

一种基于 Zynq 的 ROS 软硬件协同计算架构设计与实现

赵可可¹, 柴志雷^{1,2}, 吴 东²

(1 江南大学 物联网工程学院, 江苏 无锡 214122;

2 数字工程与先进计算国家重点实验室, 江苏 无锡 214125)

摘 要: 针对复杂机器人算法在嵌入式系统上运行时间过长的问题, 设计并实现了一种在 Zynq 平台上通过软硬件协同机制实现机器人操作系统(ROS)节点的方法, 利用 FPGA 加速 ROS 中的复杂图像算法. ARM 端软件进行接口间的转换, 可以和其他 ROS 节点正常通讯. 随后测试了双目匹配算法 census 在系统中的实现. 实验证明, 在处理复杂图像算法时相对于纯软件实现具有明显优势.

关键词: 机器人操作系统; FPGA; 软硬件协同; 立体匹配

中图分类号: TP302.1

文献标识码: A

文章编号: 1000-7180(2017)09-0087-05

Design and Implementation of a ROS Software-hardware Co-processing Compute Architecture Based on Zynq

ZHAO Ke-ke¹, CHAI Zhi-lei^{1,2}, WU Dong²

(1 School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China;

2 State Key Laboratory of Mathematical Engineering and Advanced Computing, Wuxi 214125, China)

Abstract: Aiming to solve the robot complex algorithm running time is too long in the embedded system, a node method designed by software-hardware co-processing in the Zynq for robot operating system (ROS). Complex image algorithm of ROS in FPGA accelerated and ARM handle software interface between the ROS and other normal communication node. Then the stereo matching algorithm census is tested in the system. Experimental results show that in dealing with complex image algorithm relative to the pure software implementation has obvious advantages.

Key words: robot operating system; FPGA; software-hardware co-processing; stereo matching

1 引言

目前已有的机器人软件平台中, 机器人操作系统 ROS(Robot Operating System)以其开源和良好的拓展性逐渐成为其中的主流平台^[1]. ROS 是一种分布式处理框架, 这使可执行文件能被单独设计, 并且在运行时松散耦合^[2]. 使用 ROS 开发机器人应用, 只需要关注自己所要开发的算法, 这大大简化了开发周期. 但目前在 ROS 包中, 大多数算法因为其复杂性, 只适合在 PC 平台上运行, 并不能很好地运用在移动型机器人上面^[3].

另一方面, FPGA(现场可编程门阵列)依靠其

硬件并行和可重复编程能力广泛应用于通讯和图像处理领域^[4]. 但传统的 FPGA 开发主要使用硬件描述语言 HDL(Hardware Description Language)进行硬件设计, 所有的通讯和外设都需要开发人员设计. 后来, Xilinx 公司推出了用于提升编程抽象能力的 HLS^[5](High-Level Synthesis)来提高 FPGA 程序开发效率, 但开发人员在开发图像算法时依然需要了解很多硬件细节, 这些问题制约了 FPGA 在嵌入式领域的发展^[6].

本文提出了 ROS 在 Zynq 异构平台上的软硬件协同机制方法(FPGA_ROS), 将原来 ROS 算法中不适合在 ARM 上运行的图像算法利用 FPGA

进行硬件加速,然后在 ARM 上打包成节点发布到 ROS 系统中,实现与其他 ROS 节点的通信.

2 ROS 通讯模式

在 ROS 系统中,机器人的每一部分被设计成一个节点(node),节点之间的通讯依靠话题(topic)来实现,通过不同的节点之间协作可以实现各种功能.开发者只需要关注自己所需要的功能并把它作为一个节点即可.

ROS 通过发布(publish)/订阅(subscribe)机制来传递消息,如图 1 所示,在 ROS 系统中需要包含两个部分:发布者(publisher)和接收者(subscriber).发布者首先将消息发布到话题上,接收者如果订阅这个话题,就会接收到发布者所发布的信息,多个接收者可以订阅同一个话题.

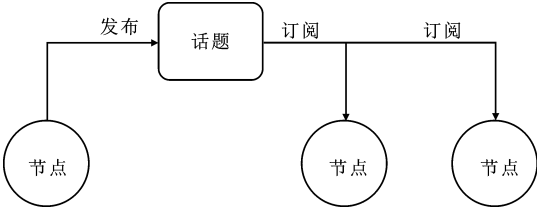


图 1 ROS 节点通讯模式

3 FPGA_ROS 机制

3.1 FPGA_ROS 介绍

FPGA_ROS 是将 FPGA 芯片作为一个 ROS 节点,同样遵循发布/订阅消息机制,可以和其他标准的 ROS 节点进行交互,FPGA_ROS 通过将需要运算的模块移交给 FPGA 处理,实现 FPGA 对 ROS 节点的加速.

FPGA_ROS 在接口上和 ROS 其他节点统一,作用只是为图像算法提供加速,提高嵌入式系统的处理速度.FPGA_ROS 相对于纯 ROS 的最大区别在于是否包含 FPGA 硬件加速.

3.2 硬件平台

FPGA_ROS 的硬件平台是 Zynq 系列芯片,Zynq 是 Xilinx 公司推出的包含 ARM+FPGA 的异构芯片,在 Zynq 系统芯片中,包含了基于 ARM 的处理系统(PS)和可编程逻辑单元(PL)^[6].图 2 为 FPGA_ROS 在 Zynq 上的实现方式,Linux 操作系统和 ROS 节点运行在 PS 端,ROS 系统中的图像算法适合在 FPGA 进行硬件加速的放在 PL 端,不适合 FPGA 加速的运行在 PS 端,PL 与 PS 之间的通讯依靠 AXI 总线来实现.

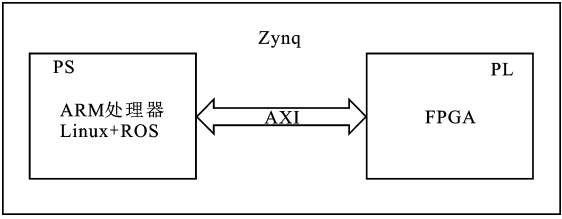


图 2 FPGA_ROS 在 Zynq 上的实现

3.3 FPGA_ROS 结构

FPGA_ROS 主要结构如图 3 所示,主要包含如下 4 个部分:

- (1)FPGA 加速模块;
- (2)ARM 与 FPGA 之间的交互接口;
- (3)订阅(Subscribe)接口;
- (4)发布(Publish)接口.

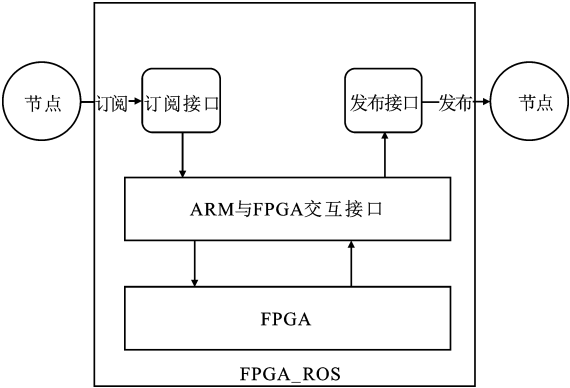


图 3 FPGA_ROS 总体结构

FPGA 加速模块主要执行图像算法的处理.图像算法采用 Vivado HLS 进行设计,使用 Vivado HLS 开发程序在保持性能的前提下加速了开发效率.

ARM 与 FPGA 之间的交互接口主要处理 PS 与 PL 之间需要处理的图像数据,图 4 所示为数据在 ZYNQ 中的传输过程.首先 PS 端从输入节点读取图像数据,通过共享内存的方式将图像数据映射到 PS 端内存去;然后内核映射数据到 PL 内存区中的读地址偏移量中,依靠 AXI 总线接口,FPGA 从基地址偏移读取读偏移量中的数据,直至读取完成.处理完成后,FPGA 将处理完成的数据通过 AXI 接口写到 PL 端写地址偏移量中,再将数据映射到 PS 内存区交给输出节点处理.

ROS 的处理包含订阅和发布接口两部分,首先,订阅接口订阅话题并接收输入数据,然后将这些数据转化为 FPGA 可识别的格式.输出接口接收 FPGA 处理完的结果,将结果打包成适合 ROS 系统的格式并发送到话题上.如果其他 ROS 节点订阅这

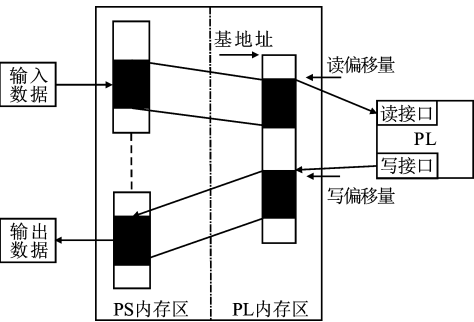


图 4 PS 和 PL 数据传输流程

个话题,能够保证获得正确的结果,这种机制可以保证软硬件协同实现的 ROS 节点(FPGA_ROS)在 ROS 系统中可以正确使用。

4 双目匹配算法 census 在 FPGA_ROS 上的实现

4.1 Census 算法

Census 匹配方法^[7-8]是一种基于区域的双目匹配算法,相较于 SAD 和 SSD 等匹配算法,对图像的结构要求不高,像素点之间的处理没有相关性,更适合在 FPGA 进行实现^[9]。

设 $I(u,v)$, $D(u,v)$ 分别为 u 行 v 列像素点的灰度值和此像素固定大小的 census 邻域窗口, $P(i,j)$ 串联起来为此像素的 census 编码 $C(u,v)$, d 为视差搜索范围, S_i 为 census 匹配的相似度算子:

$$P(i,j) = \begin{cases} 0, & I(i,j) > I(u,v), I(i,j) \in D(u,v) \\ 1, & I(i,j) < I(u,v), I(i,j) \in D(u,v) \end{cases} \quad (1)$$

$$S_i(u,v) = \sum_{u,v} \text{HAMMING}(C_i(u,v), C_r(u, v+i)) \quad i = 0, 1, \dots, d-1 \quad (2)$$

式中, $C_l(u,v)$ 和 $C_r(u,v)$ 表示基准图参考像素和对准图候选匹配像素的 census 编码。计算 d 个 S_i 数值后,比较得到最小的 S_j , j 即为最终得到的视差,如图 5 所示。

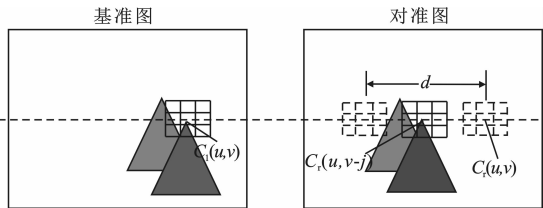


图 5 census 算法匹配原理

4.2 硬件实现

图 6 为 census 算法的硬件实现,在双目匹配

中,如果以左图做基准图,当左图像素点进行处理的时 候,与之相对应的对准图右图像素点也进行处理,因为需比较右图的 d 个像素点才能决定最终的视差值,所以右图在每一个像素点处理时需要加一个单位延迟器 z^{-1} ,在 census 编码后,对左右 census 编码进行异或操作,并统计 1 的个数,得到此像素点的汉明距离,然后对 d 个汉明距离做进行胜者为王决策 WTA(Winner-take-all),选取最小汉明距离,得到左像素点最终的视差值,因为像素点之间没有相关性,所以更适合在 FPGA 上进行硬件加速。

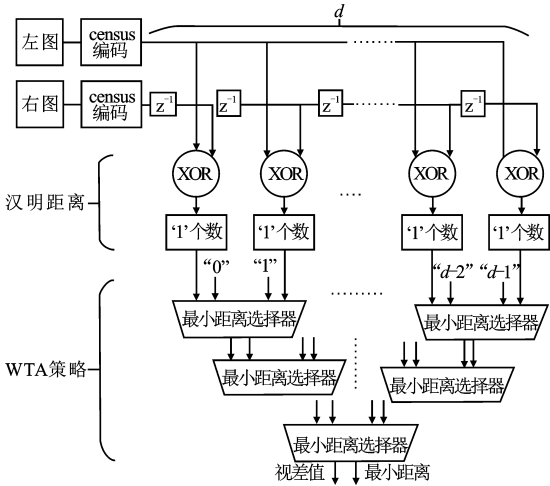


图 6 census 算法硬件实现框图

4.3 ROS 节点实现

图 7 为 census 算法在 ROS 系统中的框图,其中, left_image 和 right_image 节点负责读取双目中的左右图像,并发布到相应的话题 left_input 和 right_input 上。FPGA_ROS 节点在订阅到左右图像后,向 PL 端加载已经用硬件实现的 census 算法 IP(Intellectual Property)核,然后将图像转换后发送给 PL 端进行处理,PL 端处理完毕后打包成 ROS 图像格式发送到话题 out_image 上; display_result 节点接收话题 out_image 并将图像显示出来,其他节点也可以接收话题 out_image 并进行后续图像处理。

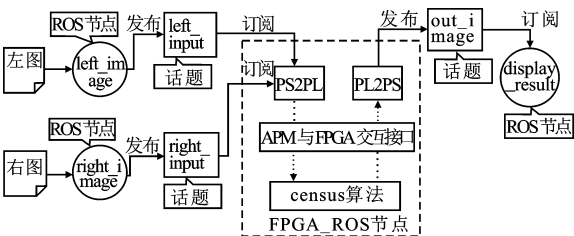


图 7 census 算法在 ROS 系统中的框图

5 综合实验

综合实现平台使用的是 Digilent 的 Zedboard 开发板,该开发板是包含 FPGA 和 ARM 的 ZYNQ (zynq7020clg484)异构芯片. 在纯嵌入式软件环境中,census 算法是在 ZYNQ 异构芯片的 PS 端上进行处理的,PS 端包含两个主频为 667 MHz 的 Cortex-A9 芯片. 对于软硬件协同的处理方式,ARM 端和 FPGA 端协同工作. 对于 PC 平台,采用的是主频为 3.3 GHz 的四核 Intel Core i5-4590 处理器,内存为 12 GB.

5.1 基于双目匹配算法软硬件协同的资源利用率

表 1 所示为基于 ZYNQ 双目匹配算法软硬件协同在 PL 端的资源消耗,从表中可以看到,BREAD_18K 和 LUT 资源消耗较多,这主要因为本文在 Census 编码中,将窗口大小 D 设置为 11,而这样需要在 BREAM_18K 中缓存左右图像各 12 行(其中多出来的一行用于流水),在运算中,需要在 FF 中保存多个 11×11 窗口大小的中间值. 缩小窗口大小可以减少资源利用率,但会增加误匹配^[10].

表 1 PL 端的资源利用率

逻辑资源	总共	利用率/%
BREAM_18K	280	34
DSP 48E	220	3
FF	106 400	13
LUT	53 200	65

5.2 用户编程模式

基于软硬件协同开发的 ROS 节点,与 PC 或者 ARM 上实现的普通 ROS 节点的使用完全相同,用户只需要知道发布和订阅的话题就可得到需要的结果. 如在 census 算法中,用户只需要发布图像到话题 left_input 和 right_input,就可以通过话题 out_image 订阅到经 FPGA 处理好的视差图像,这与在 PC 或者 ARM 上开发 census 算法的 ROS 节点所使用的方式完全相同,唯一的区别就是算法经过 FPGA 进行硬件加速.

5.3 三种平台下的实现时间比较和分析

本文测试了在这三种平台下实现 census 算法所用时间,其中,图像像素为 320×240 ,邻域窗口大小为 11,视差搜索范围 d 为 20. 如图 8 所示,基于 FPGA 加速的 Zynq 芯片只需要 0.02 s 就可以处理完一幅图片,比使用 PC 快了超过 30 倍,比 ARM 要快 100 多倍. 原因在于算法中每个像素点重复计算多次并转换为二进制比特位计算,更适合并行处理.

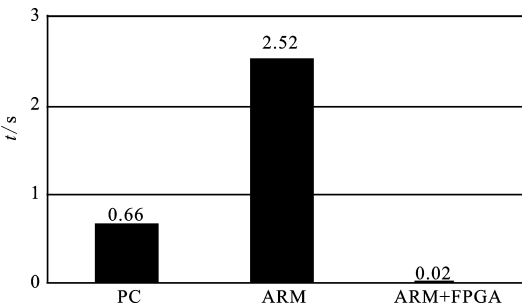


图 8 census 算法实现性能对比

图 9 为 ROS 在三种平台下从发布双目图片到接收到处理好的图片各阶段时间对比,主要包含 5 个部分:

- (1)订阅节点时间,从发布到接收到双目图片时间.
- (2)先软件处理时间,主要包括 ROS 接口转换为普通参数,软硬件协同还包括将双目图像从 PS 端映射到 PL 端的时间.
- (3)census 算法实现时间.
- (4)后软件处理时间,主要包括普通参数转换为 ROS 接口,软硬件协同还包括 PL 端映射视差图片到 PS 端的时间.
- (5)发布节点时间,从发布到接收到视差图片.

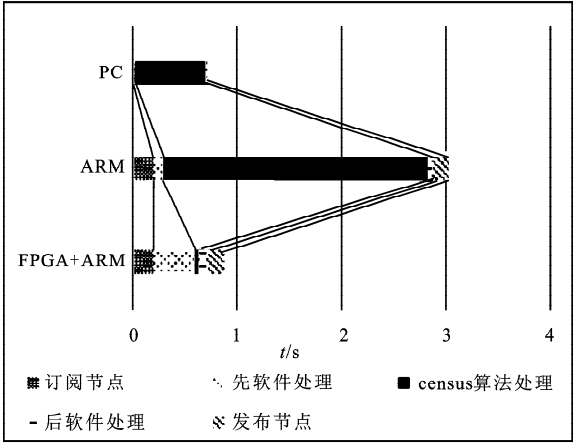


图 9 ROS 实现 census 算法各阶段时间

从图 9 可以看出,基于软硬件协同实现的 ROS 节点处理一幅图片总耗费时间为 0.88 s,而在 PC 上实现相同算法需要 0.72 s,单纯使用 ARM 实现需要 3.02 s,在 ARM 和 PC 等纯软件实现中,算法处理耗时最多,而在软硬件协同中,算法处理时间耗时最少,最多的是先软件处理时间,即从 PS 端到 PL 端的双目图像传输时间,这部分基于映射处理,需在 ARM 端顺序执行,耗时较久. 在 ROS 节点间的通讯时间上,软硬件协同和 ARM 端用时相同,而在 PC 上用时

很少。总体来看,基于软硬件协同的 ROS 节点实现要比在 ARM 上实现更有时间上的优势。

6 结束语

从以上实验可以看出,基于 Zynq 软硬件协同的 ROS 节点实现在用户编程模式上和纯软件完全相同,但经过 FPGA 硬件加速之后,大大缩减了 ROS 中复杂图像算法的运行时间,更有利于智能机器人在嵌入式领域的发展。

目前在 Zynq 上 PS 端到 PL 端的通讯耗时较长,如何减少这一块的通讯开销并将软硬件协同实现移植到其他异构芯片上是进一步需要研究的方向。

参考文献:

- [1] Barros, Taisir Tadeu Teixeira. A mobile manipulator controller implemented in the robot operating system [R]. Proceedings for the Joint Conference of ISR, 2014: 121-128.
- [2] WIKI. Open Source Robotics Foundation[EB/OL]. [2016-10-09]. <http://wiki.ros.org/>.
- [3] Yamashina, Kazushi, Ootsu K, et al. Proposal of ROS-compliant FPGA component for low-power robotic systems[C]//International Conference on Intelligent Earth Observing & Applications. IEEE, 2015.
- [4] 丁帅帅,柴志雷. 基于 HLS 的 SURF 特征提取硬件加速单元设计与实现[J]. 微电子学与计算机, 2015, 32(9): 133-137.

- [5] Martin G, Smith G. High-level synthesis: past, present, and future[J]. IEEE Design and Test of Computers, 2009, 26(4): 18-25.
- [6] 何宾. Xilinx FPGA 设计权威指南[M]. 北京:清华大学出版社, 2014.
- [7] Fire W S, Archibald J K. Improved census transforms for resource-optimized stereo vision[J]. IEEE Transaction on Circuits and System for Video Technology, 2013, 23(1): 60-73.
- [8] Heinly J, Dunn E, Frahm J M. Comparative evaluation of binary features[C]//12th European Conference on Computer Vision, Part II. IEEE, 2012.
- [9] Nakazato, Kaichiro. FPGA-based stereo vision system using census transform for autonomous mobile robot [C]// International Conference on Informatics, Electronics and Vision. IEEE, 2015.
- [10] 周龙,徐贵力,李开宇,等. 基于 Census 变换和改进自适应窗口的立体匹配算法[J]. 航空学报, 2012, 33(5): 886-892.

作者简介:

赵可可 男, (1990-), 硕士研究生. 研究方向为嵌入式系统、FPGA 设计. E-mail: yake827@foxmail. com.

柴志雷 男, (1975-), 博士, 副教授. 研究方向为嵌入式系统设计技术、高性能视觉系统、基于 FPGA 的可重构计算和 FPGA 操作系统等.

吴东 男, (1971-), 博士, 高级工程师. 研究方向为 FPGA 高效能计算机.

(上接第 86 页)

- [8] CENTER HIRISE OPERATIONS. HiRISE images of MS-L(Curiosity)[EB/OL]. [2016-10-12]. <https://www.uahirise.org/releases/msl-images.php>.
- [9] ESA/NASA/SOHO/LASCO. Space in images[EB/OL]. [2016-10-12]. <https://www.esa.int/spacelinimages/Images>.
- [10] Mikolajczyk K, Tuytelaars T, Schmid C, Zisserman A. A Comparison of affine region detectors[J]. International Journal of Computer Vision, 2005, 65(1/2): 43-72.
- [11] 朱英红. 基于特征的红外与可见光图像匹配技术研究[D]. 西安: 第二炮兵工程大学, 2012.

作者简介:

陶江 男, (1991-), 硕士研究生. 研究方向为计算机视觉. E-mail: tjdwangyi@163. com.

曹云峰 男, (1964-), 教授, 博士生导师. 研究方向为无人机飞行控制与导航.

丁萌 男, (1981-), 博士, 副教授. 研究方向为计算机视觉、无人机导航、制与控制、红外目标检测.

庄丽葵 女, (1963-), 讲师. 研究方向为视觉导航技术.

张洲宇 男, (1992-), 硕士研究生. 研究方向为计算机视觉.

钟佩仪 女, (1994-), 硕士研究生. 研究方向为计算机视觉.