Spacecube：一种快速逆向高光谱地理校正系统

Thomas P. Watson 和 Eddie L. Jacobs，终身高级会员，*IEEE*

arXiv:2601.05181v1 [eess.IV] 8 Jan 2026

摘要—**高光谱相机在采集数据的实用性方面具有诸多优势。它们对每个采样点（像素）采集数百个数据点，而非仅采集 RGB 或多光谱相机系统所能采集的少数几个数据点。航空系统可远程感测此类数据，但在分析之前，必须对数据进行地理校正，以生成一致的图像。我们发现，传统的直接地理校正方法速度较慢，且容易产生伪影。为解决这些缺点，我们提出了 Spacecube，这是一款使用 OpenGL 图形学编程技术实现完整高光谱地理校正流程的程序，其中包括我们自主研发的快速逆向地理校正技术。Spacecube 的运行速度远超实时要求，并能消除像素覆盖伪影。它支持高质量的交互式查看、数据探索及最终产品导出。我们向公众发布 Spacecube 的源代码，供业界使用。**

索引词—**高光谱、推扫式、地理校正、实时、OpenGL、开源**

I. 引言

HYPERSPECTRAL 相机可感测数百个波长波段，而RGB相机只有三个波段，单色成像仪则只有一个宽波段。图像中丰富的波长信息使多种应用成为可能，例如通过植被指数监测农业健康状况 [1]，识别感兴趣的国防目标 [2]，以及检测 [3]土壤或水中的微塑料。

高光谱相机可通过多种技术和方法实现这种感测 [4], [5], [6], [7], [8]，此外，采用特定技术的相机还可通过不同方式进行校准并集成到完整的航空感测系统中 [9], [10], [11], [12]。本文研究的系统包括一台安装在多旋翼小型无人航空系统（sUAS）上的线扫描（即推扫式）相机。

高光谱图像数据集称为数据立方体：单色图像有两个维度（NxM），而高光谱图像增加了第三个光谱维度（NxMxK）。这些维度分别称为 N=个采样点、M=条扫描线和 K=个波段，但排列顺序可能有所不同。顾名思义，数据立方体的光谱维度（K）通常与另外两个维度的大小相当；我们研究的系统采集的数据立方体包含 N=900 个采样点、M=1000 条扫描线，以及 K=300 个波段。

在航空系统中，飞机被预先编程，沿着覆盖感兴趣区域的特定航迹飞行。线扫描相机的安装方向使采样线与飞行航迹垂直，并被设定为按固定的时间间隔采集扫描线（图 1）。仅仅将这些扫描线堆叠起来，几乎不可能生成清晰易懂的二维图像（图 2）；飞机的运动十分明显，会造成显著失真，在多旋翼小型无人航空系统上尤其如此。

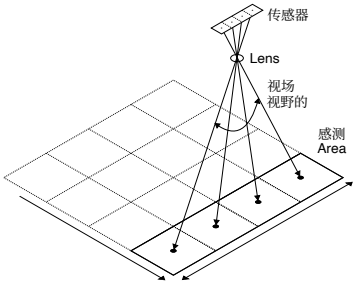

图 1. 航空线扫描系统在地面的一条扫描线上采集四个采样点

因此，我们需要知道相机采集每条扫描线时的精确位置和姿态。惯性导航系统/全球导航卫星系统（INS/GNSS，以下简称 INS）的组合可采集这些信息。这些数据用于计算每条扫描线上各采样点的地理位置，然后在称为地理校正的过程中，将它们相应地放置到图像上 [12]（图 3）。除了实现局部一致性（例如，在图像中将地面上笔直的扫描线显示为笔直），了解图像及其像素的地理位置，还能比较在不同时间、由不同系统拍摄的图像，例如商业卫星影像（图 4）；即使不同飞行之间的具体航迹会发生变化，也不影响比较。

Spacecube 是一款用于整理、校准并对高光谱线扫描相机及相关惯性导航系统数据进行地理校正的程序。我们使用 OpenGL 实现基于网格的逆地理校正技术，可即时预览并快速导出高质量的最终数据立方体。该技术为交互式图形查看器提供预览功能，并支持通过命令行栅格化工具进行导出。Spacecube 的高速处理能力有助于数据调查与探索，而网格渲染则消除了像素覆盖伪影。

II. 背景

我们介绍了高光谱相机的各种实现方式及我们做出选择的原因。我们还介绍了完整高光谱系统的组成部分，包括标定和地理校正。随后，我们简要说明逆向地理校正技术，

T. Watson 和 E. Jacobs 任职于孟菲斯大学 Herff工程学院电气与计算机工程系，地址：3720 Alumni Ave, Memphis, TN,38104 USA；电子邮箱：eljacobs@memphis.edu。



图 2. 地理校正前，线条弯曲（飞行航迹从左向右）

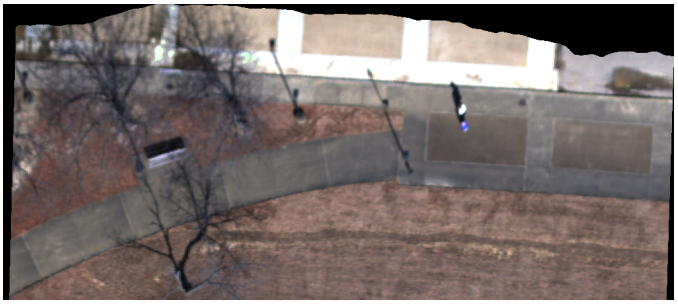

图 3. 使用 Spacecube 进行地理校正后，直线恢复笔直（飞行航迹从左向右）

包括它与其他技术的不同之处。最后，我们简要介绍现代 OpenGL 图形学处理，为第 III 节中对系统更详尽的说明做准备。

A. 高光谱相机

高光谱相机可以采用多种技术，实现对场景中每个像素的多个波长进行感测。这些技术包括微滤光片阵列 [4] 或微透镜阵列 [5], 、可调滤光片 [6], [7], 以及带有衍射光栅的线扫描相机 [8]。文中讨论的所有相机都以普通的二维宽带单色图像传感器为基础，但额外的光学元件会在场景光线到达传感器之前对其进行处理，并且需要通过额外处理和/或合成多幅图像，才能重建完整的数据立方体。

带有微滤光片阵列的相机 [4] 会在传感器前放置许多窄带滤光片，每个滤光片捕获光谱中不同的部分。



图 4. 可比较的卫星影像（影像/地图数据 ©2024 Google）

根据滤光片结构的不同，每个波段成像的区域可能并不相同，因此可能需要移动相机，才能完整成像相机前方的场景。微透镜阵列 [5] 的工作原理类似，每个镜头也都带有滤光片。使用微透镜阵列时，可以同时捕获场景的所有部分，但需要经过复杂的全光函数/光场重建才能生成图像。这两种类型通常每帧包含的信息量最大，以采样点 × 扫描线 × 波段计，但价格昂贵，且需要复杂的后处理。

采用可调滤光片的相机只需改变整个传感器所接收的波长。机械式滤光轮 [6], 可以实现这一点，但更常见的方法是使用可调法布里-珀罗干涉仪 [7]。这需要花费时间扫描所有滤光片配置并采集完整的一组波段。此外，这种方法假设场景不会变化且相机在扫描期间不会移动，因此难以用于航空场景。

在性能与成本权衡方面，最基本的配置可能是线扫描相机，它使用衍射光栅将入射波长分开 [8]。光栅以光谱轴取代了一个空间轴。因此，相机一次只能采集场景中的一条“扫描线”，但能为每个采样点提供完整光谱，生成尺寸为 Nx1xM 的图像，而非原始宽带传感器生成的 NxMx1 图像。

我们重点关注线扫描相机，因为其设计简单，适合构建成本较低的系统，而且易于应用于航空领域，并在其中发挥作用。Spacecube 专为这种类型的高光谱相机而设计。

B. 高光谱校准

所有类型的高光谱相机都需要进行标定，以生成不受特定设置、传感器和场景特性影响的数据。传感器输出的数字量（DN）与照明成正比，但要获得客观的输出并赋予数据科学意义，还必须考虑并消除若干影响因素 [9], [10]。

传感器可以在多种帧率、曝光时间和电子增益值下运行，这些因素都会影响传感器的响应（即单位照明下的 DN值）。这些设置会在飞行前和飞行期间进行调整，以优化捕获范围并避免过曝或欠曝。它们会与数据立方体一同记录，用于确定应使用的校准数据，并正确调整最终输出的比例。

即使没有光线照射，传感器也会生成非零输出值；当传感器被完全遮盖时采集的一条校准暗线会从每条扫描线中减去，以减轻这一影响。随后，使用一条从空间均匀且各波长强度已知的校准源采集的扫描线，将传感器输出值转换为辐射测量量（例如微闪烁）。这些校正消除了传感器波长相关响应的影响（例如由光谱变化的量子效率引起），以及空间非均匀性的影响（例如由镜头渐晕引起）。

Spacecube 可加载并应用校准数据，并允许用户交互式选择要应用的步骤

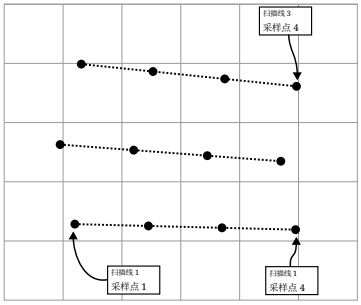

图 5. 将一个包含 4 个采样点、3 条扫描线的数据立方体中的采样点绘制到 6×5 像素的输出图像上

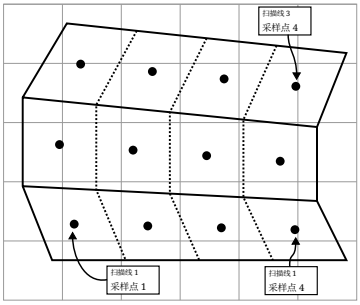

图 6. Spacecube 将每个采样点扩展到采集该采样点时成像的区域，形成覆盖网格

以探索每个步骤的效果。场景照明光谱也可用于将辐射亮度转换为反射率，并消除场景照明变化的影响。该光谱可通过测量已知反射率物体的感知辐射亮度，或借助辅助传感器（例如直接测量照明光谱的光谱辐射计）来获取。

C. 地理校正策略

文献和商业软件中已有许多高光谱地理校正系统，但大多数都采用了简单直接的直接地理校正技术 [11]。该方法独立处理每条扫描线上的每个采样点。首先，使用采样点编号和传感器参数，计算场景中成像后形成特定采样点的射线。然后，根据扫描线的位置和姿态，将这条射线变换到相机在场景中的实际位置（图 1）。射线与地面相交；交点就是该采样点地理校正后的位置，表明其数值来自何处。最后，将相应数值绘制到最终数据立方体中计算出的位置上（图 5）。对每条扫描线上的每个采样点重复此操作，即可构建完整的地理校正后数据立方体。

这种方法在概念上简便易懂，也很容易实现，并且有许多技术以此为基础。这些技术通常会根据交点生成点云 [13], [14], ，或与其他图像数据进行较大区域的特征匹配，以提高整体一致性 [12]，[15]。然而，直接地理校正在速度和覆盖范围方面存在根本性局限，而 Spacecube 解决了这些问题。

对每个采样点执行的计算并不特别复杂（具体取决于交点计算技术），但采样点数量庞大。文献中的研究人员通常报告称，地理校正数据集需要数小时，这还不包括其特定技术所需的额外时间。此外，一些系统会分别对数据集的各个部分进行地理校正，因此还需要额外时间将结果拼接成完整的数据立方体。

直接地理校正还存在覆盖不完整的问题，尤其是在力求生成接近输入数据分辨率的结果时。由于航空器的运动系统中，多个采样点可能落在同一个像素上（造成信息浪费），或者某个像素可能始终没有采样点落在其上（导致输出中出现空缺）。虽然可以通过插值技术来弥补空缺，但这会进一步增加处理时间，而且需要在地理校正后对数据立方体进行处理，从而造成不一致，并引发关于插值像素与“原始”像素之间区别的问题。

Spacecube 通过计算所覆盖区域的网格表示来解决覆盖问题，而不是将采样点视为一组点（图 6）。随后，Spacecube 使用 OpenGL 和现代图形硬件渲染该网格，从而解决速度问题；本质上，这是对直接地理校正的逆向实现。我们称这项技术为“逆向”，因为 OpenGL 会为每个经过地理校正的像素计算对应的采样点，而不是为每个待地理校正的采样点计算对应的像素。只要存储带宽足够，完整数据集就能在几秒钟内完成地理校正，而不必耗费数小时。为进行质量控制或调查分析，也可以即时查看数据子集。

D. 图形学

Spacecube 利用现代图形处理器（GPU）的强大能力，通过 OpenGL 渲染表示地理校正数据的生成网格。在 OpenGL 中，网格由一组顶点和一个连接这些顶点的三角形列表定义。

每次网格渲染操作都会将组成网格的三角形渲染到二维图像缓冲区中。该操作可以包含用户设置的参数和其他作为输入的图像。渲染数据（包括顶点、参数和图像）由着色器处理；着色器是使用类 C 的 GL 着色语言（GLSL）编写的小型程序 [16]。虽然顶点通常描述位置，图像通常包含 RGB 颜色，但其实际数据及其含义由用户定义，并且具有可扩展性。

对于每个顶点，顶点着色器根据其数据计算其在图像缓冲区中的位置。对于每个三角形，片段（即像素）着色器随后使用从其三个顶点的数据（以及图像数据）中线性插值得到的数据，计算该三角形覆盖的每个像素的值；这些值会存储在图像缓冲区中对应的像素位置。由于着色器是处理用户定义数据的完整程序，因此几乎可以实现任何计算。

Spacecube包含一组顶点/参数/图像数据描述和着色器程序，用于实现其渲染，详见第III-D节。许多机器学习和物理仿真等应用也会使用图形处理器，但采用不同的

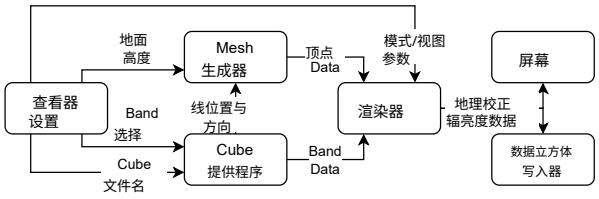

图 7. Spacecube 组件概览

编程 API，例如 OpenCL 和 CUDA。由于 Spacecube 是一款需要进行光栅化和实时渲染的交互式图形应用，我们认为使用 OpenGL 很合适，其功能也足以很好地完成我们的任务。

III. 实现

图 7 展示了 Spacecube 的各个组件，它们共同实现了一个完整的地理校正解决方案，采用的是我们的逆向地理校正技术。Spacecube 使用 Python 3.10 实现，主要借助 ModernGL 5.8 [17]调用 OpenGL 函数。Spacecube 的查看器和命令行栅格化器是两个独立的程序，不过大多数功能实际上由二者共用的库实现，也可用于其他应用程序。我们主要介绍查看器，同时也会指出栅格化器的不同之处。

Spacecube 可配置为支持任何推扫式高光谱系统。它以业界标准的 ENVI（ENvironment for Visualizing Images，图像可视化环境）格式读取和写入数据立方体。Spacecube 按波段顺序运行，因此每次只需将少数几个波段加载到内存中。Spacecube 还以数据立方体为单位运行；我们的机载系统默认将 1000 条扫描线累积到一个数据立方体中，并将每个数据立方体视为独立对象，尽管它们属于同一次数据采集。这有助于并行加载和处理数据，减少一次需要管理的数据量，从而提升性能并简化实现。

A. 查看器操作

打开查看器时，用户会提供要查看的数据立方体列表。随后，Spacecube 会加载相关元数据、处理地理校正信息、自动估算地面高度，然后加载并在屏幕的红、绿、蓝通道上显示三个选定波段。即使加载尚未完成，用户也可以使用鼠标或触摸屏输入缩放和平移场景。图 8 显示了查看器及其控件；为便于说明，部分数据立方体仅加载了一部分。

用户可以设置处理模式，以显示原始数字量化值数据、相对缩放数据（即缩放至一致的曝光），或辐射率校准数据（如果已提供合适的校准文件）。每种处理模式都可以不缩放显示（最适用于原始数据），也可以进行缩放（称为“拉伸”[18]），对所有显示通道使用同一个值，或为每个通道分别使用独立的值。缩放值根据直方图计算，使得像素值不高于 2% 时显示为黑色，不低于 98% 时显示为白色。

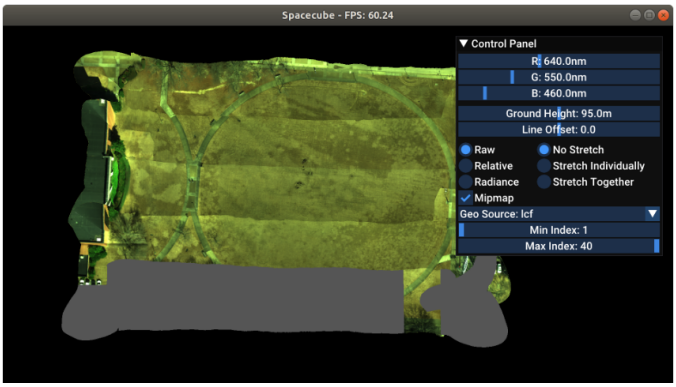

图 8. Spacecube 查看器及其控件的屏幕截图。尚未加载数据的数据立方体显示为灰色

在原始模式下，此直方图根据原始数据计算；否则，则根据屏幕内容动态计算。

用户可以选择要在每个 RGB 通道中显示的波长。Spacecube 会查找波长最接近所请求波长的波段编号，然后加载该波段的数据。用户可以选择要在屏幕上显示哪些数据立方体，这有助于裁去飞行的起始和结束部分。地面高度和地理校正数据源可以交互式更改。用户还可以选择启用多级渐远纹理 [19]，，以（略微）提升性能并减少混叠，从而平滑图像，但这会在数据立方体之间产生细小伪影。

查看器旨在保持响应迅速，以支持交互式使用。缩放和平移、直方图、处理模式以及所显示数据立方体的范围均以每秒 60 帧的速度更新。其他操作则在后台线程中处理，并在结果可用后立即显示给用户。如果用户更改参数，系统会尽力取消现有请求，以便尽快显示最新数据。查看器会在选定波段和地理校正数据加载时，将其传输至图形处理器，同时限制传输速率以保持交互性。

命令行光栅化器始终生成未缩放的辐亮度数据，从不使用多级渐远纹理，并将结果存储到按波段顺序排列的数据立方体文件中，而不是将其显示在屏幕上。它还可以处理超过三个波长；所有指定的波长都会依次进行处理。

B. 提供程序

Spacecube 通过数据立方体提供程序库抽象化数据立方体的检索和处理过程，该库之所以如此命名，是因为它提供 Spacecube 所处理的数据。这样，未来便可以针对不同的数据格式和不同类型的实验使用不同的数据源。Spacecube 将数据立方体的文件名（或其他来源）提供给提供程序，然后由提供程序负责查找数据立方体，并将数据转换为 Spacecube 可使用的通用格式。

提供程序最初只加载数据立方体元数据（例如维度和相机参数），随后提供一个句柄，供 Spacecube 在必要时检索特定的数据立方体波段或一组几何校正数据。

与 Spacecube 一样，该提供程序专为按波段顺序排列的数据立方体而设计；请求一次处理一个波段。也支持其他排列方式，但由于磁盘访问模式不理想，性能可能较差（读取特定波段时，访问文件开头 1/300 的字节，速度远快于每隔 300 个字节访问一次）。程序还提供了将所有数据预加载到内存中的功能，以便在内存容量足够的计算机上缓解这些问题。

提供程序还负责对地理校正数据进行预处理。数据从 INS 文件中读取，并首先整理为拍摄时间戳、相机地理位置（纬度、经度、海拔）和相机姿态（滚转、俯仰、偏航）。

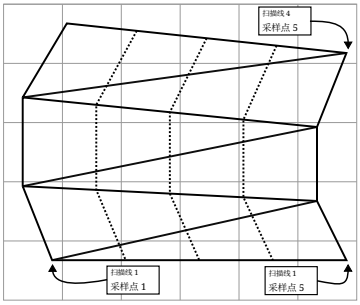

图 9. 以三角网格表示的样本覆盖区域，顶点位于线段端点处（虚线采样分隔线仅用于说明）

纬度和经度不适合直接用于运算，因为它们表示的是球坐标系，而我们希望渲染的是名义上的平面地图。因此，需要将其投影到通用横轴墨卡托 [20] (UTM) 坐标系中。加载数据时会自动选择合适的 UTM 分区和网格航向偏移量；我们假设小型无人机系统的飞行区域不会大到因分区或偏移量不理想而产生明显失真。投影后，坐标会以米为单位，存储在内部的北-东-下（NED）坐标系中。

姿态数据会经过变换，使单位旋转表示相机朝向天底，且拍摄扫描线的顶部在 NED 系统中指向北方。随后应用计算得出的 UTM 航向偏移量，并将数据作为 SciPy 1.11 旋转对象（包含多个旋转）存储在内部。

随后，使用曝光和惯性导航系统数据的时间戳，对位置数据进行线性插值，对姿态数据进行球面线性插值，从而为数据立方体的每条扫描线指定其曝光开始时刻的精确位置和姿态。然后，将这些数据以米为单位的 NED 坐标数组和 Rotation 对象的形式提供给 Spacecube，其中每个数组和对象都包含数据立方体中每条扫描线对应的一项。此预处理确保 Spacecube 在空间和时间上都能采用一致且已知的坐标系运行，不受数据立方体或地理校正数据来源的影响。

C. 网格生成器

在渲染每个数据立方体之前，Spacecube 会生成一个网格，用于表示相机采集该数据立方体时成像的地面区域。网格不包含实际的数据立方体数据，并且独立于数据立方体生成。为生成该网格，

我们假设相机在一个已知海拔（即地面高度）的平坦表面上方飞行；对于小型无人机系统扫描的农田等区域而言，这是一个合理的假设。我们还假设相机持续曝光，因此扫描线之间没有间隙；这是大多数相机都支持的一种操作模式。

请注意，我们的网格化过程将某个采样点视为覆盖一片特定区域，而不是像直接地理校正方法那样将其放置在某个特定点上。我们认为这种做法更准确，因为（我们假设）相机在扫描线采集期间持续曝光，因此在扫描线之间移动时会采集整个区域的信息。该扫描线上某个采样点覆盖的区域也并非无限小，而是由探测器的物理尺寸和光学元件决定；同样，我们假设区域之间既没有间隙，也没有重叠。不过，我们仍会像直接地理校正一样，计算每条扫描线在地面上的两个端点坐标，以确定数据立方体所覆盖区域的边界。

从相机处于单位姿态开始，使用针孔相机模型从原点投影（或反投影）出两个向量 [21], , 分别来自相机视场的两个极端，使它们的端点落在下方 1 个单位的表面上。已知相机视场角 θ ，并运用简单的三角学，可得每个端点与一个长度为 1、垂直向下的向量之间的水平距离为 $d = 1 * \tan^{-1}(\frac{\theta}{2})$。随后，两个向量端点的坐标为 $(N, E, D) = (0, \pm d, 1)$。

对于数据立方体中的每条扫描线，都会获取其曝光开始时的姿态和位置。将两个向量旋转至该姿态并平移至该位置，使其端点位于拍摄该扫描线时投影到相机传感器上的图像的 3D 端点处。对于透视校正插值 [22], , 我们根据每个端点到相机和地面的垂直距离，计算该端点的有效深度。最后，沿着两个向量各自的方向延伸，使其与地面相交；这两个二维交点即为扫描线的端点。

为覆盖成像区域，将一条扫描线的两个端点彼此相连，并用一个四边形将它们与下一条扫描线的两个端点相连；该四边形的顶点就是这四个点（自然地表示为两个三角形）。依次对数据立方体中的所有扫描线重复此过程，相机成像的整个区域便由一系列三角形覆盖，这些三角形共同构成一个无缝网格（图 9）。随后为顶点标记其深度，并将其上传至图形处理器进行渲染。

为避免数据立方体后出现间隙并涵盖其完整区域，采集序列中下一个数据立方体的第一条扫描线会被视为当前数据立方体最后一条扫描线的“下一条”扫描线。因此，尽管一次完整的数据采集由多个分别生成网格的数据立方体组成，渲染时仍会将其视为一个无缝网格。在数据采集结束时，由于不存在下一个数据立方体，最后一条扫描线被假定为零面积，因此不会被渲染。

D. 渲染器

渲染器一次处理三个波段（分别映射到红、绿、蓝通道），并渲染到一个缓冲区上。缓冲区的像素尺寸由查看器窗口大小或栅格化地面采样距离（地面采样距离） [23] 决定。缓冲区的渲染区域以中心点（东北坐标）和各维度的比例尺（米）来描述。查看器会根据用户的平移和缩放命令调整中心点和比例尺；栅格化器则根据给定的数据立方体边界计算这些参数。

每个数据立方体网格都通过一次操作完成渲染。顶点数据来自扫描线端点位置和深度。参数用于描述视区和校准模式。系统会设置图像，其中包含波段数据、校准数据，以及根据相机设置计算出的逐行相机响应。

顶点着色器会根据视图参数将顶点数据转换为屏幕坐标，从而无需更改内存中的数据，提升性能。顶点着色器还会为每个顶点添加其扫描线编号和采样点编号，这些编号根据已知的网格结构计算得出。

OpenGL 会计算网格覆盖的像素，然后对顶点数据进行插值，以计算这些像素中心处的数据立方体位置（图 10）。数据立方体位置采用透视校正插值，该方法假定连接某条扫描线及其下一条扫描线的四边形在三维空间中是平坦的，因此会将分隔采样点的扫描线渲染为直线。然而，如果两条扫描线之间存在滚转差异，四边形就无法保持平坦，并会沿着连接四边形两个三角形的边缘出现折痕。实际上，这种效应导致的位置误差平均不到一个采样点大小的 1%。在此处进行插值，而不是在地理校正后进行插值，可确保每个像素值都确实存在且几何位置准确，不会缺失，也不会由“附近”的像素填充。

对于每个被覆盖的像素，片段着色器使用计算得到的扫描线号和采样点号，从输入图像中查找相应的波段数据值、校准系数和相机响应。如第 II-B 节所述，根据该公式应用这些值以生成辐射亮度值

$$(\mathrm{cube}_{band,line,sample} - \mathrm{calib}^{\mathrm{dark}}_{band,sample}) * \frac{\mathrm{calib}^{\mathrm{rad}}_{band,sample}}{\mathrm{response}_{line}}. \quad (1)$$

输入会根据模式而变化，例如，禁用标定会固定 $\mathrm{calib}^{\mathrm{dark}} = 0$ 和 $\mathrm{calib}^{\mathrm{rad}} = 1$。

随后，该值会按显示范围进行缩放，并存储到该像素位置处的缓冲区，最终显示在屏幕上或写入数据立方体。此过程分别针对三个选定波段独立执行。

IV. 结果

我们介绍所使用的航空线扫描高光谱系统的参数，以及利用该系统采集的数据集。随后，我们使用 Spacecube（它采用我们的逆向地理校正技术）和采用直接地理校正的其他软件处理该数据集，以便比较处理性能。最后，我们比较各程序的处理结果质量。

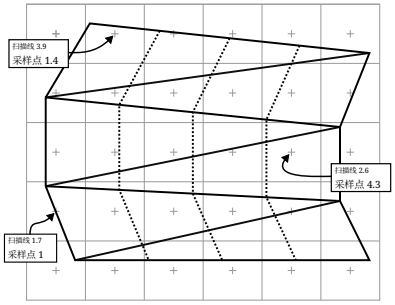

图 10. OpenGL 对顶点数据进行插值，以提供每个覆盖像素的立方体位置（虚线采样点分隔线仅用于示意）

A. 数据采集

采集时，我们使用了 Resonon Pika L [24] 机载高光谱系统，并对其进行了改装和调整，使其适配我们的小型无人机系统并提高可靠性。该系统包括一台 SBG Systems Ellipse N [25] 惯性导航系统，以及一个 Emlid Reach RS2+ [26] GNSS基准站，用于提供实时动态（RTK）校正数据。Pika L相机以每秒 249 帧的速率生成扫描线，每条扫描线包含 900 个采样点和 300 个波段；曝光时间为 3.9ms（占帧间隔时间的 97%）。惯性导航系统以 200Hz 的速率生成位置和姿态数据。该系统搭载在一架定制的 1000mm 级六旋翼小型无人机系统上运行，使用 ArduPilot [27]。

为生成可比较的结果，我们主要使用了 Resonon 的高光谱分析程序 Spectrononv3.5.5。该程序专为我们使用的 Pika L相机设计，并随相机提供。Spectronon 具有用于执行辐射定标、使用直接地理校正方法进行地理校正，以及将数据镶嵌为最终数据立方体的插件。我们还使用了行业标准工具 PARGE [28], [29] v4.1b7；它与 Pika L 部分兼容，也使用直接地理校正方法。

使用我们的采集系统，我们于2024年2月23日上午11:00在孟菲斯大学采集了一个数据集。飞行区域约为175米×125米（2.2公顷），共进行了5次飞行，名义海拔为40米（地面高度为95米），飞行速度为10米/秒，使用视场角为47.5度的镜头。根据这些飞行参数，两个轴向上的名义地面采样距离（GSD）约为4厘米。因此，我们将所有处理的地面采样距离设为4厘米，地面高度设为95米。

使用的数据总计包含40个数据立方体和40,000行数据，采集时间为2分40秒。为避免干扰和伪影，起飞、飞行中校准以及返航/着陆期间采集的其他数据未进行处理。

Space- cube 处理后的完整地理校正结果如图 11 所示。由于镜头渐晕以及平坦表面上不完全平整所产生的透视效应，立方体行重叠处存在一些伪影。此外，还存在一些小的绿色斑点，尤其是在左下角，这是由于

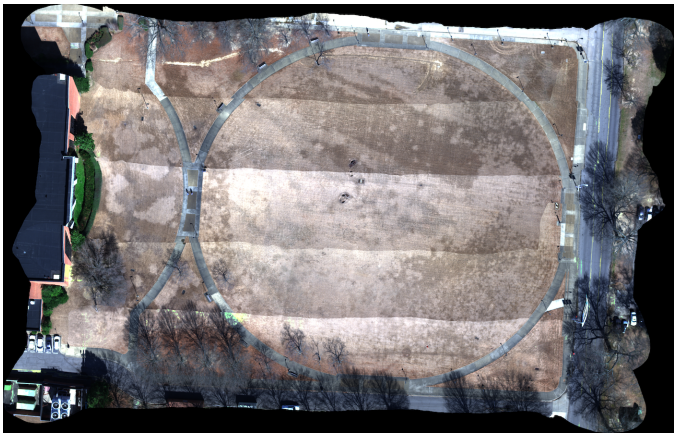

图 11. 地理校正结果

过度曝光造成的。按照设计，该结果与 Spectronon 生成的结果基本无法区分，因此我们不再占用篇幅展示 Spectronon 的输出。不过，我们将重点观察该结果的某些部分，以突出我们方法的优势。

B. 性能比较

我们在一台 ThinkPad P52 笔记本电脑上测量了各程序每个步骤的处理时间。该电脑配备 Intel Xeon E-2176M CPU（6 核，2.7GHz，64GB 内存）和 Nvidia Quadro P2000 GPU（4GB 显存）。为避免磁盘访问模式和缓存影响结果，我们将所有测试软件所用的数据文件都保存在内存盘中。结果汇总于表 I。

Spacecube 的栅格化器命令行程序在 11 秒内生成了完整的 300 波段数据立方体，其中包括应用辐亮度校准、地理校正以及镶嵌，最终生成数据立方体。数据采集历时 2 分钟 40 秒，因此其运行速度达到实时运行速度的 1,400%。

Spectronon 对 40 个数据立方体逐一进行辐亮度校准和地理校正，共耗时 7 分 16 秒，此外镶嵌还需 14 分 13 秒。Spacecube 的速度超过 Spectronon 的 11,000%，并且能够一次完成所有步骤，无需任何中间文件。

PARGE 不支持对 Pika L 进行标定，因此重复使用了 Spectronon 的标定结果。PARGE 用了 3 分 11 秒对所有数据立方体进行地理校正（采用其快速平坦地面假设）；该时间是通过处理部分数据立方体后，再按数据立方体总数折算得出的，因为 PARGE 的批处理不支持 Pika L。随后，PARGE 花了近 30 分钟进行镶嵌（同样重复使用 Spectronon 的地理校正结果，并进行了格式转换）。在最快模式下，Spacecube 的速度是 PARGE 的 19,000% 以上。

C. 质量比较

使用 Spectronon 的默认插值设置（形态学插值，像素半径为 1），我们对数据集数据立方体 26（如引言中所示）进行地理校正，并在图 12 中展示结果。

表 I 处理时间（分和秒）

处理步骤	Spectronon	PARGE	Spacecube
辐射校正	4分15秒		0分11秒
地理校正	3分1秒	3分11秒（预计）	
镶嵌	14分13秒	28分22秒	
总计	21分29秒	35 分 48 秒	0 分 11 秒



图 12. Spectronon 的地理校正能够生成视觉上无法区分的结果

该数据立方体在视觉上与图 3 中由 Spacecube 生成的数据立方体无法区分。然而，关闭插值（图 13）会产生空隙，形成未覆盖像素的条纹图案。在这两种情况下，PARGE 生成的结果在视觉上都与 Spectronon 的结果无法区分，因为它同样使用直接地理校正。

这些地理校正中的空隙是未被覆盖的像素，因为小型无人机系统的姿态略有变化，采样点恰好没有落在这些位置上。空隙是直接地理校正方法固有的，在输出地面采样距离与名义输入地面采样距离相近时尤其容易出现。随后，插值只需用附近像素的值替换每个空隙，即可将其填补。对于我们的 4 厘米地面采样距离数据集，Spectronon 建议采用 7 厘米的地面采样距离，这确实能减少空隙。不过，像素边长增加 75% 自然会扩大其面积，导致分辨率降低 300%。

虽然插值确实能弥补空缺，但它也有不足之处，尤其是在运动幅度较大和重叠区域。图 14 展示了由 Spectronon 处理的数据立方体 23：在下半部分，两种姿态重叠处出现了径向条纹，这是因为小型无人机系统转弯时发生了转向。插值器不会

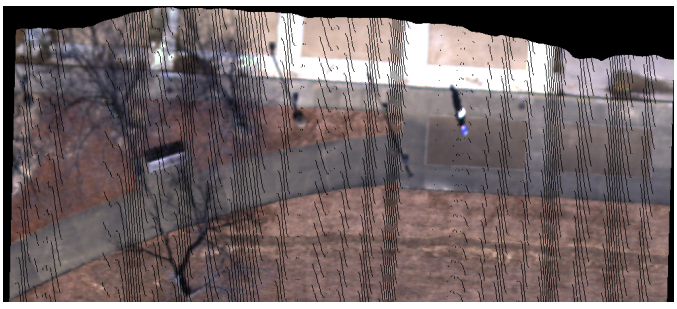

图 13. 禁用 Spectronon 的插值后，空隙显现，导致出现条纹状覆盖伪影

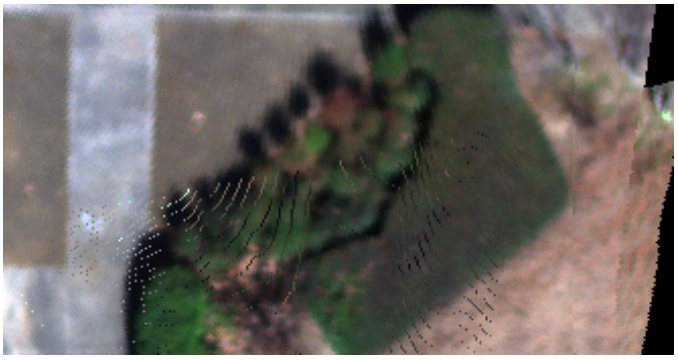

图 14. 尽管 Spectronon 进行了插值，仍可见错误像素的图案

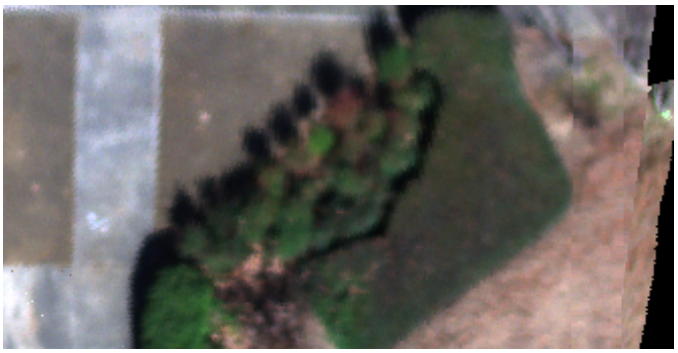

图 15. PARGE 的空缺和错误像素较少，但左下方仍有一些

将这些区域视为空缺，因为有一个采样点落在了上面。由于插值器是在地理校正后运行的，因此无法从整幅图像的角度识别出该采样点不一致。令人意外的是，PARGE 在重叠区域产生的错误像素较少，不过左下方仍有一些（图 15 ）。

Spacecube 的渲染方法（图 16）从根本上避免了产生空隙，因此无需进行插值处理，也完全避免了任何覆盖伪影。生成的网格完整覆盖了所有采集到的采样点；即使某个像素上没有任何特定采样点，OpenGL 也能确保网格覆盖的所有像素都以几何上准确的方式填充。

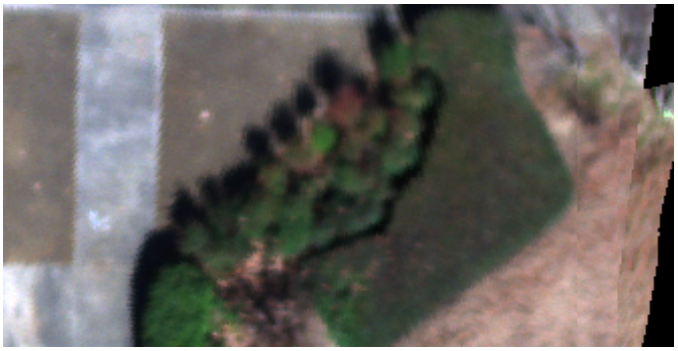

图 16. Spacecube 的网格渲染可自然避免覆盖伪影

V. 结论

与典型的直接地理校正方法相比，Spacecube 的逆向地理校正技术可将数据处理速度提升 11,000% 以上，并且不会受到像素覆盖空隙造成的伪影影响。交互式查看器结合快速平移和缩放，以及实时调整地理校正数据、辐亮度处理模式等设置，让探索性数据分析变得轻松。命令行工具可快速生成符合行业标准 ENVI 格式的数据立方体，且质量与查看器相同。即使处理数据集中的所有数据立方体和波段，Spacecube 的运行速度仍比实时处理快 1,400%，从而促成了新的应用场景。

Spacecube 的主要局限在于假设地面是平坦的。对于我们使用小型无人机系统开展的作物调查而言，这一假设基本合理；但由于地面并非完全平坦，再加上透视效应，航带重叠处会出现伪影。Spectronon 和 PARGE 等直接地理校正软件可以轻松使用数字高程图（DEM）来补偿这些影响。Spacecube 则无法做到，而要扩展算法以支持这一功能并非易事。尽管如此，这仍是我们未来的一项重要工作。Spacecube 目前也尚未实现对重叠采样点取平均值的功能，不过 OpenGL 可通过混合模式支持这一功能，我们计划也将其加入。

附录

披露声明

作者声明，他们不存在任何可能影响本研究客观性或本文撰写的经济利益、商业关系或其他潜在利益冲突。

代码、数据和材料可用性

本文所展示功能对应的 Spacecube 版本源代码将在经过同行评审的论文发表后，以 GPLv3（或更高版本）许可证发布于 https://github.com/JacobsSensorLab/spacecube-paper-release。Spectronon 可从 Resonon 网站免费下载，但地理校正插件不免费。联系开发者 ReSe Applications GmbH 可获取 PARGE 免费试用版。本研究使用的数据集可在 Zenodo 获取，采用 CC-BY-4.0 许可证，网址为 https://zenodo.org/rec ords/14814175。

致谢

我们特别感谢协助设计和操作用于采集数据的无人机和机载系统的实验室成员，包括 Angelin Favorito、Kevin McKenzie、Brianna Miller、Joe Perry、MD Rafi Ur Rahman、Shoaf Robinson、Noah Wargo 和 Ryan Williamson。若没有他们的帮助，我们就没有任何东西可以进行地理校正。

我们还要感谢 Peter Le、Patrick Leslie 和 Kevin McKenzie 对早期草稿的审阅，以及他们提供的宝贵写作意见和鼓励。

本研究由美国陆军研究实验室资助，并根据合作协议编号 W911NF-21-2-0294 完成。本文所述观点和结论仅代表作者观点，不应被理解为代表美国陆军研究办公室或美国政府明示或暗示的官方政策。尽管本文载有任何版权声明，美国政府仍有权为政府用途复制和分发再版。

参考文献

[1]P. S. Thenkabail、R. B. Smith 和 E. De Pauw，《高光谱植被指数及其与农作物特征的关系》，《环境遥感》，第 71 卷，第 2 期，第 158–182 页，2000年 2 月。[在线]。网址：https://linkinghub.elsevier.com/retrieve/pii/S0034425799000067X [2] Y. Chen Sr.、J. Zhou、X. Chen、Y. Ji 和 W. Shen，《通过高光谱成像和光谱广义角分析进行目标鉴别》，载于第十届国际数字图像处理会议（ICDIP 2018），X. Jiang 和 J.-N. Hwang 编，第 10806 卷。国际光学与光子学学会，2018年，第 105–110 页，备用出版方：国际光学与光子学学会。[在线]。网址：https://doi.org/10.1117/12.2502957 [3] A. Faltynkova、G. Johnsen 和 M. Wagner，《高光谱成像作为分析微塑料的新兴工具：系统综述及未来发展建议》，《微塑料与纳米塑料》，第 1 卷，第 1 期，第 13 页，2021年 12 月。[在线]。网址：https://microplastics.springeropen.com/articles/10.1186/s43591-021-00014-y [4] T. Guo、Z. Lin、Z. Zhang、X. Chen、Y. Zhang、Z. Hu、R. Zhang 和 S. He，《采用可重构滤波器阵列、兼具高空间分辨率和高光谱分辨率的微型高光谱成像仪》，《纳米快报》，第 24 卷，第 36 期，第 11 156–11 162 页，2024年 9 月。[在线]。网址：https://pubs.acs.org/doi/10.1021/acs.nanolett.4c01075 [5] “Cubert Ultris X”。在线。网址：https://cubert-hyperspectral.com/en/ultris-x20/ [6] “CI Systems SR-7000”，2024年。[在线]。网址：https://www.polytec.com/fileadmin/website/optical-systems/spezialkameras/pdf/PHWLCISR-7000 Hyperpectral Camera.pdf [7] “HinaLea VNIR 4200C”，2022年。[在线]。网址：https://sphereoptics.de/en/wp-content/uploads/sites/3/2023/02/HinaLea-4200C-VNIR-Hyperspectral-Camera-Specs-2022.pdf [8] C. Yang、J. H. Everitt、M. R. Davis 和 C. Mao，《基于电荷耦合器件相机的定点及机载高光谱成像系统》，《国际地理制图》，第 18 卷，第 2 期，第 71–80 页，2003年 6 月。[在线]。网址：http://www.tandfonline.com/doi/abs/10.1080/1010604030852742 [9] K. C. Lawrence、B. Park、W. R. Windham 和 C. Mao，《用于农业检测的推扫式高光谱成像系统标定》，《美国农业工程师学会汇刊》，第 46 卷，第 2 期，2003年。[在线]。网址：http://elibrary.asabe.org/abstract.asp??JID=3&AID=12940&CID=t2003&v=46&i=2&T=1 [10]P. Geladi、J. Burger 和 T. Lestander，《高光谱成像：标定问题及解决方案》，《化学计量学与智能实验室系统》，第 72 卷，第 2 期，第 209–217 页，2004年 7 月。[在线]。网址：https://linkinghub.elsevier.com/retrieve/pii/S0169743904000371 [11]M. A. Warren、B. H. Taylor、M. G. Grant 和 J. D. Shutler，《使用机载处理库（APL）处理机载遥感高光谱数据：地理校正算法说明及空间精度评估》，《计算机与地球科学》，第 64 卷，第 24–34 页，2014年 3 月。[在线]。网址：https://linkinghub.elsevier.com/retrieve/pii/S0098300413002938 [12] Y. Angel、D. Turner、S. Parkes、Y. Malbeteau、A. Lucieer 和 M. F. McCabe，《基于推扫式传感器的无人机高光谱影像自动地理校正与镶嵌》，Remote Sensing，第 12 卷，第 1 期，第 34 页，2019年 12 月。[在线]。网址：https://www.mdpi.com/2072-4292/12/1/34 [13] D. Inamdar、M. Kalacska、G. Leblanc 和 J. P. Arroyo-Mora，《直接地理配准高光谱点云的实现》，《MethodsX》，第 8 卷，第 101429 页，2021年。[在线]。网址：https://linkinghub.elsevier.com/retrieve/pii/S2215016121002223 [14]A. Finn、S. Peters、P. Kumar 和 J. O’Hehir，《使用线扫描成像传感器观测的无人机高光谱影像，自动进行地理校正、镶嵌和三维点云生成》，Remote Sensing，第 15 卷，第 18 期，第 4624 页，2023年 9 月。[在线]。网址：https://www.mdpi.com/2072-4292/15/18/4624 [15] L. Yi、J. M. Chen、G. Zhang、X. Xu、X. Ming 和 W. Guo，《用于环境监测的无人机推扫式高光谱图像无缝镶嵌》，Remote Sensing，第 13 卷，第 22 期，第 4720 页，2021年 11 月。[在线]。网址：https://www.mdpi.com/2072-4292/13/22/4720 [16] R. J. Rost、B. Licea-Kane 和 D. Ginsburg，《OpenGL 着色语言》，第 3 版。Upper Saddle River，NJ Munich：Addison-Wesley，2012年。[17] S. Dombi，《ModernGL：高性能 OpenGL Python 绑定 3.3+》，2020年 5 月，出版方：GitHub。[在线]。网址：https://github.com/moderngl/moderngl [18] R. Driggers、E. Jacobs 和 S. Young，《成像系统的信号处理与性能分析》。Artech，2008年。[19] L. Williams，《金字塔参数化》，载于《第十届计算机图形学与交互技术年度会议论文集》，1983年，第 1–11 页。[20] J. P. Snyder，《地图投影：实用手册》，美国地质调查局，华盛顿哥伦比亚特区，报告 1395，1987年。[在线]。网址：https://pubs.usgs.gov/pub/1395 [21] R. Szeliski，《计算机视觉：算法与应用》，第 2 版，计算机科学读本系列。Cham：Springer，2022年。[22] P. S. Heckbert 和 H. P. Moreton，《多边形纹理映射与着色的插值》，《计算机图形学前沿：可视化与建模》，第 101–111 页，1991年，出版方：Springer Science & Business Media。[23] R. G. Driggers，《光学工程百科全书》。纽约：Dekker，2003年，第 2 卷。[24] “Resonon Pika L”，2024年。[在线]。网址：https://resonon.com/content/files/Resonon---Camera-Data-Sheets-Pika-L.pdf [25] “SBG Systems Ellipse 系列”，2023年。[在线]。网址：https://www.sbg-systems.com/wp-content/uploads/Ellipse Series Leaflet.pdf [26] “Emlid Reach RS2+”，2024年。[在线]。网址：https://files.emlid.com/docs/Datasheet-RS2%2B-ENG-web.pdf [27] A. Tridgell、R. Mackay 等，《ArduPilot》，2024年。[在线]。网址：https://ardupilot.org/ [28] D. Schlapfer、M. Schaepman 和 K. Itten，《PARGE：基于地面控制点校准的辅助数据的参数化地理编码》，载于 SPIE 成像光谱学 IV，1998年 1 月，第 334–344 页。[29] “PARGE”，2024年 11 月。[在线]。网址：https://www.rese-apps.com/software/parge/index.html



Thomas P. Watson 分别于2019年和2021年在孟菲斯大学获得电气工程学士+CE和电气工程硕士学位，目前正在攻读博士学位。他喜欢编程、嵌入式系统和系统集成，尤其喜欢以新的方式组合旧事物。他的研究兴趣包括飞行系统和遥感，尤其关注融入实时处理。



Eddie L. Jacobs 分别于1986年和1988年获得阿肯色大学电气工程学士和电气工程硕士学位。2001年，他获得乔治·华盛顿大学电物理学理学博士学位。1989年至2006年，他受雇于美国陆军夜视与电子传感器局（弗吉尼亚州贝尔沃堡），负责电光建模与仿真成像传感器。自2006年以来，他一直担任孟菲斯大学电气与计算机工程教授。他的研究兴趣包括电光学、成像传感器和遥感。