

基于设计模式的遥感图像处理平台设计与实现

马江林, 赵志明, 杨 建, 杨 俊, 汪承义

(中国科学院遥感应用研究所国家遥感应用工程技术研究中心, 北京 100101)

摘 要: 设计模式作为一种面向对象软件开发过程中针对特定问题的解决方案, 已经成功应用到了多个软件的系统设计和架构中。遥感图像软件开发是一个复杂的系统工程, 其中遥感图像输入输出类库、算法类库和视图类库是整个软件体系架构中最重要的 3 个部分。该文结合遥感图像的特点, 利用设计模式的思想, 论述实现这些类库的过程。

关键词: 设计模式; 软件设计; 遥感图像

Design and Implementation of Remote Sensing Image Processing Based on Design Pattern

MA Jiang-lin, ZHAO Zhong-ming, YANG Jian, YANG Jun, WANG Cheng-yi

(National Engineering Research Center of Geoinformation, Institute of Remote Sensing Applications, CAS, Beijing 100101)

【Abstract】 Design pattern, as a smart design scheme for certain problems in object-originated software design, is successfully applied in many software projects. Remote sensing image software design and development is a complicated project, in which the remote sensing IO class design, algorithm class design and view class design are the most important parts. This paper elaborates how to design the three classes for remote sensing image processing based on design pattern.

【Key words】 design pattern; software design; remote sensing image

1 设计模式的基本思想

设计模式最重要的思想是“封装变化的概念”^[1]。一方面, 设计的软件具有一定的灵活性, 以适应可能的变化; 另一方面, 把这种复杂性所带来的软件复杂性进行封装, 为外界提供一个简单而又稳定的访问接口。

为此, 设计模式特别强调两个面向对象设计的原则^[1]:

(1) 优先使用类组合, 而不是类继承, 即合成/聚合复用原则: 一个新的对象里面使用一些已有的对象, 使之成为新对象的一部分; 新对象通过向这些对象的委派达到复用已有功能的目的。

(2) 针对接口编程而不是针对实现编程, 即依赖倒置原则: 抽象不应当依赖于细节; 细节应当依赖于抽象。

设计模式的分类有不同的方法, 根据目的准则, 将其分为创建型、结构型和行为型。创建型模式与对象的创建有关, 包括工厂方法等 5 种模式; 结构型模式处理类或对象的组合, 包括组合、桥接等 7 种模式; 行为型模式对类或对象怎样分配职责进行描述, 包括观察者、命令等 10 种模式。

虽然不同设计模式的设计思想是不同的, 但是它们并非都是孤立的, 而是常常组合在一起, 共同完成系统的设计。

2 遥感图像处理的特点

从程序设计的角度来看, 遥感图像处理程序应该具有以下特点:

(1) 多种图像格式的支持以及对新图像格式的快速扩展能力。作为一款通用遥感图像处理软件, 不仅要支持通用的图像格式和各种卫星的图像自定义的图像格式, 而且还要支持各种商用遥感图像软件数据格式, 对多种格式的支持是其必须解决的一个关键问题。

(2) 海量遥感数据的支持。对海量数据处理的支持是遥感图像处理软件区别于普通图像处理软件的一个重要标志。

(3) 多数据类型的支持。不同类型的遥感图像其图像中表示每个像素数值的数据深度不同。

(4) 多投影方式的支持。遥感图像处理与普通图像处理的一个差异在于遥感影像中包含着地理位置信息以及投影信息。多投影的支持是指支持不同投影之间的转换以及相应的图像几何形状的变化。

(5) 繁多的图像处理算法的支持。遥感图像处理算法从功能上可以分为图像增强、复原、纠正、镶嵌、融合、压缩、分割、分类、栅格矢量化和三维可视化等。算法是软件的灵魂, 遥感图像信息的提取, 需要多个算法配合来最终实现。

通过以上的分析可以看到, 与普通的图像处理软件系统相比, 遥感图像处理系统数据结构复杂多变, 数据来源多样, 功能点多。如果在设计开发中不采用适当的模式来消除各种依赖性, 会大大增加软件的开发成本和稳定性因素。采用设计模式是开发基于面向对象的遥感图像处理软件的必由之路。

3 设计模式在遥感图像处理软件开发中的应用

图像输入输出类库、算法类库和视图类库是通用遥感图像处理平台最为重要的 3 个部分: 输入输出类库用于解决不同数据源的读写问题; 算法类库用于解决遥感图像处理算法

基金项目: 国家“863”计划基金资助项目(2003AA135010)

作者简介: 马江林(1979-), 男, 博士研究生, 主研方向: 图像处理, 模式识别; 赵志明, 教授、博士生导师; 杨 建、杨 俊、汪承义, 博士研究生

收稿日期: 2007-02-10 **E-mail:** majianglin2003@gmail.com

问题；视图类库用来解决图像的可视化和用户交互的问题。3 个类库协同工作，共同完成了遥感图像处理和信息提取的工作。

3.1 基于工厂方法模式和单态模式的图像 I/O 类库设计

一个好的遥感图像处理输入输出类库应满足如下标准：

(1)模块化。该类应该是独立的，为应用提供统一的读写接口而与具体的针对数据处理的应用无关。该类库仅负责将图像数据从文件中读写，图像的读写功能与图像处理的算法完全隔离。

(2)可扩展性。支持新的数据格式无须改变已经存在的代码，因为很难预见未来会遇到什么样的图像格式。

为了满足上述标准，采用了设计模式中的工厂方法模式进行遥感图像 I/O 类库的设计，如图 1 所示。

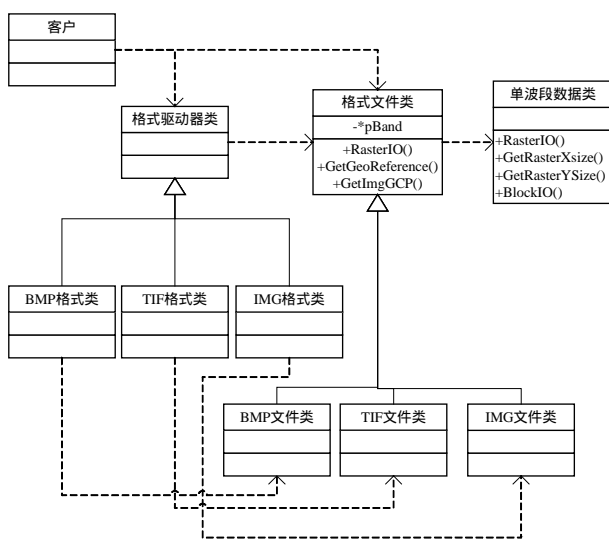


图 1 I/O 类框图

工厂方法模式是一种类的创建模式，其用意是定义一个创建产品对象的工厂接口，将实际创建工作推迟到子类中。在工厂方法模式中，核心的工厂类不再负责所有产品的创建，而是将具体创建工作交给子类去做。这个核心类仅仅负责给出具体工厂必须实现的接口，而不接触哪一个产品类被实例化这种细节。这使得工厂方法模式可以允许系统在不修改工厂角色的情况下引进新产品。工厂类与产品类往往具有平行的等级结构，它们之间一一对应。

基于工厂方法模式的图像 I/O 类库由 4 部分组成：

(1)抽象工厂。格式驱动器类对应于抽象工厂。在抽象工厂中，定义了不同格式读写的接口。

(2)具体工厂。BMP 格式类、TIF 格式类、IMG 格式类等对应着具体工厂。在具体工厂中，对具体格式读写的规则进行了定义和实现。如针对 BMP 格式，笔者根据 BMP 文件格式的特点，对 BMP 格式的读写规则进行了定义，同时对 BMP 格式相关的信息进行了设置。

(3)抽象产品。格式文件类是工厂模式所创建对象的超类，代表不同图像文件的公共接口，如文件中的投影坐标系、控制点、图像文件中波段的个数等信息。

(4)具体产品。具体产品是由具体工厂生产的，如 BMP 格式类工厂负责对 BMP 文件的读写。BMP 文件类就是具体产品。

为了支持海量数据，数据读写部分的实现是通过格式文件类的分块读写机制来保证的。在每个格式类中都重载了按

块读写的函数，相应的格式类根据该图像格式的特点对其进行重载，如 IMG 格式是按照 64 × 64 进行分块存储的，而 TIF 格式是按行存储的。

与普通图像处理不同，很多遥感图像的格式不像 BMP 图像那样有 BMP 后缀作为文件的标示。因此，在文件读写的时候就存在寻找合适的格式驱动器的问题。笔者借鉴文献[2]中的 clone-factory 方法，在调用图像 I/O 类的时候，生成一个格式驱动器管理类，负责对所支持的图像格式类型进行管理。为了保证程序中有唯一的格式驱动器管理类，采用了单态设计模式。当打开一幅遥感影像图的时候，从格式驱动器管理类对象中寻找合适的格式驱动器，创建相应的图像数据集类。

采用工厂方法模式非常适合对新图像格式的支持而不改动原先读写类库，为了支持一种新的图像格式，仅需要做如下工作：(1)在格式驱动器类中根据新图像格式如头文件的大小和结构信息，进行读写规则的定义；(2)对新图像格式文件中分块读写函数，进行重新定义和实现；(3)将格式驱动器类注册到格式驱动器管理类中。

3.2 基于策略模式、模板模式和外观模式的算法类库设计

从程序设计的角度看，图像处理算法可以分 3 个不同的层次：(1)针对内存块的基本算法；(2)针对文件操作的基本算法；(3)基于数据流模型的算法流程。

针对内存块的基本算法不去考虑遥感图像海量的特点，只针对特定的数据块进行算法的设计而不去考虑数据是如何组织放置于内存块的；针对文件操作的基本算法，考虑如何将海量的遥感数据填充到算法所需要的内存块中以及图像附加信息处理；针对数据流模型的算法是将基本算法组织成数据流管道，完成复杂的算法流程。层次(1)和层次(2)是层次(3)的基础，而层次(1)同时也是层次(2)的基础。

(1)基于策略模式的针对内存块的基本算法

基本图像处理算法根据对内存块中数据处理方式的不同，可分为基于点处理的算法、基于邻域运算的算法、基于正交变换的算法、基于多波段的算法和基于几何变换的算法。

1)基于点运算的算法包括灰度反转、指数变换、对数变换、线性变换、阈值变换、窗口变换、直方图均衡等。这类算法的特点是，像素点灰度变换的规则一旦定义，运算后新像素的值可以唯一确定。

2)基于邻域运算的算法包括边缘检测、图像平滑、中值滤波、图像腐蚀、图像膨胀等。这类算法的特点是输出的像素值是由周围像素决定的。

3)基于正交变换的算法包括 FFT、小波变换、离散余弦变换、沃尔什变换等。这类算法的特点是需要先对行进行变换，然后再对列进行变换。

4)基于多波段的算法包括 KL 变换、RGB/HIS 彩色变换、缨帽变换等。这类算法的特点是输出的像素是由该点在不同波段上灰度值运算而获取的。

5)基于几何变换的算法包括图像平移、镜像变换、转置变换、缩放变换和旋转变换。这类算法的特点是给出一个分块的内存区域，返回一个新的内存分块区域。

基于上面的分析，可以看到基本图像处理算法之间的区别仅在于它们的行为，即对某特定内存块处理策略的差异。因此，采用了策略模式对基本算法进行设计。

策略模式的用意是针对一组算法，将每一个算法封装到具有共同接口的独立的类中，从而使得它们可以相互替换。

策略模式使得算法可以在不影响到客户端的情况下发生变化。基于策略模式，笔者设计的算法类库如图 2 所示。

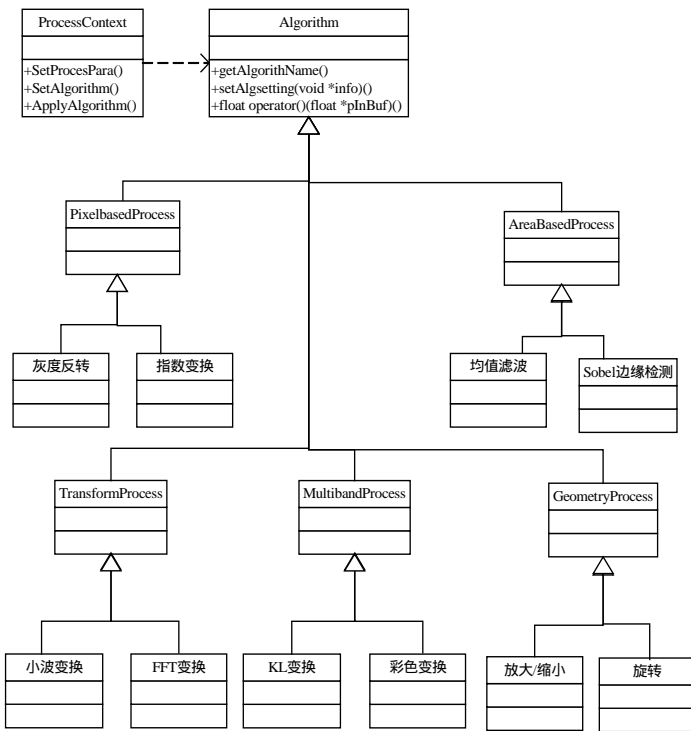


图 2 针对内存块基本图像处理算法类库框图

这样可以将大部分算法统一在一个框架下面，通过将算法与图像文件数据分离机制，提高了算法复用程度，使得不仅算法可以应用于基于文件的图像处理程序中，同时可以将算法方便地应用到视图显示处理中。

(2)基于模板模式的针对文件操作的基本算法

针对文件处理的基本算法与针对内存块的算法不同，遥感图像海量的特点，使得直接将图像数据填充入内存是不合适的。同时，遥感影像具有地理信息、投影信息、彩色表和元数据等信息，这些信息会随着采用算法的不同发生变化。

虽然针对文件的算法实现复杂，但实现流程可以模板化。不管采用哪一种算法，都需要选择合理的内存分配方案；都需要选择针对内存块操作的算法；都需要对文件的辅助信息进行修改和写入。根据这样的特点，针对文件的基本算法采用了模板模式，见图 3。

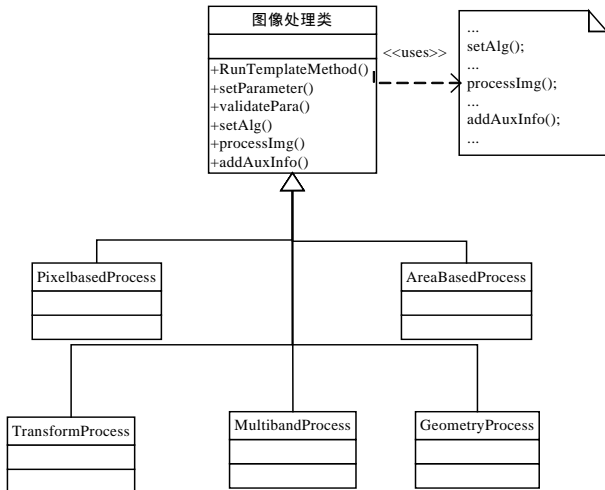


图 3 针对文件的基本图像处理算法类库框图

模板方式模式特别适合这种轮廓和骨架类似的算法，笔者定义了针对算法的流程模板方法 RunTemplate Method()，在该模板中定义了针对文件处理算法的一般流程，而在具体模板类中对模板方法中的基本方法进行了重载，如 processImg()函数包含了不同方法内存分配的不同方案。在基于点像素运算的方法中，内存分配方案是单波段分块读写处理的方案；基于区域内存分配方案跟基于点运算的方案类似，但是强调了分块临近边缘点内存中元素的处理；基于多波段处理的采用了基于多波段读写的分块方案；基于变换域的方法是采用了临时文件的方法进行分块读写的。

(3)基于门面模式的数据流模型算法

遥感图像处理算法复杂，但是很多算法可以由若干基本算法组合实现的，比如基于彩色变换的图像融合算法，其流程见图 4。

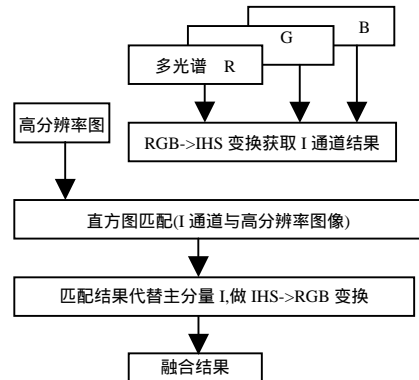


图 4 基于 IHS 变换的像素级融合流程

为了能够让像融合算法这样复杂的算法有一个通用方便的接口，调用更为方便，采用了门面设计模式对其进行了包装，如图 5 所示。图像缩放类、图像直方图规定类、图像彩色变换类、图像波段合成类组成了一个处理流水线，形成了一个计算框架来完成融合处理。

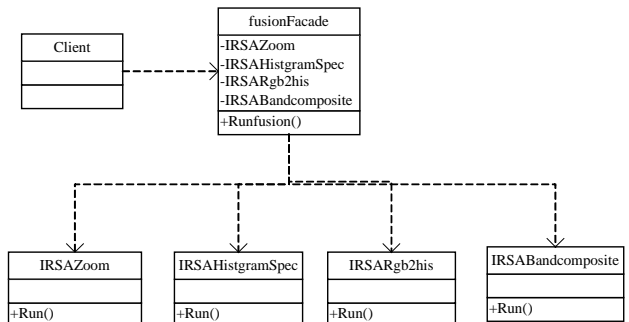


图 5 采用门面模式的融合类图

3.3 基于多种模式的视图类库设计

在遥感图像处理软件中，多个模块涉及到图像的显示问题，如纠正、镶嵌、分类和专题图制作等。若每个模块都写自己的显示模块，不仅会重复劳动，同时会带来代码维护的问题，因此，设计了一个图像显示的中间件：视图类库。

由于 GIS 视图类库和遥感图像处理视图类库的相似性，因此借鉴了文献[3-4]中对 GIS 视图类处理的一些解决方案：

(1)采用桥梁模式进行图像的显示，将抽象部分和实现部分分离，提高了视图的可重用性。

(下转第 71 页)