

一种面向多核处理器的高效并行 PCA-SIFT 算法*

刘 仲,邢彬朝,陈跃跃

(国防科技大学 计算机学院,湖南 长沙 410073)

摘要:提出一种面向多核处理器的并行 PCA-SIFT 算法,采用数据级并行方法实现并行的特征提取和特征点匹配,将计算任务分配到各个 DSP 核并行处理,充分开发多核处理器的多级并行性。实验结果表明,并行 PCA-SIFT 算法对各种不同图像形变的图像具有良好的适应性,具有接近串行 PCA-SIFT 算法的图像匹配能力,平均加速比达 3.12。

关键词: 并行;PCA-SIFT;图像匹配;多核处理器

中图分类号:TP391.4 文献标志码:A 文章编号:1011-2486(2012)04-0103-05

An efficient parallel PCA-SIFT algorithm for multi-core processor

LIU Zhong, XING Binchao, Chen Yueyue

(College of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract: An efficient parallel PCA-SIFT algorithm for multi-core processor is proposed, which adopts data-level parallel method to implement parallel key points extraction and matching. The computing tasks were assigned to four DSP cores for parallel processing, which fully develops multi-level parallelism of multi-core processor. The experimental results show that the parallel PCA-SIFT algorithm has good adaptability for various distortion images and the image matching ability is close to the sequential algorithm with the average speedup 3.12.

Key words: parallel; PCA-SIFT; image matching; multi-core processor

随着数字图像处理和微电子技术的快速发展,图像匹配技术在飞行器辅助导航、人脸识别、图像拼接、图像检索和辅助医学诊断等方面得到了广泛应用^[1-2],这些应用要求高效、甚至实时的图像匹配算法。Lowe^[3]提出的 SIFT (Scale Invariant Feature Transform) 图像匹配算法是目前匹配效果较好的一种算法,对于图像的尺度变化、旋转、亮度、噪声甚至仿射变换具有良好的适应性。Yan Ke^[4]将主成分分析法 (Principal Component Analysis, PCA) 应用到 SIFT 算法的特征描述符生成中,降低了描述符的维数,加快了特征提取和特征匹配速度。但是 PCA-SIFT 算法的计算复杂度高,单芯片上的 PCA-SIFT 算法难以满足实时应用的需求,随着多核处理器的出现,基于多核处理器开发并行 PCA-SIFT 算法成为提高算法性能的一种有效方法。

1 多核处理器体系结构

YHFT-QDSP^[5]是一款高性能浮点异构多核 DSP 芯片,定点运行性能可达到 11.2GIPS,浮点

运行性能可达到 6.8GFLOPS。如图 1 所示, YHFT-QDSP 芯片由一个 32 位定点运算的 RISC 核和四个 32 位浮点运算的 DSP 核组成, RISC 核运行嵌入式操作系统,提供外部接口用于处理与用户交互、外围设备管理等应用,通过 MB 总线对四个 DSP 核进行控制和通信。

DSP 核间提供共享数据缓冲池 (Share Data Pool, SDP) 和 QLink 两种数据通信方式。SDP 用于 DSP 核间的少量、快速的数据通信和同步、控制信息的传输。QLink 是 DSP 核与 4×4 的 16 位交叉开关之间的接口,用于实现核间的大块数据交换。

2 串行 PCA-SIFT 算法流程

PCA 是一种对数据进行统计分析的技术,通过对数据进行简化,有效地找出数据中最“主要”的元素和结构,去除噪音和冗余,将原有的复杂数据降维。PCA-SIFT 算法将 PCA 理论应用于 SIFT 描述符的生成。

* 收稿日期:2011-11-16

基金项目:国家自然科学基金资助项目(60503042)

作者简介:刘仲(1971—),男,湖南邵东人,副研究员,博士,硕士生导师,E-mail:zhongliu@nudt.edu.cn

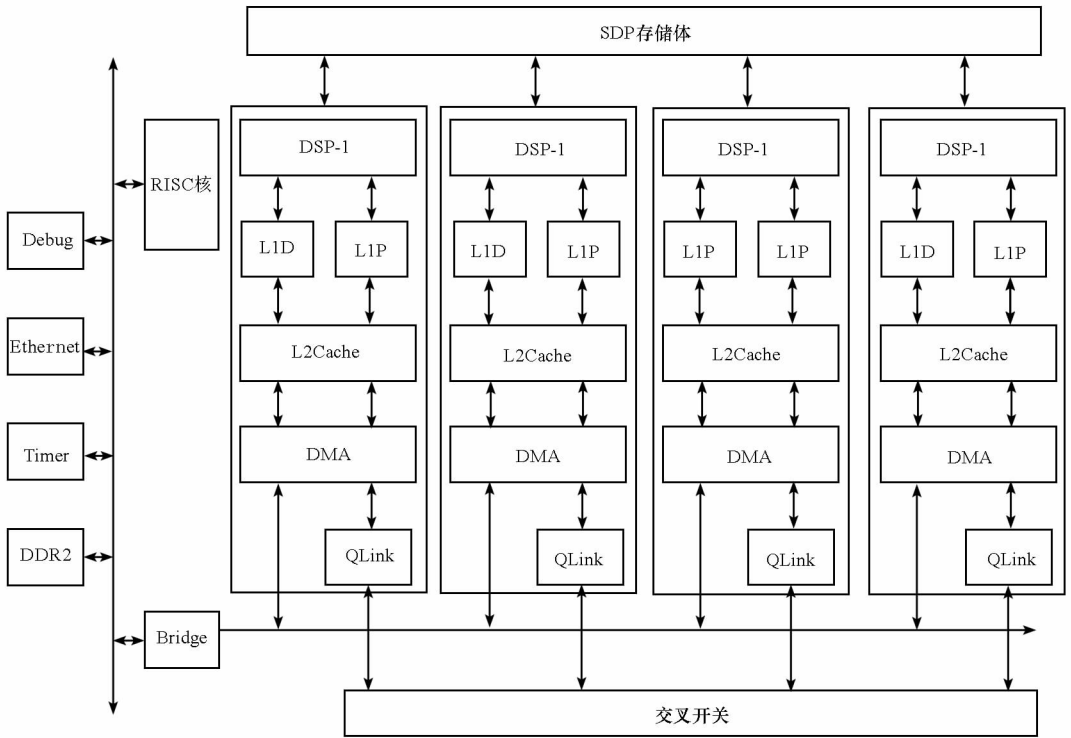


图 1 YHFT-QDSP 的体系结构

Fig. 1 Architecture of YHFT-QDSP

2.1 创建尺度空间

尺度空间是为了模拟图像的多尺度特征。一幅二维图像 $I(x, y)$ 的尺度空间用它与高斯函数 $G(x, y, \sigma)$ 的卷积来表示: $L(x, y, \sigma) = G(x, y, \sigma) \otimes I(x, y)$, 其中, $L(x, y, \sigma)$ 表示尺度函数, \otimes 表示卷积, $G(x, y, \sigma)$ 用式(1)表示:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x+y)^2}{2\sigma^2}} \quad (1)$$

其中, σ 是尺度因子, 它用于控制一幅图像被平滑的程度。

PCA-SIFT 算法用 DoG 构造的高斯差分金字塔来表示尺度空间, 而 DoG 用尺度图像的差值来表示, 如式(2)所示:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) \\ &\quad - G(x, y, \sigma)) \otimes I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (2)$$

在构建高斯金字塔时, 首先第一阶第一层将原始图像扩大 2 倍, 与不同尺度因子的高斯核 $G(x, y, \sigma)$ 进行卷积形成高斯金字塔的第一阶。第二阶的第一层图像是对第一阶的中间层图像下采样得到, 并与不同尺度因子高斯核 $G(x, y, \sigma)$ 进行卷积形成其余几层图像; 其余阶的形成方法类似。当高斯金字塔创建后, 对第一阶的相邻两层图像作差形成高斯差分金字塔的第一阶, 但层数比高斯金字塔少一层, 其他阶依次类推。

2.2 检测极值点

在高斯差分金字塔中, 除了每一阶的最低层和最高层图像, 其余每层图像的每个像素要和同尺度图像的相邻 8 个像素, 以及相邻尺度的上、下层图像所对应的各 9 个像素, 共 26 个像素值作比较, 若该点所对应的像素值为最大值或者最小值, 则将这个点记为一个极值点。

2.3 极值点的精确定位和过滤

由于图像的离散性, 使得检测到的极值点在连续图像函数中, 并不一定是极值点。所以, 要对极值点的位置进行精确定位; 此外, 有的极值点的值很小, 这样的点容易受到噪声的干扰而变得不稳定, 要给予去除。PCA-SIFT 算法对局部极值点进行三维二次函数拟合来精确定位特征点的二维图像位置和尺度。

2.4 极值点的主方向

为使特征描述符具有旋转不变性, 需要根据极值点邻域像素的梯度分布特性为每个极值点指定一个或多个方向基准。在高斯金字塔中, 像素点 (x, y) 处的梯度和方向的计算, 如式(3)所示:

$$\begin{aligned} m(x, y) &= \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \\ \theta(x, y) &= \arctan\left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}\right) \end{aligned} \quad (3)$$

统计极值点邻域像素点梯度的梯度方向直方图, 选取具有最大梯度值的柱所对应的方向作为

主方向。当存在一个大于最大梯度 80% 的另一个或者多个梯度柱时,将其所对应的方向作为辅方向;当一个特征点有辅方向时,把它也作为特征点的主方向,即一个特征点具有多个主方向;根据每个主方向分别计算特征描述符,即一个特征点有多个特征描述符,这可以增加匹配的鲁棒性。当选取主方向后,要对它进行插值,使其角度更加精确。

2.5 计算 PCA-SIFT 描述符向量

PCA-SIFT 算法通过投影矩阵计算生成描述符向量。投影矩阵是通过大量的有代表性图像的学习得到的,只需要计算一次,就可以对所有这种类型的图像使用。生成 PCA-SIFT 描述符向量的步骤如下:

(1) 在特征点的所在尺度上,把以特征点为中心的坐标轴逆时针旋转到特征点的主方向,用旋转后的坐标取相应的像素值;若旋转后坐标超出了图像的边界,则其像素值设为边界像素值;取一个大小为 41×41 的像素块。

(2) 除了像素块的外围像素,按如下公式计算每个像素点的垂直和水平梯度:

$$gx = \text{pixel}(r, c + 1) - \text{pixel}(r, c - 1)$$

$$gy = \text{pixel}(r + 1, c) - \text{pixel}(r - 1, c)$$

并将其放入一个向量中,得到一个 $39 \times 39 \times 2 = 3042$ 大小的梯度向量。

(3) 把第 2 步中生成的梯度向量进行归一化,将向量元素值限制在 0.2,再次进行归一化。

(4) 用大小为 36×3042 的投影矩阵乘以第 3 步归一化后的梯度向量,得到一个 36 维的向量,这个向量就是 PCA-SIFT 的描述符。

PCA-SIFT 的特征匹配采用遍历法,即实时图的每一个特征点在参考图的所有特征点中,计算与该点具有最小欧式距离 d_1 和次小欧式距离 d_2 的特征点,若其比值小于 0.7,则认为与该特征点具有最小欧式距离 d_1 的特征点为匹配点。

3 并行 PCA-SIFT 算法

3.1 并行特征提取

YHFT-QDSP 由四个 DSP 核构成,每个核都有独立的存储空间。基于 YHFT-QDSP 的并行算法主要有任务级并行和数据级并行两种方法,任务级并行是指根据算法的流程,按照每个子过程耗时的长短,对其进行流水线式的任务分割,实现整个任务的流水作业;数据级并行是指把待处理的数据均匀分配到四个 DSP 核,实现数据并行处

理的过程。

由上节可以看出,PCA-SIFT 算法具有较高的计算复杂性,算法的主要耗时部分是创建尺度空间和生成描述符,算法前后关联性较强,均匀分割任务困难。并且,若采用任务级并行方法,DSP 核间的中间数据通信量将非常大,不利于提高加速比。因此,我们采用对图像进行分块提取特征点的方法,即数据级并行方法,DSP 核间只需要传输特征点数据,数据通信量小,有利于提高加速比。因为 PCA-SIFT 算法提取的特征点属于图像的局部特征,所以,对图像分块分别提取特征点是可行的。

如图 2 所示,PCA-SIFT 算法的数据级并行方法的基本思想是:把图像分成四份,分别让四个 DSP 核对相应部分提取特征点。为了减少图像边缘对特征提取的影响,让每个 DSP 核多取一定宽度的图像,记为 d 。图中的实线表示将图像进行了四等份划分,实现和虚线之间的距离为 d ;DSP0 所取的图像为虚线所示图像边界,类似地,其他三个核也多取宽度为 d 的图像,以尽量减少图像边缘对特征提取的影响。 d 值的大小可以进行调整,实验表明,当 d 取 10~20 时,效果较好。

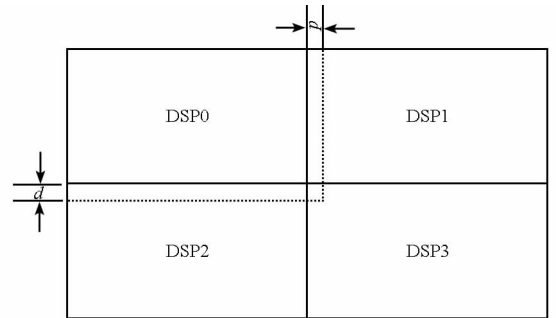


图 2 基于图像分块的并行特征提取
Fig. 2 Parallel extracting keypoints based on segmenting image

实现时,向四个 DSP 核的存储空间中加载两幅完整的图像,分别称为参考图和实时图;每个 DSP 核根据其 ID(取值范围是 0、1、2、3)号分别读取相应部分的图像;每个 DSP 核对所读取的图像同时进行特征提取。当特征提取完成后,四个 DSP 拥有参考图的特征点集合分别记为 A_0 、 A_1 、 A_2 和 A_3 ,实时图的特征点集合分别记为 B_0 、 B_1 、 B_2 和 B_3 ,这些特征点都存放于 DSP 核指定的存储空间中,用于后续的特征匹配。

3.2 并行特征点匹配

为了确保图像匹配的准确性,需要将实时图的每一个特征点在整幅参考图特征点中找它的匹

配点,以获取全部的匹配特征点集合。假定实时图的特征点集合 B0、B1、B2、B3 的特征点数目分别为 $n_0、n_1、n_2、n_3$,总数为 n ;参考图的特征点集合 A0、A1、A2、A3 的特征点数目分别为 $m_0、m_1、m_2、m_3$,总数为 m ,则总的特征点匹配次数可以按照如下公式计算:

$$\begin{aligned}
 Total &= \sum_{i=1}^n \sum_{j=1}^m b_i a_j = \sum_{i=1}^{n_0} \sum_{j=1}^m b_i a_j + \sum_{i=1}^{n_1} \sum_{j=1}^m b_i a_j \\
 &+ \sum_{i=1}^{n_2} \sum_{j=1}^m b_i a_j + \sum_{i=1}^{n_3} \sum_{j=1}^m b_i a_j = \left(\sum_{i=1}^{n_0} \sum_{j=1}^{m_0} b_i a_j \right. \\
 &+ \sum_{i=1}^{n_0} \sum_{j=1}^{m_1} b_i a_j + \sum_{i=1}^{n_0} \sum_{j=1}^{m_2} b_i a_j + \sum_{i=1}^{n_0} \sum_{j=1}^{m_3} b_i a_j \left. \right) \\
 &+ \left(\sum_{i=1}^{n_1} \sum_{j=1}^{m_0} b_i a_j + \sum_{i=1}^{n_1} \sum_{j=1}^{m_1} b_i a_j + \sum_{i=1}^{n_1} \sum_{j=1}^{m_2} b_i a_j \right. \\
 &+ \sum_{i=1}^{n_1} \sum_{j=1}^{m_3} b_i a_j \left. \right) + \left(\sum_{i=1}^{n_2} \sum_{j=1}^{m_0} b_i a_j + \sum_{i=1}^{n_2} \sum_{j=1}^{m_1} b_i a_j \right. \\
 &+ \sum_{i=1}^{n_2} \sum_{j=1}^{m_2} b_i a_j + \sum_{i=1}^{n_2} \sum_{j=1}^{m_3} b_i a_j \left. \right) + \left(\sum_{i=1}^{n_3} \sum_{j=1}^{m_0} b_i a_j \right. \\
 &+ \sum_{i=1}^{n_3} \sum_{j=1}^{m_1} b_i a_j + \sum_{i=1}^{n_3} \sum_{j=1}^{m_2} b_i a_j + \sum_{i=1}^{n_3} \sum_{j=1}^{m_3} b_i a_j \left. \right)
 \end{aligned}$$

从上式可以看出,特征点的匹配完全可以分布到四个 DSP 核进行并行处理。根据 YHFT-QDSP 的体系结构特点,我们设计了数据级并行的特征点匹配方法,即每个 DSP 轮流作为主控结点,而与各个 DSP 核并行的为主控 DSP 核执行计算任务并返回计算结果,如图 3 所示。DSP 核间的数据通信采用 QLink 与 SDP 相结合的方式,其中特征点数据采用适合大块数据传输的 QLink,特征点数目和数据同步信号采用适合小批量数据传输的 SDP,充分开发了 YHFT-QDSP 的多级并行性,加快了匹配的速度。

如图 3 所示,以 DSP0 作为主控 DSP 核为例,详述多核并行匹配的实现步骤如下:

(1) DSP0 把特征点集合 B0 的描述符数据,通过 QLink 发送至 DSP1、DSP2 和 DSP3;发送完毕后,把 B0 的描述符的个数,用 SDP 发送至 DSP1、DSP2 和 DSP3,并打开对这三个 DSP 核的信号灯,以示数据发送完毕;

(2) DSP0 在特征点集合 A0 中寻找实时图 B0 中每一个特征点的最小距离点和次小距离点,存于 C0。DSP1、DSP2 和 DSP3 检测到 DSP0 对其开放的信号灯后,读取数据并关闭 DSP0 所打开的信号灯;然后,DSP1、DSP2、DSP3 分别对所接受的每一个描述符在其相应的参考图特征点集合 A1、A2 和 A3 中,寻找与它具有最小距离和次小距离的特征点;计算完毕后,三个 DSP 核都把计算结果用 QLink 发送至 DSP0,分别存储于 C1、

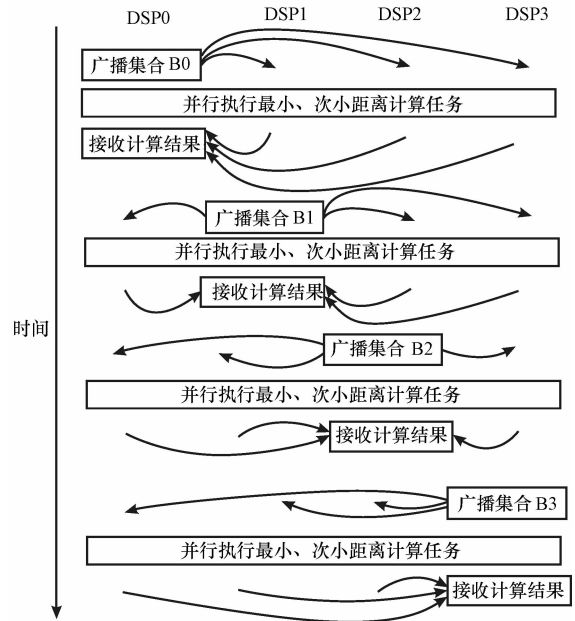


图 3 基于 YHFT-QDSP 的并行特征点匹配

Fig. 3 Parallel keypoints matching based on YHFT-QDSP C2、C3,并分别打开对 DSP0 的信号灯。

(3) DSP0 检测到信号灯后,读取数据并关闭信号灯;对于 B0 中的每一个特征点 b ,都有参考图中的 8 个特征点的信息(特征点的坐标和该点与特征点 b 的距离),从中选出最小距离和次小距离,若两者比值小于 0.7,则认为这两个点为匹配点;否则,不匹配,由此得到 B0 的特征匹配点集合。当 DSP0 处理完成时,打开对 DSP1 的信号灯。

DSP1、DSP2、DSP3 分别作为主控 DSP,重复类似于步骤(1)、(2)、(3)的处理过程,全部计算完成以后,得到参考图像与实时图像的全部特征匹配点集合。最后,采用剔除算法剔除误配点,得到正确的匹配特征点集合。

4 性能测试与分析

评价并行图像匹配算法的主要准则是加速比和重复率比。加速比是同一个任务在单处理器系统和并行处理器系统中运行消耗时间的比率,用来衡量并行系统或程序并行化的性能和效果。本文中,加速比 $= T_{DSP} / T_{YHFT-QDSP}$,其中 T_{DSP} 表示 PCA-SIFT 串行算法在单核 DSP 上的运行时间, $T_{YHFT-QDSP}$ 表示并行 PCA-SIFT 算法在 YHFT-QDSP 上的运行时间。重复率是指两幅图像的匹配点数和两幅图像所提取特征点数的平均值的比值,用来衡量算法的图像匹配能力。重复率 $K = C(I_1, I_2) / N$,其中 $C(I_1, I_2)$ 表示两幅图像的匹配点数, N 表示两幅图像所提取的特征点数量的平均值。

重复率比 = $K_{DSP}/K_{YHFT-QDSP}$, 其中 K_{DSP} 表示 PCA-SIFT 串行算法在单核 DSP 上的重复率, $K_{YHFT-QDSP}$ 表示并行 PCA-SIFT 算法在 YHFT-QDSP 上的重复率, 重复率比反映了并行算法相对串行算法的图像匹配能力, 重复率比越接近 1 表示并行算法的图像匹配能力越接近串行算法。

本节用尺度变化、旋转、光照明暗变化、噪声影响和仿射变换五种图像形变的样本对设计的并行算法进行综合测试, 从加速比和重复率比两方面分析与评价并行 PCA-SIFT 算法。实验中, 图像样本大小为 360×144 , 格式为 pgm。

图 4 分别对比了串行和并行 PCA-SIFT 算法针对尺度变化图像、旋转图像、光照变化图像、噪声影响图像和仿射变换图像的匹配结果。

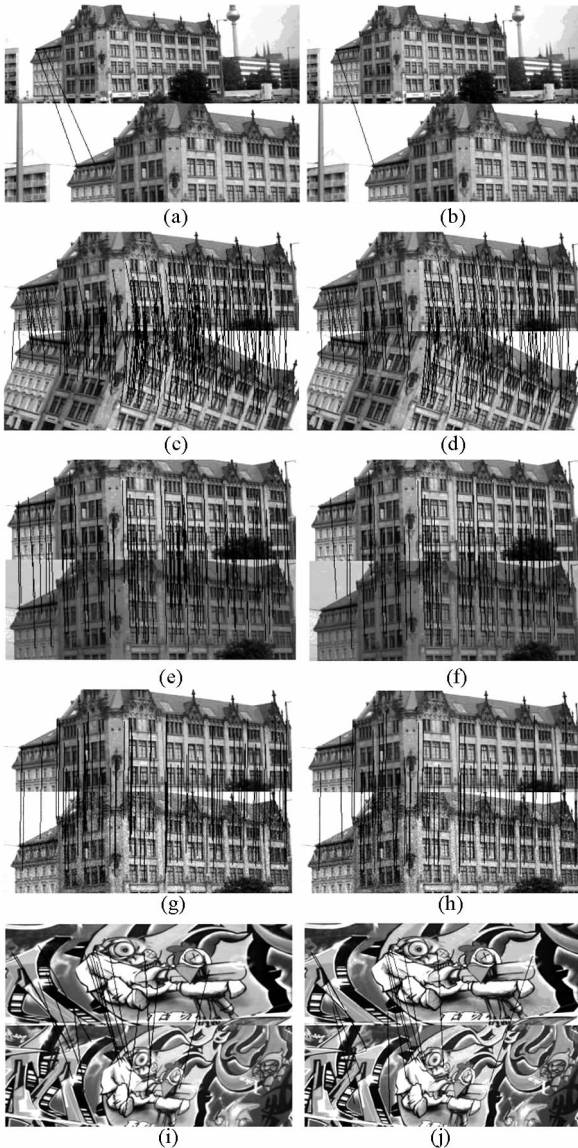


图 4 串行和并行 PCA-SIFT 算法的图像匹配结果对比

Fig. 4 Compare the image matching results of the serial and parallel PCA-SIFT algorithm

表 1 总结了 5 种不同图像形变样本的匹配结果。从表中可以看出, 并行 PCA-SIFT 算法的平均加速比为 3.12, 并行算法的平均重复率比为 71.1%, 表明并行 PCA-SIFT 算法与串行算法的图像匹配能力相当, 尤其是对仿射变换、光照明暗变化、旋转图像的匹配很接近。从图 4 也可以看出 PCA-SIFT 并行算法的图像匹配效果很好, 能够满足图像匹配要求。

表 1 不同图像形变样本的匹配性能结果

Tab. 1 Matching result of different deformation images

参数	尺度变化	图像旋转	光照明暗变化	噪声影响	仿射变换	平均
加速比	2.79	3.13	3.28	2.89	3.52	3.12
重复率比	50%	72.7%	75.5%	72.3%	86.7%	71.1%

5 结论

本文提出一种面向多核处理器的并行 PCA-SIFT 算法, 设计了数据级并行的并行特征提取和并行特征点匹配算法。根据 YHFT-QDSP 的体系结构特点, 计算任务在各个 DSP 核并行处理, DSP 核间的数据通信采用 QLink 与 SDP 相结合的方式, 其中特征点数据采用适合大块数据传输的 QLink, 特征点数目和数据同步信号采用适合小批量数据传输的 SDP, 充分开发 YHFT-QDSP 体系结构的多级并行性, 加快了并行 PCA-SIFT 算法的执行速度。实验结果表明, 并行 PCA-SIFT 算法对各种不同图像形变的图像具有良好的适应性, 具有接近串行 PCA-SIFT 算法的图像匹配能力, 平均加速比达 3.12。

参考文献 (References)

- [1] Morel J M, Yu G. ASIFT: a new framework for fully affine invariant image comparison [J]. SIAM J. Imaging Sciences, Society for Industrial and Applied Mathematics, 2009, 2(2): 438 - 469.
- [2] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features [C]//IEEE Conference on Computer Vision and Pattern Recognition, 2001: 511 - 518.
- [3] Lowe D C. Distinctive image features from scale-invariant keypoints [J]. International Journal of Computer Vision, 2004, 60(2): 91 - 110.
- [4] Ke Y, Suktharankar R. PCA-SIFT: A more distinctive representation for local image descriptors [C]//Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04), 2004:511 - 517.
- [5] Chen S M, Wan J H, Lu J Z, et al. YHFT-QDSP: high-performance heterogeneous multi-core DSP [J]. Journal of Computer Science and Technology, 25(2):214 - 224, 2010.