

文章编号 1004-924X(2010)10-2262-07

运动模糊视频图像在图形处理器平台上的实时恢复

王 晶¹, 李 仕²

(1. 中国科学院 长春光学精密机械与物理研究所, 吉林 长春 130033;

2. 杭州电子科技大学 图形图像研究所, 浙江 杭州 310018)

摘要:提出了一种图形处理器优化编程方法,用于实现运动模糊视频图像的实时恢复处理。根据计算统一设备架构(CUDA)的硬件框架特征对 GPU 的线程块及线程数量进行优化配置,并引入了一种自动内存接合访问的方法,使得 GPU 的硬件资源得到充分利用。根据图像频谱的对称性去除冗余信息,减少了图像算法在频谱滤波时的数据量,使得 GPU 对内存的访问次数下降,从而提升了算法效率。实验表明,本文提出的 GPU 方案的计算性能比传统的 CPU 平台方案提升了一个数量级,半频谱滤波设计使总时间开销减少 20% 以上,实验结果证明了本文方案的可行性及有效性。

关键词:视频图像;图像恢复;图形处理器;计算统一设备架构;优化编程

中图分类号: TN941.1; TP391.4 **文献标识码:** A **doi:** 10.3788/OPE.20101810.2262

Real-time restoration of motion-blurred video images on GPU

WANG Jing¹, LI Shi²

(1. *Changchun Institute of Optics, Fine Mechanics and Physics,*
Chinese Academy of Sciences, Changchun 130033, China;

2. *Institute of Graphics & Image, Hangzhou Dianzi University, Hangzhou 310018, China)*

Abstract: A Graphic processing Unit(GPU) optimization programming method is presented to apply to the real-time restoration of motion blurred video images. The blocks and threads run on the GPU are optimally set based on the hardware structure of Compute Unified Device Architecture (CUDA), and a memory access method is introduced to implement automatic coalesced access. These are required to make sure the full utilization of the GPU's hardware resource. According to the symmetry property of FFT spectra, the redundant information in the frequency spectrum is eliminated and the number of frequency data filtered by the image algorithm is decreased, by which the amount of GPU memory access for realizing the algorithm optimization is reduced and the computing efficiency is improved. The experiment indicates that the proposed GPU project can improve the computing performance by 10 times as compared with the conventional CPU project, and the design of half-spectrum filtering can reduce the above time consumption by 20%. The experimental results confirm the feasibility and the validity of proposed method.

Key words: video image; image restoration; Graphic processing Unit(GPU); Compute Unified Device Architecture(CUDA); optimization programming

收稿日期:2010-06-07;修订日期:2010-08-06.

基金项目:国家 863 高技术研究发展计划资助项目(No. 2008AA121803);国家 973 重点基础发展规划资助项目(No. 2009CB72400607)

1 引 言

计算机的发展水平日新月异,PC 机的 CPU 已在短短的几年内从原有的单核发展到现今常见的双核及四核,2010 年上半年,INTEL 甚至展示了一款被称为单芯片云计算的 48 核 CPU。然而,CPU 的运算能力还是受到了新技术强有力的挑战,甚至威胁到其在传统大、中型计算机领域的地位,这个挑战者正是以超越摩尔定律速度发展的图形处理器(GPU)^[1]。

目前,GPU 已从原先单纯的图形处理领域发展延伸到通用计算领域,利用硬件的可编程性和并行性可实现一些复杂模型的快速通用计算。由于 GPU 出色的浮点运算能力、超大的内存带宽和相对较低的价格,基于 GPU 的通用计算(General Purpose Computing on GPU,GPGPU)已经对许多应用领域产生了极大的推动作用。如 Moreland 等^[2]在 GPU 上实现了 FFT 算法;Kruger 等^[3]在 GPU 上完成了求解大型稀疏线性方程组的高斯-赛德耳迭代法和共轭梯度法;Bolz 等^[4]则在 GPU 上实现了稀疏非结构化矩阵求解;Harris 等^[5]利用 GPU 来加速计算云层的动态变化;Cuntz 等^[6]在 GPU 上利用粒子水平集方法可视化三维流场;Zhou 等^[7]在 GPU 上构建 kD 树并将之用于实时光照。

本文拟解决的运动模糊视频图像实时恢复处理问题是图像处理领域的一个经典命题。该问题长期以来一直被图像恢复算法的庞大运算量所困扰,使得相对成熟的图像处理技术迟迟不能在工程化应用中展现它应有的价值。例如在航空侦察飞机上的航拍视频,由于速高比的存在^[8]使得视频图像具有运动模糊,如果该视频图像不能得到实时恢复,侦察飞机上的视频跟踪系统将无法正常工作。因此,本文提出了一种基于 GPGPU 处理技术的运动模糊视频图像实时恢复处理算法,介绍了该算法的优化配置并进行了实验验证。

2 基于 GPU 平台的帧图像恢复算法

2.1 维纳滤波

运动模糊图像恢复算法主要分为空间域解卷积算法和频率域滤波算法。空间域解卷积算法的

运算量过大,为提高图像算法的处理速度,通常是采用循环矩阵对角化的数学方法,利用 FFT 算法对高运算量的空域解卷积算法进行提速^[9-10]。为此,本文直接采用常见的频域滤波算法-维纳滤波^[11]来恢复运动模糊视频的帧图像。当图像噪声为白噪声时,得到如下维纳滤波表达式:

$$F(u, v) = \frac{1}{H(u, v) \sqrt{\frac{\|H(u, v)\|^2}{\|H(u, v)\|^2 + k}}} G(u, v), \tag{1}$$

式中: $G(u, v)$, $F(u, v)$, $H(u, v)$ 分别代表滤波前、后图像及点扩散函数的频谱数据。式(1)中的 k 是信噪比相关的系数,通过 k 值的交互选择可以找到最好的视觉效果。实践表明,当 k 处于 $[0.001, 0.01]$ 时,恢复效果比较理想^[8]。

2.2 基于 GPU 的并行处理

GPU 的架构特点是适合拿来作数据的并行计算。GPGPU 的思想是将一个问题大量地拆解成多个相同、且彼此不相关的小问题,交给 GPU 来做并行处理,达到加速计算的效果^[12]。模糊视频是基于时间的序列,对之进行并行处理的方案应该是在下一帧图像传来之前,将当前帧图像数据进行并行运算的拆解,通过对帧图像的并行加速处理来实现整个视频数据的实时处理。

由维纳滤波的表达式(1)可发现:滤波的本身是对频谱 $G(u, v)$ 上的数据进行逐个处理, $F(u, v)$, $G(u, v)$, $H(u, v)$ 在滤波时,各数据点对应的下标是相同的,且各频谱数组的相邻数据在滤波时是相互独立的。因此帧图像的维纳滤波可以被分解成一系列相同、且彼此不相关的滤波运算,

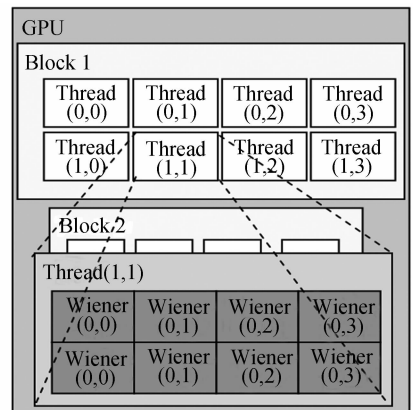


图 1 维纳滤波并行处理图解

Fig. 1 Illustration of parallel Wiener filters

每一组数据的滤波都可以指定一个线程来处理, 以此实现算法的并行加速。假设一个线程一个周期时间内可以完成一个式(1)所示的滤波计算, 那同时有 N 个线程并行执行时, 则一个周期时间可以同时完成 N 个滤波计算。图 1 为维纳滤波在 GPU 上并行处理的图解。

3 计算统一设备架构(CUDA)

为克服一般 GPGPU 硬件编程工作量大、程序可读性较差及难以维护调试等不足, 2006 年, NVIDIA 公司推出基于 G80 显卡的开发工具包 CUDA^[13] 来支持图形处理器的通用计算。CUDA 将 GPU 视作一个数据并行计算设备, 并且无需把这些计算映射到图形 API^[14], 使得 GPGPU 编程能通过 C 语言和一些 CUDA 的扩展来实现, 从而减少 GPU 编程的开发难度。考虑到 CUDA 能为高性能计算提供并行数据缓冲和线程执行管理器的支持, 并优化了 GPU 和 CPU 之间的数据通信路径, 所以本文采用 CUDA 作为 GPU 的编程环境。

3.1 CUDA 架构

CUDA 的硬件架构以 NVIDIA 在 2007 年发布的 GeForce 8800 为例, 该 GPU 包含 16 个流多处理器 (SM), 每个 SM 包含 8 个线程处理器 (TP), 共 128 个处理核心。SM 采用单指令、多线程 (SIMT) 架构将每个线程映射到一个 TP, 各 TP 使用自己的指令地址和寄存器独立执行。SIMT 单元以 32 个并行线程为一组来创建、管理、调度和执行线程, 这样的线程组称为一个 warp。构成 warp 的各个线程在同一个程序地址一起启动, 也可独立执行。

3.2 Block 及线程的设置

CUDA 以 Block 块为单位做线程管理, 每一个 Block 最多有 16 个 warp, 即一个 Block 里面最多有 512 个线程。Block 里的线程个数要为 32 的整数倍才能避免在执行到个别 warp 组时出现部分硬件资源闲置的情况。每个 SM 有 8 个 SP, 最多能同时跑 8 个 Block 的线程, 但 SM 在分配资源的时候, 一次把所有的寄存器和共享缓存分配给所有的 Block 的所有的线程。所以当 Block 及 Block 内的线程数过多的时候会因为单个线程

可以利用的寄存器和共享缓存变少而导致程序的执行效率降低。

4 算法的优化配置

维纳滤波的运算量主要来自图像的正逆 FFT 运算及频域滤波。CUDA 有提供专门的 FFT 库函数 CUFFT。借用 CUFFT, 下文给出维纳滤波时的一些其它优化配置。

4.1 频谱对称性质

二维离散傅里叶变换的表达式如(2)式所示:

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) W_M^{ux} W_N^{vy}, \quad (2)$$

式中: M, N 表示复数二维序列 $f(x, y)$ 的维度, $W_n = e^{-i2\pi/N}$ 。当 $f(x, y)$ 的虚部全部为“0”时, 由式(2)能得到如式(3)所示的共轭关系。

$$F(u, v) = F^*(M-u, N-v). \quad (3)$$

在数学上可以证明: 两复数间的基本数学运算(加、减、乘、除)的结果与这两复数的共轭复数之间的相应基本数学运算的结果互为共轭^[15], 所以可以只对 FFT 频谱的一半进行频域滤波, 然后通过式(3)的共轭对称关系获得完整的维纳滤波结果。

4.2 GPU 编程优化

设普通 GPU 上的 SM 为 m 个, 为充分占用 SM, 将其划分为 $2m$ 个 Block 块, 使每个 SM 对应 2 个 Block 块; 为充分运用 warp 并行, 及平衡 SM 上的寄存器和共享缓存的使用情况, 在每个 Block 块上开辟 256 个线程, 则 GPU 上共有 $512m$ 个线程在做并行滤波处理, 每个线程分配一定的运算量。当采用半频谱滤波方式时, GPU 上每个线程的工作量能减半。

每帧图像的数据量远大于 GPU 上的共享缓存大小, 将一帧图像数据全部存入共享缓存是不可能的, 所以每帧图像的数据只能先存放在显卡内存上。GPU 对显存数据进行访问时, 每次的访问时间需要 400~600 个时钟周期, 因此需充分借用共享缓存做中间数据的存储, 来避免 GPU 频繁地访问显存; 另外, 可将二维的视频帧图像数据以一维数组的形式展开, 使之自动满足内存对齐的要求以实现显存的接合访问(使每半 warp 的 16 个 threads 同时访问显存)。具体优化过程见图 2。

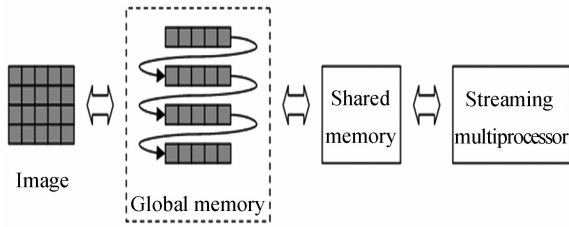


图 2 GPU 优化编程的数据流程

Fig. 2 Illustration of GPU data processing

4.3 数据分析

单帧视频图像一次完整的维纳滤波由正、逆各一次的 2 维 FFT 运算及频谱滤波组成。一次 N 点的 FFT 的运算量为 $2N\log_2 N$ 次实数乘法和 $3N\log_2 N$ 次实数加法^[16]。每个线程分担到的运算量为 $2N(\log_2 N)/512m$ 次实数乘法和 $3N(\log_2 N)/512m$ 次实数加法。GPU 线程的运算单元是 fused multiply-add 单元, 可以同时进行一次浮点乘法 and 一次浮点加法。因此, 一次正、逆二维 FFT($1\ 024 \times 1\ 024$) 运算, 每个线程分担到的 FFT 运算时间合计大约为 100 个时钟周期, 频谱滤波时每个线程所需的运算时间约为 64 个时钟周期, 每个线程在一次完整维纳滤波中合计运算时间为 164 个时钟周期。

为提高 GPU 运算速度, 用接合访问的方式将显存的访问时间平摊到半 warp 的 16 个线程。对显存完成一次存/储过程, 每个线程分摊到的访问时间为 50~75 个时钟周期。可见显存的访问时间即使经过接合访问的方式进行分摊后开销还是很大, 而采用半频谱滤波方式的主要目的也是为减少对显存的访问次数。

针对一帧 $1\ 024 \times 1\ 024$ 分辨率的视频图像进行维纳滤波, 每个线程分摊到的时间开销是: 全频谱运算合计需要 $100 + 64 + 60 = 224$ 个时钟周期 (显存访问时间假设为 60 个时钟周期), 半频谱运算合计需要 $100 + 64/2 + 60/2 = 162$ 个时钟周期, 半频谱运算所需时间约为全频谱运算的 72.3%。

5 实验结果

本文实验所用 CPU 为 INTEL 公司在 2008 年 11 月发布的 Core i7 920@2.66GHz, 所用 GPU 为 NVIDIA 公司在 2007 年 11 月发布的 GeForce 8800 GTS (500M)。

5.1 不同平台对比实验

为验证本文所提方案的可行性, 作者先分别在 CPU 平台及 GPU 平台上针对不同分辨率图像做维纳滤波时间测试对比实验。两平台上所运行的维纳滤波均用实数 FFT 优化配置过。

图 3 为对比实验的测试结果, 它表明: 虽然实验所用 GPU 与 CPU 的硬件水平相差一年, 但在本试验同等工作强度下 GPU 的处理效率依旧领先 CPU 一个数量级, 展示了很强的运算性能。该结果证明本文 GPU 方案能够满足对 25 frame/s 的 PAL 格式视频进行帧图像的实时恢复处理的需求。

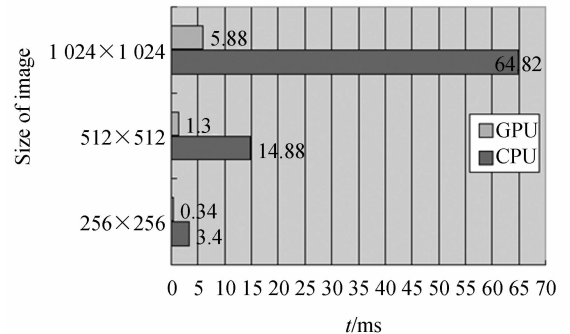


图 3 CPU 平台与 GPU 平台上维纳滤波对比测试结果

Fig. 3 Test results of Wiener filters on CPU and GPU

5.2 优化配置的实验验证

为验证文中第四节所提优化方法的有效性, 本文在 GPU 平台上分别进行全频谱及半频谱维纳滤波实验, 并做实验数据对比分析, 其中对全频谱进行维纳滤波的程序代码由 CUDA 自带的例程“convolutionFFT2D”修改而来。在半频谱的维纳滤波实验中, 使用 CUFFT 库函数的实数 FFT 函数来生成实验所需的半频谱, 使得半频谱数据在显存上连续地存储, 以保证频谱数据的接合访问。由于 CUFFT 的实数 FFT 函数本身相对复数 FFT 函数并不带来运算量上的减少^[13], 所以 CUFFT 实数 FFT 的使用只是影响对显存上数据的访问次数。

对不同尺寸图像的全频谱及半频谱分别进行维纳滤波所得实验数据如图 4 所示。图 4 的折线曲率显示, 随着图像尺寸的增大, 两种维纳滤波所耗用时间增幅均要大于各自运算量的增幅, 印证了 GPU 对显存上数据访问的开销要大于运算量的开销; 图 4 的折线曲率同时表明, 传统的全频

GPU上不同算法时间消耗对比图

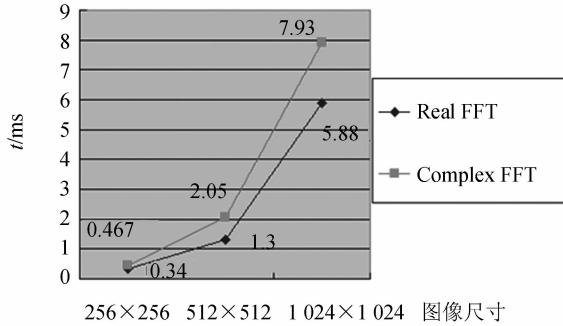


图 4 GPU 平台上不同维纳滤波方式所得实验数据
Fig. 4 Results of Wiener filters with difference FFTs on GPU

半频谱滤波与全频谱滤波的时间比值图

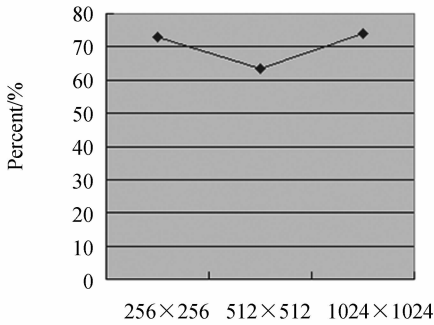


图 5 两种频谱滤波的时间比值图
Fig. 5 Ratio of two filter results

谱维纳滤波在图像尺寸增大的时候,所耗用的时间增幅要比本文所提的半频谱维纳滤波所耗用时间的增幅要大,即随着图像尺寸的增大,本文方法所耗用时间的增加比例比传统方法要小,达到了方法的预期目的。图 5 数据为将图 4 中两种算法的对应时间数据进行比值所得的结果,数据显示本文方法能将 CUDA 自带例程上传统方法的时间开销减少 20% 以上,验证了 4.3 节中数据分析的结果。

5.3 半频谱维纳滤波图像恢复效果

实验通过架在匀速转动平台上的逐行扫描相机(如图 6 所示)获得室外的运动模糊视频图像(如图 7 所示),用最优窗算法^[17-18]对之进行边界预处理,然后使用本文提出的方法进行运动模糊恢复,所得恢复图像如图 8 所示。图 8 中的远、近景信息的细节均得到一定程度的恢复,该实验结果进一步验证了本文方法的可行性。

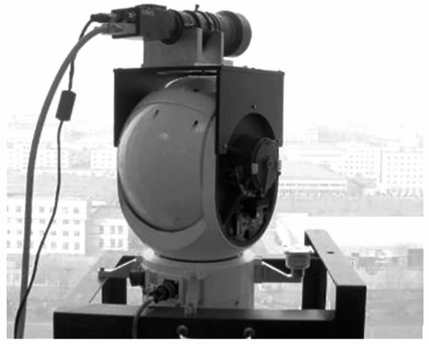


图 6 运动模糊视频实时采集平台
Fig. 6 Platform for motion-blurred video capturing



图 7 真实的运动模糊帧图像
Fig. 7 Real motion-blurred frame



图 8 恢复处理的帧图像
Fig. 8 Restored frame

6 结 论

本文立足于解决运动模糊视频图像的实时恢复处理问题。实验结果表明,用 GPU 处理 1024×1024 视频单帧图像只需 5.88 ms,满足视频帧图像实时恢复处理的要求,验证了文中所提方案

的可行性。

在算法的优化配置部分, CUFFT 库里的实数 FFT 函数实质上是复数傅里叶变换, 它只是用传统实数 FFT 的接口形式对 CUFFT 中的复数 FFT 进行封装, 并没有利用实数 FFT 的对称特性减少 FFT 函数在 GPU 上的运算量。因此, 文中实数 FFT 的使用本身并没有带来运算量的减少, 只是帮助求得无冗余信息的频谱, 使得频域滤

波的数据量减半, 减少了 GPU 对显存的访问次数。

实验结果表明, GPU 对显存访问次数增加所带来的时间开销要比运算次数增加所带来的时间开销大, 证明了立足于减少显存访问次数的改进方法的可行性及有效性。本文方法并不是仅适用于维纳滤波, 它可以被推广到其它图像频域滤波的应用上。

参考文献:

- [1] 吴恩华. 图形处理器用于通用计算的技术、现状及挑战[J]. 软件学报, 2004, 15(10):1493-1504.
- WU E H. State of the art and future challenge on general purpose computation on GPU [J]. *Journal of Software*, 2004, 15(10):1493-1504. (in Chinese)
- [2] MORELAND K, ANGEL E. The FFT on a GPU [J]. *Proceedings of the ACM SIGGRAPH/Eurographics Conference on Graphics Hardware, San Diego*, 2003:112-119.
- [3] KRUGER J, WESTERMANN R. Linear algebra operators for GPU implementation of numerical algorithms [J]. *ACM Transactions on Graphics*, 2003, 22(3):908-916.
- [4] BOLZ J, FARMER I, GRINSPUN E, et al.. Sparse matrix solvers on the GPU: Conjugate gradients and multigrid [J]. *ACM Transactions on Graphics*, 2003, 22(3):917-924.
- [5] HARRIS M J, BAXTER W V, SCHEUERMANN T, et al.. Simulation of cloud dynamics on graphics hardware [C]. *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware, San Diego*, 2003:92-101.
- [6] CUNTZ N, KOLB A, STRZODKA R, et al.. Particle level set advection for the interactive visualization of unsteady 3D flow [C]. *Proceedings of the Eurographics/IEEE-VGTC Symposium on Visualization, Eindhoven*, 2008:719-726.
- [7] ZHOU K, HOU Q M, WANG R, et al.. Real-time Kd-tree construction on graphics hardware [J]. *ACM Transactions on Graphics*, 2008, 21(5): 231-241.
- [8] 贾平, 张葆, 孙辉. 航空成像移模糊恢复技术[J]. 光学精密工程, 2006, 14(4):697-703.
- JIA P, SUN H, ZHANG B. Restoration of motion-blurred aerial image [J]. *Opt. Precision Eng.*, 2006, 14(4):697-703. (in Chinese)
- [9] BIENOND J, RIESKE J, GERBRANDS J J. A fast Kalman filter for images degraded by both blur and noise [J]. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1983, 31(5): 1248-1256.
- [10] LIKHTEROV B, KOPEIKA N S. Motion-blurred image restoration using modified inverse all-pole filters. [J]. *Journal of Electronic Image*, 2004, 13(2):257-263.
- [11] BANHAM M R, KATSAGGELIS A K. Digital image restoration [J]. *IEEE signal proc. mag.*, 1997, 14(2):24-41.
- [12] OWENS J D, HOUSTON M, LUEBKE D, et al.. GPU computing [C]. *IEEE Proceedings*, 2008:879-899.
- [13] NVIDIA CUDA programming guide. Version 3.0 [M]. Santa Clara: NVIDIA Corporation, 2010.
- [14] OWENS J D, LUEBKE D, GOVINDARAJU N, et al.. A survey of general-purpose computation on graphics hardware [J]. *Computer Graphics forum*, 2007, 26(1):80-113.
- [15] SMITH S W. *The Scientist and Engineer's Guide to Digital Signal Processing* [M]. California: California Technical Publishing, 1997.
- [16] 李仕, 孙辉, 张葆. 运动模糊图像的实时恢复算法 [J]. 光学精密工程, 2007, 15(5):767-772.
- LI SH, SUN H, ZHANG B. Restoration of motion-blurred images [J]. *Opt. Precision Eng.*, 2007, 15(5):767-772. (in Chinese)
- [17] TAN K C, LIM H, TAN B T G. Restoration of real-world motion-blurred images [J]. *CVGIP*,

1991,53:291-299.

[18] TAN K C, LIM H, TAN B T G. Windowing

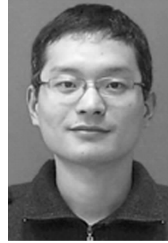
techniques for image restoration [J]. *CVGIP*, 1991,53:491-500.

作者简介:



王 晶(1966—),女,吉林九台人,研究员,硕士生导师,1987年于东南大学获得学士学位,主要从事光电成像与测量技术的研究。Email: wangjingyuan@ sina.com

通讯作者:



李 仕(1984—),男,浙江苍南人,博士,讲师,2010年于中科院长春光机所获博士学位,主要从事数字图像处理与分析、GPU加速计算研究。E-mail: brightlishi@gmail.com

● 下期预告

光谱成像仪 CCD 组件稳态/瞬态热分析与验证

郭 亮^{1,2},吴清文¹,颜昌翔¹,刘 巨¹,陈立恒¹

(1. 中国科学院 长春光学精密机械与物理研究所,吉林 长春 130033;

2. 中国科学院 研究生院,北京 100039)

光谱成像仪 CCD 器件温度过高产生的热噪声和暗电流会导致成像质量下降,采用有限元数值分析方法,建立了 CCD 组件传热的数值模型。根据 CCD 组件的结构特点和导热路径,应用有限元热分析软件 IDEAS-TMG 建立了有限元热分析模型,在给定温度边界条件下对 CCD 组件进行了稳态和瞬态仿真分析。给出了 CCD 组件的热响应性能、组件中关键部件的稳态温度分布云图以及随时间变化的瞬态温度曲线。稳态分析结果表明 CCD 器件工作过程中平均温度水平为 27.1 °C;瞬态分析结果表明 CCD 器件在工作时的升温速率为 2.5 °C/min,最高温度为 37.8 °C。验证试验结果与数值分析结果吻合较好,验证了数值分析的正确性和温度预示的有效性,试验过程中 CCD 器件的温度水平为 26.8 °C,温升速率为 2.4 °C/min。所获得的稳态和瞬态分析结果能够满足热控指标要求,为提高 CCD 组件的可靠性和热设计优化提供了理论依据,对热试验以及在轨工作温度预测有重要的指导意义。