­­摘 要

地理信息系统是一种专门用于采集、存储、管理、分析和表达空间数据的信息系统，它既是表达、模拟现实空间世界和进行空间数据分析的“工具”，也可看做是人们用于解决空间问题的“资源”，同时还是一门关于空间信息处理分析的“科学技术”。[1]

2000年以来，随着组件化平台的发展，GIS系统也从一开始的一体软件，核心软件发展到基于Com/ActiveX的组件化平台[2]。现在的GIS系统开发人员可以很快的基于GIS软件服务商提供的SDK和GIS服务，实现一个个性化的GIS系统解决方案。

近些年，随着地理信息系统应用的深入，涌现出大量分析、挖掘、重组地理信息系统数据的需求，以及在新的技术条件下地理信息系统的新发展和突破，促使我们深入学习地理信息系统基础理论、算法和技术知识。

自从苹果公司在2007年推出iPhone手机以来，基于智能终端的广义移动GIS在各行各业普遍应用。传统的GIS应用正在向移动GIS发展，基于位置的服务 LBS(Location Based Service)和移动位置服务 MLS(Mobile Location Service)已经成为移动GIS的主要方向之一。

以谷歌、百度为首的互联网服务商，已经提供了面向消费领域的GIS服务及开发工具，但是相较桌面GIS组件，这些面向消费的GIS开发工具缺乏标绘，专题地图等创造生产力的开发功能。论文主要研究工作以在IOS系统下开发带有可扩展标绘GIS组件为主要对象，同时考虑新开发组件的未来可用性相关问题，并实施相关研究工作。本课题研究如何利用IOS下的Quartz 2D和OpenGL ES技术实现GIS组件的开发，从而为今后的开发、设计人员能够创建更好的基于IOS系统的GIS组件打下基础。

在研究过程中，探索实现GIS基本功能的原理和方法，学习已有的GIS组件中，分析本GIS组件针对的业务领域的需求，结合其具体业务案列制定合适的GIS组件。

本课题通过软件复用的思路和方法，对所需GIS组件功能进行框架设计和研究，取得如下成果：通过Quartz 2D和OpenGL ES技术为绘制业务中所需的GIS数据搭建出一个GIS组件——IOS.GIS.Common.Tools，其具有GIS基本的功能：包括坐标转换、测量工具、专题地图、图层管理、手工标绘，同时可以兼容ArcGIS，谷歌地图，百度地图等第三方IOS地图开发工具。

**关键词** GIS；组件化；IOS；矢量标绘；

Abstract

**Keywords** more than 5 words.

目 录

摘 要 I

Abstract II

目 录 III

第1章 绪论 1

1.1 研究意义 1

1.2 研究背景 2

1.3 研究内容 5

1.4 论文结构 5

第2章 移动GIS应用概述 6

2.1 GIS及移动GIS功能分析。 6

2.2 空间数据结构 8

2.2.1 栅格数据 8

2.2.2 矢量数据 8

2.3 地理信息可视化 9

2.3.1 地理信息可视化的目的 9

2.3.2 地理信息可视化的方式方法 9

2.3.3 地图数据来源分析 9

2.4 课题GIS功能需求 10

2.4.1 空间数据结构定义 10

2.4.2 空间数据处理 10

2.4.3 地理数据可视化 11

2.4.4 图层管理 11

2.5 本章小结 11

第3章 IOS.GIS.Common.Tool 功能包概要设计 12

3.1 当今GIS系统应用的架构设计 12

3.1.1 GIS解决方案 12

3.1.2 现有移动GIS的组件架构设计 13

3.2 IOS.GIS.Common.Tool功能包 14

3.2.1 总体设计 14

3.2.2 位置服务子系统 15

3.2.2.1 地理坐标模块 15

3.2.2.2 移动定位服务模块 15

3.2.3 矢量数据子系统 16

3.2.3.1 矢量数据结构模块 16

3.2.3.2 矢量数据持久化模块 16

3.2.3.3 矢量数据可视化模块 17

3.2.4 栅格数据子系统 18

3.2.4.1 栅格数据结构模块 18

3.2.4.2 栅格数据源模块 18

3.2.4.3 栅格数据可视化模块 18

3.2.5 地图UI子系统 20

3.2.5.1 地图浏览模块 20

3.2.5.2 图层定义模块 20

3.2.6 设计的可裁剪性 21

3.2.6.1 位置服务子系统的裁剪 22

3.2.6.2 矢量数据相关GIS模块的裁剪 23

3.2.7 设计的扩展性 24

3.2.7.1 扩展符号数据类型和符号数据构造器 24

3.3 本章小结 24

第4章 地理坐标模块详细设计 25

第5章 移动定位服务模块详细设计 26

第6章 栅格数据结构模块详细设计 27

6.1 瓦片地图技术 27

6.1.1 瓦片地图技术应用场景 27

6.1.2 瓦片地图技术原理概述 27

6.2 瓦片金字塔模型研究 27

6.2.1 瓦片金字塔的分辨率 27

6.2.2 瓦片金字塔的构建 28

6.3 瓦片地图的定义 29

6.3.1 瓦片地图参数 29

6.3.1.1 缩放级数 29

6.3.1.2 地图分辨率 30

6.3.1.3 比例尺 30

6.3.1.4 瓦片的尺寸规格 30

6.3.1.5 瓦片的矩阵范围 31

6.3.2 瓦片地图的拓扑关系 32

6.4 瓦片地图的数据结构 33

6.4.1 瓦片的数据结构 33

6.4.2 瓦片矩阵的数据结构 34

6.4.3 瓦片金字塔的数据结构 36

6.5 栅格数据源模块详细设计 37

6.5.1 基于WMTS接口的地图源 38

6.5.1.1 WMTS标准介绍及实现 38

ServiceMetadata 38

Tile 40

6.5.1.2 以Bing地图为例说明WMTS数据源的消费方式 41

加载Bing地图的实现 41

Bing地图常量 41

Bing地图比例尺和缩放等级 42

Bing瓦片像素坐标系 43

Bing瓦片坐标系和四叉树键值编码 43

Bing地图图像元数据 44

请求元数据 44

元数据响应信息： 44

BingDataSource初始化Bing地图和加载瓦片地图 46

请求元数据构造瓦片金字塔 46

请求瓦片图像数据 47

计算四叉树键值 48

6.5.2 基于静态图片的地图源 49

6.5.2.1 使用静态图片渲染瓦片地图的参数转换 49

6.5.2.2 以百度地图为例实现基于静态地图的渲染 50

第7章 栅格可视化模块详细设计 51

第8章 地理图形操作模块详细设计 52

8.1 矢量元素图层 52

8.1.1 矢量图层基本元素 52

8.1.2 矢量图层类关系 53

8.1.3 矢量图层绘制流程 53

8.1.4 矢量图层绘制算法 54

8.2 地图浏览接口 54

8.3 坐标转变换接口 54

8.4 图层管理 54

第9章 矢量标绘模块 55

第10章 数据管理模块 56

第11章 位置服务模块 57

第12章 案例验证 58

第13章 结 论 59

第14章 参考文献 60

第15章 致 谢 63

1. 绪论

## 研究意义

从2000年至今，GIS组件化发展迅速，ESRI和MapInfo分别推出了MapObjects和MapX等功能强大的组件系统。GIS组件具有小巧灵活、价格便宜、易于应用、对于GIS基本知识要求不高等优点。

同时随着智能终端的普及，移动GIS组件也有了大规模的发展，以谷歌和百度为首的互联网服务商纷纷提供了基于自家平台的面向消费领域的 移动GIS组件，例如：百度地图SDK，天地图SDK等。

但是这些移动GIS组件，相对于现有的桌面GIS组件来说，其提供的开发能力相对集中在基于位置的服务 LBS(Location Based Service)和移动位置服务 MLS(Mobile Location Service)之上的路由选择、设施查询以及数据采集等服务消费领域，对于企业级应用中所需要的标绘，专题地图等功能少有涉及。

同时在不同的应用背景下，采用不同的应用服务商来寻求低成本高效益这样的混合开发模式也被越来越多的企业所采纳。这样的趋势也影响了GIS组件开发，例如ESRI公司提供的ArcGIS组件库就可以将谷歌地图或者Bing地图作为第三方地图来源使用。相对而言，现有IOS平台下的GIS组件更加封闭，倾向于只接纳自家的地图等服务，而不跟第三方服务商融合。

论文主要研究工作以在IOS系统下开发带有可扩展标绘GIS组件为主要对象，兼容多种平台为目标，同时考虑新开发组件的未来可用性相关问题，并实施相关研究工作。其最终目标为了服务于今后更多的基于IOS的GIS相关系统，提供一个高可复用性的GIS组件，降低开发成本。

利用已有的GIS组件进行二次开发是国内GIS应用的很大一部分趋势，不容否认这种形式的复用，会使二次开发变得非常的简单，从而，从软件开发的各个方面提高了效率，减少了成本[3]。GIS应用的开发人员更专注于业务需求，不必学习过多的GIS基础知识。然而，这样的简单复用同时也导致了开发人员缺乏基本的地理信息知识，在GIS系统向第四代GIS软件转化，进而具备支持数字地球 (区域、城市) 的能力成为OS、DBMS 之上的主要应用集成平台, 实现由二维处理向多维处理的转变;由面向地图处理向面向客观空间实体及其时空关系处理 的转变;由以系统为中心 向以数据为中心,实现空间数据共享与服务的转变;由管理型向分析决策型的转变过程中，应对大规模涌现出来对于地信数据进行分析、挖掘、重组的需求乏力，以及不能及时利用在新的技术条件下地理信息系统的新发展和突破。

GIS应用所涉及的行业和领域越来越多，专精于某一个GIS应用领域，就必须深刻学习GIS本身的理论和技术知识，同时拥有一套自己的GIS组件，独立开发的GIS组件灵活性大，更容易响应自己专精领域的发展和变化，也可以在经过一段时间研究后，也可以发展成对于特定领域特定问题的解决方案框架。

## 研究背景

1. **地理信息系统（GIS）概述**

地理信息系统是以采集、存储、管理、描述、分析地球表面及空间和地理分布有关的数据的信息系统。它是以地理空间数据库为基础，在计算机硬、软件环境的支持下，对空间相关数据进行采集、管理、操作、分析、模拟和显示，并采用地理模型分析方法，适时提供多种空间和动态的地理信息，为地理研究、综合评价、管理、定量分析和决策服务而建立起来的一类计算机应用系统。简而言之，地理信息系统是以计算机为工具，具有地理图形和空间定位功能的空间型数据管理系统[4]

地理信息系统的基本应用功能主要为以下5类：

1. 位置：即对象的位置在哪里？通常利用特定坐标系和投影方式确定空间对象的位置，如坐标或街道编码。确定位置 后再通过查询等其他方式获得其他的特性。
2. 条件：即满足某些特定条件的对象在哪里？通常利用空间对象的属性信息列出条件表达式或组合表达式，进而获取满足这些表达式的对象的空间位置信息，通常会在屏幕上以高亮方式显示出来。
3. 趋势：即特定地区发生的特定事件及其随事件变化的过程。它需要综合既有数据，以识别已经发生或者正在发生变化的地理现象。趋势的确定通过一个数据集合来进行，当数据越完备的时候，确实分析可能越趋于正确。地理信息系统还可以根据产生的趋势快速生成一些特定能量数据以及说明该趋势的附图等。
4. 模式：即特定地点存在的空间实体的空间分布及空间关系的模式问题。模式分析揭示了地理实体间的空间关系。模式的确定需要长期的观察，理解数据之间的潜在关系及规律。
5. 模拟：即特定地点若具备某种特定条件将会发生什么问题。模拟是以模式和趋势为基础的，在它们的基础上，模拟将建立现象和因素之间的模型关系，从而发现具有普遍意义的规律。
6. 国内外GIS发展概况

在国际上，GIS是从二十世纪六十年代中期开始发展起来的技术。1963年，加拿大测量学家R.F Tomlinson首先提出了地理信息系统这一术语，并建成世界上第一个GIS(加拿大地理信息系统CGIS)。不久，美国哈弗大学提出了较完整的系统软件SYMAP。这可算是GIS的起步。在七十年代，一些经济发达国家先后建立了许多专业的GIS系统。并且随着八十年代计算机网络技术的兴起，以及民用电脑的普及，地理信息产业链趋于完整。截止1995年，市场上有报价的软件已打上千种，并且涌现出一些有代表性的GIS软件。

总之，用户的需求，技术的进步，应用方法论的提高，以及相关组织结构因素，都深深影响着地理信息系统的发展。

国内的GIS研究发展起步于七十年代，但是发展迅速，在政务业务和少量的大型企业应用方面都有具体的应用，例如：地理信息系统已在资源开发、环境保护、城市规划建设、土地管理、农作物调查与结产、交通、能源、通讯、地图测绘、林业、房地产开发、自然灾害的监测与评估、金融、保险、石油与天然气、军事、犯罪分析、运输与导航、110报警系统公共汽车调度等方面。

在传统的GIS系统开发领域，国内主要是基于现有的GIS组件进行二次开发。这些二次开发基于国外成熟度较高的GIS组件，例如：ESRI的ArcGIS、MapInfo的MapX等。随着国外专业GIS组件发展越发完善，相应的GIS组件也越来越封闭，二次开发的能力以及对于GIS的研究发展也都受到了限制。

同时，互联网企业近年来大举进入GIS领域，很多的网站都提供了地图功能及一些搜索，寻路的服务，例如：百度地图，MapABC，天地图，高德等。可惜的是，这些互联网地图服务商，大多数专注于消费领域的业务，所提供的API对于企业服务的支持十分有限。并且由于企业应用对于数据安全的敏感性很高，所以在企业级开发领域很少会将自己的数据托付给这些互联网地图服务商。

在国家“十二五”规划中，提出了产业化发展具有自主知识产权的GIS和物联网等相关领域系统软件。长期以来，地理信息系统领域存在着研究和市场脱离的现象，对于GIS软件的研究和开发主要在高校和科研院所中进行，与市场脱节。从“九五”期间，这种情况得到了根本的改善，建立并且发展了一批以GIS为主要经营方向，具备深厚研发能力的企业。在“十二五”期间，国家更加重视GIS的发展，重点关注了谷歌地图，bing地图提供的在线地图服务在消费端及物流等领域的应用，以及我国大型桌面系统主要采用美国地图软件的现状，提出要加大对研发有自主知识产权GIS的支持力度，例如“天地图”公共服务平台；以及在物联网、政务网等GIS应用领域推广自研GIS系统。在国家的大力支持下，一批研发GIS系统和采集地图信息的企业获得了很快的发展，例如：超图公司及其SuperMap产品，高德公司及其高德MapABC产品等。

总之，很多优秀国内产品的出现，有力缓解了国外GIS系统独大的局面，市场竞争也越发激烈。国内产品，在大型多用户GIS软件领域中，国产软件主要面临ARC/INFO的巨大压力，在小型和桌面系统领域（国产软件主要集中在这一方面）里，主要面临MapInfo的攻势，最近，AutoDesk也加入了角逐。由于国外软件厂商的经济实力比较雄厚，产品升级速度很快，在市场开拓方面有丰富的经验并进行大量投入，已经在中国占有了相当的市场，特别是针对国产GIS企业及其软件的崛起采取了相应的措施。因此，今后几年，市场竞争将会十分激烈。

1. 国内外地图组件发展现状

GIS组件作为用户二次开发的工具，为用户带来了极大的方便。现在主要流行的产品有：

1. MapX

MapX提供了一个可以嵌入到企业应用的GIS组件。MapX为开发人员提供了一个快速、易用、功能强大的地图化组件。

MapX定义了一个类体系，以有效的组织图形元素，图层，属性数据等对象。MapX的主要功能包括：显示MapInfo格式的地图；对地图进行放大、缩小、漫游、选择等操作；专题地图；图层控制；数据绑定；动态图层和用户绘图图层；生成和编辑地图对象；简单地理查询：边界查询，地址查询。

1. ArcObject

ArcObject 是 ESRI 公司 ArcGIS 家族中应用程序的开发平台，它是基于 COM 技术所构建的一系列 COM 组件集。ArcGIS 的所有产品都是以ArcObject 组件为基础平台搭建起来的。通过 ArcObject 可完成以下甚至更多的 GIS 功能：空间数据的显示、查询检索、编辑和分析；创建各种专题图和统计报表；高级的制图和输出功能；空间数据管理和维护等等。ArcObject 已经提供了许多底层的基本功能，而使用者的任务是按照应用需求将这些底层功能组装集合成一个更强大的 COM 对象。由于 ArcObject 是基于微软的 COM 技术来构建的，因此它的开放性和扩展性很强大[15]。

新一代ArcGIS中的许多应用层模块,如ArcMap、ArcCatalog和ArcScene等,都是以AO组件为其内核构建的。ESRI运用AO的对象模型建立ArcGIS，对于用户来说，ESRI的基本对象组件和其它第三方组件没有任何不同，用户可以运用同样的方法对ArcGIS进行扩展。只要安装了ESRI的桌面产品ArcGIS Desktop，用户就可以使用组件进行二次开发。此外，ESRI还提供了可以独立安装使用的组件开发包MapObject和ArcEngine，这些组件开发包对AO进行了二次封装，用户可以不必安装ArcGIS Desktop应用程序即可使用GIS组件，并且开发的程序也无须运行在安装过ArcGIS Desktop的环境中,只要安装ArcEngine的运行时即可。这更加方便了用户使用GIS组件并且节省了开发的成本。

1. SuperMap IS

SuperMap IS是北京超图地理信息技术有限公司自主研发的Internet GIS开发平台。

Supermap是中科院地理资源中心即超图公司开发出的国内比较优秀的地理信息系统产品.可视化编 程语言平台提供了较好的组件集成功能,Supermap是一个标准的控件GIS产品,可以在多种语言上使用, 具有容易实现系统的无缝集成特性、良好的扩展性、强大的伸缩性和流行的可视化程序设计风格等特点, 最主要的是其能够与大型数据库产品如ORACLE,SQI.SERVER连接,使得在系统开发中一个数据库产品 就能完成所有数据(图形数据、文本数据)存储和管理,使得数据间的访问更为有效。

1. IOS简介

IOS是由苹果公司为iPhone手机开发的操作系统。它主要是给 iPhone、iPodtouch以及iPad使用。就像其基于的 MacOSX操作系统一样，它也是以 Darwin为基础的。原本这个系统名为iPhoneOS,直到2010年6月7日WWDC大会上宣布改名为IOS。

IOS的系统架构分为四个层次：核心操作系统层 (the Core OS layer)，核心 服 务层 (the Core Services layer)，媒 体层 (the Media layer)，可轻触层 (the Cocoa Touch layer)。系统操作占用大概240M 的存储器空间。

iOS应用开发使用的语言是Objective-C,它是一个类C、C++的开发语言,有着与C相似的语法,用着和C++相同概念的面向对象的编程方法,并在其基础上添加了有着自己特征的新概念,如“类别”、“协议”,而这些都与 C++或JA V A中有些概念相似,当然另类的语句书写方法, 让初学者有些不适应,但强大的开发工具——XCode,还 是给应用开发增色不少。

## 研究内容

本论文将涉及到的研究内容有以下几个方面：

首先，需要分析GIS的基本功能及移动GIS的特征功能，主要内容是空间数据文件的存储方式及加载、及对于矢量数据数据格式的定义及读取。

其次，在IOS上实现GIS矢量数据的绘制功能，包括投影坐标转换、图层管理、用户交互，基本地图渲染。

继而，对于在GIS地图基本功能实现的基础上，设计图标标绘系统，封装绘制逻辑，对外提供矢量绘制接口、图标属性设置功能接口、图标查询体系及图标互动功能，和对于数据的可视化展示接口。

进而，实现移动GIS的定位、地址解析、信息采集等功能，以及尝试兼容现有的一些地图服务。

最后，利用开发的GIS组件进行二次开发，构建实际GIS应用程序。

## 论文结构

第1章绪论部分，主要针对本课题的研究意义、研究背景以及研究内容进行了简述；

第2章GIS及移动GIS的基本功能与空间数据部分，简要介绍了两种数据结构，以及数据来源。重点论述了本课题需要为特定用途空间数据处理工具提供的GIS具体功能；

第3章IOS.GIS.Common.Tool 功能包的总体设计，以及其他功能的配合使用方案；

第4章制图和矢量标绘模块设计与实现，重点介绍了制图和矢量标绘模块基于第三章的需求所进行的详细设计以及实现；

第5章对于利用IOS系统的定位功能，实现本GIS组件的位置信息服务功能，以及对于一些现有互联网地图服务的兼容方案。

第6章利用IOS.GIS.Common.Tool功能包进行二次开发，通过此章节展示对IOS.GIS.Common.Tool的使用，以及验证本功能包的实际开发效率。

1. 移动GIS应用概述

## GIS及移动GIS功能分析。

GIS(Geographic InformationSystem)是专门用采集、贮存、管理、分析和表达 空间数据的信息系统，移动GIS是GIS在移动设备如PDA、智能手机、Pad上的应用。

21世纪初，随着互联网的普及，尤其是在各大公司提供了瓦片地图的基础上，WebGIS得到了广泛的应用。

从2007年苹果公司推出iPhone手机后，移动互联网高速发展。移动互联网的发展，催生了移动GIS(Mobile GIS)，促进了基于位置服务(LBS)的发展，扩大了GIS的用户群。利用移动互联网技术，GIS 实现了从WebGIS向Mobile GIS的转变，使通过手机也可以浏览地理信息。在LBS方面，定位服务实现了从GPS 定位向手机定位的转变，克服了传统GPS定位对卫星信号的要求问题。

从功能上来看 ,移动GIS以下列方式传递地图信息：文本、图像、语音、视频。相对于传统GIS，移动GIS在以下几种应用方向上，更具有显著的优势：

1. 路由选择

路由选择是移动终端基于GIS最流行的实现方式。移动用户可以向制图系统表明他的起点，目的地和通过位置。其中一些输入也能从GPS直接获取。GPS设备与机动终端同时使用,并利用快速的转播速度,移动用户也能获得路线指导。移动终端可采用文本或语音方式从服务器上获取详尽的路由结果，并将结果突出显示在移动终端上。

1. 查询

根据用户所给的位置信息，通过搜索地图数据库查询用户所需地点。用户可以在一定的半径之内查询有关银行、宾馆、医院等公用设施的信息。如果移动终 端上配有GPS，用户就可以从GPS上获得自身的位置。

1. 数据采集

当外出采集数据的时候，由于没带大的移动存储设备, 而且所采集的空间数据又非常大,此时就可以通过移动GIS把采集到的空间数据直接传到空间数据服务器上。同时, 服务器又可以通过处理将有用的数据再传到移动 GIS终端上,以满足外出采集数据所必需的基本数据内容。

从应用类型来看，移动GIS主要有两种应用类型：

查询类应用模式，核心GIS数据一经发布，则不能有应用客户端修改。应用程序以只读的方式对核心GIS数据进行访问。常见的移动GIS应用，如车载导航，景区导游机；移动平台上的谷歌地图、百度地图等，均属于此类应用。

更改类应用，即应用程序在运行时可能会更改GIS数据，例如对于图层的增加，对于地理对象的修改。此类应用主要是应对数据采集的需求。

从与网络的关系来看，移动GIS主要有两种联网模式：

1. 离线模式

离线模式是指移动空间位置信息的提供与获取不需要依赖网络，所有服务都是通过移动终端自己完成，该模式要求移动终端具有独自提供与获取移动空间信息服务的能力。所以移动终端需要移动GIS的支持，具有存储、显示、查询和检索空问信息的能力，甚至一些简单的空间分析能力。该模式下的移动空问信息服务的建立主要是空间数据和服务功能的准备。空问数据可以通过ActiveSync从桌面计算机中获取，或者是通过Flash Memory直接存储：服务功能主要是针对专项需求的专业服务。移动空问信息服务的空间数据通过移动GIS在移动终端本地存储和管理，服务功能也是通过移动GIS在本地完成。由于这种模式下的移动空问信息服务全部在移动终端完成，所以系统的响应速度快，但是在对实时更新等需求兼容性不好。

1. 在线模式

移动GIS的在线体系，其建立过程与服务方式都与离线体系不同。按其服务方式的不同，又可分为两种模式：“有线下载，无线服务”模式和无线网络模式。

有线网络模式与离线模式的主要区别就是在建立移动空间信息服务过程中对空间数据的准备方式不同，前者需要移动终端设备与服务中心的服务器通过有线网络建立连接，下载所需的空间数据并存储到本地。

该模式需要提供从网络下载空间数据的工具，而且下载的空间数据需要与系统具有很好的兼容性和集成性。因为提供的服务都是在移动终端本地完成，所以同离线模式一样，具有数据访问速度快，系统响应及时的优点，但比离线模式更有利的一点就是：该模式只要在Intemet的任一结点上便可以更新空间数据和相关专题信息，但是这种数据和信息的更新不是实时的，要受到时间和地点的限制。

无线网络模式就是指移动终端设备通过无线通信技术与无线互联网技术与服务器建立连接，实时地获取所需的空间数据和专题信息数据。在这种模式下，所有的服务、空间数据及专题信息数据都放在互联网上，服务的请求和响应都需要无线网络的支持。网络的资源是丰富的，提供的服务也比较全面、多样化，该模式能让用户获取最新的数据和最优的服务，而且所有的移动终端都连接成一个有机整体，为数据的动念更新提供基础，为人们在同常生活中地紧密联系提供了方便。如一个调度系统，指挥中心需要清楚所有移动终端的具体位置，这样才能实时地、高效地进行调度处理。

## 空间数据结构

设计和使用GIS的首要工作就是根据系统的功能获取所需的空间数据。

在地理信息系统中，描述地理位置及其空间关系的空间特征数据是地理信息系统所特有的数据类型，主要以矢量数据结果和栅格数据结构两种形式存储。

### 栅格数据

栅格数据主要是按网络单元的行列排列、具有不同灰度、颜色等的阵列数据。每一个单元（象素）的位置由它的行列号定义，所表示的实体位置隐含在栅格行列位置中，数据组织中的每个数据表示地物或现象的非几何属性或指向其属性的指针。

栅格图又叫位图、点阵图、像素图等，是基于行列组成的方格数据模型,通过记录每个小格子中像素值来描述一副图像,常见格式如.bmp,.jpeg等格式均是栅格图。

目前互联网地图，主要是利用四叉树算法，将栅格图按照地理关系组织成为一个具有空间索引关系的以影响金字塔方式存储的栅格数据集。

### 矢量数据

矢量数据是[计算机](http://baike.baidu.com/view/3314.htm" \t "_blank)中以矢量结构[存贮](http://baike.baidu.com/view/7186087.htm" \t "_blank)的内部数据，一般通过记录坐标的方式来尽可能将地理实体的空间位置表现的准确无误，显示的图形一般分为矢量图和位图。在矢量数据结构中，点数据可直接用[坐标](http://baike.baidu.com/view/161356.htm" \t "_blank)值描述；线数据可用[均匀](http://baike.baidu.com/view/327848.htm" \t "_blank)或不均匀间隔的顺序坐标链来描述；面状数据（或多边形数据）可用边界线来描述。

矢量图也称为[面向对象](http://baike.baidu.com/view/125370.htm" \t "_blank)的图像或绘图图像，是计算机图形学中用点、[直线](http://baike.baidu.com/view/15102.htm" \t "_blank)或者多边形等基于[数学](http://baike.baidu.com/view/1284.htm" \t "_blank)方程的几何图元表示图像。[矢量图](http://baike.baidu.com/view/138039.htm" \t "_blank)形最大的优点是无论放大、缩小或旋转等不会失真；最大的缺点是难以表现色彩层次丰富的逼真图像效果。

## 地理信息可视化

### 地理信息可视化的目的

现代信息可视化的主要目的是支持信息的访问和探索活动，探索活动伴随可视化过程一起进行，地图被看成关于地理空间数据的界面，同时也保留它作为表达形式的传统角色，而制图者与用图者则合二为一。

### 地理信息可视化的方式方法

GIS系统提供了对地理信息的可视化处理。

二维地理信息可视化目前是空间信息可视化的主流方法。

通常意义上的 GIS，其可视化采用的方法即是二维图形图像学方法，其可视化方法主要是对传统地图学以及制图学可视化方法的数字化实现。

二维空间信息可视化常见的数据结构有矢量数据结构和栅格数据结构。

矢量数据可视化是一种符号化方法，应使用基本图形元素（如点、线、面）工具及符号库中的视觉变量。矢量数据可视化基于GIS组件提供对于图形的实时绘制功能。

栅格数据结构主要与影像数据可视化关联。在这个过程中提高影像的方法，主要有空间索引和影响金字塔等。影像数据的可视化，在地图服务器端已经完成切片、组织和预制，需要GIS组件提供对于这些切片的访问和渲染功能。

### 地图数据来源分析

移动GIS的地图数据来源分为三种：离线地图、互联网地图和OGC服务。

离线地图是指存储在客户端本地中的地图文件，其中常见的地图格式有：

Shapefile, Mapfile, KML, OSM, MBTiles几种。

为了实现地理信息系统的开放性和互操作,OGC制定了一系列的地理数据和操作软件开发规范，其中基于XML的WMS是地图服务标准之一。

随之互联网、3S技术的普及，互联网提供商提供了大量的地图服务，比较典型的有GoogleMaps, MapBox online, BingMaps, 百度Map，Arcgis online等。同时一些企业也会在公司内部建立专属地图服务，通过内部网络或者互联网，比较典型的地图服务端是ArcGIS Sever。

## 课题GIS功能需求

综上所述，本课题希望在IOS平台提供一个能够具备一定的数据解析能力、电子地图基本操作功能，以及相对易于使用的标绘功能的GIS组件。

### 空间数据结构定义

移动GIS支持栅格类型和矢量类型的数据结构定义。

在结构标准定义之上，GIS组件进一步对于矢量数据元素间的关系和栅格数据间的关系进一步加以定义，如表 2.4‑1 GIS数据结构定义能力要求。

表 ‑1 GIS数据结构定义能力要求

|  |  |
| --- | --- |
| **需求** | **描述** |
| 栅格数据结构定义 | 提供栅格数据结构和具有关联的栅格数据集合结构定义。 |
| 矢量数据结构定义 | 提供矢量数据结构和具有拓扑关系的矢量数据集合结构定义。 |

### 空间数据处理

GIS系统以对空间数据的读取和编辑功能为基础，针对不同领域的需求进行扩展。在实际使用过程中，人们首先要在特定坐标系中对空间数据进行定位，继而根据实际需求完成空间数据的分析和处理工作。

随着移动设备的普及，人们可以简单有效的获取实时GPS数据。针对这种数据的进一步应用，加速了LBS服务在移动GIS上的普及。

GIS组件提供的数据处理功能，如表 2.4‑2 GIS空间数据处理能力要求

表 ‑2 GIS空间数据处理能力要求

|  |  |
| --- | --- |
| **需求** | **描述** |
| 空间数据的读取 | 提供针对网络地图，第三方本地地图文件和GIS模块提供的数据存储文件的读取功能。 |
| 空间数据的编辑 | 提供将经过处理的空间数据的存储功能。 |
| LBS服务 | 提供对于百度地图的LBS服务访问的接口，及LBS服务提供者定义，可以替换为其他服务提供商。 |
| 坐标系转换 | 提供针对常用坐标系间的转换，以对百度自定义坐标系的转换功能。 |

### 地理数据可视化

GIS系统提供了对地理信息的可视化处理。GIS组件不仅提供了渲染数据的功能，同时也提供了对于某些地理数据的编辑功能。具体描述如表 2.4‑3 GIS地图数据可视化的能力要求

表 ‑3 GIS地图数据可视化的能力要求

|  |  |
| --- | --- |
| **需求** | **描述** |
| 切片地图绘制 | 提供获得互联网地图提供的地图切片的算法。 |
| 基础矢量图形绘制 | 提供将矢量数据的地理坐标系转平面坐标系和绘制功能。 |
| 常用统计控件 | 提供以矢量图形方式展现的常用统计控件。 |
| 特定符号标绘 | 提供以矢量图形组合的方式提供特定的符号显示和编辑功能。 |

### 图层管理

GIS组件通过图层的方式容纳地理信息的图形显示元素，并且将这些图层置于同一个屏幕坐标系下，共享用户对于地图的平移和缩放操作。具体描述如表 2.4‑4 GIS图层管理能力需求

表 ‑4 GIS图层管理能力需求

|  |  |
| --- | --- |
| **需求** | **描述** |
| 图层容器 | 加载相应类型的图层，提供统一的屏幕坐标系，以及平移缩放功能 |
| 图层定义 | 定义矢量图层，切片地图图层和专题地图图层，同时在各种图层上提供平移缩放接口 |
| 图层编辑 | 支持新建、保存、排序和删除图层功能。 |

## 本章小结

本章对当前GIS系统具有的功能进行了简述，简要分析了GIS数据存储的方式，移动GIS应用本身的特征需求，得出了本组件的目标功能需求。本组件的目标功能需求可以分4部分：空间数据结构定义、空间数据处理、地理数据可视化、图层管理 。从这4部分来看：空间数据定义、空间数据处理、地理数据可视化不同的系统中都会出现，前3部分可做成通用的组件和控件复用，有利于组件功能的扩展和复用，使得二次开发人员可以专注于业务领域，在各自的领域越做越深。

1. IOS.GIS.Common.Tool 功能包概要设计

## 当今GIS系统应用的架构设计

一个GIS系统往往需要对GIS数据、基本空间处理功能与各种应用模型进行整合，而系统整合方案在很大程度上决定了系统的适用性。

本课题对当今国内流行的GIS应用方案进行了研究以供参考。

### GIS解决方案

成熟的商业化GIS系统是WebGIS和传统桌面GIS应用的常见解决方案，即通过商业GIS开发系统提供的可视化的开发环境开发环境，基于GIS组件提供的功能进行二次开发，进行快速的二次开发。方案如图：

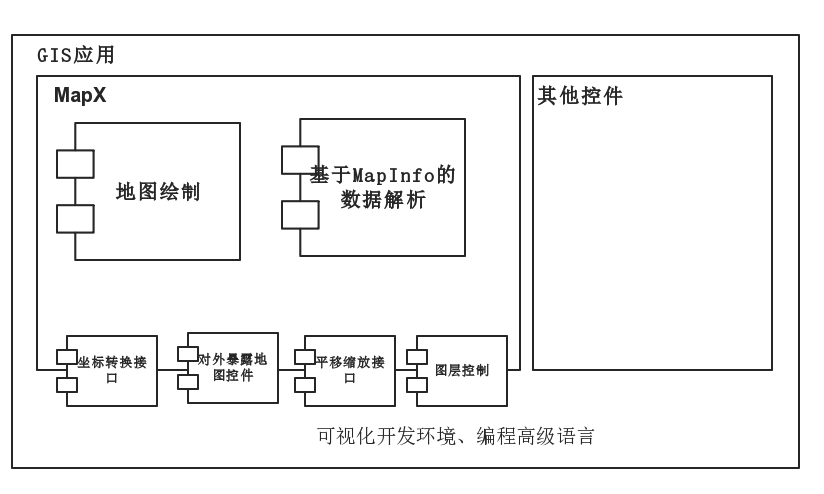


图 ‑1 商业GIS组件开发方案

GIS应用中关于地理信息部分的业务是以商业GIS为基础进行开发的，商业GIS一般涵盖了基本的GIS应用功能，这些功能以组件形式提供给开发人员，一般功能组件如表：

表 ‑1 商业组件常见功能列表

|  |  |
| --- | --- |
| **功能** | **说明** |
| 坐标转换接口 | 坐标系间的转换，包括地理坐标间的转换、地图坐标和屏幕坐标的转换。 |
| 对外暴露地图控件 | 提供实例化地图控件的接口 |
| 平移缩放接口 | 根据需求应用设备输入进行对地图的放大、缩小、漫游操作的操作接口。 |
| 图层控制 | 对图层进行管理，将业务逻辑相关的地图元素放到指定的图层上，方便管理 |

不容否认，商业GIS组件可以执行高效率、流畅的系统整合。但是在一些特殊的领域，会有一些特殊的需求，与一般的商业GIS应用有不同之处。 传统商业GIS组件针对移动设备，尤其是触屏为操作界面的移动手机等设备的兼容不好，对于GPS等数据没有内建支持。这些功能的缺乏桎梏了二次开发人员在业务领域的开拓能力。例如从位置服务来看，移动设备往往通过自身携带的GPS功能就可以完成地图定位和导航服务

### 现有移动GIS的组件架构设计

参考了现在流行的百度、MapABC、谷歌等移动GIS IOS SDK。现在的移动GIS SDK功能主要集中在加载固定的地图，LBS服务和在地图上绘制简单的标志。

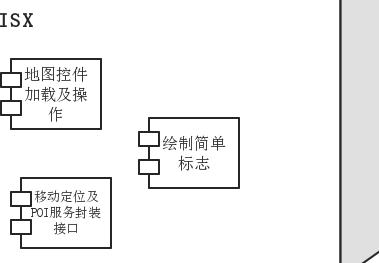
相比较后，可以发现移动GIS提供的地理信息绘制功能很简单，标绘和地图加载两个基本功能点不足以支持复杂的商业开发。

图 ‑2 移动GIS架构图

## IOS.GIS.Common.Tool功能包

### 总体设计

在进行总体设计的时候，参考了上述的GIS应用架构设计，依照本课题提出的移动GIS需求，提供了一套名为IOS.GIS.Common.Tool的移动GIS SDK设计方案。

在设计过程中，参照了“高内聚，低耦合”的模块设计原则。高内聚指每模块应该尽可能集中做好一件事，低耦合指模块间的依赖尽可能的少，避免“牵一发而动全身”的情况。故而，本课题在设计IOS.GIS.Common.Tool时，将相同类别的需求抽象出来，归类，从而确定提供的功能，具体实现可以由子类完成。

IOS.GIS.Common.Tool由4个子系统组成，分别为矢量数据系统、栅格数据系统、位置服务系统、地图UI。如图 3.2‑1 IOS.GIS.Common.Tools 子系统关系图

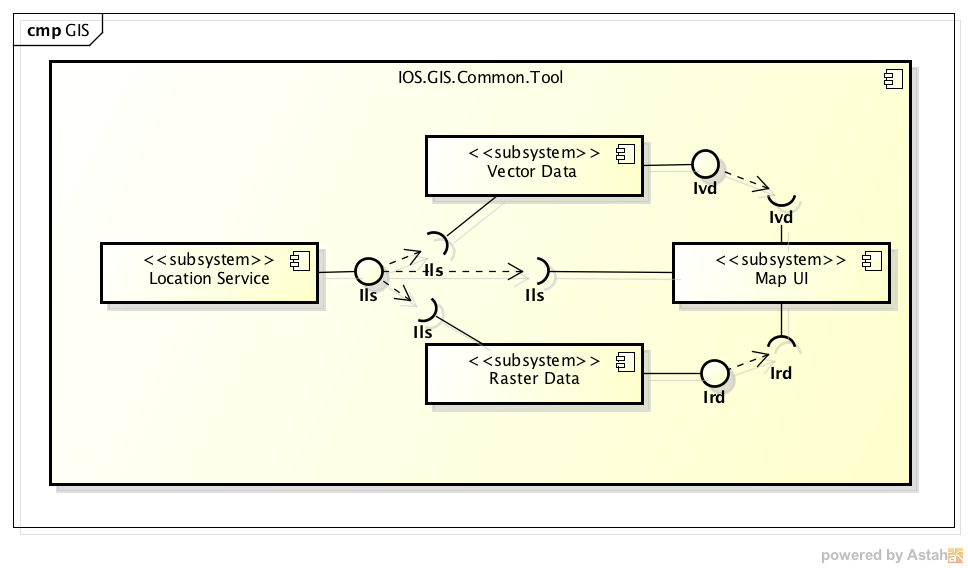


图 ‑1 IOS.GIS.Common.Tools 子系统关系图

### 位置服务子系统

位置服务子系统主要是通过对地理坐标的处理，提供了多种坐标系间的转换功能、不同地点间相对位置的判定以及基于位置的系统外联功能。

位置服务子系统主要由三个模块组成：地理坐标模块和移动定位服务模块。

#### 地理坐标模块

地理坐标模块主要是作为IOS.GIS.Common.Tools的基础服务模块，通过对于地理坐标的处理，将空间坐标投影为平面后，进一步判定地图上两个点间不同的方位角和距离，详细划分如表 3.2‑1 地理坐标模块功能列表

表 ‑1 地理坐标模块功能列表

|  |  |
| --- | --- |
| **功能** | **说明** |
| 地理坐标转换接口 | 提供将不同地理坐标转换为WGS-84 |
| 地理坐标转投影坐标接口 | 提供将不同空间坐标系下的地理坐标转换为Web Mercator投影坐标系下的平面坐标 |
| 相对位置关系判定接口 | 提供判断地图上两个点的方位角和距离判断。 |

#### 移动定位服务模块

移动定位服务模块主要为IOS.GIS.Common.Tools获得设备当前位置、周边信息查询、签到等功能。具体划分如表 3.2‑2 移动定位服务模块功能列表

表 ‑2 移动定位服务模块功能列表

|  |  |
| --- | --- |
| **功能** | **说明** |
| 移动设备定位接口 | 提供获取iOS设备定位信息功能 |
| 周边信息获取接口 | 提供通过互联网LBS服务商获得相应 |
| 位置签到接口 | 提供将iOS设备当前位置及附加信息记录到服务器的功能 |

### 矢量数据子系统

IOS.GIS.Common.Tools SDK通过矢量数据子系统，可以提供对矢量数据结构、拓扑关系的判定，以及对数据持久化、数据可视化的支持。

矢量数据子系统由3个模块组成：矢量数据结构模块、矢量结构数据持久化模块、矢量数据可视化模块。

#### 矢量数据结构模块

矢量数据结构对矢量数据模型进行数据的组织;它通过记录实体坐标及其关系,尽可能精确地表示点、线、多边形等地理实体,坐标空间设为连续,允许任意位置、长度和面积的精确定义。矢量数据结构直接以几何空间坐标为基础,记录取样点坐标。

同时通过符号数据将空间对象的属性信息和空间信息结合。

矢量数据结构模块主要提供对于两种矢量数据类型的描述，符号数据和几何体数据，以及通过描述符号数据的层次挂席和几何体的拓扑关系来展现矢量数据的集合信息。具体划分如下表 3.2‑3 矢量数据模块功能列表：

表 ‑3 矢量数据模块功能列表

|  |  |
| --- | --- |
| **功能** | **说明** |
| 基于对象的符号数据结构接口 | 提供对于对于符号数据的通用接口、关联图片符号数据结构、关联几何体符号数据结构描述。 |
| 基于对象的几何体数据结构接口 | 提供对于几何体的数据结构描述。 |
| 几何体的拓扑关系接口 | 提供对于几何体间的拓扑关系类型定义和拓扑关系判定的算法 |
| 符号数据的层次关系接口 | 提供符号数据的关联层次的数据结构描述 |

#### 矢量数据持久化模块

矢量数据持久化模块实现了矢量数据的存储和读取，具体划分如下表 3.2‑4 矢量数据持久化模块功能列表

表 ‑4 矢量数据持久化模块功能列表

|  |  |
| --- | --- |
| **功能** | **说明** |
| 符号数据存储接口 | 提供对于对于符号数据及层次关系的持久化功能 |
| 符号库接口 | 提供了对于预定义符号库的读取，和对于注入自定义符号库的支持。 |
| 符号数据接口 | 提供了对于预定义符号类型的读取，支持注入自定义符号类型，同时也提供了对于自定义符号数据源的支持。 |
| 几何体持久化接口 | 提供了对于基本几何元素、拓扑关系矩阵、关联表的持久化支持。 |

#### 矢量数据可视化模块

矢量数据可视化模块提供了对于几何体和符号数据可视化，以及将矢量数据分图层展示的支持。详细划分如表 3.2‑5 矢量数据可视化模块

表 ‑5 矢量数据可视化模块

|  |  |
| --- | --- |
| **功能** | **说明** |
| 符号数据可视化接口 | 提供对于符号数据的可视化、定位、以及常用统计控件和对于符号数据通过UI添加修改删除的功能，支持自定义符号可视化功能注入。 |
| 几何体可视化接口 | 提供对于几何体的可视化功能，和几何体的自定义编辑功能，支持注入自定义几何体可视化功能。 |
| 矢量图层可视化接口 | 提供将展示同层矢量数据的图层，这些图层提供对于矢量数据的定位，显示可视化结果以及平移缩放功能，以及将重建已经保存的图层，和创建新图层的功能。 |

### 栅格数据子系统

通过栅格数据子系统中的栅格数据结构模块、栅格数据源模块和栅格数据可视化模块，提供了对于栅格数据结构的通用支持、数据信息的加载和可视化功能的支持。

#### 栅格数据结构模块

IOS.GIS.Common.Tools SDK支持的栅格数据主要为以影像金字塔形式组织的瓦片地图系统。

瓦拍金字塔模型通过由瓦片单元，邻接瓦片数组和四叉树模型三部分组成，实现了对于瓦片地图单位数据和拓扑关系的定义。

详细划分如表 3.2‑6 栅格数据结构功能列表

表 ‑6 栅格数据结构功能列表

|  |  |
| --- | --- |
| **功能** | **说明** |
| 瓦片单元接口 | 提供了对于最小地图瓦片单元的数据结构定义。 |
| 邻接瓦片数组接口 | 提供了对于按照邻接关系组织的同分辨率层瓦片数组数据结构定义。 |
| 四叉树接口 | 提供了对不同分辨率层次上下层间的父子关系定义，以四叉树结构形式呈现。 |

#### 栅格数据源模块

IOS.GIS.Common.Tools SDK扩展自WMTS 多种瓦片地图数据源。

具体功能划分如表 3.2‑7 栅格数据源功能列表

表 ‑7 栅格数据源功能列表

|  |  |
| --- | --- |
| **功能** | **说明** |
| 数据源提供器接口 | 支持注入自定义数据源和提取数据源解析器 |
| 数据源解析器接口 | 提供了提供了数据源解析器的定义，和基于WMTS的Bing地图，ArcGIS地图的解析，以及利用静态地图方式通过瓦片金字塔方式组织的百度地图数据源解析。 |

#### 栅格数据可视化模块

IOS.GIS.Common.Tools SDK支持基于WMTS瓦片地图的可视化。

具体功能划分如表 3.2‑8 栅格数据可视化功能列表

表 ‑8 栅格数据可视化功能列表

|  |  |
| --- | --- |
| **功能** | **说明** |
| 瓦片地图图片接口 | 提供图片的下载和缓存功能。 |
| 瓦片地图图层接口 | 提供对于当前比例尺下瓦片地图的定位和显示，以及不同层级间的瓦片地图切换功能。 |

### 地图UI子系统

地图UI子系统，为IOS.GIS.Common.Tools SDK提供了对于可视化图层的管理和用户对于地图界面漫游和缩放的支持。主要分为两大模块：图层定义模块和地图浏览模块。

#### 地图浏览模块

表 ‑9 地图浏览模块功能列表

|  |  |
| --- | --- |
| **功能** | **说明** |
| 地图容器接口 | 提供加载图层到地图中，以及排序和编辑功能对于地图界面的平移和缩放操作，以及对于这种操作的传导机制。 |
| 屏幕坐标转换接口 | 平面坐标、地理坐标和屏幕坐标转换。 |

#### 图层定义模块

表 ‑10 图层定义模块功能列表

|  |  |
| --- | --- |
| **功能** | **说明** |
| 图层定义接口 | 提供图层定义和图层读取器的定义。 |
| 屏幕坐标转换定义接口 | 提供屏幕坐标转换定义和坐标转换供应器定义。 |

### 设计的可裁剪性

IOS.GIS.Common.Tools SDK，很重要的一个设计思路就是希望不同侧重点的用户可以基于本SDK进行剪裁，找到适合自己业务领域的版本进行二次开发。

故而本SDK以一组IOS静态链接库(.a)的形式发布，在使用时候可以通过引用不同的静态库组合实现对于IOS.GIS.Common.Tools所提供的功能裁减和重组。

本文将以对于位置服务子系统的裁剪和矢量数据相关系统的裁剪为例，来介绍本SDK的可裁剪性。

#### 位置服务子系统的裁剪

根据具体需求将地理坐标模块单独裁剪出来，功能及调用方式如表 3.2‑11 地理坐标模块单独裁剪

表 ‑11 地理坐标模块单独裁剪

|  |  |
| --- | --- |
| **二次开发** | **具体说明** |
| 功能描述 | 提供给二次开发人员对GIS地理坐标转换的功能 |
| 发布形式 | 地理坐标模块以IOS静态链接(.a)的形式发布。 |
| 调用形式 | 通过引用该静态链接库的方式，获得对于对于 |

此模块属于底层服务模块，可以供不熟悉GIS地理换算的二次开发人员调用，来实现不同地理坐标系统间的转换，或者对于两地位置的相对判定。

应用设计图如图图 3.2‑2 地理坐标模块裁剪应用设计图

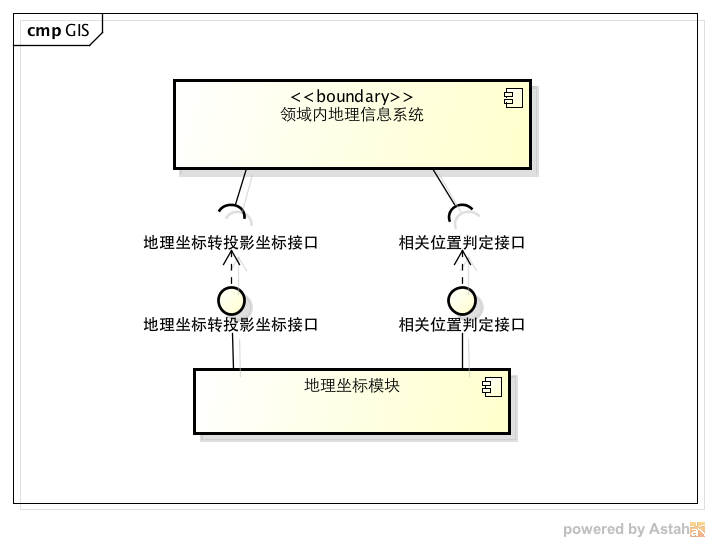


图 ‑2 地理坐标模块裁剪应用设计图

#### 矢量数据相关GIS模块的裁剪

根据实际需求将地理坐标模块、图层定义模块、矢量数据结构模块 、矢量数据可视化模块及地图浏览模块单独裁减出来，构成一个基本的移动GIS模块。

表 ‑12 矢量数据相关GIS模块裁剪

|  |  |
| --- | --- |
| **二次开发** | **具体说明** |
| 功能描述 | 提供给二次开发人员基本的移动GIS组件功能 |
| 发布形式 | 发布一组IOS静态链接(.a) 。 |
| 调用形式 | 1. 初始化地图控件：设计地图数据源、初始位置和默认比例尺。 2. 新建矢量标绘图层到地图控件上，共享地图的平移、缩放等功能。 3. 将数据管理模块解析的数据源制定给图层，绘制矢量图形 |

应用设计图如下图 3.2‑3 矢量数据相关GIS模块裁剪使用

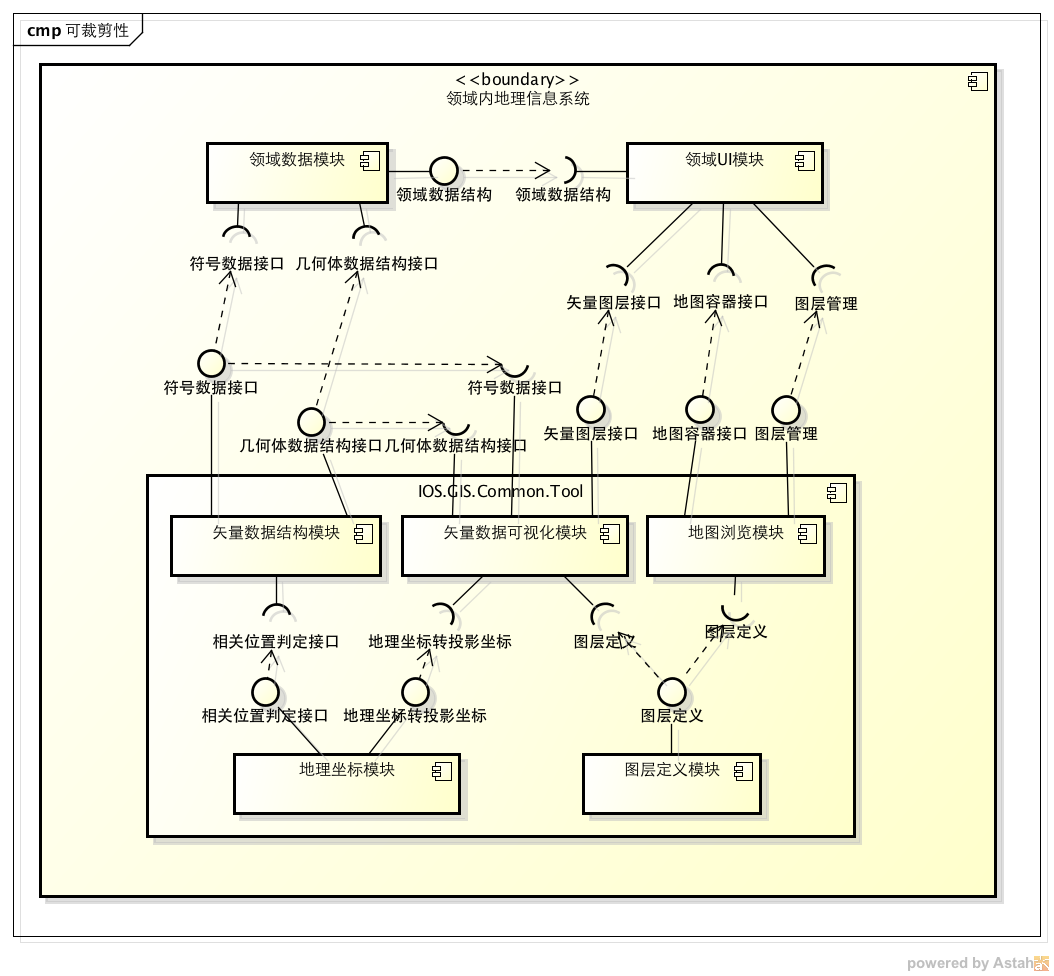


图 ‑3 矢量数据相关GIS模块裁剪使用

### 设计的扩展性

#### 扩展符号数据类型和符号数据构造器

矢量数据子系统提供了一个很重要的功能就是将持久化的数据转换为标准形式的符号变量。

符号变量是对现实中的空间数据的抽象，处于不同的领域或者相同领域的不同业务上下文下，开发人员对于符号变量的抽象方式也不尽相同。

因而在设计矢量数据持久化模块时，就考虑到了通过允许第三方开发人员对符号数据构造器进行扩展的方式，来增加IOS.GIS.Common.Tools SDK的业务扩展能力，同时也可以使第三方开发人员不再关心现有的符号数据构造过程和详细的构造流程。

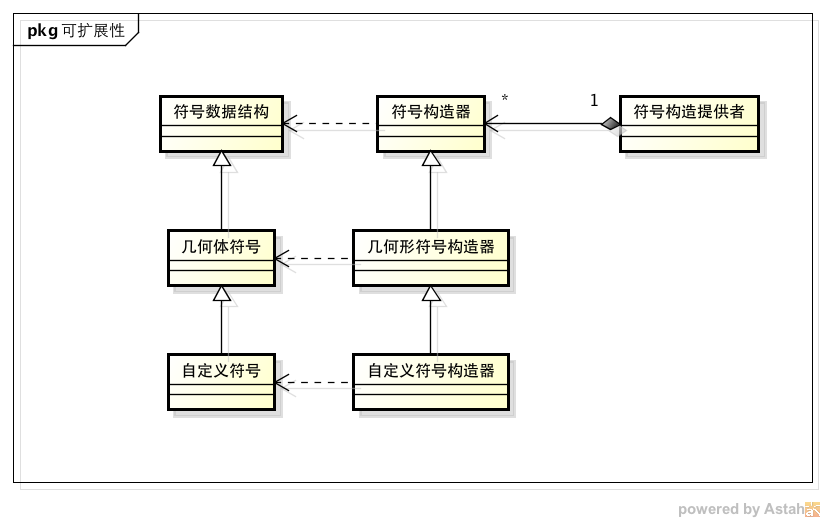


图 ‑4 符号构造器及符号扩展

## 本章小结

本章描述了IOS.GIS.Common.Tool SDK的总体设计。

首先，通过学习当前流行的GIS应用组件设计方案和移动GIS的SDK，找到了不同方案的共性和不足之处。

其次，根据本课题实际要解决的问题，以及上述学习获得经验，依据“高内聚，低耦合”的原则进行了设计，提出了IOS.GIS.Common.Tool整体设计和各模块的规划。

最后讨论了IOS.GIS.Common.Tool模块设计在软件扩展性方法的优势。IOS.GIS.Common.Tool 允许针对不同侧重点的客户进行功能裁减，找到合适的部分进行二次开发。同时高可扩展性的设计方案，也有利于IOS.GIS.Common.Tool在今后的应用中通过集成更多现有的服务来提高IOS.GIS.Common.Tool的业务功能。

1. 地理坐标模块详细设计
2. 移动定位服务模块详细设计
3. 栅格数据结构模块详细设计

## 瓦片地图技术

### 瓦片地图技术应用场景

地图是GIS操作的入口，请求、载入和渲染地图的效率直接影响了使用者的体验和工作效率。

在分布式网络环境下，建立在量普通PC服务器通过Internet互联，对外作为一个整体提供存储服务为基础上的大规模分布式存储系统，带来了低成本强可扩展性的高性能存储方案。因此空间数据的组织模型一定要和在分布式网络环境中进行数据存储的模式相适应。

目前以谷歌和百度为代表的互联网地图服务商和以ArcGis为代表的传统地图服务器提供方都已经通过采用瓦片地图的方式组织地图数据，充分利用了分布式存储和服务能力。

同时传统GIS通过实时请求来获得实时的地图数据，对于提供地图数据的系统而言，随着地图消费终端的增多，网络传输和运输压力也随之快速增长。而以瓦片为单位组织空间数据就可以获得局部访问直接、缓存替换简单、分布存储快捷、优化策略多样等诸多优势。

### 瓦片地图技术原理概述

瓦片地图技术是指将配置好的确定地理空间范围的地图，依照一定的数学规则，采用预先配置好的多级显示比例尺（即将连续地图通过划分比例尺的形式转换为多级的离散地图），由服务器端提前把地图图片切割为统一规格的瓦片矩阵（128\*128或256\*256），并遵循固有的命名规则和缩放比例，存储为不同服务器的树状结构的文件系统，并建立瓦片存储路径和地图坐标的映射关系。

## 瓦片金字塔模型研究

影像金字塔是一种栅格数据的组织方式，指在统一的空间参考下，分层表示不同的分辨率的栅格数据。

### 瓦片金字塔的分辨率

地图的分辨率是指在一定的比例尺下显示地图时，屏幕的一个像素点所代表的空间长度。

每一层分辨率相同，不同层的分辨率按照从上到下，以几何关系缩小，所对应的比例尺成等倍数依次放大，通常采用4的倍数缩小，这种倍数关系被称为缩放级别。

本文中为了论述方便，先约定金字塔模型的最顶层为第1层，最下层为第N层。

不同层的分辨率从上往下成几何倍数缩小，如第一层的分辨率为R，则第N层的分辨率为R/4N-1；不同层的切片个数从上到下按照几何关系增加，如果缩放倍数为4的话，假设第一层瓦片数量为1，则第N层瓦片数量为4（N-1）。

即第1层只有一个覆盖整个地图范围的瓦片；在第2层时，每一个瓦片只能显示第1层单个瓦片1/4的地图范围了；在第3层时，每一个瓦片只能显示第1层瓦片1/16的地图范围了。单个瓦片显示的地图范围从顶层到底层逐渐变小，而每一层的瓦片集合涵盖的地理范围不变。如图 6.2‑1 瓦片地图分辨率所示

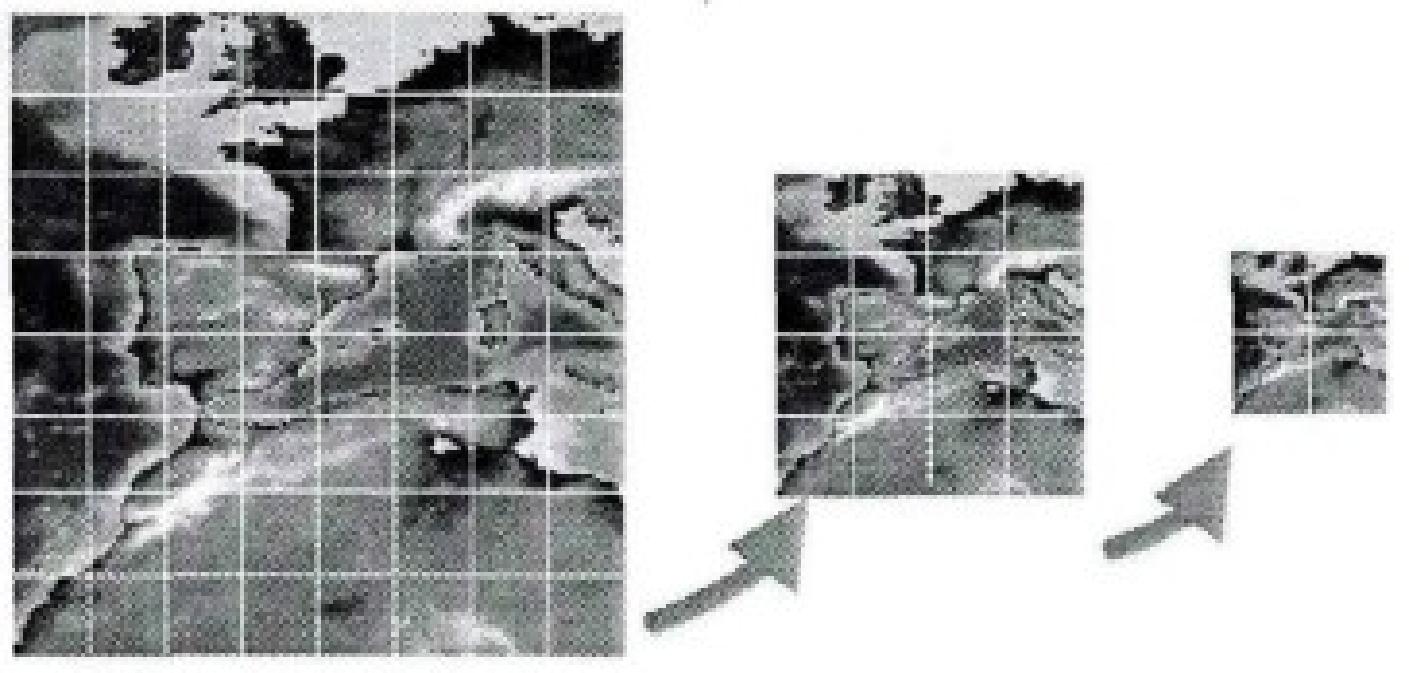


图 ‑1 瓦片地图分辨率

### 瓦片金字塔的构建

预定瓦片地图服务提供的地图缩放级别为N。

第一步将地图的分辨率最低的地图作为模型的最N层，从该层地图左上角为起点依照上述规则进行切割，将地图切割为相同分辨率的矩形瓦片，并指定瓦片的保存格式。各层的瓦片编号从左上角开始，从左至右，从上到下依次进行。如图 6.2‑2 瓦片地图的编号顺序：

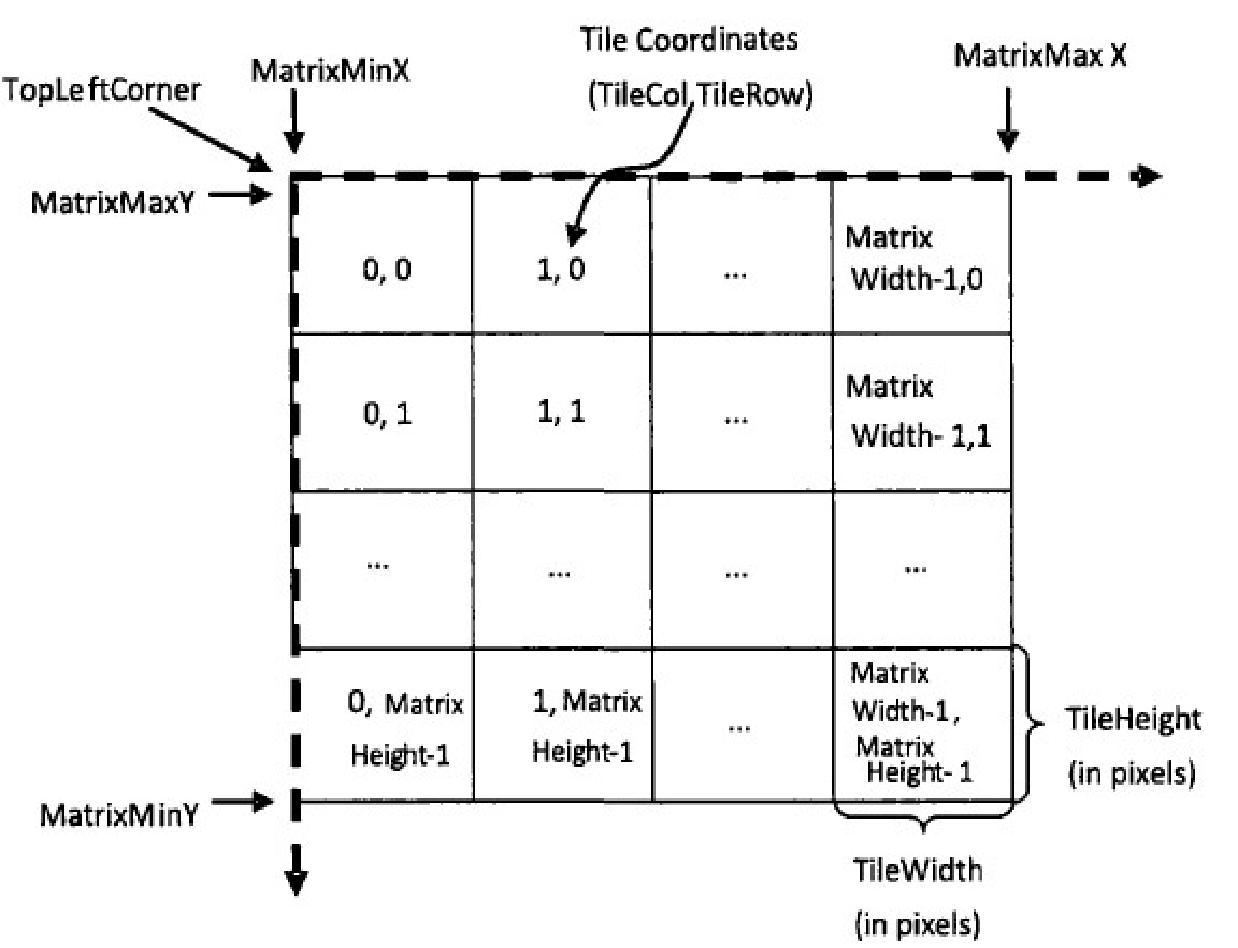


图 ‑2 瓦片地图的编号顺序

然后每4个像素为一个单元合并为1个像素，将最底层的地图按此法生成第N-1层，再次切割，这样就得了第N-1层的瓦片集合；采用相同的方法生成N-2, N-3直到第0层为止。

这样就生成了全部需要的瓦片集合，如图图 6.2‑3 金字塔模型图所示，地图的分辨率从最顶层到最底层依次降低，每一层的瓦片数量构成一个公比为4的等比数列，从下至上依次减少。

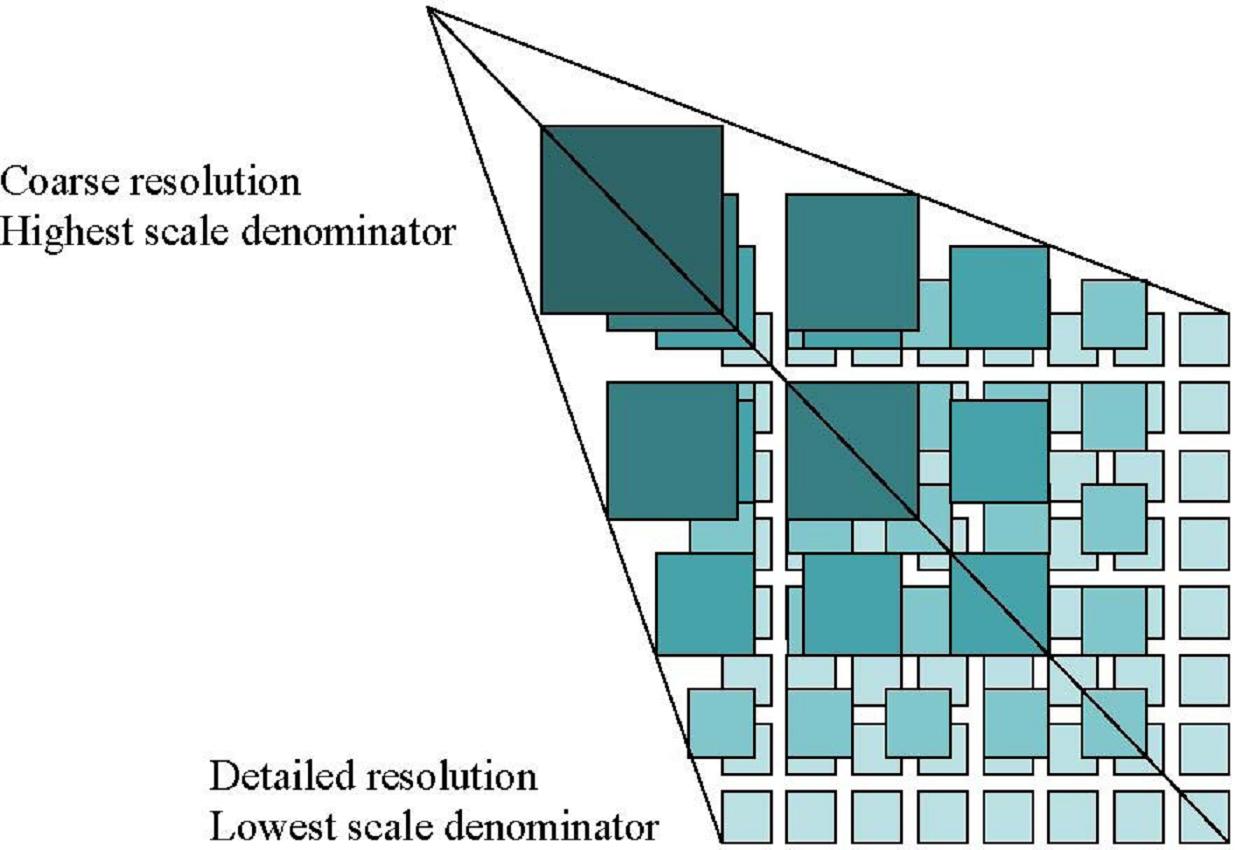


图 ‑3 金字塔模型图

## 瓦片地图的定义

制作瓦片地图，既需要方便服务器端的存储和传输，也要考虑到如何在客户端进行拼接，从而达到快说而无缝显示的目的。

因此对于需要对瓦片地图进行定性的描述，供服务器端生成和描述瓦片地图，同时也供客户端读取和处理瓦片地图。

瓦片地图的定义描述了瓦片地图和瓦片地图的拓扑关系。

### 瓦片地图参数

在地图切割前，定义好一组标准的参数，可以确保地图切分过程的一致性和切分后个瓦片地图信息的完整性。

为了满足一般性需求，在一定地理范围内，按照比例尺从小到大的顺序，提供多级别的瓦片矩阵，供用户完成从全局到局部细节的浏览就已经足够。

一个瓦片地图可以通过瓦片地图分辨率、缩放级数、比例尺、瓦片的尺寸规格、瓦片矩阵涵盖的空间范围、空间参考系和图片格式。

#### 缩放级数

按照金字塔模型的构建思想，缩放后的比例尺为最大比例尺乘以缩放级数的2倍，并定义最大比例尺对应的缩放级数为0。

#### 地图分辨率

地图的分辨率（Resolution）是在已知的地理范围内和缩放级数下，切片地图的一个像素点所代表的地理空间长度，单位为米/像素。

在切分地图的时候，首先需要确定一个地图分辨率作为最低分辨率，然后其他各层分辨率在这个地图分辨率的基础上，呈以2为系数的几何增长，直到放大级数满足上限为止。

地图分辨率的换算与所使用的投影方法相关。现在的网络地图一般采用Web墨卡托投影法，web墨卡托投影法各个方向均等扩大的特性，基准纬线处无变形，从基准纬线处向两级逐渐增大。故而地图分辨率的计算公式是一个与缩放级别和纬度相关的公式，如

式 ‑1

其中156543.04是赤道周长，单位为米。

简单而言，在常规精度要求下，可以取地图在赤道上的分辨率。

#### 比例尺

在已知的地理范围内和缩放级数下，屏幕上的线段长度与实地相应线段长度之比，它表示地图图形的缩小程度。

显示设备的DPI提供了屏幕上一个像素单位和距离单位的换算常数。

故而从客户端来看，地图分辨率的计算可以通过地图的当前比例尺和显示设备DPI，以及单位英寸转单位米的换算来确定。

式 ‑2

一般情况下，Meter\_Per\_Inch取0.0254。

#### 瓦片的尺寸规格

瓦片的规格是指瓦片的几何大小，以像素为基本单位。则瓦片的高和宽分为定义为TileHeight、TileWidth；每个瓦片都是所对应的瓦片矩阵的一部分，在瓦片矩阵坐标系下，瓦片在对应的投影坐标系中的的宽、高分别为TileSpanX和TileSpanY。

换算关系如：

式 ‑3

式中 pixcelSpan—每个参考坐标系单位对应的像素单位数

*0.28是默认的一个像素是0.28mm\*0.28mm，是指米转换为参考坐标系对应的长度单位的比例。这里的是1：1投影图像上1像素与当前放缩系数下的地图上像素间的转换关系。*

#### 瓦片的矩阵范围

在给定的放缩级数下，确定地理范围内的投影地图资源的矩形边界被称为瓦片的矩阵范围，瓦片就是对边界范围内的地图进行分割，每个瓦片就是一个分割单位。如图 6.3‑1 瓦片地图的空间阵范围：

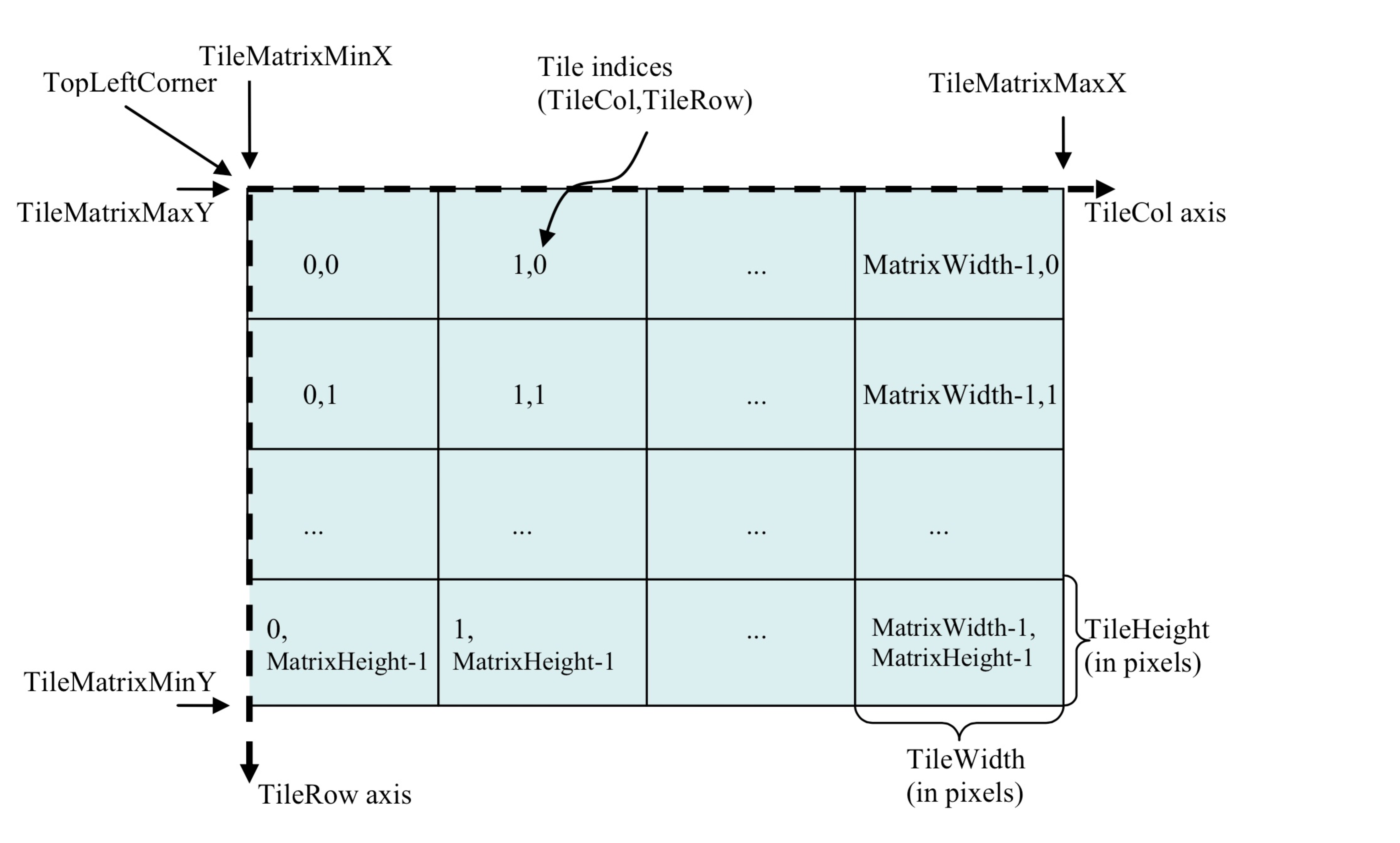


图 ‑1 瓦片地图的空间阵范围

图中的地图边界所对应的矩阵的左上角为(TileMatrixMinX，TitleMatrixMaxY)，而右下角为（TileMatrixMaxX，TitleMatrixMinY）。

式 ‑4

在矩阵中的每一个瓦片，将在矩阵中左上角的坐标作为唯一标示，指明这个瓦片在瓦片矩阵中的相对位置。标示为（TileCol，TileRow）

### 瓦片地图的拓扑关系

如图 6.3‑2瓦片地图的四叉树结构图，最大缩放级数为2，固定地理范围的地图，被切割为3层地图。其中第2层地图，瓦片显示的地理范围最大，但是分辨率最低；第1层的四张瓦片分别对应第2层瓦片的左上部、右上部、左下部和右下部，分辨率是第二层瓦片的2倍；第1层的4张瓦片以从左至右的顺序分别对应第0层的从左至右的4组瓦片。每层间的瓦片数量呈以4为公布的等比数列递减。与图 6.2‑3 金字塔模型图对比，四叉树对于瓦片地图的层级关系描述更加准确。

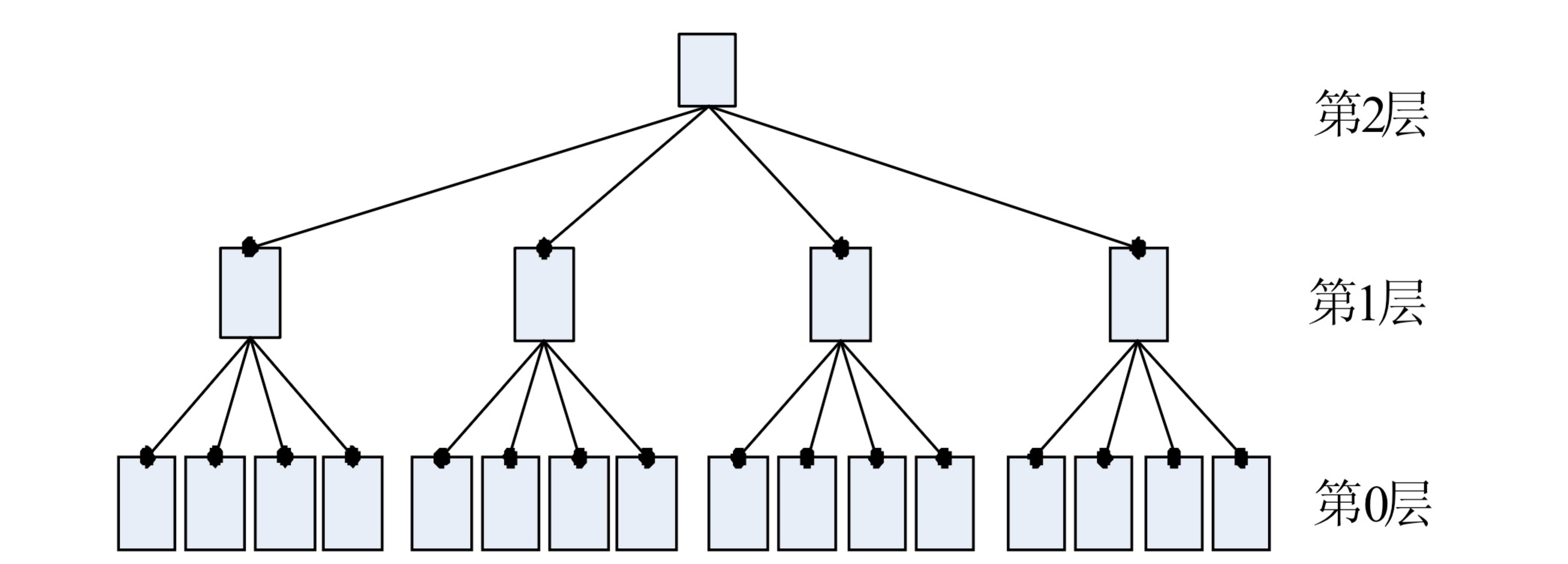


图 ‑2瓦片地图的四叉树结构图

为了便于下面的运算，这个给出下由一个瓦片坐标推导出其相邻瓦片坐标和父瓦片坐标的公式。

如图 6.3‑3 金字塔模型的瓦片拓扑关系例图所示，

在任意L层，L层的1个瓦片最多和8个瓦片相邻。

L层的1个瓦片在L-1层有4个子瓦片，且在L+1层1个瓦片会有一个父瓦片对应。

（，N为放缩级别数量）

推导过程如下：

设定L层任意一个瓦片的坐标为（TileC，TileR），其在整个金字塔中的坐标为（TileC，TileR，N），其中TileR为瓦片所在的行号，TileC为这个瓦片所在的列号，N为缩放级数。

* 该瓦片相交于瓦片4边的瓦片1、3、4、6所对应的座位依次为（TileC+1，TileR，N）、（TileC，TileR-1，N）、（TileC，TileR+1，N），（TileC-1，TileR，N），仅相交于该瓦片4个顶点的0、2、5、7分别为（TileC-1，TileR-1，N），（TileC-1，TileR+1，N）、（TileC+1，TileR-1，N）、（TileC+1，TileR+1，N）。
* 四个子瓦片0、1、2、3的坐标分别为（2\*TileC，2\*TileR，N-1），（2\*TileC+1，2\*TileR，N-1），（2\*TileC，2\*TileR+1，N-1），（2\*TileC+1，2\*TileR+1，N-1）。
* 父瓦片的坐标为（floor（TileC/2）,floor(TileR/2)，N+1）,其中floor为向下取整函数。

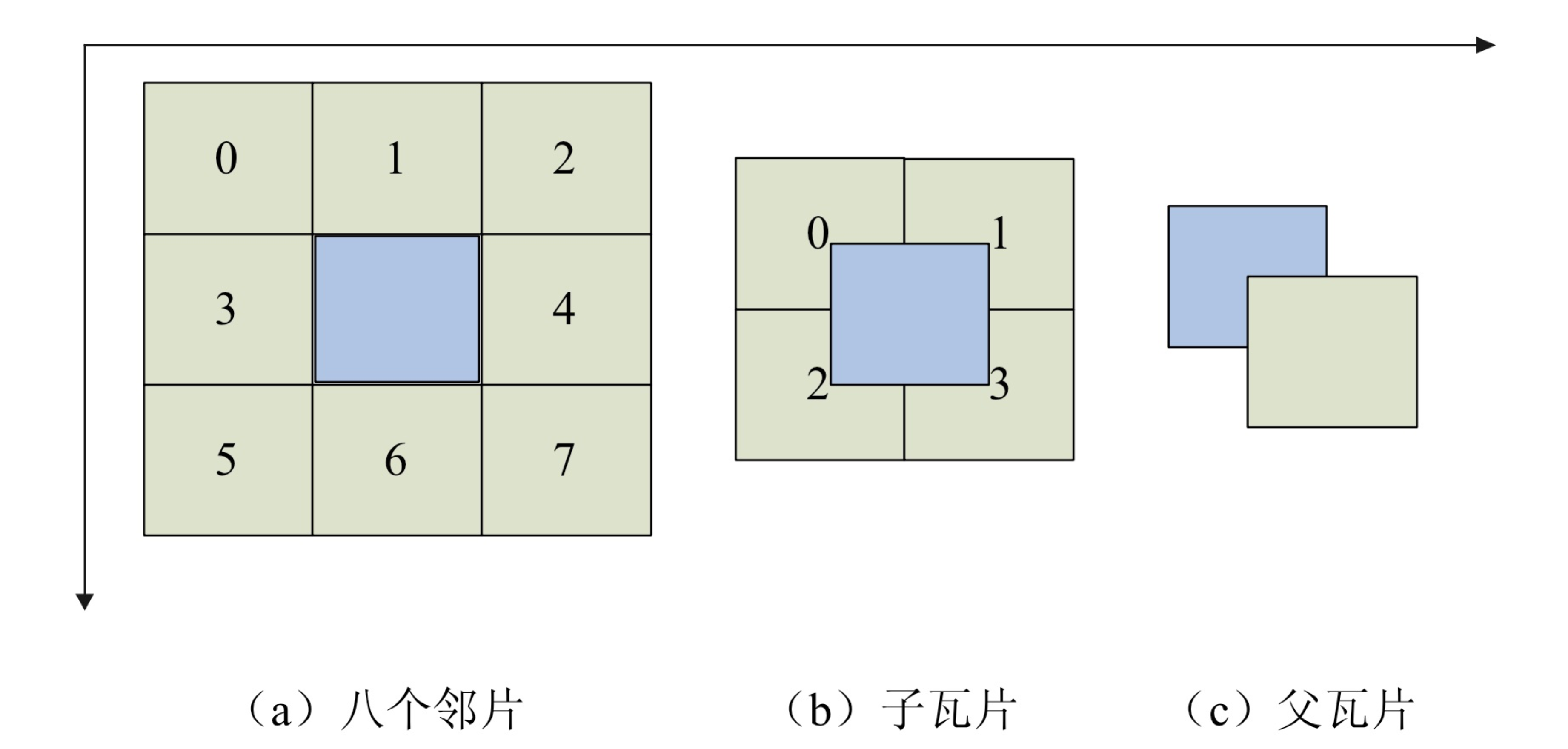


图 ‑3 金字塔模型的瓦片拓扑关系例图

## 瓦片地图的数据结构

为了更好的完成移动设备程序向地图服务器请求瓦片和在显示设备上渲染瓦片，我们需要了解一个瓦片信息有：尺寸规格、涵盖的地理范围、行列号、放缩级数、图片格式，该瓦片所在的瓦片矩阵涵盖的地理范围、尺寸规格、参考坐标系、放缩级数，地图服务器提供的获取瓦片图片的方式，以及瓦片金字塔所对应的放缩倍数和地理学比例尺的对应关系。

其中因为瓦片地图所使用的参考坐标系涵盖了地理坐标系和投影坐标系，且地理坐标系和投影坐标系存在一定的转换关系，所以可以通过地理坐标系标示地理范围。而地图服务器提供的获取瓦片图片的方式和图片格式属于数据源的内容，不在这里定义。

### 瓦片的数据结构

瓦片数据结构（Tile）中定义了瓦片在所对应的矩阵中的行列号，瓦片规格、地理范围和缩放级数。

字段的具体划分见表 6.4‑1 瓦片的数据结构（Tile）字段

表 ‑1 瓦片的数据结构（Tile）字段

|  |  |  |
| --- | --- | --- |
| **名称** | **类型** | **说明** |
| RowIndex | 整型 | 行号 |
| ColumnIndex | 整型 | 列号 |
| Width | 浮点 | 像素为单位的瓦片宽度，一般而言为256 |
| Height | 浮点 | 像素为单位的瓦片高度，一般而言为256 |
| ZoomLevel | 整型 | 放缩级数 |
| MinGl | 自定类型 | 瓦片所对应的最小经纬度 |
| MaxGl | 自定类型 | 瓦片所对应的最大经纬度 |

功能的具体划分见表 6.4‑2 瓦片的数据结构（Tile）功能

表 ‑2 瓦片的数据结构（Tile）功能

|  |  |  |
| --- | --- | --- |
| **名称** | **返回值** | **说明** |
| GetParent | Tile | 获得对应的父级瓦片 |

UML描述如图 6.4‑1 瓦片数据结构（Tile）

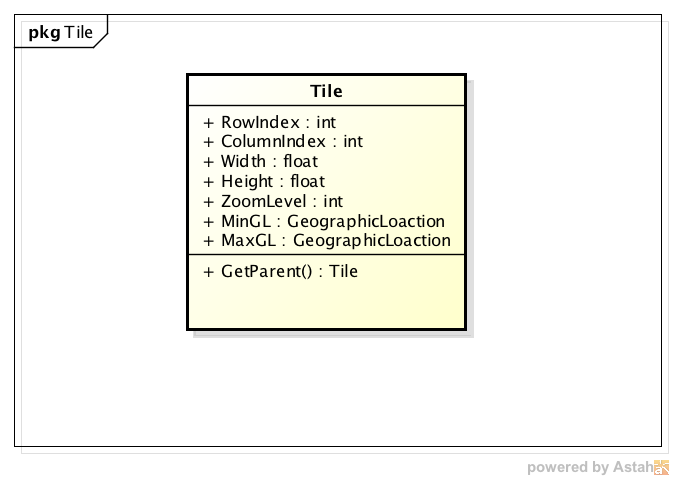


图 ‑1 瓦片数据结构（Tile）

### 瓦片矩阵的数据结构

瓦片矩阵的数据结构（TileMatrix）提供了矩阵涵盖的地理范围、尺寸规格、参考坐标系、放缩级数以及瓦片数组。

字段的具体划分见表 6.4‑3 瓦片矩阵的数据结构（TileMatrix）字段

表 ‑3 瓦片矩阵的数据结构（TileMatrix）字段

|  |  |  |
| --- | --- | --- |
| **名称** | **类型** | **说明** |
| Tiles | 瓦片二维数组 | 矩阵内的瓦片集合 |
| Width | 浮点 | 像素为单位的矩阵宽度 |
| Height | 浮点 | 像素为单位的矩阵高度 |
| ZoomLevel | 整型 | 放缩级数 |
| MinGl | 自定类型 | 矩阵所对应的最小经纬度 |
| MaxGl | 自定类型 | 矩阵所对应的最大经纬度 |

功能的具体划分见表 6.4‑4 瓦片矩阵的数据结构（TileMatrix）功能

表 ‑4 瓦片矩阵的数据结构（TileMatrix）功能

|  |  |  |
| --- | --- | --- |
| **名称** | **返回值** | **说明** |
| GetTilesByParent | Tile[] | 根据传入的父瓦片获得子瓦片集合 |
| GetTile | Tile | 根据传入的行号列号获得对应瓦片 |

UML描述图 6.4‑2 瓦片矩阵的数据结构(TileMatrix)

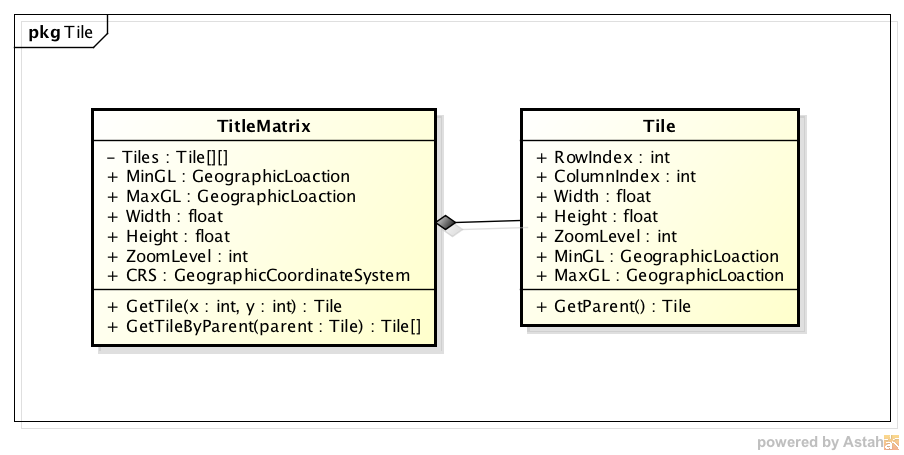


图 ‑2 瓦片矩阵的数据结构(TileMatrix)

### 瓦片金字塔的数据结构

瓦片金字塔的数据结构（TilePyamid）提供了瓦片矩阵数组和缩放级数数组信息

字段的具体划分见表 6.4‑5 瓦片金字塔的数据结构(TilePyamid)字段

表 ‑5 瓦片金字塔的数据结构(TilePyamid)字段

|  |  |  |
| --- | --- | --- |
| **名称** | **类型** | **说明** |
| Matrixs | 矩阵二维数组 | 金子塔内各层矩阵 |
| Zooms | 缩放级数 | 提供缩放级数和比例尺对应集合 |

功能的具体划分见表 6.4‑6 瓦片金字塔数据结构(TilePyamid) 功能

表 ‑6 瓦片金字塔数据结构(TilePyamid) 功能

|  |  |  |
| --- | --- | --- |
| **名称** | **返回值** | **说明** |
| GetMatrix | 矩阵 | 根据传入的放缩级别值获得矩阵 |

UML描述图 6.4‑3 瓦片金字塔的数据结构（TilePyramid）

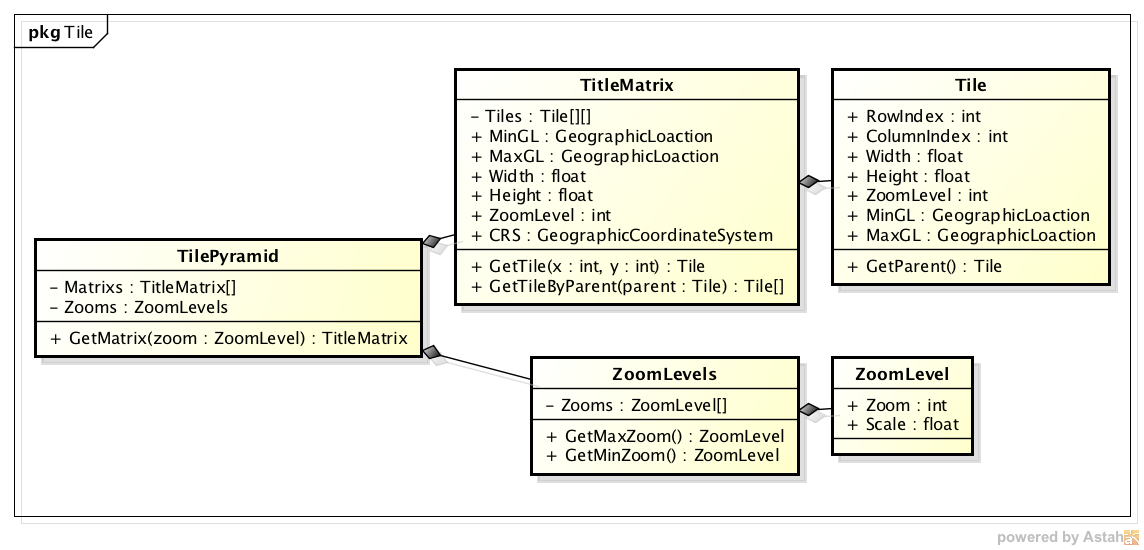


图 ‑3 瓦片金字塔的数据结构（TilePyramid）

## 本章小结

在本章深入学习了瓦片地图技术的原理、公式和实现方式，并且得到了适用于本开发组件的瓦片金字、瓦片矩阵、瓦片和相关的放缩级数一整套栅格数据结构集合，为下面加载栅格地图的开发奠定了基础。

1. 栅格数据源模块详细设计

地图信息系统的客户端通过消费基于Web瓦片地图的服务，获得了在网络压力和用户体验的改善。同时，服务器端通过提供基于Web瓦片地图的服务，实现了对于客户端请求地图数据的预生成，提高了响应速度，更加易于通过横向扩展来提高系统性能。

因此，瓦片技术已经成为了在线地图的通用标准技术。

瓦片地图服务的服务协议主要采用OGC的WMTS协议，或者采用扩展自WMTS定义的接口。

现在的一部分互联网服务提供商对于自家的服务采取闭环策略，例如瓦片地图服务的接口和协议不再向公众开放，转而通过提供以中心坐标、地理范围和放缩级数为唯一标识的静态图片，供消费端镶嵌地图用。对于这样的地图服务，我们可以将同一放缩级数下的静态地图集合看做一个瓦片矩阵，而多个放缩级数下的静态地图集合构成了瓦片金字塔。

## 基于WMTS接口的地图源

OGC推出的WMTS 标准通过定义了系统平台中的各种资源、用户请求的资源、服务元数据文档、瓦片和要素文档的语义,同时还定义了不同类型架构的客户端与服务器间的交互机制。

在本文中使用基于Restful风格设计的服务器客户端交互方式。

### WMTS标准介绍及实现

ServiceMetadata、Tile、FeatureInfo三种元素构成了WMTS标准的基本数据定义，实现了对于基于WMTS地图服务提供的资源以及资源所依赖的信息的描述。

#### ServiceMetadata

ServiceMetadata描述了服务中可用的资源以及消费该服务的必备条件，是对WMTS在SOA架构下的GetCapabilities请求的返回值，同时也是WMTS在Restful架构风格下的消费地图服务的起始路径的返回值。

ServiceMetadata由对于特定的地图服务器的元数据ServiceIdentification和关于服务提供商的信息的ServiceProvider，用于描述在SOA架构下提供的服务操作功能的OperationsMetadata，提供了瓦片地图数据元数据的Contents以及对于各地图图层对应的专题信息描述的Themes共5部分组成。

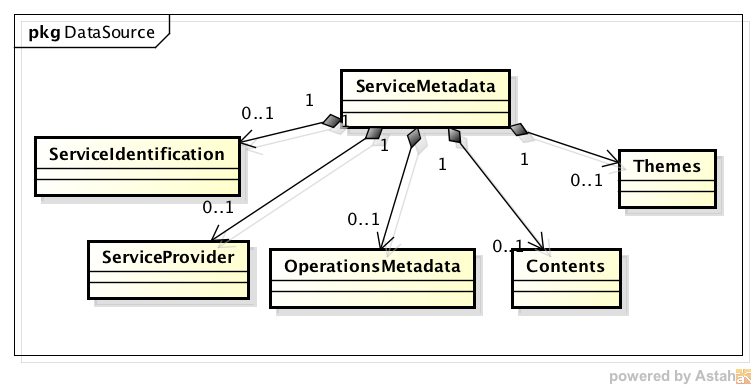


图 ‑1 ServiceMetadata数据结构

这里简要介绍下ServiceMetadata的Contents子集。

如图 7.1‑2 Layer结构摘要，Contents中提供了对于可访问的地图图层的一般性描述和附属的维度、符号、图片格式以及每图层的瓦片矩阵说明。

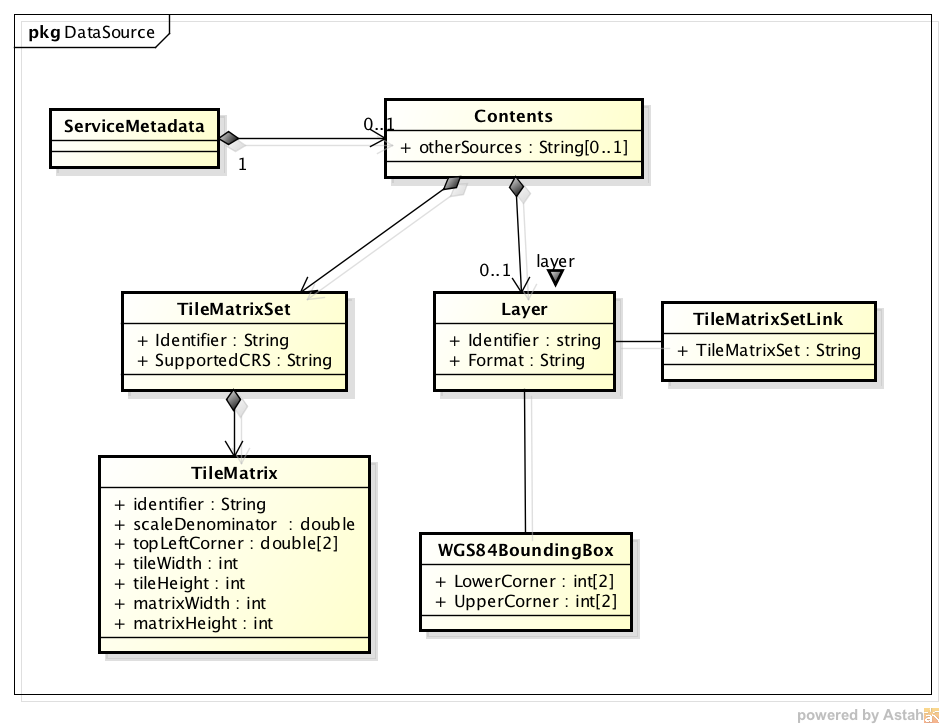


图 ‑2 Layer结构摘要

具体描述见表

表 ‑1 Contents类结构描述

|  |  |  |
| --- | --- | --- |
| **名称** | **类型** | **说明** |
| layer | Layer | 瓦片金字塔的元数据，具体见表 7.1‑2 Layer类结构描述 |
| tileMatrixSet | TileMatrixSet | 每个layer下所有的瓦片矩阵集合，具体见表 6.5‑3 TileMatrixSet 类结构描述 |

表 ‑2 Layer类结构描述

|  |  |  |
| --- | --- | --- |
| **名称** | **类型** | **说明** |
| Identifier | String | 瓦片金字塔的唯一标示 |
| tileMatrixSetLink | TileMatrixSetLink | 该瓦片金字塔所有瓦片矩阵的引用，以及该瓦片矩阵的行列号范围（可选） |
| wGS84BoundingBox | WGS84BoundingBox | 该瓦片金字塔对应的地理范围 |
| Format | String | 瓦片对应的图片格式 |

表 ‑3 TileMatrixSet 类结构描述

|  |  |  |
| --- | --- | --- |
| **名称** | **类型** | **说明** |
| Identifier | String | 瓦片金字塔的唯一标示 |
| scaleDenominator | Double | 该瓦片图层的比例尺分母 |
| topLeftCorner | Double[2] | 该瓦片矩阵在CRS下的坐标 |
| TileWidth | Double | 一个瓦片的像素宽度 |
| TileHeight | Double | 一个瓦片的像素高度 |
| MatrixHeight | Double | 瓦片图层的像素高度 |
| MatrixWidth | Double | 瓦片图层的像素宽度 |

#### Tile

本论文主要论述基于Restful架构风格提供的接口。

客户通过URL想服务器发出请求来获得对应瓦片，请求Url格式如下：

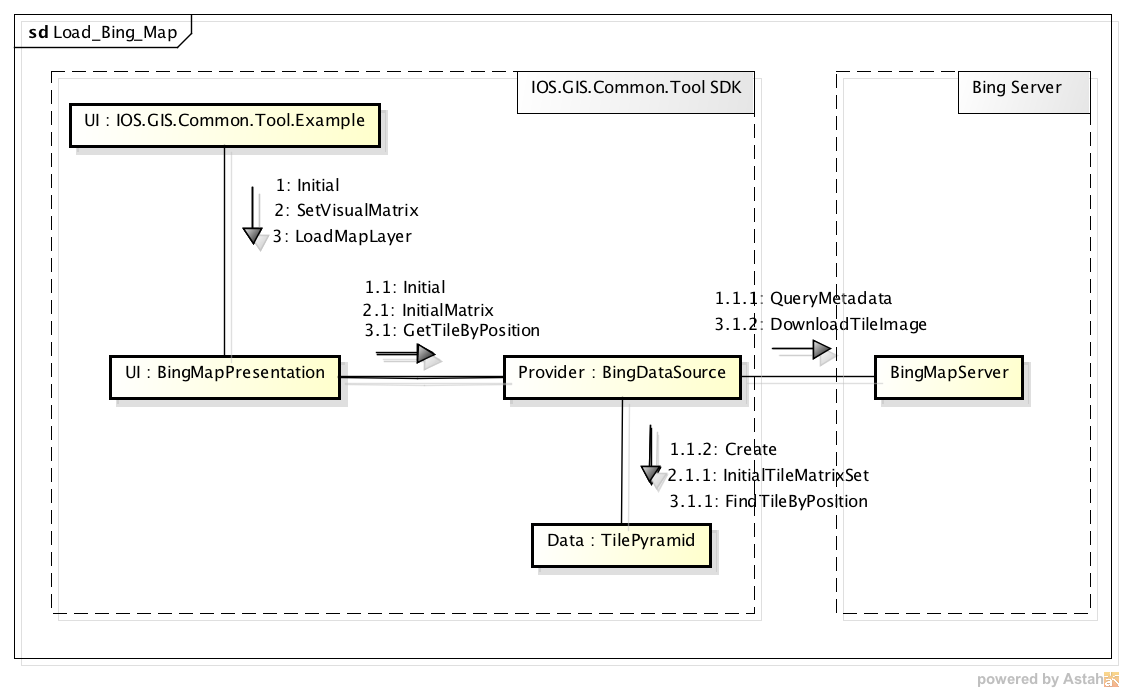
http://host/{layer}/{CRS} /{TileMatrix}/{TileRow}/{TileCol}.{format}

返回值：对应格式的图片。

### 以Bing地图为例说明WMTS数据源的消费方式

微软提供的Bing地图提供了基于WMTS的地图数据服务。微软在WMTS上针对Bing地图服务的一些特点做了一些修改，例如提供了JSON格式的数据描述以及Rest接口，瓦片矩阵信息不出现在metadata中而是全局通过统一的算法换算和比例尺。

使用IOS.GIS.Common.Tool.BingDataSource通过Bing地图的Rest服务来获得瓦片地图的基本流程，如图 7.1‑3 IOS.GIS.Common.Tool加载Bing地图过程。从服务器加载bing地图主要由两个步骤组成：加载Bing地图元数据和加载Bing地图瓦片矩阵。

 图 ‑3 IOS.GIS.Common.Tool加载Bing地图过程

#### 加载Bing地图的实现

在WMTS标准基础上，微软根据Bing地图服务的特点，进一步将地图的元数据拆分为两部分：地图元数据常量和地图图像元数据。

##### Bing地图常量

微软地图常量包括地图参考坐标系、地图分辨率、比例尺、像素坐标系、瓦片坐标系和，具体如下表：

表 ‑4 微软地图常量

|  |  |
| --- | --- |
| **常量** | **具体说明** |
| 地图坐标系参考 | Bing地图采用的大地坐标系是WGS-84，投影是墨卡托投影坐标系。 |
| 地图分辨率 | 分为18级，从1级开始，具体见Bing地图比例尺和缩放等级 |
| 比例尺 | 分为18级，从1级开始，具体见Bing地图比例尺和缩放等级 |
| 像素坐标系 | 坐标范围在(256\*2^图像等级–1，256\*2^图像等级- 1) |
| 地理范围 | 世界地图 |
| 瓦片坐标系 | 具体见Bing瓦片坐标系和四叉树键值编码 |

##### Bing地图比例尺和缩放等级

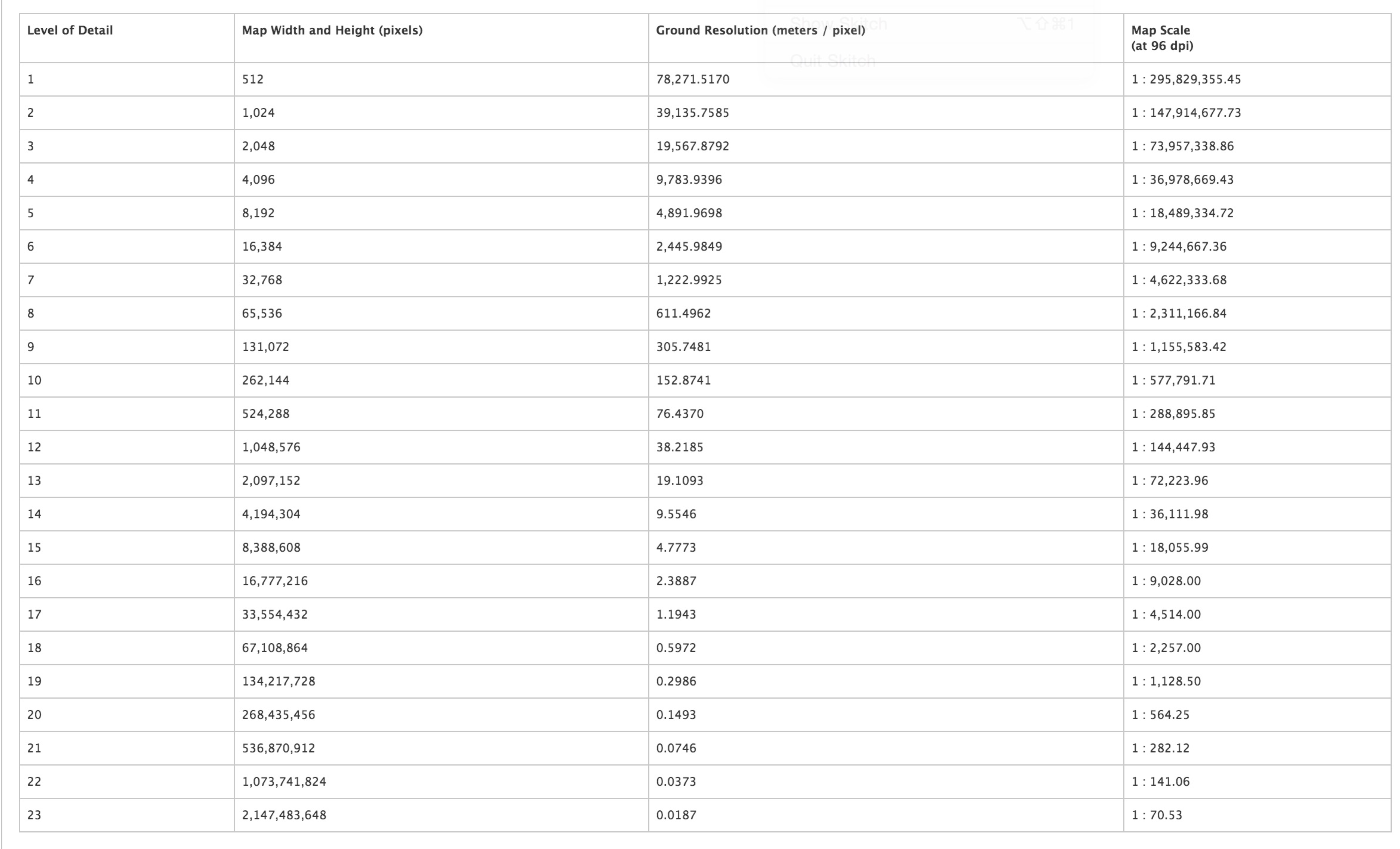


图 ‑4 Bing地图比例尺和缩放等级

##### Bing瓦片像素坐标系

在不同缩放级别下，瓦片矩阵的分辨率是不同的，所以每一级别下都有以自己级别像素宽高为准的坐标系，这些坐标系遵循着统一的规则和换算公式：

1. 原点在地图的左上角，向右向下为正方向；
2. 每一集的坐标系最大值为(地图宽度 - 1, 地图高度 - 1)
3. 地图宽度 = 地图高度 = （以像素为单位）

如图 7.1‑5 像素坐标系示例，该图像对应的瓦片矩阵的缩放级别为3，像素坐标范围为（0,0）至(2047，2047)

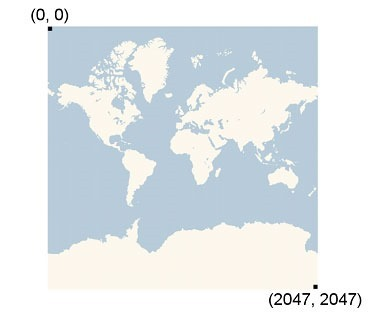


图 ‑5 像素坐标系示例

##### Bing瓦片坐标系和四叉树键值编码

Bing地图在实现瓦片金字塔时，定义瓦片为边长256像素的正方形瓦片，并规定：

1. 原点在左上角，向右向下为正方向
2. 坐标系范围为（0，0）~（）
3. 每个放缩级别各边瓦片 =

微软通过使用四叉树键值编码(quardtree keys，缩写为quadkeys)为瓦片金子塔的每一个瓦片设置了唯一标示，将三维索引（放缩级别，瓦片X坐标，瓦片Y坐标）简化为了一维索引形式quadkey。

计算一个瓦片的Quadkey的编码规则如下：

1. 确定瓦片所在放缩级别；
2. 计算瓦片X坐标，计算瓦片Y坐标
3. 将瓦片坐标Y和X的2进制值进行交错组合，并转换为4进制
4. 将4进制值保存为字符串，这个字符串就是该瓦片的QuadKey

例如：

定义瓦片A的放缩级别为3，坐标值为（3，5）。

经转换可得。

特别需要指出，quadkey的字符串长度正好等于对应瓦片的放缩级别，最后一位字符的数值与对应瓦片在父瓦片的4个子瓦片队列中的索引一致（索引顺序为从左到右，从上到下）。

##### Bing地图图像元数据

Bing地图通过Restful服务的形式提供了的元数据。

###### 请求元数据

Restful服务的Url为：

<http://dev.virtualearth.net/REST/v1/Imagery/Metadata/imagerySet?key=BingMapsKey>

传入参数如下表

表 ‑5 Bing地图元数据请求参数说明

|  |  |
| --- | --- |
| **参数名称** | **具体说明** |
| imagerySet | 请求地形图像类型：  Aerial - Aerial imagery.  AerialWithLabels - Aerial imagery with a road overlay.  Road - Roads without additional imagery. |
| BingMapKey | 向Bing地图服务申请的访问地图的秘钥。 |

###### 元数据响应信息：

对于元数据的请求，Bing地图服务提供了5种响应状态：

表 ‑6 响应状态

|  |  |
| --- | --- |
| **响应状态** | **具体说明** |
| 200 | 成功 |
| 400 | 无效的请求。 |
| 401 | 请求未得到授权 |
| 500 | 服务器错误 |
| 503 | 服务不可用 |

响应信息中包含了对于地图元数据的具体描述，

表 ‑7 响应信息数据结构

|  |  |  |
| --- | --- | --- |
| **字段名称** | **数据类型** | **具体说明** |
| ImageUrl | URI | 获得瓦片图像的Url模板 |
| imageUrlSubdomains | String | ImageUrl中对应的subdomain |
| imageWidth | Int | 瓦片图像的宽度 |
| imageHeight | Int | 瓦片图像的高度 |
| zoomMin | Int | 该瓦片金字塔最小的放缩等级 |
| zoomMax | int | 该瓦片金字塔最大的放缩等级 |

响应信息示例：



图 ‑6 Bing地图元数据

其中imageUrl对应的值为

表 ‑8 Bing地图返回值

|  |  |
| --- | --- |
| **字段名称** | **具体说明** |
| ImageUrl | http://ecn.{subdomain}.tiles.virtualearth.net/tiles/r{quadkey}.jpeg?g=129&mkt={culture} |
| imageUrlSubdomains | [“t0”,”t1”,”t2”,”t3”] |

ImageUrl模板中的占位符：

表 ‑9 Bing地图请求Url模板占位符

|  |  |
| --- | --- |
| **字段名称** | **具体说明** |
| subdomain | 响应请求的服务器子域 |
| quadkey | 请求瓦片的quadkey |
| culture | 请求瓦片上附加标示的语言类型，中文为cn |

##### BingDataSource初始化Bing地图和加载瓦片地图

如图 6.5‑7 BingDataSource类结构图，IOS.GIS.Common.Tool通过Bing地图Restful服务加载Bing地图通过BingDataSource完成以下两个步骤：请求元数据和请求瓦片图片。BingDataSource具体如下表 6.5‑10 Bing地图字段和

表 ‑10 Bing地图字段

|  |  |  |
| --- | --- | --- |
| **字段名称** | **类型** | **具体说明** |
| tilePyramid | TilePyramid | Bing地图对应的瓦片金字塔 |
| ImageUrlTemplate | String | 组织好的瓦片图片Url模板 |

表 ‑11 Bing地图方法

|  |  |  |
| --- | --- | --- |
| **方法名称** | **类型** | **具体说明** |
| Initial | N/A | 初始化瓦片金字塔 |
| GetTileByPosition | Url | 获得瓦片对应的地图UI |
| InitialMatrixSet | N/A |  |

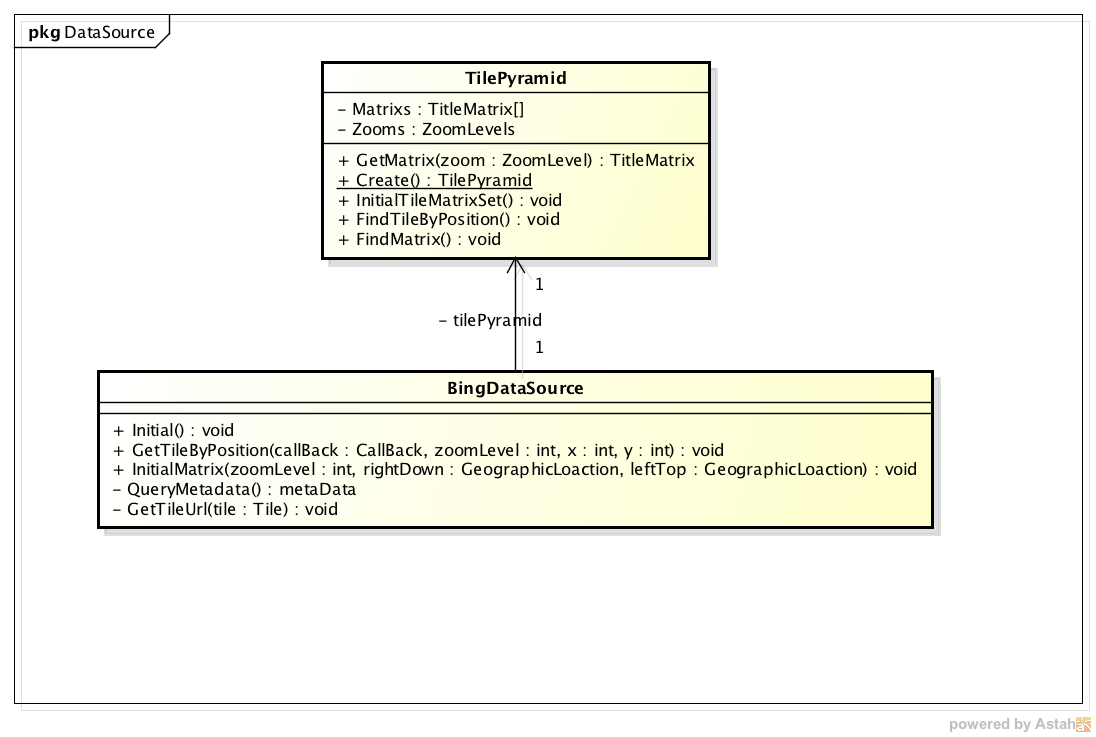


图 ‑7 BingDataSource类结构图

###### 请求元数据构造瓦片金字塔

请求元数据发生在地图初始化时构造对应的瓦片金字塔时，这时候通过加载Bing地图元数据和地图常量，就可以构建出瓦片金字塔的基础骨架。

如图 6.5‑8 初始化瓦片金字塔，在初始化图层时，BingDataPresentation会调用BingDataSource的Initial方法来完成瓦片金字塔的初始化，初始化需要确定Bing地图的放缩范围、参考坐标系等：

1. 读取地图常量，获得坐标系信息、默认放缩级别范围等信息
2. 请求瓦片地图元数据，获得图片Url模板及实际支持放缩级别范围
3. 根据地图常量和实际放缩级别创建瓦片金字塔。

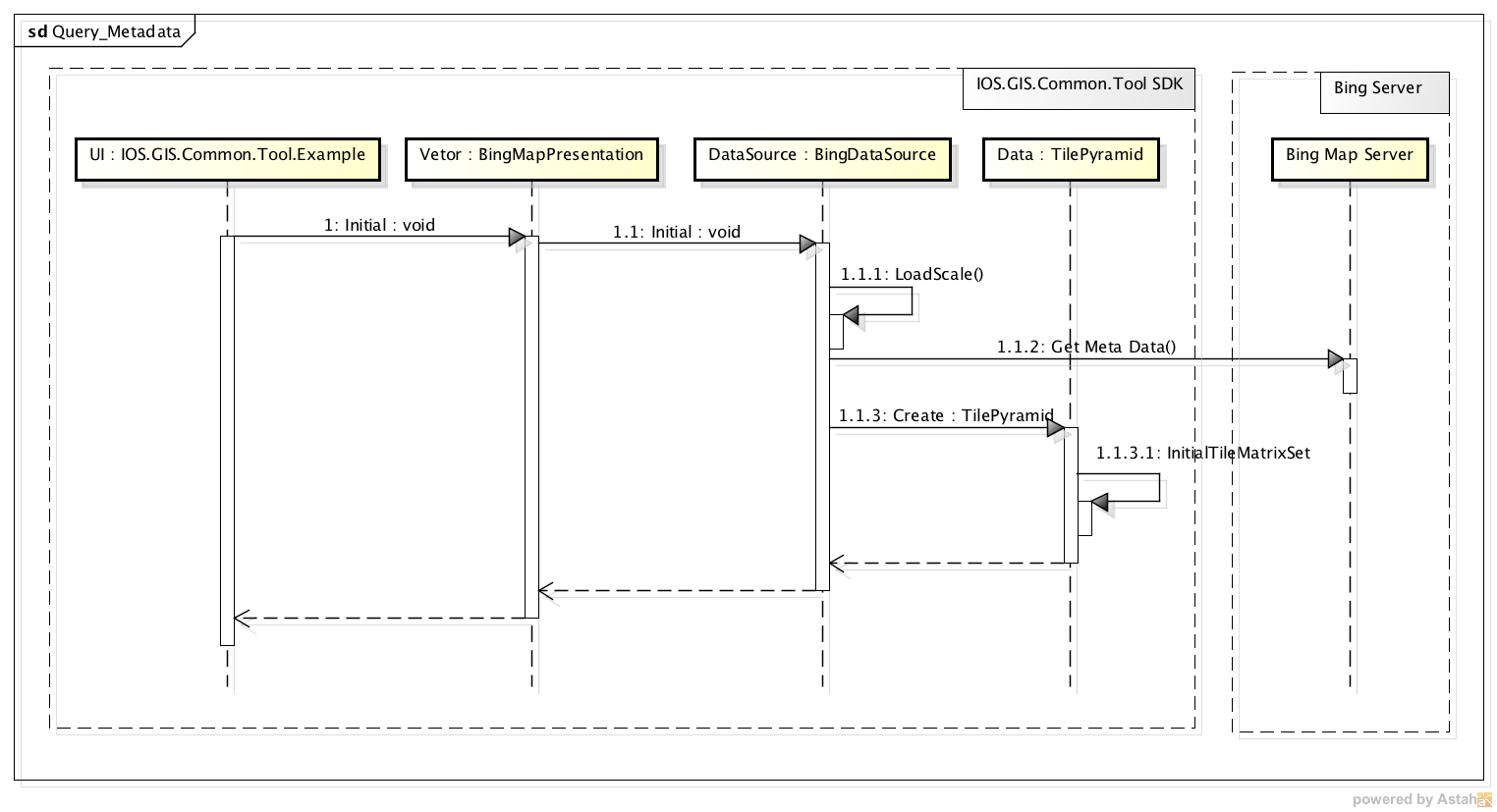


图 ‑8 初始化瓦片金字塔

###### 请求瓦片图像数据

而当移动客户端渲染地图时，IOS.GIS.Common.Tool会根据瓦片金字塔记录的信息，向服务请求瓦片对应的图片。如图 6.5‑10 渲染瓦片图像，渲染地图界面上的一个瓦片图像时：

1. 通过坐标和放缩级别找到对应的瓦片数据
2. 将瓦片转换为quadkey，并且通过替换图片Url模板的参数获得该瓦片对应图片的Url
3. 从服务器现在图片并渲染

当图形界面加载地图图层时，渲染瓦片单元图像的操作是并发进行的。

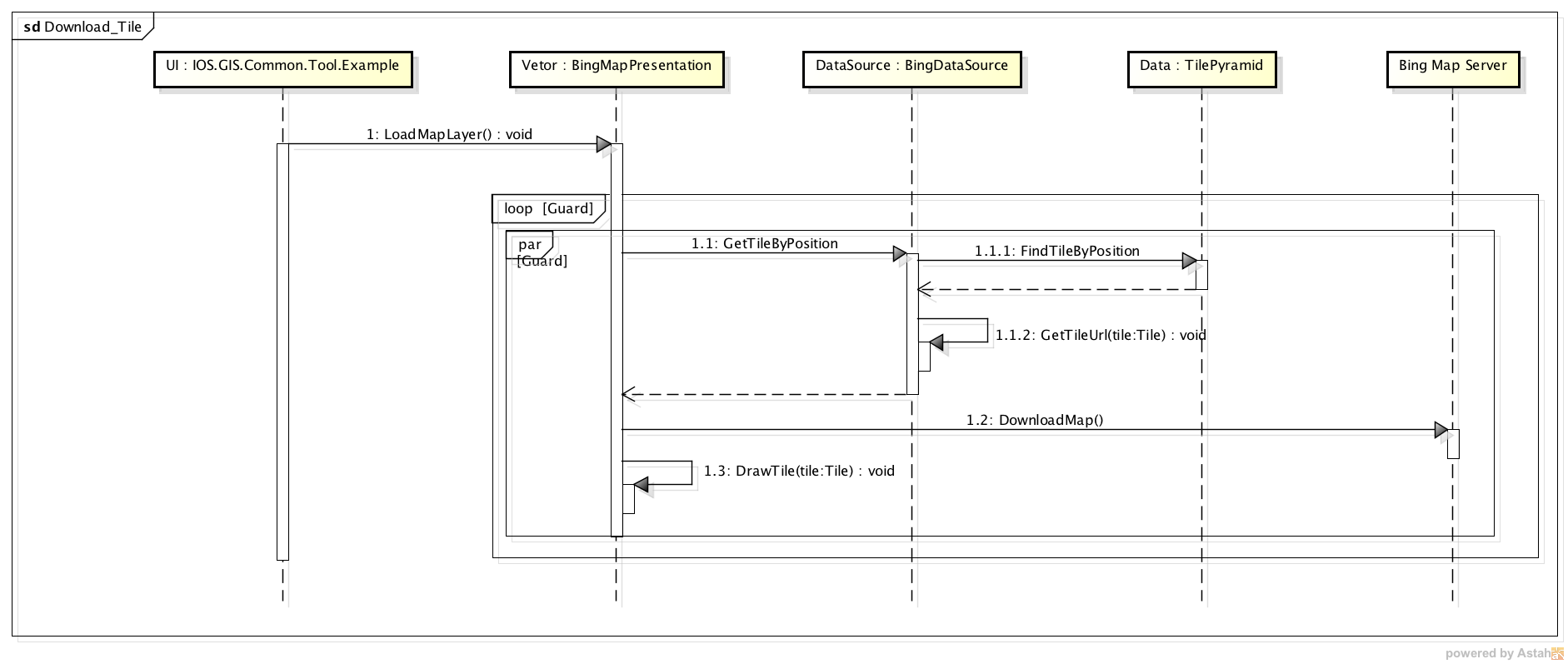


图 ‑9 渲染瓦片图像

###### 计算四叉树键值

BingDataSource很重要的一个功能是计算特定缩放级别下的瓦片的四叉树键值。具体实现流程如下图 6.5‑10 四叉树键值：

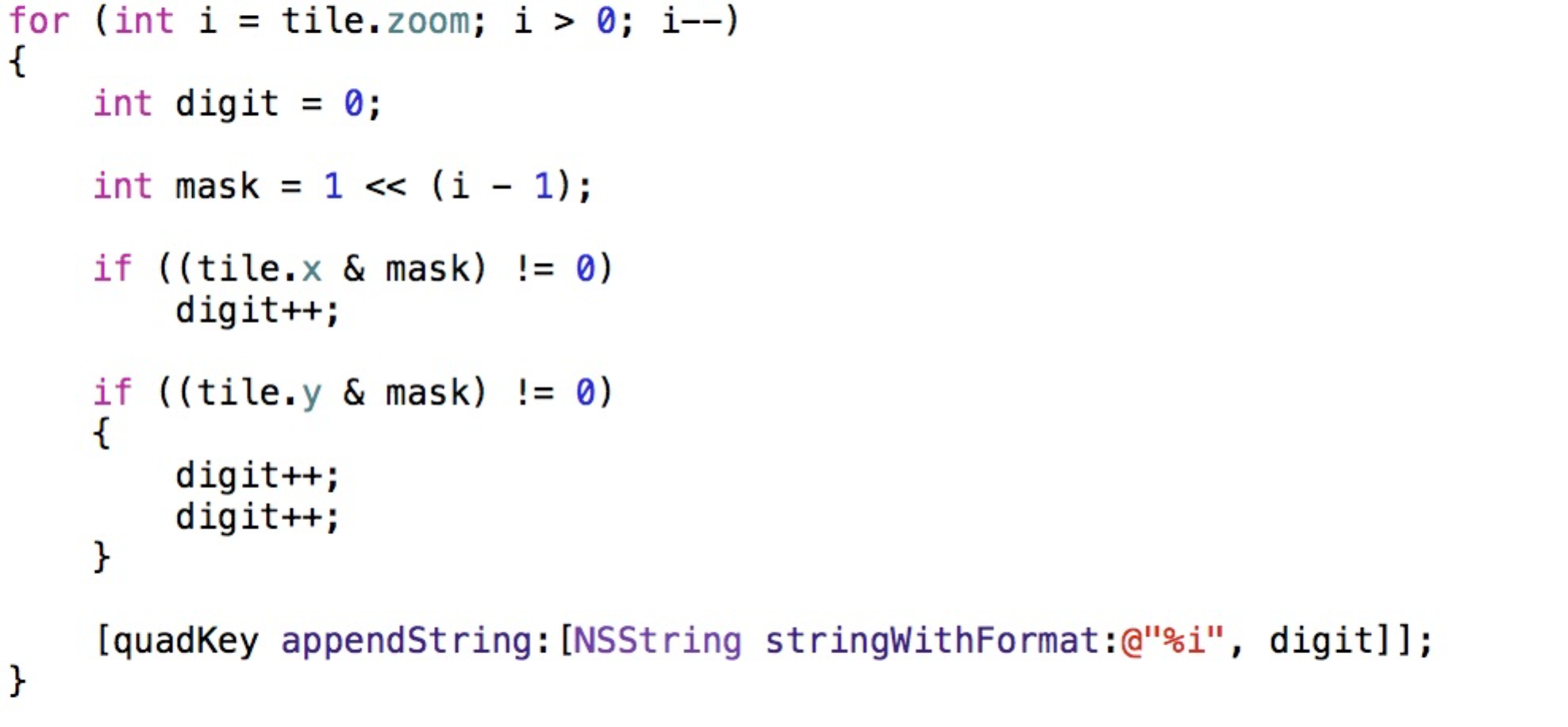


图 ‑10 四叉树键值

### 基于静态图片的地图源

最近几年，很多地图服务商如百度、搜狗等，已经不再对外开放瓦片地图服务，同时为了满足图形化地理位置的需求，这些服务商相继提供了静态地图图片作为数据源。

静态地图图片指的是，以在某一缩放等级下，某一地点为中心，一定距离范围内的地形作为一张图片提供给消费端。这样的图片一般通过以下信息来索引：

#### 使用静态图片渲染瓦片地图的参数转换

如果需要利用静态图片来渲染瓦片地图，需要先建立静态地图索引和对应瓦片地图的索引关系。

即：

将瓦片索引

转化为

这是一个在消费端重建瓦片金字塔的过程，按照6.2瓦片金字塔模型研究可得：

在相同放缩级别下假设有一瓦片（a, x, y）对应的静态地图为(a, centerPoint, xLength, yLength)。

其中：

a——放缩级别。

x——瓦片在对应矩阵坐标系下的x坐标。

y——瓦片在对应矩阵坐标系下的y坐标。

centerPoint——静态地图中心店在WGS84下的经纬度坐标。

xLength ——静态地图x方向边长，以像素为单位。

yLength ——静态地图y方向边长，以像素为单位。

假设中心点转化为a级别下的矩像素坐标为（Xcpixel，Ycpixel），a级别下的瓦片矩阵边长为c，以像素单位。

则可得公式：

根据不同地图服务商提供的换算公式，可得中心店centerPoint的经纬度坐标和像素坐标的转换。

以百度地图为例，平面坐标系的原点与经纬度的原点一致，即赤道与0度经线相交的位置。

*像素坐标 = |平面坐标 × 2 zoom - 18|*

可得公式组

其中

——中心点坐标在墨卡托投影坐标系的x值。

——中心点坐标在墨卡托投影坐标系的y值。

WGS84空间坐标转墨卡托投影坐标公式见地理坐标模块详细设计。

由任意一点坐标公式

可得图上任意点转获得对应的静态图片的请求参数。

#### 以百度地图为例实现基于静态地图的渲染

下面以继续以百度地图的静态地图API为例，说明IOS.GIS.Common.Tool SDK是如何实现通过静态地图组织瓦片金字塔的。

百度地图静态地图的API为

http://api.map.baidu.com/staticimage?center={x,y}&width={width}&height={height}&zoom={zoom}

|  |  |  |
| --- | --- | --- |
| **字段名称** | **类型** | **具体说明** |
| x,y | 平面直角坐标 | 请求静态图片的中心坐标 |
| Width | 宽度 | 请求静态图片的宽度，这里和瓦片地图的宽度相符（256个像素） |
| Height | 高度 | 请求静态图片的高度，同宽度 |
| Zoom | 放缩级别 | 详见图 6.5‑11 百度地图放缩级别和比例尺 |



图 ‑11 百度地图放缩级别和比例尺

由6.5.1节给出的公式，可以将某放缩级别下，根据一个瓦片的行列号，快速换算得到瓦片的中心坐标，从而得到以上4个要素，组织成一个图片API，向百度地图请求对应图片。

如图 6.5‑12 加载百度静态地图，静态地图的加载同样分为两步：

1. 初始化瓦片矩阵
2. 渲染可视区域

和通过WMTS服务来加载地图相比，通过静态图片加载地图少了在初始化瓦片矩阵时请求元数据，多了在每个瓦片请求对应图片前，将瓦片行列，转化为中心点坐标，总体流程一致。

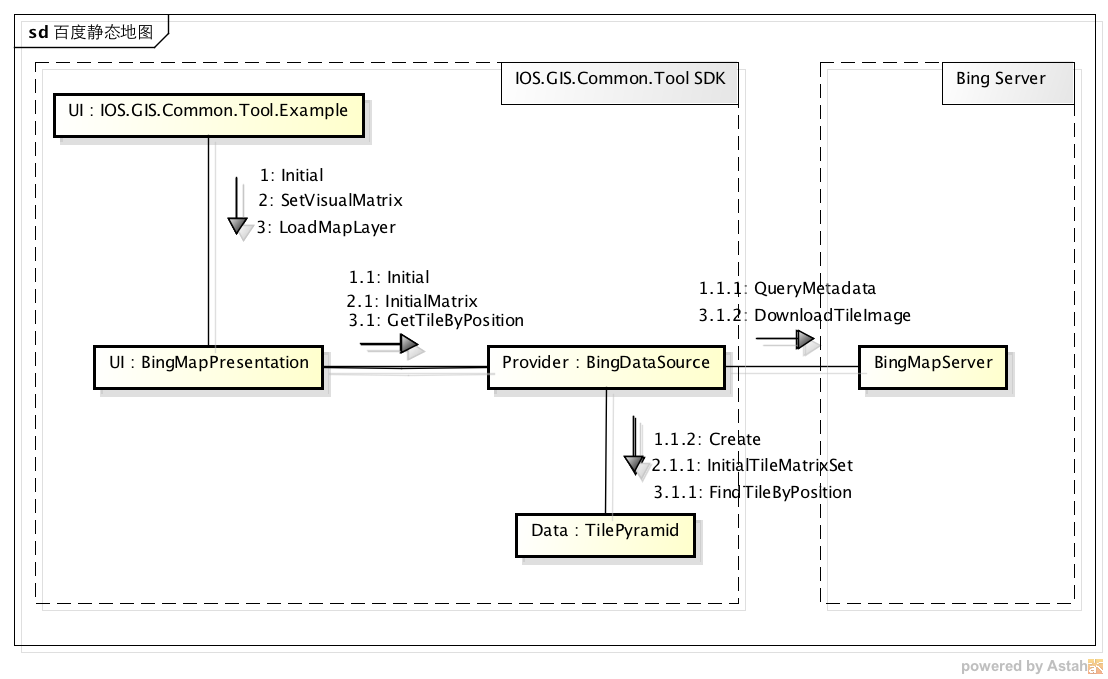


图 ‑12 加载百度静态地图

因而，如图 6.5‑13 百度静态地图渲染，整个栅格数据源被抽象为一个名为TileMapDataSource的基类，BaiduStaticDataSource通过继承IloadStaticMap获得瓦片行列转换为标准Url的能力，重写了继承于父类的GetTileByUrl。

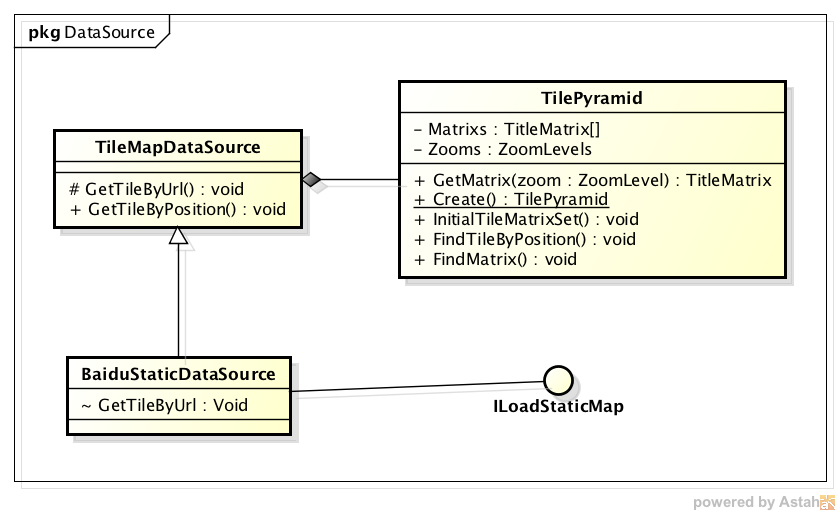


图 ‑13 百度静态地图渲染

## 本章小结

本章，在前3章的基础上，通过研究瓦片地图技术，了解了瓦片地图的定义和数据结构；通过了解WMTS标准，实现了一个基于Bing地图的栅格数据源。

同时，考虑到互联网地图服务商逐渐限制自家瓦片地图的开发访问，提出了一个基于静态地图图片构架瓦片地图的方案。并且通过数学计算，获得了在特定坐标系和地理范围下每个瓦片对应的静态图链接。

1. 栅格可视化模块详细设计

本章将继续阐述IOS.GIS. Common.Tool SDK如何通过TileMapPresentatio nView将瓦片地图渲染到UI界面上，和基于用户交互等原因造成地图可视范围和比例尺的变化导致的瓦片地图重载。

这里需要特别指出的是，关于地图UI坐标系以及平移缩放造成的坐标系变换，将在地图浏览模块中具体阐述。

如图，描述了一个通过IOS.GIS. Common.Tool SDK构造的地图应用加载Bing地图的过程，向地图服务器请求数据的具体流程不在这里赘述，主要阐述瓦片地图渲染到界面的过程。

综合性能和开发效率等方面的需求，TileMapPresentationView使用了CATiledLayer作为瓦片地图的可视化图层。

TileMapPresentationView的平移缩放功能是基于移动地图界面对客户操作的响应和处理完成的。本章主要介绍了TileMapPresentationView在程序初始化时对瓦片地图图片的加载过程，和在放缩和平移处理后对瓦片地图的重载过程。而TileMapPresentationView与主界面交互的问题将放在地图UI模块部分具体论述。

## CATiledLayer

### CATileLayer简介

CATiledLayer在iOS中提供的一种高性能layer，它通过对图层切片的异步加载提高了在渲染高像素图片或类似场景时的用户体验。通过对绘画内容的几何变换和实现了对于基于切片的图形变换功能，图像的放缩序列为2的等比序列，放缩总级数见表 8.1‑1 CATiledLayer属性

表 ‑1 CATiledLayer属性

|  |  |
| --- | --- |
| **功能名称** | **具体说明** |
| levelsOfDetail | 放缩的级别数量 |
| levelsOfDetailBias | 可缩小的级别数量 |
| tileSize | 每个切片的大小 |

可得公式：

当levelsOfDetail < levelsOfDetailBias时，放缩总级数不受限制。

在iOS系统中，CATiledLayer一般配合ScrollView使用，这里给出CATiledLayer的放缩级数和ScrollView的zoomScale的换算关系。

令 ，level为当期的放缩级数，则可得：

如图 8.1‑1 放缩级别和比例尺换算示例所例

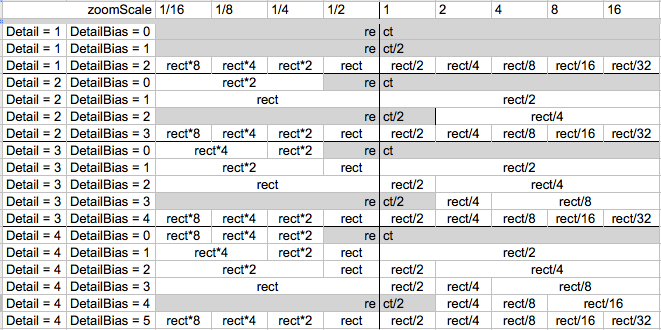


图 ‑1 放缩级别和比例尺换算示例

### 在UIView中使用CATileLayer作为默认图层

UIView及继承于UIView的子类，默认使用的都是CALayer。如果需要在TileMapPresentationView中重写*+ (Class)layerClass*方法。

具体代码如图 8.1‑2 UIView中启用CATileLayer的方法：

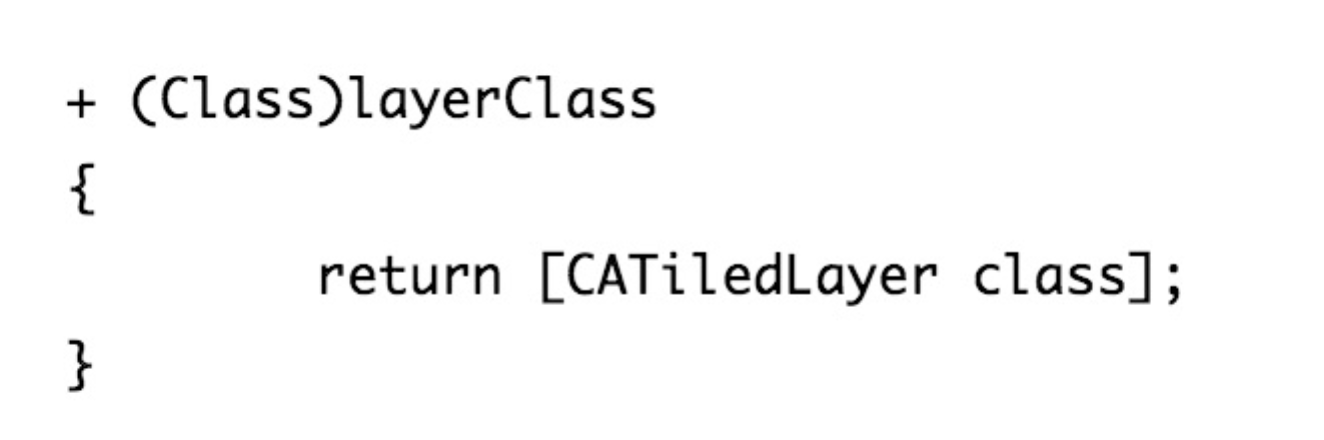


图 ‑2 UIView中启用CATileLayer的方法

## 在TileMapPresentationView上绘制瓦片地图

配合CATileLayer，TileMapPresentationView还需要实现绘制瓦片地图图片的功能，才能将一个瓦片地图完整有效率的呈现给使用者。

下面，从在iOS上绘制图片和在IOS.GIS. Common.Tool SDK中通过TileMapPresentationView完成地图绘制的流程两方面阐述在TileMapPresentationView将地图呈现给用户的流程。

### iOS上绘图的方法简介

iOS支持两套图形API：Core Graphics（QuartZ 2D） 和OpenGL ES。 OpenGL ES是跨平台的图形API，属于OpenGL的一个简化版本。QuartZ 2D是苹果公司开发的一套API，它是Core Graphics Framework的一部分。

Core Graphics Framework是一套基于C的API框架，使用了Quartz作为绘图引擎。它提供了低级别、轻量级、高保真度的2D渲染。该框架可以用于基于路径的绘图、变换、颜色管理、脱屏渲染，模板、渐变、遮蔽、图像数据管理、图像的创建、遮罩以及PDF文档的创建、显示和分析。

IOS.GIS. Common.Tool SDK通过Core Graphics Framework来完成图片的绘制工作。

#### 利用UiView的drawRect绘图

对瓦片地图的加载和重绘，发生在UIView的drawlayer方法触发时。

当用户进行地图的放缩和平移操作时，UIView的drawRect方法会被出发， 对于当前可视区域对应瓦片区域的换算和处理，将在这个时机完成。

对应方法的签名如图 8.2‑2 drawLayer方法签名：

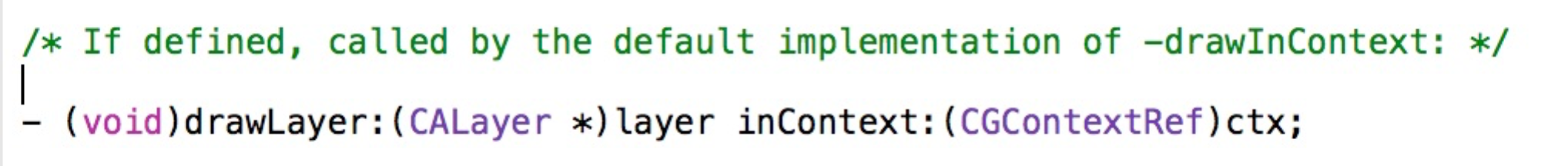


图 ‑1 drawLayer方法签名

表 ‑1 drawLayer方法签名

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数类型** | **具体说明** |
| layer | CALayer | 当前接受绘制图形行为的layer |
| ctx | CGContextRef | 绘制上下文的指针，可以在相应方法下得到当前绘制区域的相关信息 |

#### CATileLayer的初始化

对CATileLayer的初始化完成在UIView的initWithFrame过程中。

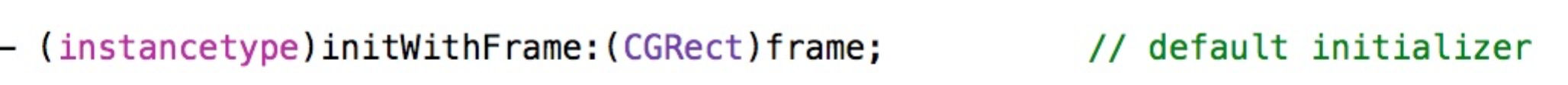


图 ‑2 UIView的initialFrame签名

表 ‑2 initialFrame方法签名

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数类型** | **具体说明** |
| frame | CARect | 所处view的大小范围 |



在初始化TileMapPresentationView时，使用瓦片金字塔中最顶层的瓦片矩阵相关信息初始化tiledLayer。tiledLayer的属性具体设置如表 8.2‑2 CATileLayer初始化参数：

表 ‑3 CATileLayer初始化参数

|  |  |
| --- | --- |
| **功能名称** | **具体说明** |
| levelsOfDetail | 瓦片金字塔的缩放级数 |
| levelsOfDetailBias | 瓦片金字塔的缩放级数 |
| tileSize | 最小放缩级数所对应的瓦片像素边长 |

#### view的坐标体系和图片在view上的定位

在iOS中，view的坐标系都是以左上为原点，向右向下为正方向的。每种view的三种位置属性：frame、bounds、center。

如图 8.2‑2 iOS中的位置属性：

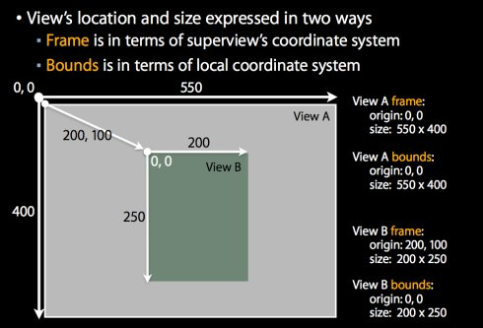


图 ‑3 iOS中的位置属性

由图可得：定义了2个View，ViewA和ViewB,ViewA是ViewB的父View。

* frame定义了一个view在其父view中的几何关系，即大小和起始点，采用的坐标系定义和父view的坐标系定义一致。
* bounds定义了一个view在其自身坐标系下的几何关系，起始点和坐标系原点一致。
* center：该view的中心点在父view坐标系统中的位置和大小，参考坐标系为父view的坐标系。

在章节《8.2在TileMapPresentationView上绘制瓦片地图》中，使用的默认位置信息是参考bounds的，坐标系为view的坐标系。

#### 瓦片矩阵的放缩级数、像素坐标和行列坐标范围

如8.2.1.1利用UiView的drawRect绘图提到的，当用户对地图进行平移放缩操作后，UIView的drawRect方法会被触发。

获得当前的放缩级数，及瓦片矩阵像素坐标和行列坐标范围，在drawRect方法中完成。

设rect为当前可视区域的矩形边框；

bounds为CATiledLayer的矩形边框，因为在初始时采用最低放缩级别来设置，则bounds同时为最低缩放级数下的瓦片矩阵的矩形边框。

在drawRect中获取rect和bounds的方法，如图 8.2‑4 获取瓦片地图范围

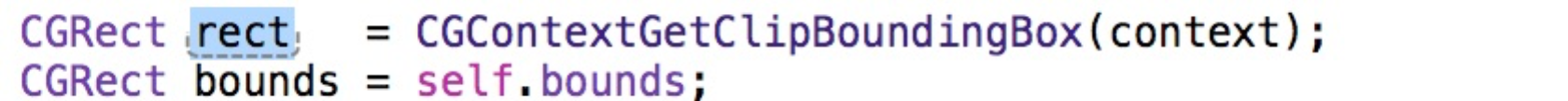
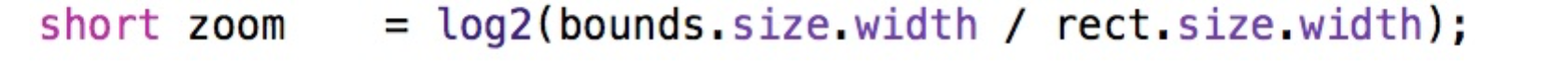


图 ‑4 获取瓦片地图范围

##### 获得放缩级数

设zoom为放缩级数，scaleRule为比例尺，则根据8.1.1 CATileLayer简介所述，可得



##### 获得瓦片矩阵的像素坐标

因为在初始时采用最低放缩级别来设置CATiledLayer，故当前的可是区域下的坐标系就是瓦片矩阵的像素坐标系。

因此设rect就为当前瓦片矩阵的矩形边框，设瓦片矩阵的左上角坐标为(x1,y1)，右下角坐标为(x2,y2)，边长为size，可得

##### 获得行列坐标

根据6.3.1.4瓦片的尺寸规格所给公式，设瓦片矩阵的左上角、右下角行列坐标为（TileCol1，TileRow1）、（TileCol2，TileRow2），可得

#### 利用UIKit绘制图片

UIKit基于Core Graphic，在当前绘图上下文中，将绘制图像的功能接口进行了二次封装，例如通过封装绘制图片的功能接口，是的UIImage对象具备了在当前上下文中绘制自身的功能。

具体调用流程如图 8.2‑5 UIKit绘制图片的流程

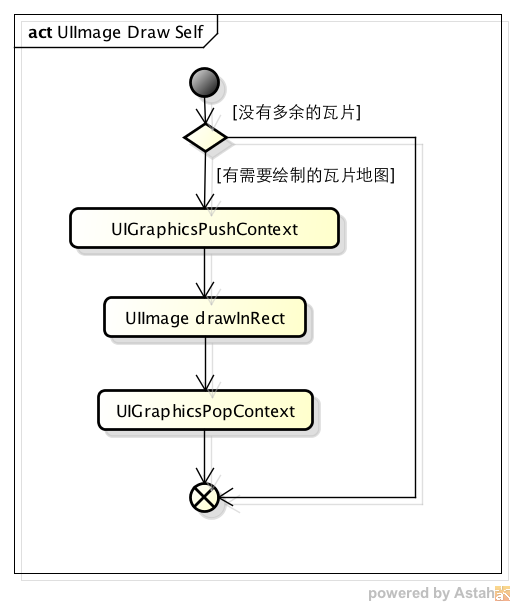


图 ‑5 UIKit绘制图片的流程

涉及的功能接口说明：

表 ‑4 UIImage绘制功能接口说明

|  |  |
| --- | --- |
| **功能名称** | **具体说明** |
| UIGraphicsPushContext | 将当前绘图上下文标记为目标。 |
| drawInRect | UIImage的实例方法，根据传入左上角像素坐标和边长作为图片的frame，将图片绘制在layer上。 |
| UIGraphicsPopContext | 将当前绘图上下文的绘制标记去掉。 |

### 绘制地图的流程综述

#### 初始化地图的流程

IOS.GIS.Common.Tools SDK通过TileMapPresentationView初始化地图的流程如图：

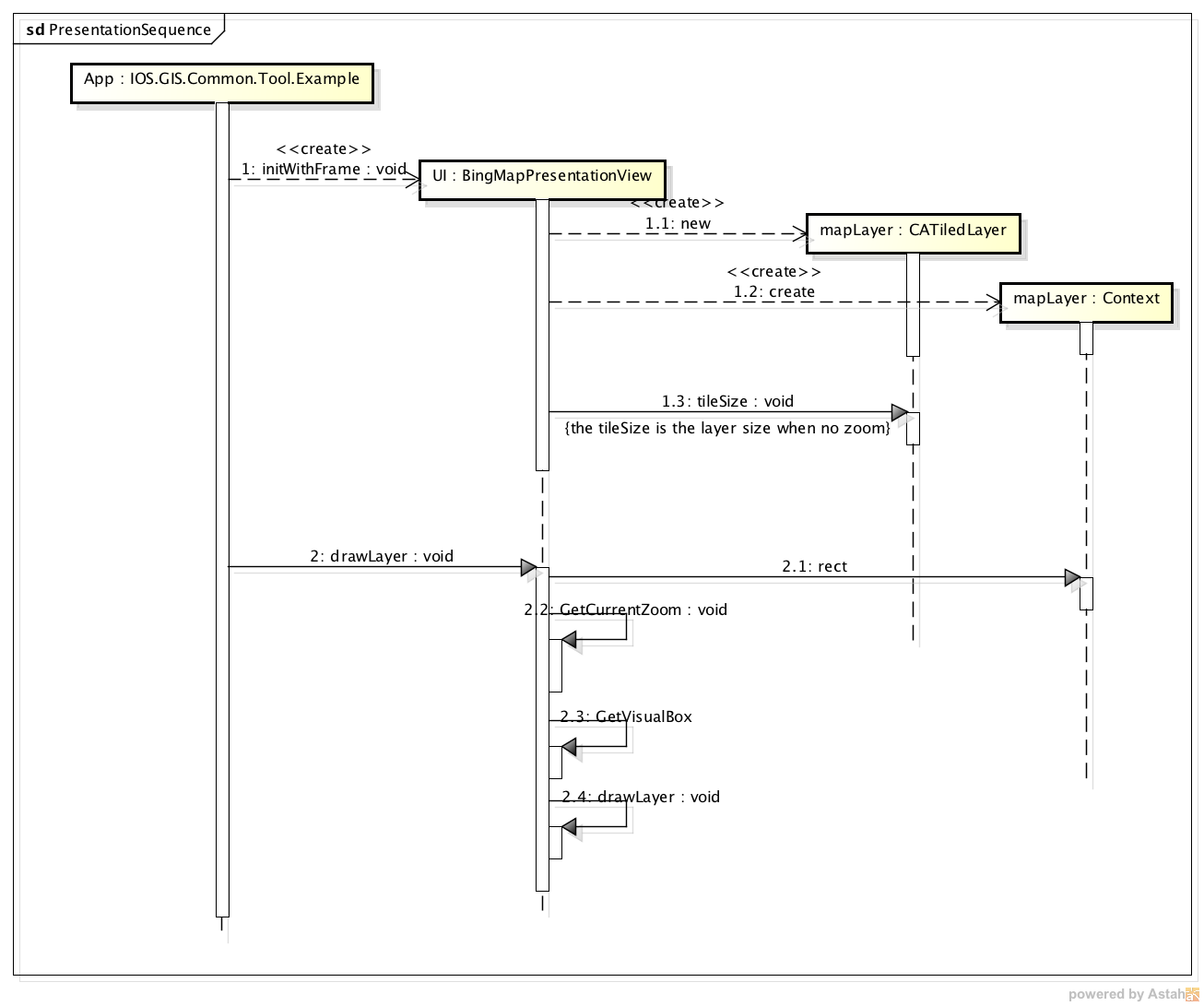


图 ‑6 IOS.GIS.Common.Tools初始化瓦片地图流程

#### 重载地图的流程

当用户对地图做了相应操作，TileMapPresentationView会重载地图。

流程如：

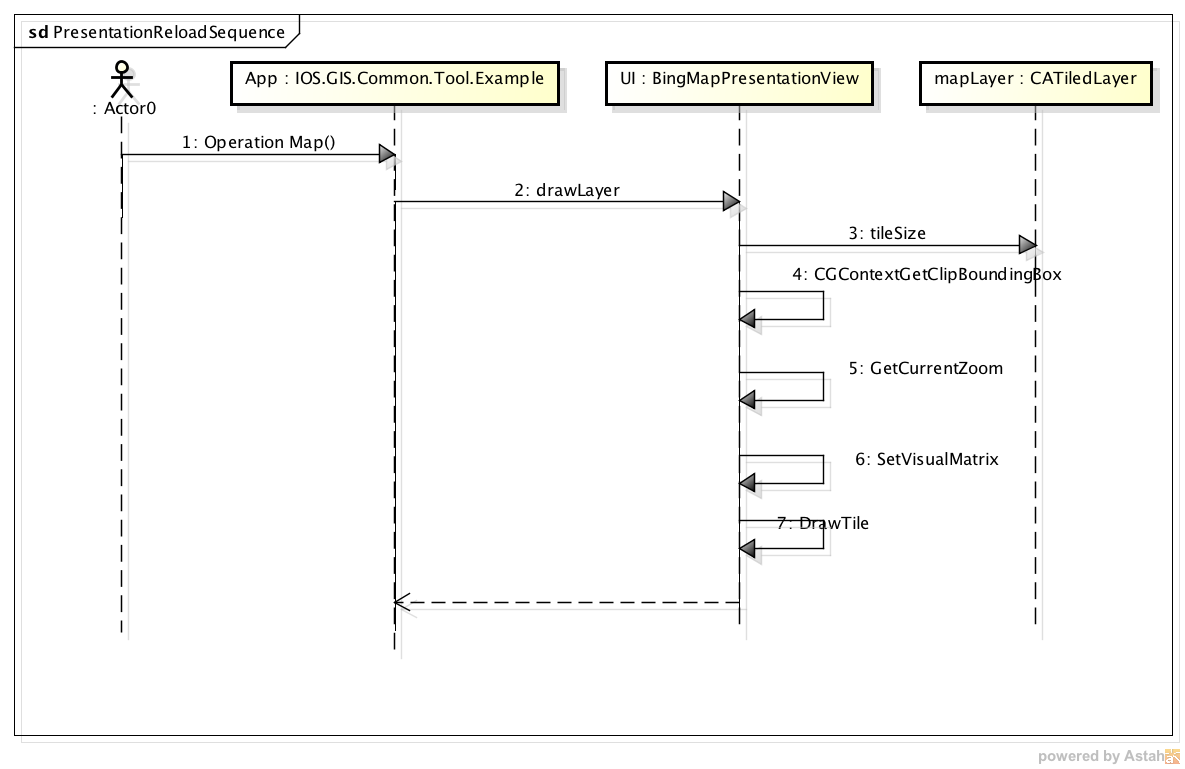
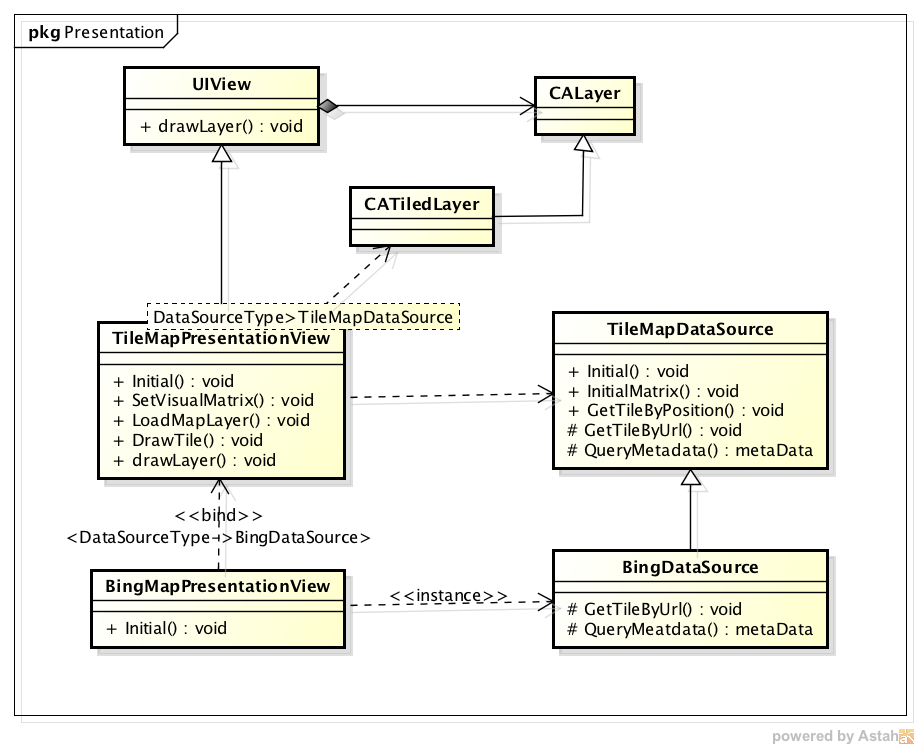
****

图 ‑7 重载地图流程

#### TileMapPresentationView的类结构



1. 地理图形操作模块详细设计

地理图形操作模块实现了用户对GIS应用的操作，是IOS.GIS.Common.Tool 的基础模块。

地理图形操作模块实现的功能包括：栅格地图图层、矢量元素图层、图层管理、坐标转变换接口、元素选择接口、地图浏览接口。

## 矢量元素图层

矢量元素图层是渲染按照矢量数据结构存储的信息的功能图层。

矢量图层提供了对于矢量图层上基本元素的绘制实现，对于复杂元素的绘制由基本元素的绘制复合而成。

~~矢量数据结构是根据矢量数据模型对数据进行管理的。相较栅格结构，矢量结构存储的信息更为精确，允许对任意位置、长度和面积的精确定义。~~

~~矢量数据结构直接记录取点坐标。按照这样的数据组织方式，可以获得精美的地图。~~

~~本论文中采用的矢量地图数据结构的编码方式是实体式~~。

### 矢量图层基本元素

矢量图层上的基本元素主要由几何元素和标志物对象两部分组成。

现代GIS系统主要将地理信息元素抽象为空间数据模型处理。空间数据模型通常分为场模型和对象模型两大类。本论文中主要采用对象模型。

对象模型是指将 研究的地理空间看成一个空域，空间实体和现象作为独立的对象分布在空域中。

对象模型描述的几何体分为四类：点、线、面和几何体集合。

* 点对象   
  点或节点、点状实体。点是有特定位置、维数为0的物体。  
  具体有以下类型的点：实体点、标记点、内点、节点和特征点等不同类型。实体点用来表示一个实体；标记点用于定位标记；内点用于记录多边形的属性，存在于多边形内；节点表示线的终点和端点；特征点表示线段和弧段的内部点。
* 线对象 具有相同属性的点的轨迹。线或折线由一系列的有序坐标表示，有长度、弯曲度、方向性等特性，线状实体包括线段，边界、链、弧端等。
* 面对像 是对湖泊、岛屿、地块等一类现象的描述，又一封闭曲线加内点来表示。具有面积、范围、周长、独立性或与其他地物相邻、内岛屿或锯齿状外形、重叠性与非重叠等特性。
* 几何体集合 现实世界的各种现象比较复杂，往往由不同的空间对象组合而成，复杂实体又简单实体组合表达。

标量对象主要指在图层上显示的图片等元算。

### 矢量图层类关系

表格 4-3 矢量图形类关系

Figure 4-3 vector class diagrams

|  |  |
| --- | --- |
| **类** | **说明** |
| VectorLayerView | 矢量标绘图层，负责调用矢量图形的绘制算法 |
| Annotation | 矢量图形画法基类 |
| PointAnnotation | 点图形画法 |
| PolylineAnnotation | 线图形画法 |
| PolygonAnnotation | 面图形画法 |
| ShapeAnnotation | 复杂图形画法 |
| MarkerAnnotation | 标志物图形画法 |
| ShapeAdapter | 矢量图形匹配画法转化器 |

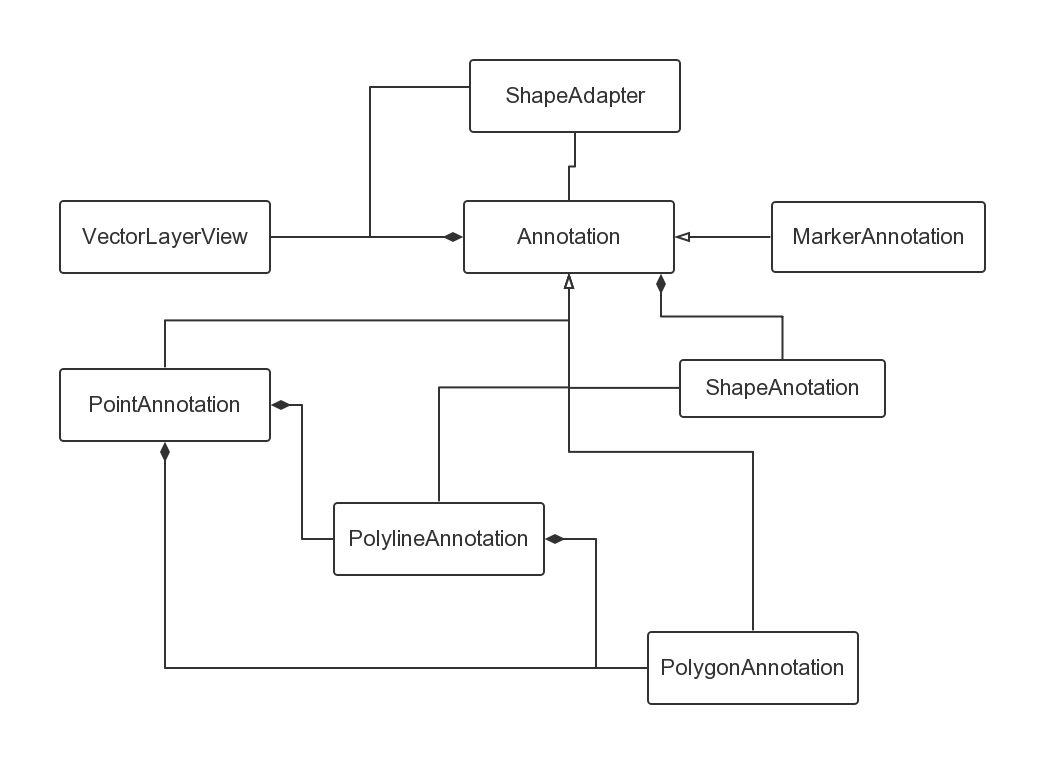


图 4-6 矢量图层类关系

Figure 4-6 vector layer class

### 矢量图层绘制流程

矢量图层绘制流程由获得图上几何元素、加载画法和绘制三步骤组成。

1. 获得几何元素  
   通过矢量标绘模块获得当前图层上的几何元素，这部分将在矢量标绘模块具体描述。
2. 构建四叉树记录  
   利用四叉树，记录当前几何元素的位置。这个将在地图浏览中详细描述。
3. 加载画法  
   通过ShapeAdapter对象，将几何元素和画法类进行匹配，获得当前图层上的对应画法对象。
4. 绘制  
   通过画法在地图上渲染对应的几何元素

### 矢量图层绘制算法

* 算法应用技术
  + CAShapeLayer
  + UIBezierPath
  + CABasicAnimation
  + CAScrollLayer
  + CGRect
  + CGRectInset
* 算法类
* 算法绘制流程
  + 图形绘制
  + 动画处理

## 地图浏览接口

## 坐标转变换接口

## 图层管理

1. 矢量标绘模块
2. 数据管理模块
3. 位置服务模块
4. 案例验证
5. 结 论

本文的研究成果是设计并实现了基于WPF技术的GIS组件WPFMapTagControl，并基于WPFMapTagControl进行了GIS应用的开发，从而证明了WPFMapTagControl的GIS基本功能，也为以后开发基于WPF技术的GIS应用开发提供了方法和模板。

本文的具体成果包括：

应用WPF技术实现绘制功能，在GIS数据读取部分实现的基础上实现对GIS矢量数据绘制的功能，包括投影坐标转换、图层管理、用户交互、地图渲染、属性数据。这些GIS基本功能可以便捷的提供给二次开发人员一个成熟的GIS地图组件，通过接口的调用和配置即可应用GIS的地图功能。

在GIS地图基本功能实现的基础上设计扩展性强的图标标绘功能，对图标标绘功能进行封装，对外提供鼠标标绘、自动标绘系列接口，图标属性设置功能系列接口，获取图标属性功能系列接口，图标属性更改功能系列接口，图标查询功能系列接口、内部监听事件。标绘部分为之后的开发奠定了一种二次开发标绘功能的模式，即通过接口调用以及事件监听的方式实现标绘功能；也为标绘功能本身的扩展打下了在设计上的基础。

论文存在的不足及进一步工作：

GIS组件开发涉及的内容非常广泛，本文只是在GIS基本功能的开发做了一些研究，功能对比商业性GIS软件产品比较简单，在数据解析方面只能适应特殊行业，尚不能对商业的GIS数据进行解析，所以下一步的研究重点之一是在现有数据解析模块的基础上扩展其解析商业GIS数据的功能。

1. 参考文献

1 [刘明皓](http://www.amazon.cn/s/ref=dp_byline_sr_book_1?ie=UTF8&field-author=%E5%88%98%E6%98%8E%E7%9A%93&search-alias=books). 地理信息系统导论 重庆出版社. 2010.9~1

2 Ivaar Jacobson,Martin Griss, Patrik Jonsson. Software Reuse Architecture, Process and Organization for Business Success. In China Machine Press. 2003.1: 20~30

3 周顺平，王海龙. 基于MAPGIS组件的二次开发. 地球科学：中国地质大学学报. 2002: 45~46

4 地理信息系统概论. 高等教育出版社. 2008-4: 23

5 刘明德，林杰斌. 地理信息系统GIS理论与实务. 清华大学出版社. 2006.12. 1~18

6 Coppock J T, Rhind D W, The history of GIS. Longman Inc. 2004: 27~40

7 吴信才,等. 地理信息系统原理与方法. 电子工业出版社. 2002：1～4

8 鲍虎军等. 我国GIS技术与应用的现状和对策. 2005: 1~10

9 刘光. 地理信息系统二次开发教程组件篇. 清华大学出版社. 2002：23~34

10 MapX Developer’s Guide.MapInfo Corporation. 2000, 5：10~13

11 李连营，王涛等. 基于MapX的GIS应用开发.湖北.武汉大学出版社.2003: 10~20

12 蔡志明. 利用MapX控件在VB上实现GPS导航系统.福建电脑. 2006：5~10

13 蒋波涛. 插件式GIS应用框架的设计与实现. 电子工业出版社. 2008: 13~16

14 Michael Waltuch, Allan Lafram boise, etal.Exploring ArcObjects.ESRI, 2001: 53~60

15 尹建忠，何政伟，贺奋琴，基于ArcGIS的土地资源信息系统(LRIS)开发研究. 新疆石油学院学报. 2003: 76~79

16 蔡勒. 为我们的世界建模. 人民邮电出版社. 2004: 3~14

17 Adam Nathan. WPF揭秘. 2008: 345~367

18 李应保. WPF专业编程指南. 电子工业出版社. 2010: 230~243

19 余英, 梁刚. Visual C++实践与提高-COM和COM+篇.中国铁道出版社. 2001: 21~30

20 邹正华. 基于COM技术的GIS组件的研究和开发. 江西：南昌大学信息工程学院计算中心，2007: 90~98

21 GIS空间分析理论与方法. 武汉大学出版社. 2010: 5~9

22 PaulA.Longley, Geographical Information systems, Volume 1, Principles and Technical Issues, SecondEdition, 电子工业出版社, 2004, 9: 9~18

23 刘湘南. GIS空间分析原理与方法. 北京科技出版社. 2008: 8~19

24 张超. 地理信息系统实习教程. 高等教育出版社, 2003: 3~12

25 胡鹏，吴艳兰，李圣权，代琪编. GIS原理学习: 3~9

26 夏春林. GIS软件使用与开发. 中国矿业大学出版社. 2008: 7~20

27 蒋波涛. 插件式GIS应用框架的设计与实现. 电子工业出版社. 2008: 61~78

28 软件设计\_百度百科. http://baike.baidu.com/view/575391.htm?fr=ala0\_1

29 Ian Sommerville. 软件工程. 2007: 78~90

30 软件设计的基本原理. http://www.educity.cn/zk/rjgc/200609010841261605.htm

31 周国庆. DirectX游戏编程. 清华大学出版社. 2010: 120~123

32 Chris Andrade. Professional WPF programming : .NET development with the Windows Presentation Foundation. 2009: 398~400

33 丁士锋. 精通C#3.0与.NET3.5高级编程. 清华大学出版社. 2009: 167~180

34 Martin Fowler. 分析模式：可复用的对象模型. 机械工业出版社. 2010: 120~121

35 内森. WPF揭秘. 人民邮电出版社. 2008: 400~404

36 安德雷德. WPF高级编程. 清华大学出版社. 2008: 340~350

37 张晗雨. WPF全视角分析. 机械工业出版社. 2009: 411~413

38 王占全. 地理信息系统（GIS）开发工程案例精选. 北京人民邮电出版社. 2005: 9~14

39 王亚民. 地理信息系统及其应用. 西安电子科技大学出版社. 2006-6-1: 8~45

40 MSDN-VS2008. Ms-help://MS.MSDNQTR.v90.en/dv\_vswpfdesigner/html/feb6238d-8916-4d 94-8016-115ffded80c6.htm

41安德森. WPF核心技术. 人民邮电出版社. 2009: 84~98

42 毕晓普. C#3.0设计模式. 机械工业出版社. 2009: 57~78

43 弗里曼. Head first design pattern. 中国电力出版社. 2007: 98~102

44 丁士锋. 精通C#3.0与.NET3.5高级编程. 清华大学出版社. 2009: 85~104

1. 致 谢

感谢我的导师，李健老师以深厚的理论功底、严谨的治学态度、敏锐的科学洞察力和清晰合理的思维方式指导我开展研究工作，使我受益非浅。他在我的论文中，给我细致耐心的指导，也给我很多建议。他严谨的治学态度给我留下了深刻的印象，将在今后的工作、学习中一直激励着我。

感谢项目组的其他同学，在这个项目组中，每个人都有着很强的团队精神和专业技术，给我以很大的帮助。没有大家通力合作，就不可能完成这样大型的软件项目。

感谢研究生三年来每一个关心、帮助过我的老师、同学，我从他们身上不仅学到了知识，还有做人的道理，处事的原则。

感谢我的父母和朋友，在平时的学习、生活中对我的关心与照顾，使我充满了前进的动力。

感谢研究生科技办的王猛老师和常宝英老师在我日常的学习和生活中给予的帮助。

最后，感谢论文评审委员会的老师对我论文的指正和他们辛勤的工作。