectories [C] // IEEE Intelligent Vehicles Symposium, Gold Coast City, Australia, 2013: 1259-1264.

- [2] 颜瑞. 基于 CUDA 的立体匹配及去隔行算法[D]. 杭州:浙江大学信息与电子工程学院, 2010. Yan Rui. The stereo matching and de interlacing algorithm based on CUDA[D]. Hangzhou: College of Information and Science and Electronic Engineering, Zhejiang University, 2010.
- [3] 李金奎. 一种分层置信传播立体匹配并行算法[J]. 通信技术, 2012, 45(7): 57-61. Li Jin-kui. Parallel HBP algorithm for stereo matching[J]. Communications Technology, 2012, 45(7): 57-61.
- [4] Felzenszwalb P F, Huttenlocher D P. Efficient belief propagation for early vision[J]. International Journal of Computer Vision, 2006, 70(1): 41-54.
- [5] Ohta Y, Kanade T. Stereo by two-level dynamic programming[C] // Proceedings of the 9th International Joint Conference on Artificial Intelligence, Los Angeles, USA, 1985: 120-1126.
- [6] Lai Chi-Hua , Hsieh Kun-Yuan, Lai Shang-hon, et al. Parallelization of belief propagation method on embedded multicore processors for stereo vision [DB/OL]. [2015-12-26]. [http://pllab.cs.nthu.edu.tw/act/ESWeek2008/ESWeek\\_files/estimemedia2008-kyhsieh.pdf](http://pllab.cs.nthu.edu.tw/act/ESWeek2008/ESWeek_files/estimemedia2008-kyhsieh.pdf).
- [7] Guan S S, Klette R. Belief-propagation on edge images for stereo analysis of image sequences[J]. Lecture Notes in Computer Science, 2008, 4931: 291-302.
- [8] Vaudrey T, Klette R. Residual images remove illumination artifacts! [J]. Lecture Notes in Computer Science, 2009, 5748: 472-481.
- [9] Scharstein D, Szeliski R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms [DB/OL]. [2015-12-29]. <http://vision.middlebury.edu/stereo/taxonomy-IJCV.pdf>.
- [10] Klette R. Concise Computer Vision[M]. London: Springer, 2014.
- [11] Sun Jian, Shum Heung-yeung, Zheng Nan-ning. Stereo matching using belief propagation[J]. IEEE Trans Pattern Analysis Machine Intelligence, 2003, 25: 787-800.
- [12] IBM. CBE programmer's guide v3. 1 [DB/OL]. [2015-12-30]. <http://www.ibm.com/developer-works/power/cell>.