

分类号_____

密级_____

UDC _____

编号_____

中国科学院研究生院

博士学位论文

太阳望远镜高时空分辨率终端系统研制

肖江

指导教师 _____ 邓元勇 研究员

_____ 中国科学院国家天文台

申请学位级别 _____ 博士

学科专业名称 _____ 天文技术与方法

论文提交日期 2009年5月 论文答辩日期 2009年5月

培养单位 _____ 中国科学院国家天文台

学位授予单位 _____ 中国科学院研究生院

答辩委员会主席 _____

**A dissertation submitted to National Astronomical Observatories for the degree
of Doctor of Philosophy with a Major in Astronomical Instrument**

**Development of High-temporal and spatial Terminal
System for Solar Telescope**

**By
Jiang Xiao**

Directed By Prof. Yuanyong Deng

**National Astronomical Observatories
Chinese Academy of Sciences**

摘要

本论文的工作是在系统分析了国家天文台怀柔太阳观测基地太阳望远镜采集系统的基础上,围绕着实现太阳望远镜高速数据处理终端和传输系统而展开,完成了太阳望远镜高时空分辨率实时观测的两套重要系统研制:(1)高速实时相关采集系统;(2)高精度自动导行系统。

高速实时相关采集系统是目前实现太阳磁场望远镜高时空分辨率观测的一项重要课题,实现该技术的关键在于大数据量的实时处理,本文系统分析和研究了计算机处理器的发展历史,提出了将最新的 GPU (Graphic Process Unit) 处理器应用于天文终端设备的解决方案,并开展了关键技术的攻关,建立了一套基于 CUDA 的 35 厘米太阳磁场望远镜的新一代终端系统,获得成功的试观测并投入使用。由于 GPU 在多线程并行计算上特有的诸多优势,使得该技术在类似天文终端设备要求的领域具有潜在应用价值。

本文的第二部分系统地探讨和分析了目前最先进的基于大面阵 CCD 的全日面自动导行方案,通过计算机模拟,采用“二值化”方法,提高了数据处理的质量。将该导行方案应用到全日面 H α 太阳望远镜上,通过试观测,在连续工作 1 小时的情况下该系统导行精度优于 1", 达到系统设计需求。

为解决高速数据传输及数据自动寻址问题,本论文第三部分还提出了采用 USB2.0 标准设计控制系统接口的思想,并通过一系列工程实践,成功地应用于多通道望远镜多波带控制系统的调制上,极大地提高了数据处理效率。

关键词: 太阳磁场、高时空分辨率、实时相关、自动导行、USB2.0 接口

Abstract

Through analyzing the acquisition system of the solar telescope in Huairou Solar Observing Station(HSOS) of National Astronomical observatories(NAOC), this dissertation focused on high-speed data processing and transmission terminal system and fulfills two key systems for high temporal and spatial observation: (1) The High-speed real-time Image- Processing System with Correlation Algorithm (CA); (2) The high-precision Tracking and Guiding System with Full Solar Disk Image .

The High-speed real-time Image- Processing System with CA is an important method for high temporal and spatial observation. The key point in this system is the large-scale data processing. After reviewing the history of computer processor, we puts forward a method for data acquirement and process system by use of the concept of GPU (Graphic Process Unit) . After tackling key technical problems, a new system has been set up for the 35cm solar magnetic field telescope of HSOS, and an experimental observation has been successfully carried out. Because the GPU has much more advantages on parallel computation, this new technique has many potential applications in astronomical.

The most advanced full-disk image automatic guiding system of solar telescope is discussed in Part 2.Through a series of computer simulation, binarization method is adopted which largely improves the processing quality for large-scale data. This new system is applied to the Full-disk H α Telescope, and achieved tracking precision of 1'' during one hour test observation.

In order to solve the problems in data high-speed transmission and data automatic addressing,the improvement of control system interface by USB2.0 are introduced in Part 3.Through a series of experimental applications, the new interface was applied to multi-pass band control system and largely improved the speed of respond.

Key Words: Solar Magnetic Field, High Temporal and Spatial, Real-time Correlation, Automatic Tracking and Guiding Method, USB 2.0 Interface

目 录

第一章 引言	1
1.1 太阳磁场测量研究的特点和意义	1
1.2 太阳矢量磁像仪的原理及发展	1
1.3 科学计算的方法和特点	5
1.4 大规模科学计算的现实意义和实用价值	6
1.5 太阳望远镜实现高速大规模科学计算的必要性	6
1.6 本论文研究的目的	7
1.7 本论文的主要内容和研究成果	8
第一部分 用 GPU 并行处理器实现大数据量实时相关图像处理	11
第二章 新一代处理器设备——GPU 图像处理器在并行计算应用中的优越性	11
2.1 传统 CPU 在大数据量计算中存在的问题	11
2.2 计算机 GPU 发展的历史与趋势	13
2.3 CUDA 技术的原理和特点	16
2.4 天文上应用 CUDA 的可行性以及 CUDA 并行计算性能测试	18
第三章 基于 CUDA 的 35CM 磁场望远镜实时相关终端系统	28
3.1 基于 CUDA 的 TESLA C1060 高性能计算设备	28
3.2 35CM 磁场望远镜实时相关终端系统	30
3.3 总结与讨论	47
3.4 CUDA 技术在天文观测中的应用前景	51

第二部分 运用“二值化”方法的面阵 CCD 导行在太阳磁场观测中的应用 55

第四章 太阳磁像仪自动导行方案的发展趋势 55

4.1 太阳磁像仪使用导行系统的必要性 55

4.2 太阳磁像仪导行方案的发展历程 57

4.3 大面阵 CCD 自动导行方案在工程上面临的主要技术瓶颈 59

第五章 面阵 CCD 导行——基于质心算法的全日面像自动导行方案 60

5.1 基于质心算法的大面阵 CCD 全日面导行的基本思想 60

5.2 基于质心算法的全日面导行方案的原理 60

第六章 面阵 CCD 导行——基于质心算法的全日面像导行系统 62

6.1 CUDA 技术在面阵 CCD 质心计算中的劣势 62

6.2 利用“二值化”方法实现实时大数据量质心计算 63

6.3 基于质心算法的全日面导行方案的模拟 65

6.4 基于质心算法的全日面导行方案的系统设计 66

6.5 基于质心算法的全日面导行方案测试结果 71

6.6 结论与讨论 73

第三部分 USB2.0 接口在望远镜高速控制系统中的应用研究 75

第七章 CYPRESS EZ-USB2.0 在太阳望远镜控制系统中的应用 75

7.1 原有技术的瓶颈和采用通用串行总线的意义 75

7.2 USB 2.0 协议通信的基本原理 76

7.3 CYPRESS 公司 EZ-USB 2.0 芯片的主要特点 78

7.4 EZ-USB2.0 芯片主机通信过程 79

7.5 基于 CYPRESS EZ-USB 2.0 芯片的接口设计 82

7.6 总结与讨论 93

第八章 结束语	96
博士期间作者发表的论文和申请的专利	98
致谢	99
参考文献:	100

第一章 引言

1.1 太阳磁场测量研究的特点和意义

万物生长靠太阳,对于人类来说,光辉的太阳无疑是宇宙中最重要的天体。太阳的光和热温暖着地球,维持着人类以及地球上一切生命活动所必需的生存环境。同时,太阳也是地球上除了原子能以外的所有能源的直接和间接的创造者^[1],我们时时刻刻都感受着太阳作用的结果——风云雨雪,昼夜交替,四季变化……太阳与人类有着非常密切的联系。

太阳是一颗典型的恒星,对于天文学而言,她是唯一一颗可以被精细观测的恒星天体,关于她的研究对于了解广阔的恒星世界有着普遍的意义。对于太阳辐射和太阳活动规律以及日地关系的研究,极大地推动了物理学的发展,产生了较多新兴的研究领域,此外,太阳活动现象会对日地空间环境和地球高空大气产生扰动,会引起一系列重要的地球物理现象,如太阳质子事件,磁暴,平流层升温等,这些事件会影响到空间飞行安全、人造卫星寿命、地物探矿、气象和水文预报等与国民经济有关的部门^[1]。以太阳为核心的行星系统,提供了研究生命起源的最佳样本,关于太阳的研究,对了解生命的起源也有重要意义。

磁场是产生太阳大气结构和太阳活动现象的重要因素,这也是目前人类可以通过观测获取的太阳表面的资料。通过高时空分辨率的太阳磁场观测可以分析太阳系统的物理特征和太阳活动的物理机制,太阳磁场的研究一直是太阳物理学界研究的前沿和热点。

1.2 太阳矢量磁像仪的原理及发展

太阳磁场的测量归功于 1896 年 Zeeman 效应的发现,即磁场能够引起单色谱线分裂的理论。1908 年, Hale 根据这一原理利用高分辨率光谱仪和偏振分析器组成的太阳磁场观测仪器,观测到来自太阳黑子区域的单色光谱线有明显的分裂,进而推算出太阳黑子的磁场强度达 2000—3000 高斯^[2]。法国太阳物理学家 Lyot 和 Ohman 分别于 1933 年、1938 年发明了双折射滤光器,从而第一次获得太阳的二维单色像^{[3][4]}; 1952 年, Babcock 父子利用光电原理,成功地解

决了弱磁场的测量问题,使太阳磁场的研究从黑子区域扩展到整个日面^[5];1960年,前苏联克里米亚天文台的 Stepanov 等人又发明了能够测量太阳横向磁场的光电矢量磁像仪^[6];1968年,Beckers 首次将双折射滤光器用于太阳磁场测量,在同一时间内得到太阳视面磁场图像,大大地提高了望远镜的时间分辨率^[7]。进入上世纪七、八十年代,二维光谱仪(同时获得光谱信息和二维空间信息的仪器)的重要性越来越为太阳物理学家所重视,为此,法国天文学家 Mein 等人研制了MSDP 光谱仪 (Multi-Channel Subtractive Double Pass Spectrograph)^[8],八十年代末,中国太阳物理学家艾国祥等人提出多通道滤光器的概念,即在同一时间内可以得到多条太阳谱线的矢量磁图,从而实现对太阳磁场的准三维观测,并于九十年代研制成功^{[9][10]}。

根据原子光谱理论,在无外磁场作用时($B=0$),原子在高能级和低能级之间跃迁时,便发出频率为 ν_0 的单色光;当存在外磁场时($B\neq 0$),处在磁场中的原子能级将发生空间量子化,从而相应的发射线也将分裂出子线,这就是 Zeeman 分裂子线。相应的,如果是谱线被原子吸收而发生的从低子能级向高子能级的跃迁就是逆 Zeeman 效应。太阳磁场观测通常采用简单的、只有三分裂的谱线,即谱线在外磁场中分裂为三条子线:一条 π 子线,分裂后波长 λ_0 不变;两条 σ 子线,其波长在分裂后则漂移到 $\lambda_0\pm\Delta\lambda$ 。这种特殊的三分裂称为正常 Zeeman 三分裂,其裂距 $\Delta\lambda = 4.67\times 10^{-13} g\lambda^2 B$,式中 g 为磁敏因子, B 为以高斯为单位的磁场强度, λ 是以埃为单位的无磁场时的单色光波长。

上述原子光谱理论讨论的是纯发射线和纯吸收线的情况,但实际上,由于光线通过的太阳大气物质不断吸收辐射又不断发生辐射,从而形成了所谓的太阳夫琅和费谱线。夫琅和费谱线的 Zeeman 分裂子线的强度和偏振状态不能借用上述两种情况简单推测,而必须建立磁场存在时的谱线辐射转移方程,并在某些假定条件下求解,得到 Zeeman 分裂各子线的强度与偏振状态。通常,人们采用 Stokes 参数来描述夫琅和费谱线的真实状态。

Unno 在 1956 年首先得到一套 Stokes 参量转移方程^[11],并在假定磁场为均匀、谱线为真吸收以及线吸收系数与连续吸收系数之比与深度无关等条件下,求得理论 Stokes 轮廓,为现有的天体磁场测量奠定了理论基础。Unno 在最初的求解 Stokes 参量转移方程的过程中,没有考虑磁光效应引起的偏振面的旋转。

1962 年 Rachkovsky 对此进行了改进, 加入了代表磁光效应的系数, 建立了 Unno-Rachkovsky 方程组^[12]。Unno 方程组和 Unno-Rachkovsky 方程组为处理现有的斯托克斯参量仪在取得四个 Stokes 参量的观测资料后进一步处理的理论基础。在此之后不少太阳物理学家从事此方向的研究, 例如 Jefferies, Lites 和 Skumanich 从经典力学的角度对 Stokes 参量的辐射转移做了细致的推导和描述^[13], E. Landi Degl’Innocenti 从量子力学的角度对偏振光的辐射转移做了更为精确的推导和描述^[14]。

目前世界上测量太阳向量磁场的望远镜分为空间望远镜和地基望远镜。空间太阳望远镜观测始于第二次世界大战后, 至今已有半个多世纪, 它的观测内容和技术不断发展。在 20 世纪 40 至 50 年代, 主要是利用探空火箭观测太阳的紫外和 X 光辐射。六十年代后, 开始用人造卫星对太阳进行更加多样化的大量观测。进入八十年代以后, 以高空间分辨率和高偏振精度观测太阳磁场, 成为一些太阳物理学家追逐的目标, 1995 年 10 月, 欧美合作研制的观测卫星“太阳和日球天文台”(Solar and Heliospheric Observatory, “SOHO”)发射运转成功, 首次得到在太空观测到的太阳磁场图像^[15]。地基望远镜受到地球大气的影响, 观测能力不如空间望远镜, 但地基望远镜建设成本低, 便于维护, 在一些观测台址仍然有很高的观测质量, 比较著名的如: 中国国家天文台怀柔观测基地的多通道太阳望远镜, 美国大熊湖天文台、美国马歇耳空间飞行中心, 日本国立天文台, 美国夏威夷大学 Mees 天文台、美国 Kitt 峰天文台等。

太阳望远镜发展至今其磁场测量仪有两大类: 双折射滤光器型和光谱仪型。属于前者的有中国国家天文台怀柔观测基地的多通道太阳望远镜, 美国大熊湖天文台、美国马歇耳空间飞行中心的向量磁像仪, 日本国立天文台的太阳耀斑望远镜等; 属于后者的有美国夏威夷大学 Mees 天文台的斯托克斯参量仪、美国 Kitt 峰天文台的光谱磁像仪^{[16][17]}, 俄国 Sayan 天文台的向量磁像仪^[18]等等。双折射滤光器型的优点是可以直接得到太阳的二维图像, 获得二维单色像以及此图的时间分辨率远远高于光谱仪型, 但是无法得到谱线轮廓; 光谱仪型的优点是可以观测到谱线轮廓, 但是时间分辨率较低, 不利于观测太阳短时爆发活动。

尽管各天文台太阳观测仪器各有特色, 但是他们的基本原理都是先让望远镜将来自太阳的光线通过偏振光分析器和分光系统, 得到处于不同偏振状态的

太阳图像，组成不同的 Stokes 参量组合，然后从这些组合中把各个 Stokes 参量解出来，再通过求解 Stokes 参量转移方程，进一步得到磁场向量 $B(B, \gamma, \chi)$ 的三个要素，即磁场强度 B ，磁场向量与视线的夹角 γ ，以及横向磁场的方位角 χ 。

怀柔多通道太阳望远镜包含几套相似的矢量磁像仪，具有代表性的是 35cm 太阳磁场望远镜，其偏振光分析器如图 1-1 所示，包括一个 $\lambda/4$ 波片，一个用来调制偏振状态的 KD^*P 电光晶体，其后还有一个光轴方位角为 0° 的偏振片。 $\lambda/4$ 波片位于偏振分析器的最前面，根据观测太阳横场或纵场的实际要求，确定它是否进入光路。偏振分析器的工作原理是：通过光路中偏振元件的不同组合状态，只让 Stokes 参数中一种偏振光完全通过，与之正交的偏振光被吸收。磁场望远镜的具体观测模式如下：不用 $\lambda/4$ 波片，只用 KD^*P 调制器时，获得 Stokes 参数 V ；当 KD^*P 调制器前加上光轴为 0° 或 45° 的消色差 $\lambda/4$ 波片时，可以测得 Stokes 参数 Q 或 U 。从 Q 、 U 和 V 可以求出纵向磁场的大小 $B_{//} = C_{//}V$ ，横向磁场的大小 $B_{\perp} = C_{\perp} (Q^2 + U^2)^{1/4}$ ，横向磁场的方位角 $\varphi = 0.5 \text{tg}^{-1}(U/Q)$ 。其中， $C_{//}$ 和 C_{\perp} 分别为纵向磁场和横向磁场的定标系数，可以通过求解上节的提到的辐射转移方程得到^[12]，也可以通过实测作经验定标^[19]。

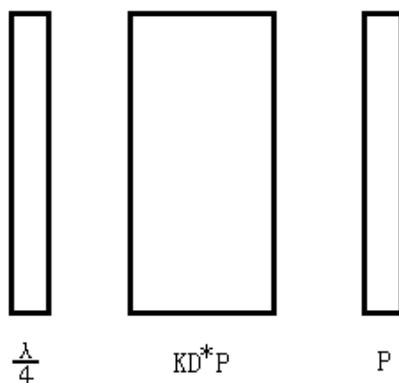


图 1-1 偏振光分析器的组成

从以上的描述中，我们可以发现高质量的太阳观测仪器的关键在于高时间分辨率，高空间分辨率，高偏振精度观测。高空间分辨率要求望远镜定位准、近地面大气湍动影响小。高时间分辨率要求图像采集的时间尽可能短（秒量级以下）。但是太阳本身进入望远镜的与磁场有关的有效光量是非常微弱和有限的，这就需要 CCD 长时间积分，或者将多帧叠加。如国家天文台怀柔观测站的太

阳磁场望远镜,在单帧采样时间 20ms 的情况下,信号比噪声大约大 70 倍左右,只能测到百高斯以上的强磁场;要想测到 10 高斯以下的弱磁场,需要 256 帧以上的叠加,也就是说提高磁场测量精度要求长时间积分。

另一方面高时间,高空间分辨率,高偏振精度观测对望远镜终端数据处理能力也提出了很高的要求,这也是一直困扰天文仪器发展的因素之一。随着科学 CCD 的发展,大面阵 CCD 广泛应用于天文设备,由此产生了巨大的数据量。以怀柔太阳观测站 35 厘米磁场望远镜为例,每张采集图像大小高达 8M,在太阳活动较多的情况下,每天的数据量可以超过 10G。实时相关,自动导行处理都要求实时完成,因此太阳望远镜对终端设备的数据处理能力的要求是远远高于一般天文观测设备的,这也是当前太阳望远镜高质量实时观测面临的瓶颈之一。

1.3 科学计算的方法和特点

科学计算是为解决科学和工程中的数学问题而利用计算机进行的数值计算。自然科学规律通常用各种类型的数学方程式表达,科学计算的目的是寻找这些方程式的数值解。这种计算涉及庞大的运算量,简单的计算工具难以胜任。在计算机出现之前,科学研究和工程设计主要依靠实验或试验提供数据,计算仅处于辅助地位。计算机的迅速发展,使越来越多的复杂计算成为可能。利用计算机进行科学计算带来了巨大的经济效益,同时也使科学技术本身发生了根本变化:传统的科学技术只包括理论和实验两个组成部分,使用计算机后,计算已成为同等重要的第三个组成部分。

科学计算主要包括建立数学模型、建立求解的计算方法和计算机实现三个阶段。建立数学模型就是依据有关学科理论对所研究的对象确立一系列数量关系,即一套数学公式或方程式。复杂模型的合理简化是避免运算量过大的重要措施,这也可以通过计算机软件优化来实现。数学模型一般包含连续变量,如微分方程、积分方程。它们不能在数字计算机上直接处理。为此,先把问题离散化,即把问题化为包含有限个未知数的离散形式(如有限代数方程组),然后寻找求解方法。计算机实现包括编制程序、调试、运算和分析结果等一系列步骤。软件技术的发展,为科学计算提供了合适的程序语言(如 FORTRAN、ALGOL)和软件工具,使工作效率和可靠性大为提高^[20]。

天文学作为理学学科的一个分支,它的发展是与数值计算密切相关的。天

文学中的许多理论公式，数值模拟都涉及科学计算的问题，因此，科学计算的技术和方法对于天文学的发展来说，有着十分重要的意义。

1.4 大规模科学计算的现实意义和实用价值

随着计算机技术的不断发展，科学计算逐步发展成广泛应用的新兴交叉学科。科学计算已与理论和实验研究一起成为当今世界科学研究的主要手段。科学研究、高新技术和重大工程中的科学计算问题越来越需要由可信的计算技术来完成。

目前科学计算需要解决的主要问题，如：工业问题中的建模分析与优化，金融风险分析与预测，地球系统模拟，大气、海洋、地下水和石油等复杂流体计算，材料物理中的多尺度计算，复杂生命系统的计算、模拟与控制技术，高性能计算方法和技术，并行计算、网络计算中的关键技术，计算机辅助技术，大规模科学计算软件平台，可信计算系统的体系结构与关键技术，网络系统的安全机制，可信计算系统开发与管理等^[21]。

在天文学领域,随着望远镜和探测器的技术水平日益强大,天文数据如同 IT 界中反映计算能力随时间指数增长的摩尔定律一样迅速膨胀,其增长一倍的时间已经不到一年,分布在世界各地的 TB (万亿字节) 甚至 PB (千万亿字节) 量级的天文数据对数据的存储、检索、处理、统计分析等提出了新挑战。目前天文学领域中的,宇宙大尺度结构形成、星系相互作用、星系形成和行星流体动力学等的物理模型和数学模型都面临数据处理能力的瓶颈。

但是由于计算机的科学计算能力仍然有限,例如在天文研究方面只能处理粗糙的数学模型,或进行中、短期的预测,因此要进行更精确的数学模型计算,必须配备更强大的计算机。越来越多的基础学科和工程技术部门已提出超过现有计算能力的大型科学计算问题。这些问题的解决,有赖于两方面的努力:一是大大提高计算机的运算速度,这有赖于硬件处理能力的改善;一是创造出更高效的计算方法,这是从软件着手,在不影响结果的情况下,精简计算冗余。

1.5 太阳望远镜实现高速大规模科学计算的必要性

如 1.2 节的阐述,随着科学 CCD 的发展,大面阵 CCD 广泛应用于天文设备。目前国家天文台怀柔太阳观测基地的望远镜终端系统已经全部采用面阵

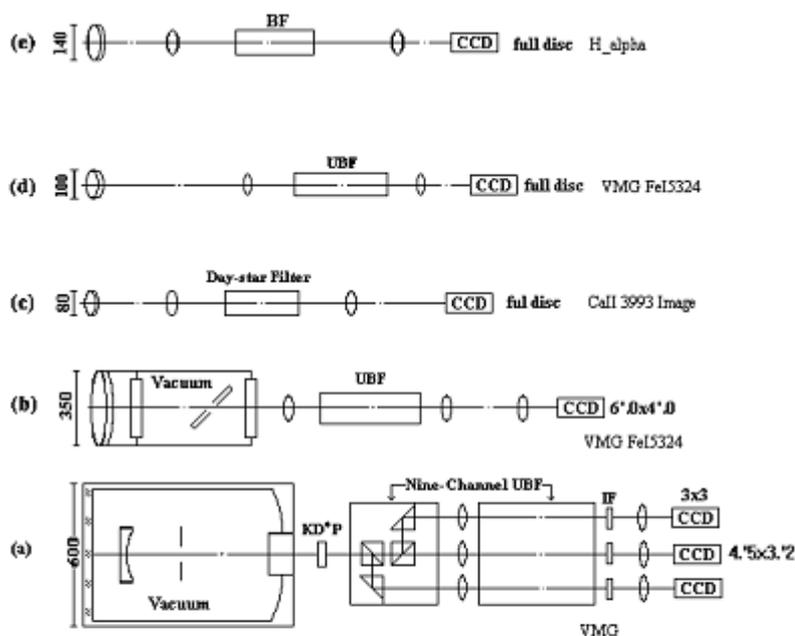
CCD,产生了巨大的数据量,由此带来了海量数据处理的难题。

目前太阳磁场望远镜的海量数据处理主要存在于两个方面:一个是图像相关需要在极短的时间内完成整幅窗口图像的计算,之后进行图像的叠加处理,之前的系统迫于计算机处理能力的限制,不能实现实时的图像相关,导致磁场观测资料时空分辨率下降;一个是高精度的自动导行系统需要实时计算面阵 CCD 每一幅图像的质心,这也带来了大数据量的处理问题,需要选用更强大的处理器和更好的方法解决数据处理的瓶颈问题。

1.6 本论文研究的目的

本论文研究的目的在于结合当代太阳物理研究的最新需求和现代高科技领域特别是计算机领域的最新技术,通过一系列望远镜终端设备的改造,采用新的计算机处理设备,优化软件算法,极大提高太阳望远镜终端系统的计算能力,进一步提升望远镜高质量观测的潜力,为展开空间太阳望远镜终端系统的研究打下基础。

多通道太阳望远镜(MCST)如图 1-2 所示,共有五个部分组成:60CM 九通道太阳望远镜(NCST)(1994 年建成)、35CM 太阳磁场望远镜(SMFT)(1986 建成)、14CM $H\alpha$ 望远镜(1990)、10CM 全日面磁场望远镜(1990)、8CM 全日面望远镜(1990)。多通道望远镜以二维太阳光球、色球矢量磁场和视向速度场为主要观测内容,为太阳磁活动的研究提供实测基础,为日地关系、空间天气监测提供实时警报,并在太阳活动周、磁场内禀性质、三维磁结构等太阳物理前沿课题的观测研究中发挥重要作用。



BF: birefringent filter UBF: universal BF
 KDP: KDP modulator IF: interference filter

图 1-2 多通道望远镜光学系统概图^[22]

该望远镜设计建造于上世纪八十年代及九十年代初，电子和终端设备采用的是当时先进的元器件，但随着现代高科技的飞速发展，这些设备目前已经大大落后于先进水平了，因此仪器的更新改造势在必行。但另一方面，由于望远镜采用真空折射式设计，结构非常紧凑，很难从硬件上进行大的改动，例如当今太阳塔式望远镜和新建造望远镜所常采用的相关跟踪器、自适应光学系统等很难应用在多通道望远镜上。因此本论文将结合该仪器的特点和现代高科技水平，对望远镜电子设备中可进一步提高时间和空间分辨率的自动导行系统和图像处理系统进行研究，提高终端数据处理能力，以期获得更高质量的太阳磁场观测资料。

1.7 本论文的主要内容和研究成果

本论文在第一章从理论上回顾了太阳磁场的测量原理及 Stokes 参数的数学描述，给出了本课题研究的背景，研究的意义及成果。

第二章从计算机处理器的发展历史入手，通过计算比较 CPU 和 GPU 的处理能力，提出符合 35 厘米磁场望远镜高速相关采集系统设计方案，并指出 CUDA 技术在天文领域应用的可行性。

论文第三章着重于 35 厘米磁场望远镜终端系统的方案设计、平台建设与软件工具。新系统成功地应用于常规观测，改正了图像叠加误差，增加了图像中更多的细节，大大提高了望远镜图像采集质量。这是国内第一套基于 GPU 并行处理器实现大窗口实时相关的终端系统。

在第四章详细介绍了太阳磁像仪导行系统的发展历程。作者力求通过本章介绍，指出太阳望远镜导行系统的发展趋势及其重要性。

第五章分析了基于大面阵 CCD 全日面导行方案的基本思想和原理，并结合望远镜的轴系控制系统得出误差像素和校正脉冲之间的数学关系。

第六章对 GPU 和 CPU 在计算质心算法情况下进行处理速度的分析和比较，给出了“二值化”方法的理论依据，然后进行了全日面导行方案的模拟，最后介绍了工程实现的软硬件系统和系统的测试结果，实现了导行精度优于 1”，这是国内第一台基于大面阵 CCD 用质心算法实现自动导行的望远镜。

第七章重点研究了望远镜控制系统的接口设计。研究接口设计的目的在于随着数据量的几何级数增长，大量的数据传输不能仅仅依靠原有的串行控制接口，需要有一种新的接口来满足这种需要。采用 USB2.0 技术可以大大提高数据传输的效率和稳定性。将当今先进的 CYPRESS EZ-USB 串口通信技术应用用于多波带控制系统的设计，极大的提高了数据的传输速率。

最后是结束语,致谢，参考文献和作者博士期间发表的论文及申请的专利。

第一部分 用 GPU 并行处理器实现大数 据量实时相关图像处理

第二章 新一代处理器设备——GPU 图像处理器在并行计算 应用中的优越性

2.1 传统 CPU 在大数据量计算中存在的问题

从上世纪四十年代世界第一台电子数字计算机“埃尼阿克”(英文缩写词是 ENIAC, 即 Electronic Numerical Integrator and Calculator, 中文意思是电子数字积分器和计算器)诞生起, 计算机技术取得了空前的发展, 晶体管计算机, 大规模、超大规模计算机不断涌现, 发展迅速, 由此引发了第三次科技浪潮。尽管目前芯片集成能力越来越高, CPU 浮点运算能力越来越强, 内存越来越大, 但是计算机的体系结构依然没有发生重大的变化——冯·诺依曼架构仍然是计算机架构的主要思想。

冯·诺依曼理论的要点是: (1) 程序和数据都用二进制数表示 (2) 存储程序控制, 要求计算机完成的功能, 必须事先编制好相应的程序, 并输入到存储器中, 计算机的工作过程是运行程序的过程 (3) 机器以 CPU 为中心 (4) 指令由操作码和地址码构成^[23]。冯·诺依曼理论解决了曾经一直困扰人们的数值计算机应该采用什么样结构的问题。人们把冯·诺依曼的这个理论称为冯·诺依曼体系结构。从 ENIAC 到当前最先进的计算机都采用的是冯·诺依曼体系结构。所以冯·诺依曼被人们称之为数字计算机之父。

在过去十年中, 由 Intel、IBM、SUN、AMD 和富士通生产的通用 CPU 虽然有了很大发展, 但性能提高速度却已经不能与上世纪八十年代末九十年代初相比。CPU 面临着巨大的挑战。如图 2-1 可以看出, 从 2003 年开始提升 CPU 浮点的运算能力的步伐已经是步履维艰了, 浮点运算能力一直在 3.0GHz 左右徘徊。出现这样的瓶颈一方面来自于“功率墙(Power Wall)”——集成电路的功率消耗的物理限制以及“存储墙 (Memory Wall)”——从 CPU 核心到内存之间

的数据交换因受制于带宽而产生的延迟，另一方面来自于通用计算程序中过低的指令级并行。

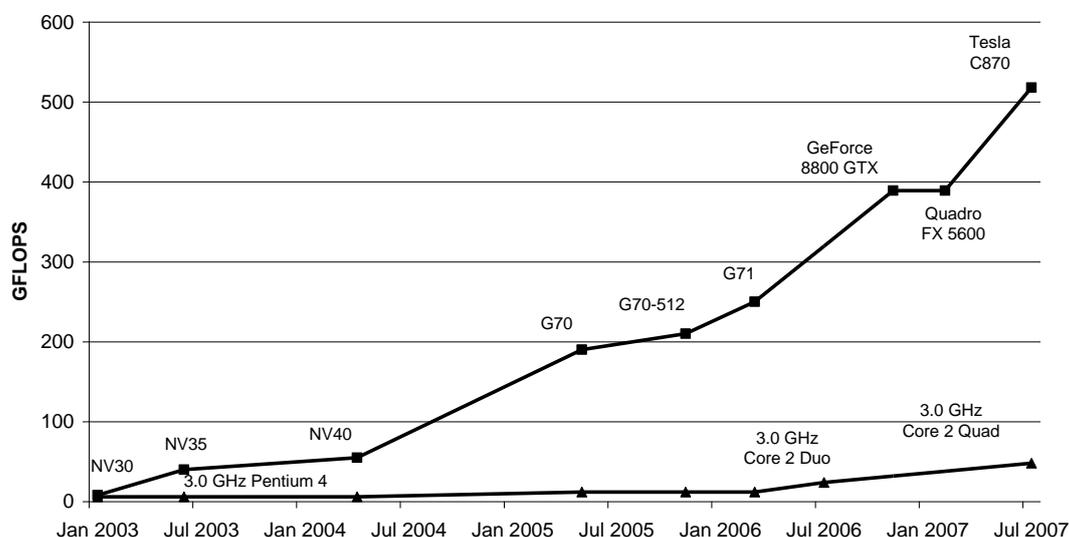


图 2-1 GPU 和 CPU 在过去几年中浮点运算能力的性能比较^[24]

然而，人们对计算的需求是永无止境的，如在气象、地质，资源以及时系跟踪等领域，它们对程序处理速度的要求都是相当高的，否则就会导致结果出现偏差或者失去其意义。目前对于海量数据的高速处理，一般都要通过高性能计算机来完成。谈到高性能计算很容易让我们联想起刚刚过去的全球 Top500 超级计算机排行榜，位列榜首的 IBM 超级计算机“走鹃”计算速度超过千万亿次^[25]。“走鹃”共包含 3456 个 Tri-Blade 节点，占地面积 557 平方米，连接光纤长达 91.7 公里，每小时耗电达到 3000 度，总造价超过 1.3 亿美元。

虽然并不是每台超级计算机都像“走鹃”有那样昂贵的身价，占用那样大的空间，但是总整体上来看，这些被安置于国家实验室、国家核安全局、超算中心等关键部门，进行核研究、航空、能源、基因或者气候方面研究的超级计算机们免不了都具备“大块头、高能耗、高价格”这三道门槛，对于中小型科研单位、企业研发中心或者高校等同样需要高性能计算的客户来说，面对高价格和惊人的能耗，使用这些超级计算机的成本是高昂的。

传统的超级计算机集群架构是将多台服务器通过高速网络连接成分布式运算的集群技术，通过增加计算节点，以提升系统的计算性能。集群的节点越多，

采购成本、系统功耗和占用空间都会相应增加，管理也越复杂。

再从计算资源分配上来看，传统的超级计算平台造价昂贵，计算资源一般由多个科研项目和科研人员共享，而大多数科研院所也不可能永远都能有充裕的资金将集群升级，随着计算需求的不断增加，往往会出现计算平台拥挤的情况，造成计算效率的低下。这种平台模式就好像一辆公共汽车，所有的人员不得不起挤。以上海超算中心为例，上海超算中心曙光 4000A 超级计算机，自 2004 年 11 月投入使用后系统平均可用率为月均 99.94%，月均利用率在 80% 以上，到 2007 年 11 月达到 91.03%^[26]（如图 2-2 所示），使用超级计算机要提前申请，按照机时付费，成本高昂！

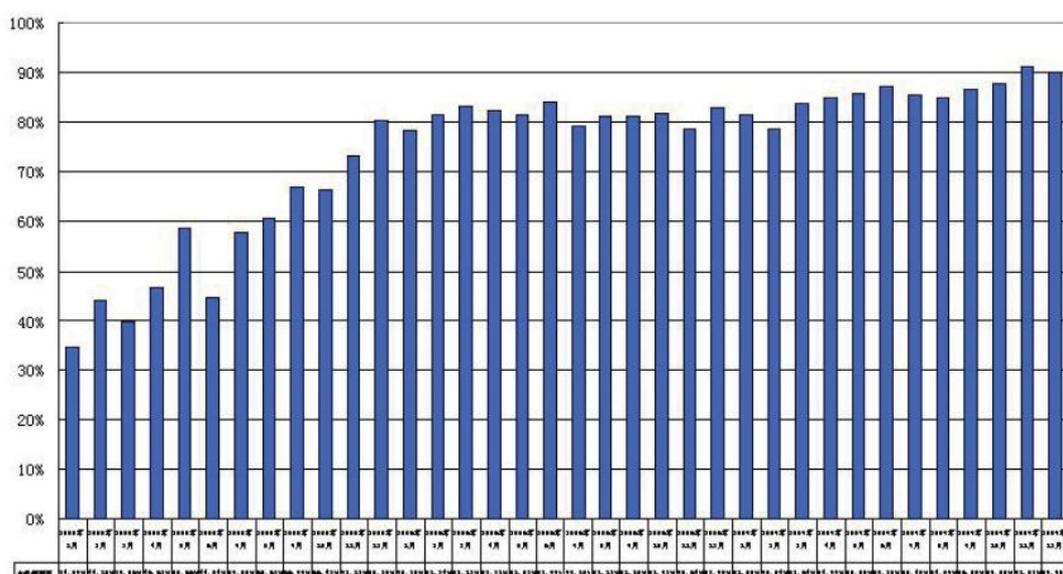


图 2-2 曙光 4000A 主机利用率（2005 年 1 月——2007 年 12 月）^[26]

传统高性能计算的高门槛和应用中的瓶颈已经越来越不能满足海量数据增长的需求，科研人员需要体积更小，成本更低的，应用更方便的计算设备。历史上任何技术的发展和进步，都是由用户和市场的内在需求驱动的，GPU 通用计算技术也正是在这种背景下诞生了。

2.2 计算机 GPU 发展的历史与趋势

可以这么说，未来科学和工程的突破在于计算！人们不遗余力的开发一代又一代的大型超级计算机就是为了满足这种不断膨胀的需求。在天文学领域，

海量数据处理是经常面对的一个重要问题，从现有的技术看，要提高运算的速度，一般通过以下几个方面的改进措施来进行：（1）采用新的处理速度更快的硬件设备，如更快的超级计算机，更大的内存，以及更快的 I/O 设备等。这是从根本上提升并行计算能力的途径。（2）更加优化的程序设计方法和函数库，如在程序中引入多线程、并行等处理方法。（3）采用优化的软件，这也是一种简便有效而且成本相对较低的方法^[27]。如何解决天文学中海量数据的处理，是困扰天文学家的重要课题，这也是我选择这个课题的原因，对于天文学家来讲，如果每一个人都能拥有一台属于自己的超级计算机，那科学生产力无疑将大大提高！

采用基于 CUDA 的 GPU 并行计算就属于上面介绍的第一和第二种方法的结合。CUDA（Compute Unified Device Architecture）是一个新的基础架构，是一个软硬件协同的完整的解决方案。这个架构可以使用 GPU 来处理复杂的科学计算问题，特别是极大数据量的并行计算问题。为什么 GPU 就具有这样的潜力呢？

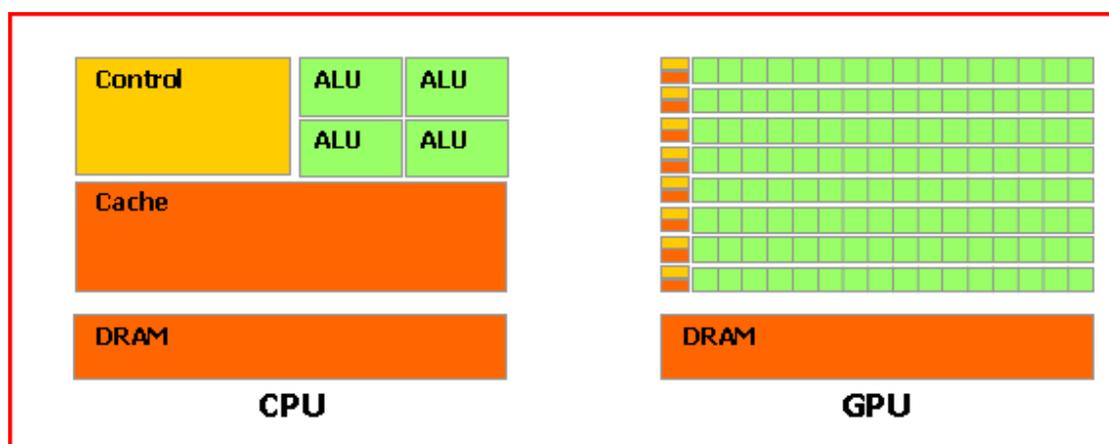


图 2-3 GPU 将更多晶体管用于数据处理^[28]

图 2-3 是典型的 GPU 和 CPU 芯片晶体管版图对比，从图中我们可以看到，CPU 中大部分晶体管主要用于构建 Control 控制电路(如：分支预测、决断等)和 Cache，只有少部分的晶体管来完成实际的运算工作(ALU)。GPU 的控制相对简单，Cache 较小，它通过一次载入成百上千个线程来隐藏这种数据传输的延迟，所以大部分晶体管可以组成各类专用电路(ALU)，多条流水线，使得 GPU 的计算速度有了突破性的飞跃、拥有了惊人的处理浮点运算的能力。现在 CPU 的

技术进步正在慢于摩尔定律，而 GPU 的运行速度已超过摩尔定律，每 6 个月其性能加倍。

GPU 发展的主要推动力是计算机在数据处理速度方面一直不能满足用户对于图形计算方面的要求。人们总是想得到更加绚烂的，更逼真的图形图像，解决这一问题的方法就是提高计算机处理器的运算精度和速度。在 GPU 诞生之前，人们将图形、视频等需要大量运算的功能都交与 CPU 完成。随着图像、视频质量的不断提高，庞大的图形数据造成了 CPU 的巨大负荷。为了满足海量数据的图形计算需求，显示芯片向高速、高集成度发展，当显示芯片实现的计算能力也无法满足快速增长的图形计算需求时，图形计算功能被脱离出来单独作为一个芯片设计。GPU(Graphic Process Unit 图形处理器)就是专门负责图形计算的处理器。最新的 Nvidia 公司生产的 GPU 包含了 240 个流处理器，在某些大规模并行计算的应用上，相对于 CPU 来说性能提高可达 100 倍以上^[29]。

从 GPU 的发展历史看，GPU 天生就适宜做并行计算，最新的 GPU 浮点计算能力已经大大超过了 CPU 和其他的处理器。但是在几年以前，GPU 还仅仅应用于图形计算领域，可编程性差，运算精度也不高，这对于计算资源来说是一个极大的浪费。为开发 GPU 在更多领域的应用潜力，人们开始研究如何能够利用 GPU 完成通常意义上的数据运算，这被称之为 GPGPU(General-Purpose computing on Graphic Process Unit, 基于 GPU 的通用计算)。

实现 GPU 的通用计算也经历了一个发展过程。第一代 GPU 的计算是基于矢量的运算(可以理解为基于数组的计算)，因此第一代 GPGPU 采取的技术方案是先将需要处理的计算问题转化为能够用矢量表示的问题，然后这些矢量通过 DirectX 或者 OpenGL 这样的图形 API 完成计算并取回结果。由于这些图形 API 会使用 GPU 来完成相应的计算，因此也就通过这种间接的方式实现了 GPGPU。但是这种方式存在着很大的不便。首先，这种间接实现方案大大增加了程序的复杂度。其次，即使要解决的问题可能根本就与图形无关，但是由于通用计算必须间接使用图形 API，因此就要求应用程序要依赖于这些图形库，开发人员也必须掌握相应的 API 函数。在这种情况下，GPU 的计算效率受限于 API，且开发效率较低，不能实现跨平台跨语言跨操作系统开发^[30]。

2.3 CUDA 技术的原理和特点

随着 GPU 制造技术的不断进步, GPU 本身的计算方式由基于矢量计算转为基于标量的并行计算。当摆脱了架构和计算方式所带来的限制之后, GPU 所能处理的问题由图形领域扩展到了通用计算领域。并且需要有一种比较灵活的开发方式,能够让用户直接使用的计算能力。

目前的 GPU 制造商有两个在制程工艺和市场占有率上处于绝对领先地位公司。一个是 AMD 公司(收购了 ATI 公司),还有一个是 Nvidia 公司。我们认为 Nvidia 公司通用计算 GPGPU 技术要比 ATI 略有领先一些,这主要基于以下几点:(1) Nvidia 公司的 CUDA 驱动较早的封装到显卡驱动中去,关于算法的资料也较多,比 AMD 公司要领先一步,市场占有率较高。对于计算机行业,市场的份额、占有率在很大程度上支持着产品提供商不断的改进优化就有的产品推出新产品。(2) 我们仔细分析了 AMD 和 Nvidia 公司的产品架构,他们的 GPGPU 采取了不同的体系结构,我们认为,相比较而言, Nvidia 公司的 GPGPU 体系结构 CUDA 有着更大的计算优势, GPU 芯片中的运算单元能全部有效的充分利用。而 AMD 公司的 GPGPU 尽管有着更多的运算核心,但是从结构上看依然没有摆脱原有图形运算单元的模式,运算单元的核心在一些情况下不能完全利用。

Nvidia—CUDA 是一个新的基础架构,这个架构可以使用 GPU 来解决商业、工业以及科学方面的复杂计算问题,它提供了硬件的直接访问接口,具备较灵活的开发方式,编程语言是基于 C 语言的拓展集合,使开发者能够在用户的强大计算能力的基础上建立起一种效率更高的密集数据计算解决方案,而不必像传统方式一样必须依赖图形 API 接口来实现 GPU 的访问,这大大降低了学习 CUDA 的门槛。CUDA 在架构上改变了传统 GPU 做法,它将像素着色器和顶点着色器制成统一渲染架构来使用 GPU 提供的硬件资源,从而给大规模的数据计算应用提供了一种比 CPU 更加强大的计算能力^[29]。

CUDA 的软件堆包含三个层次(如图 2-4 所示):设备驱动程序,应用程序编程接口(API)及其运行时库,以及两个较高级别的通用数学库,即 CUFFT 和 CUBLAS。CUFFT 和 CUBLAS 是基于 CUDA 技术所提供的应用开发库,分别称为离散快速傅立叶变换和离散基本线性计算库。这两个数学运算库所解

决的是典型的大规模的并行计算问题，也是在密集数据计算中非常常见的计算函数。开发人员在数学库的基础上可以快速、方便的建立起自己的计算应用。此外，用户也可以在 CUDA 技术基础上实现出更多的开发库。应用程序编程接口（API）及其运行时库提供了应用开发接口和运行期组件，包括基本数据类型的定义和各类计算、类型转换、内存管理、设备访问和执行调度等函数。设备驱动程序基本上可以理解为是 CUDA-Enable 的 GPU 的设备抽象层，提供硬件设备的抽象访问接口。CUDA 提供的应用程序编程接口（API）也是通过这一层来实现各种功能的^[31]。

基于 CUDA 开发的程序代码在实际执行中分为两种，一种是运行在 CPU 上的代码(Host Code)，这种代码的文件后缀是.c 或者.cpp 文件，它是整个架构的主程序，在我们的程序中采用微软公司提供的 cl.exe 编译器交给 CPU 处理；一种是运行在 GPU 上的设备代码(Device Code)，文件以.cu 后缀结尾，使用 Nvidia 公司提供的 nvcc.exe 编译器交给 GPU 处理。不同类型的代码由于其运行的物理位置不同，能够访问到的资源不同，因此对应的运行时组件也分为公共组件、宿主组件和设备组件三个部分。从以上的分析中，我们可以把设计的程序架构看成是“CPU+GPU”架构，GPU 作为 CPU 的协处理器，负责处理海量数据并行计算部分。

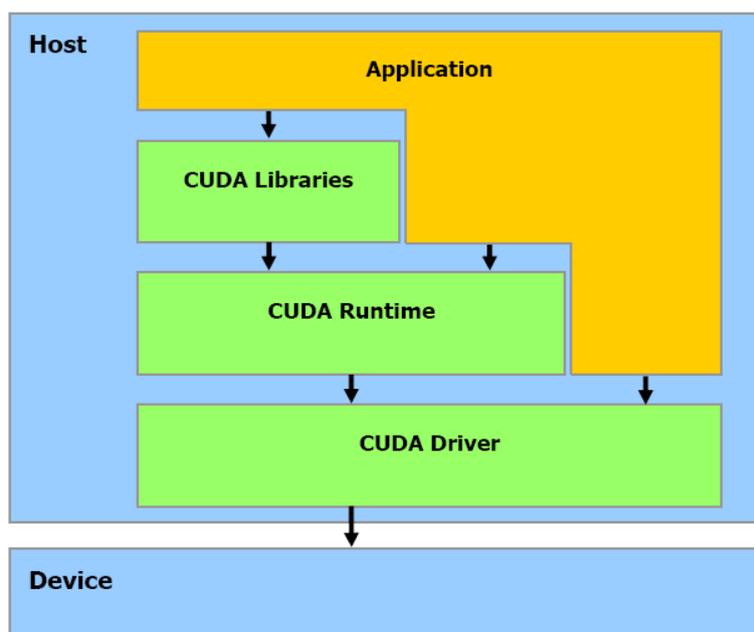


图 2-4 计算统一设备架构软件栈^[31]

2.4 天文上应用 CUDA 的可行性以及 CUDA 并行计算性能测试

2.4.1 天文学家采用 CUDA 技术进行并行计算有着巨大的需求

当天文学开始用 CCD 取代了照相底板的时候，就开始了天文仪器信息化的进程，来自于观测（数字化巡天）和模拟（比如宇宙结构形成、超新星爆发）的数据容量开始指数增长。现在全球的天文数据量已经达到了 1PB，而且还在以 2T/天的速度增长。数十 TB 的数据处理已经十分寻常，几 PB 的数据集很快就会出现。

天文学中的海量数据给计算机科学的发展提出了严峻的挑战。天文学家在处理海量数据的时候很多都涉及矩阵乘加运算，离散傅立叶变换（如图 2-5），三角函数等，比如在行星际模拟，巡天，实时图像处理等领域，太阳耀斑，超新星、伽玛爆、微引力透镜事件，行星掩食，星体耀变，超大质量黑洞的吸积耀变，快速移动天体等等的瞬变天象都需要能够快速反应的观测系统。有些重要的天象只会存在几秒，对它们后续特征的研究势必会拓展我们对宇宙的认识和理解，带来新的发现。这就需要快速的计算机处理系统！在国外 CUDA 技术已经被应用于天文领域^{[32][33]}，当然这种应用大多是实验性的，毕竟如上文所提到的 CUDA 技术刚刚兴起，其编译环境可调试性都不如人意，但是它有着巨大的潜力在于批量的并行计算，这成为天文学家解决长期以来计算机数据处理能力低的唯一选择。

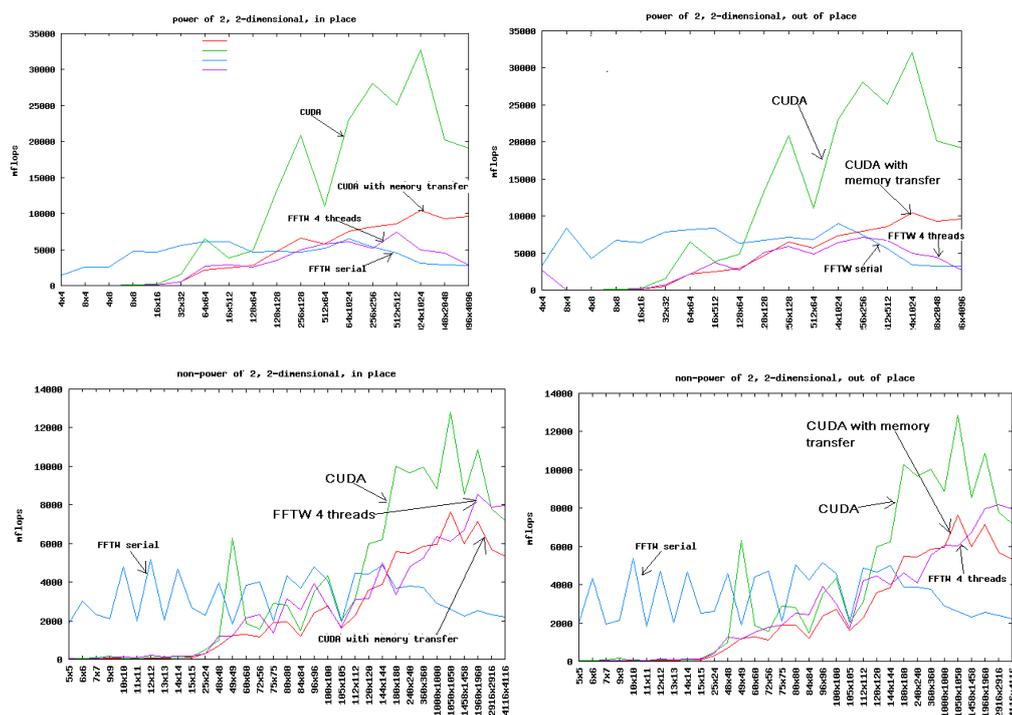


图 2-5 CUDA 与 FFTW 在不同数据量情况下的速度比较

可以看出数据量越大 CUDA 并行计算优势越明显！ [34]

2.4.2 GPU&CPU 并行计算性能测试基本配置

表 2-1 是 Nvidia 公司三大产品线支持可编程 GPU 的主要显卡列表。

NVIDIA Geforce 系列	Tesla 系列	Quadro 系列
Geforce GTX295	Tesla S1070	Quadro FX 5800
Geforce GTX285	Tesla C1060	Quadro FX 5600
Geforce GTX280	Tesla S870	Quadro FX 4800
Geforce GTX260	Tesla C870	Quadro FX 4700 X2
Geforce GTS250	Tesla D870	Quadro FX 4600
Geforce GTS150		Quadro FX 3700

表 2-1 Nvidia 公司主要产品线列表

目前 Nvidia 公司的产品除了 Tesla S1070, Tesla C1060 以外, 其他产品都只支持单精度浮点运算, 这对于大部分天文的需求来讲是可以接受的, 且随着通用 GPU 的不断发展支持双精度浮点运算的 GPU 会越来越多。

我们随机产生数据进行单精度浮点测试。测试的目的是确保数据计算的高速可靠性, 我们比较了 GPU 和 CPU 在计算矩阵乘法和快速傅立叶变换上性能的差异, 选取这两个进行比较有以下几个原因: (1) 矩阵乘法和快速傅立叶变换是天文学家进行数据处理常用的处理方法; (2) 这两项测试都能实现较大程度的并行性, 没有更多的分支、判断结构; (3) 现今在科学界最流行的 FFTW 使 Intel 公司的 CPU 在计算傅立叶变换时发挥极大性能, 这相对于 Nvidia 的 CUFFT 库有更好的可比性。

我们选用的是丽台公司 PX8800GT 显卡^[35], 该显卡采用 Geforce 8800GT 显示芯片。该芯片嵌入了 G92 显示核心, 112 个流处理器, 内核时钟为 600M, 内存带宽为 57.6GB/sec, 支持并行数据高速缓冲 (Parallel Data Cache), 与之相比较的 CPU 是 Intel Pentium D 830 3.0GH, 双核, 主频为 3.0GHz, 内存带宽为 4429.8MB/s, 峰值运算能力大约为 31.4GFLOPS, Simithfield 核心, LGA775 封装技术, Socket 775 插口, 制造工艺为 90nm, 前端总线为 800 MHz, 外频为 200MHz, 一级缓存为 12K μ Ops (微系统指令) +6KB (L1 Trace+L1 Data), 二级缓存为 1MB \times 2, 支持 SSE, SSE2, SSE3。该处理器的浮点运算能力基本代表 Intel 系列高端 CPU 的水平。

测试方法是通过 79M 的数据量对 Nvidia 的 GPU 核心芯片 (G92) 和 Intel Pentium D 830 进行矩阵乘法和快速傅里叶变换测试, 通过比较两者运算速度得出结论。

下面分别是测试环境下 GPU 和 CPU 的软件配置:

在 CUDA 的软件层面, NVIDIA 编译器 nvcc.exe 是其中的核心^[36]。CUDA 程序是 GPU 和 CPU 的混合编码, 其程序文件包括 .cu、.cpp 或 .c, 程序中混合着 C 语言函数和 CUDA 语言函数。程序首先通过 C 编译器编译将 GPU 和 CPU 的代码分离, GPU 代码被编译成能交给 GPU 计算的代码, 而 CPU 代码编译成标准 CPU 机器码。因此一个完整的 CUDA 软件开发环境需要有两个编译器, 一个是面向 CPU 的 C 编译器, 另一个是面向 GPU 的编译器 nvcc.exe。CUDA 可

以支持多种运行在 Windows XP 和 Linux 操作系统下的编译系统，在本次测试中，考虑到通用性，我们采用的是 Microsoft Visual Studio 2008^[38]，其 C 编译器为微软公司提供的 cl.exe。如图 2-6 是 CUDA 软件架构示意图。

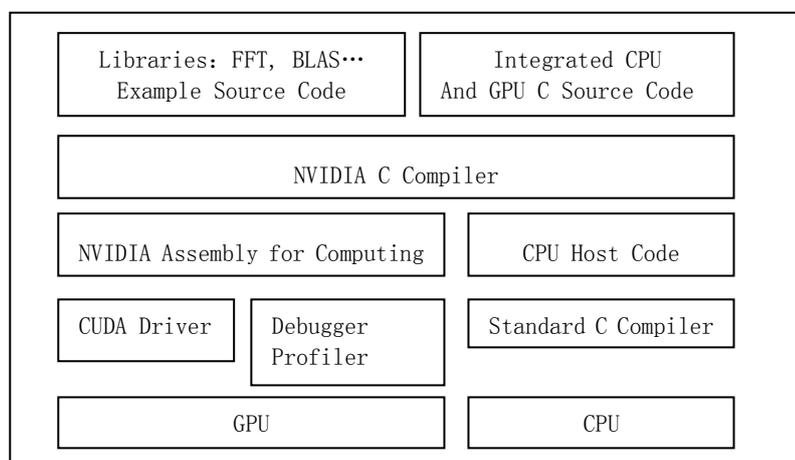


图 2-6 CUDA 软件架构示意图^[24]

从图 2-6 我们可以看到在 CUDA 软件堆的最上层，为我们提供了 CUBLAS (CUDA Basic Linear Algebra Subprograms, 简称 CUBLAS)^[40]和 CUFFT(CUDA Fast Fourier Transform)^[41]函数库，它是基于 NVIDIA CUDA 驱动程序上的子程序。可以访问 NVIDIA GPU 的计算资源。CUBLAS 和 CUFFT 函数库自身包含 API 层，可以不与 CUDA 驱动程序直接相互作用。这是经过 Nvidia 公司优化过的函数库，在进行大规模矩阵乘法运算和快速傅里叶变换的测试中，该函数库将提供 GPU 最极致的运算性能。

为配合 Intel CPU 的测试，我们选用的是英特尔® 数学核心函数库（英特尔® MKL）。该函数库是经过高度优化和广泛线程化的数学例程函数库，专为需要高性能的科学、工程、天体物理及分子动力模拟等领域的应用而设计。该函数库能针对不同的处理器采用不同的线程处理方式，使处理器的运算性能得以充分发挥。该函数库中包括了经过高度优化的 BLAS 例程，它们提供的性能改善非常显著，远远超出其他替代性实现^[37]，MKL BLAS 函数库能发挥 Pentium D830 处理器在矩阵运算中的最高运算能力。

我们采用 FFTW 进行快速傅里叶变换的测试。FFTW 是目前运行速度最快最流行的 FFT 软件包，该软件包用 C 语言开发，其核心技术（编码生成器）采

用面向对象设计方式和面向对象语言 Caml 编写。FFTW 能自动适应 CPU 环境，可移植性强，运行速度快，可进行一维和多维的离散傅里叶变换 (discrete Fourier transform)，其数据类型可以是实型，复型或半复型，用 FFTW 进行傅里叶变换能将 CPU 的运行能力发挥到极致^[39]。

测试包括三个指标：(1) 数据计算量 (2) 运算时间 (3) 浮点运算能力。选用这三个指标是考虑到从天文学家的角度看，这三个指标基本能体现出两种计算设备的性能差异。

2.4.3 大规模矩阵乘法性能测试

测试中采用了五种不同的程序写法：(A)基于 CUDA CUBLAS 函数库的程序代码；(B)基于 Nvidia CUDA 通用内核的程序代码；(C)基于 CUDA 驱动 API 的程序代码；(D)基于 Intel MKL 函数库的程序代码；(E)普通 C 语言代码。调用 srand^[38]，rand 函数产生一个行数为 5*512，列数为 3*512 的 A 矩阵，产生一个行数为 3*512，列数为 8*512 的 B 矩阵，将产生的随机数保存到.txt 文件下。五种不同的程序算法得出同样的正确的计算结果，把他们放到一个程序段中进行编译，测试程序分以下几步完成：

首先在 CPU 中建立存储空间，将.txt 文档中的数据调入主机 (host) 的内存中，在 GPU (device) 存储空间中建立矩阵和矢量对象，填入主机 (host) 内存中的数据，调用一连串的 CUBLAS 功能，最后返回计算结果到主机。CUBLAS 提供的辅助函数可以在 GPU 空间建立和结束对象，可以从这些对象处写入和重新得到数据^[40]。

(A)CUBLAS 测试代码

```
status = cublasInit();//初始化 CUBLAS 函数
status = cublasAlloc(size_A, sizeof(d_Ablas[0]), (void**)&d_Ablas);
status = cublasAlloc(size_B, sizeof(d_Bblas[0]), (void**)&d_Bblas);
status=cublasAlloc(size_Cblas,sizeof(d_Cblas[0]),(void**)&d_Cblas);
//在 GPU 上创建存储空间,d_Ablas, d_Bblas, d_Cblas 分别用于存放矩阵 A,
//矩阵 B, 和矩阵 A*B 的结果
cublasSgemm('n', 'n', 5*32*16, 8*32*16, 3*16*32, alpha, d_Ablas, HA, d_Bblas,
HB, beta, d_Cblas, HC); //单精度 BLAS3 矩阵乘法函数
cublasGetVector(size_Cblas, sizeof(h_Cblas[0]), d_Cblas, 1, h_Cblas, 1);
```

(B)GPU 通用内核测试

GPU 通用内核矩阵乘法包含两个文件：一个是 Mul.cu 文件(主程序文件)，另一个是 Mul_kernel.cu 文件(内核文件)。主文件调用内核文件用于 GPU 并行处理，内核文件将 A 和 B 划分成若干个小块，每个小块分别交给 GPU 不同的 Block 来处理，每个块大小为 32*32，也就是说每个 block 并行处理 32*32 个线程，运算速度大大加快。以下是并行运算代码：

```
float* d_A;
float* d_B; //在 GPU 上分配存储空间，用于存放 A,B 矩阵
CUDA_SAFE_CALL(cudaMalloc((void**)&d_A,mem_size_A));
CUDA_SAFE_CALL(cudaMalloc((void**)&d_B,mem_size_B));
CUDA_SAFE_CALL(cudaMemcpy(d_A,h_A,mem_size_A,cudaMemcpyHostToDevice));
CUDA_SAFE_CALL(cudaMemcpy(d_B,h_B,mem_size_B,cudaMemcpyHostToDevice));
//将 CPU 上的数组元素传入 GPU
dim3 threads(32, 32);
dim3 grid(WC / threads.x, HC / threads.y); //将 GPU 运算单元中分成 16*16 运算//块，需要分配的格点的坐标为 C 的列数和行数与 Block 大小的比值
Mul<<< grid, threads >>>(d_C, d_A, d_B, WA, WB); //执行矩阵乘法内核运算
for (int k = 0; k < BLOCK_SIZE; ++k)
    {Csub += AS(ty, k) * BS(k, tx);}
int c = wB * 32 * by + 32 * bx;
C[c + wB * ty + tx] = Csub; //矩阵乘法内核，返回计算结果到主机
```

(C)CUDA 驱动 API 性能测试

CUDA 驱动 API 程序的编程思路与内核程序类似，其主要思想是先在 CPU 上分配内存空间，再将 CPU 上的数据传入 GPU 中，再在 GPU 进行并行处理，但是驱动 API 在 CUDA 软件堆中属于较底层函数库，其编写起来较内核程序繁琐，但是效率更高。以下是 CUDA 驱动 API 运算部分的程序代码，其中 matrixMul 是内核执行程序，BLOCK_SIZE 是自定义线程块的大小：

```
CU_SAFE_CALL(cuFuncSetBlockShape(matrixMul,BLOCK_SIZE,BLOCK_SIZE,1));
CU_SAFE_CALL(cuFuncSetSharedSize(matrixMul,2*BLOCK_SIZE*BLOCK_SIZE*size
of(float)));
CU_SAFE_CALL(cuParamSeti( matrixMul,5*512*8*512,d_C ));
CU_SAFE_CALL(cuParamSeti( matrixMul,3*512*5*512,d_A ));
CU_SAFE_CALL(cuParamSeti( matrixMul,3*512*8*512,d_B ));
CU_SAFE_CALL(cuParamSeti( matrixMul,3*512,WA ));
CU_SAFE_CALL(cuParamSeti( matrixMul,8*512,WB ));
```

```
CU_SAFE_CALL(cuParamSetSize( matrixMul,20));
CU_SAFE_CALL(cuLaunchGrid( matrixMul, WC / BLOCK_SIZE, HC / BLOCK_SIZE));
```

(D) Intel MKI 矩阵乘法

Intel MKI 库函数内容极其丰富, 涵盖线性代数, 随机数产生, FFT 等方面, 为测试 CPU 的矩阵运算性能, 我们采用的是线性代数函数库 (BLAS), 该函数库包括 BLAS1, BLAS2, BLAS3 三种类型的运算^[37], 我们选用 BLAS3 (Matrix-To-Matrix) 进行运算。以下是 Intel MKL-blas 运算代码:

```
cblas_sgemm(CblasColMajor, CblasNoTrans, CblasNoTrans, HA, WB, HB, alpha, h_A, HA,
h_B, HB, beta, Cintelblas, HC); //矩阵 A 与矩阵 B 相乘
//并将结果返回到主机 Cintelblas 矩阵
```

编译 MKL 库函数, 需要在函数中添加: MKL.h 头文件, 并且要将应用程序的 lib 库指向 MKL 函数库, 连接器命令行添入 cublas.lib mkl_c.lib mkl_c_dll.lib mkl_intel_thread.lib, 为使测算函数运行时间的准确还需调用函数 QueryPerformanceFrequency(); QueryPerformanceCounter(), 来计算开销的 CPU 时钟周期数。

(E) 普通矩阵乘法

普通矩阵乘法代码就是采用普通的 C 循环语句做的矩阵乘法, 通过反复调用自定义函数实现矩阵乘法运算。以下是矩阵乘法代码:

```
Mul(float* C, const float* A, const float* B, unsigned int HA, unsigned int WA, unsigned
int WB)
{ for (unsigned int m = 0; m < HA; ++m)
    for (unsigned int n = 0; n < HB; ++n)
    { double sum = 0;
        for (unsigned int k = 0; k < WA; ++k)
        { double a = A[m * WA + k];
            double b = B[k * WB + n];
            sumC += a * b;
        }
    }
C[m * WB + n] = (float)sumC;
```

2.4.4 大规模快速傅立叶变换性能测试

这里主要包括三项测试指标: (1) 数据量大小 (2) 运算量 (3) 运算时间

测试中分别调用了(A)CUDA CUFFT 函数库的程序代码；(B)FFTW 函数库的程序代码。测试程序用矩阵乘法部分产生的数据，完成了数据的两次一维傅里叶变换，分以下几步完成：

(A) CUFFT 程序

首先需要将.txt 文档中的数据装入 GPU memory 中，定义数据为复数的内存空间：

```
Complex* h_signal = (Complex*)malloc(sizeof(Complex) * SIGNAL_SIZE);
//建立 plan,完成 FFT 变换
CUFFT_SAFE_CALL(cufftExecC2C(plan1, (cufftComplex *)d_signal, (cufftComplex *)d_signal, CUFFT_FORWARD));
CUFFT_SAFE_CALL(cufftExecC2C(plan2,(cufftComplex*)d_filter_kernel, (cufftComplex *)d_filter_kernel, CUFFT_FORWARD));
```

(B) FFTW 程序计算

FFTW 库函数程序首先需要产生相应的 libfftw3-3.lib libfftw3f-3.lib 或 libfftw3l-3.lib 文件，这些文件是编译文件调用的库函数，需要用户自己在 DOS 环境下编译生成^[39]。FFTW 库函数程序与 CUFFTW 库函数程序类似，其过程也是首先建立一个 plan，再完成傅里叶变换函数的调用，程序如下：

```
plan=fftw_plan_dft_1d(SIGNAL_SIZE,h_signal,h_signal,FFTW_FORWARD,FFTW_ESTIMATE);
plan=fftw_plan_dft_1d(FILTER_KERNEL_SIZE,h_filter_kernel,h_filter_kernel,FFTW_FORWARD,FFTW_ESTIMATE);
```

2.4.5 大规模矩阵乘法及快速傅立叶变换测试结果

图 2-7 是运算测试结果。



图 2-7 大数据量矩阵乘法性能测试结果

从图 2-7 可以看出，在 79M 数据量，32.2 Billion 运算量的情况下，我们得到以下结论：（1）利用 GPU 进行大规模矩阵乘法运算能力最强，CUBLAS 平均浮点运算能力能达到 530.28GFLOPS，利用通用内核矩阵和 CUDA 驱动程序（API）其浮点运算能力分别可以达到 444.299 GFLOPS 和 430.925 GFLOPS，而利用 CPU 最优化的函数库 Intel MKL 其浮点运算能力仅为 21.47GFLOPS，与 GPU 最优化的函数库 CUBLAS 相比相差 25 倍。（2）利用 GPU 进行快速傅里

叶变换所需要的处理时间为 64.27ms,而利用 FFTW 在 CPU 上计算相同的数据量处理时间为 1000ms。由此可见,在大规模并行运算方面通用 GPU 的运算能力更加强大。

第三章 基于 CUDA 的 35CM 磁场望远镜实时相关终端系统

如前文所述，太阳本身进入望远镜的与磁场有关的有效光量是非常微弱和有限的，这就需要 CCD 长时间积分，或者将多帧叠加。实时相关的作用就是可以矫正望远镜跟踪误差，将每一幅需要叠加的图像对齐后再叠加。从怀柔多通道望远镜的紧凑结构出发，我们选用的终端计算设备应该满足以下几点：(1) 利用怀柔基地现有的 CCD 采集卡，需要将数据采集卡和数据计算设备集成到一块母板上，并有高速的数据传输接口；(2) 数据计算设备应当具备在相机极大帧速率情况下，实时数据处理的能力。

3.1 基于 CUDA 的 Tesla C1060 高性能计算设备

Nvidia 公司的显卡有三个产品线：Geforce、Quadro 和 Tesla 系列，目前市场上这三个系列的产品都支持 CUDA 并行计算，但是在功能上各有侧重。

Geforce 产品主要是面向消费级的用户，主要是提供对游戏、影音、商用图形的良好支持。而 Quadro 系列显卡则针对专业图形设计用户，而且 Quadro 显卡顾名思义是安装在图形工作站平台内的，在点线硬件抗锯齿功能、OpenGL 硬件逻辑操作、重叠图形处理等方面有很大优势。用户定位不同，导致工作站显卡和消费娱乐显卡在品质精度、软件驱动和价格等方面有着较大的差别^[42]。

Tesla 是 Nvidia 公司为高性能计算专门开发出来的产品系列，包括 Tesla C870, Tesla S870, Tesla D870, Tesla C1060, Tesla S1060。其中 Tesla C1060, Tesla S1060 支持双精度浮点运算。Tesla 产品的优势在于质量过硬，计算准确性高。前面提到的 Geforce 和 Quadro 系列产品由于主要面向的用户不是用于高性能计算，因此在计算精度和图像计算的准确性上较 Tesla 产品更低^[42]。

我们经过反复的比较考虑，选用的是 Tesla C1060 处理器，从天文学的需求出发，双精度的浮点运算尽管在实时相关计算中不需要，但是在其他领域是必需的，另外 Tesla C1060 可以和 CCD 的图像采集卡集成在一块主板上，数据传输速率更高，质量也有保证，也节省了工程人员的开发成本。图 3-1 是 Tesla C1060 产品的技术指标。

Form Factor	10.5" x 4.376", Dual Slot
# of Tesla GPUs	1
# of Streaming Processor Cores	240
Frequency of processor cores	1.3GHz
Single Precision floating point performance (peak)	933
Double Precision floating point performance (peak)	78
Floating Point Precision	IEEE 754 single & double
Total Dedicated Memory	4GB GDDR3
Memory Speed	800MHz
Memory Interface	512-bit
Memory Bandwidth	102GB/sec
Max Power Consumption	200 W peak, 160 W typical
System Interface	PCIe x16
Auxiliary Power Connectors	6-pin & 8-pin
Thermal Solution	Active fan sink
Programming environment	CUDA

图 3-1 Tesla C1060 性能指标^[43]

我们采用 Nvidia 公司推荐的 DELL T7400 图形工作站做为搭载显卡的工作站，表 3-1 是 DELL T7400 图形工作站的技术指标。

表 3-1 DELL T7400 图形工作站参数^[44]

Dell T7400 WorkStation	
CPU	XeonE5430 (2.66GHz/1333MHz/12MB)
Graphic Card	256M Nvidia NVS 290
RAM	2G
HDD Hard Disk Driver	7200rpm/SATA2/320G
Operation System	Windows XP-Pro 32bit
	DVD-RW

选用 DELL T7400 图形工作站作为搭载 Tesla 计算卡的主机，主要考虑到观

测系统的需要。首先观测系统除了安装基本的板卡(网卡、显卡)之外,需要安装图像采集卡、数字 IO 卡,并且考虑到今后三通道望远镜观测终端也集成到该系统中,因此,该系统需要极大的扩展能力。Dell T7400 WorkStation 具有 7 个可扩展插槽。1 个 PCI-e x8 插槽,布线 x4; 2 个 PCI-e x16 Gen 2 显卡插槽; 3 个 PCI-X 64bit/100MHz 插槽,支持 3.3v 或通用卡; 1 个 PCI 32bit/33Mhz 插槽,四核 CPU 具有极强的处理能力^[40]。

安装的操作系统是 Nvidia 公司推荐的 Windows XP—Pro 32bit 操作系统。

3.2 35CM 磁场望远镜实时相关终端系统

前文中,我们着重讨论并搭建了基于 CUDA 的并行计算系统,基于这些研究,我们设计了一个用于怀柔 35CM 太阳磁场望远镜上的终端设备并已投入常规观测。本部分将对这一系统进行介绍。

3.2.1 相关函数在天文观测中的应用

假设 $g(x,y),h(x,y)$ 是两幅要进行相关的图像,即能量有限的二维离散函数。图像的大小分别是 $A \times B, C \times D$, 两者的相关函数的定义如下:

$$R_{fg}(m,n) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} h(x,y)g(x-m,y-n) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} g(x,y)h(x+m,y+n) \quad (1)$$

上面公式中 $m=0,1,2,\dots, M-1; n=0,1,2,\dots, N-1; M=A+C-1; N=B+D-1$ 。

从上面的相关函数的定义式中可以证明:

$$F\{R_{fg}(m,n)\} = F\{h(x,y)\} \times F^*\{g(x,y)\} = H(u,v) \times G^*(u,v) = R(u,v) \quad (2)$$

在公式(2)中 $H(u,v)$ 是 $h(x,y)$ 的傅立叶变换形式, $G^*(x,y)$ 是 $g(x,y)$ 的傅立叶变换的共轭。由相关函数在频域运算的形式推知,可以通过快速傅立叶变换算法实现相关运算,在此处只需要对 $G(x,y)$ 取共轭,得到 $G^*(x,y)$, 然后与 $H(u,v)$ 进行点乘,即可获得相关函数的傅立叶变换 $R(u,v)$ 。在 $R(u,v)$ 的逆变换矩阵(相关矩阵)中的最大值元素所在的位置即是两幅图像相关性最强的位置^[45]。

由于系统处理的图像仅存在微小的线性平移,因此,对目标图像从相关的最大位置开始进行线性移位必然可以实现其与基准图像的跟踪对齐。经过移位对齐后的两幅图像进行叠加,由于消除了两者之间的位置交错,从而可以更好地保持两幅图像的细节,实现分辨率的提高。

对太阳而言,其可观测的微弱磁场大约为几高斯,空间分辨率的范围在几角秒之间。要得到几高斯的灵敏度,目前国家天文台怀柔太阳站(HSOS)的矢量视频磁场仪至少需要在一个极性的方向上叠加数百帧。在这种观测模式下,图像的空间分辨率是由地球大气的抖动情况和望远镜自身的跟踪精度决定的。视宁度情况可以由 Fried 参数确定,对于可见光太阳观测来说,世界上多数观测站的 Fried 参数的均值都不会好于 $2''$ [46]。另一方面,风也是长时间叠加观测中的一个极有害的因素,因为世界上多数的太阳观测站不是位于高山上,就是坐落在水边。在这样的地点,即使望远镜自身的跟踪精度很高,因为风的原因造成的太阳图像移动也是在所难免的,这将大大降低长时叠加观测的空间分辨率。基于上面的原因,太阳深度叠加观测甚至无法得到几高斯的分辨率,相反这样差的分辨率甚至影响微弱磁场的探测。

相关跟踪技术是消除图像线性移位的有效方法,在上面的例子中线性移位主要来自风,跟踪误差和大气抖动的一阶成分。这就意味着相关跟踪技术可以以优于 Fried 参数的分辨率获得图像。在过去的三十年相关技术已经被应用于太阳观测的多领域,例如:相关跟踪器、自适应光学(Von der Luhe, 1992)。虽然这些技术是非常有效的,但它们并不能直接应用于怀柔站的矢量磁场测量仪,因为这些技术要求在望远镜系统中做大的硬件方面的调整,而怀柔多通道望远镜可利用空间小,结构紧凑,不太可能在硬件上做更多的改造。

因此可以考虑的方法是从软件着手。用软件实现相关也经历了一个发展过程。怀柔太阳基地原有的技术方式是将采集的图像存到硬盘上,等待观测完成后,成批处理实现相关-移位-叠加的过程[46][47]。这样的相关过程是非实时的,这样的系统存在几个问题:(1)是工作量大,耗时,需要投入一定的人力去完成这项工作(2)不同的人不同情况下选定的相关窗口不一样,图像对齐校正的系统误差也不一样。

用软件实现实时相关的瓶颈在于数据量大,对于一幅 992×1000 像素的

CCD 采集图像，每一个像素需要完成一次傅立叶变换，一次乘法，一次傅立叶反变换，一次求极值查找。目前性能最佳的英特尔至强 CPU 还不能在相机曝光时间内完成如此大数据量的计算。在这种情况下，怀柔基地发展出基于小窗口的实时相关技术，从日面图像中心裁剪出不大于 128×128 Pixel 大小的图像，用这个图像实现实时的相关-移位-叠加^{[48] [52]}，但是这种方式选取的窗口小且具有随意性，程序运行慢，仍然较难满足常规观测的需要。

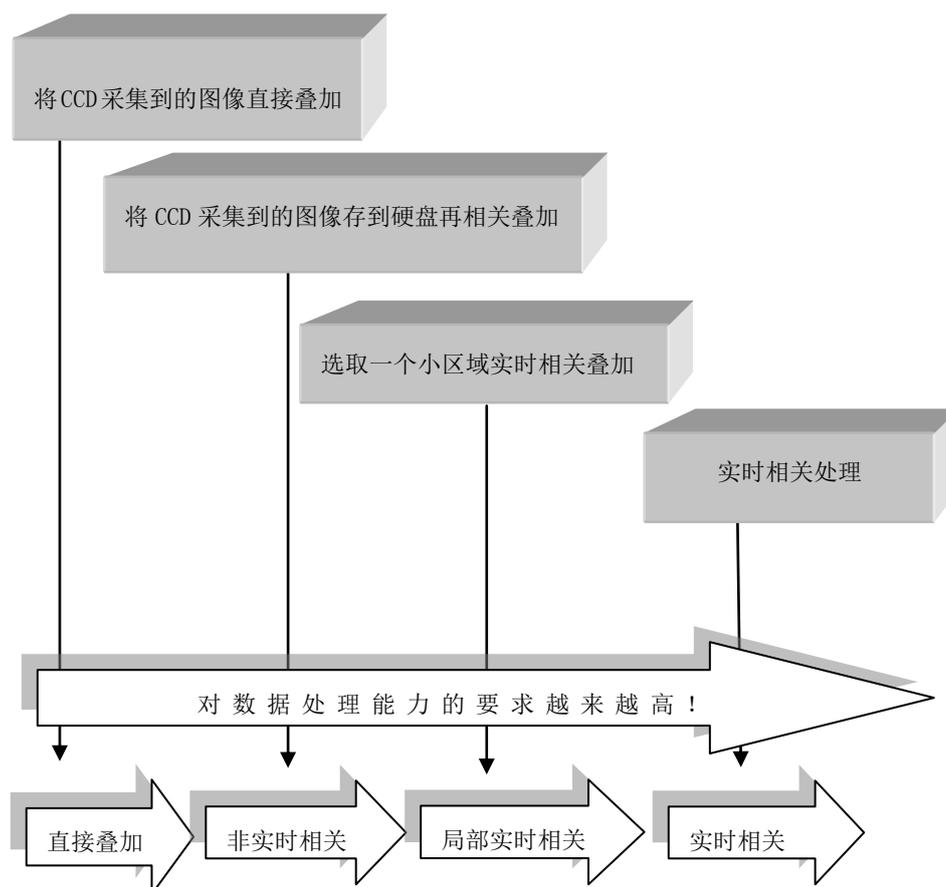


图 3-2 太阳望远镜图像相关系统发展过程

对于每一个像素都要完成相关运算，用 CPU 顺序执行的体系结构难以满足实时观测的要求。采用 GPU 并行计算可以实现在一个单位时间内完成成百上千次运算的目标。

并行计算在计算机领域是一门非常重要的学科，随着计算机 CPU 多核化的不断发展，并行计算获得了前所未有的重视，参考文献【27】较为全面地综述了并行计算在各个方面的最新进展。从现有的条件看，目前最高端图形工作站

的代表 DELL T7400 可以放入 8 个 CPU 核，一个时间单元可以启动 8 个线程。尽管如此，这仍然不能满足实时相关的需求。

根据前面的描述，在 TESLA C1060 数据计算设备中，我们一次可以启动 512×512 个线程（我们的相机是 992×1000 Pixel），在一个单位时间内完成巨大的数据处理量。

3.2.2 基于 CUDA 的相关算法

相关跟踪算法流程如下：

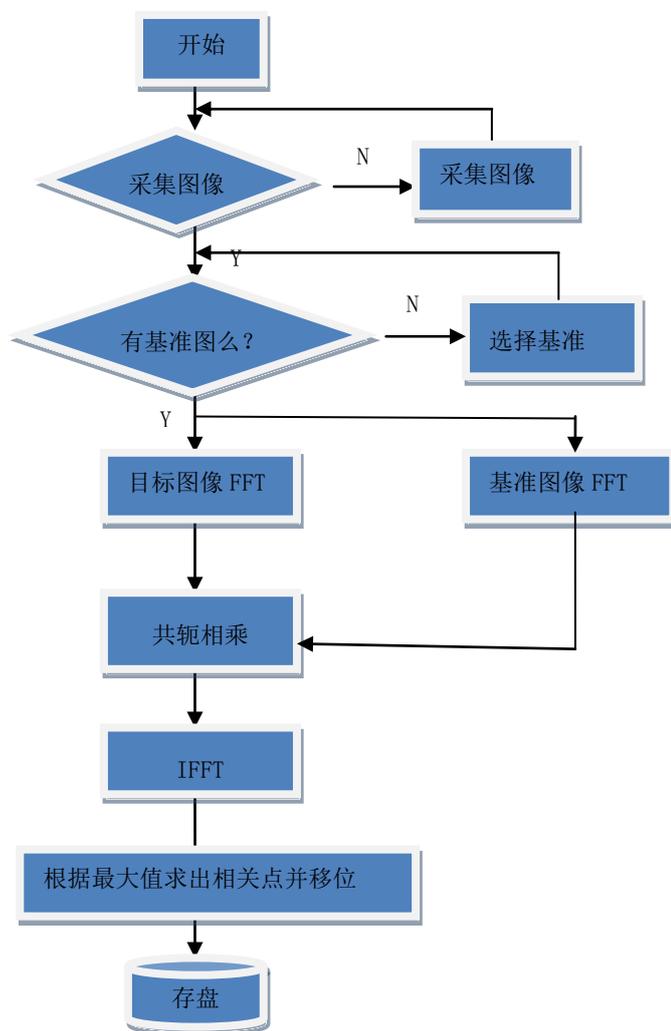


图 3-3 实时相关算法流程

从上图中我们可以看到，对于相关跟踪算法分为以下四个步骤：

1. 目标图像傅立叶变换，求共轭

2. 与基准图像进行点乘
3. 求 IFFT 变换
4. 求所得数组的最大值

要实现实时的相关处理，上面几个步骤必须在有限的时间内完成。新一代终端系统选用 992*1000 Pixel 的百万像素 CCD，在双通道情况下最大帧速率为 48 F/s,平均每一帧处理时间约为 20 毫秒^[49]，上述 5 个步骤所需要的时间不得超过 20 毫秒。因此我们测试和比较了当今最流行了基于 Intel 志强 CPU 处理器的 FFTW 库和基于 CUDA 的 CUFFT 库完成实时相关的情况。如下图所示：左边是 FFTW 完成实时相关需要的 7 个步骤，右边是 CUFFT 完成实时相关需要的 9 个步骤。从下图可以看出，CUFFT 和 FFTW 相比，多出来的两个步骤是创建 GPU 显存数组，并将内存中的数据拷贝到 GPU 显存中去。

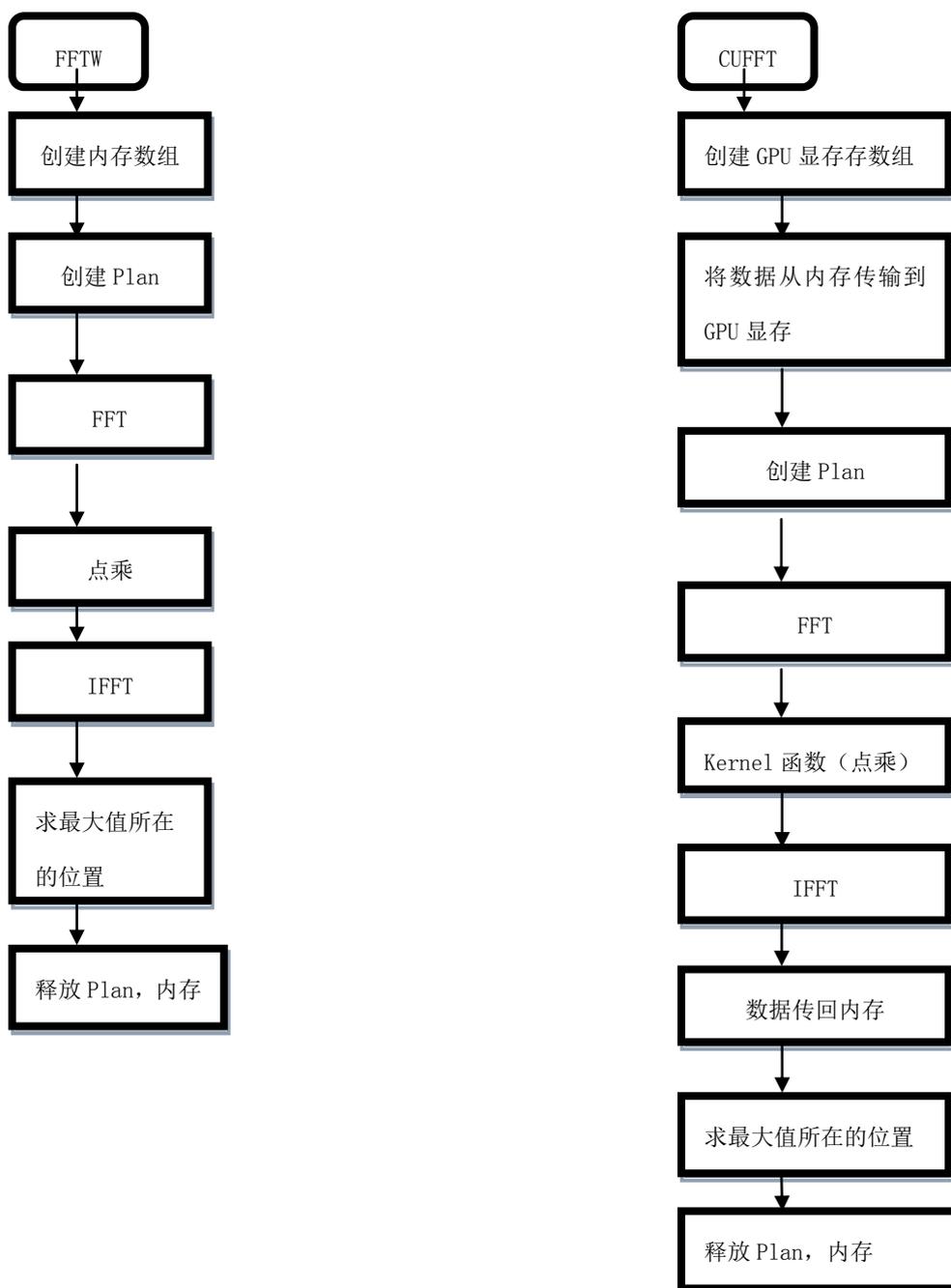


图 3-4 GPU 和 CPU 相关算法流程对比

FFTW:

表 3-2 FFTW 在不同窗口大小下时间开销列表

窗口大小	128×128	256×256	512×512	768×768	992×1000	1024×1024
整型数组强制转换浮点型数组	3.3ms	4.8ms	12.4ms	15.6ms	16.1ms	25.3ms
创建内存数组	1ms	1ms	2.0ms	2.6ms	3.2ms	4ms
创建 Plan	0.5ms	0.5ms	0.5ms	0.5ms	0.5ms	0.5ms
FFT	3.1ms	4.1ms	16.5ms	30.8ms	37.8ms	30.9ms
点乘	2ms	3.7ms	20.7ms	25.3ms	28.1	32.5ms
IFFT	2.8ms	4.5ms	15.8ms	14.3ms	39.2ms	31.8ms
求最大值所在位置	1.5ms	2ms	15.3ms	20.1ms	37.9ms	40.8ms
释放 Plan,内存	0.5ms	0.5ms	0.5ms	0.5ms	0.5ms	0.5ms
总计耗时:	14.7ms	21.1ms	83.7ms	109.7ms	173.3ms	166.3ms

CUFFT:

表 3-3 CUFFT 在不同窗口大小下时间开销列表

窗口大小	128×128	256×256	512×512	768×768	992×1000	1024×1024
整型数组强制转换浮点型数组	0.2ms	0.5ms	1ms	1.5ms	2ms	2ms
创建 GPU 显存数组	0.1ms	0.2ms	0.5	0.814ms	0.814ms	1ms
将数据从内存传送到 GPU 显存	0.2ms	0.5ms	1ms	2ms	2ms	2ms
创建 Plan+ 释放 Plan	0.5-0.6ms	0.5-0.6ms	0.5-0.6ms	0.5-0.6ms	0.5-0.6ms	0.5-0.6ms
FFT	0.6ms	1ms	1ms	2ms	3.7ms	2ms
Kernel 函数 (点乘)	0.4ms	0.2ms	0.4ms	0.5ms	0.4ms	0.5ms
IFFT	0.6ms	1ms	1ms	2ms	3.8ms	2ms
求最大值所在位置	0.1ms	0.1ms	0.5ms	5.2ms	3.1ms	6.2ms
总计耗时:	2.7ms	4.0ms	5.9ms	14.514ms	16.4ms	16.2ms

FFTW 是由麻省理工学院计算机科学实验室两位博士生 Matteo Frigo 及 Steven G. Johnson 在 1997 年所发明的，是计算离散 Fourier 变换 (DFT) 的快速 C 程序的一个完整集合。它可以计算一维或多维、实数或复数以及任意规模的 DFT。FFTW 还包含对共享和分布式存储系统的并行变换，它是当今最流行的快速 Fourier 变换函数库。根据 FFTW 开发手册，为极大的发挥 FFTW 和

CUFFT 的变换效率，我们裁剪的图像大小都是 2 的指数次幂。FFTW 还支持维度系数为 3, 5, 7 质数的编译^[39]，由于完整图像的大小是 992×1000 Pixel，因此我们选择了裁剪窗口 768×768 Pixel ($768 = 3 \times 2^8$)。

从上述的两张表中我们可以看出：FFTW 在傅立叶变换和反变换以及点乘上的耗时明显超过 CUFFT，尽管 CUFFT 要比 FFTW 多出“创建显存”和“传输数据到显存”两个步骤，但是 CUFFT 多线程隐藏了这两个步骤的时间延迟。在不同的图像尺寸下，CUFFT 运算速度是 FFTW 的 2 倍以上，而且数据量越大差异越明显。综合考虑时间因素(如前文所述，根据相机帧速率的计算，相关算法的时间不能超过 20ms)并且尽量选择较大的相关窗口，我们选取 512×512 作为图像窗口的大小。

我们将“目标图像 FFT”，“共轭相乘”，“IFFT”，“根据最大值求出相关点并移位”四个部分写入到一个 .cu 文件的主程序中，这样做主要是出于以下几点考虑：

(1) 可移植性。可移植性是衡量程序质量的重要指标之一，在怀柔太阳基地除了 35 厘米磁场望远镜，还有三通道望远镜，小磁场望远镜等，这些望远镜的计算机体系结构或者配置不同，很容易造成程序之间不兼容。将相关跟踪算法单独写入到文件中，可以不用修改或者只需要较少的修改即可在其他种类的计算机上运行，这样也有利于怀柔观测基地的技术积累

(2) 易于调试。调试是程序设计中的重要环节。好的程序要采用简单易懂的代码，同时要实现功能划分。将相关跟踪算法单独写入到 .cu 文件中，可以尽量精简不必要的代码，用 NVIDIA 公司提供的 NVCC 编译器单独调试程序文件，及时快速的发现错误。

3.2.3 终端硬件系统同步信号设计

在地基太阳磁场望远镜中，很多偏振分析器采用电光晶体 KD*P 作为调制元件，它的调制速度很高，可以在很短的时间内完成两个正交偏振分量之间的切换，以降低地球大气的影响^[50]。国家天文台怀柔太阳基地 35 厘米磁场望远镜是通过采集同步机构和 KD*P 高压控制机构控制晶体，以实现不同波段的观测。

3.2.3.1 终端硬件系统高压控制机构设计

怀柔太阳观测基地 35 厘米磁场望远镜有三组高压，分别是：500 伏，890 伏和 980 伏。5324 和 4861 波段分别观测太阳光球和色球，Q 和 U 代表横场，L 代表纵场，V 代表速度场。通过不同的选通控制信号选择高压。

下面是不同观测波段下的继电器组合：

表 3-4 高压组合控制逻辑

观测内容	第一组继电器 (500 伏)	第二组继电器 (890 伏)	第三组继电器 (980 伏)
5324+L			√
5324+Q			√
5324+U			√
5324+V	√	√	
4861+L		√	
4861+V	√		

从上图我们可以画出一个状态图，事实上一共有六个状态，用 1101 (L5,Q5,U5)，0101(L4)，1100(V5)，0011(V4)选通信号。

我们选用 PCI1730^[51]发送高压控制信号的板卡。以下是 PCI1730 的主要技术指标：

- 32 路隔离 DIO 通道（16 路输入和 16 路输出）
- 32 路 TTL 电平 DIO 通道（16 路输入和 16 路输出）
- 高输出驱动能力
- 中断能力
- 2 个 20 针接口用于隔离数字量 I/O 通道，2 个接口用于 TTL 数字量 I/O 通道

- 用于隔离输入和输出通道的 D 型接口

选用 PCI1700 基于以下几点考虑:

(1): 光耦和。可以很好的隔离千伏高压信号和计算机板卡

(2): PCI1730 一共 16 路数据输出, 具有较大的扩展性, 便于今后使用更多的控制信号控制观测波段。

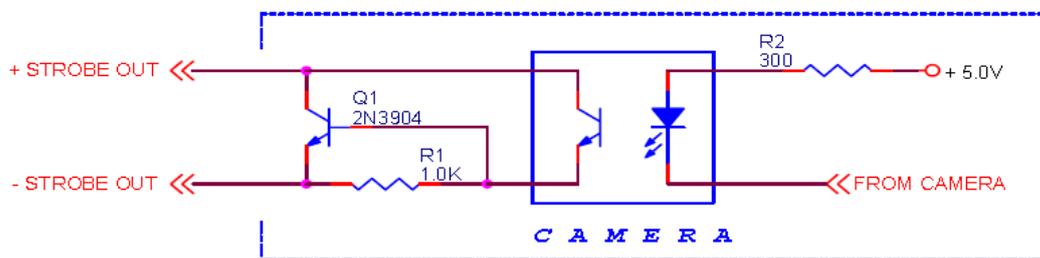
在观测程序中, 添加库文件 Adsapi32.lib, 调用 I/O 函数发送高压组合逻辑:

```
static PT_DioWritePortByte ptDioWritePortByte;
static PT_DioReadPortByte ptDioReadPortByte;
ptDioWritePortByte.port = m_iChannel;
ptDioWritePortByte.mask = 0xff;
ptDioWritePortByte.state = 0x68//向端口发送1101000, 选择高压, 观测内容 '5324' //&&
'L'
ErrCde = DRV_DioWritePortByte (DriverHandle,(LPT_DioWritePortByte)
&ptDioWritePortByte;
```

3.2.3.2 终端硬件系统采集同步机构设计

采集同步机构设计的目的在于使得相机曝光和高压翻转信号同步, 同步的要求就是信号翻转在曝光之前或之后发生, 这也是 CCD 应用中特殊的地方, 因为对于普通的工业用 CCD 相机, Strobe (闪光灯) 信号总是和曝光信号同步的, 正因为这种特殊的天文需求, 我们在同步信号控制机制上花费了不少的功夫。相机没有提供对 Strobe 信号进行测试的工具, 造成了我们测试中的困难。为完成 Strobe 信号的调试, 我们设计了一个实验来验证 Strobe 信号和曝光信号的时序关系, 实验的基本思路是: 搭建电路, 用相机的 Strobe(闪光灯)信号点亮一个 LED 管, 同时相机对准 LED 管进行拍摄, 假设 Strobe 信号是在曝光信号之后或者之前发生, 那么相机曝光的一瞬间 LED 是没有被点亮的, CCD 拍摄出来的 LED 管也是不亮的; 反之, CCD 拍摄出来的 LED 管是亮的。由于, 我们的需求是在曝光之前状态发生改变, 因此我们需要调整 Strobe 信号, 使得 CCD 拍摄出来的 LED 是不亮的。

通过查找相机内部电路资料, 我们可以画出相机 IMPERX IPX-1M48-L 的 Strobe 信号在相机内部的连接图。

图 3-5 Strobe 信号内部逻辑连接图^[49]

如图 3-5 可以看出, Strobe 信号是通过一个光耦驱动三极管 2N3904 工作的, 如果没有上拉电阻, 信号产生的电荷将会聚集在三极管 2N3904 上, 而示波器不会有信号输出。为解决这个问题, 我们加入上拉电阻, 用 Strobe 端子点亮 LED。如下图:

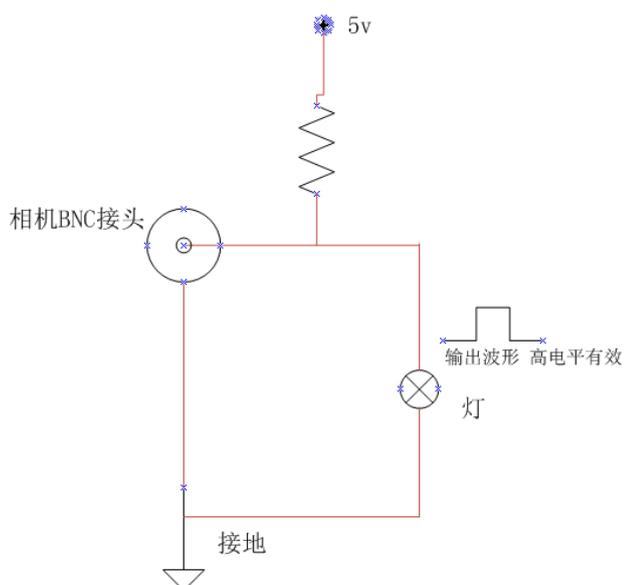


图 3-6 Strobe 信号外部逻辑连接图

按照图 3-6 的设计我们测试了发光管。发光管光线很暗, CCD 无法看到发光管发亮, 通过示波器发现电压仅为 0.8v, 这是由于相机内部三极管 2N3904 驱动能力不够造成的。为此我们在输出端增加了 IRF020 MOSFET 组成共源级放大电路^[53], 加入共源级放大电路后电路驱动 LED 能力明显提高, LED 发光能够被 CCD 清晰的看到。

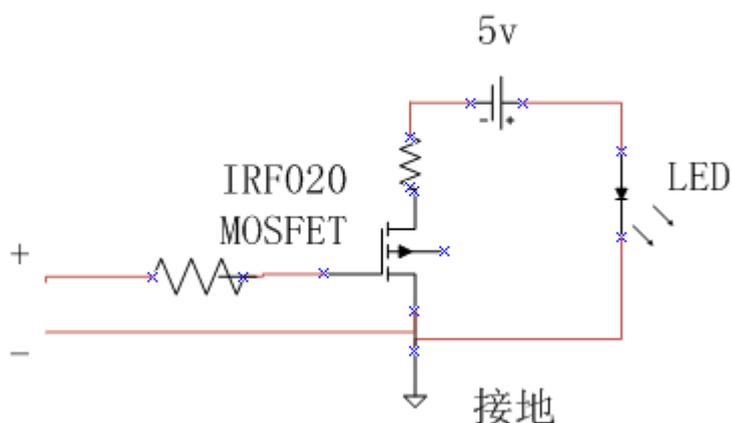


图 3-7 共源级驱动电路^[53]

IMPERX IPX-1M48-L 相机提供了丰富的相机命令参数，可以根据用户需要控制相机的初始化，曝光时间，帧速率，Strobe 信号位置等，我们发送 ssp 命令（控制 Strobe 信号位置的命令）调整 Strobe 信号相对于曝光信号的位置，同时用 CCD 对准 LED。当 LED 发亮，而 CCD 拍摄到的 LED 逐渐变暗直到消失，那么这正好是在曝光完成的时候 Strobe 信号发生了！



图 3-8 Strobe 信号测试实验电路

图 3-8 是整个实验的情况，上左图是用相机拍摄 LED 管在计算机上显示的情况，上右图是 LED 管被点亮以后，示波器显示 LED 输出波形。

采集同步机构的设计需要将确定的 Strobe 信号和图像采集信号同步起来。图像采集每完成一帧信号翻转一次再与 Strobe 信号整合成占空比为 50% 的信号，控制高压翻转，这样就能确保每次高压翻转和图像采集同时完成。为此我们设计电路，制作控制盒。

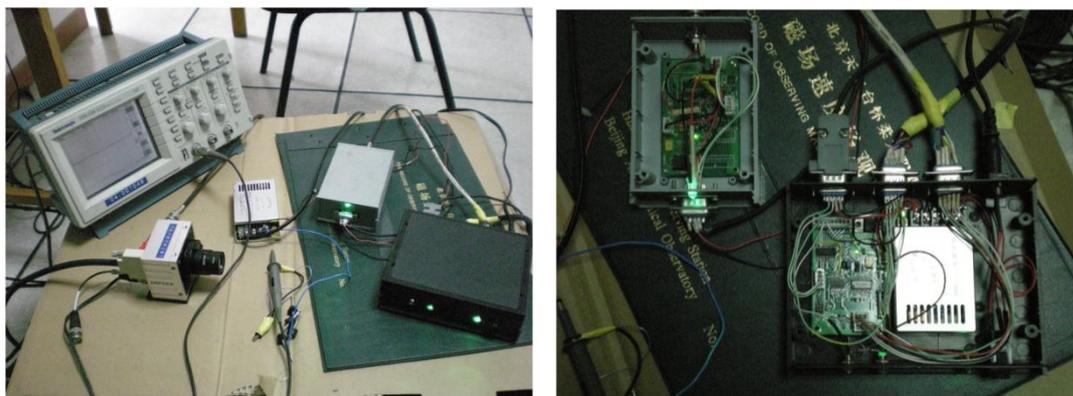


图 3-9 同步信号控制盒

如图 3-9，左图是测试时的情景，左图中有两个控制盒，分别是同步控制盒（黑盒子）和 Strobe 信号复原器（白盒子），同步控制盒同步 Strobe 信号和图像采集信号，它的左边是专为 Strobe 信号设计的复原器，由于 Strobe 信号从望远镜端传输到计算机有较远的传输距离，存在较大的信号干扰，Strobe 信号复原器可以中继信号，并将信号复原传输给同步控制盒。右图是同步控制盒和 Strobe 信号复原器内部电路板。

3.2.4 基于 CUDA 的终端系统用户界面设计

这一部分主要阐述新一代终端处理系统的软件设计部分。在 CUDA 编译环境下编写相关函数代码，做为 kernel 函数并行处理数千个线程。

观测程序的设计必须符合观测需求，基于这个出发点，我们在软件设计中充分的考虑了天文学的需求，并且重新设计了新的用户界面。如图 3-10:



图 3-10 观测程序整体界面设计

用户界面的基本思路是把日常观测中常用的参数放到左边的可浮动工具栏中，便于操作使用，如：宁静度，清晰度，定标系数，像质等，这些都是生成 Fits 文件需要的参数。最下边的“更新”按钮将会保存“波长”和“观测内容”（(5173、4861 和 L、Q、U、V) 一共 16 种组合）下的各种参数。这样设计的目的是由于 FITS 文件的参数较多，如果观测员在每次观测一个活动区的时候都要填入所有的参数将会使观测工作变得繁琐，因此我们在设计中把第一次填入的 16 组组合对应的参数保存在 Access 数据库中，在第二次观测时就可以直接调用，而不需要全部重新填入或者只需要修改其中较少的几个参数，然后“更新”即可！

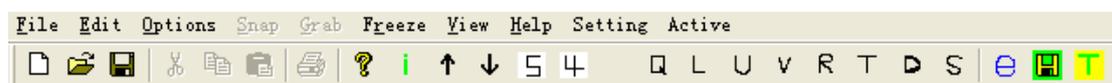


图 3-11 观测程序工具栏快捷方式设计

在工具栏中有“5173，4861”和“L、Q、U、V”快捷方式，他们不同的组合产生不同的参数值，以供观测员使用和修改。

活动区

日期: 2009- 3-19

时间: 19:06:58

X: 207.911

Y: -9.283

HSOS Number: 09011

AR Number: NA

下一条

上一条

到最后

添加

更新

Decarl

Carrington

Global

确定

取消

图 3-12 观测程序添加活动区界面设计

添加活动区部分观测员可以填入活动区号、日期、时间，并且自动把每一个活动区的相关信息保存在 Access 数据库中，在程序左边栏可以自动显示并计算出每一个活动区的卡林顿坐标，笛卡尔坐标以及 global 坐标，便于观测员选择使用。

程序接受用户的输入，把用户输入的命令编译解释后执行，执行单元分成三部分：采集图像，命令参数，数据库管理。下面简单介绍一下这三部分：

(a) 采集图像：应用相机厂商提供的显示模块（SAPERA 库），从图像板传输到宿主机的图像被显示模块直接显示到屏幕上。在显示模块的程序部分，利用 GPU 完成图像数据的相关计算，并将图像对齐叠加保存。

(b) 相机管理：应用相机厂商提供的命令控制相机的曝光时间，增益，底值，数据传输等。怀柔太阳观测基地是常规观测基地，一般是在当地培训观测人员，为便于操作，减少外语可能带来的理解上的困难，整个界面都设计成中文，这样就降低了培训观测人员的成本。

(c) 数据库管理。观测结果被以 FITS 格式保存在硬盘上，同时被纪录的还有相关的索引信息，这里应用微软公司的 ODBC 数据库接口将索引保存在 ACCESS 数据库里。Microsoft Access 数据库管理系统是 Microsoft Office 套件的重要组成部分^[54]，先后出现了 Access 97、Access 2000 和 Access 2007。其中 Access 2007 是最新版本。Access 适用于小型商务活动，用以存贮和管理商务活动所需要的数据。Access 不仅是一个数据库，而且它具有强大的数据管理功能，它可以方便地利用各种数据源，生成窗体(表单)，查询，报表和应用程序等^{[67][68]}。根据我们的需求，采用 Microsoft Access 数据库系统可以满足常规观测时数据检索管理的需要，在我们编制的程序中，保存并自动维护着一个数据库^[55]，这个数据库管理着活动区号相关信息和 Fits 文件相关信息，天文学家可以查找和使用数据。由于 Access 数据库是和其他 office 套件一同销售的，因此不需要额外安装数据库，使用时在很大程度上与其他 office 套件类似，因此，对于天文学家来讲无需花太多时间来学习，简单的操作就能轻松检索数据库。图 3-13 是数据库的结构图 and 数据库的相关图：

	字段名称	数据类型
	自动编号	自动编号
	文件名	文本
	自编位置索引	文本
	宁静度	数字
	清晰度	数字
	像质	数字
	定标系数	数字
	曝光时间	文本
	叠加帧数	文本
	消去底值	文本
	波长	数字
	观测内容	数字
	注释	文本
	温度	文本
	观测者	文本
	日期	日期/时间
	增益	数字
	天气	文本

图 3-13 数据库中主表（文件索引表的数据结构）

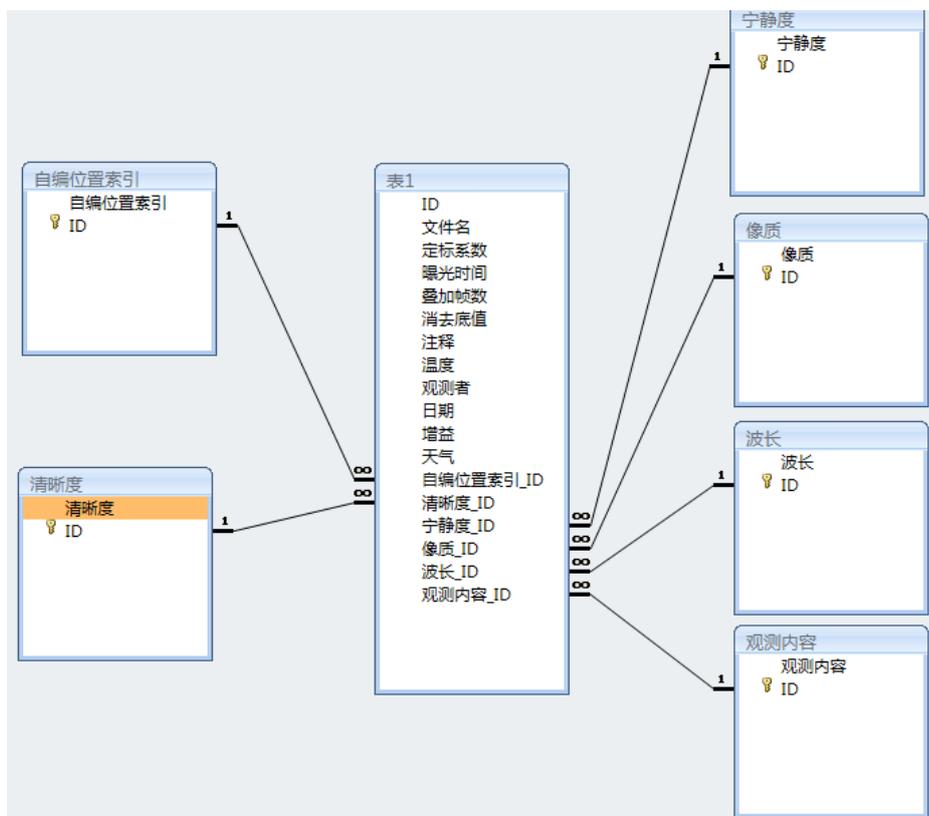


图 3-14 数据库中各表之间的关系

整个应用程序设计实现了良好的用户体验，目前该观测系统已经投入常规观测。

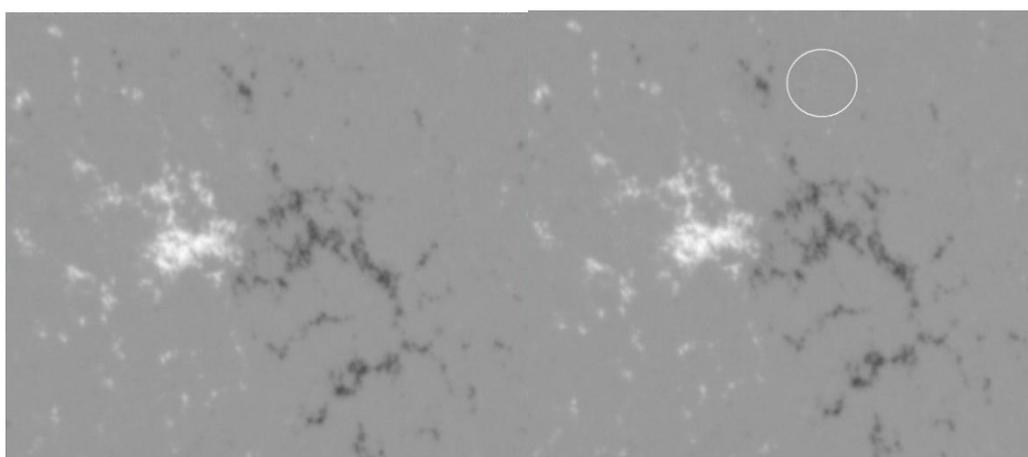
3.3 总结与讨论

表 3-5 列出了系统在更新后同原系统的比较

	以前的系统	新的系统
像素大小	9um	9um
面阵	640×480	992×1000
速度	12.5 帧/秒 (隔行扫描)	48 帧/秒 (最大) (逐行扫描)
处理方式	151 图像处理器	CUDA

计算机	P4	至强四核+TeslaC1060
操作系统	WINDOWS 2000	WINDOWS XP
存储方式	Fits	Fits

2009年4月18日，该系统进行了首次试观测，图 8-11 显示了该系统配合怀柔 35 厘米磁场望远镜观测到的太阳的局部磁图。

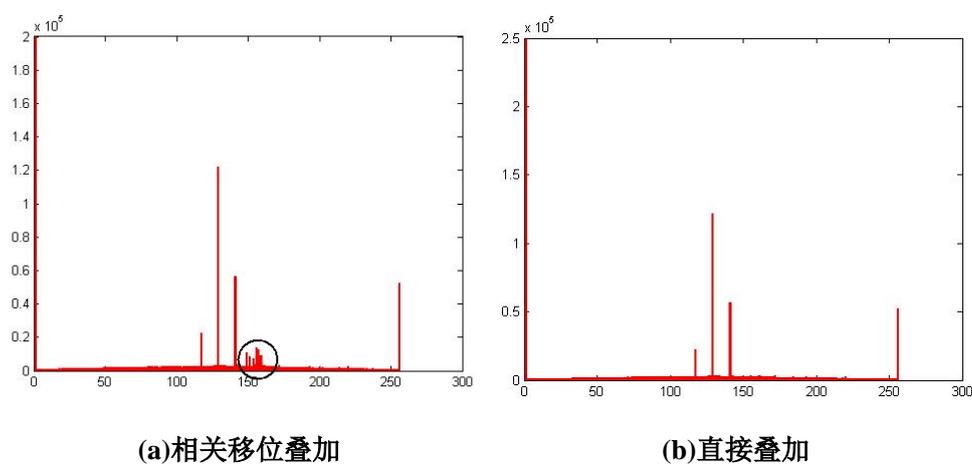


(a)相关移位叠加

(b)直接叠加

图 3-15 磁图对比

从磁图直观上能够看到移位叠加后的数据比直接叠加的图像数据保持了更多的细节，磁场更强，图像反差的对比度更高。



(a)相关移位叠加

(b)直接叠加

图 3-16 磁图数据分布情况对比

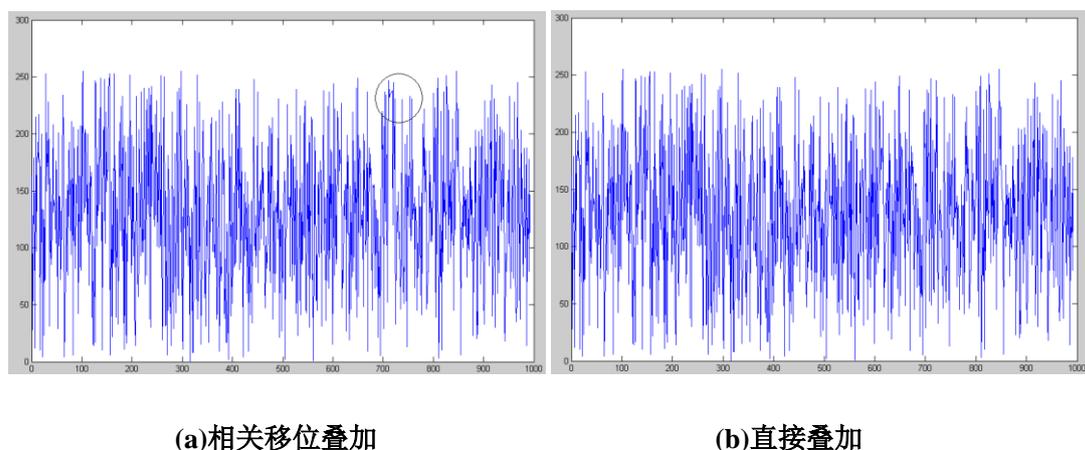


图 3-17 两幅磁图的中线处的灰度轮廓线

从图 3-16, 3-17 的磁图数据分析中能够看到移位叠加后的数据比直接叠加的图像数据在分布上有更多变化, 表示叠加之后的图有更多的细节。

以上介绍了 CUDA 技术在天文上应用的一个实例--基于 CUDA 的 35 厘米磁场望远镜终端系统。现在我们总结一下应用此方案带来的好处。

3.3.1 可扩展可裁剪可移植, 适合大数据量并行计算的应用

基于 CUDA 的实时相关方案具有在应用环境改变后只需改变软件而硬件保持不变的能力, 这样就使同一硬件平台可以适应很多不同的需要。CUDA 系统受到的限制是计算机 PCI×16 插槽以及操作系统, 而这些都不涉及光路, CCD 和其它硬件电路的改动。对于 CUDA 厂商 (NVIDIA) 来说同一系列的 CUDA 驱动可以兼容不同品牌的 GPU 产品, 这样应用者就可以在相同的计算机平台上选择不同的 GPU 计算卡以满足不同规模的需要, 开发者也可以通过这样的裁剪降低造价。这个特性在天文上很有用, 因为天文应用的特点是没有很大需求量, 但需求变化多端。这样我们就可以根据需求创造不同的 CUDA 超级计算平台, 针对不同的需求修改程序中线程的分块和寻址, 可以将不同波长 (可见光, 红外, 射电) 的探测器统一到相同硬件平台下。

3.3.2 高开发效率，获得的知识具有普遍性

当所有的探测器都集中到兼容的 CUDA 计算平台上时，开发者首先省掉的是采购计算机硬件的风险和时间，而且由于采用统一的硬件平台使开发者可以将从一个设计中取得的经验移植到另一个应用上，这一方面提高了开发效率，另一方面开发者从一个项目中获得的知识能够得到广泛的应用。

3.3.3 成本可以接受

应该指出的是基于 CUDA 的计算机平台系统的成本对于大规模应用来讲是很高的，因为 Tesla 卡（或者其他高端支持 CUDA 的消费级显卡）的售价相对于商品化的 CPU 来说是比较高的，计算机平台也要选用图形工作站级别的产品。所以这样的系统我们认为不是很适合大规模的应用。但天文应用通常是单件或者小批量的，而在天文仪器开发中研发的费用会占很大的比例，所以在硬件平台上多投入的资金，会从节省的研发费用里得到补偿。此外，多个 CUDA 计算平台可以作为多个望远镜数据终端，平均来看这也是降低了成本。CUDA 是一项新兴的技术，这项技术会随着技术的成熟和市场普及率的提高不断降低成本，通过构造 35 厘米磁场望远镜的终端系统，我们认为这个方案在成本上是可以接受的。表 3-5 是浪潮公司用于出售的基于 CUDA 的超级计算机，可以看出传统超级计算机的成本远远高于基于 CUDA 的超级计算机。

表 3-6 基于 CPU 和基于 GPU 高性能计算机比较^[56]

	浪潮“倚天”桌面超级计算机	传统 x86 万亿次 Cluster 计算机
性能	万亿次	万亿次
机房	不需要	需要满足一定承重要求的机房
体积	1 台普通 PC Server 大小	至少占用 1 个 42U 机柜位置
散热	不需要专门配备空调散热	需要专业的机房空调设备
功耗	小于 1000w	5000w 左右
噪音	20 至 40db	60db 以上
价格	5 万	50 万

3.3.4 开放系统使维护简单，升级容易，便于后续和改进

NVIDIA 公司为 CUDA 解决方案的库函数提供了源代码，如 CUFFT, CUBLAS 库源代码，不像专有的系统只提供程序的接口，而天文上的需求往往是特殊的，需求往往与商业上的产品不同。开放的源代码使用户可以对其进行修改，以得到更符合要求的系统。

3.3.5 跨平台跨操作系统并且使用简单

当我们把天文仪器接入 CUDA 平台系统，我们就有可能使客户端的程序具有跨平台跨操作系统的能力。因为 CUDA 程序是基于 C 语言的扩展，在不同的操作系统环境下程序只需要少许的改动就可以移植，而且调用方式都是类似的。同时 CUDA 语言支持不同的 GPU 硬件平台，这样甚至不用修改就可以使客户端程序具有跨平台的能力。

3.4 CUDA 技术在天文观测中的应用前景

在 35 厘米磁场望远镜终端系统研制取得成功以后，我们还计划将 CUDA 技术应用到中国南极天文台二期建设项目的预研制中，CCD 大小提高到 $10k \times 10k$ 。基于 CUDA 的 35 厘米磁场望远镜终端系统的试制成功，充分说明该技术对于提高运算速度减少研制时间有着重要意义，同时，其较低的开发成本（硬件无需再扩展），也使其具有很大的天文应用前景，如 LAMOST 望远镜中需要数十个 CCD，可以采用 CUDA 技术，利用一台基于 Tesla 的图形工作站进行并行处理，在一个单位时间完成所有 CCD 数据的处理。另外，由于该系统的编译软件和其他调试工具（如：Profiler）越来越便于使用，可以在开发中提高设计的正确性。总之 CUDA 超级计算系统在天文观测中具有很好的应用前景。

现代的天文望远镜有可能具有很多通道，需要很多探测器同时进行观测。下面我们通过怀柔的多通道望远镜介绍一下基于 CUDA 的解决方案。

最初的怀柔多通道望远镜多达 14 个通道，每个通道都需要一个 CCD 相机，如果按照传统的办法就需要同样多台的 PC 机，这样很不便于操作。望远镜建成时的系统如图 3-18 所示。受当时电子技术的限制，有很多 CCD 的信号纪录还只能是由录像机完成，被 151 图像处理器数字化的图像则被重新采样成 512×512 大大降低了系统的空间分辨率。同时也需要很多计算机来进行数据采

集，导致观测人员需要同时照看很多监视器，繁琐的重复性工作量很大。

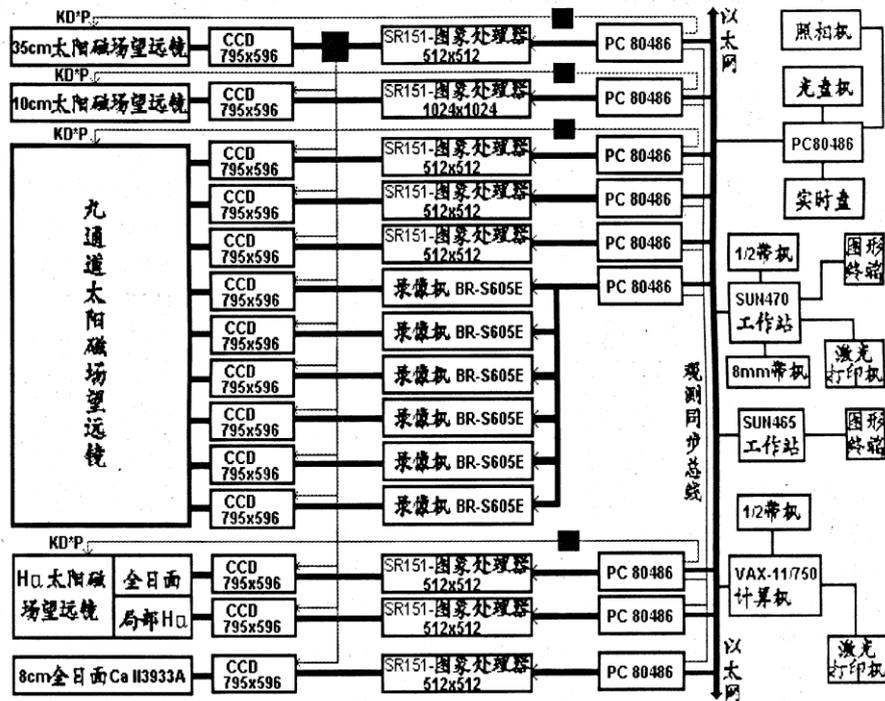


图 3-18 原有的怀柔多通道望远镜接收系统^[57]

经过一代又一代太阳物理研究者的努力，现有的观测系统已经发生了改变。目前投入常规观测的主要有三通道望远镜和 35 厘米磁场望远镜，需要两套观测终端，图像直接叠加，如图 3-19 所示。

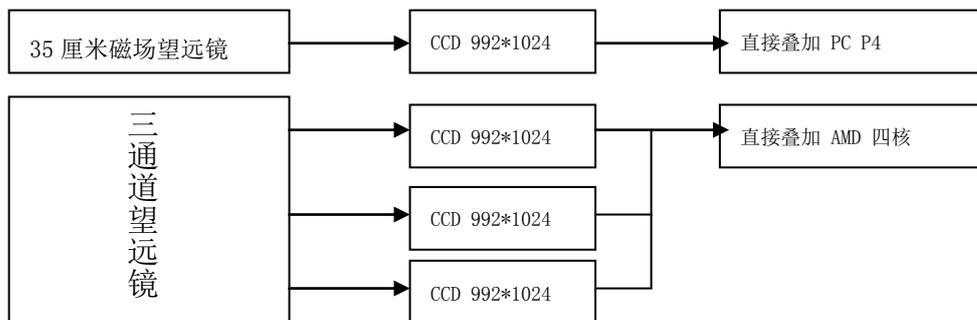


图 3-19 目前怀柔多通道望远镜接收系统

图 3-20 是新方案的系统图。应用了我们新的 CUDA 计算平台可以得到简化，只需一台计算机就可以进行所有的数据采集，实时相关和监控的工作，应用 WINDOWS 强大的图形功能使计算机能实时的显示所有的相机采集的图像（尺寸有所缩小）。

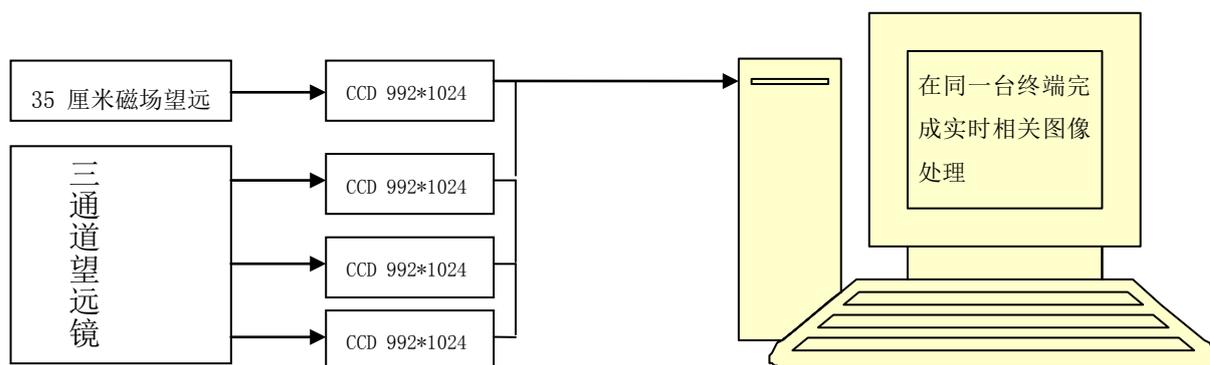


图 3-20 构造中的新的怀柔多通道望远镜接收系统

这里充分体现了使用 CUDA 计算平台的一大好处：扩展方便！只需购买一台图像工作站，并插入多个 Tesla 超级计算卡就可以实现一个控制台管理多个设备的目的。随着制造工艺的不断进步，晶体管集成度将会越来越高。在不远的将来，Tesla 系列计算卡运算性能将会成倍增长。所有这些，都使得我们所研制的系统具有了广阔的天文应用前景。

第二部分 运用“二值化”方法的面阵 CCD 导行在太阳磁场观测中 的应用

第四章 太阳磁像仪自动导行方案的发展趋势

4.1 太阳磁像仪使用导行系统的必要性

当今国际上较大型的太阳望远镜基本上为赤道式，大型地平式太阳望远镜则较少，这当中除了涉及光学，机械的问题外，导行系统也是其中的一个难点。太阳望远镜多采用赤道式望远镜主要是因为天体的视运动可以很容易地利用赤经轴的匀速转动来补偿，在赤道式望远镜的视场中太阳位置没有相对转动，同时赤道式望远镜在观测条件最好的天顶位置没有盲区，而地平式望远镜对太阳进行跟踪时，会产生像场旋转^[58]。

怀柔太阳观测基地的多通道望远镜就是赤道式望远镜。除了第一部分阐述的通过望远镜终端系统对图像进行矫正，提高望远镜的跟踪精度也是减少观测扰动的重要方法。由于太阳观测的高空间分辨率是其他天文观测所不具备的优势之一，因此导行系统成为太阳望远镜中广泛使用和必不可少的附属设备。太阳观测中使用导行系统主要有以下几方面原因：1. 观测环境中风，轻微地震等自然现象带来的误差；2. 望远镜极轴精度所带来的系统误差；3. 齿轮的准周期误差^[19]；4. 望远镜应力弯曲（特别是对于大的望远镜）；5. 日轴方位角变化（ $\text{MAX}=1''/\text{min}$ 在春分秋分点）5. 在早晚时候，地平线上大气折射率随高度的变化快，从而导致太阳视方位的变化^[59]。以上所有这些误差都会带来望远镜跟踪误差，其结果就是采集的图像不能完全对齐，存在晃动，因此，需要自动导行技术实时矫正。如何保证太阳望远镜的长时间稳定跟踪是一个重要课题！

单从跟踪太阳的方法来看，太阳跟踪分为手动跟踪和自动跟踪两种^[60]。手动跟踪精度差，只能间歇运行，而且国家天文台怀柔太阳基地是常规观测站，手动

跟踪增加了观测员的观测负担，大型望远镜手动跟踪绝不可行。自动跟踪是用步进电机作为动力来驱动跟踪系统。自动跟踪方法分为两类：一类是采用程序控制，用电动机驱动跟踪系统，按照一定的程序跟踪太阳，这属于视日运动轨迹跟踪，这称为恒动，其缺点是存在累积误差，并且自身不能消除；另一类是利用光电转换装置，当太阳偏转时产生电压信号传到控制步进电机上，转动跟踪装置使其对准太阳，这称为自动导行。

对于国家天文台怀柔太阳观测基地的磁场测量，由于磁场信号相对其背景信号很弱，现行的磁场观测方法要求观测时用较长的曝光时间，通过多幅图叠加换取磁场灵敏度的提高，在这种情况下，为确保科学目标的实现必须有一套高质量的稳定的跟踪系统，原有系统采取的解决方案是“恒动跟踪+自动导行”得以实现。

对于恒动跟踪方法是根据相应的公式和参数计算出白天太阳的实时位置然后发出指令给步进电机，去驱动太阳跟踪装置，以达到对太阳实时跟踪的目的^[60]。

恒动轨迹跟踪方式的原理图如下：

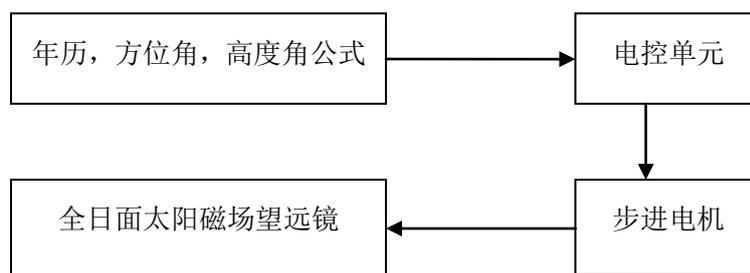


图 4-1 恒动跟踪系统原理

恒动轨迹跟踪方式是望远镜长时间跟踪太阳的有效方法，但是这种方法本身存在积累误差，并且由于望远镜系统本身的误差，恒动跟踪的精度仍然不能满足天文学家角秒级跟踪误差的需求。

怀柔太阳观测基地全日面H α 望远镜原有的系统即是采用恒动跟踪方式，再配合四个光电转换器（即“光电跟踪法”）实时矫正恒动跟踪的误差^[59]。但是在实际的观测中，这种跟踪方式精度低，在太阳像大小发生变化时需要人工调整光电转换器的位置，当探测器没有指向太阳时会产生输出，这就存在望远镜失控的安全隐患。

为进一步提高观测质量，解决望远镜跟踪难题，怀柔太阳基地进行了科研攻关。事实上，可以将跟踪分为两个部分：恒动和自动导行。由于恒动系统本身是通过公式推导得出抵消地球自转的速度，可改造的余地小，造成跟踪扰动的根本原因在于在本章开始时提到的五个因素。因此，要实现角秒级的自动跟踪系统就必须改造现有的自动导行方式。

4.2 太阳磁像仪导行方案的发展历程

随着观测研究需求的提高和科技水平的不断进步，太阳望远镜的自动导行技术经历了一个点...线...面的辩证发展过程。由于光电导行系统属于技术范畴，各研制单位都有一定的保留，所以在调研过程中，无法接触到过多的资料，但是从历史上看，现有的导行技术主要有以下三种：

(A) 光电跟踪导行法——光电跟踪是指利用光敏传感器来测定赤经和赤纬的偏差，当偏差超过一个阈值时通过电机驱动跟踪调整装置转动，减小偏差直到使光敏传感器电压重新平衡实现对太阳的自动导行。

光敏传感器结构如下图。在太阳像的边缘对称设置四个硅光电池，当太阳像偏移时，由于硅光电池接收到的信号不平衡会在赤经赤纬上产生电动势，导行系统根据电动势的大小调整望远镜的观测方位，从而使偏移的太阳像回到正确位置。该套跟踪系统采用硅光电池来进行检测，结构简单，能及时测出观测目标是否偏移，在当时没有较好的探测器件的情况下，应该说是难能可贵的，但这对于实现高精度跟踪的要求是远远不够的，实际使用效果有限，并且整个系统由模拟电路构成的，很难把智能元件加到控制系统里面，导致该方案缺乏必要的保护。

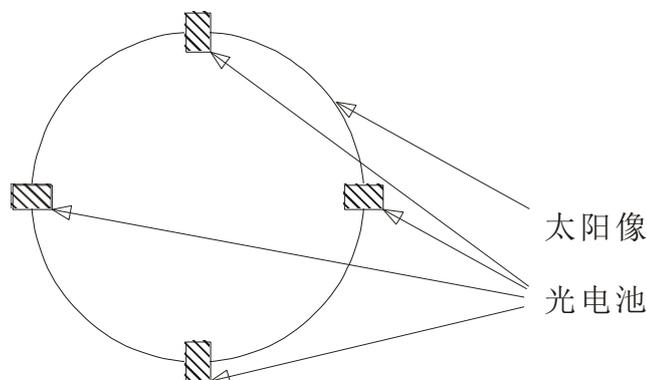


图 4-2 怀柔的多通道望远镜的原有的导行方案

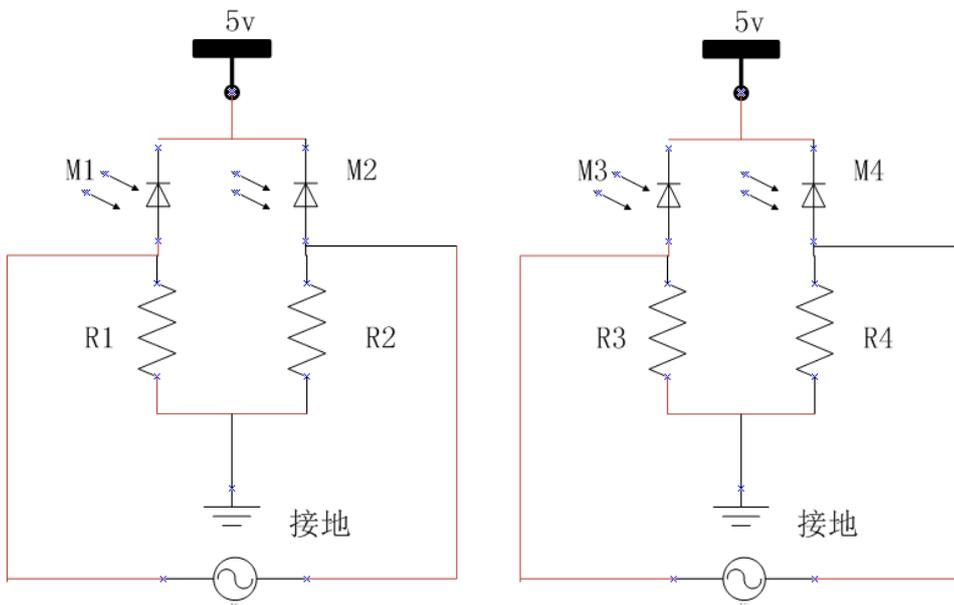


图 4.3 赤经方向，赤纬方向电压比较示意图

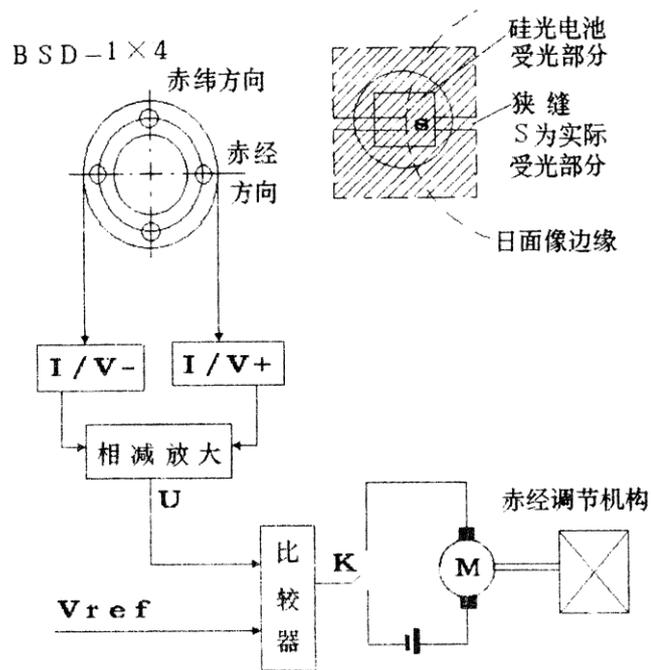


图 4-4 传统日光导行系统误差处理示意框图^[61]

(B)线阵 CCD 导行法——通过抽取线阵探测器上的几个点,几条线作为导行依据,这种导行技术目前应用在西班牙卡那利岛上的德国 GCT^[62]、日本的 YOHKOH 卫星^[63]以及我国曾经研制的球载望远镜上。从查阅的资料中,最新的一种比较详细的线阵 CCD 导行方案是在云南 1 米红外太阳望远镜上采用线阵 CCD 检测太阳的边缘从而得到太阳的重心^[58]。将重心的实际位置与理论位置相减,得到偏差量,将此偏差量消除系统误差,传给望远镜控制系统。这种技术的缺点是 CCD 利用率低,导行精度不高,仍然难以满足高精度太阳观测的需要。

(C)现有的导行系统中还存在一种采用光栅钢带码盘和小电荷耦合器件 (CCD Charge Coupled Device) 配合局部太阳像光路的导行方法,具有代表性的如国家天文台怀柔太阳基地的小磁场望远镜,该方法为现有技术中相对比较先进的一种导行方法,通过计算跟踪期内望远镜实际转过的角度与太阳理论位移的偏差,实时发出速度微调指令,形成位置闭环跟踪;再通过增加的胡氏光路,利用小面阵视频 CCD 实现二次自动导行^[65]。但是该导行系统虽采用了视频的面阵 CCD,但是其靶面小,从太阳上抽取的点数少,且需要附加导行光路,所以系统较复杂,成本开销大,不适合利用在空间类望远镜的导行观测中,以及对现有望远镜系统的更新改造上。

4.3 大面阵 CCD 自动导行方案在工程上面临的主要技术瓶颈

从上一节的分析中我们可以看到,太阳望远镜的自动导行技术所经历的发展历程是与终端系统数据处理能力相适应的,提高导行精度的根本在于提高计算机的数据处理能力。要实现大面阵 CCD 的自动导行就必需解决大数据量的实时计算问题。

在下一章中,将详细介绍基于质心算法的大面阵 CCD 导行方案,通过“二值化”方法优化,可以提升终端系统处理的精度和效率,从而实现高精度的自动导行系统。

第五章 面阵 CCD 导行——基于质心算法的全日面像自动导行方案

5.1 基于质心算法的大面阵 CCD 全日面导行的基本思想

面阵 CCD 导行方案由于要求更高的实时处理能力, 最早使用于空间观测中, 例如 SOHO^[61], 近年来开始在一些在研的设备中使用, 遗憾的是目前还查不到实际使用的文献。多年来怀柔太阳基地对导行系统进行了认真的研究, 在文献【59】中对基于质心算法的导行方案进行了理论验证, 得出了如下结论: 在探测器面阵的 AD 量化位数为 8, 填充因子为 1 的情况下, 该方法的定位精度可以达到 1/70 像元分辨率。根据这一理论结果我们进行了工程应用方面的设计, 在导行系统使用了以下新方法: (1) 用大面阵 CCD 采集全日面图像, 利用工作站对图像进行“二值化”处理, 用质心算法实时计算图像位置; (2) 用嵌入式终端设备处理校正数据, 实时控制电机驱动设备。这种方法简化了光路, 提高了导行系统的精度, 降低了薄云, 雾等对跟踪精度的影响。

当使用面阵 CCD 作为导行时, 通常以太阳像的中心作为参考位置。求中心的算法有多种, 但差别不大。因此本文中仅讨论以位置为加权平均的质心作为太阳像中心的方案。

5.2 基于质心算法的全日面导行方案的原理

导行系统采用面阵 CCD 导行, 根据每幅图像的质心判断太阳的实际位置。求质心的算法有多种, 但差别不大, 在文献【59】中, 作者提出了采用加权平均的质心作为太阳像中心的方案, 其原理如下:

定义太阳像的质心为:

$$X = \sum xf(x, y) / \sum f(x, y); \quad (1)$$

$$Y = \sum yf(x, y) / \sum f(x, y). \quad (2)$$

式中, $f(x, y)$ 表示 (x, y) 点的像素值, X 、 Y 分别表示计算后的质心位置。

假设第一幅图像的中心为 X_0 、 Y_0 ，则偏差计算如下：

$$OffsetX = X - X_0;$$

$$OffsetY = Y - Y_0.$$

根据以上理论，由于 $offsetX$ ， $offsetY$ 表示当前太阳位置与开始跟踪时太阳位置在赤经和赤纬方向上的偏差，所以设计的望远镜轴系控制系统每 $2 \times 10000 \times 10000$ 个脉冲望远镜在赤经或者赤纬方向上转动一圈，由此可以计算：

$$\begin{aligned} & \text{每校正一个角秒发送的脉冲个数} \\ & = (2 \times 10000 \times 10000) / (360 \times 60 \times 60) \approx 154.32 \end{aligned}$$

整个太阳像视场为 $32' \times 32'$ ($1920'' \times 1920''$)，由于自动导行采用的相机面阵为 $2029(H) \times 2044(V)$ *pixel*， 1pixel 约等于 $1.06''$ ，因此偏移 1pixel 对应发送 145.58 个脉冲。

第六章 面阵 CCD 导行——基于质心算法的全日面像导行系统

6.1 CUDA 技术在面阵 CCD 质心计算中的劣势

在第一部分中，我们详细阐述了 GPU 在并行计算中的优势，简单的说，CPU 是串行计算的领先者，而 GPU 是并行计算的领先者。CPU 在操作系统、系统软件、应用程序、通用计算、系统控制等领域有着较深的技术功底；GPU 通用计算虽然出现时间较晚，但是由于采用了与 CPU 完全不同的处理器架构，在处理并行计算方面优势明显，GPU 可以同时完成成千上万的任务。相对于 CPU，GPU 非常擅长于图形类矩阵运算以及非图形类并行数值计算，因此数学模型的并行化对于 GPU 性能的发挥至关重要。

基于质心算法的大面阵 CCD 自动导行需要实时计算 CCD 采集图像的质心。如 5.2 节所示，质心算法是较为典型的串行处理模型，每一次计算都需要将所有像素逐个累加，再求平均，这样的非并行化的数学模型导致 CUDA 计算效率低下，造成这种情况的主要原因是由于 GPU 中存储器和处理器数据交换需要较多的时间开销。如表 6-1 所示，利用计算机随机生成的整形数据（由于将采用的相机为 8 位，所以数据都小于 2^8 ）模拟 GPU 和 CPU 在不同 CCD 面阵下的质心计算速度比较。

表 6-1 GPU 和 CPU 在不同 CCD 面阵下的质心计算速度比较

	CPU (Intel 志强处理器)	GPU (Tesla)
CCD:2000*2000Pixel	7.1ms	8.5ms
CCD:992*1000Pixel	4.1ms	4.3ms
CCD:640*480Pixel	3.1ms	3.5ms

表 6-1 表明 CUDA 技术在质心算法模型中并没有明显优势，会导致计算时间开销增大，成本增加。

基于以上比较，不能利用 GPU 实现大数据量的质心计算，而在计算数据优化的基础上利用 CPU 进行质心计算。

6.2 利用“二值化”方法实现实时大数据量质心计算

为提高运算速度，我们考虑从软件着手优化算法以提高运算能力。从质心算法的执行流程来看，可优化的余地小。通过分析数据的构成形式，我们考虑采用“二值化”方法对采集到的像素值进行处理。

首先我们要证明采用“二值化”方法对计算精度的影响。在文献【59】中阐述了以下结论：

如图 6-1 是两张太阳白光像一张是 8BIT 量化，另一张是“二值化”之后的结果，从图中可以看出“二值化”后，基本看不出什么细节，这正是我们所需要的，因为在高 BIT 量化的情况下有可能对千分之一像元分辨率的图像移动产生响应，这不是我们需要的，因为太阳上的结构在高精度的 AD 变换后会对计算质心产生影响。我们实际上是需要没有细节的太阳像，而低 BIT 的 AD 变换会抹掉低反差的结构，这对我们计算质心是有好处的。

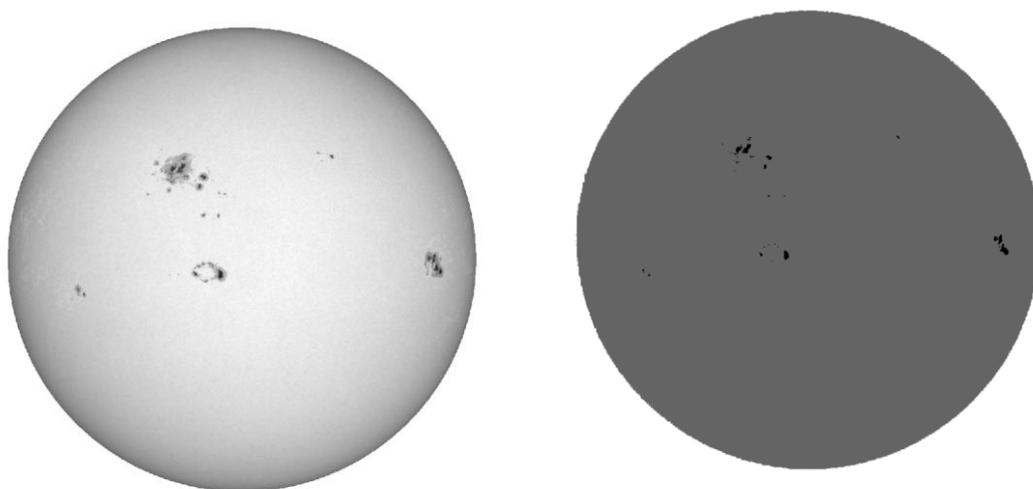


图 6-1 8 位和 1 位的全日面像

上图是一张 MDI 2003 年 10 月 28 日的全日面白光像。左图是 8 位量化，右图是 1 位量化(灰色区域被量化成 1 黑色部分为 0)。可以看出在一位量化时太阳上的活动区所影响面积大大缩小，还可以适当的调整阈值以进一步减少太阳上的活动区对定位精度的影响。

在采用“二值化”方法量化时，薄雾等的影响将不会很明显。虽然比较

厚的云会产生比较大的误差，但是在这种天气下观测已经不会再继续，主要的要求是导行机构不失控，不会对安全造成影响。所以从这一点上来说，产生误差是可以被接受的。

此外，由于该望远镜系统设计建造于上世纪七、八十年代及九十年代初，光学设备和电子设备都采用当时先进的器件，但随着时间的推移，部分设备老化，通过测试发现，光路会出现杂散光的现象。为此，导致 CCD 所采集的图像噪声较大，另外每一帧图像的曝光时间短，图像的位置和形状容易受大气抖动的影响，采用“二值化”方法也可以很好的解决这个问题，降低杂散光的影响。

“二值化”方法是：取太阳像中心（ $2000 \times 2000 \text{ pixel}$ ）面积区域，计算其像素点的期望值 p_0 ，定义太阳像中某一点像素值为 p ：

$$\text{if } p > p_0 \text{ then } p = p_0 ; \text{ if } p < p_0 \text{ then } p = 0$$

由图像知识可知，采用“二值化”方法后，图像过滤掉了太阳表面一些大反差结构（如：黑子，耀斑等），还过滤掉了杂散光等干扰信号，得到稳定清晰的图像，它的中心更加精确的反映了真实太阳像的中心。

高 BIT 量化会增加计算成本，降低跟踪精度，采用低 BIT 量化可以加快大数据量的处理时间。如表 6-2 所示在 $2000 \times 2000 \text{ Pixel}$ 面阵下不同 BIT 量化位数的数据处理时间（英特尔志强 2.66GHz 四核 CPU, 2G 内存）。

表 6-2 不同 BIT 量化下的数据处理时间

量化位数\处理时间	处理时间
1BIT	3.4ms
8BIT	7.1ms
16BIT	10.5ms

从上图可以看出，采用“二值化”方法可以提高数据处理能力数倍，这对于提升系统的响应速度有着重要意义。

6.3 基于质心算法的全日面导行方案的模拟

为保证模拟结果的可信,我们还选取了怀柔太阳基地 2006 年全日面 H α 磁场资料进行了测试。2006 年度全日面 H α 磁场资料采用 640 \times 480 Pixel CCD 进行拍摄,因此我们也采用 640 \times 480 面阵的 CCD 进行模拟。

首先用 C 语言在计算机上产生一个 640 \times 480 的二维数组,用命令 `dir *.fit/b>txt type txt` 生成文件序列索引^[66],然后依次复原显示文件夹中原有的 H α 图像,确保图像的正确显示。然后再整体移动图像,判断计算出来的数据和理论值是否吻合。程序界面如图 6-2 所示。



图 6-2 全日面导行模拟程序设计

上图中点击“导行”按钮就能自动显示图像原有坐标和移动两个像素之后的坐标,“文件序号”显示了文件在被测图像中的排序,“误差”对话框显示了理论值和实际值的误差。

下面需要做的工作是移位,将整幅图像向一个方向移动 2 个像素。在实际观测中,由于望远镜晃动以及大气扰动产生的移位并不是循环移位,因此我们在设计中也就不整体向一个方向移位 2 个像素,以产生模拟的效果。

通过计算 10,50,100 幅图的质心坐标,再求每一幅图与 2 的差的绝对值,再

求标准方差，结果如表 6-3 所示。

表 6-3 自动导行误差分析

帧数	标准差	偏差百分比
10	0.054	2.7%
50	0.032	1.6%
100	0.012	0.6%

从以上结果可以看出，采用质心算法计算大面阵 CCD 质心获得的定位精度在 AD 变换的精度足够的情况下完全可以达到亚角秒级；采用基于质心算法的大面阵 CCD 自动导行其误差结果与 AD 变换的精度有关，提高 AD 变换的精度能提高定位的精度。

6.4 基于质心算法的全日面导行方案的系统设计

6.4.1 全日面导行系统软件设计

基于以上的理论分析和模拟结果，我们开始了基于大面阵 CCD 自动导行技术工程上的设计应用。导行系统的闭环控制结构是实时反馈系统，设计系统的基本流程是：采用大面阵 CCD 探测器首先采集第一幅图像，实时计算太阳像的质心，存储作为起始值；然后对实时采集的每幅图像计算太阳像的质心，并与起始值比较，得出偏差值，根据偏差值向终端设备实时发出速度微调指令，从而实现闭环跟踪控制。

闭环控制系统框图如图 6-3 所示。

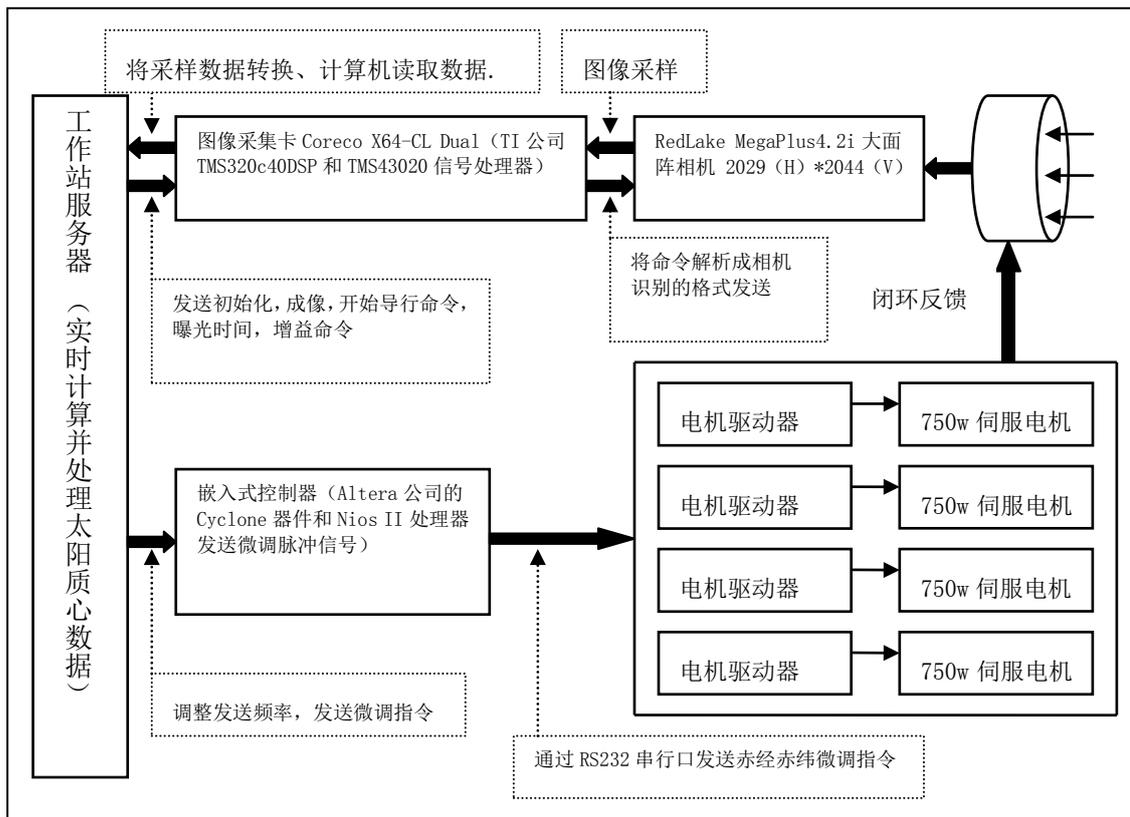


图 6-3 闭环控制系统流程图（RA：赤经 DEC:赤纬）

在该系统中我们采用的 CCD 相机是 RedLake 公司的 MegaPlus 4.2i^[66]，这是一款高性能的 8 位数字输出相机，时钟频率为 10MHz，可以根据曝光时间调整采样速率，面阵大小为 2029(H)×2044(V) pixel，有 400 多万个像元，像元为 9 μm，CCD 芯片可以对整个太阳成像，相机部分基本参数如图 6-4 所示：

Black Level:	Clamped to black reference at the start of each frame.
Gamma:	Unity.
Scanning:	Non-Interlaced progressive scan
Synchronization:	Internal.
Dynamic Range:	Greater than 65dB at the input of the A/D Converter.
Pixel Clock Rate:	10 MHz.
Frame Rate:	2.1 frames per second @ strobe illumination 1.8 frames per second @ 50 millisecond exposure time 1.0 frames per second @ 500 millisecond exposure time

NOTE: Frame Rate = $1/(485 \text{ milliseconds} + \text{exposure time} + 15 \text{ millisecond shutter transition time})$.

4.10.2 Camera Mechanical

Housing:	All aluminum gasket sealed case.
Dimensions:	
<i>F-mount:</i>	4.45"H x 3.90"W x 5.84"L (113.0 x 99.1 x 148.3mm)
<i>C-Mount:</i>	4.45"H x 3.90"W x 4.68"L (113.0 x 99.1 x 118.8mm)
Lens:	C-Mount or F-Mount.
Weight:	Approximately 3 pounds (1.26 Kg).
Mount:	Four ¼ - 20 threaded holes (2 on top/2 on bottom).
Vibration:	3G, sinusoidal from 5 to 150 Hz.
Shock:	20G (nonoperating).

4.10.3 Temperature

Operating:	0 to 35°C (32 to 95°F), non-condensing (Image quality will degrade with increasing temperature)
Storage:	-25 to +80°C (-13 to 176°F), non-condensing.

4.10.4 Humidity

Operating:	<80% @ 35°C (95°F).
Storage:	<40% @ 80°C (176°F).

图 6-4 MegaPlus 4.2i 相机部分参数^[69]

根据该相机的特点，我们根据相机手册编写了成像程序，并将导行控制程序也嵌入到成像程序中。成像情况如图 6-5 所示。

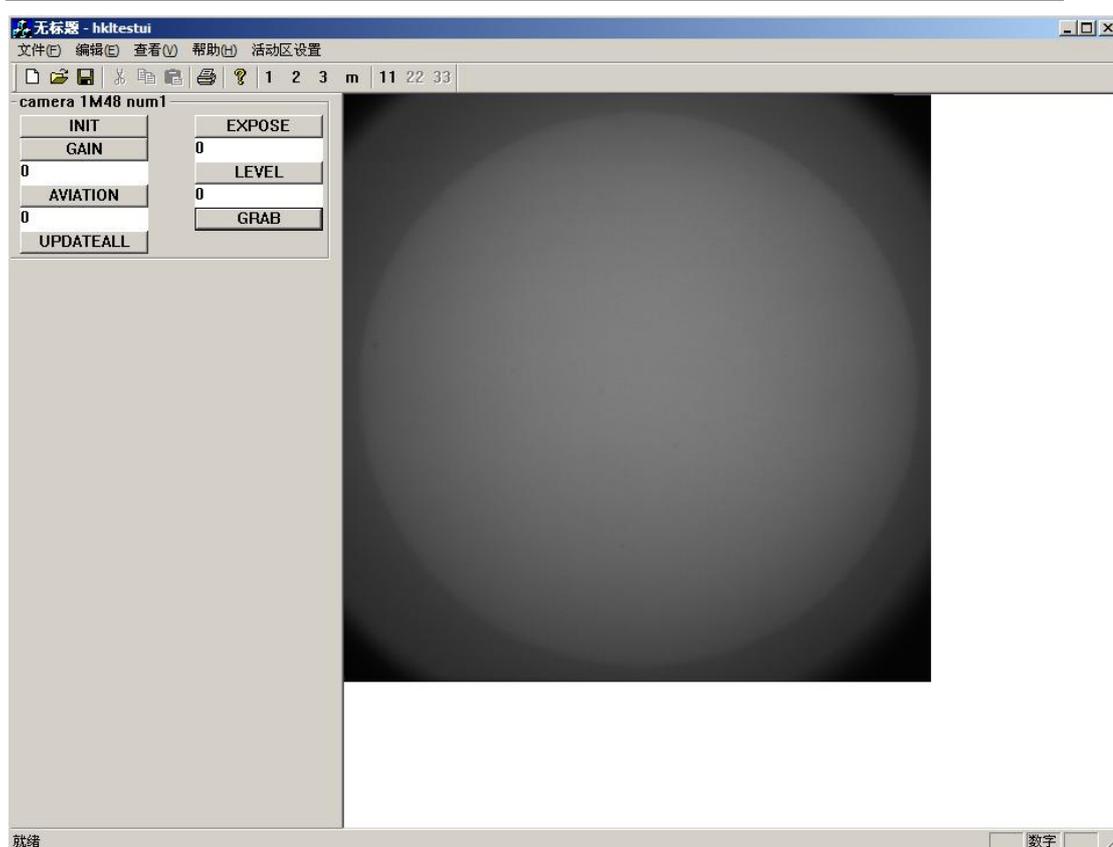


图 6-5 导行系统工作界面

该应用界面可以发送命令操纵 CCD 相机参数，等待位置调整完毕后，点击 Aviation 即可保存第一张基准图像，后续采集数据将与这张图像对比，并发送反馈信号。

6.4.2 全日面导行系统硬件设计

导行系统要和望远镜控制系统进行数据交互，所以它们之间需要进行通信。通常通信的类型可以分为两种：并行通信和串行通信。所谓的并行通信，即八个通道一次的传输量为 8 位(1 个字节)；而串行通信则是一次只传输了一个位(也就是一个标准电位的状态)。两者之间的数据传输速率相差了 7 倍。并行通信虽然速度快，但是稳定性不如串行通信。这是因为数据电压传输的过程中，容易因线路的因素而使标准电压发生变化(最常见的就是电压衰减问题)，因而使数据传输的数据发生错误，如果传输线较长，电压衰减效应及相干扰问题会更加明

显，数据的错误也就会容易发生^[58]。相比之下，串行通信一次只传输一个位，处理的数据电压只有一个，因而比较不容易把数据漏失。在自动导行系统中，所传输的数据量并不大，并且传输的距离较远，因此我们选择串行通信。

串行通信经过多年使用和发展，其接口标准已经有多种，典型的如 RS-232，RS-485，USB 接口，IEEE-1394 等。IEEE-1394 和 USB 接口技术传输速度最快，但它们都有共同的缺点：电缆传输长度较短，IEEE-1394 的电缆传输长度仅为 4.5 米，USB 也仅为 5 米，而 RS232 和 RS485 的传输距离最大可为 4000 米。此外，IEEE-1394 和 USB 接口设计相对于 RS232 和 RS485 更加复杂。通过反复考虑，我们选用的是 RS232 作为数据传输的接口。RS232 接口理论最高数据传输速率为 115200bps，理论传输距离可达 4000 米^[70]，并且结构简单，在数据终端的计算机背面板上即有 RS232 数据传输接口，无需重新设计接口电路，方便易用，满足自动导行的需要。

RedLake 公司的 MegaPlus 4.2i 相机采用 RS-232 接口发送相机参数控制命令，为实现与计算机采集卡的兼容，我们选用 Imperx 公司的转换器 Adapt-A-Link FAMILY 将 RS232 接口转换成 Cameralink 接口，下图是整个采集系统的示意图。

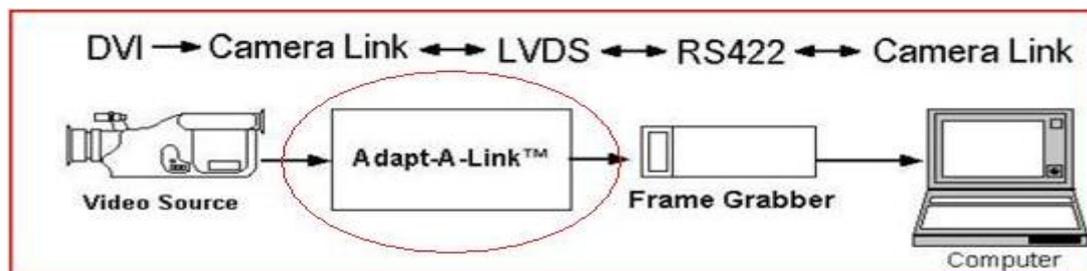


图 6-6 加入转换器之后的采集系统示意图

RS232-Cameralink 转换器的主要技术指标如下。

Input	24 RS422 data inputs 4 RS422 enable inputs 4 RS422 control outputs 1 RS422 data clock 1 RS232 serial interface
Output	Base Camera Link
Clock	33 MHz max
Power	6.0 VDC 300 mA
Dimensions	5.0" x 3.0" x 1.5"

图 6-7 RS232-Cameralink 转换器主要技术指标

此外，为满足大面阵导行所带来的大数据量要求，我们选择了 Altera 公司

的 Cyclone 器件(1PEC100 芯片)和 Nios II 处理器的硬、软件的组合方案。硬件配置和软件设计使用 Quartus II, SOPC Builder 和 Nios II IDE 完成。(关于器件特点、软件使用方法这里就不作详细介绍,可以参看 Altera 公司提供的手册),采用 SOPC 技术来完成控制系统逻辑设计,系统设计的复杂程度大大降低,设计时间大大缩短。基于从速度调节和系统升级等方面的考虑我们选择伺服电机作为驱动设备。我们选用的电机驱动器是松下公司 MINAS/MSMA 的伺服电机,输出功率为 750W。

图 6-8 是设计的电机控制电路板。

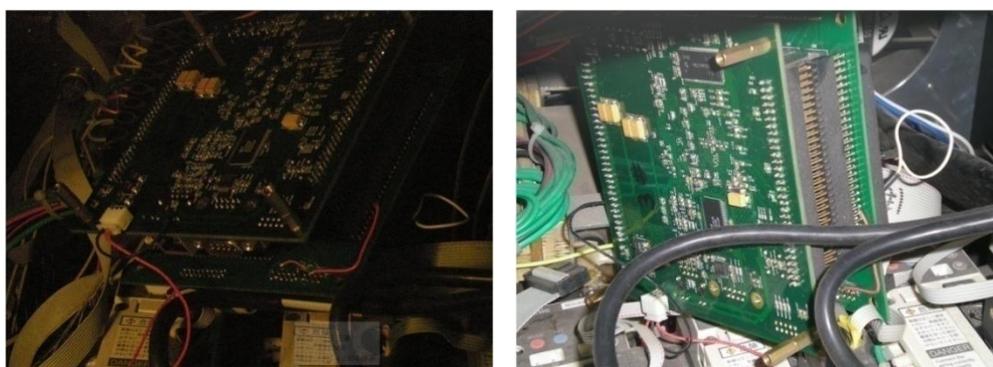


图 6-8 控制电路板(分上下两块,上层为主控电路板,下层为辅助接口电路板)

6.5 基于质心算法的全日面导行方案测试结果

最终导行系统的工作站和伺服电机控制系统如图 6-9、图 6-10 所示。



图 6-9 工作站计算机系统

图 6-10 伺服电机驱动系统

该导行系统设计之初要求望远镜导行精度能达到 $1''/30\text{min}$ (标准偏差),系统安装调试完毕后,2007年11月27日进行了跟踪导行测试,如图 6-11 所

示(纵坐标表示太阳像偏离初始位置的像素值,横坐标表示测试时间),在曝光时间为40ms,每6幅图校正一次的情况下,测试跟踪60min结果为:DEC方向标准偏差为 $0.2134''$,RA方向标准偏差为 $0.9395''$,高于预期目标。另外对长周期(即采样时间为4h)和恶劣天气(有薄云,微风)也进行了导行测试,测试结果RA方向跟踪精度 $1.1384''$,DEC方向为 $0.2516''$ 为这说明该导行方法非常有效。

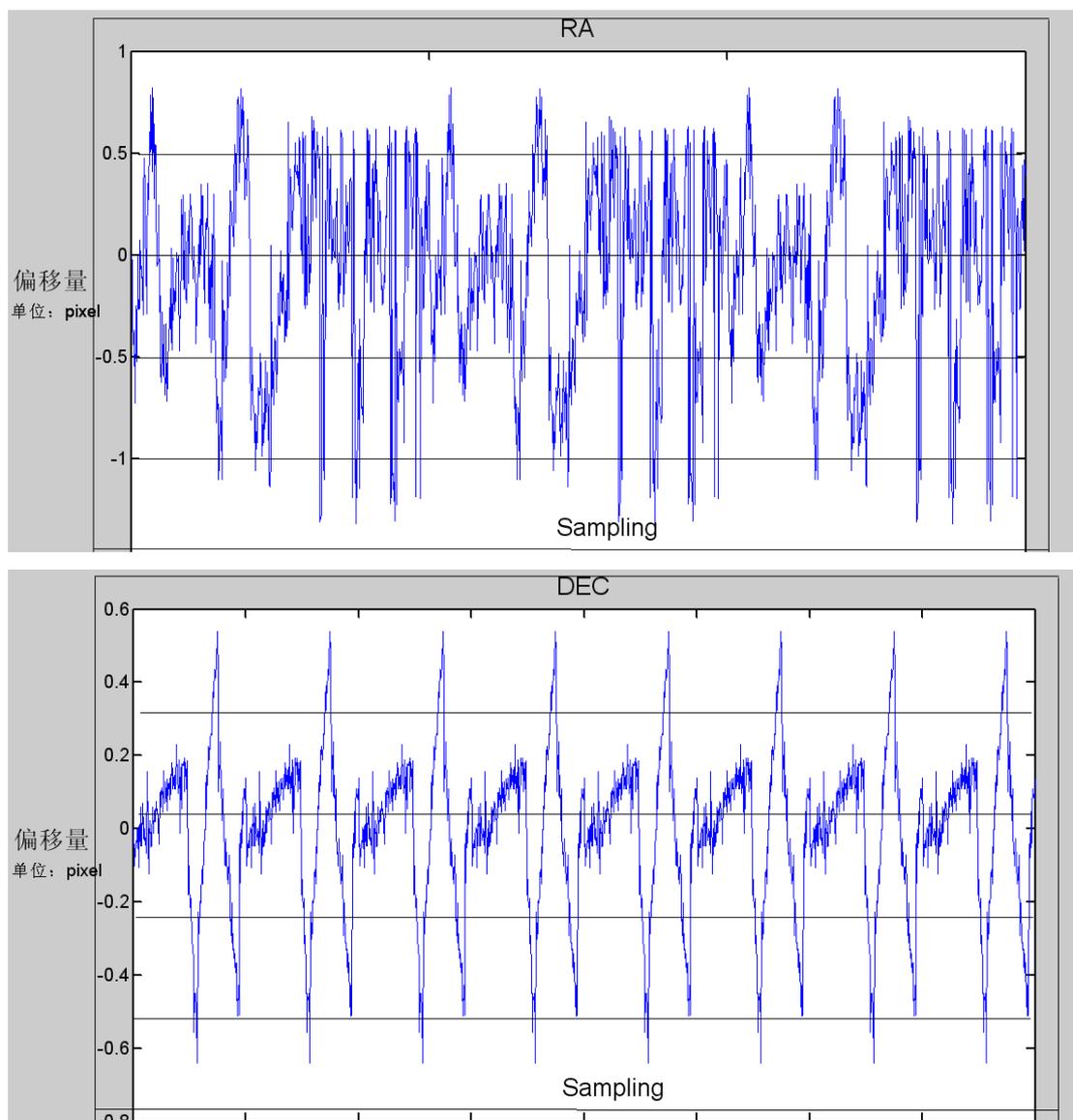


图 6-11 导行 60 min 测试结果 (RA:赤经方向 DEC: 赤纬方向),

纵坐标表示太阳像偏离初始位置

6.6 结论与讨论

通过全日面导行系统的工程实践我们可以得到如下结论：

(1) 基于质心算法的大面阵 CCD 自动导行，当采用填充因子为 100% 探测器时，可以实现导行精度在 1 个角秒误差范围内。

(2) 应用测量太阳质心的算法，太阳上的一些结构会对最终结果产生影响，这个是该方法比较大的缺点^[59]。但是由于太阳上主要的背景结构的反差都较小，采用“二值化”方法，可以进一步减小太阳上的活动区对于定位精度的影响，采用“二值化”方法还可以提高 CCD 像素处理速度，这对于实现高速高精度的自动导行起着重要作用。

第三部分 USB2.0 接口在望远镜高速控制系统中的应用研究

第七章 CYPRESS EZ-USB2.0 在太阳望远镜控制系统中的应用

7.1 原有技术的瓶颈和采用通用串行总线的意义

前面两部分主要分析和探讨了望远镜的高速实时相关系统和高精度自动导行系统，这两项技术的实施涉及到多方面的问题，其中最重要的一点就是要求匹配的高速数据传输接口。事实上，除了数据高速传输问题，望远镜还存在着数据的自动寻址问题，在望远镜机械结构允许的情况下，提高数据的传输速度和实现数据自动寻址可以进一步提高望远镜的响应效率和精简望远镜的电路连接线，这对于提高太阳望远镜的高时空分辨率观测水平具有重要意义。

为进一步提高怀柔太阳望远镜观测质量的潜力，也为了对正在进行的红外太阳望远镜控制系统做技术准备，我们对数据的高速传输和寻址问题进行了预研究。由于高速实时相关系统和高精度自动导行系统虽然需要高速传输但目前暂不需要考虑数据的自动寻址问题，因此从长远考虑，我们将新的控制接口设计实验于多通道望远镜多波带调制控制系统上。多波带调制系统原有的 RS-485 是串行接口，其理论最大通讯速度仅为 10Mbps（理论极限值），半双工的，可扩展能力有限，不支持热插拔。为与之兼容，我们广泛的比较了目前常用的串行接口设计。从电气标准及协议来看，目前较多使用的串行接口主要有：RS-232、RS-422、RS-485、USB 等。

RS-232、RS-422 与 RS-485 传输速度都不及 USB 2.0，并且可传输的距离短，不支持热插拔，不能区分多个设备，这几项接口技术发展缓慢。而 USB 是近几年发展起来的新型接口标准，主要应用于高速数据传输领域，发展迅速。与其它设计方案相比(如 RS-485 等)它可以在不断电的情况下直接将外设连接到接口上，且马上被系统识别。USB 2.0 接口独立供电，其理

论最大传输速率为 480Mbps。一个 USB 控制器可以级联多达 127 个外设，每个外设间的距离可达 5 米，利用集线器可达 30 米^[73]。通过修改 USB 设备驱动程序和固件程序（Firmware），为每个设备“编号”，就可以利用最多可配置 127 个设备的优点，不论插入主机的顺序如何，固定识别 USB 设备，固定对某个设备进行操作。这也就为计算机控制多通道大型望远镜提供了一个很好的解决方案。

7.2 USB 2.0 协议通信的基本原理

如图 6-1 是 EZ-USB 2.0 芯片与计算机主机之间交互的流程。从图中可以看出 USB 设备与主机之间通信的层次分为三层，从系统级到物理层，每一层都完成特定的数据传输接受任务，USB 本身具有数据指挥中心“CPU”，数据的交换是双向的。

但是 USB 设备与主机之间通信的关系不能完全看作是两台微机之间的通信。几乎所有的相关协议和设备开发提供的 USB 控制器资料上都在反复强调一个问题：“Host”只能是主机。也就是说在处理主机与设备之间的通信时，所有的请求，数据与控制信号都能够而且只能被主机响应后才能实现其功能。例如：在数据传输类型之一的中断传输中，接口控制器可以向主机提供中断传输数据请求，但是只有在主机轮询到该设备所在的端口时，该请求才能被响应^[74]。

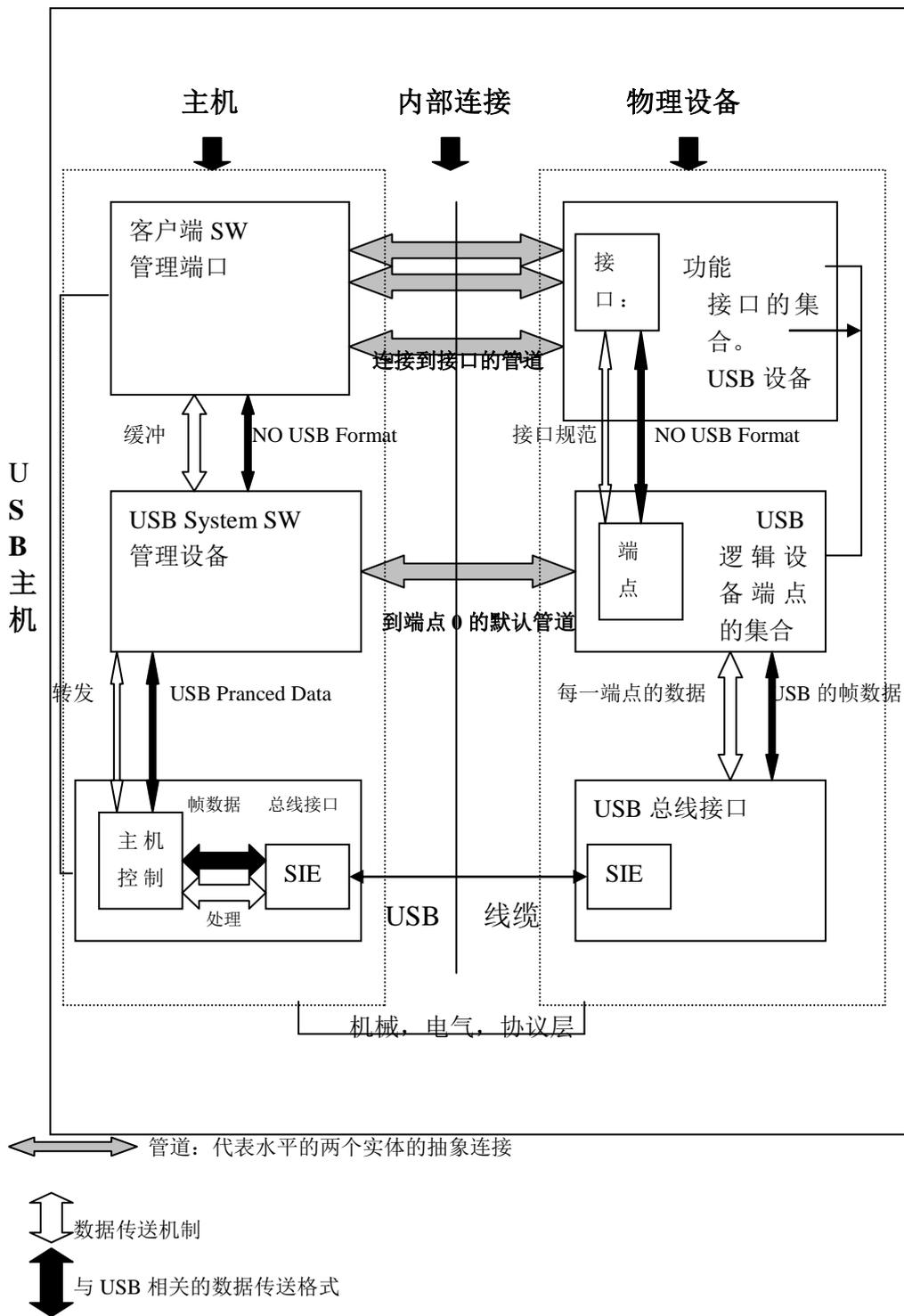


图 7-1 USB 设备与主机的结构图^[74]

将上图简化之后如图 7-2 所示，显示了一个多层次结构的通信模型，它突出了端点和管道所扮演的角色。

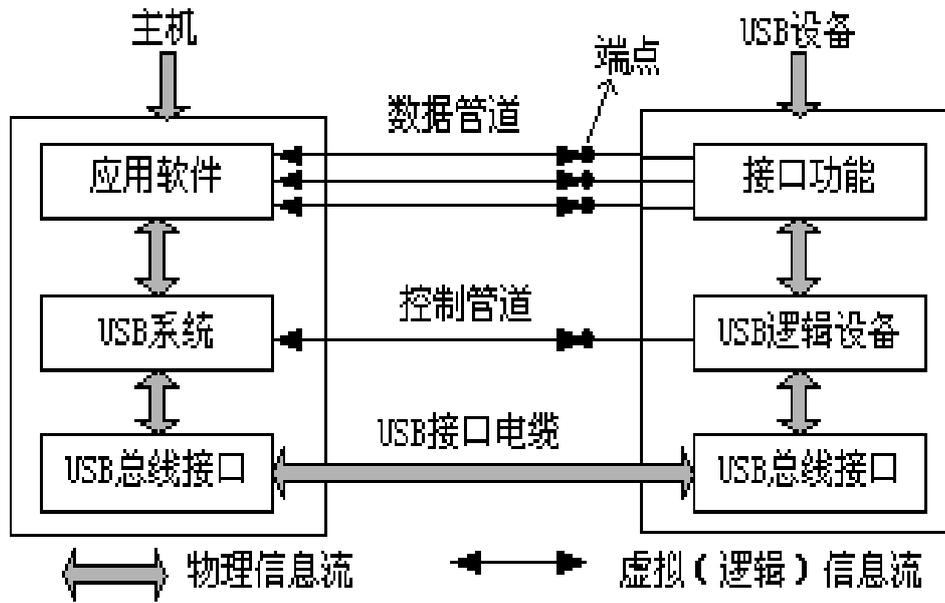


图 7-2 USB 互连通信模型^[73]

USB 接口还有一个特点，普通的计算机端口之间地址是相互独立的，连接到该端口上的设备单独的享用自已的数据通道，设备数据的发送与接收在设备间是独立的。在 USB 接口中，为了最大限度的节省主机资源，当有两个 USB 接口时，他们会汇总到一个叫做根集线器（Root Hub）的地方，并由一个主机控制器所控制并共享一个数据通道。通常把每个 USB 接口称为一个端口。也许在每一个端口上都可以接入 USB 设备，但在实际工作中，在某一时刻，只有一个设备与主机进行通信。每个设备只能等候主机轮到与自己通信时才能占用数据通道。连接到主机上的设备越多,每个设备的可用的通信时间也就越少^[75]。

7.3 CYPRESS 公司 EZ-USB 2.0 芯片的主要特点

USB 是真正的即插即用的接口，可以自动的监测与设置，当启动计算机后需要再连入一个外围设备时，不再需要关闭计算机打开机箱插入板卡，只需要将配有 USB 接口的外设插入相应的计算机机箱上的接口，剩下的事情便完全由 USB 外设和主机完成。USB 设备的通用性也是其特点之一，现在市场上可以方便地得到 USB 接口的键盘、鼠标、光驱、硬盘、摄像头等一系列产品。由于不同的外设可以使用同一个 USB 接口，因此，操作系统不需为每种设备都配置不同的中断和 I/O 口，从而最大限度的节省了计算机系统的资源。USB 只使用 4

根线缆便可完成繁重的数据传输。它们分别是电源线+5 伏、地线、两条差分的数据线 (D+, D-), 这样使得 USB 接头相当小巧。

经过反复调研, 美国 CYPRESS 公司 2002 年推出的一款 EZ-USB FX2TM (又称为 CYPRESS 68013) 芯片满足项目要求。EZ-USB FX2TM 是世界上第一款集成 USB 2.0 的微处理器, 它集成了 USB 2.0 收发器、SIE(串行接口引擎)、增强的 8051 微控制器和可编程的外围接口。在 FX2TM 中, 智能 SIE 可以硬件处理许多 USB 1.1 和 USB 2.0 协议, 从而减少了开发时间和确保了 USB 的兼容性。CYPRESS 公司特别设计了 CY7C 68013 的开发板, 并且带有一个开发包, 内含开发一个 USB 外设所必需的驱动程序, 应用程序以及一个完整的固件程序的框架^[76], 这大大节省了开发时间。

7.4 EZ-USB2.0 芯片主机通信过程

参考 CYPRESS 公司提供的的设计手册, EZ-USB 2.0 芯片(即 CY7C68013) 结构图如下图所示, 它有三种封装形式: 56SSOP, 100TQFP 和 128TQFP。

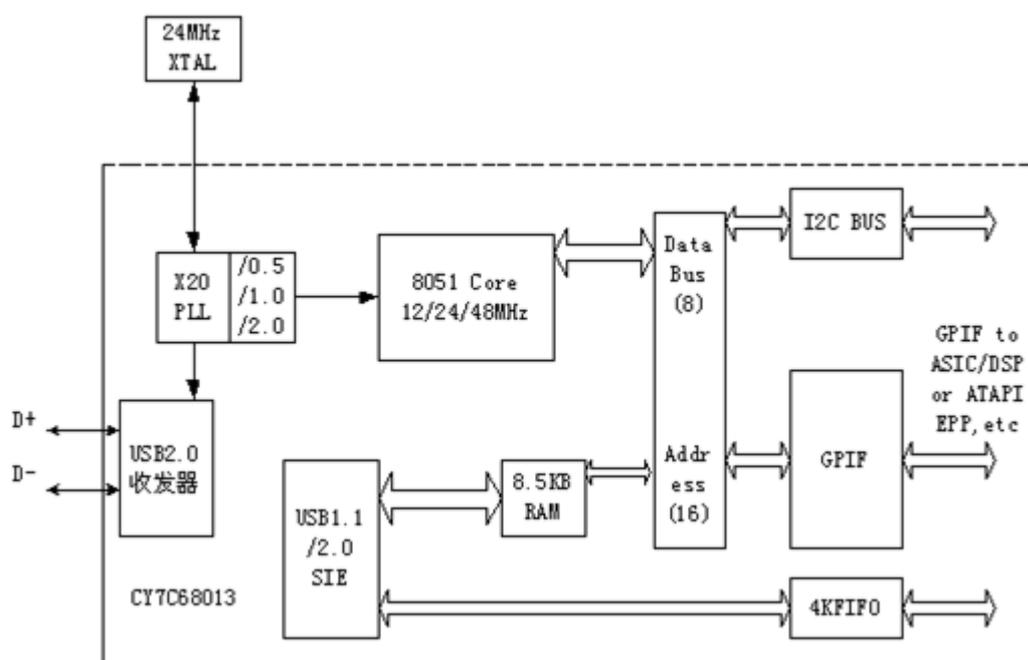


图 7-3 CY7C68013 结构图^[76]

从图 7-3 中可以看到 CY7C68013 集成了以下几个基本特性:

- USB2.0 收发器、SIE (串行接口引擎) 和增强 8051 微处理器;

●软件运行：8051 程序从内部 RAM 开始运行，可以借助下列几种方式进行程序装载：

- (1) 通过 USB 下载；
- (2) 从 EEPROM 中装载；
- (3) 通过外部存储器设备。

●四个可编程 BULK/INTERRUPT/ISOCHRONOUS 端点；
可选双、三和四缓冲

●8 位或 16 位外部数据接口

●通用可编程接口（GPIF）

- (1) 可以直接连接到并口，8 位和 16 位；
- (2) 可编程波形描述符和配置寄存器；
- (3) 支持多个 Ready 输入和 Control 输出。

●集成标准 8051 内核，且具有下列增强特性：

- (1) 可以达到 48MHz 时钟；
- (2) 每条指令占四个时钟周期；
- (3) 三个定时/计数器；
- (4) 扩展的中断系统；
- (5) 两个数据指针。

●3.3V 电源系统；

●智能串行引擎（SIE）；

●矢量 USB 中断；

●独立的数据缓冲区供 SETUP 和 DATA 包控制传输；

●集成 I2C 控制器，运行速度可达 100 或 400KHz；

●8051 可运行在 48MHz,24MHz 或 12MHz

●四个FIFO

(1)内部自带FIIFO降低系统成本

(2)自动转换16位总线

(3)主/从模式运行

(4)FIFO可以使用外部时钟或异步滤波信号

(5)简单的接口到ASIC和DSP IC

●专门的 FIFO 和 GPIF 自动矢量中断；

- 多达 40 个通用目的 I/O 口
- 四种封装-128 脚 TQFP,100 脚 TQFP,56 脚 QFN 和 56 脚 SSOP

一般来说，在设备列举过程中，应该完成以下基本流程：

- 给请求的设备分配一个地址
- 从设备中读取所需要的描述符
- 加载相应的设备驱动程序
- 根据设备所反映的数据来决定设备的最终配置

从望远镜多波带控制系统对于USB接口的需要出发，我们对于USB芯片启动模式进行的认真的设计，从公司提供的700多页的手册中我们可以看到USB芯片有三种启动方案，分别是(1)无EEPROM或是无效的EEPROM(2)“C0”EEPROM(只是装载VID/PID/DID)(3)“C2”EEPROM(将整个固件装入片内的RAM)

上电时，内部逻辑会检查连接到I2C总线上的EEPROM中的第一个字节（0xC0或0xC2）。如果是0xC0，就会使用EEPROM中的VID/PID/DID来替代内部存储值；如果是0xC2，内部逻辑就会把EEPROM中的内容装入到内部RAM中；如果没有检查到EEPROM，FX2就会使用内部存储的描述符来枚举。FX2缺省的VID/PID/DID是0x04B4/ 0x8613/ 0xxxxxyy。

表 7-1 默认 FX2 的 ID 值

Default VID/PID/DID		
Vendor ID	0x04B4	Cypress Semiconductor
Prod ID	0x8613	EZ-USB FX2
Device release	0xXXYY	Depends on revision (0x04 for Rev E)

注： I^2C 总线 SCL 和 SDA 管脚必须上拉,即使在没有 EEPROM 连接的时候。否则检测模式不能正常工作。

怀柔多通道望远镜多波带控制系统有 64 个电机需要通过主机进行控制，因此需要在 USB 芯片中写入“地址”（ID 值）以区分控制系统使用的哪个电机。为达到这个目的必须将存储器中的内容（地址）完全的装入 CYPRESS 内部的 RAM 中，因此我们选用的是第三种启动模式。

下面以一个完整的例子来说明一个完整的设备列举过程，这里假定的环境

为 FX2 控制芯片, 固件由 EEPROM 下载到 FX2 的内置 RAM 中, 具体步骤为:

- i. USB 设备从总线上得到电力供给
- ii. 复位电路将 USB 芯片暂时保持在重启状态
- iii. FX2 检测到有 EEPROM 连入到 IIC 总线上。FX2 从 EEPROM 中读取 VID 和 PID 值, 并用其代替自身内部的 VID 和 PID 的值。
- iv. 主机检测到设备已经连入, 并向设备请求其 ID 值, 设备用代表用户所连入设备的 ID 值来下载固件设备的 ID 值进行回应。
- v. 主机接收到这些 ID 值, 并开始装载其驱动程序。
- vi. 在驱动程序的导引下, FX2 开始执行从外部将固件下载到内部 RAM 的步骤。
- vii. 驱动程序使 FX2 中的 8051CPU 芯片重新启动, 用户的固件程序开始运作。
- viii. 固件使 USB 在电气上与总线断开连接。
- ix. 主机发现了初始设备的脱离, 于是便将驱动程序从其内存中清除。
- x. 固件使 USB 设备重新连入总线。
- xi. 主机检测到了设备的连入, 开始询问其 ID 值。
- xii. 固件以代表用户已经配置好的设备的 ID 值进行回应。
- xiii. 主机收到 ID, 开始装载典型驱动或是用户自己所做的驱动。
- xiv. 到这一步, 主机和 FX2 固件便连接完毕, 开始执行具体的应用程序。

7.5 基于 CYPRESS EZ-USB 2.0 芯片的接口设计

7.5.1 基于 CYPRESS EZ-USB 2.0 芯片接口的软件设计

为了解决区分多个 USB 设备的问题, 必须考虑为每一个设备“编号”, 考虑过几个方案, 第一个方案是修改 USB 芯片中的固件程序, 增加自定义请求部分。通过应用程序向每一个 USB 设备以广播的方式发送请求, 如果某个 USB 设备发现这个请求的前几位地址位, 与自己的编号相同, 则响应该请求, 其它设备处于等待状态; 第二个方案就是采用轮询方式, 应用程序轮询每一个 USB 接口, 当轮询到某个 USB 设备时, 如果发现该设备的编号正是自己的目的地址时, 则要求该 USB 设备响应该请求。第三种方案就是修改驱动程序和固件程序, 将功能驱动程序与总线驱动程序“绑定”起来, 这样当 USB 设备插入主机, 装

入驱动程序后，主机立即就能发现这是“第几号”USB 设备。

通过深入研究，发现前两种方案不能在物理层上将 USB 设备区分开来，也就是需要人先植入“编号”，然后通过应用程序的与固件程序通讯的结果由人来判断这是第几号设备，计算机应用程序是不会显示这是第几号设备的，这仅仅是人能区分多个设备，但是计算机却是不能识别的，因为如果在物理层上没有区分，那么主机就会认为这些 USB 设备是无区别的，并给每个 USB 设备随机编号。因此最终选定了第三个方案。

以下就是第三个方案的设计思路：

1)首先 EZ_USB68013 芯片以 C2 方式启动，在芯片的 IIC 总线上悬挂 EEPROM 24CL64 存储器。在 24LC64 芯片里写入固件程序，其中包括有关 80C51 的操作。

2)在芯片的后面几行（0x60）写入一个标识符，用以标记 USB 设备的编号。
具体的来讲就是增加代码

```
BYTE xdata Digit[]_at_0x60 = {0xC2,0x47,0x07,0x09};  
BYTE code Digit[]_at_0x60 = {0xC2,0x47,0x07,0x09};
```

3)标识符响应开发包的驱动程序，利用 68013 芯片 A0(上传/下载 RAM)永远有效，叫主机通过驱动程序发送一个 IRP 请求包（先要构造一个 URB）发送 A0 请求，而 68013 方面则把请求包要求的 RAM 中的字节发送给主机来响应 A0 请求。

以下是设计框图。

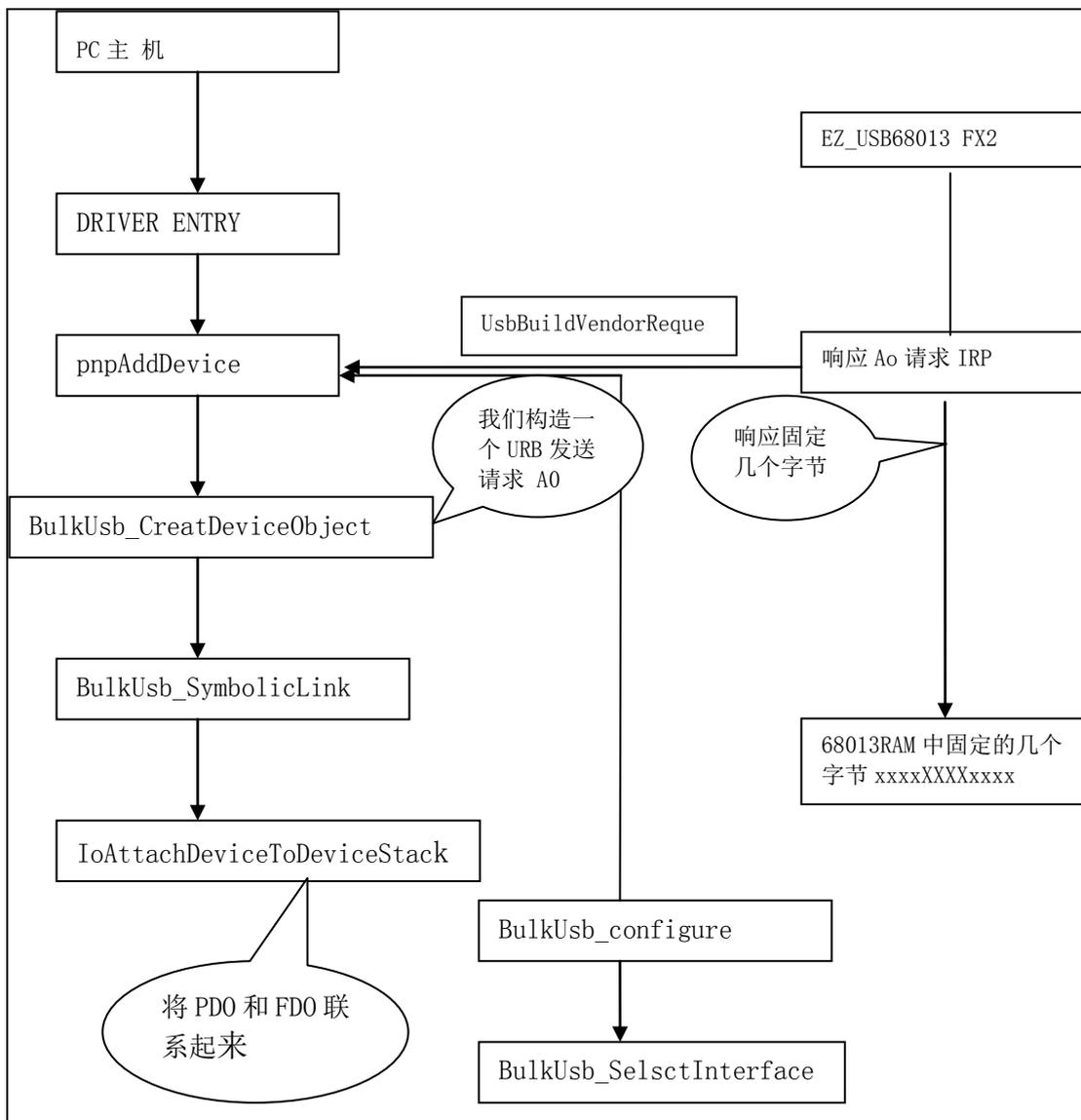


图 7-4 函数调用关系

在这项设计中我们是利用 Windows 2000 DDK 对 EZ-USB FX2TM 芯片进行驱动程序开发^[78]。由于多通道望远镜多波带控制系统的特殊需要，驱动程序的开发是完成控制系统接口设计的一个难点。

Windows 2000 DDK 定义了驱动程序的分层，以适用于即插即用系统。各个不同层次的驱动程序一起构成了一个设备驱动程序栈。

如下图 7-5 所示，在 WDM 驱动程序模型中，每个硬件设备至少有两个驱动程序，其中一个驱动程序为功能（function）驱动程序，即通常硬件设备的

驱动程序。它了解使硬件工作的所有细节，负责初始化 I/O 操作，有责任处理 I/O 操作完成时所带来的中断事件，有责任为用户提供一种设备适合的控制方式。另一个驱动程序为总线 (bus) 驱动程序，它负责管理硬件与计算机的连接。例如：PCI 总线驱动程序检测插入到 PCI 槽上的设备并确定设备的资源使用情况，它还能控制设备所在的 PCI 槽的电流开关。

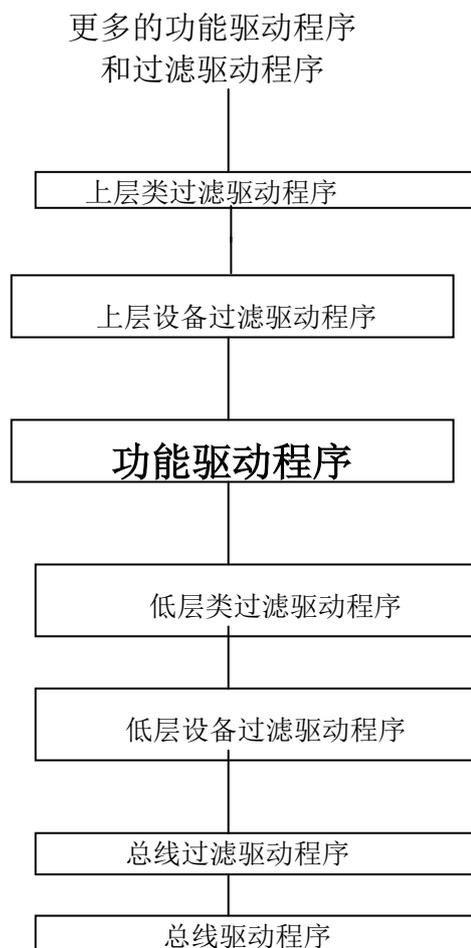


图 7-5 设备驱动程序栈

在功能驱动程序之上以及功能驱动程序和总线驱动程序之间可能存在着各种类型的过滤驱动程序 (Filter Driver)。过滤驱动程序也可以创建过滤设备对象 FiDO(Filter Device Object)。位于 FDO 上面的过滤器设备对象称为上层过滤器，位于 FDO 下面（但仍在 PDO 之上）的过滤器设备对象称为下层过滤器。过滤驱动程序是一个中间层的驱动程序，它可以截获并处理经过它的 I/O 请求。过滤驱动程序用于变更标准设备驱动程的行为，但它不是重写整个总线驱动程序或者类驱动程序，只是修改一些感兴趣的动作而已。图 7-5 中，有多种过滤驱

动程序，过滤驱动程序总是相对于一个特定的驱动程序层而存在的，并帮助其完成一些辅助功能。总线过滤驱动程序作用于连接到特定总线驱动程序的所有设备，类过滤驱动程序在指定类的每个设备栈中安装，而设备过滤驱动程序针对一个特定的设备。低层的过滤驱动程序在功能驱动程序下，而上层的过滤驱动程序在设备栈中功能驱动程序之上。过滤驱动程序只是在设备第一次安装时装入，所以，在设备栈构造之后就不能插入过滤驱动程序了。在某些情况下，例如当 USB 设备插入 USB 总线时，低层过滤驱动程序可以修改功能驱动程序要执行的总线操作流。

驱动程序的具体实现分为以下几个部分：

- a) 首先是一个为 AddDevice,DispatchPnP,DispatchPower 和 DispatchCreat 例程设置入口点 DriverEntry 例程。
- b) 一个完成下面内容的 AddDevice 例程：
 - 调用 IoCreatDevice 创建一个独立的功能设备对象，并发送请求，接受响应数据。
 - 卸载刚刚建立的功能设备对象，再建立新的设备对象。
 - IoAttachDeviceToDeviceStack 把功能层和物理层“绑定”起来。
 - 调用 IoRegisterDeviceInterface 为它的设备设置一个接口。
 - 为 IRP_MJ_PNP 请求编写一个基本的 DispatchPnP 例程。该 DispatchPnP 例程处理具体的 PnP IRP。当驱动程序处理启动的 IRP,它必须调用 IoSetDeviceDevice In terface 以激活它先前注册的接口。
 - 为 IRP_MJ_POWER 编写一个基本的 DispatchPower 例程。
 - 为 IRP_MJ_CREAT 编写一个基本的 DispatchCreat 例程。
- c) 编译程序
- d) 编译驱动程序
- e) 安装驱动程序

驱动程序的编写工作完成后，下一步就是响应驱动程序。从图 7-4 中可以看到，上层的固件程序要和驱动程序通信，否则就是“协议出错”。

固件程序以.iic 文件的形式，固化在 EEPROM 24LC64 中^[76]。当用户用 EZ-USB FX2TM 芯片来做 USB 设备时，开发包所提供的固件架构可以简化固件程序的编写，并且加速 USB 外围的开发。针对用户而言，这个固件架构可以

利用 8051 程序代码来实现 EZ_USB 芯片的起始设置，USB 标准设备请求的处理以及 USB 闲置模式的电源管理服务。因此，用户仅需要提供 USB 描述符表以及实现外围功能的程序代码，即可完成完全兼容的 USB 设备，而 CYPRESS 公司开发包提供了副函数钩子与范例程序代码来帮助用户加速处理这个步骤。

固件架构实现了非常简易且兼容的设备(如图 7-6)，在程序起始时，固件架构会执行下列步骤：

- 1) 首先，设置所有的内部状态变量，也即是设置起始的初值。
- 2) 然后就会调用用户的初始设置函数 TD_Init()。待返回后，固件架构就会设置 USB 接口成为未配置状态，并且使能中断。
- 3) 此时，紧接着在 1s 的间隔内，开始重新列举（Renumerate）设备，直到设置（SETUP）封包收到端点 0 为止。
- 4) 一旦 SETUP 封包被检测到后，固件架构就会启动与其合作的工作分配器。
- 5) 而这个工作分配器机会顺序重复的执行下列的工作：

✧ 步骤一：调用用户函数 TD_Poll()

✧ 步骤二：是否决定标准设备请求是未定(或等待决定的)。如果已经觉得，它将会剖析所收到的命令请求，并且紧接着加以响应。

✧ 步骤三：是否觉得 USB 核心已经报告了 USB 中止（suspend）事件。如果已决定，它会调用用户函数 TD_Suspend()。若取得成功的返回，它将会测试回复（resume）事件。反之，如果未检测到，它将会把微处理器放入中止模式中。当回复事件被检测到时，它将会调用 TD_Resume()，并且连续的跳回至步骤三。

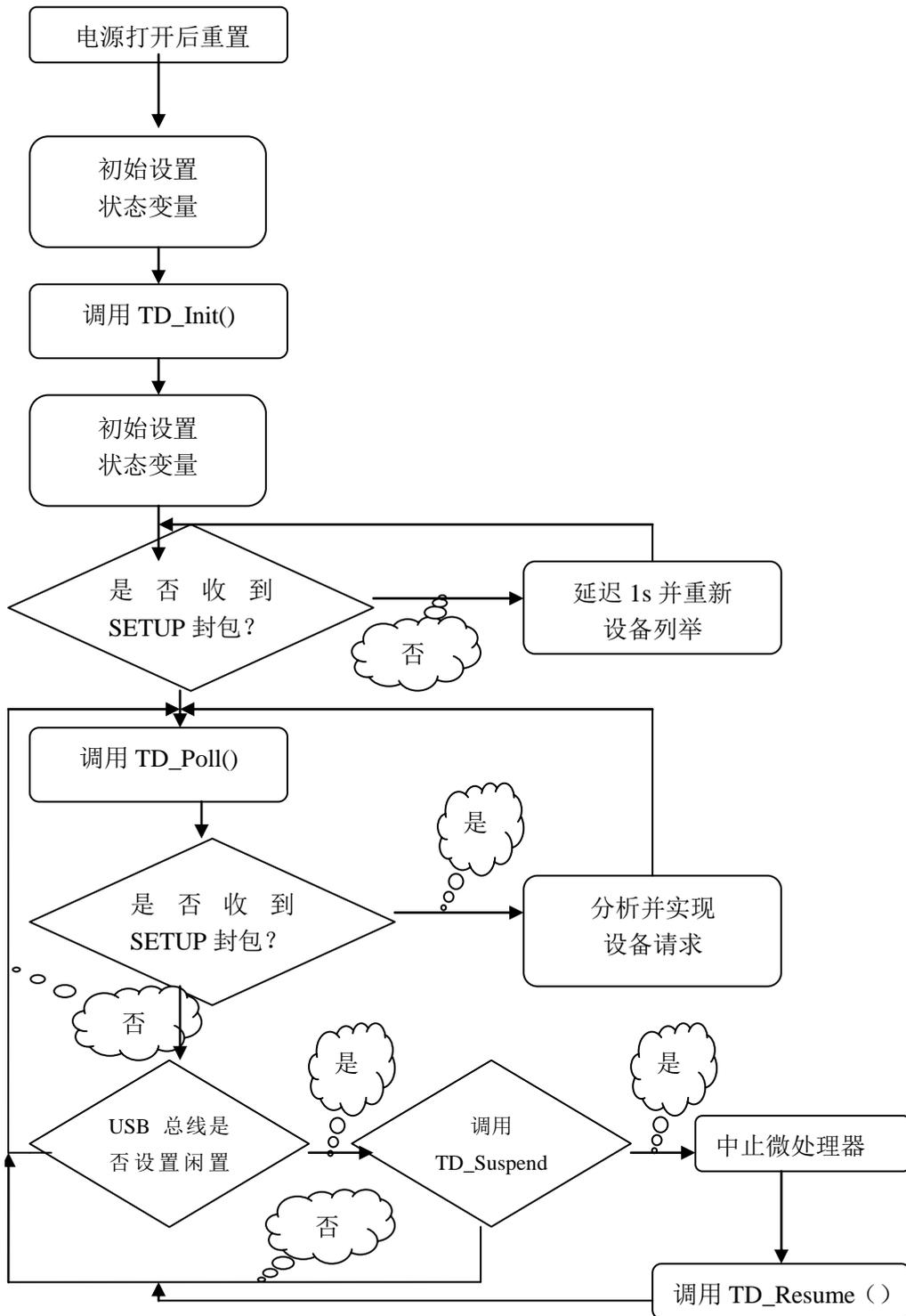


图 7-6 固件工作框图

固件架构是由 Keil C51 编译器与其整合开发工具所编写和构建出来的。也

就是说这些源文件必须在 Keil C51 编译器中加以执行，而其余的开发工具则不保证其能够正确的工作。根据设计方案，可以看出固件程序必须响应驱动程序的 A0 请求，才能达到相互通信的目的。通过学习 EZ_USB FX2 使用手册，可以发现 FX2 提供了一个供应商特别请求：“Firmware Load”，0xA0(作为 bRequest 的值，在请求的第 0 个字节 bmRequestType 为 “x10xxxxx”时，即暗示一个供应商特别请求时，0xA0 的值是永远有效的)。固件下载的请求是永远有效的，所以，即使在重列举之后下载的特征仍然有可能被用到。

本部分另一个创新点在于增加了一个自定义请求，它可以随意读取 EEPROM 24LC64 的数据，就好像“编程器”一样，在线编程。

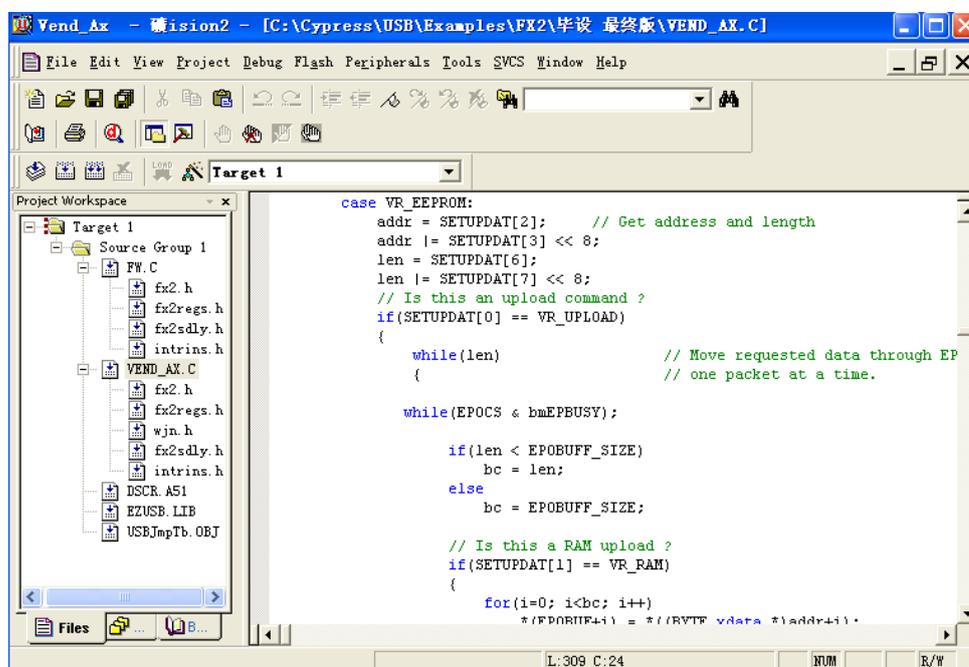


图 7-7 固件程序编译界面

CYPRESS EZ-USB FX2TM 设备将 RAM 划分为两个区域：一个是提供给程序代码与数据的；另一个则是给 USB 缓冲区与控制寄存器。如图 7-8(8k RAM) 所示。

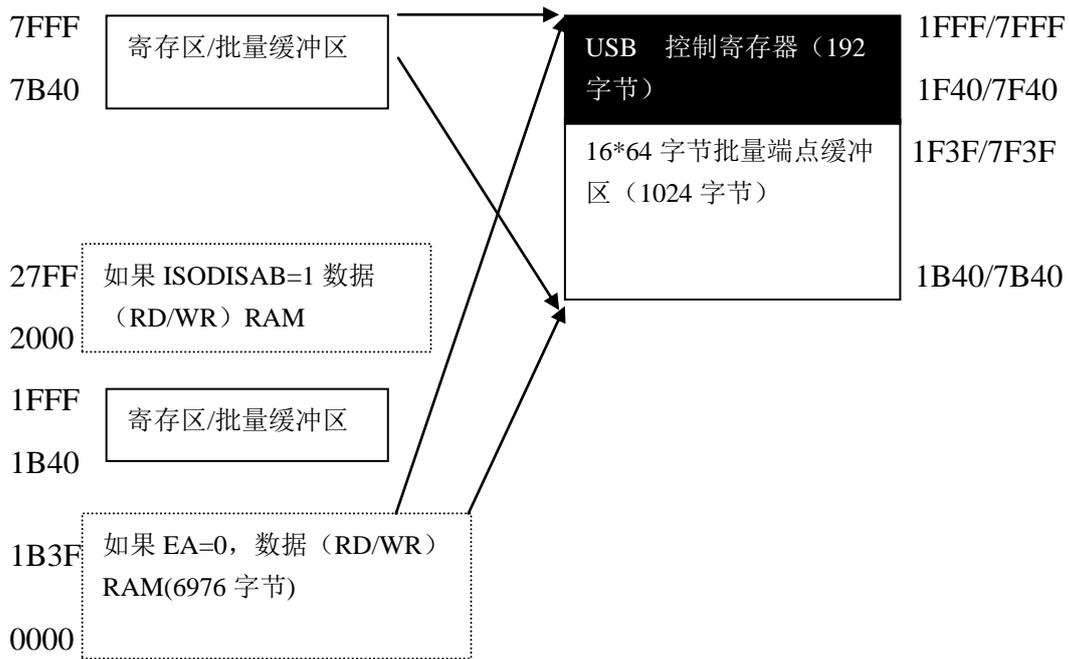


图 7-8 EZ-USB FX2 8K 内存映像图-----以十六位寻址

上图显示了 EZ-USB FX2 的内存结构，采用的是将程序和数据合用一个存储器空间结构，称为普林斯顿（Princeton）结构。根据普林斯顿结构的特点给每个设备添加一个编号，在 vend_ax 中添加代码：

```
BYTE xdata Digit[10]_at_0x60 = {0xC2,0x47,0x07,0x09}; //将 0x60 段定义为数据//段代码
```

```
BYTE code Digit[10]_at_0x60= {0xC2,0x47,0x07,0x09}; //将 0x60 段定义为程序代//码
```

在上面的语句中就是以 0xC2,0x47,0x07,0x09 响应驱动程序，从而“告诉”主机这个 USB 设备的编号是 0xC2,0x47,0x07,0x09。

编译该程序生成.hex 文件，再用命令 hex2bix -c 0x00 -f 0xC2 -i -p 0x0210 -v 0x4705 -o VEND_AX.iic VEND_AX.hex 把.hex 文件编译成.iic 文件，通过 CYPRESS3681 开发包提供的控制面板 control panel 烧录到 EEPROM 24LC64 中去。

7.5.2 基于 CYPRESS EZ-USB 2.0 芯片接口的硬件设计

CYPRESS 公司提供的开发包中包括了该公司开发板的原理电路图，通过分析该原理图我们发现这块开发板仅仅是作为学习之用，并不具有实际的工程意义，严格的说这是一块学习板，它能够完成外围 I/O 测试，端点对的测试，批量传输的测试，等时传输的测试，以及自定义请求的测试，如下图所示。

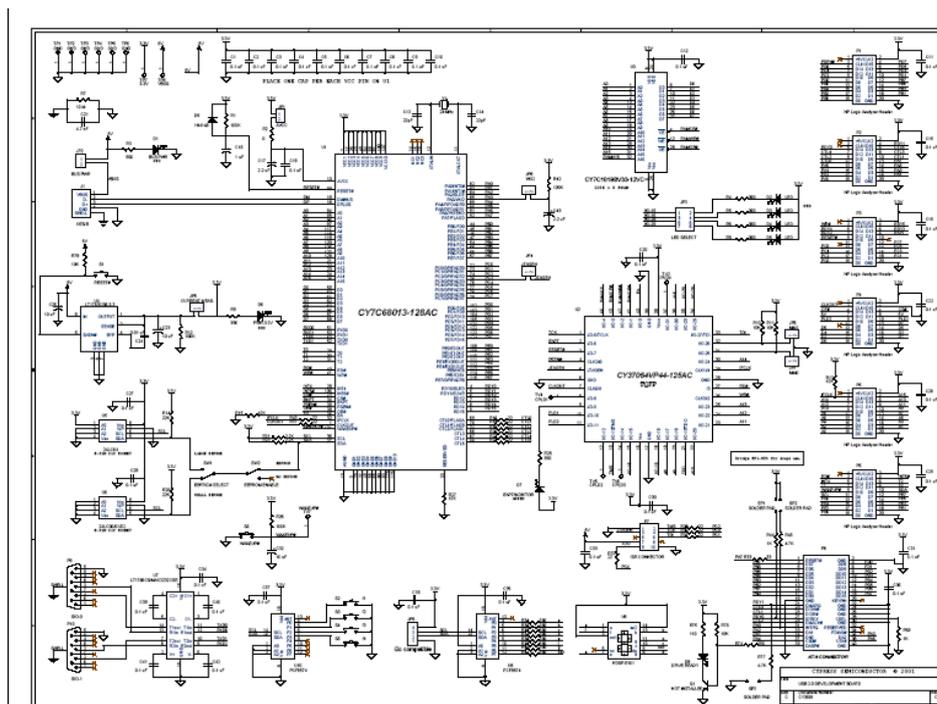


图 7-9 CYPRESS 公司提供的开发板原理图

根据工程的需要利用 CYPRESS Semi-conductor 公司的 EZ-USB FX2TM 设计新的 PCB 板接口电路（如下图），该图中以 EEPROM 24LC64 作为固件程序的初始存放区，以“C2”方式启动（“C2”EEPROM(将整个固件装入片内的 RAM)）



图 7-10 基于 CYPRESS EZ-USB FX2TM 芯片的 PCB 板

电路中以外部供电方式向 USB 芯片提供电源的电路板电源接口为 15 伏，经过芯片转换为 5 伏和 3.3 伏（CYPRESS EZ-USB FX2TM 芯片供电电压为 3.3 伏）。

根据图 7-4 的程序设计方案，可以看出固件程序必须响应驱动程序的 A0 请求，才能达到相互通信的目的。通过研究文献[76]，可以发现 FX2 提供了一个供应商特别请求：“Firmware Load”，0xA0（作为 bRequest 的值，在请求的第 0 个字节 bmRequestType 为“x10xxxx”时，即暗示一个自定义特别请求时，0xA0 的值是永远有效的）（见附件二）。固件下载的请求是永远有效的，所以，即使在重列举之后下载的特征仍然会被用到。

在工程设计中作者还增加了一个自定义请求，它可以随意读取 EEPROM

24LC64 的数据，就好像“编程器”一样，在线编程。

7.6 总结与讨论

图 7-11 是基于 Cypress EZ-USB 2.0 芯片的控制接口电路显示的参数，图中显示了第 0 号设备的基本运行情况。

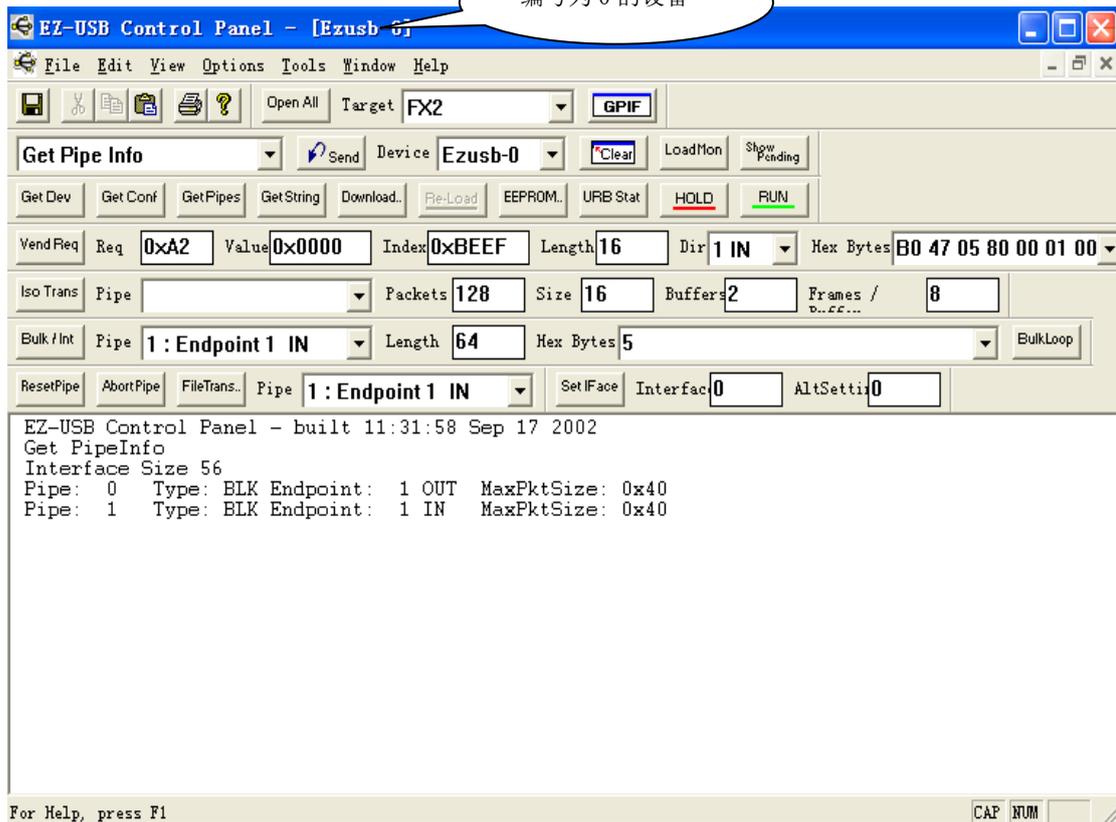


图 7-11 编号为 0 的 USB 设备

下表列出了新的接口电路同原有接口电路的比较。

表 7-1 原有接口电路和现有接口电路的比较

参数指标	以前的接口	新的接口
传输速度(理论值)	10Mbps	480Mbps
可扩展接口	允许连接 128 个收发器	可以级联 127 个外设
通讯方式	半双工	双工
是否支持热插拔	不支持	支持
外设最大间距	4000 英尺	3810m

通过对原有的多通道太阳望远镜多波带控制系统的深入研究,从设计 PCB 电路板,驱动程序,固件程序等方面着手,用最新的 USB 2.0 接口改造原有的控制接口,极大的提高了数据传输速度,也实现了一个 USB 接口控制多个电机的目标(如图 7-11 所示:第 0 号设备)。提出的“编号”的设计思想,为今后改造其他控制接口做好了准备。

以上介绍了将 USB2.0 接口技术应用在天文上的一个实例--基于 USB2.0 的多波带控制系统设计。现在我们总结一下应用这种接口带来的好处。

7.6.1 可扩展性强

USB2.0 接口技术支持热插拔,操作方便,可以扩展多个外设,如果需要扩展,只需要对软件做出细小的修改即可,使同一硬件平台可以适应很多不同的需要,就可以在相同的 USB 接口电路板上连接不同外部设备以满足不同规模的需要。而对于原有的 RS485 接口,由于采用九芯插头,从接插本身的角度来考虑,用户使用并不方便,不支持热插拔,连接线较多,容易接错。

7.6.2 传输速度快

原有的接口 RS-485 理论极限速度为 10Mbps,而 USB2.0 接口可以达到 480Mbps,USB 接口是最近几年广泛使用的接口,传输速度快,且发展迅猛,是商用设备的主要接口之一,目前已经有 USB3.0 试验性接口,但尚没有广泛使用,可以确定的是 USB3.0 相对于 USB2.0 还会有更大的提高。

7.6.2 成本低廉

USB 接口是目前商用机器上广泛使用的通讯接口,产量大,应用广泛,设备中嵌入 USB 接口已经成为一种趋势和潮流,因此该设备的成本相对低廉,并且可以根据需要裁剪。此外,由于这种接口设备应用广泛,开发者技术获取的渠道较多,这也将大大减小开发的时间成本。

第八章 结束语

太阳磁场的观测研究，是太阳物理研究的重要课题。当今太阳物理研究的重大成果和主要困难都与太阳磁场的观测及理论研究有关。以高时间分辨率、高空间分辨率来观测太阳磁场，可以更加深入地了解太阳活动的物理机制，为目前困扰太阳物理界的一些难题提供观测资料，同时，提供预报太阳活动以及相关灾害的多种参数。发展高时间分辨率、高空间分辨率的太阳磁场观测是以实验物理为主的太阳物理研究永恒的主题。

本论文正是基于以上背景，以实现高速度，高时空分辨率太阳磁场观测为目的，结合当代太阳物理研究的需求和现代的高科技水平，研究实现可用于太阳望远镜和其他类似望远镜的实时相关处理系统,导行方案和高速控制接口。

本文的主要贡献有：

1. 完成了具有强大处理能力的新一代高速实时相关处理系统的设计研制。该终端系统采用 GPU 处理器实时并行处理大批量数据，并成功的应用于怀柔太阳观测基地 35cm 磁场望远镜，大大提高了观测质量，获得了满意的太阳观测资料。将 GPU 技术应用于天文学的研究目前在国际上尚属前沿课题。此项技术的应用成功，大大提高了天文观测终端设备的处理能力，可以说，它将望远镜终端系统带入了并行处理时代。

2. 重点讨论并完成了处于国际水平的基于质心算法的大面阵 CCD 全日面像自动导行方案，通过计算机模拟计算，利用“二值化”模型优化了计算数据，得到该方案导行精度的主要参数及技术指标，并应用于怀柔太阳基地 H α 全日面磁场望远镜，取得了预期的效果。这是国内第一台利用大面阵 CCD 进行高精度自动导行技术的太阳望远镜，该技术的研制成功将为未来太阳望远镜自动导行技术的发展提供宝贵借鉴和依据。

3. 提出了望远镜控制系统采用 USB 接口技术的新设计，并成功的应用于多通道太阳望远镜的多波带调制系统上，大大提高了数据的传输效率并实现了数据的自动寻址。这项工作的完成，提高了控制系统的响应速度和稳定性，精简了太阳望远镜控制系统的线路结构，对于未来红外太阳望远镜控制接口的设

计具有一定的指导意义。

总之，通过本论文的研究论证及已经取得的部分结果，为下一步的研究工作打下了坚实的基础；为太阳望远镜跟踪精度与终端处理技术方面的研究提供了有益的参考和帮助；尤其为将 GPU 并行计算应用于望远镜的终端系统这项技术的发展提供了技术储备和借鉴。

博士期间作者发表的论文和申请的专利

发表的论文:

1. [题目] 基于 USB 协议的多波带控制系统设计
[作者] 肖江 胡柯良 林佳本 申基 邓元勇
[刊物] 天文技术与方法, Vol.4 No.4 Sep.2007 , pp 360-368.
2. [题目] 用大面阵 CCD 实现全日面像自动导行
[作者] 肖江 胡柯良 林佳本 申基 邓元勇
[刊物] 光学 精密工程, Vol.16 No.9 Sep.2008, pp1589-1594
3. [题目] CUDA 矩阵乘法并行计算和快速傅里叶变换性能测试
[作者] 肖江 胡柯良 林佳本 申基 邓元勇
[刊物] 计算机工程, No.12 Sep.2009

申请的专利:

发明专利: 全日面导行方法和系统 申请号: 200810101387.4

致谢

在此论文即将结束之际，心中感慨万千。回忆这几年走过的求学之路，步步都离不开师长、同仁给予的关怀与鼓励。

首先，我要感谢我的导师邓元勇研究员。他治学严谨，却勇于创新；他思维敏捷，却肯于脚踏实地；他对科学那种锲而不舍的执著精神深深感染着我。在他的悉心指导与影响下，我不仅在专业能力上获得了长足的长进，而且对科研方法和系统研究的全局把握方面有了一定的能力，在我的论文写作过程中，花费了很多精力，从文章整体构思、章节编排到文字修改，处处包含着他的关心与指导，使我受益匪浅，用良师益友称呼他是再贴切不过的了。我很庆幸在我的人生路上曾有一位这样的导师为我引路。

感谢导师胡柯良副研究员，在我的课题研究中给予的大力支持和帮助，他不仅对我的课题提出很多有益的建议，而且亲自验证我的研究结论。在他的悉心关怀和指导下，我的科研工作取得了长足的进步。他国际化的视野，严谨勤奋的作风，都让我永志不忘，深刻影响着我今后的工作和生活。如果没有他的悉心指导完成本博士论文中的工作是不可想象的，谨在此向他表示崇高的敬意和最衷心的感谢！

感谢林佳本工程师在望远镜终端系统实验时提供的宝贵帮助，使我的工作得以顺利进行；感谢申基同学在望远镜自动导行系统测试时提供的宝贵意见；感谢南京天文光学技术研究所章海鹰高工、倪厚坤技师在工作中对我的鼓励；同样感谢研究生部杜红荣和艾华两位老师在我担任国家天文台学生会主席期间对我工作的关心和帮助。

感谢张洪起研究员、李威站长对本人工作的大力支持；感谢张枚研究员、苏江涛副研究员、王东光高级工程师、林刚华高级工程师对我工作的支持；感谢汪国萍老师对我工作和生活上的热情关怀和帮助，感谢怀柔太阳观测基地全体老师和同学在技术和其他各方面给予的帮助与合作。

参考文献:

- [1] 林元章, 太阳物理导论, 科学出版社, 2000
- [2] Hale,G.E.,Ap.J.,1908,28,315
- [3] Lyot,B.,Comptes Rendus,1933,197,1593
- [4] Ohman,Y.,Nature,1938,pp141-157
- [5] Bacbcock,H.W.,Ap.J.,1953,pp118-387
- [6] Stepanov V.E and Severny A.B.,IZV.Crim.Astrophy.Obs.28(1962),116
- [7] Beckers,J.M.,Solar Phys.,1968,3,258
- [8] Mein, P.; Blondel, M., Multi-Channel Subtractive Double Pass Spectrograph, Solar Physics, 27, 48
- [9] 艾国祥, 胡岳风, 中国科学 A,8(1986a),pp889-896
- [10] 艾国祥, 胡岳风, 中国科学 A,9(1986b),987
- [11] Unno,W.,Publ.Astron.Soc.Japan,8,108.
- [12] Rachkovsky, D. N., 1962, Izv. Krymsk. Astrofiz. Obs., 27, 148
- [13] Jefferies, J., Lites, B. W. and Skumanich, A.,1989, ApJ, 343, 920
- [14] Landi, D. E., 1982, Solar Phys., 79, 291
- [15] P.H.Scherrer, R.S.Bogart, et al., 1995, Solar Phys, vol.162, pp129-188.
- [16] Livingston, W., et al, Applied Optics, 1976 Vol.15, No.1, pp40
- [17] D.F.Elmore, B.W.Lites, et al., SPIE. 1992, Vol.1746, pp22-33
- [18] Grigoryev, V. M., Osak, B. F, Kobanov, N. I., et al, 1981, Issled. Geoman. Aeron. Fiz., Soluta, 56, 129
- [19] 王同江, 艾国祥, 邓元勇, 1996, 北京天文台台刊, 28, 31
- [20] <http://www.hudong.com/>
- [21] 国家自然科学基金“十一五”发展规划
- [22] 林佳本, 山东大学硕士论文, 2002
- [23] 张基温, 计算机组成原理教程, 清华大学出版社, 2008
- [24] NVIDIA_CUDA_Programming_Guide_2.0, <http://www.nvidia.com>
- [25] 宋宜昌, 世界最快的超级计算机, 时事报告, 2008, 7, 64-65
- [26] 曙光 4000A 计算机系统三年运行情况分析, 高性能计算发展与应用 2008, 22 (1), 43-50
- [27] 莫则尧, 陈军, 曹小林等译. 并行计算综论. 北京: 电子工业出版社, 2005
- [28] NVIDIA CUDA 统一计算架构编程模型, <http://www.nvidia.com>
- [29] CUDA——走向 GPGPU 新时代, 程序员, 2008, 3
- [30] CUDA 软硬件环境简介, 2008, 3
- [31] 邓培智, CUDA 编程模型, 2008,5
- [32] Edited by Hubert Nguyen Foreword by Kurt Akeley Fast N-Body Simulation with CUDA , GPU GEMS3: p677-695
- [33] Aarseth, S. 2003,Gravitational N-Body Simulations,Cambridge University Press.

- [34] http://www.science.uwaterloo.ca/~hmerz/CUDA_benchFFT/
- [35] Technical Brief: NVIDIA GeForce 8800 GPU Architecture Overview.Santa Clara, California. 2006.11
- [36] The CUDA Compiler Driver NVCC, http://www.nvidia.com/object/cuda_develop.html
- [37] Intel Math Kernel Library for the Windows* Operating System Users' Guide, <http://www.intel.com>
- [38] <http://msdn.Microsoft.com>
- [39] Matteo Frigo, Steven G. Johnson., FFTW, <http://www.fftw.org>
- [40] CUDA CUBLAS LIBRARY, Sep.2007, http://www.nvidia.com/object/cuda_develop.html
- [41] CUDA CUFFT LIBRARY, Sep.2007, http://www.nvidia.com/object/cuda_develop.html
- [42] geforce 和 quadro 的九大区别, <http://cuda.csdn.net/>
- [43] <http://www.nvidia.com>
- [44] <http://www.dell.com>
- [45] 章毓晋, 图像处理和分析, 清华大学出版社, 2002: 47~48
- [46] 邓元勇, 张斌, Application of a Large Capacity Real - time Image Acquisition System(I) “Deep Integration ” Magnetogram of Quiet Region on the Sun” [J].Astrophysics Reports, 1999; 34: 15~19
- [47] 邓元勇, 张斌, 王东光.New Image Acquisition System in Huairou Solar Observing Station[J].Astrophysics Reports, 2000; 35: 6~7
- [48] 林佳本, 邓元勇, 胡柯良, 局部相关跟踪算法在太阳磁场观测中的应用, 计算机工程与应用, 2006.27: P203-205
- [49] LYNX_Hardware_User_Manual, <http://www.imperx.com>
- [50] 王东光, 中国科学院研究生院博士论文, 2003
- [51] Device Driver User Manual For PCI-1730, <http://www.advantech.com>
- [52] 实时相关跟踪图像处理系统(英文), 天文研究与技术 , Vol 3 (4), 2006.12
- [53] 李哲英, 电子技术及其应用基础 模拟部分, 高等教育出版社, 2003
- [54] 张强, Access 2007 入门与实例教程, 电子工业出版社, 2007
- [55] (美) 巴尔特著, 谢晖, 许伟译, Access 2007 开发指南, 人民邮电出版社, 2008
- [56] <http://www.langchao.com/>
- [57] 张斌等, 多通道太阳望远镜后端数据处理系统, 光学技术增刊, 152-158, 1995
- [58] 秦伟, 中国科学院研究生院硕士论文, 2004
- [59] 胡柯良, 中国科学院研究生院博士论文, 2005
- [60] 张兴磊, 杨丽丽, 张东风, 高龙, 张治坤 太阳自动跟踪系统设计, 农业工程技术(新能源产业), 2008 (6): P11-13
- [61] 陈新知, 多通道组合太阳望远镜驱动电控设计报告, 光学技术增刊, 1995, p167-p173

- [62] Kuveler G., Wiehr E., Thomas D., et al., “Automatic Guiding of the Primary Image of Solar Gregory Telescopes”, Solar physics, 1998, 182, 247-255
- [63] Wulser J., Hudson H., Nishio M., et al, “Precise Determinations of the Coordinate system for YOHKOH Telescopes and the Application of a Transit of Mercury”, Solar Physics, 1998, 180 131-156
- [64] Thompson W. and Carter M., “EUV Full-sun Imaging and Pointing Calibration of the SOHO/CDS” Solar Physics, 1998, 178, 71-83
- [65] 吴春晖, 朱庆生, 周小军, 全日面太阳光学和磁场望远镜的自动跟踪与导行方法, 天文研究与技术,2007.6: P147-152
- [66] Davis Chapman 著 孟文 译, VC++.NET 入门, 中国电力出版社, 2003
- [67] David Sceppa,ADO 编程技术, 清华大学出版社,2001
- [68] 施伯乐等译, 数据库处理--基础、设计与实现(第七版), 电子工业出版社,2001
- [69] The MegaPlus -Model 4.2i Camera User’s Manual. [http:// www.redlake.com](http://www.redlake.com)
- [70] MarkNelson 著, 串行通信开发指南, 中国水利水电出版社, 2001
- [71] Y.Y. Deng, G.X. Ai, J.S. Wang, et al., “Reports on the Test Observation of Multi-Channel Solar Telescope”, 1997, Solar Physics,, 173, 207-221
- [72] Ai, G., and Hu, Y., “Multichannel Birefringent Filter. I. Principle and Video Spectrograph”, 1987, Scien. Sinica A, Volume 30: p 868-876
- [73] 周立功, PDIUSB12 USB 固件编程与驱动开发, 北京航空航天大学出版社, 2002
- [74] 边海龙, 贾少华, USB 2.0 设备的设计与开发, 人民邮电出版社, 2004
- [75] USB 2.0 规范, <http://www.usb.org/>
- [76] EZ-USB FX2 Manual Technical Reference, <http://www.cypress.com>, 2002
- [77] CY7C68013EZ-USB FX2™ USB Microcontroller High-speed USB Peripheral Controller, <http://www.cypress.com>, 2002
- [78] Windows 2000 DDK, <http://www.microsoft.com/whdc/devtools/WDK/>