



### “祝融号”火星车立体视觉算法并行设计与实现

毛晓艳, 苗志富, 陈建新, 李志平, 滕宝毅, 邢 琰

#### Parallel Design and Implementation of Stereo Vision Algorithm of Zhurong Mars Rover

MAO Xiaoyan, MIAO Zhifu, CHEN Jianxin, LI Zhiping, TENG Baoyi, and XING Yan

在线阅读 View online: <https://doi.org/10.15982/j.issn.2096-9287.2022.20210101>

### 您可能感兴趣的其他文章

#### Articles you may be interested in

#### “祝融号”火星车图像压缩软件的设计与实现

Image Compression Software Design for Zhurong Mars Exploration Rover

深空探测学报(中英文) . 2021, 8(5): 503-510

#### “祝融号”火星车遥操作技术

Teleoperation Technology of Zhurong Mars Rover

深空探测学报(中英文) . 2021, 8(6): 582-591

#### 基于双目视觉的非合作目标自主姿态估计方法

Autonomous Attitude Estimation Method for Non-Cooperative Target Based on Stereo VISION

深空探测学报(中英文) . 2019, 6(4): 341-347

#### 火星车图像压缩存储系统与关键算法设计

Design of Image Compression Storage System and Key Algorithm for Mars Rover

深空探测学报(中英文) . 2020, 7(5): 458-465

#### 火星车机构集成控制系统设计与实现

Design and Implementation of Integrated Control System for Mars Rover Mechanism

深空探测学报(中英文) . 2020, 7(5): 450-457

#### 火星车系统动力学建模与仿真分析

Dynamics Modeling and Simulation Analysis of Mars Rover System

深空探测学报(中英文) . 2020, 7(5): 437-444



关注微信公众号, 获得更多资讯信息

# “祝融号”火星车立体视觉算法并行设计与实现

毛晓艳<sup>1,2</sup>, 苗志富<sup>1</sup>, 陈建新<sup>1</sup>, 李志平<sup>1</sup>, 滕宝毅<sup>1,2</sup>, 邢 琰<sup>1,2</sup>

(1. 北京控制工程研究所, 北京 100190; 2. 空间智能控制技术重点实验室, 北京100190)

**摘要:** 针对地外天体的巡视探测任务, 环境感知计算量大影响行走效率的问题, 提出了立体视觉算法及并行优化设计与硬件协同来提高效率的方案。通过前向模型一次查表、省略冗余计算、三维点云生成与快速滤波等并行优化算法, 立体视觉的计算量显著减少。再通过硬件的并行实现, 感知效率比“玉兔二号”提升了8倍。“祝融号”火星车采用该算法自主避障, 完成了安全高效的1 km行走距离, 飞行在轨验证结果表明该设计在地外天体未知环境和资源约束条件下快速有效, 可作为后续深空任务的基石。

**关键词:** 祝融号; 立体视觉; 并行设计; 前向模型; 冗余计算

**中图分类号:** V448.2

**文献标识码:** A

**文章编号:** 2096-9287(2022)02-0202-09

**DOI:**10.15982/j.issn.2096-9287.2022.20210101

**引用格式:** 毛晓艳, 苗志富, 陈建新, 等. “祝融号”火星车立体视觉算法并行设计与实现[J]. 深空探测学报(中英文), 2022, 9(2): 202-210.

**Reference format:** MAO X Y, MIAO Z F, CHEN J X, et al. Parallel design and implementation of stereo vision algorithm of Zhurong Mars rover[J]. Journal of Deep Space Exploration, 2022, 9(2): 202-210.

## 引言

火星与地球最近的距离约5 500万km, 通讯时延有十几分钟, 测控站无法通过地面监测和遥操作模式实现实时的信息交互, 这需要火星车具有较强的自主性, 自主避开障碍实现安全行走。

目前在火面成功着陆并进行巡视的火星车只有美国和中国。美国从1996年开始, 陆续发射了火星着陆巡视器, 迄今为止已有4代。最早的“索杰纳”(Sojourner)采用了激光条纹仪的探测方式。从“火星探测漫游者”(Mars Exploration Rover, MER)项目, 火星车开始采用双目感知的自主避障方式, 其采用20 MHz的RAD6000处理器, 处理时间2~3 min<sup>[1-2]</sup>。“好奇号”(Curiosity)火星车在继承MER的基础上, 升级为RAD750的233 MHz处理器, 提高了自主行走的速度, 行走速度约20 m/h<sup>[3]</sup>。2021年2月着陆的“毅力号”(Perseverance)火星车是采用最新技术的巡视器, 它配备了现场可编程门阵列(Field-Programmable Gate Array, FPGA)的专用计算机来实现自主导航, 可达120 m/h的行走速度, 是“好奇号”的5~6倍<sup>[4]</sup>。

“天问一号”探测器于2021年5月15日成功着陆于火星乌托邦平原, “祝融号”火星车于5月22日驶离着陆平

台, 开始了火星漫漫之旅。“祝融号”也采用了双目感知的方式自主避障行走。而环境感知涉及复杂的图像计算, 大运算量影响其行走效率。行走效率的提升一方面来源于中央处理器(Central Processing Unit, CPU)计算能力的提高; 另一方面来源于算法的优化设计缩减复杂度。这种感知方式在国内也有很多的研究<sup>[5]</sup>, 但地面所获得的处理效率和星上不可同日而语, 高处理效率往往掩饰了对算法的优化需求。由于国外技术的封锁以及航天产品的高可靠性要求, “祝融号”不能一味选择目前处理速度最快的芯片。

本文详细地介绍了“祝融号”立体视觉算法的优化并行设计。在可获取性和高可靠性的权衡下, 考虑工程实际情况, 实现了从算法优化和硬件协同设计来提升效率。

## 1 立体视觉算法

考虑到火星表面崎岖不平的地理因素, 为实现在未知环境下自主漫游, 火星车采用光学相机的被动立体视觉成像算法对未知环境进行三维重建, 作为自主规划的前提条件。

立体视觉技术兴起于20世纪60年代, 麻省理工学院成功对三维积木进行重建之后, 立体视觉走进了人

们的视野。Huffman、Waltz等又将二维图像处理扩展到三维，解决了阴影等难题，立体视觉研究从此起步。双目立体视觉测量的基本原理如图1所示<sup>[6]</sup>。其中： $O_L$ 为双目中左相机的光心， $O_R$ 为右相机的光心。ImageL为左相机的成像平面，ImageR为右相机的成像平面， $W$ 为物方点， $m_1$ 为 $W$ 点在左相机成像平面所成像点， $m_2$ 为 $W$ 点在右相机成像平面所成像点。通过拍摄的左右相机图像，找到 $m_1$ 和 $m_2$ 的对应关系，即可确定空间中的两条直线，其交点坐标即为 $W$ 在相机系下的坐标值。

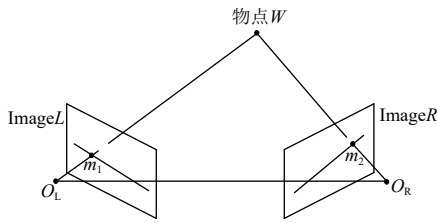


图1 双目立体视觉测量的基本原理

Fig. 1 Basic principles of stereo vision measurement

双目立体视觉技术的一般思路：通过图像内参数校正、极线校正、滤波、稠密匹配和三维计算，完成所拍摄环境的三维重建，给自主避障规划进行安全路径选取。

立体视觉的匹配方法是重难点，是保证输出效果的本质原因，也是影响效率的关键一步。目前，从种类上来说匹配算法可以分为局部、全局及半全局算法<sup>[7]</sup>。局部匹配采用基于窗口的代价聚合算法；全局算法不做代价聚合，通过优化一个包含局部数据项和平滑项的全局能量函数估计图像之间的相关性；半全局算法将二维优化问题转换到一维，采用动态规划沿着多个方向进行一维能量最小化来代替二维全局能量的最小化。近些年人工智能的兴起，又涌现出了一批基于深度学习的人工智能立体匹配方法，如HsMnet方法<sup>[8]</sup>，采用具有分层结构的神经网络模型，通过训练完成视差的恢复。LiStereo<sup>[9]</sup>方法以自我监督的方式训练模型，从而减少了必须收集大量基准真实标签的代价。

经过半个多世纪的发展，现阶段立体视觉技术已较为成熟。当前遇到的主要问题是，在面对遮挡纹理、弱纹理、深度不连续区域等还没有得到很好的解决，尤其是视觉技术受光照的影响很大，光线过暗或过亮、光照分布不均匀等情况都会影响其结果。这些都严格制约了立体技术在工程中的应用。无论理论依据与实验测量有多完美，最终都需要应用到实际工程中检验。现阶段立体视觉研究的方向都集中在对精度和效率的优化上，最大限度地保证实用性和并行化。

## 2 立体视觉工程实现方案选择

“天问一号”是中国的首次火星探测任务，受到质量功耗和计算性能等条件约束，无法选用GPU等高速并行计算的设备，只能考虑在现有计算能力上进行方案优化选择。

为保证行走安全，需要对几倍车宽的区域进行感知。为提高感知效率，实现一次感知的范围尽量宽，“祝融号”采用120°视场的鱼眼相机进行环境感知，视场宽的相机畸变很大，鱼眼相机更是具有特殊的大畸变，需进行后续的计算处理，首先要进行畸变校正，将左右图像消除鱼眼变形后再消除内外参数变形，变成极线对正的图像，才能进行左右图像的稠密匹配。鱼眼避障相机的成像模型原理如图2所示，需先进行鱼眼的标定和去畸变<sup>[10-11]</sup>。其中： $o$ 表示像平面的投影中心； $o'$ 表示主点； $P$ 表示物方点； $p$ 表示鱼眼模型表面的投射点； $p'$ 表示在鱼眼成像平面的投影坐标； $p''$ 表示 $P$ 的透视投影点； $x_c'o'y_c'$ 面是鱼眼相机的投影平面； $x_coy_c$ 面是满足透视投影关系的成像平面；相机的焦距为 $f$ ，也表示的是鱼眼模型的球面半径。

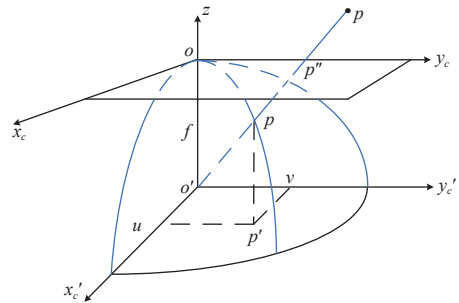


图2 鱼眼成像模型

Fig. 2 Fish-eye imaging model

考虑立体相机的左右相机成像亮度不一致的问题，除了设计相机的同步采集信号之外，在对图像进行匹配代价计算时，采用了一种左右图像灰度差异估计和灰度分布的均衡化处理方法，减小亮度差异对后续匹配的影响。

对目前的多种匹配方法进行分析，因火星环境的未知性和存储量及计算量的限定，网络训练的方法首先不推荐。全局方法的结果较优但计算量大不予选取。半全局方法综合了全局的优点效率，但是对低纹理的处理效果不佳，计算量相对还是较大。稠密匹配方法采用了模板相关匹配的差绝对值和方法（Sum of Absolute Difference, SAD），该方法的性能较折中，在计算量较小的情况下性能较优，最主要的是便于实现并行计算<sup>[12]</sup>。

$$SAD_{LR}(x, y, \delta_{LR}) = \sum_{i=-\frac{1}{2}(WinX-1)}^{\frac{1}{2}(WinX-1)} \sum_{j=-\frac{1}{2}(WinY-1)}^{\frac{1}{2}(WinY-1)} |I_L(x+i, y+j) - I_R(x+i-\delta_{LR}, y+j)| \quad (1)$$

其中:  $I_L$  表示左图像像素灰度;  $I_R$  表示右图像像素灰度, 其匹配窗口大小为  $WinX \times WinY$ ;  $\delta_{LR}$  表示搜索的视差范围。

$SAD_{LR}(x, y, \delta_{LR})$  最小的位置就是最佳匹配处, 完成匹配后, 所有的像素点都得到了视差结果, 对视差图中的噪声设计滤波算法, 去除斑块噪声, 保证其输出的可用性。视差图通过小孔成像的公式计算可得到三维坐标值并进行滤波而后输出。滤波后的三维坐标再通过适宜度的分析得到对应地形的可通过性评价, 通过规划形成自主避障行走的轨迹。

视差图通过小孔成像的公式计算可得到三维坐标值并进行滤波而后输出。滤波后的三维坐标再通过适宜度的分析得到对应地形的可通过性评价, 通过规划形成自主避障行走的轨迹。

上述的算法过程, 按照普通的流程运算, 也可以得到三维数据的输出, 但耗时很长。本文介绍的并行优化设计内容主要针对效率进行提升, 力求在受限的硬件资源下完成较高效率的立体视觉实现, 涉及鱼眼校正、稠密匹配、三维点云生成和滤波, 对这三个方面进行并行优化设计以提高立体视觉的运算效率。

### 3 并行设计算法

#### 3.1 前向模型的查表结构

“祝融号”火星车采用了视场角达到  $120^\circ$  的鱼眼相机, 其畸变大, 需进行严格的校正修订后才能进行相关匹配。传统采用的是从前向后的正向校正方法, 即取原始图像像素点, 按照畸变公式进行校正, 然后修正鱼眼变形, 投射到透视成像的平面模型上, 将该点的灰度按照浮点像素进行加权投影, 形成新的校正后图像。由于鱼眼变形在4个边角上的变形量最大, 导致该处的原始图像像素修正以后坐标分布变得很广且稀疏, 无法根据该值填补校正后的像素灰度, 在4个边角处形成空洞, 影响了后续的处理。

本文提出了基于宽视场鱼眼相机前向模型的快速图像校正方法, 采用后向前的非线性方程逆向求解实现了畸变校正后的像素与原始图像像素的对应关系<sup>[13]</sup>, 保证每个校正后的像素都有与之相对应的原始像素关系, 从而解决了宽视场相机对角极限角度下的像素空

洞问题。

正向模型的公式为

$$\mathbf{a}_c = \mathbf{a}_d + \mathfrak{J}_D(\mathbf{a}_d, \delta) \quad (2)$$

$$\mathfrak{J}_D(\mathbf{a}_d, \delta) = \begin{bmatrix} \bar{x}_d(k_1\rho_d^2 + k_2\rho_d^4 + k_3\rho_d^6) + b_1\bar{x}_d + b_2\bar{y}_d + \\ [2p_1\bar{x}_d\bar{y}_d + p_2(\rho_d^2 + 2\bar{x}_d^2)] \\ \bar{y}_d(k_1\rho_d^2 + k_2\rho_d^4 + k_3\rho_d^6) + \\ [p_1(\rho_d^2 + 2\bar{y}_d^2) + 2p_2\bar{x}_d\bar{y}_d] \end{bmatrix} \quad (3)$$

其中:  $\mathbf{a}_c = [x_c, y_c]^T$  表示无畸变的正确坐标;  $\mathbf{a}_d = [x_d, y_d]^T$  表示带畸变的坐标;  $\mathfrak{J}_D(\mathbf{a}_d, \delta)$  表示畸变带来的偏差。

$$\bar{x}_d = x_d - x_0 \quad (4)$$

$$\rho_d = \sqrt{\bar{x}_d^2 + \bar{y}_d^2} \quad (5)$$

其中:  $(x_0, y_0)$  表示中心偏量;  $(k_1, k_2, k_3)$  表示径向畸变系数;  $(p_1, p_2)$  表示偏心畸变系数;  $(b_1, b_2)$  表示像平面畸变系数<sup>[14]</sup>。

前向模型的公式则变为

$$\begin{aligned} \mathbf{a}_d &\approx \mathbf{a}_c - [\mathbf{I} + \mathbf{D}(\mathbf{a}_c)]^{-1} \mathfrak{J}_D(\mathbf{a}_c, \delta) \\ &= \mathbf{a}_c - \mathfrak{J}_D^*(\mathbf{a}_c, \delta) \end{aligned} \quad (6)$$

式 (6) 中

$$[\mathbf{I} + \mathbf{D}(\mathbf{a}_c)]^{-1} \approx \frac{1}{d_{11}(\mathbf{a}_c) + d_{22}(\mathbf{a}_c) + 1} \quad (7)$$

经过简化计算推导而得<sup>[15]</sup>

$$\mathfrak{J}_D^*(\mathbf{a}_c, \delta) = \mathfrak{J}_D(\mathbf{a}_c, \delta) \times \frac{1}{4k_1\rho_c^2 + 6k_2\rho_c^4 + 8k_3\rho_c^6 + 8p_2\bar{x}_c + 8p_1\bar{y}_c + b_1 + 1} \quad (8)$$

其中

$$\begin{cases} \bar{x}_c = x_c - x_{c0} \\ \bar{y}_c = y_c - y_{c0} \\ \rho_c = \sqrt{\bar{x}_c^2 + \bar{y}_c^2} \end{cases} \quad (9)$$

在图3中, (a) 是原始鱼眼图像, (b) 是通过正向模型得到的鱼眼校正结果图, (c) 是通过前向模型得到的鱼眼校正结果图。可以看出, 在边角处 (c) 的效果很平滑, 无空洞现象。

为实现快速校正过程, 采用查表的方法, 将鱼眼的畸变校正、内外参模型和前向模型统一解算, 直接存储为对应的坐标表格, 通过查找对应的像素, 仅进行一步灰度加权就可得到新的修正位置上的纹理信息, 从而得到极线对正图像。

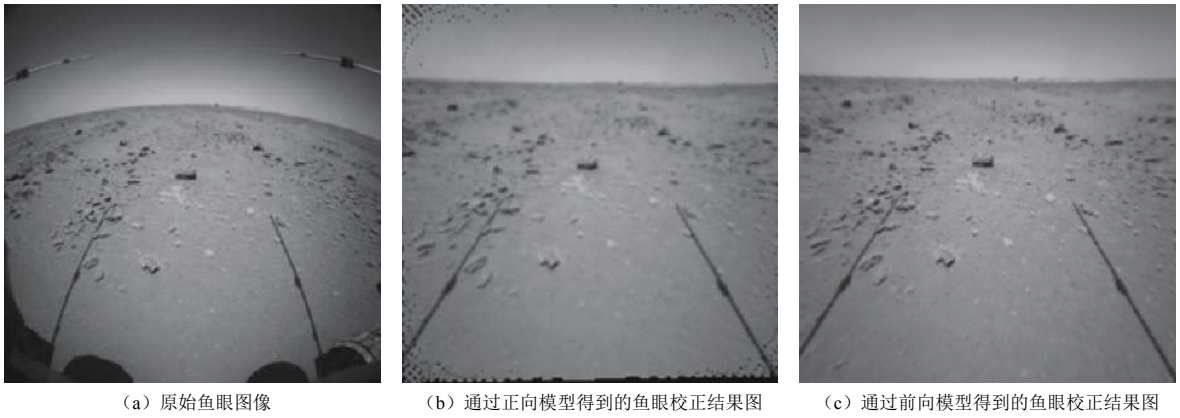


图 3 鱼眼避障相机的后向/前向校正结果

Fig. 3 Backward/forward correction results of fish-eye hazard avoidance camera

### 3.2 冗余计算省略

SAD配准方法在直接编程下可以很容易实现，但在主频不高的星载计算机直接计算需要较长的时间。为满足星上的时间计算要求，对SAD嵌套循环算法进行专门的优化设计，采用大量迭代运算消除冗余计算量，代替传统的嵌套循环。

冗余计算量如图4所示，红色的虚线框表示上一行的匹配窗口，品红色的实线框表示下一行的匹配窗口，在图像窗口向+Y方向滑动的过程中，可以看出白色区域的计算量是不变的，用上一步红色虚线框内的值加上最下面一行（灰色的）的绝对差值和，再减去最上面一行（灰色的）的绝对差值和，得到品红色窗口内的SAD值。其中，最上面和最下面一行（右图中蓝色框）的绝对差值在图像向+X方向滑动的过程中也可获得，最后每一步的新增加计算只有4个黑色角点的绝对差值相减的计算，白色和灰色区域的计算都可以通过保留上一步的值而直接得到。

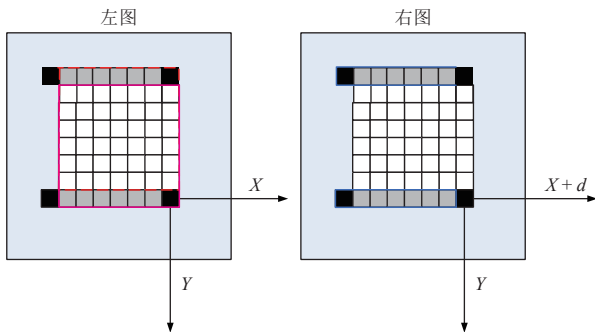


图 4 冗余计算量示意图

Fig. 4 Schematic diagram of redundant calculation

最终通过保留中间的不变量将每一步移动的SAD简化为仅计算4个黑色的边角点，消除了中间白色和灰色的大面积冗余计算。

$$SAD(x, y + 1, d) = SAD(x, y, d) + U(x, y + 1, d) \quad (10)$$

$$U(x, y + 1, d) = U(x - 1, y + 1, d) + |I_L(x + n, y + n + 1) - I_R(x + d + n, y + n + 1)| - |I_L(x - n - 1, y + n + 1) - I_R(x + d - n - 1, y + n + 1)| + |I_L(x - n - 1, y - n) - I_R(x + d - n - 1, y - n)| - |I_L(x + n, y - n) - I_R(x + d + n, y - n)| \quad (11)$$

其中： $SAD(x, y, d)$ 表示 $(x, y)$ 点像素的在视差 $d$ 处的以 $(2 \times n + 1)$ 大小窗口进行匹配的绝对差值和； $I_L$ 表示左图图像像素的灰度； $I_R$ 表示右图图像像素的灰度； $U(x, y, d)$ 表示 $(x, y)$ 点像素在视差 $d$ 处的绝对差值和的增量，可描述为4个角点的迭代，将每步新增的计算量大幅度降低，在完成这一步后，针对计算机的处理性能和其需要左右相互匹配校验的特点，对匹配算法进一步优化。

为提高匹配准确度，剔除不佳匹配点，采用右图向左图匹配来校验左图向右图匹配效果的方法，当两个结果接近时，才认为是正确的匹配点。图5显示了以左图为基准向右图匹配，与以右图为基准像向左图匹配时，像素坐标与对应SAD值的关系。在左往右匹配寻找最佳位置时，同时记录这些值，通过对应关系可以直接得到右图中点右往左匹配的SAD值，直接省略了右图往左图的匹配计算。最后比较左往右最佳位置与其对应右图中的点右往左的匹配值，两者小于设定阈值，将匹配结果输出，否则视为噪声点滤除，这一步极大提高了输出视差的有效率。

$$d_{LR}(x_L, y_R) = \operatorname{argmin}_{d_{\min} < \delta < d_{\max}} \{SAD_{LR}(x, y, \delta)\} \quad (12)$$

$$d_{LR}(x_L - d_{LR}(x_L, y_L), y_L) = -d_{LR}(x_L, y_L) \quad (13)$$

$SAD_{LR}(x, y, \delta_{LR})$ 最小的位置是最佳匹配处，得到的是左往右匹配的最佳视差位置 $d_{LR}(x_L, y_L)$ ，如果是正确的匹配点，则右往左的最佳对应点应该为 $(x_L - d_{LR}, y_L)$ 点， $(x_L, y_L)$ 表示像素位置。在左

图往右图匹配寻找视差 $d_{LR}$ 时,通过存储适当大小的左图  
为基准向右图匹配的SAD值,无须重新计算右图往

左图匹配的SAD值,直接获得 $d_{LR}$ ,使得稠密匹配中计  
算量最小化。

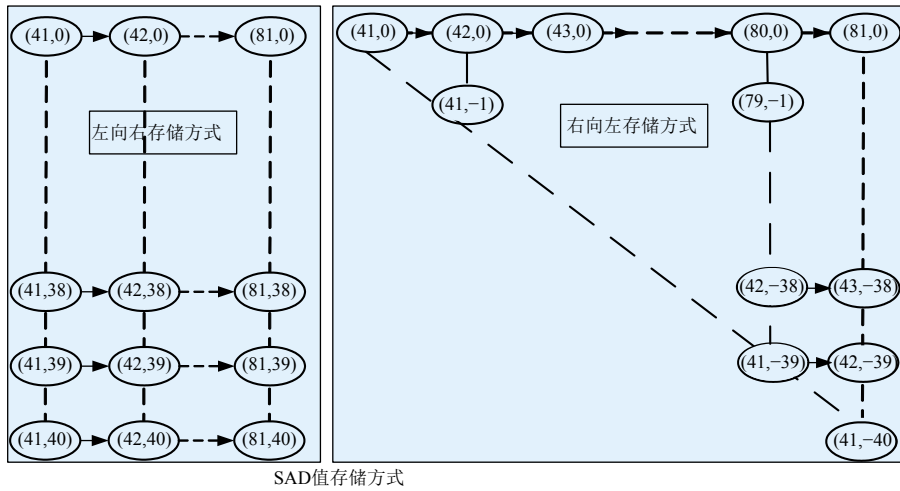


图5 左图往右图/右图往左图配准的SAD存储方式

Fig. 5 SAD storage mode of left to right/ right to left matching

在软件的实现中,采用链表结构和堆入堆出的赋值方式进行巧妙设计,实现代码最优。最佳匹配点的左右点匹配信息也被保留下来,作为亚像素拟合的输入,提升配准精度。

通过上述的算法优化设计,计算效率相比直接编程实现可以提升近10倍,且不随匹配窗口的增大而增加计算量,硬件实现也具有明显的优势。

### 3.3 三维点云生成与快速滤波

匹配完成后,立体视觉恢复出来的是视差数据,在视差数据上进行一次块状滤波,去除一些噪声,然后根据公式转换成三维点云数据,即

$$\begin{cases} Z_{HCam} = \frac{f_c \times b_c}{d_{LR}} \\ X_{HCam} = \frac{Z_{HCam} \times j}{f_c} \\ Y_{HCam} = \frac{Z_{HCam} \times i}{f_c} \end{cases} \quad (14)$$

其中:  $(X_{HCam}, Y_{HCam}, Z_{HCam})$  表示三维坐标;  $f_c$  为相机焦距;  $b_c$  为双目相机基线;  $d_{LR}$  是匹配得到的视差。

三维点云数据量非常大,而且这些数据在空间上并非均匀分布,针对这一问题,提出了一种高效三维点云滤波技术,解决大量点云数据进行多次重复遍历、处理用时较长、效率低的问题,实现了快速有效的三维地形点云输出。具体的优化设计过程:在生成三维稠密点云地形数据之前,首先构建一张均匀网格的地形图;然后在计算点云数据时对网格地形图填充

并做均值化处理,完成三维稠密点云地形数据到二维均匀网格数据的转化,同时完成地形数据的第一层滤波;在此基础上采用类似图像滤波的方法对均匀网格数据进行第二层滤波,从而完成三维地形数据的生成和滤波。这种方法在数据存储量和计算量方面都有较大改善,三维地形滤波算法的效率得到明显提高。另外,通过采用三维地形数据分层滤波的方法,提高了三维恢复的精度,保证了自主避障规划结果的安全性和有效性<sup>[6]</sup>。表1给出算法优化前后三维地形滤波用时的比较,可以看出优化后滤波用时明显减小

表1 三维地形滤波算法优化前后用时比较

Table 1 Comparison of 3D terrain filtering time before and after algorithm optimization

次数	三维地形滤波总用时	
	优化前/s	优化后/s
1	142.388	4.340
2	141.447	4.416
3	142.621	4.600
4	143.036	4.626
5	142.404	4.516
均值	142.379	4.500

## 4 硬件并行实现

为支持环境感知算法的并行执行,硬件采用如图6所示的并行计算架构。该架构采用哈弗存储方式,并包含两个独立的执行单元。每个执行单元含有独立

的寄存器组，可以执行访存、逻辑、算术运算等所有程序的指令，两个执行单元之间可通过寄存器组进行数据交互。

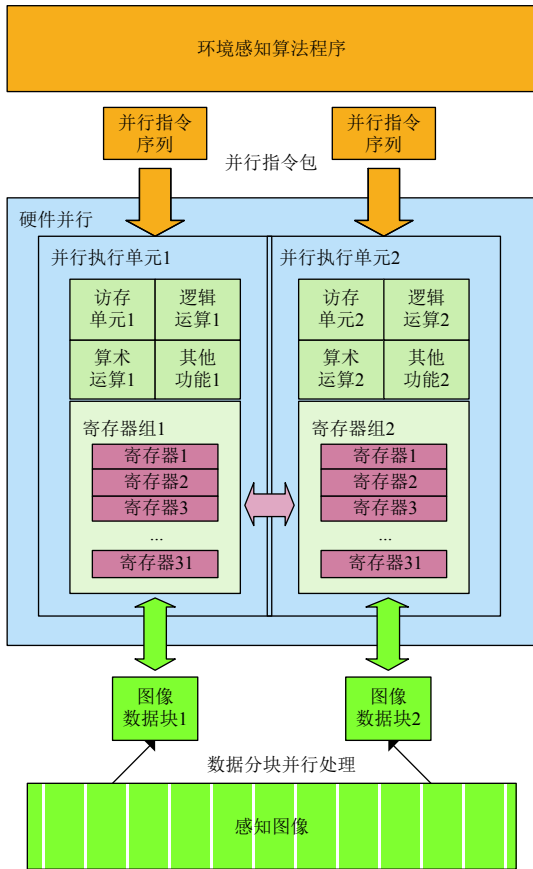


图6 硬件并行计算架构

Fig. 6 Hardware architecture of parallel computing

该硬件设计主要从以下3方面提高感知算法的并行执行。

1) 提高存储访问并行度，硬件采用哈弗存储方式，将指令和数据独立存储，实现对指令、图像数据的并行访问。

2) 提高指令并行度，两个执行单元可同时执行不同的指令包，每个指令包由并行指令序列组成。不同类型的指令，在执行单元的访存、逻辑、算术运算单元等功能模块上并行执行。通过将感知算法中包含的大量循环语句展开优化，使循环体中的语句同时在多个运算单元中执行。另外，将算法中包含的大量分支判断语句改进成条件运算语句，减少程序跳转，避免处理器指令流水线的断流，实现指令的并行。

3) 提高数据处理并行度，图像处理算法具有对大量数据进行相同运算操作的特点，将感知图像数据进行分块，两个执行单元可对不同的数据块进行并行处

理，尤其是对感知算法性能影响较大的数据处理步骤，采用单指令多数据（SIMD）汇编指令进行编程，实现对数据的并行处理。

## 5 火星车在轨飞行验证及结果

“祝融号”火星车行驶里程超过了1 km，并超过100 d的寿命期，是中国目前在地外天体上行走最远的表面巡视器。能高效完成这个行驶里程，得益于其高效安全的自主避障行走方式。“祝融号”通过立体视觉算法优化和硬件并行设计完成一次自主避障的时间目前在20 s左右，与未优化的“玉兔二号”效率相比提升了近8倍。在火面行走时，“祝融号”采用盲走和自主避障行走的组合方式，提升了地面一次操作的可行行走距离。而“玉兔二号”采用的主要是盲走方式，受限地面操作，效率较低，一次行走不超过10 m。自主避障行走使得“祝融号”的效率显著提升，一次行走距离可达20 m。“祝融号”火星车几乎每日都采用“盲走+自主规划移动”的策略，即在地面操作员利用导航地形相机图像“看得清”的范围内（约10 m）采用盲走移动，“看不清”的拓展范围（一般10 m）采用自主避障规划移动。6月6日火星车第一次自主避障规划行驶9.063 m；6月7日火星车第一次视觉里程移动8.638 m；7月22日自主避障规划模式行驶的最长距离15.12 m。

在目前的1 km总里程中，其自主里程已经占到整个里程的40%以上，已经超过“勇气号”（Spirit）的27%。“祝融号”90个火星日行走的总路线图及局部放大图如图7所示。图7中绿色的圆点表示的是每一次的导航点，导航点之间的黄色曲线表示盲走，绿线表示的是自主避障行走，在多次导航任务中，盲走和自主避障行走各占一半，其中一步探测避障相机的图像和感知视差图如图8所示。（b）图和（d）图为通过立体视觉并行算法计算出来的视差图，可直接转化为DEM（Digital Elevation Model）数据进行规划。可以看出无论是在较平缓的沙丘，还是障碍密集的地形上，“祝融号”都能有效识别出环境中的起伏，自主避障10 m完成的轨迹效果图如图9所示。

图9为“祝融号”火星车的导航相机拍摄的图片，其中根据遥测数据描绘出了其自主避障的行走轨迹线，黄色的方框为其轮子的轨迹位置，蓝色的线条表示火星车的中心移动轨迹。火星车的移动轨迹是星上感知规划算法根据立体相机对实际地形进行感知规划并控制行而获得的，初始给定的目标点在障碍的后方，火星车立体视觉算法可有效识别地形上的障碍，规划避开障碍，并保证安全到达目标点。

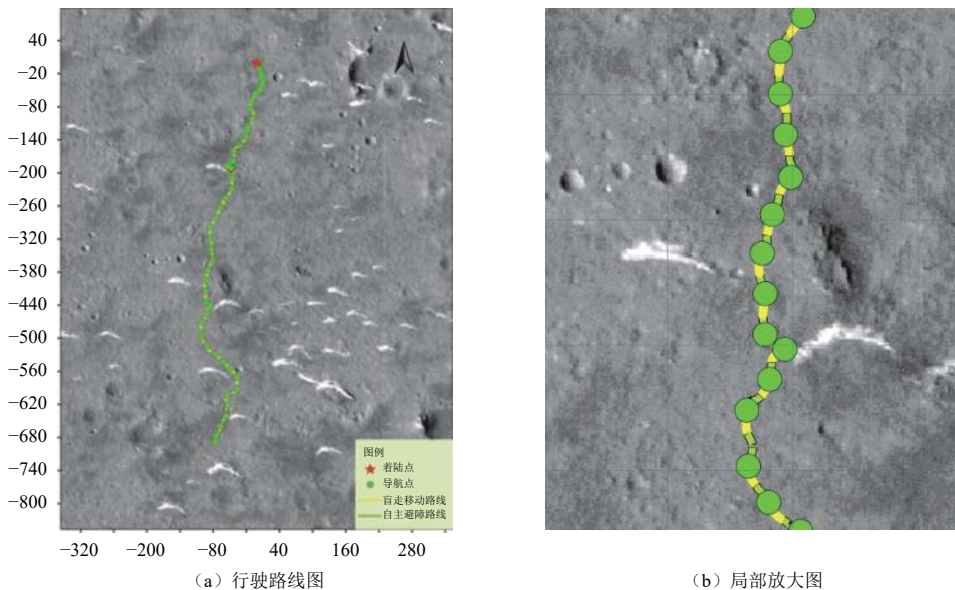


图7 “祝融号”90个火星日行驶路线图及局部放大图

Fig. 7 Route map of Zhurong 90 Mars days and partial enlarged view

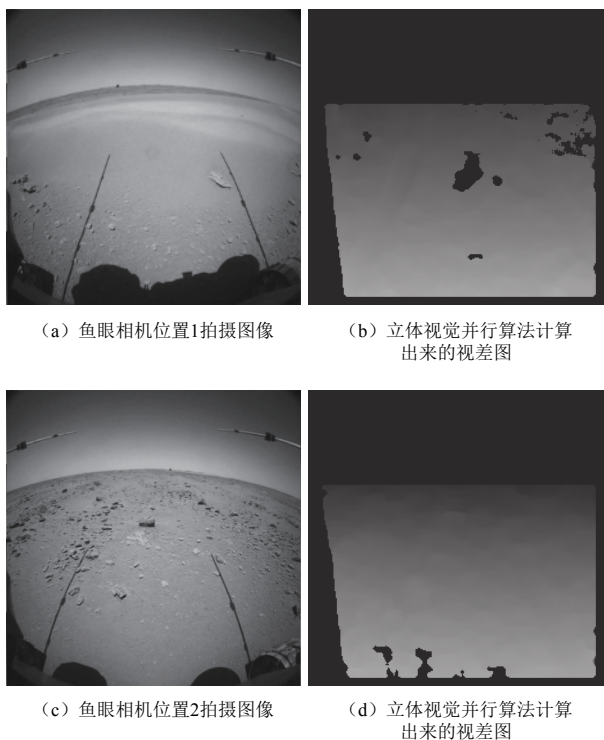


图8 避障相机图像及感知视差图

Fig. 8 Images of hazard-avoiding camera and disparity maps

立体视觉算法并行设计在“祝融号”火星车上得到了成功验证，通过可靠的火星车在未知环境下的自主行驶能力，将一次地面感知可以行走的距离提升到20 m以上。通过并行设计提升了速度，保证了在能源限定和时长等约束条件下火星车长距离的可靠安全行驶，行走距离远超“玉兔二号”，成为名副其实走得稳走得

快的地外天体表面移动平台。在后续的延展任务中，若适当放宽能源和通讯限定，火星车能够实现更远距离的时速以超过40 m/h的自主避障行走。携带立体视觉并行设计的“祝融号”将在火星上继续漫游，走更远的路、看更美的景、探测更多的未知天地。

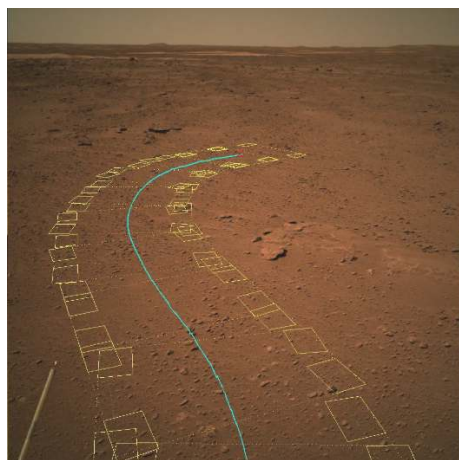


图9 火星表面自主避障行走的效果图

Fig. 9 Results of autonomous obstacle avoidance moving on Mars surface

## 6 结论

本文主要介绍了中国的第一辆火星车“祝融号”采用的立体视觉并行算法，通过多重优化设计算法与硬件并行处理，在实际工程中完成了可靠安全的高效率自主感知，探测器行走各性能表现突出。但是需要看到的是感知技术不断在进步，更多更好的方法也在不断涌现，综合工程可实现性的更优方法研究和新的硬



件实现技术是今后仍需努力的方向。

## 参 考 文 献

- [1] LEGER P C, TREBI-OLLENNU A, WRIGHT J, et al. Mars exploration rover surface operations: driving spirit at Gusev crater[C]//2005 IEEE International Conference on Systems, Man and Cybernetics. Waikoloa, HI, USA: IEEE, 2005.
- [2] BIESIADECKI J J, BAUMGARTNER E T, BONITZ R G, et al. Mars exploration rover surface operations: driving opportunity at Meridiani planum[J]. IEEE Robotics & Automation Magazine, 2006, 13(2): 63-71.
- [3] NACHON M, BORGES S, EWING R, et al. Coupling Mars ground and orbital views: generate view sheds of Mastcam images from the Curiosity rover using ArcGIS and public datasets[J]. Earth and Space Science, 7(9): 1-13.
- [4] NASA. NASA's Mars Curiosity Debuts Autonomous Navigation [EB/OL]. (2013-8-27)[2021-10-11]. <https://mars.nasa.gov/news/nasas-mars-curiosity-debuts-autonomous-navigation/>.
- [5] 魏祥泉, 黄建明, 顾冬晴, 等. 火星车自主导航与路径规划技术研究[J]. 深空探测学报(中英文), 2016, 3(3): 275-281.  
WEI X Q, HUANG J M, GU D Q, et al. Researches on the techniques of autonomous navigation and path planning for mars rover[J]. Journal of Deep Space Exploration, 2016, 3(3): 275-281.
- [6] KLETTE R, SCHLUNS K, KOSCHAN A. Computer vision: three-dimensional data from images[M]. Singapore: Springer-Verlag Ltd., 1998.
- [7] HIRSCHMULLER H. Stereo Processing by Semiglobal Matching and Mutual Information[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 30(2): 328-341.
- [8] YANG G, MANELA J, HAPPOLD M, et al. Hierarchical deep stereo matching on high-resolution images[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR). Long Beach, CA, USA: IEEE, 2019.
- [9] ZHANG J, RAMANAGOPALG M S, VASUDEVAN R, et al. LiStereo: generate dense depth maps from lidar and stereo imagery[C]//2020 IEEE International Conference on Robotics and Automation(ICRA). Paris, France: IEEE, 2019.
- [10] 王保丰, 毛晓艳, 唐歌实, 等. 月球车鱼眼相机的标定方法研究[J]. 宇航学报, 2011, 32(4): 933-939.  
WANG B F, MAO X Y, TANG G S, et al. A calibration method for fish-eye camera of lunar rover[J]. Journal of Astronautics, 2011, 32(4): 933-939.
- [11] 毛晓艳, 张晋, 张巍, 等. 机器视觉中光学成像传感器安装方法[P]. 北京: 国家专利局, 2017.
- [12] SZELISKI R. Computer vision: algorithms and applications[M]. Newyork: Springer, 2010.
- [13] MAO X Y, HUANG X Y. The simulation method for wide-angle fish-eye image with distortion[C]//2016 12th World Congress on Intelligent Control and Automation (WCICA). Guilin: WCICA, 2016.
- [14] FRASER C S. Photogrammetric camera component calibration: a review of analytical techniques[M]. Berlin: Springer Publication, 2001.
- [15] MALLON J, WHELAN P F. Precise radial un-distortion of images[C]//Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004. Cambridge, UK: IEEE, 2004.
- [16] 邢琰, 刘祥, 滕宝毅, 等. 一种适用于低信噪比图像的障碍快速识别方法[P]. 北京: 国家专利局, 2019.

作者简介:

**毛晓艳** (1977-), 女, 研究员, 主要研究方向: 表面巡视器导航方面的研究。

通讯地址: 北京海淀区友谊路104号院(100094)

电话: (010)68111183

E-mail: cicely\_mao@163.com

# Parallel Design and Implementation of Stereo Vision Algorithm of Zhurong Mars Rover

MAO Xiaoyan<sup>1,2</sup>, MIAO Zhifu<sup>1</sup>, CHEN Jianxin<sup>1</sup>, LI Zhiping<sup>1</sup>, TENG Baoyi<sup>1,2</sup>, XING Yan<sup>1,2</sup>

(1. Beijing Institute of Control Engineering, Beijing 100190, China;

2. Science and Technology on Space Intelligent Control Laboratory, Beijing 100190, China)

**Abstract:** Aiming at the problem that the large amount of environmental perception calculation affects the walking efficiency in the rover detection task of celestial bodies outside the earth, the scheme selection, the parallel optimization engineering design and the efficient hardware implementation of stereo vision algorithm were proposed. Through the optimization design of algorithms suitable for parallel implementation, such as one-time look-up table of forward camera model, omitting redundant calculation, 3D point cloud generation and fast filtering, the amount of calculation for stereo vision was significantly reduced. Through the parallel implementation of hardware, the perception efficiency was improved by 8 times compared with that of “Yutu 2” Lunar rover. “Zhurong” Mars rover extensively used the algorithm for autonomous obstacle avoidance and completed a safe and efficient 1km-distance walk. The flight verification results show that the design is safe and effective under unknown environments and resource constraints of celestial bodies outside the earth, and can be used as the cornerstone of subsequent deep space missions.

**Keywords:** Zhurong Mars rover; stereo vision; parallel design; forward camera model; redundant calculation

## Highlights:

- One-time look-up table method for forward model is proposed.
- Redundant calculation during matching window sliding is reduced.
- The corresponding relations of left to right forward and right to left reverse matching are established to reduce the amount of calculation while guaranteeing the robustness.
- The single point calculation is directly filled into the 2D map and filtered while the output process of 3D dense point cloud is cancelled.

[责任编辑: 杨晓燕, 英文审校: 宋利辉]