

引用格式: SUN Zhelun, XIAO Biyong, ZHAO Zulaing, *et al.* Loading and service implementation of GF-1 satellite data based on Open Data Cube[J]. Remote Sensing Technology and Application, 2023, 38(4): 794-802. [孙哲伦, 肖弼雍, 赵祖亮, 等. 基于开放数据立方体的GF-1卫星数据加载与服务[J]. 遥感技术与应用, 2023, 38(4): 794-802.]

DOI: 10.11873/j.issn.1004-0323.2023.4.0794

# 基于开放数据立方体的GF-1卫星数据加载与服务

孙哲伦<sup>1</sup>, 肖弼雍<sup>1</sup>, 赵祖亮<sup>2</sup>, 俞国江<sup>1</sup>, 曹倩倩<sup>3</sup>, 王萌<sup>3</sup>, 姚晓闯<sup>1</sup>

(1. 中国农业大学土地科学与技术学院, 北京 100193;

2. 浙江省发展规划研究院, 浙江 杭州 310030;

3. 中国科学院空天信息创新研究院, 北京 100094)

**摘要:** 开放数据立方体系统是新一代开源对地观测数据管理与应用系统, 可以解决遥感卫星大数据在管理、服务及应用方面面临的诸多问题, 然而, 该系统目前仅支持国际主流卫星数据的加载和服务, 针对国产数据仍无法使用。以国产卫星数据高分一号WFV数据为例, 进行基于开放数据立方体系统的国产卫星数据的加载和服务的设计与实现。首先针对GF-1的数据特点编写产品定义和提取元数据脚本, 以建立索引和摄取数据, 完成国产卫星数据的加载; 然后基于已加载数据, 使用Web UI接口开发了GF-1数据的水探测服务。研究结果证明了使用开放数据立方体系统管理国产卫星数据的可行性, 并为其他基于该系统的国产卫星数据的加载和服务提供了技术参考。

**关键词:** GF-1卫星数据; 数据管理; 开放数据立方体

**中图分类号:** TP75 **文献标志码:** A **文章编号:** 1004-0323(2023)04-0794-09

## 1 引言

自1972年美国国家航空航天局发射陆地一号卫星(Landsat 1)以来, 人类利用卫星进行对地观测活动已有近50 a历史。在此其间, 对地观测卫星技术随着第三次科技革命飞速发展, 尤其是进入21世纪以来, 对地观测卫星的数量和观测性能都有着突飞猛进的增长。更高的时间分辨率、空间分辨率和光谱分辨率, 使得新一代地球观测卫星创造的数据量越来越大, 数据量从此不再是限制遥感应用的主要制约, 遥感大数据时代的大门就此开启<sup>[1]</sup>。

在遥感大数据时代, 如何加快数据准备和访问速度、丰富数据分析手段和提高数据分析效率成为了用户面临的主要问题。此外, 尽可能降低获取和分析卫星数据所需的技术门槛, 也成为了不同专业背景的遥感数据使用者的一致诉求。为了满足上

述的需求, 完成向遥感大数据时代的过渡, 更先进的算法、工具和数据管理解决方案需要被开发和应用<sup>[2]</sup>。新一代的地球观测数据管理方案大多结合云计算等技术, 摆脱了传统的本地处理和数据分发方法, 在降低数据存储和处理所需的工作成本的同时, 使得科研工作者的使用不再受限于计算机硬件和网络环境<sup>[3]</sup>。目前最主流的对地观测数据云平台当属由谷歌公司开发的谷歌地球引擎(Google Earth Engine, GEE)。GEE平台包含有PB级的数据, 不仅涵盖了Landsat系列, MODIS系列, Sentinel系列卫星的各级产品数据, 还涵盖了各类社会经济数据。同时, GEE可以通过Web API和IDE为用户提供较为强大的云计算服务, 使得用户可以借助云端强大的算力实现遥感数据的调用、分析、输出, 这一方面使得科研人员节约了大量处理数据的时间, 另一方面大大降低了遥感数据分析的硬件和技术

收稿日期: 2022-05-12; 修订日期: 2023-07-11

基金项目: 国家重点研发计划项目(2019YFE0127000、2021YFE0102300)。

作者简介: 孙哲伦(2000—), 男, 黑龙江哈尔滨人, 从事遥感卫星数据组织与管理的研究。E-mail: sunzhelun@cau.edu.cn

通讯作者: 姚晓闯(1986—), 男, 河南登封人, 副教授, 主要从事时空大数据技术及应用研究。E-mail: yxc@cau.edu.cn

门槛,拓宽了对地观测数据的应用领域和场景<sup>[4]</sup>。这也是谷歌地球引擎自发布以来被全球范围内的研究人员广泛应用的原因。

然而,Google Earth Engine由谷歌公司运营,其数据存储和计算服务器均位于海外,出于服务器连接稳定性和数据安全等方面的考虑,GEE在处理国产对地观测卫星数据的问题时显然并不能胜任。因此,开源的地球观测数据分析管理平台——开放数据立方体(Open Data Cube, ODC)成为了国产对地观测卫星数据可行的解决方案。ODC与GEE相比,具有便于在本地部署,便于进行小范围数据分析,数据分析接口较为丰富等一系列技术优势。开放数据立方体已经成为全球地球综合观测系统(GEOSS)和由我国牵头成立的亚太区域综合地球观测系统(AOGEOSS)的重要技术平台<sup>[5]</sup>。从理论和实践层面都可以看出,ODC系统可以较为成功的解决国产卫星大数据管理和应用的难题,因此使用ODC作为国产卫星数据的管理解决方案是完全可行的。

当前,国内关于开放数据立方体的研究工作尚处于初期阶段,研究成果较少。Open Data Cube目前只支持Landsat 5/7/8, Sentinel-1/2, ALOS-1/2, ASTER, MODIS等数据,尚未支持中国卫星数据<sup>[6]</sup>。此外,现有的研究并未涉及国产卫星数据基于ODC的服务实现,仅停留在存储的研究和利用Python接口进行简单分析应用的层面。为解决上述问题,基于Linux环境部署ODC系统,在系统上实现了国产对地观测卫星高分一号(GF-1)WFV数据的加载,并利用已加载国产卫星数据进行了水探测服务案例的开发,在此基础上讨论了基于ODC系统的Web UI——Data Cube UI设计国产卫星数据服务的一般方法。本研究是对使用ODC的国产卫星数据管理模式的探索,将为相关的应用提供有效的参考。

## 2 ODC系统的部署

### 2.1 ODC介绍

Open Data Cube的前身是澳大利亚科工局和地调局开发的澳大利亚地学数据立方体(Australian Geoscience Data Cube, AGDC)。2018年,为更好满足全球用户对对地观测数据的需求,进一步提升对地观测数据的影响力,在地球观测卫星委员会(CEOS)的牵头组织下,开源数据管理平台Open

Data Cube被正式建立。ODC吸收融合了AGDC和GEE的诸多优点,在确保用户对数据的完全控制的同时,可以大大简化遥感数据准备过程,提升数据管理和查询效率,并提供了命令行、代码编辑器、Web UI等一系列丰富接口以便于用户与数据进行交互<sup>[7]</sup>。此外,ODC还提供了一个由一系列数据结构和工具组成的开源通用分析框架,有助于大型网格化数据收集、组织与分析。

经过数年的发展,ODC已经建立了集开发、科研、应用为一体的较为完善的生态圈(如图1所示),在全球范围内不断扩大着影响力。ODC官方于2019年开始全球多个国家和地区国家和地区进行定制化开发和业务化部署,并计划在2022年在20个国家或地区完成系统部署<sup>[8]</sup>。目前已完成部署的ODC区域系统有非洲区域数据立方体(Africa Regional Data Cube)和瑞士数据立方体(Swiss Data Cube)、澳大利亚数字地球(Digital Earth Australia)等<sup>[9]</sup>。基于ODC系统开发的中国数据立方体China Data Cube(CDC)在应用中也取得了良好的表现,目前正处于积极建设中<sup>[10]</sup>。



图1 ODC生态示意图

Fig.1 ODC ecosystem

### 2.2 ODC的组成和架构

ODC系统大致由3部分组成(如图2所示): ①开放数据立方体核心(ODC Core):是数据准备模块,用于实现数据的预处理和标准化,以便于后续基于时间序列的分析。②ODC应用(ODC Apps):是数据交互软件,用于查看元数据信息和管理数据。③ODC算法(ODC Algorithms):用于实现遥感

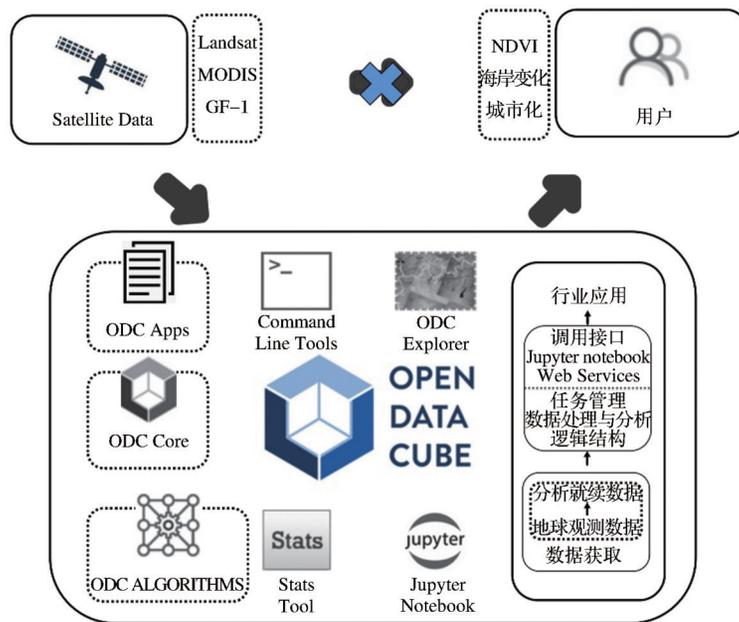


图2 ODC的组成与架构

Fig.2 ODC components and framework

数据的计算和分析<sup>[11]</sup>。通过上述3个模块的相互配合,用户在ODC系统上可以以自动化的方式完成预处理、建立索引、数据查询分析的完整工作流,从而降低了管理对地观测数据的时间成本。

为增加系统的可扩展性,提高系统与不同对地观测数据集的适配能力,降低系统的维护成本,开发者为Open Data Cube设计了一种分层架构<sup>[11]</sup>,该架构共分为4层(如图2所示),按照从底向上的顺序分别为:数据获取、逻辑结构、调用接口、具体应用<sup>[3]</sup>。

(1)数据获取层:其基本功能是管理用户写入的地球观测数据和经过预处理的数据。其中的摄取数据模块可以对ARD数据进行进一步管理,并且用NetCDF格式的储存单元来分块储存数据,使得数据的存储和调用效率获得提升。

(2)逻辑结构层:是架构中最核心的部分,一方面负责底层数据的处理、分析和可视化呈现,另一方面负责ODC任务的管理调度等全局任务。

(3)调用接口层:提供一系列可供开发者调用的接口,包括终端命令、Web接口、Python(Jupyter Notebook)接口等。用户开发的应用可以通过接口层提供的接口与逻辑结构层进行交互。

(4)具体应用层:指用户通过接口直接对数据进行分析,或设计一系列服务,以便将遥感数据应用于植被监测、海岸线动态监测、非法采矿监测、作物产量估计等具体方面。

### 2.3 ODC系统在Linux环境下的部署

研究基于64位Ubuntu 20.04进行部署, Linux系统运行环境为Oracle VM虚拟机。部署分为两个步骤:第一步是进行Datacube Core的安装:在虚拟机准备完毕后,开始安装ODC运行的依赖环境,主要有Python, GDAL, rasterio, netCDF4, PostgreSQL数据库等。第二步是Data Cube UI的安装:环境配置之后,安装Datacube Core并创建Datacube配置文件。然后安装docker和Data Cube UI,并使用控制台命令为Data Cube UI配置端口。所有安装完成后启动虚拟环境和服务器进行测试,结果成功连接至UI Web,即表明系统部署成功。

## 3 GF-1卫星数据的加载

高分一号发射于2013年4月26日,是我国高分辨率对地观测系统重大专项的首发星,其在提高我国高分辨率自给率,推动我国卫星工程技术的提升方面有着重要的意义<sup>[12]</sup>。高分一号搭载了4台16m分辨率宽幅相机(WFV),成像幅宽可达800km左右,并具有蓝、绿、红、近红外4个观测波段。高分一号卫星数据延续时间较长,有着较好的空间分辨率和覆盖度,因此具有很高的应用价值,同时也具有十分庞大的数据量。因此,探索使用Open Data Cube加载高分一号卫星数据的技术路线,在提升高分一号卫星数据生产力,发挥其数据应用潜力方面有着重要意义。

在ODC系统中加载卫星数据一般分为4个步骤:①定义产品;②提取元数据;③建立索引;④摄取数据。ODC官方提供了大致的技术路线,然而,

针对国产卫星数据,每一步骤中都需要根据国产卫星数据的特点进行额外的改动。加载国产卫星数据的全部流程如图3所示,步骤具体介绍如下:

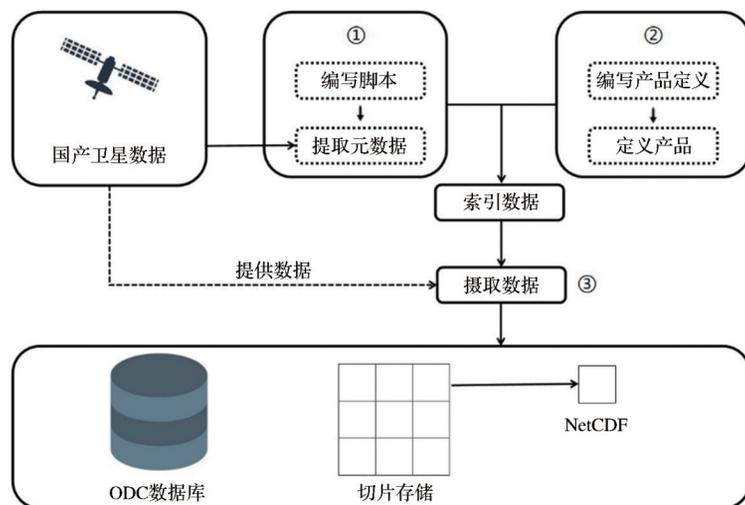


图3 向ODC中加载GF-1卫星数据数据的步骤

Fig.3 Steps for loading GF-1 data to ODC

### 3.1 定义产品

该步骤的目的是对GF1影像数据的类型进行正式定义和相关说明。在执行索引任务之前,需要为遥感数据集建立一个元数据文档,描述数据表示的内容、数据来源,以及数据存储的格式等信息,这个元数据文档即为ODC中的产品定义(product definition)。产品定义是一个格式为YAML的文件(同时支持JSON格式),文件中包含卫星数据的一系列信息,包括名称,产品类型,空间参考,波段信息等,考虑到单波段储存,还需将影像空间范围、成像时间、No Data值等信息储存入产品定义文件中。产品定义存储在索引中,用于搜索、查询和访问数据。ODC虽然提供了Landsat、MODIS等卫星影像数据的定义文件,但并未提供GF1等国产卫星影像数据的定义文件,我们针对此问题编写了根据高分一号卫星数据的特点编写了GF-1卫星的产品定义文件GF1.yaml(图4(a)),并成功运行命令将其添加进数据库。

### 3.2 提取元数据

近年来,在卫星数据元数据的整理阶段,由于现有软件不支持或元数据格式不统一,技术人员不得不采用传统手工作业方式,将元数据项的内容逐项填写,这浪费了大量的人力和时间。因此在ODC中,这一方式将通过标准化、自动化的方式实现,其具体实现方式就是提取元数据的Python脚本。

ODC目前只提供了Landsat系列和Sentinel系列卫星元数据的提取脚本,因此我们需要根据高分一号卫星数据的特点重新编写适用的脚本(图4(b))。脚本主要使用了yaml,GDAL等库,其大致结构为:①读取高分一号卫星元数据信息。②读取波段信息,生成波段文件储存路径。③将元数据信息和波段信息共同写入到yaml文件中进行输出,输出文件即为提取所得到的元数据。值得注意的是,高分一号影像4个波段的数据储存在一个TIFF格式文件中,这点与Landsat数据有很大不同,因此需要使用GDAL库将4个波段分离并分开存储,才能得到4个波段数据的正确路径。

### 3.3 建立索引

在向ODC加载了高分一号卫星的产品定义,并成功提取元数据之后,接下来的步骤是建立索引(index),即将提取所得的元数据与产品定义进行匹配,并将卫星影像数据路径存入数据库的过程。这一步可通过ODC自带命令实现。建立索引之后,该景卫星数据在ODC系统中便处于可查询状态,可以继续进行摄取操作。

### 3.4 摄取数据

摄取(ingest)是将用户所导入的GF1影像数据进行系统的切块、压缩等一系列操作,最后生成便于ODC分析计算的NetCDF格式的数据切片,这种数据格式可以不依赖于计算机和应用程序。由于

```

1 name: GF1
2 description: GF1 sence
3 metadata_type: eo
4
5 metadata:
6   platform:
7     code: GF1
8   instrument:
9     name: WFV1
10  product_type: STANDARD
11  format:
12    name: GeoTiff
13
14 measurements:
15   - name: 'band1'
16     aliases: [band_1, blue]
17     dtype: int16
18     nodata: -9999
19     units: 'reflectance'
20   - name: 'band2'
21     aliases: [band_2, green]
22     dtype: int16
23     nodata: -9999
24     units: 'reflectance'
25   - name: 'band3'
26     aliases: [band_3, red]
27     dtype: int16
28     nodata: -9999
29     units: 'reflectance'
30   - name: 'band4'
31     aliases: [band_4, nir]
32     dtype: int16
33     nodata: -9999
34     units: 'reflectance'

```

(a) 产品定义

```

: def sf_yaml(path):
    path_last = os.path.basename(path)
    path_tif = path_last + '.tif'
    path_xml = path_last + '.xml'
    path_xml = os.path.join(path, path_xml)
    path_ds = os.path.join(path, path_tif)
    ds = gdal.Open(path_ds)
    print(ds)
    im_width = ds.RasterXSize #栅格矩阵的列数
    im_height = ds.RasterYSize #栅格矩阵的行数
    im_bands = ds.RasterCount #波段数
    im_data = ds.ReadAsArray(0, 0, im_width, im_height) #获取数据
    im_geotrans = ds.GetGeoTransform() #获取仿射矩阵信息
    im_proj = ds.GetProjection() #获取投影信息
    im_blueBand = im_data[0, 0:im_height, 0:im_width] #获取蓝波段
    im_greenBand = im_data[1, 0:im_height, 0:im_width] #获取绿波段
    im_redBand = im_data[2, 0:im_height, 0:im_width] #获取红波段
    im_nirBand = im_data[3, 0:im_height, 0:im_width] #获取近红外波段
    proj = ds.GetProjection()
    id1 = uuid.uuid1()
    path_band = os.path.join(path, 'bands')

    path1 = os.path.join(path_band, 'blue.tif')
    writeTiff(im_blueBand, im_width, im_height, 1, im_geotrans, im_proj, path1)
    path2 = os.path.join(path_band, 'green.tif')
    writeTiff(im_greenBand, im_width, im_height, 1, im_geotrans, im_proj, path2)
    path3 = os.path.join(path_band, 'red.tif')
    writeTiff(im_redBand, im_width, im_height, 1, im_geotrans, im_proj, path3)
    path4 = os.path.join(path_band, 'nir.tif')
    writeTiff(im_nirBand, im_width, im_height, 1, im_geotrans, im_proj, path4)

```

(b) 提取脚本

图4 高分一号卫星产品定义与与提取源数脚本(部分)

Fig.4 Product definition and data preparation script for GF-1 satellite (partly showed)

卫星影像以切片的形式存储在数据库中,因此小范围分析和像元级的分析数据分析效率都会大大提高。

#### 4 GF-1 卫星数据水探测服务的实现

Open Data Cube 系统为用户提供了丰富的数据交互接口,包括了命令行工具, Jupyter Notebook, ODC Explorer, Web UI, Web 服务等。其中,其 Web UI——Data Cube UI 兼有数据管理和数据分析功能,便于用户进行二次开发,且界面易于操作,对没有信息技术专业背景的用户较为友好,因此有较高的应用价值。在成功加载了高分一号的数据之后,我们基于 ODC 的 Web 接口 Data Cube UI 设计并实现了高分一号卫星数据的水探测服务案例。

##### 4.1 Data Cube UI 架构

Data Cube UI 建立在 Python Web 框架 Django 的基础上,同时应用了 Celery, PostgreSQL, Apache/Mos WSGI 等技术<sup>[13]</sup>,其具体架构逻辑可分为前端和后端两部分:

(1) 后端包含了数据、数据分析算法和任务:①数据。Data Cube UI 中的数据主要分为遥感影像数据和 Django 数据两部分,卫星遥感数据存储在 Open Data Cube 中,Web UI 的数据(Django Model)存储在 Django 的数据文档 models 中,二者的数据库均基于 PostgreSQL 实现。②数据分析算法。Data

Cube UI 的算法基于 Django app 实现,所有的算法均继承自 dc\_algorithm 这一基类,官方提供了植被指数计算,水探测,海岸线变化探测等数据分析算法,用户也可以根据实际需要定义新算法。③任务(task)是算法执行的具体过程,其基于 Redis + Celery 实现。任务根据用户设定的算法参数,在后端获取数据并将其分割为易于分析的大小,对数据执行算法,并将处理结果输出,是前端与后端沟通的桥梁。

(2) 前端包含了模板(template),视图(view)。前者是 html 文件,包含网页页面的内容,格式等,可以通过函数从后端获取数据。后者是一个 Python 函数,用于根据用户的 Web 请求返回 Web 响应。二者共同控制页面的显示和用户的前端交互。

##### 4.2 水探测服务的创建

如上文所述,Data Cube UI 中内置了植被指数计算、水体探测、海岸变化探测等一系列算法,这些算法涵盖了对地观测数据的大部分的使用场景。本文基于一景已经成功加载的哈尔滨地区 2019 年 10 月 26 日的一景高分一号 WFV 数据,应用 Data Cube UI 内置的 Water Detection 算法,创建了该区域的水探测服务。创建服务的主要步骤为:

(1) 添加区域(areas):区域(areas)是显示在算法页面的单元,用户使用时,在选择算法之后进入到区域选择页面,选择分析案例所在的区域即可开

始分析。首先,在控制台中运行 Data Cube UI 的管理脚本,创建超级用户,以便于对后续对 Django 数据库的直接操作。然后,运行服务器,在浏览器中接入 Django Admin 管理工具的端口,并以超级用户身份登录。登陆成功之后,先在 satellite 类中创建新的卫星类型 GF-1,然后在 dc\_algorithm 类中的 areas 下根据地区地理位置范围、所选用的数据等信息,创建新区域并将其命名为 dongbei\_1。

(2) 将区域与算法匹配:在数据库中创建新区域的记录之后,其并不能在页面中显示,需要进行匹配操作。选择 Dc\_algorithm > Applications > water\_detection, 然后在 Areas 选项卡中勾选新创建的区域,并在下方的 Satellites 选项卡中勾选 GF-1。操作完成之后点击保存,即完成了区域的添加。此时进入 Data Cube UI 中的水探测页面,可以看到新创建的区域 dongbei\_1 (图 5)。



图5 成功添加区域后的“区域选择”页面

Fig.5 “Region selection” page after adding new page successfully

上述步骤完毕之后,服务即创建完成,点击图标即可进入数据分析页面(图6),在设定参数之后点击提交,即可使用加载的 GF-1 数据完成水探测分析。该流程同样适用于使用其他内置算法创建国产卫星的服务。

### 4.3 自定义服务的创建

Data Cube UI 有较好的扩展性,支持用户开发自定义数据分析算法,这使得用户不仅可以使⽤内置算法创建服务,还可以根据实际⽤户需求创建自定义服务。由于上文已经讨论了基于已有算法创建服务的步骤,因此创建自定义服务的关键在于如何创建自定义算法。创建新算法本质上是新建 Django app 的过程,大部分操作均需要通过 Django 的管理文件 manage.py 交互实现。由于篇幅所限,本文仅讨论创建基础应用(如波段运算)的方法。以波段运算算法为例,创建一个自定义算法有以下3个步骤(图7):

(1) 创建基本文件(Generating Base Files):该步骤是创建一切新算法的初始步骤。包括复制并

重命名基类中的基本文件,创建新应用的数据模型(model),为后续步骤创建指引等操作。

(2) 迁移(data migration):该步骤是新建 Django app 所必须进行的操作,其目的是在新的 app 中生成基本表。

(3) 算法实现(algorithm implementation):完成上述两个步骤之后,新算法的代码和基本数据结构已经完成构建,接下来需要根据在步骤(1)中创建的指引进行具体函数修改。在新 app 目录下的 task.py 文件中搜索“#TODO”标签,按照指引并根据波段计算公式修改对应部分<sup>[14]</sup>。

综上所述,可以得到基于 Data Cube UI 创建卫星数据的一般流程(如图7)。在创建服务时,首先应该根据实际应用场景,判断 Data Cube UI 内置的算法能否满足使用需求,若能满足,直接在数据库中创建区域并将区域与算法匹配即可。若不能满足,则需要根据实际需求自行创建算法,然后再进行区域的创建与匹配。

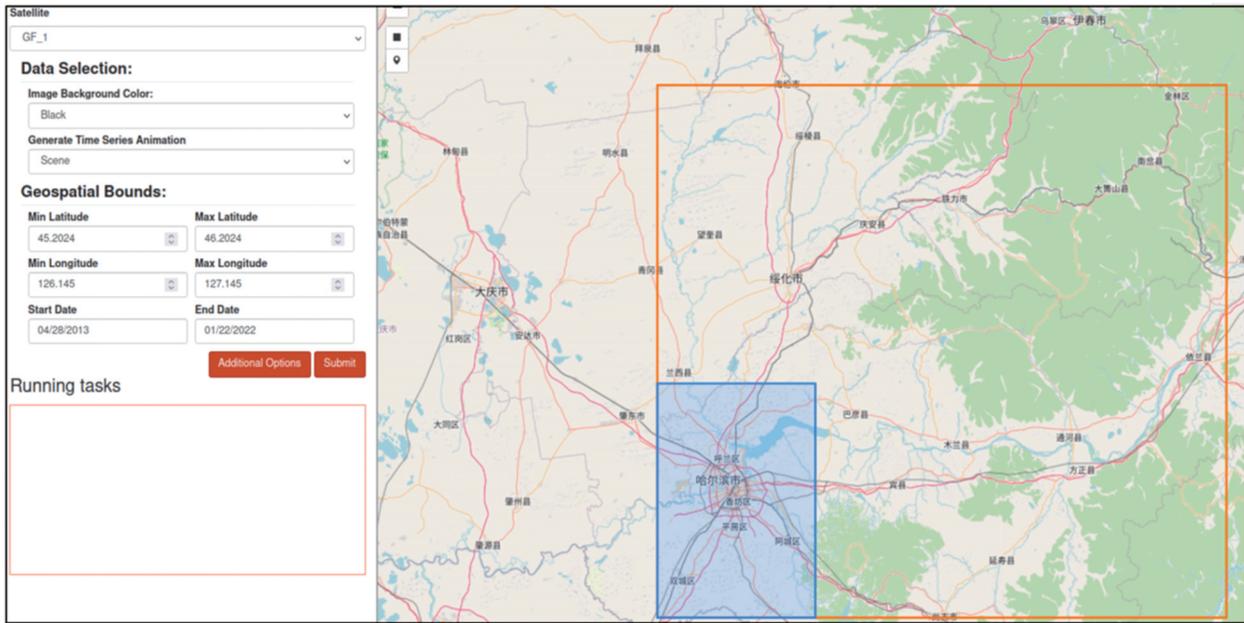


图 6 水探测分析页面

Fig.6 water detection analysis page

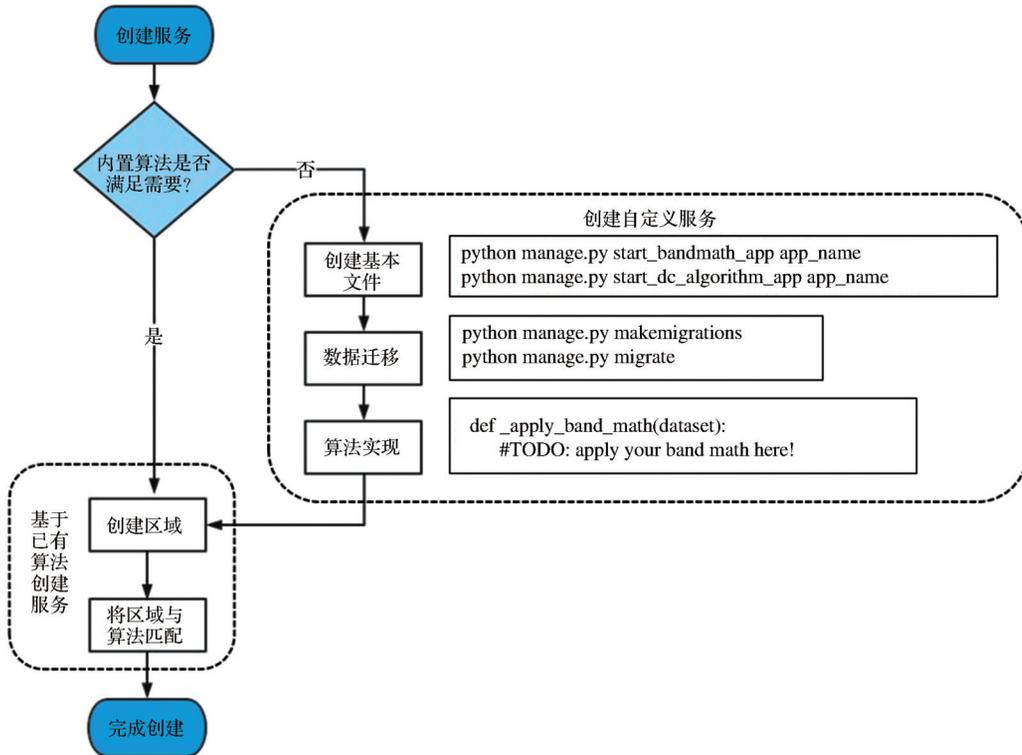


图 7 基于 Data Cube UI 创建卫星数据服务的流程

Fig.7 Workflow for creating satellite data services based on Data Cube UI

## 5 讨论

随着遥感技术的发展和进步,遥感卫星的种类、数量不断增多,卫星数据的时间、空间、光谱分辨率不断提高,遥感数据的种类和数量也在不断增加<sup>[15]</sup>。遥感大数据时代的来临对于广大的科研工

作者来说,既是机遇,也是挑战。先进有效的方式数据管理方案,可以将海量的对地观测数据转化为澎湃的生产力,因而可以使遥感科学和遥感相关应用领域发生翻天覆地的变化。我国的对地观测卫星事业正处于上升期,国产卫星带来的海量数据,

需要一套行之有效新型数据管理解决方案,而Open Data Cube作为一个开放、高效、架构先进的系统,可以成为国产遥感大数据管理的有效工具。

本研究的首要目的就是探索使用Open Data Cube系统管理国产卫星数据的技术路线。为了达到这一目的,需要解决两个问题:第一,国产卫星数据在ODC上的加载问题。ODC系统目前对中国卫星数据并不支持,这就需要根据高分一号WFV数据的特点编写产品定义和提取元数据脚本,完成定义产品、建立索引、摄取数据的完整流程,从而将国产数据加载到ODC上,即使国产卫星数据与ODC系统成功“适配”。第二,国产卫星数据服务的开发问题。为充分释放数据的生产潜力,在成功加载国产卫星数据后,还需利用ODC的相关接口,基于已加载数据创建易于使用的数据分析服务。基于上述思路,以目前数据量较大,使用频率较高的高分一号WFV数据为例,实现了在ODC上加载国产卫星数据并创建服务的技术路线,从而为基于Open Data Cube的国产卫星数据管理提供了技术经验和先验知识。

本研究也有一些局限性:首先,由于本文着重探讨加载国产对地观测数据的一般方法,故仅向Open Data Cube加载了高分一号WFV数据作为实验,并未对其他国产卫星数据的加载进行具体的尝试。其次,受限存储条件和设备算力,在研究中没有大规模地加载数据,对向ODC批量加载国产卫星数据的方法和效率没有进行深入的探讨。再次,在服务设计环节,我们没有对水探测服务的处理效率进行定量的探究。这些不足有待于后续相关研究进行完善和补充。

未来可以从两方面入手,继续深化本文的研究工作。一是可以探索多源国产卫星数据在ODC上的加载。近年来,我国的对地观测卫星事业取得了长足发展,相继发射了搭载不同类型传感器、涵盖不同分辨率的多个系列的遥感卫星,已经建立起陆地、海洋、大气三大系列的遥感卫星系统<sup>[16]</sup>。在本研究的基础上,探索使用ODC系统对多源国产对地观测数据进行的有效管理的方法,可以充分挖掘现有数据的潜力,对我国对地观测卫星体系未来的发展具有重要的意义。二是可以充分利用ODC官方提供的接口,针对现有的国产对地观测数据开发多种服务,从而更好地满足实际应用的需求。例如,以植被指数计算模块为基础,结合时序遥感

影像数据,开发植被物候探测服务;以不透水面指数计算模块为基础,结合历史遥感影像数据以及人口等其他类型数据,开发城市化进程探测服务等。

## 6 结 论

本文首先分析了遥感大数据时代给数据管理带来的挑战,并在横向对比的基础上,提出了适合国产卫星数据的新型数据管理解决方案——Open Data Cube系统。随后以高分一号WFV数据为例,讨论了在Linux环境下向ODC加载国产卫星数据的方法,并在成功加载后使用已加载数据,基于ODC的Web UI设计了水探测服务案例。具体的成果有:

(1) 在Linux环境下部署ODC系统,并按照ODC的官方规范,根据高分一号卫星的特征创建了GF-1 WFV的产品定义。

(2) 使用Python编写了提取高分一号卫星元数据的脚本,使用脚本成功提取了元数据。最终成功建立索引,摄取数据,将高分一号WFV数据加载至ODC系统中。

(3) 基于ODC提供的Web UI接口Data Cube UI,创建了哈尔滨区域高分卫星数据的水探测服务案例。并在此基础上探讨了使用Data Cube UI创建国产卫星数据服务的一般方法。

本文所探讨的在Linux环境下的ODC系统中加载国产卫星数据,并使用Web UI创建服务的技术路线,填补了国内关于开放数据立方体研究的空白,为使用ODC管理国产卫星数据的相关研究提供了先验知识,为国产卫星数据的高效管理提供了新思路和切实可行的办法。研究也存在一定的局限性:没有探讨其他国产卫星数据的加载流程,没有探讨大规模加载数据的方法,没有对定量探究水探测服务的处理效率,这些不足有待于后续的研究进行验证和补充。

### 参考文献(References):

- [1] LI Guoqing, PANG Lushen. A new age of public-oriented Earth observation development[J]. *Scientia Sinica Informationis*, 2017, 47(2): 193-206.[李国庆, 庞禄申. 公众化驱动的地球观测发展新时代[J]. *中国科学:信息科学*, 2017, 47(2): 193-206.]
- [2] CASUA F, MANUNTAA M, AGRAMB P S, *et al.* Big Remotely Sensed Data: Tools, applications and experiences [J]. *Remote Sensing of Environment*, 2017(202): 1-2.
- [3] LI Junjie, CHEN Shubo, ZHANG Wen, *et al.* Management and application of domestic satellite imagery based on Open

- Data Cube[J]. *Journal of Geo-information Science*, 2020, 22(9): 1861-1867.[李俊杰,陈舒博,张文,等.基于ODC的国产卫星影像存储与应用研究[J].*地球信息科学*, 2020, 22(9): 1861-1867.]
- [4] GORELICK N, HANCHER M, DIXON M, *et al.* Google Earth Engine: Planetary-scale geospatial analysis for everyone[J]. *Remote Sensing of Environment*, 2017, 202: 18-27.
- [5] GU Xingfa, ZHOU Xiang, ZHANG Songmei, *et al.* Asia-Oceania GEOSS: A GEO initiative to implement GEOSS for sustainable development in Asia-Oceania[J]. *Journal of Remote Sensing*, 2018, 22(4): 666-671.[顾行发,周翔,张松梅等.亚洲大洋洲区域综合地球观测系统计划进展[J].*遥感学报*, 2018, 22(4): 666-671.]
- [6] Open Data Cube.[EB/OL]. <https://www.opendatacube.org/installation>, 2021, 2022.
- [7] Open Data Cube. ODC Whitepaper[EB/OL]. [https://www.opendatacube.org/files/ugd/f9d4ea\\_1aea90c5bb7149c8a730890c0f791496.pdf](https://www.opendatacube.org/files/ugd/f9d4ea_1aea90c5bb7149c8a730890c0f791496.pdf), 2017, 2022.
- [8] Open Data Cube.he "Road to 20" International Data Cube Deployments[EB/OL]. [https://www.opendatacube.org/\\_files/ugd/8959d666661f43c6c0461497854700a123cc59.pdf](https://www.opendatacube.org/_files/ugd/8959d666661f43c6c0461497854700a123cc59.pdf), 2018-12, 2022.
- [9] Open Data Cube.Open Data Cube Overall[EB/OL]. <https://www.opendatacube.org/overview>, 2021, 2022.
- [10] YAO X C, LIU Y B, CAO Q Q, *et al.* China Data Cube (CDC) for Big Earth Observation Data: Lessons learned from the design and implementation[C]// 2018 International Workshop on Big Geospatial Data and Data Science (BGDDS). Wuhan, China, Sept. 2018.
- [11] LEWIS A, SIMON OLIVER S, LYMBURNER L, *et al.* The Australian Geoscience Data Cube—Foundations and lessons learned[J]. *Remote Sensing of Environment*, 2017, 202: 276-292.
- [12] BAI Zhaoguang. The technical characteristics of GF-1 satellite[J]. *Erospace China*, 2013(8): 5-9.[白照广.高分一号卫星的技术特点[J].*中国航天*, 2013(8): 5-9.]
- [13] RATTZ J, AMA A. CEOS Open Data Cube UI Guide[EB/OL]. [https://github.com/ceos-seo/data\\_cube\\_ui/blob/master/docs/ui\\_guide.md](https://github.com/ceos-seo/data_cube_ui/blob/master/docs/ui_guide.md), 2021-10, 2022.
- [14] RATTZ J, AMA A. Open Data Cube UI Algorithm Addition Guide[EB/OL]. [https://github.com/ceos-seo/data\\_cube\\_ui/blob/master/docs/add-ing\\_new\\_pages.md](https://github.com/ceos-seo/data_cube_ui/blob/master/docs/add-ing_new_pages.md), 2021-9, 2022.
- [15] LI Guoqing, HUANG Zhenchun. Data infrastructure for remote sensing Big Data: Integration, management and on-demand service[J]. *Journal of Computer Research and Development*, 2018, 22(4): 666-671.[李国庆,黄震春.遥感大数据的基础设施:集成、管理与按需服务[J].*计算机研究与发展*, 2017, 54(2): 267-283.]
- [16] ZHAO Wenbo, LI Shuai, LI Bo, *et al.* Development strategy of the new-generation effectiveness-oriented Earth-Observation System[J]. *Strategic Study of CAE*, 2021, 23(6): 128-138.[赵文波,李帅,李博,等.新一代体系效能型对地观测体系发展战略研究[J].*中国工程科学*, 2021, 23(6): 128-138.]

## Loading and Service Implementation of GF-1 Satellite Data based on Open Data Cube

SUN Zhelun<sup>1</sup>, XIAO Biyong<sup>1</sup>, ZHAO Zulaing<sup>2</sup>, YU Guojiang<sup>1</sup>,  
CAO Qianqian<sup>3</sup>, WANG Meng<sup>3</sup>, YAO Xiaochaung<sup>1</sup>

(1. College of Land Science and Technology, China Agricultural University, Beijing 100193, China;

2. Zhejiang Development & Planning Institute, Hangzhou 310030, China;

3. Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China)

**Abstract:** The Open Data Cube system is a new generation of open-source earth observation data management system, which can solve many challenges faced by Chinese satellite big data in the management and application fields. However, the system currently supports only international mainstream satellite data loading and services and is still not available for Chinese satellite data. Taking GF-1 WFV data as an example, we loaded and created a service of Chinese satellite data based on the Open Data Cube system. Firstly, according to the characteristics of GF-1 data, the product definition and metadata extraction script are written to index and ingest data, to complete the loading of Chinese satellite data. Then based on the loaded data, the water detection service of GF-1 satellite data is developed using ODC's Web UI interface. This research proves the feasibility of using the Open Data Cube system to manage Chinese satellite data and provides technical references for other Chinese satellite data loading and service implementation based on this system.

**Key words:** GF-1 Satellite Data; Data Management; Open Data Cube