

基于 SLAM 的连续运动测量坐标转换算法研究

瞿安琪, 柴晓冬, 郑树彬, 李立明, 张磊

(上海工程技术大学 城市轨道交通学院, 上海 201620)

摘要:近年来, 视觉导航开始成为人们的研究热点; 在轨道检测中, 现有运动状态下的测量方法所采用的坐标系都是建立在测量系统自身的一种连续动态变化的坐标系, 不能与静态基准点建立明确的解析联系; 针对这一问题, 结合同步定位与地图创建的思想, 提出了一种新的检测算法; 基于 SLAM 原理, 通过双目视觉系统的坐标转换方法, 建立动态坐标系与静态基准点之间的关系, 求得旋转平移矩阵, 将后面 N 个位置的坐标均转换到初始坐标系下, 实现连续运动测量; 通过实验, 设计运动路线, 验证坐标转换算法, 并得到了建立在世界坐标系下的空间线形曲线; 实验结果表明: 转换后的坐标, x 、 y 、 z 三个方向的最大误差不超过 0.734 mm。

关键词:同步定位与地图创建; 坐标转换; 连续测量; 世界坐标系

Continuous Movement Measuring Coordinate Transformation Algorithm Based on SLAM

Zi Anqi, Chai Xiaodong, Zheng Shubin, Li Liming, Zhang Lei

(College of Urban Rail Transportation, Shanghai University of Engineering Science, Shanghai 201620, China)

Abstract: In recent years, the visual navigation began to become the focus. In rail detection, the existing sports under the state of measuring methods adopted by the coordinate system are based on the measurement system itself a continuous dynamic changes of the coordinate system, cannot establish clear resolution contact with static reference point. In order to solve this problem, a new detection algorithm is proposed combining with simultaneous localization and mapping. In this paper, a relationship between a dynamic coordinate system and the static reference point was built to realize continuous motion measurement, which was based on the principle of simultaneous localization and mapping (SLAM) and the coordinate transformation method of binocular vision system. We designed a movement course in the experiment to verify the coordinate transformation algorithm, and got a space linear curve based on the world coordinate system. Experimental result shows that after the transformed coordinates, x , y , z three directions of the maximum error is less than 0.734 mm.

Keywords: simultaneous localization and mapping; coordinate transformation; continuous motion measurement; world coordinate system

0 引言

轨道是机车车辆运行的基础, 直接影响列车行驶安全以及舒适性。由于行车安全的要求, 高速铁路、城市轨道交通线路必须具有高精度的几何线形参数, 做到高平顺性。目前, 高速无砟轨道结构中人们仍借助全站仪、GPS 精测网进行非连续测量, 这种在静止状态下才能进行测量的方法效率较低, 无法较好地保证长距离的轨道线路的高平顺性要求。而现有的运动状态下的测量方法所采用的坐标系都是建立在测量系统自身的一种连续动态变化的坐标系, 这种动态坐标系还不能与静态基准点建立明确的解析联系, 也就无法解决现实生活中的一些问题^[1-2]。

近年来, 视觉的方法越来越多的应用到轨道检测、机器人等领域。视觉导航更是开始成为人们的研究热点, 它具有获取信息完整、探索范围比较大等优点, 在机器人导航领域有着重要的研究意义。视觉导航系统中, 目前应用最多的是在移动机器人上面安装一个或若干个视觉传感器, 通过控制移动机器人, 拍摄周围环境的局部图形, 然后通过特征提取、图像匹配、环境建模等^[3], 创建地图实现定位, 即利用了 SLAM 的

基本原理。本文将 SLAM 原理与连续运动测量结合起来, 在 SLAM 原理下, 通过双目视觉系统的坐标变换方法, 将动态测量值转换到世界坐标系下, 实现非接触连续运动的测量。

1 SLAM 原理

在 SLAM 研究中, 有 3 个关键性的问题, 即: 1) Where am I? 2) Where am I going? 3) How should I get there? 要求机器人在一个完全未知的环境中从一个未知的位置出发, 利用传感器对环境的观测递增地建立环境的导航地图, 同时根据已建立的地图同步确定自己的位置^[4]。其基本原理图如图 1 所示。

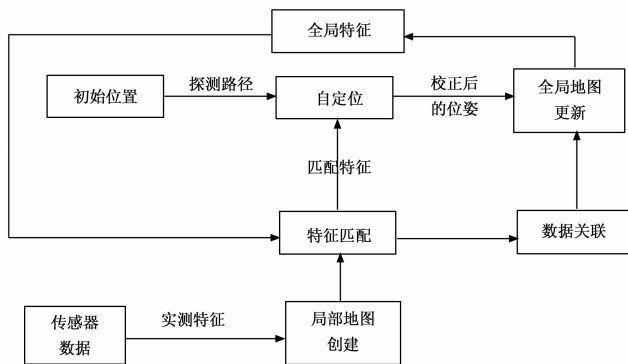


图 1 SLAM 原理图

本文将两个 CCD 相机安装在控制平台的固定位置, 通过设计好的路径连续地运动获取物体几何信息, 进行坐标转换, 获得空间线形曲线, 从而实现全局定位与地图创建。

收稿日期:2015-08-05; 修回日期:2015-09-16。

基金项目:国家自然科学基金资助(51478258); 上海市科委重点支撑项目资助(13510501300); 上海市研究生教育创新计划学位点引导布局与建设培育项目(13sc002)。

作者简介:瞿安琪(1990-), 女, 山东邹城人, 硕士研究生, 主要从事智能检测与机器视觉方向的研究。

柴晓冬(1962-), 男, 安徽人, 教授, 硕士研究生导师, 主要从事智能信息处理、光信息处理方向的研究。

2 立体视觉基本原理

1) 立体视觉介绍:

立体视觉是指从多幅拍摄的图像中获取目标物体三维几何信息的一种方法,它的目的在于重构场景的三维几何信息。立体视觉的研究主要包括图像获取、摄像机模型、特征提取、图像匹配和深度计算等部分。

双目立体视觉的方法,即从所拍摄的两幅图中获取物体三维几何结构信息,从而恢复出空间线形。双目视觉在生活中并不少见,其中最完美的双目视觉结构就是人的双眼。当用双眼观察物体时,会有远近的感觉,而使用两个相机实现立体视觉系统正是参照此原理,通过建立所拍摄的两幅图像之间的位置关系,根据对应点的视差恢复三维信息,计算深度^[5]。

2) 空间点重建:

对于一个三维多面体,如果知道了它的各个顶点的三维坐标与相邻关系,那么这个多面体的形状和位置也就是唯一确定的。利用立体视觉的方法获取物体的三维点坐标,也是立体视觉中最基本的。假设,在图 2 中空间中任意点 $\mathbf{X}$ 在两个相机上的投影点分别为 $\mathbf{X}_L$ 和 $\mathbf{X}_R$,两相机的投影矩阵分别为 $\mathbf{M}_1$ 与 $\mathbf{M}_2$,于是有

$$\mathbf{Z}_{c1} = \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = \mathbf{M}_1 \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11}^1 & m_{12}^1 & m_{13}^1 & m_{14}^1 \\ m_{21}^1 & m_{22}^1 & m_{23}^1 & m_{24}^1 \\ m_{31}^1 & m_{32}^1 & m_{33}^1 & m_{34}^1 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \\ 1 \end{bmatrix}$$

其中, $(u_1, v_1, 1)$ 与 $(u_2, v_2, 1)$ 分别为点 $\mathbf{X}$ 在两个相机中对应点 $\mathbf{X}_L$ 和 $\mathbf{X}_R$ 的齐次坐标; $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, 1)$ 为点 $\mathbf{X}$ 在世界坐标系下的齐次坐标。联立上式,消去可得:

$$(u_1 m_{31}^1 - m_{11}^1) X + (u_1 m_{32}^1 - m_{12}^1) Y + (u_1 m_{33}^1 - m_{13}^1) Z = m_{14}^1 - u_1 m_{34}^1$$

$$(v_1 m_{31}^1 - m_{21}^1) X + (v_1 m_{32}^1 - m_{22}^1) Y + (v_1 m_{33}^1 - m_{23}^1) Z = m_{24}^1 - v_1 m_{34}^1$$

$$(u_2 m_{31}^2 - m_{11}^2) X + (u_2 m_{32}^2 - m_{12}^2) Y + (u_2 m_{33}^2 - m_{13}^2) Z = m_{14}^2 - u_2 m_{34}^2$$

$$(v_2 m_{31}^2 - m_{21}^2) X + (v_2 m_{32}^2 - m_{22}^2) Y + (v_2 m_{33}^2 - m_{23}^2) Z = m_{24}^2 - v_2 m_{34}^2$$

由解析几何的知识可知,直线 $O_L X_L$ 与直线 $O_R X_R$ 一定相交,即上述方程组一定有解,而且解是唯一的^[6-7]。由此就可以确定空间中点的三维信息,即实现自定位。

3 坐标转换算法原理

连续运动测量基本原理框图如图 2 所示。

以左摄像机为例,初始化位置记为 P_1 ,将标定板固定,并在标定板上建立绝对世界坐标系,则标定板和摄像机之间的旋转矩阵和平移向量 $[R_0; t_0]$ 可以求得。

1) 前后位置关系:

记前后位置公共特征点在位置 P_1 的坐标为 (x_{1i}, y_{1i}, z_{1i}) ,在位置 P_2 下的坐标为 (x_{2i}, y_{2i}, z_{2i}) ,则 P_1 和 P_2 之间的位置关系可以表示为:

$$\begin{bmatrix} x_{1i} \\ y_{1i} \\ z_{1i} \\ 1 \end{bmatrix} = R_1 \begin{bmatrix} x_{2i} \\ y_{2i} \\ z_{2i} \\ 1 \end{bmatrix} + t_1$$

$$\text{其中旋转矩阵 } \mathbf{R}_1 = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \text{ 平移矩阵 } \mathbf{t}_1 = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}。$$

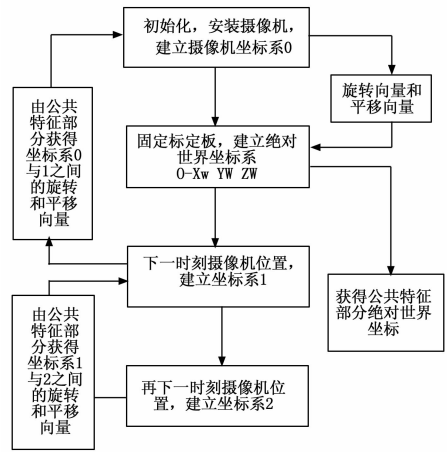


图 2 测量原理框图

$$\text{记 } \mathbf{A} = \begin{bmatrix} x_{z2} & y_{z2} & z_{z2} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & x_{z2} & y_{z2} & z_{z2} & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_{z2} & y_{z2} & z_{z2} & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{z2} & y_{z2} & z_{z2} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & x_{z2} & y_{z2} & z_{z2} & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_{z2} & y_{z2} & z_{z2} & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{X} = \begin{bmatrix} r_{11} \\ r_{12} \\ r_{13} \\ r_{z1} \\ r_{z2} \\ r_{z3} \\ r_{z1} \\ r_{z2} \\ r_{z3} \\ r_1 \\ r_2 \\ r_3 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} x_{z1} \\ y_{z2} \\ z_{z3} \\ \vdots \\ x_{1n} \\ y_{1n} \\ z_{1n} \end{bmatrix}$$

构造伪逆矩阵 $(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{A} \cdot \mathbf{A} \mathbf{X} = \mathbf{b}$, 则最小二乘解为 $\mathbf{X} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$ 。

即可求得,位置 P_1 和位置 P_2 的之间的 $[R_1; t_1]$ 。

以此类推,可以求得连续运动状态下所有位置之间的 R, t 。

2) 递推关系:

通过连续测量,可以获得每个位置上数据点的坐标,而又求得了前后位置之间的关系。这样就可以将所有位置上的坐标转换到初始位置 P_1 的世界坐标系下,从而反映出整个变化的情况。最后一个位置上面的点在世界坐标系下的坐标可以通过前面位置的旋转和平移递推公式求得。即可以通过位置 1 在相机坐标系下的坐标和相机与世界坐标系之间的旋转、平移矩阵获得位置 1 上的点在世界坐标系下的坐标。即:

$$\begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix} = [R_0; t_0] \cdot \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} \quad (1)$$

其中: $(x_1, y_1, z_1, 1)$ 为 P_1 位置的相机所拍摄的坐标,

$(X_0, Y_0, Z_0, 1)$ 为其在绝对世界坐标上点的坐标。

同理, P_1 和 P_2 之间的关系为:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} = [R_1; t_1] \cdot \begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix} \quad (2)$$

由此递推可得最后一个位置上面的点在世界坐标系下的坐标:

$$\begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix} = [R_0; t_0][R_1; t_1] \cdots [R_{n-1}; t_{n-1}] \begin{bmatrix} x_n \\ y_n \\ z_n \\ 1 \end{bmatrix} \quad (3)$$

4.4 算法验证与分析

实验采用如图 3 所示的控制平台进行验证。

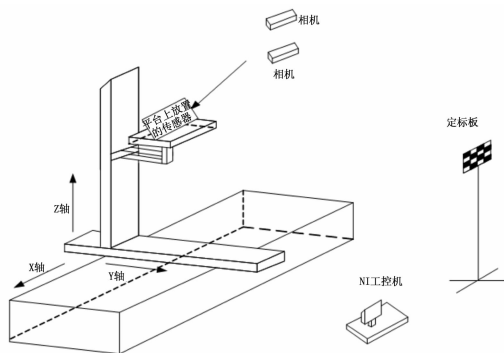


图 3 实验平台示意图

如上图所示,整个实验平台由三部分组成:NI 工控机、三维移动平台、检测平台。实验平台中 X 向位于最底端,可以左右移动;Y 向支座设置在 X 向底座上,可以前后移动;而 Z 向支架位于 Y 向支架的上表面,可以上下移动。3 个方向移动的最小单位均为 0.05 mm,重复精度为 0.02~0.04 mm。在检测平台上固定着两个相同型号的 CCD 相机;相机正前方放置与之垂直的定标板^[8]。

具体实验步骤如下：

1) 为了得到高精度的旋转矩阵, 在左右相机都能获取完整标定板信息的前提下, 尽量多的改变标定板的姿态, 同步采集图像 20 次进行定标, 获得相机的 $\mathbf{M}$ 矩阵;

2) 在初始位置建立世界坐标系, 确定相机坐标系与世界坐标系之间的位置关系;

3) 按照设计的路线移动控制平台, 拍摄图像, 提取当前



图 4 实验平台

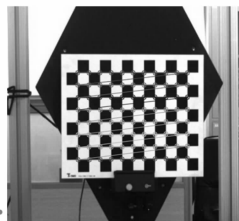


图 5 CCD 相机标定界面

位置与上一位置的公共特征点，确定前后位置的旋转矩阵和平移矩阵，对特征点进行匹配；

4) 继续移动, 重复步骤 3), 获取所有点的三维坐标;

5) 将后面所有位置特征点的坐标转换到初始位置的摄像机坐标系下, 然后再转到世界坐标系下, 从而获得空间线形曲线。

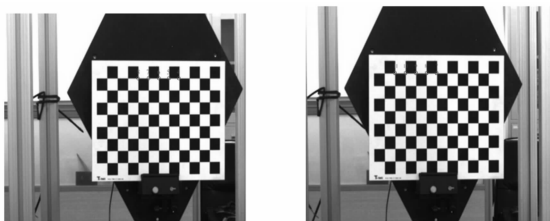


图 6 前后位置特征点提取对比图

表 1 所示为移动到位置 1、位置 2 和位置 3 时所取点转换到初始位置的坐标, 由转换后点的坐标可知, 三轴的最大误差不超过 0.734 mm。

坐标本实验中得到的空间线形曲线如图 7 所示, x 轴向绝对差为 7.104 mm, y 轴向绝对差为 105.549 mm, z 轴向绝对差

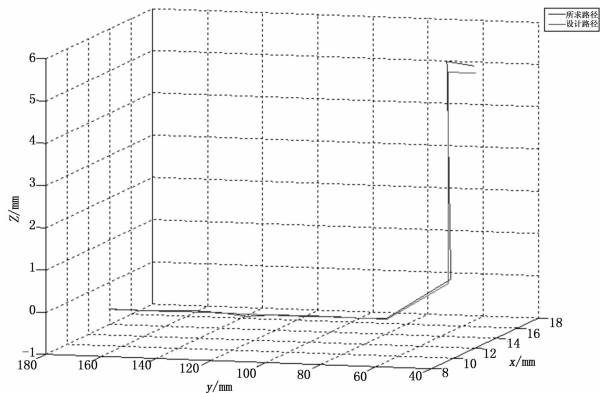


图 7 获得的空间线形曲线

表 1 位置 1、2 上的点转换到初始位置下的坐标

点号	1	2	3	4	5	6	7	8
坐标	$\begin{bmatrix} 99.50233 \\ -41.62987 \\ 822.21321 \end{bmatrix}$	$\begin{bmatrix} 89.67947 \\ -41.49898 \\ 819.79879 \end{bmatrix}$	$\begin{bmatrix} 79.76407 \\ -41.29574 \\ 819.58482 \end{bmatrix}$	$\begin{bmatrix} 69.83310 \\ -41.26779 \\ 817.15907 \\ 69.92934 \\ -41.31517 \\ 817.49254 \end{bmatrix}$	$\begin{bmatrix} 60.08264 \\ -41.29006 \\ 816.10782 \\ 59.79792 \\ -41.36494 \\ 816.37154 \end{bmatrix}$	$\begin{bmatrix} 49.86938 \\ -41.29062 \\ 816.20929 \end{bmatrix}$	$\begin{bmatrix} 39.92954 \\ -41.22923 \\ 815.25984 \end{bmatrix}$	$\begin{bmatrix} 30.08574 \\ -41.04571 \\ 812.30183 \end{bmatrix}$
点号	7	8	9	10	11	12		
坐标	$\begin{bmatrix} 40.40898 \\ -41.05182 \\ 814.52661 \end{bmatrix}$	$\begin{bmatrix} 30.05706 \\ -41.12708 \\ 812.14222 \end{bmatrix}$	$\begin{bmatrix} 20.12307 \\ -41.16644 \\ 812.31834 \end{bmatrix}$	$\begin{bmatrix} 10.35778 \\ -40.83537 \\ 807.82512 \\ 10.01477 \\ -40.69145 \\ 807.26611 \end{bmatrix}$	$\begin{bmatrix} 0.59748 \\ -40.97581 \\ 809.36656 \\ 0.61204 \\ -40.90625 \\ 808.92763 \end{bmatrix}$	$\begin{bmatrix} -9.90029 \\ -41.09266 \\ 809.71264 \end{bmatrix}$		

基于 MPI 的卷积计算并行实现

鲁金, 马可, 高剑

(西安电子工程研究所, 西安 710100)

摘要: 针对传统的卷积并行计算模型中, 存在着大量的消息传递, 负载均衡等问题; 提出一种新的基于 MPI 同步模型的并行卷积算法; 该模型采用消息传递的方式进行进程间的通信, 同时有效平衡负载, 避免大量的消息传递; 通过分析该模型的加速比和效率, 实验结果表明, 此方法显著提高了并行效率和长序列的运算速度, 充分发挥了节点间分布式存储和多核并行处理的优势, 是一种有效可行的并行策略。

关键词: 卷积计算; 并行; 消息传递接口; 负载均衡

Implementation of Parallel Convolution Based on MPI

Lu Jin, Ma Ke, Gao Jian

(Xi'an Electronic Engineering Research Institute, Xi'an 710100, China)

Abstract: Convolution computing plays an important role in scientific computing. There exist massive message passing, load-unbalancing in traditional MPI model. Concerning this question, a new parallel convolution based on MPI model was proposed in this paper. This model prevents massive message passing, and keeps load-balancing. By analyzing the speedup ratio and efficiency, the experimental result shows that the new MPI parallel programming obviously promotes parallel efficiency and large-scale sequence's speed, which exerts the advantages of distributional memory between nodes and parallel computing. It is an effective and feasible parallel strategy.

Keywords: convolution computing; parallel; MPI; load-balancing

0 引言

近年来, 受到程工艺的限制, 单核处理器的性能已接近极限, 通过提高单处理器的时钟频率来提高计算机性能的方法越来越难以达到良好的效果。因此, 多核技术成为 CPU 制造商显著提高处理器的性能的共同解决方案^[1-2]。多核设备为应用程序提供了并行计算的硬件平台, 使计算机的计算速度得到了巨大提高。而另一方面, 在数字信号处理领域中, 要处理的数据量越来越大, 实时性的要求也越来越高, 将并行处理的思想运用在信号处理中, 构建一个具有并行性的系统, 必将是一个未来发展的趋势。

卷积计算作为信号与系统时域分析的一种重要方法。在科学计算领域中起着重要的作用^[3], 广泛应用于通信, 航空航天, 生物医学工程, 雷达信号处理等工程领域。因此, 将并行化的思想应用在卷积计算中, 从而大大加快卷积计算的速度, 提高卷积计算的效率, 具有重要的研究意义。

1 MPI 并行

并行计算是指在并行系统上, 将一个大的任务分解为多个小的子任务, 分配给不同计算单元上, 各个计算单元之间相互协同, 并行地执行各个子任务, 最后汇合同步, 从而达到加速求解任务的目的。

消息传递接口 MPI^[4] (message passing interface, MPI) 是目前最流行的并行编程模型。它并不是一种新的编程语言, 而是一个可以被 C、C++ 和 Fortran 等程序调用的函数库。不仅具有易移植、高效率等多种优点, 而且可以广泛应用于分布式系统中。在 MPI 程序中, 运行在一个核上的程序称为一

个进程。两个进程可以通过调用函数来进行通信: 一个进程调用发送函数, 另一个调用接收函数。将消息传递的实现称为消息传递接口, 即 MPI。

MPI 有多种免费的、高效的实现版本, 包括 MPICH, LAM 和 MPI/Pro 等, 其中 MPICH 是目前最主流的一种版本, 大多并行计算机厂商都提供对它的支持。

2 卷积计算的并行模型

假设 $x_{(n)}$ 是长度为 N_1 的序列, $h_{(n)}$ 是长度为 N_2 的序列, 则由 $x_{(n)}$ 和 $h_{(n)}$ 卷积所得的结果 $y_{(n)}$ 可表示为:

$$y(n) = x(n) * h(n) = \sum_{m=-\infty}^{+\infty} x(m)y(n-m) = \sum_{m=low}^{to p} x(m)y(n-m) \begin{cases} to p = \min(n, N_1 - 1) \\ low = \max(0, n + 1 - N_2) \end{cases} \quad (1)$$

为了能直观地理解卷积公式, 如图 1 所示, 固定 $h_{(n)}$ 序列不动, 将 $x_{(n)}$ 序列以 $x_{(0)}$ 作对折, $y_{(0)}$ 等于 $h_{(0)}$ 和 $x_{(0)}$ 的乘积, 然后将 $x_{(n)}$ 序列往右平移一位, 再作乘积相加, 对应 $y_{(1)}$, 并以此类推, 求的 $y_{(2)}, y_{(3)}, y_{(4)}, \dots$ 。

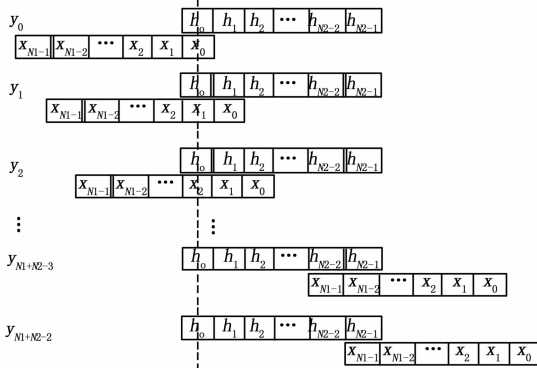


图 1 卷积计算

收稿日期: 2015-08-05; 修回日期: 2015-09-11。

项目基金: 武器装备预先研究资助项目(40405040301)。

作者简介: 鲁金(1988-), 男, 浙江杭州人, 硕士, 工程师, 主要从事并行计算在现代雷达中的应用方向的研究。

2.1 传统基于 $y_{(n)}$ 分配的 MPI 模型

传统的基于 MPI 卷积计算的并行方法是根据 $y_{(n)}$ 来分配任务, 如图 2 所示, 基本思想如下:

1) 把 $x_{(n)}$, $h_{(n)}$ 分别广播给 N 个进程, 主进程申请 $N1 + N2 - 1$ 个存储空间, 用以存放计算结果 $y_{(n)}$;

2) 为每个进程申请 $\text{ceil}((N1 + N2 - 1) / N)$ 个存储空间 temp (ceil 为向上取整);

3) 以循环分配的方式划分每个进程的任务 $y_{(k)}$ 如图 1 所示, 第 k 个处理器只计算 $y_{(k)}$, $y_{(k+N)}$, $y_{(k+2N)}$, $y_{(k+3N)}$..., 并通过如上计算公式将计算结果按顺序放在该进程申请的存储空间 temp 中;

4) 通过 MPI_Barrier 同步;

5) 通过循环 $(N1 + N2 - 1) / N$ 次, 利用 MPI_Gather^[5-6] 依次把 temp 中的数据搬运到主进程 y 中。

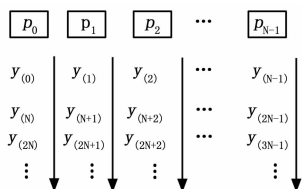


图 2 每个进程的任务分配

虽然传统的卷积并行计算相对于普通的串行计算在一定情况下的速度有一定提升, 但也存在一些不足: 1. 负载不均衡, 每个 $y_{(n)}$ 的计算量不同, 如 $y_{(0)}$ 只需一次乘法计算, $y_{(1)}$ 需要两次乘法和一次加法计算, 而 $y_{(3)}$ 需要 3 次乘法和两次加法计算, 这会增加单个进程的空闲时间; 2. 进程间通信量大, 进程间的通信严重影响执行的效率^[7]。整个计算过程中, 除了初始化时的广播外, 还有 $(N1 + N2 - 1) / N$ 次的聚集 (gather) 操作, 花费了大量的额外开销。

2.2 基于矩阵分割的 MPI 模型

为了解决上述传统卷积并行计算存在的一些不足, 本文提出一种基于矩阵分割的 MPI 模型。为了便于分析, 我们建立一个由 $N1$ 个 $x_{(n)}$ 序列和 $N2$ 个 $h_{(n)}$ 序列组成的 $N1 * N2$ 矩阵模型, 如图 3: 可以直观的看出, 卷积结果 $y_{(k)}$ 正好等于第 k 条斜对角线的所有元素相加。这样, 就把求卷积的过程转化为求斜对角线上的元素, 并相加。

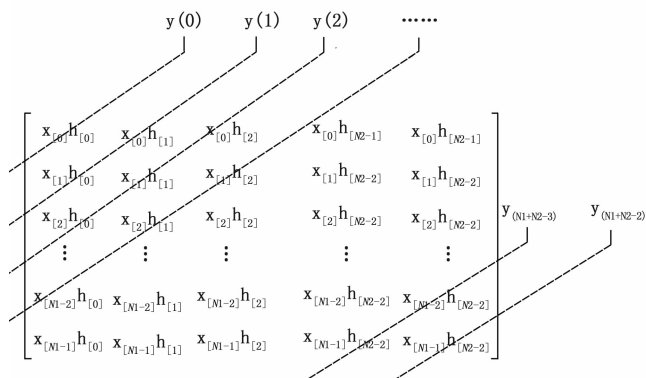


图 3 矩阵分割

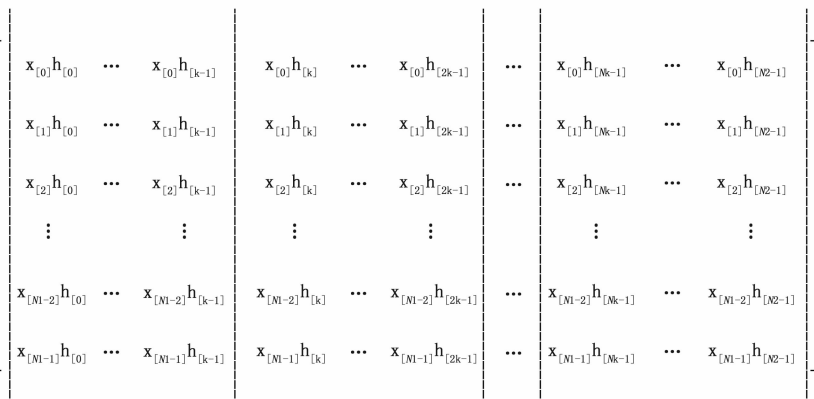


图 4 矩阵分割

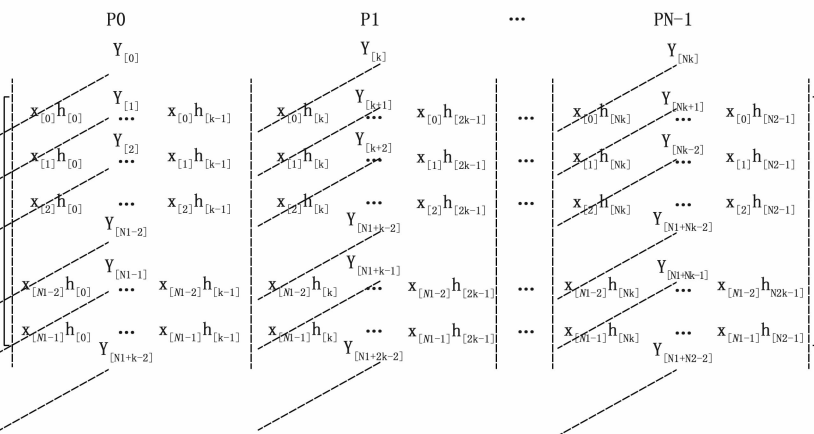


图 5 矩阵分割

以并行方式在求矩阵斜对角线元素的和时, 本文按图 4 方式划分并行任务, 即按列将整个 $N1 * N2$ 矩阵等分成 N 个子块 (N 为进程的个数), 不能等分的向上取整 (如: 列数 $N2 = 100$, 进程数 $N = 8$, 则前 7 个子块每块连续分配 13 列, 第 8 块分配最后 9 列)。然后将每一块分配给 N 个进程, 每个进程求解对应子块的斜对角线元素的和, 如图 5 所示。最后通过集合通信的方式将每个子块已经求的结果作归约^[8]计算。从而就求出了整个矩阵斜对角线元素的和, 即卷积的最终结果。整个计算流程如图 6 所示, 具体步骤如下:

- 1) 将 $x_{(n)}$ 广播给 N 个进程;
- 2) 将 $h_{(n)}$ 按块等分成 N 块 (不能等分的向上取整), 再将每一块依次广播给 N 个进程;
- 3) 为每个进程申请 $N1 + N2 - 1$ 个存储空间 y , 并初始化为 0;
- 4) 每个进程计算对应的子块中斜对角线元素的和, 并赋值给对应的 $y_{(n)}$, 如图 3 所示。

5) 通过 MPI_Reduce 将每个进程的 $y_{(n)}$ 相加, 最后的结果就是 $x_{(n)}$ 和 $h_{(n)}$ 的卷积和 $y_{(n)}$ 。

可以看到, 在整个计算过程中, 每个进程的独立性较强, 仅在最后作数据归约计算的时候又一次集合通信, 大大降低了进程间的通信量。另外, 当 $N2$ 能被 N 整除时, 每个进程的任务负载完全相同。这也大大减少了进程为了同步而等待所消耗的开销。

3 实验性能分析

为了测试这种编程模型的性能, 将其对串行模型、传统的

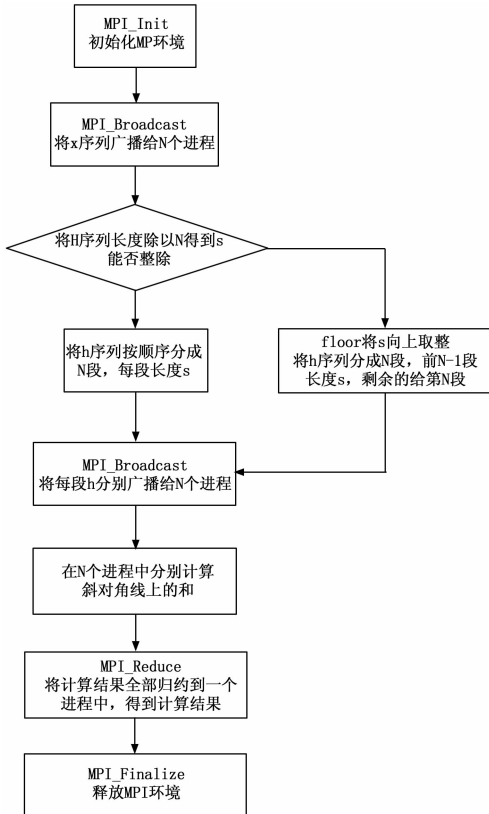


图 6 并行卷积计算流程图

MPI 并行模型进行模拟数值实验, 实验环境是: CPU: intel i7 @3.4 GHz; 内存 4.00 GB; 操作系统: linux Fedora 17; MPI 函数库: MPICH2。在固定处理器数目的条件下 ($P=8$) 选取不同的序列长度作等长卷积, 对本文基于矩阵分割的 MPI 算法和传统基于 $y(n)$ 分配的 MPI 算法、普通串行算法在执行时间方面进行了实验测试, 运行时间如图 7 所示。

由图 6 的实验数据可知, 随着卷积序列长度的增大, 尤其当序列长度大于 1024 点以后, 基于矩阵分配的 MPI 算法有明显的效率提升, 充分证明该模型具有很高的可行性。

4 结束语

通过对不同长度作卷积计算的测试, 实验表明, 本文提出的基于矩阵分割的 MPI 算法具有更高的执行效率, 计算时间

(上接第 291 页)

为 5.217 mm。而实际设计的路线: x 轴的绝对位移为 6 mm; 误差为 1.104 mm; y 轴的绝对位移为 105 mm, 误差为 0.549 mm; z 轴的绝对位移为 5 mm, 误差为 0.217 mm。

5 结语

本文基于 SLAM 原理, 对连续的运动轨迹进行测量, 并通过双目视觉系统的坐标变换方法, 将动态测量值转换到世界坐标系下, 从而将动态坐标系与静态基准点建立明确的解析联系。实验证明, 本文的方法可以进行连续长距离的非接触快速检测, 具有转换精度高、重复性好等特点, 可实现连续运动的测量, 并能够有效地检测出空间线形曲线。

参考文献:

[1] 赵国堂. 高速铁路轨道几何状态测量与轨道精调 [R]. 北京: 京

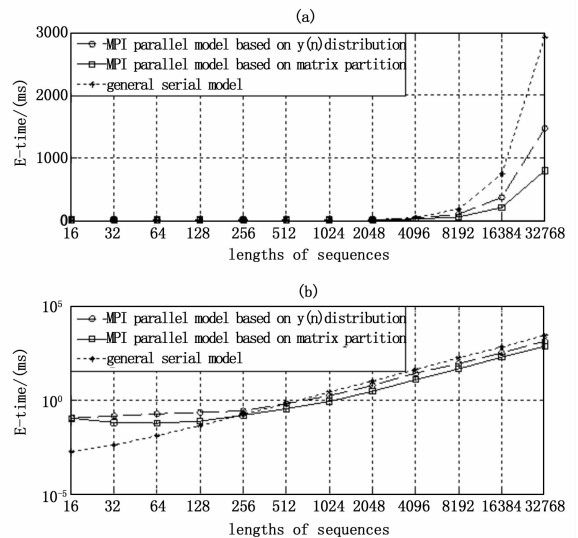


图 7 卷积计算效率对比

大大缩短, 更加充分发挥节点间分布式存储和多核并行计算的优势。在多核处理器环境下该模型可以有效提高并行计算性能, 是一种高效可行的并行编程策略。但当序列长度大到一定的值时, 在时域上作卷积并行计算效果并不明显, 这时应该考虑转化到频域中再作并行 FFT 的计算。

参考文献:

- [1] Michael J. Quinn. Parallel Programming in C with MPI and OpenMP [M]. 北京: 清华大学出版社, 2005.
- [2] Calvin Lin, Lawrence Snyder. 并行程序设计原理 [M]. 北京: 机械工业出版社, 2009.
- [3] 章隆兵, 吴少刚, 蔡飞. PC 机群上共享存储与消息传递的比较 [J]. 软件学报, 2004 (6): 842-849.
- [4] Peter S. Pacheco. 并行程序设计导论 [M]. 北京: 机械工业出版社, 2013.
- [5] Faye Gebali, Mahafza. 算法与并行计算 [M]. 北京: 清华大学出版社, 2012.
- [6] Snir M, Otto S, HussLederman S, Walker D, Songarra J, MPI: The Complete Reference, [M]. London: MIT Press, 1996.
- [7] 曹振南. 高性能计算的性能评测与性能优化 [D]. 北京: 北京科技大学, 2003.
- [8] 钟光清, 郑灵翔. 一种基于循环并行模式的多核优化方法 [J], 厦门大学学报 (自然科学版), 2010 (6): 789-792.
- [9] 沪高速铁路股份有限公司, 2010.
- [10] Shen S L, Wub H N, Cui Y J, et al. Long-term settlement behaviour of metro tunnels in the soft deposits of Shanghai [J]. Tunneling and Underground Space Technology, 2014, 40: 309-323.
- [11] 韩锐. 未知环境下基于 SLAM 的移动机器人导航算法研究 [D]. 武汉: 武汉理工大学, 2006.
- [12] 郑宏. 移动机器人导航和 SLAM 系统研究 [D]. 上海: 上海交通大学, 2007.
- [13] 吴福朝. 计算机视觉中的数学方法 [M]. 北京: 科学出版社, 2008.
- [14] Zhang Z Y, Membe S. A flexible new technique for camera calibration [J]. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2000, 22 (11): 1330-1334.
- [15] 马颂德, 张正友. 计算机视觉 [M]. 北京: 科学出版社, 1997.
- [16] 罗伟. 多传感信息融合的轨道长波不平顺检测实验平台研究 [D]. 上海工程技术大学, 2015.