
密级： 保密期限：

北京邮电大学

硕士学位论文



题目：基于移动协作感知平台的
数据可视化研究与设计

学 号：2012111367

姓 名：袁龙运

专 业：计算机科学与技术

导 师：龚向阳

学 院：网络技术研究院

2014 年 12 月 30 日

独创性（或创新性）声明

本人声明所呈交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：_____ 日期：_____

关于论文使用授权的说明

学位论文作者完全了解北京邮电大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京邮电大学。学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。（保密的学位论文在解密后遵守此规定）

非保密论文注释：本学位论文不属于保密范围，适用本授权书。

本人签名：_____ 日期：_____

导师签名：_____ 日期：_____

基于移动协作感知平台的数据可视化研究与设计

摘 要

通信、微电子、集成电路和软件等技术的快速发展,使得移动终端得到了迅速的普及,而智能移动终端设备庞大的用户群体结合其丰富的计算、感知和通信能力使得以移动协作感知(Mobile Collaborative Sensing)技术进行低成本大范围的感知数据收集成为可能。通过移动协作感知技术收集的环境、交通等数据,在环境保护、公共设施服务和提升全民健康水平等方面有着广泛的应用。而通过数据可视化(Data Visualization)技术对收集的基于地理位置的感知数据进行简单、直观的呈现,可以更容易的掌握数据分布、变化以及不同数据之间的关联,为我们解决相关的问题提供指导信息。

本论文在对移动协作感知背景和数据可视化相关技术进行分析的基础上,分析并设计了一个移动协作感知平台,并对其中的数据可视化系统进行了详细的设计和实现。该数据可视化系统能够在 web 界面对收集到的感知数据进行基于地理位置的可视化,包括对展示数据的兴趣点(POI, Point of Interest)在地图上的聚合,以及对收集到的基于地理位置的图片、PM 2.5 等感知数据的动态展示。

论文首先对移动协作感知的研究背景和相关的 web 端的数据可视化技术进行简要分析;之后对移动协作感知平台的架构进行设计,在此基础上对其中的 web 数据可视化系统进行需求分析和总体设计;接下来对数据可视化系统的各个子模块进行详细的设计和实现,包括在 web 前端对感知数据的可视化模块和服务器端的感知数据的处理模块;然后对数据可视化系统进行了功能性的测试,表明可视化系统达到了对感知数据进行可视化的预期目标;最后,论文对全文进行了总结,对本文的不足和下一阶段的工作进行简要介绍,并总结了作者在研究生期间的所有工作和成果。

关键词 移动协作感知 数据可视化 兴趣点

RESEARCH AND DESIGN OF DATA VISUALIZATION BASED ON MOBILE COLLABORATIVE SENSING PLATFORM

ABSTRACT

With the rapid development of communications, microelectronics, integrated circuits and software technology, the mobile terminal has been rapidly gaining popularity. Smart mobile devices combined its large user with rich computing, sensing and communication capabilities to mobile collaborative sensing technology makes sensing data collection with a wide range of low-cost becomes possible. By moving mobile collaborative sensing technology to collect environmental, traffic and other data, the data has a wide range of applications in environmental protection, public facilities improvement and national health condition enhancement. Through data visualization technique sensing data can be presented in a simple and intuitive way, so we can more easily grasp the changes and the association between different data, which can provide guidance for us to solve related problems.

In this thesis, based on the background of mobile collaborative sensing and related technologies of data visualization, a web-side data visualization system based on mobile collaborative sensing platform is designed and implemented. The system is able to visualize the collected sensing data in the web interface based on location, including the display of data POI (Point of Interest) polymerization on the map, as well as the dynamic display of collected location-based data, such as pictures, PM 2.5.

First, this thesis introduced the mobile collaborative sensing research background and related web-side data visualization techniques briefly; After that the architecture of mobile collaborative sensing platform is designed, on the basis of that the requirement analysis and system design

of data visualization system is made; Next, detailed design and implementation of each sub-module are depicted in detail, including web front-end data visualization module and server-side data processing module; Then the whole system functional tests show visualization system to achieve the desired design goals; and finally, the summary the whole work, the inadequate and the future work of this thesis are briefly described, and the summary all of the work and achievements of author during the graduate students are given.

KEY WORDS: mobile collaborative sensing; data visualization; point of interest

目 录

第一章 绪论	1
1.1 研究背景	1
1.2 拟解决问题	2
1.3 论文主要工作	3
1.4 论文结构	4
第二章 相关概念及技术	6
2.1 数据可视化	6
2.1.1 地理位置数据可视化技术	7
2.1.2 图片数据的可视化技术	8
2.1.3 数字的图表可视化技术	9
2.2 浏览器与服务器之间的通信技术	10
2.2.1 HTTP 请求应答模型	10
2.2.2 Ajax 技术	11
2.2.3 数据通信格式	12
第三章 基于移动协作感知平台的数据可视化系统的需求分析和总体设计	14
3.1 移动协作感知平台的需求分析和总体设计	14
3.1.1 移动协作感知平台的应用场景	14
3.1.2 移动协作感知平台的需求分析	15
3.1.3 移动协作感知平台的总体设计	16
3.2 数据可视化系统的需求分析	18
3.3 数据可视化系统的总体设计	20
3.4 数据可视化系统的交互的设计	21
第四章 数据可视化系统的详细设计与实现	24
4.1 数据库设计	24
4.1.1 用户信息表	24
4.1.2 POI 表	25
4.1.3 POI PM 表	25
4.1.4 图片信息表	26
4.1.5 数据库接口设计	27
4.2 服务器端的处理模块	28
4.2.1 POI 计算部分	28

4.2.2 POI PM 计算	30
4.2.3 响应前端请求部分.....	31
4.2.4 用户注册登录处理.....	32
4.3 web 前端展示模块.....	33
4.3.1 地图显示区域.....	36
4.3.2 图片显示区域.....	37
4.3.3 曲线图区域.....	39
4.3.4 柱状图区域.....	40
第五章 数据可视化系统的测试.....	42
5.1 测试环境.....	42
5.2 测试目的.....	42
5.3 测试方案及过程.....	43
5.3.1 服务器端的处理模块测试.....	43
5.3.2 web 前端展示模块测试.....	44
5.4 测试总结.....	53
第六章 结束语.....	54
6.1 全文总结.....	54
6.2 不足和下一步工作.....	54
6.3 研究生期间的工作.....	55
6.3.1 参加的项目.....	55
6.3.2 完成的文档.....	56
参考文献.....	57
致 谢.....	59
作者攻读学位期间发表的学术论文目录.....	60

第一章 绪论

1.1 研究背景

得益于通信、电子和软件等技术的不断发展，移动终端得到了迅速的普及，移动电话也由最初只具有通话功能发展成为集成了强大的计算、感知和通信能力的智能移动终端设备。移动终端领域的技术的最新进展以及移动终端的普遍存在促使科研人员提出了一种新型的数据获取方式——移动协作感知（Mobile Collaborative Sensing）^[1-4]，使得我们可以用很低的成本进行大范围的感知数据的收集。移动协作感知的基本思想是由参与者利用手机等移动终端设备采集并共享基于位置的感知数据，并基于这些采集到的数据提供各种新型服务。

与传统的传感器网工作方式不同，移动协作感知不需要预先部署特定传感设备，它将普通手持/车载智能终端用户设备所嵌入的多种传感器作为基本感知单元，保证了信息获取的多元化与及时性，实现了感知任务（来源于任务需求方）与感知动作（来源于任务参与者）的充分互动，感知数据的有效收集和分析，进而形成具有明显交互特性的“移动协作感知网络”。利用有意识或无意识的用户参与，感知系统具有了覆盖面广、数据丰富、不需要额外硬件投资等优点，能够向基于移动协作感知网络的应用提供全面、便捷、深度的数据，在完成大规模、复杂的社会感知任务方面拥有广阔的应用前景。

近年来，国外已经有大量关于参与式感知技术的应用和研究。比较著名的参与式感知研究小组包括美国麻省理工大学 SENSEable City Lab、加州大学洛杉矶分校 CENS Lab、美国达特茅斯大学 Mobile Sensing Group、美国哥伦比亚大学 Sensor Networks Group、IBM 研究院、微软研究院等。

我国国民经济和社会发展“十二五”规划中强调要加大环境保护力度，建立健全公共服务体系等。考虑到我国城市环境中具有人口密度大和移动设备普及度高的特点，移动协作感知技术部署成本极低，在环境保护、公共设施服务、提升全民健康水平和灾难救援等方面有着广泛的应用前景。例如在环境保护方面，收集参与者上传的环境感知数据或图片，获取城市细粒度实时空气质量信息；灾难救援中，通过参与者提供的受灾情况照片及位置信息等，分析灾害在不同区域所造成的损失，为灾难救援决策的制定提供指导。

而对通过移动协作感知技术所收集到的数据的查看需要用到数据可视化的相关技术,通过对数据的可视化展示,我们可以以直观的图形的方式了解分析我们收集的数据。

我们对数据可视化的理解可以这样来表述:通过特定的可视化工具和技术以地图、图表、标签、动画或任何可以使内容更容易理解的图形方式来进行数据呈现。同时数据可视化技术不仅能够帮助我们完成数据的最终呈现,而且对发现数据中新的信息也能够起到非常关键的作用。我们可以这样认为,数据可视化的本质就是借助图形化手段,清晰有效地传达我们想要的信息,并使通过数据表达出来的内容更容易被理解。

为了实现信息地有效传达,数据可视化需要做到兼顾美学形式与实现功能两个方面,通过直观地形式表现出数据的关键特征,实现对数据集的深入了解。要达到数据可视化效果的完美呈现,需要做到内容充实、直观高效、形式新颖并且符合审美这四个方面。

随着新的数据可视化技术和工具的出现,数据可视化可以实现用户与数据之间的交互。新的数据可视化技术的应用标准应该向以下 4 个方面努力:1) 直观性:直观、形象地对数据进行呈现;2) 关联性:挖掘需要呈现的数据之间的关联;3) 艺术性:增强数据呈现的艺术效果,符合审美规则;4) 交互性:可以实现用户与数据的交互,增强用户对数据的控制。其中,直观化和关联化强调数据可视化的功能性,艺术性和交互性则强调数据可视化的美学设计和智能化。达到了直观化和关联化,就能够对数据进行直观、形象地呈现,进而找出数据之间独特的关联性,而智能、交互性强的动态的数据可视化过程可以实现信息的传输和双向沟通^[5]。

1.2 拟解决问题

移动协作感知是通过参与者的智能移动终端的传感器进行感知数据的收集并通过网络将收集到的感知数据上传给服务器,在服务器端进行存储、分析和处理。而数据可视化可以将经过分析处理的数据直观、形象的展示出来,让我们更容易了解收集数据的分布以及变化规律。

基于上述研究目标,本论文主要解决以下几个问题:

(1) 移动协作感知平台的设计

移动协作感知平台的系统架构与传统无线传感器网络基本相同,传统无线传感器网络通常包括感知、网络传输、数据处理、应用四个层面:感知层通过各种传感技术获取有用的感知信息;网络传输层通常包括接入和传输,通过结合各种

通信和网络技术,将各种分散的来自感知层的信息接入现有通信基础设施,最终传送到应用系统服务器;数据处理层利用机器学习,数据挖掘等技术,从收集到的感知数据中获取所需要的信息,然后进行分析、储存、管理等工作,再反馈给关联对象;应用层面具有智能信息处理的特性,帮助进行智能的决策和调节控制。

我们需要针对移动协作感知平台的每一层功能,对其总体架构进行具体设计,以使该平台符合我们进行感知数据收集、传输、存储和管理的要求。

(2) 数据可视化系统的研究与设计

移动协作感知平台在应用层可以利用处理后的感知数据开发各种各样的应用,而数据可视化系统作为应用层的应用,需要对收集的感知数据进行 web 端的展示,包括对感知数据的地理位置信息在地图上的聚合展示,对收集的图片、光照、噪声以及通过图片分析的 PM 2.5 等数据的变化展示。

在感知数据的可视化过程中需要解决如下几个问题:如何从服务端获取需要展示的数据;数据从服务端传输到 web 端的格式;以及感知数据在 web 端的展示形式。

1.3 论文主要工作

本论文的主要任务就是基于我们设计的移动协作感知平台,对通过移动协作感知平台的客户端收集的各种感知数据进行分析处理之后,在 web 前端进行各种形式的可视化展示。具体的工作内容包括以下三点:

1. 查看移动协作感知技术相关文献资料,研究移动协作感知技术提出的背景和研究现状。

移动通信业务和互联网相关业务作为当今世界发展最快、市场潜力最大的两大业务,它们的增长速度都是任何人没有预料到的。截止到目前,全世界的移动用户数量已接近世界人口的总数,而互联网用户的数量也已经超过 27 亿^[6]。现在有超过数亿人每天携带智能手机。这些随处可见的设备利用交互式或自主式的方法,能够收集越来越多的数据,并且能够对数据信息进行分类:如图像信息,声音信息,位置信息和其他数据信息。这些移动设备跨越了多种应用领域,使得它们互相关联。如:医疗保健,安全,环境监测,数据传输等应用领域,并且形成了一个新的研究领域,即移动感知。在手机中装上传感器,以这种方式就可以收集到大量的数据。例如:可以利用移动感知来追踪并监控多个数据点,并且可以提供有关环境监测的动态信息,以此来了解城市的交通状况。Burke 在 2006 年明确提出移动感知的概念^[7],他描述了一个可以提高数据的质量、可信度、安全性和共享性的体系架构。

参与式感知(Participatory Sensing)是目前移动物联网领域的研究热点之一。在部分国外相关文献中, Volunteer Sensing^[8]、Public Sensing^[9]、Urban sensing^[10]、Mobile Crowdsensing^[11]、Community Sensing^[12]等也表示与参与式感知相近的含义。

2. 与项目小组成员协作, 对移动协作感知平台进行总体设计。

基于对移动协作感知平台的系统架构的分层, 在我们的移动协作感知平台上, 我们将对每一层进行对应的设计。

移动协作感知平台的客户端负责完成感知层的功能, 即通过调用移动终端上的传感器进行感知数据的收集并上传到服务器, 同时我们会根据终端的状态(电量、网络状态、充电状态等)进行收集和上传策略的调整以减少终端资源的消耗; 网络传输层的功能我们采用的是利用现有的移动数据网络或者 WiFi 网络等进行数据的传送; 感知平台的服务器会负责数据处理层面的功能, 包括对感知数据的接收、分析处理(感知网络节点的异构性导致数据的不准确、不真实, 必须进行相应的去除冗余、修正错误, 从而得到可靠的数据, 保证后续工作的持续进展)和存储到数据库的工作; 在应用层, 我们会通过 web 界面对经过分析处理的感知数据进行各种形式的可视化展示。

3. 数据可视化系统的设计与实现

数据可视化系统的主要功能是在 Web 端对移动协作感知平台参与者上传到服务器端的数据进行可视化展示。

在 web 端进行数据展示的数据可视化系统可以采用 B/S 结构进行实现。在服务器端利用 Java 中的 servlet 技术进行浏览器请求的响应和处理, 利用数据库接口进行请求数据的获取; 而数据的传输可以采用基于 HTTP 的 Ajax 技术传输 JSON 字符串来完成; 浏览器端数据的请求可以通过 HTTP 获得, 而对获取的不同类型的数据要根据其数据特点进行不同形式的可视化展示: 感知数据的地理位置信息可以通过地图上 POI 进行聚合展示, 而图片信息可以通过幻灯片的切换进行直观的展示, 图片的 PM 2.5、POI 的 PM 2.5、POI 的光照、POI 的噪声等数据信息可以通过曲线或者柱状图等形式进行展示。

1.4 论文结构

本论文的结构和章节安排如下:

第一章为绪论。本章首先阐述本论文的研究背景, 然后对本论文的主要研究内容和本论文的组织结构进行介绍。

第二章对本论文涉及到的可视化技术进行分析。本章首先对数据可视化概念

进行介绍，然后对本论文使用到的数据可视化相关技术进行逐一介绍：对收集数据的位置信息进行展示的 Google Map 技术，用于实现图片幻灯片展示功能的 jQuery 库技术，用于感知数据及相关计算数据进行图形展示的 Highcharts 技术以及用于前端页面与服务器进行数据通信的 Ajax 技术。

第三章对基于移动协作感知平台的数据可视化系统的需求分析和总体设计进行阐述。本章首先对移动协作感知平台进行需求分析和总体设计，然后对数据可视化系统与移动协作感知平台的关系和交互进行论述，最后对数据可视化系统的总体以及各个模块分别进行设计。

第四章将详细介绍数据可视化系统的详细设计和实现细节。本章将分别对数据可视化各个模块进行详细设计并实现，数据可视化系统主要包括进行 web 前端展示的模块和服务端端的负责处理前端请求的模块。本章还会对前端展示所涉及到的数据在数据库中存储的表的结构进行介绍。

第五章将对基于移动协作感知平台实现的数据可视化系统进行测试。本章将对测试的目的、测试的环境以及测试的过程和结果分析进行详细的阐述，验证所实现的系统的正确性。

第六章总结全文。本章将对全文进行总结，并指出目前工作的不足之处和下一阶段的工作目标。

第二章 相关概念及技术

本章将在论文研究背景和研究内容介绍的基础上,对论文涉及的数据可视化的概念进行介绍。而由于我们的可视化系统的展示是基于 web 页面的,我们将对用到的 web 端的数据可视化技术以及浏览器与服务器之间的通信技术进行阐述分析。

2.1 数据可视化

数据可视化技术起源于二十世纪六十年代的计算机图形学,人们使用计算机进行图形和图表的创建,完成对提取出来的各种数据的可视化,将数据的各种属性和变量展示出来。计算机硬件的快速发展,促使数据采集设备和保存设备的快速发展,帮助人们创建出更复杂、规模更大的数字模型。而为了创建这些规模庞大的数据集,我们就需要更高级的计算机图形学方面的技术和知识。数据可视化应用领域的不断增加,展示平台的拓展,表现形式的不断变化,以及诸如实时动态效果、用户交互使用等效果的不加入,促使数据可视化概念的边界不断扩大。

饼状图、直方图、曲线图以及柱状图等是我们最熟悉也是最原始的统计图表,这些图表是数据可视化技术最常见的应用。作为统计学工具,数据可视化技术给我们提供了一条快速认识数据集的捷径,使他成为一种令人信服的数据表现形式。我们可以在很多的 PPT、报表、解决方案以及新闻报刊中见到统计图形,它们可以传达出存在于数据中的基本信息。但是,这些最原始的统计图表只能呈现出基本的信息或者说对定量的数据进行可视化。

在面对十分复杂或大规模的异构数据集时,比如财务报表、商业数据分析、人口分布情况、用户行为数据等,数据可视化技术所面临的状况会复杂得多。可能要经历包括数据采集、数据管理、数据分析或数据挖掘等在内的一系列复杂的数据处理流程,然后再由设计师来设计一种可视化形式,是立体的还是二维的、动态的还是静态的、实时的或者是允许交互的。然后由工程师设计对应的可视化算法,确定可视化技术的实现手段,可能包括数据建模方法、能够进行大规模数据处理的体系架构、涉及到的动态交互技术等。如果涉及到动画,动画工程师还要考虑表面材质、动画的渲染方法等问题,而交互设计师也会加入,对用户的交

互模式进行设计。

数据可视化系统的开发同其他类型的系统的开发模式一样,需要根据系统的目的和用户群来选用不同的表现形式。对于同一份数据,我们可以将其可视化成多种截然不同的形式。对于一份不停变化、可读性强的图表,它的可视化目标可能是为了对数据进行观测、跟踪,所以需要强调实时性、变化以及运算能力;而对于一份可以检索的交互式的图表来说,它的目的可能是为了进行数据分析,所以要强调数据的呈现度;我们可以生成分布式的多维的图表来发现数据之间的潜在关联;也可以利用丰富的颜色和动画创建生动、明了、具有吸引力的图表,从而帮助用户快速理解数据的含义或变化。

2.1.1 地理位置数据可视化技术

得益于智能移动终端和定位技术的快速发展,在我们进行感知数据的收集时,可以同时获得所采集到的数据的地理位置信息。在对感知数据进行可视化时,可以同时为地理位置进行可视化,从而将地理位置和感知数据联系起来,可以更直观的了解感知数据的分布情况,并可以对同一位置的感知数据随时间的变化情况进行分析。

在 web 端对表示地理位置的经纬度信息的可视化最好的方式是通过在线地图对该经纬度信息在地图上进行标记。

Google 地图提供丰富的 API 可供开发者使用,开发者可以利用该 API 开发满足自己需求的基于地理位置的地图应用,并可以提供世界范围内的地图数据,可以满足我们的项目对收集的感知数据的地理位置信息进行展示的需求。

Google 公司推出 Google 地图,可以向全世界的用户免费提供地图查看、地点搜索以及逐级缩放的地图服务。而为了使自己的地图服务得到更广泛的应用,Google 还向 web 应用开发者提供了开放式的地图服务应用程序接口(Google 地图 API)^[13]。借助 Google 地图的 API,开发者可以在自己的 web 应用程序中嵌入 Google 地图,从而让世界上所有对基于地理位置的服务感兴趣的人都能够自主开发基于 Google 地图服务的应用。

Google 地图的特点如下^[14]:

- 1) 地图操作简单。Google 地图的操作非常简单,可以对其自由移动和缩放。
- 2) 地图的预生成功能。在服务器端地图会被预先处理成小的图片碎片,切块后进行编码并保存。当对地图进行移动或者缩放操作导致地图可视范围的变化时,只需要下载新显示的区域所对应的图片,在此利用了 Ajax 技术以及浏览器的多线程下载功能。另外,已经下载过的图片会被浏览器缓存起来,再次在地图上显示时不用重新访问服务器进行下载,这样既能快

速的进行地图展示，还可以节省用户的带宽和资源。

- 3) 路线查询功能。在 Google 地图上还可以进行距离测量、路径分析以及路线查询等操作。

Google 地图 API 是 Google 公司向应用开发者提供的免费地图 API。使用 Google 地图的 API，应用开发者不用搭建自己的地图服务器来存储地图数据，而是直接将 Google 地图的数据嵌入到自己的网站中，利用 Google 地图的数据为自己网站的用户提供位置服务。

Google 地图 API 除了可以帮助 web 应用的开发者利用 JavaScript 将 Google 地图嵌入到自己的 Web 应用中之外，还允许开发者利用 JavaScript 进行应用拓展的开发。该 API 提供了大量的操作工具对地图进行处理，同时可以通过简单并且丰富的接口函数向地图中添加自己想要的内容，从而使开发者能够在自己的 web 网站上创建功能非常全面的地图应用。例如给地图添加定制的图标、叠加层，或者响应用户在地图上的点击或者缩放的操作。

2.1.2 图片数据的可视化技术

移动协作感知平台需要收集环境相关的数据，其中包括 PM 2.5 值，但是移动终端不具备获取 PM 2.5 值的传感器，所以需要通过使用相机拍摄照片然后分析该地点的 PM 2.5 值，在此过程中会拍摄大量的图片，而对这些图片的展示会让用户对该地点的 PM 2.5 值有一个直观的认识。所以数据可视化系统需要对采集的图片信息进行可视化展示，而比较好的展示方式是对某一地点的图片进行幻灯片形式的切换展示。

jQuery 作为开源函数库，包含详细的使用文档和开发说明^[16]。使用 jQuery 可以很方便的实现各种动画效果，处理 HTML、events 事件，还可以为网站提供 Ajax 交互。基于 jQuery 的以上特性，我们可以使用 jQuery 技术开发用于图片切换展示的插件。

jQuery 是一个轻量级的 JavaScript 函数库，对于使用 JavaScript 语言进行编码的程序员来说，它是一个非常有用的函数库。作为 JavaScript 新手，jQuery 能够帮你解决文档对象模型(Document Object Model, DOM)脚本和 Ajax 开发中的复杂问题，而作为一个资深的 JavaScript 专家，jQuery 则可以帮你减少 DOM 脚本和 Ajax 开发中无聊的重复工作。利用 jQuery 丰富的函数库，程序员可以减少 DOM 脚本库的调用以及重复代码的编写，而只关注程序的逻辑实现。利用尽可能少的代码实现尽可能多的功能，正是 jQuery 的核心原则。

jQuery 作为一个 JavaScript 的开源函数库，与使用原生的 JavaScript 进行编码相比，具有以下优点^[15]：

- 1) 对各种浏览器的兼容。由于各种各样的原因,浏览器对 JavaScript 语法的支持是有区别的,而使用 jQuery 则可以将这种差异隐藏起来。对于 jQuery 的用户来说,这种由浏览器的实现不同造成差异对用户是不可见的,用户不用关心使用 jQuery 编写的程序将在哪种浏览器下运行,也可以说, jQuery 是跨浏览器的。它支持的浏览器包括 Chrome、IE 6.0+, FF 1.5+, Safari 2.0+, Opera 9.0+等各种主流浏览器。
- 2) 前端页面行为与样式的分离。利用 jQuery 能将前端页面中用于控制页面行为的 JavaScript 代码和用于控制页面样式的 HTML 代码分离开来,便于代码的维护。即通过 js 文件引用, jQuery 代码和 HTML 代码可以分别编写,这样不但可以提高 HTML 代码的可读性,也方便前端页面的功能实现和维护。
- 3) jQuery 支持 CSS1-CSS3 的全部语法,以及基本的 XPath。利用 jQuery 不但可以实现对不同版本的样式库的支持,还可以通过对 XPath 的解析实现对 XML 的支持。
- 4) 简单易学。jQuery 库代码简练、文档丰富,非常便于学习。原生 JavaScript 的广泛使用会导致代码重复从而导致代码的可读性的降低, jQuery 正是针对 JavaScript 的这个缺点编写的。jQuery 可以通过强大的函数封装功能来大大减少程序员编写的代码数量。
- 5) 便于扩展。作为开源的函数库, jQuery 提供的接口可以免费使用,来实现自己的扩展功能,并将自己实现的扩展功能提供给他人使用。而自己也可以免费使用他人公开的扩展插件,提高自己编写的代码质量。

2.1.3 数字的图表可视化技术

移动协作感知平台收集的环境相关数据除了 PM 2.5 值,还包括光照、噪声等数据,而这些数据都是以数字的形式采集并存储的,所以对这些数据的展示要采用与图片不同的形式。在 web 端对数字格式的数据进行展示可以采用曲线或者柱状图的形式,同时曲线图还可以表示一段时间的变化趋势,有助于了解环境数据的变化情况。

Highcharts 是一个使用 HTML5/JavaScript 技术实现的图表库,能够让 web 应用开发者很方便的在自己的 web 应用程序中添加不同类型的交互性图表,包括直线图、曲线图、面积图、柱状图、饼状图等。并且免费提供给个人学习以及非商业用途使用^[17]。我们可以使用 Highcharts 来进行 web 端的数字数据的图表展示。

Highcharts 适用于所有的主流浏览器,如: IE、Firefox、Chrome、Safari,甚

至包括 iPhone/iPad 和 Android 等移动设备上的浏览器。并且，在 iOS 和 Android 操作系统上，对多点触控的支持可以提供无缝的用户体验。Highcharts 在标准浏览器使用 SVG 进行图形渲染，而在老版 IE 中的图形使用 VML 进行绘制。

Highcharts 的配置使用也非常的简单。它的运行只需要引入两个 js 文件，即 Highcharts.js 以及 jQuery、MooTools、Prototype、Highcharts Standalone Framework 几个框架中的一个框架文件。而这两个文件的引入方式有两种，一种是在线引入 js 文件的方式：

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.js"></script>
<script src="http://code.highcharts.com/highcharts.js"></script>;
```

另一种是将所需要的 js 文件下载到本地，通过引入本地文件的方式将文件引入项目：

```
<script src="js/jquery.js"></script>
<script src="js/Highcharts.js"></script>。
```

Highcharts 也提供更换图表主题和图表导出等功能。要实现这些功能需要引入额外的 js 文件。Highcharts 也提供将图表导出为常见图片文件或打印图表的功能，而使用该功能，只需要引入 Highcharts 文件库中的 exporting.js 文件即可。

2.2 浏览器与服务器之间的通信技术

2.2.1 HTTP 请求应答模型

HTTP^[18] (HyperText Transfer Protocol, 超文本传送协议) 是互联网上应用最为广泛的协议。它定义了浏览器向网络上的服务器请求资源以及服务器返回给浏览器资源的方式。通过 HTTP 协议请求的资源由统一资源标示符来标示，HTTP 是应用层的协议，它是万维网进行可靠文件交换的重要基础。

HTTP 协议的主要有以下特点：支持 B/S (Browser/Server, 浏览器/服务器) 模式，应用广泛；向服务器发起请求时只需传递请求方法和路径。通常采用 POST 和 GET 两种方式，两种方式规定的客户端与服务器交互方式不同；HTTP 的服务器程序规模很小，通信速度快；传输类型可以通过 Content-Type 标记，传输对象灵活可变；无连接协议，每次链接只能处理一个请求，服务器处理完客户端请求并接收客户端应答以后即断开连接，从而节省传输时间；对事物缺乏记忆功能，无状态协议，因此信息若需要后续处理则不需进行重传。

由于它的简单性、快速性和灵活性，在系统设计时服务器与客户端间的通信采用了该协议。它的工作过程如图 2-1 所示，客户端需要向服务器请求资源或上传数据，则对服务器发起请求 (get/post)，服务器则向客户端返回对该请求的应

答，返回的资源可以有很多表达方式（图片、文字、富媒体等）。本论文在数据上传和客户端验证的过程中都采用了 HTTP 协议进行通信。

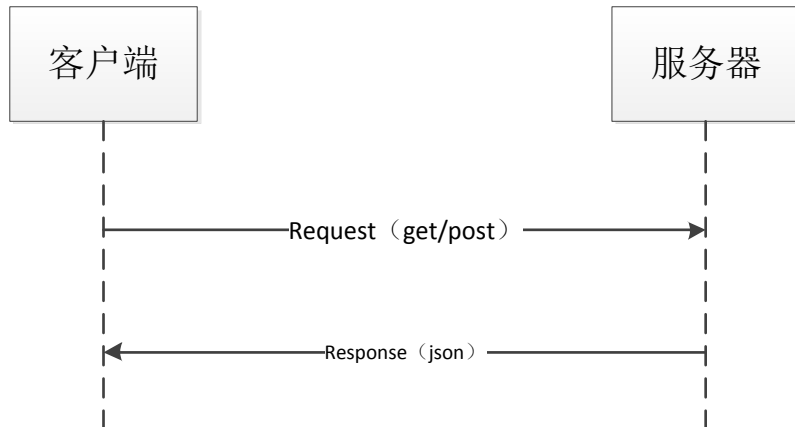


图 2-1 HTTP 请求应答模型

2.2.2 Ajax 技术

Ajax (Asynchronous JavaScript and XML)，意思是异步 JavaScript 和 XML。使用 Ajax 技术可以构建丰富的 Web 应用，使得在线的 Web 应用与桌面应用程序有相同的使用体验。Ajax 是对几项现有技术的结合，主要包括基于 XHTML 和 CSS 标准的表示；使用 DOM 进行动态显示和交互；使用 XMLHttpRequest 对象与服务器进行异步通信；使用 JavaScript 绑定一切元素。

在传统的 Web 应用中，用户与服务器采用同步的方式进行交互，其交互流程如下：用户首先向服务器发送一个 HTTP 请求，然后，服务器处理该请求并向发送请求的用户返回用户所请求的页面。可以看出，这个过程会导致不连贯的用户体验：服务器在处理用户请求的时候，用户是处于等待状态的，而在等待服务器的返回期间屏幕是不会显示任何内容的，这会严重影响用户的使用体验。

而与传统的 Web 应用不同，Ajax 技术采用异步交互过程。这种交互模式会在用户与服务器之间引入一个中间处理单元，用来改变用户与服务器交互过程中处理-等待-处理-等待的流程。在访问采用 Ajax 技术实现的网站时，浏览器在执行任务时会加载一个能够处理 Ajax 请求的引擎。该引擎主要用于处理用户界面与服务器之间的交互，采用 JavaScript 语言编写，并且通常位于一个隐藏的框架中。Ajax 引擎允许用户与应用软件之间进行异步的交互，而该交互过程是独立于用户与网络服务器之间的交流的。

Ajax 技术的核心是基于 JavaScript 实现的 XMLHttpRequest 对象，利用该对象可以完成异步请求。开发人员可以使用 XMLHttpRequest 对象向服务器发送请求并接收和处理响应，而不会阻塞用户的其他应用。我们可以用 JavaScript 调用

Ajax 引擎来来执行那些不需要重新载入整个页面的需求，比如内存中的数据编辑、页面导航等功能。

传统 Web 应用与基于 Ajax 技术的 web 应用的模型对比如图 2-2 所示^[19]。

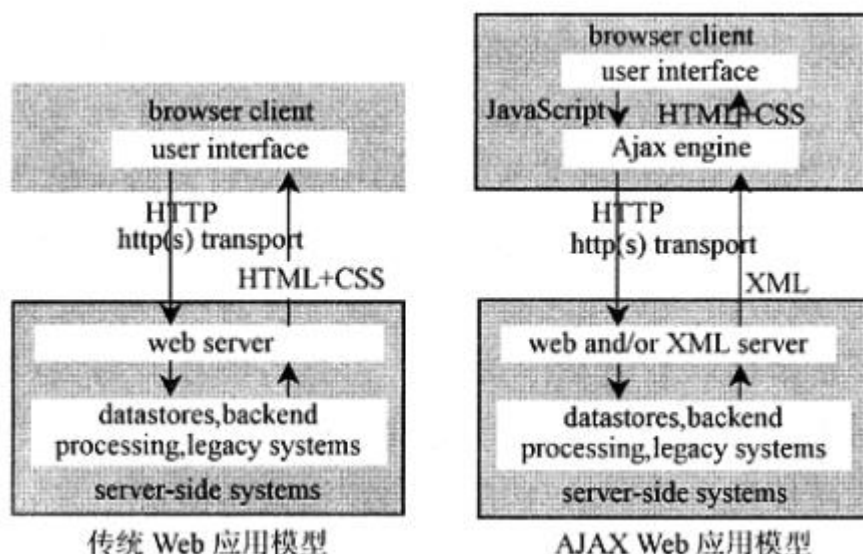


图 2-2 两种 Web 应用模型的对比

基于 Ajax 技术的 Web 应用模型可以为用户提供很大的灵活性，在客户端执行屏幕内容更新时，具有以下优势：

- 1) 可以减轻服务器的负担。Ajax 技术“按照需求请求数据”的原则可以在很大程度上减轻冗余请求对服务器造成的压力。
- 2) 不用刷新整个页面即可对页面内容进行更新，减少用户的等待时间。Ajax 使用 XMLHttpRequest 对象向服务器发送请求并得到响应，能够在不重新载入整个页面的情况下获取所需的更新数据，而只有当数据完全被接收到之后才会更新相应部分的内容。用户几乎感觉不到这个瞬间更新的过程。
- 3) 使用 Ajax 技术可以减轻服务器和带宽的负担。能够把服务器负责的工作转移一部分到客户端上执行。
- 4) Ajax 技术有自己的标准，是一种被广泛支持的技术，而且无需下载任何插件即可使用，可以给用户带来很好的体验。

我们可以在很多应用中使用 Ajax 技术，主要是数据分类良好、交互较多、需要频繁读数据的 Web 应用。

2.2.3 数据通信格式

JSON^[20] (JavaScript Object Notation) 是一种轻量级的数据交换格式。由于 JSON 是基于 JavaScript 的一个子集，这意味着在 JavaScript 中处理 JSON 数据不

需要加载任何特殊的工具包（除了像 IE 6/7 这样的旧版本浏览器不支持 JSON 的解析和序列化，需要加载 json2.js 库）。相比 XML，JSON 容易阅读、解析速度更快、占用空间更少，C、C++、Java、Python、Ruby 等语言都有相应的 JSON 库支持。JSON 的这些特性使得它成为了前后端之间理想的通信格式，也是未来的流行趋势。

数据可视化系统在设计中使用 Ajax 通信方式，制定 JSON 格式的 API，不仅减少了单次数据传输字节流的大小，提升了用户体验，同时也便于扩展，让用户能够感受到更多由 Web 2.0 所带来的新特性和功能。

第三章 基于移动协作感知平台的数据可视化系统的需求分析和总体设计

本文设计的数据可视化系统是基于移动协作感知平台的并且作为移动协作感知平台的一部分。本章首先对我们搭建的移动协作感知平台进行需求分析和总体设计,并对平台的模块组成和数据可视化系统在移动协作感知平台中的位置进行设计,然后对数据可视化系统进行需求分析,最后针对数据可视化系统需求分析的结果对数据可视化系统进行总体设计。

3.1 移动协作感知平台的需求分析和总体设计

3.1.1 移动协作感知平台的应用场景

硬件的高速发展使得智能手机内置了丰富的传感器,给移动协作感知提供了良好的硬件基础。智能手机近些年得到飞速的普及,给移动协作感知提供了良好的用户基础。设计合适的移动协作感知方案,智能手机就可以作为传感器节点和移动感知数据收集工具。移动协作感知系统的应用场景如图 3-1 所示。凡是包含内置传感器并且能够连接网络的设备,都能够成为移动协作感知的参与者。

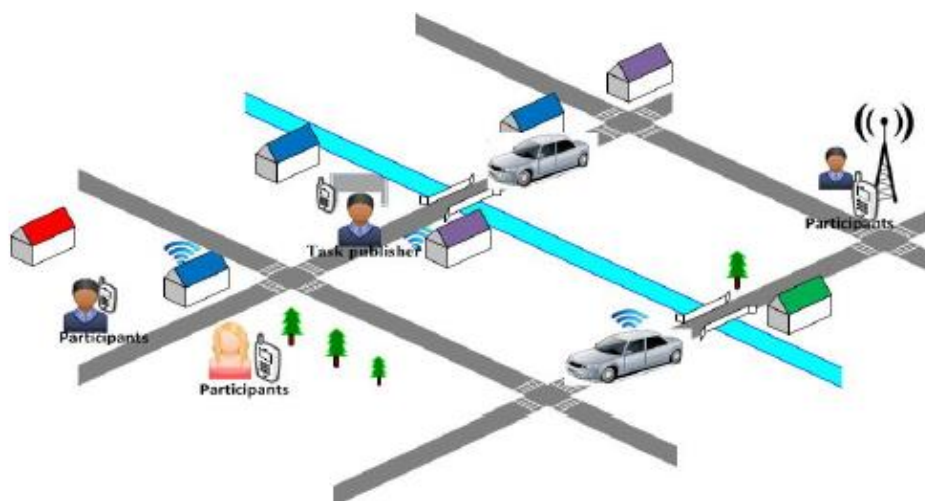


图 3-1 移动协作感知的应用场景

移动协作感知平台的组成如图 3-2 所示。它由三部分构成:任务发布者,参

与者和服务器平台。任务发布者通过移动客户端和 Web 端发布任务信息（所需数据类型、数据的位置等信息）以及相应的激励信息。服务器在接收该任务以后，通过多播或者广播的方式，将任务信息发布给参与者，并且从所有具有参与意愿的用户中选取若干个参与者完成本次任务。参与者在接收任务后，到指定位置采集指定类型的传感数据，通过网络将数据上传到服务器。服务器通过对数据进行分析，给出激励。

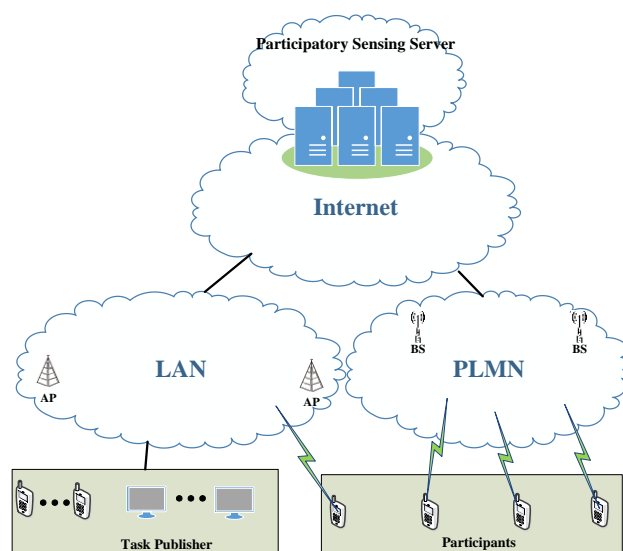


图 3-2 移动协作感知平台

3.1.2 移动协作感知平台的需求分析

本论文所设计的移动协作感知平台是移动设备采集传感器数据，对数据进行上传，在服务器上对数据进行存储和分析，用户通过上传数据获得奖励，并且能够对用户所拍摄的图片进行分析，得出该场景所对应的 $PM_{2.5}$ 值的系统。根据上述需求分析中所提到的功能性需求和非功能性需求，整个系统的网络拓扑设计如图 3-3 所示。

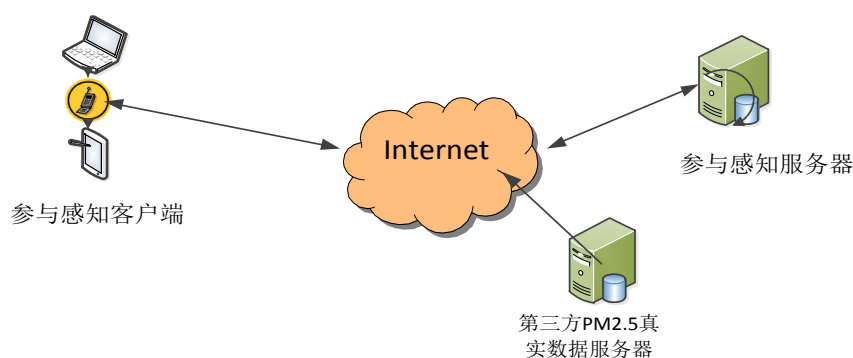


图 3-3 系统拓扑结构

移动协作感知平台主要包括客户端、服务器端和第三方 $PM_{2.5}$ 数据服务器

三大部分，其中：

1) 客户端包括移动客户端和 web 端。移动客户端主要负责数据的收集和上传至服务器的工作，包括感知数据，如图片、光照、噪声等数据以及收集到的感知数据的地理位置信息，还有与安装客户端的终端有关的状态信息如终端的网络状况、电量、充电状态等信息，客户端同时还负责对收集的图片进行分析以得到拍照地点的 PM 2.5 值以及对部分收集数据的展示工作；web 前端主要负责移动协作感知平台用户的注册登录、个人信息管理以及个人上传数据的分析展示，并对经过服务器处理、存储在数据库中的数据在 web 页面上进行可视化的展示，以方便用户通过收集到的数据对数据收集点的相关环境状况有清晰、直观的认识和了解。

2) 服务器端主要负责上传数据的分析与管理工作。包括数据的接收与存储、冗余数据的清洗与融合、缺失数据的补齐、用户轨迹数据的分析与预测以及当数据缺失时激励信息的发布和管理。

3) 第三方 PM 2.5 数据服务器主要是为了配合移动客户端的 PM 2.5 分析模块的需求。该服务器存放爬取到的指定区域内 PM 2.5 监测站点的监测数据。该数据将作为移动客户端 PM 2.5 分析模块的对比信息，通过误差对比，对移动客户端的模型进行训练和调整。

3.1.3 移动协作感知平台的总体设计

移动协作感知平台的主要任务就是搭建一个集数据收集、传输、存储、处理和展示于一体的平台，该平台通过在参与者的移动终端设备上安装的客户端软件来收集图片、噪声、光照以及地理位置信息等数据，然后通过无线网络传送到服务器，在服务器端对数据进行融合处理，最后将整理后的数据在 web 端进行展示。同时，该平台可以通过参与者的位置信息对其运动轨迹进行分析和预测，然后通过激励机制对参与者发出激励以使参与者到达数据缺失的区域帮助我们收集所需要的数据。据此，我们设计出移动协作感知平台如图 3-4 所示的系统功能框架图。

客户端主要由用户 UI、控制模块、采集模块、图像模块、数据库模块以及通信模块组成。用户在客户端 UI 界面上进行操作，在控制模块的逻辑控制下，客户端通过数据采集模块进行图片、光照、噪声以及地理位置信息等数据的采集，同时，图像处理模块会对收集的图片进行分析从而得到所采集到的图片的 PM 2.5 值。采集和分析得到的数据首先会存储在客户端数据库中，然后客户端会根据当前数据采集终端的网络和电量状况等信息采取相应的上传策略调用通信模块将采集和分析得到的数据上传至服务器端。客户端在进行数据展示时也会通过通信

模块向服务器请求相应的数据。

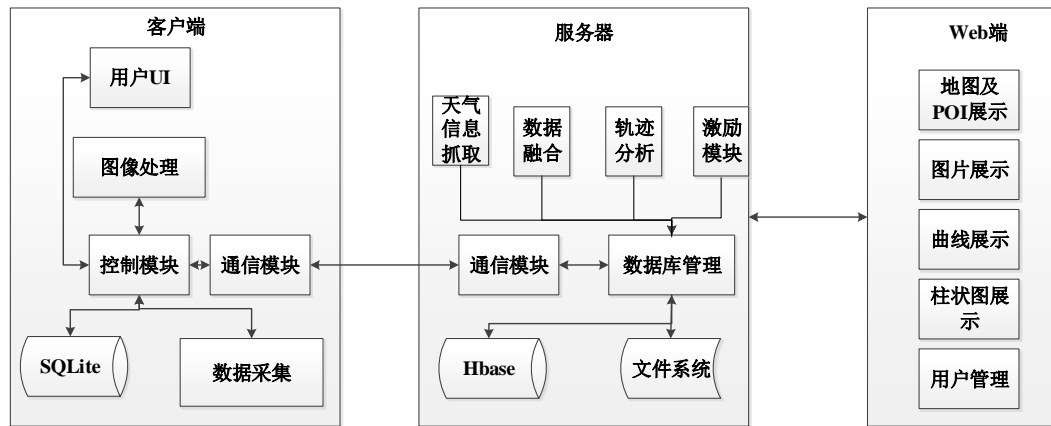


图 3-4 移动协作感知系统功能框架图

服务器端主要由通信模块、数据库管理模块、数据融合模块、轨迹分析模块以及激励处理模块组成。通信模块负责接收解析客户端的数据上传及下载请求，然后向数据库管理模块发送或请求数据。数据库管理模块通过分布式数据库进行相应的数据存储或返回客户端请求的数据。在分布式数据库将数据存储好后，数据融合模块会进行冗余数据的清洗与融合、缺失数据的补齐等工作，并把处理好的数据重新写入数据库，而通过算法无法补齐的数据会结合轨迹分析模块对收集数据的地理位置信息的分析和预测结果调用激励模块向可能出现在数据缺失地点的用户发出激励信息，激励用户采集并上传在这个位置我们所需要的数据。

浏览器主要负责 Web 前端的展示功能，包括移动协作感知平台的用户注册登录，查看个人基本信息以及上传的数据的分析展示，从分布式数据库中读取展示所需的数据并在 web 端对数据进行图表等形式的可视化展示。

根据系统的功能框架，将该系统的模块分为：表示层、应用层、控制层、存储层和物理层。如图 3-5 所示，图左侧模块为客户端模块，右侧则是服务器以及 web 展示模块。

表示层包括了客户端和服务器的展示模块，客户端通过用户 UI 进行展示，而服务器的数据则通过 Web 展示模块进行展示；应用层包含了客户端的 PM2.5 分析和服务器的各个算法模块，应用层的模块会对系统中的数据信息进行应用、分析和处理；控制层负责整个系统的逻辑控制和数据读写控制。在客户端逻辑控制与数据控制相分离，而服务器的控制模块主要是进行数据控制，各个模块间的逻辑关系通过远程调用来实现；存储层复杂系统数据的存储。客户端应用轻量级数据库 SQLite 进行数据存储，服务器端除了数据存储外还需要存储相应的照片信息，采用了分布式数据库 HBase 和文件系统对数据进行存储；物理层是最接近硬件的部分，负责通过读取硬件获得数据采集部分，通过数据通信模块将服务器与客户端相连接。

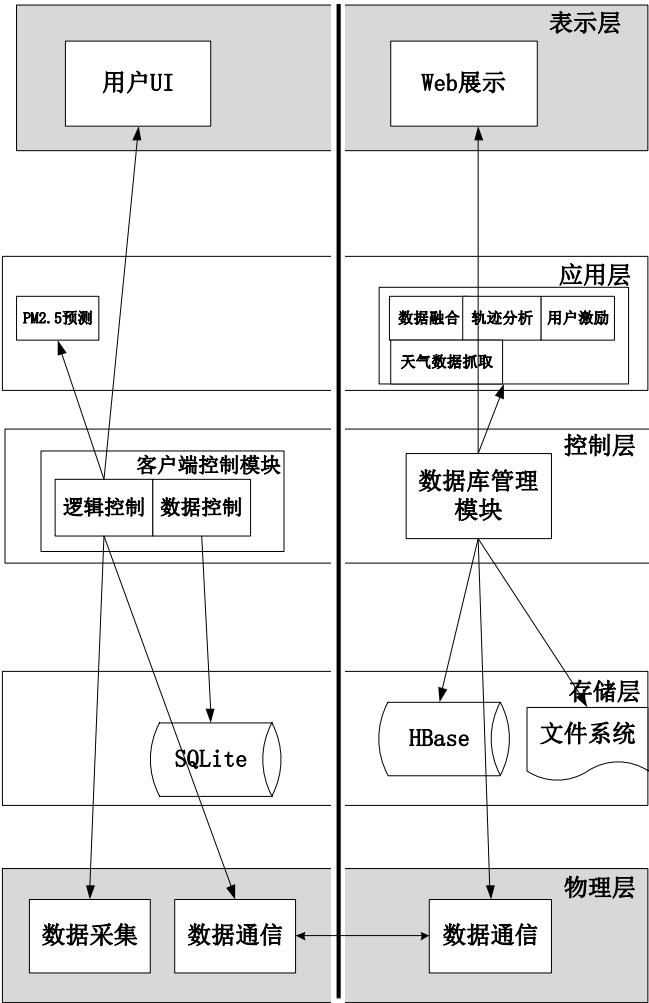


图 3-5 移动协作感知平台系统架构图

3.2 数据可视化系统的需求分析

通过前面对移动协作感知平台的需求分析和总体设计，我们知道，数据可视化系统的主要功能就是在 web 端对移动客户端收集到的感知数据进行可视化的图表展示。则我们可以得到数据可视化系统的拓扑结构如图 3-6 所示：

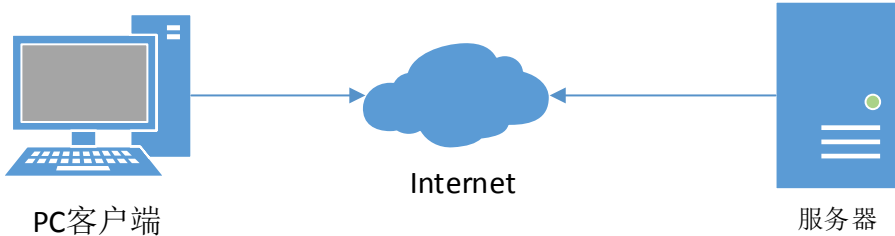


图 3-6 可视化系统拓扑图

用户通过 PC 客户端的浏览器登陆我们的 web 站点，查看可视化数据的展示效果，浏览器通过向移动协作感知平台的服务器请求数据，服务器返回所请求的

数据给浏览器，然后浏览器将感知数据以特定的形式展示给用户。

数据可视化系统的服务器需要响应浏览器的数据请求，同时要负责采集图片所对应的 POI 的计算，我们从客户端收集到图片会包含经纬度信息，但与在地图上显示的 POI 没有直接的对应，为了在点击 POI 时显示对应的图片，我们需要通过图片的经纬度信息确定图片所对应的 POI。

在 PC 客户端的 web 界面上主要完成用户注册登录、管理个人信息以及查看个人上传数据的统计展示功能，以及对所有收集到的感知数据的展示功能。其中个人数据包括用户名、邮箱、激励值等基本信息以及用户上传的图片、噪声等数据信息。而我们需要展示的感知数据主要包括图片、光照强度、噪声大小、图片的 PM_{2.5} 值、图片对应地点的 PM_{2.5} 的值以及展示数据的 POI 点的地理位置信息：

1) 对展示数据的 POI 点的地理位置的展示可以使用在线地图服务，在地图上可以根据 POI 的经纬度信息在地图上准确定位该点。但是当我们对移动协作感知数据收集者收集数据的 POI 在地图上进行展示时，需要考虑地图可视范围内 POI 的数量，大量的 POI 在地图上展示不仅会影响地图可视效果，还会影响地图性能，从而影响用户在使用可视化界面时的体验，所以我们需要对展示数据的 POI 点根据地图的不同缩放级别进行聚合，提高可视化界面的使用体验。

2) 展示数据的 POI 在地图缩放级别减小时会进行聚合，从而减少地图可视范围内的 POI 数。由于与 POI 对应的 PM_{2.5} 值、光照强度、噪声大小为具体的数值，并且随着时间的变化会得到不同的数值，则可以考虑通过随时间变化的曲线图来对这些数值进行可视化。其中聚合 POI 的各项数值可以通过被聚合的 POI 的相应的数据通过特定算法计算得出。

3) 对收集到的图片的展示可以采用幻灯片切换的方式，通过这种方式可以对图片采集地点的 PM_{2.5} 值有一个直观的认识。同时，图片 PM_{2.5} 值的展示应该与图片相对应，即在图片切换的同时图片 PM_{2.5} 值的展示效果也会跟随变化。由于图片 PM_{2.5} 值是一个具体的数值，所以可以考虑使用柱状图的高度来表示 PM_{2.5} 数值的大小。

4) 默认情况下，我们会展示地图中显示的所有 POI 所对应的图片。同时，地图中显示的 POI 是可以被点击的，在 POI 被点击后，图片展示区域应该只展示被点击的 POI 所对应的图片。

可以看出，在整个数据可视化的展示需求中，我们可以把地图作为整个展示效果改变的触发点。地图中展示的是有数据展示需求的 POI 点，在对地图进行缩放、拖动等操作从而改变地图可视范围时，地图上显示的 POI 会发生变化。因此，与 POI 对应的 PM_{2.5} 值、光照强度、噪声大小、图片以及图片的 PM_{2.5} 值

等数据都会跟随地图中显示的 POI 的变化而变化。

3.3 数据可视化系统的总体设计

根据我们对数据可视化系统的需求分析可以得出数据可视化系统的主要功能是对感知数据的 web 端可视化展示和服务端对 web 端的数据请求的处理和相关数据的计算。据此可以得到可视化系统的功能框架图如图 3-7 所示：

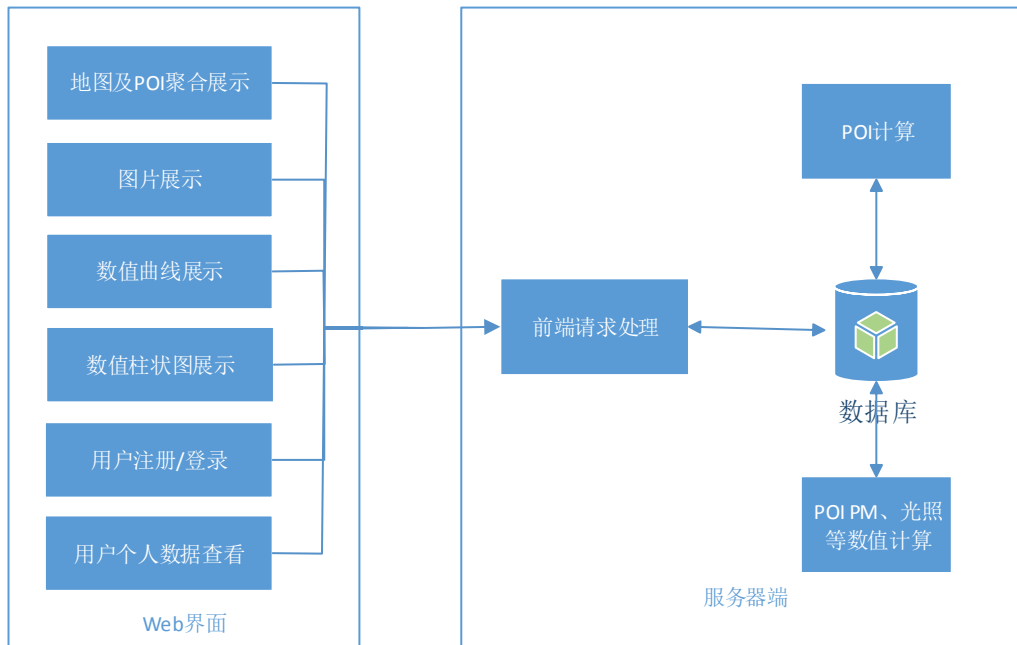


图 3-7 数据可视化系统功能框架图

数据可视化系统的功能分为 web 前端和服务端两个部分，各个部分的功能模块如图 3-7 所示。Web 前端要完成用户注册/登录、用户个人数据查看、地图及 POI 的聚合展示、图片的展示、采集数据的曲线和柱状图的展示；服务器端则负责对前端展示请求的处理、图片对应 POI 的计算以及 POI 的 PM 2.5、光照、噪声等数据的计算的功能。

数据可视化系统的主体流程是 web 页面的注册/登录请求、个人数据展示或者数据展示请求被发送到服务端，服务端通过数据库接口返回 web 端所需要的数据在 web 页面进行可视化展示，则可以设计出如图 3-8 所示的数据可视化系统的总体架构：表现层即用户可见注册/登录、个人数据查看以及感知数据的可视化界面，当用户在 web 界面上进行不同的交互操作时会产生不同的请求，此时业务逻辑层根据接收到前端页面请求的类型调用相应的数据库接口从数据库获取相应的数据并返回给前端页面，然后前端页面根据返回的数据进行页面的展示。

其中，表现层的 web 展示功能与业务逻辑层的交互可以通过 Ajax 技术发送 HTTP 请求来实现。在 web 端可以根据不同的展示需求向服务器发送不同的数据

请求，服务器根据接收到的前端页面的请求调用不同的数据库接口从数据库获取相应的数据并返回给前端页面进行数据展示。

业务逻辑层与数据访问层的交互主要是数据库提供的数据库接口。除了接收前端请求并向前端返回数据库返回的数据之外，业务逻辑层也接收数据库发送的客户端上传的图片的经纬值，并根据经纬度值计算出该经纬度所对应的 POI，再调用数据库接口将该 POI 值返回给数据库写入图片的 POI 属性列。业务逻辑层同时还要负责地图上展示的 POI 的 PM 2.5 值、光照、噪声等数据的计算。

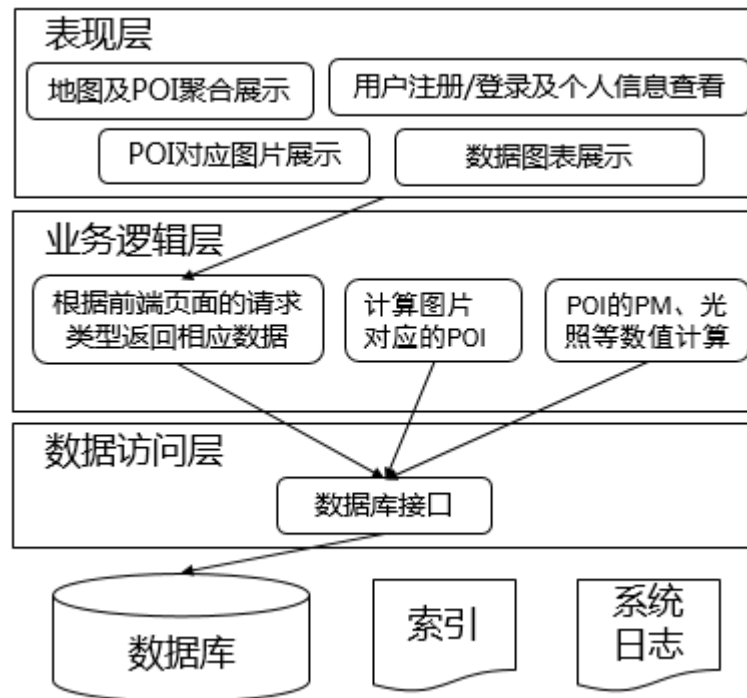


图 3-8 数据可视化系统的总体架构

3.4 数据可视化系统的交互的设计

移动协作感知平台通过移动客户端收集数据并将收集到的数据传送到服务器端，数据在服务器端经过融合处理之后会存储在分布式数据库中。数据可视化系统的用户注册/登录以及感知数据的 web 展示过程和服务器端的处理数据的过程都需要与移动协作感知平台进行交互。

在感知平台的用户的注册过程中，需要通过数据库验证用户注册所使用的用户名跟感知平台现有用户的用户名没有重复，并存储注册成功的用户的密码、邮箱等基本信息。则 web 展示与服务器和数据库的交互过程如图 3-9 所示。

而在感知平台的用户登录过程中需要通过数据库对用户输入的用户名和密码对进行验证，如果用户名和密码都正确则显示该用户的个人基本信息以及该用户所上传的感知数据的统计分析结果。则登录过程与数据的交互流程如图 3-10

所示。

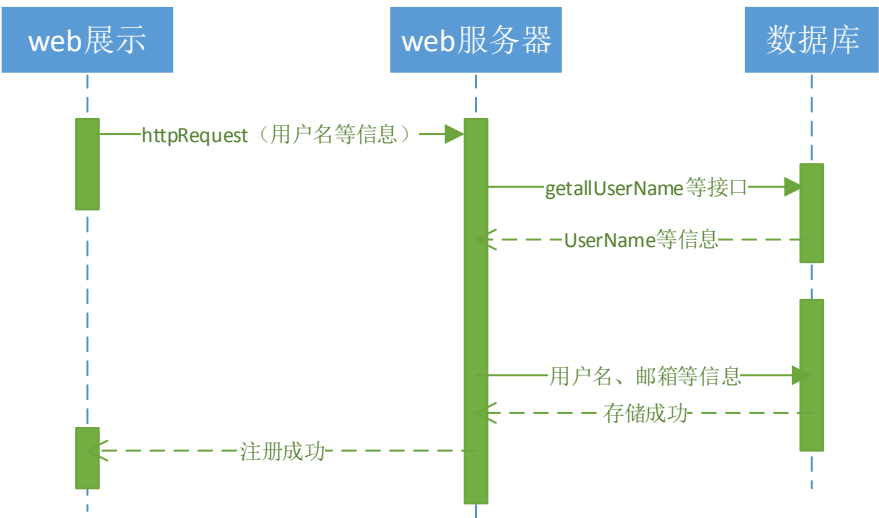


图 3-9 用户注册过程的交互

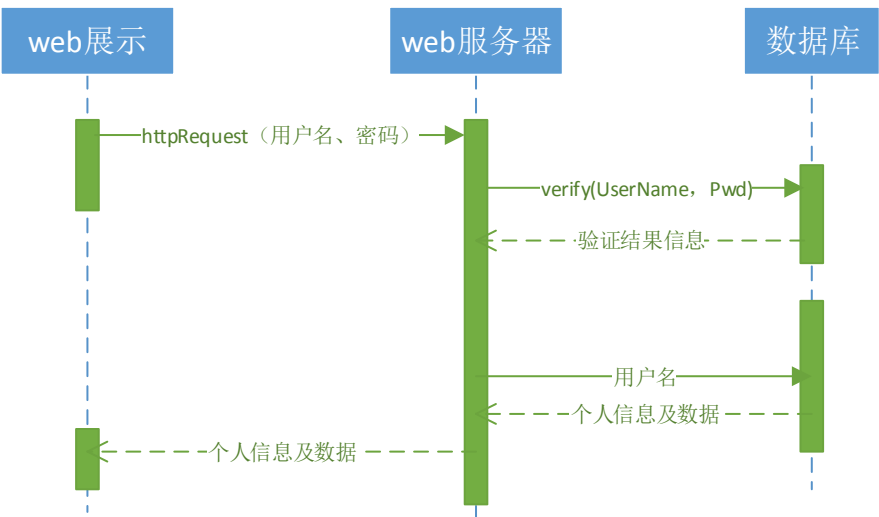


图 3-10 用户登录过程的交互

数据可视化系统的 web 展示模块在进行数据的展示时，会向服务器端发送数据请求，服务器端接收请求并调用数据库接口从数据库获取请求的数据并打包成 JSON 字符串格式，然后返回给 web 展示模块进行数据的解析和展示。该过程的交互如图 3-11 所示。

在 web 展示的过程中，我们可以获得在地图上展示的 POI 的 ID 及其经纬度信息用于 POI 在地图上的显示。然后再通过 POI ID 可以得到 POI 一段时间的 PM 2.5 值、光照强度、噪声大小等数据用于变化曲线的展示，也可以得到 POI 所对应的图片的存储位置用于图片展示部分，得到图片 PM 2.5 数据用于表示图片 PM 2.5 的柱状图的显示功能。

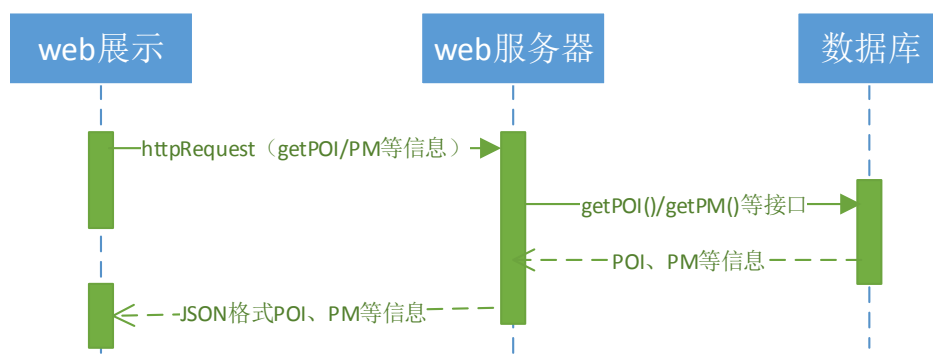


图 3-11 web 展示模块与数据库交互流程

数据可视化系统与分布式数据库的另一个交互场景涉及到图片所对应的 POI 的计算以及 POI 的 PM 2.5、光照、噪声等数据的计算。我们从客户端收集到的图片会包含经纬度信息，但与在地图上显示的 POI 没有直接的对应，为了在点击 POI 时显示对应的图片，我们需要通过图片的经纬度信息确定图片所对应的 POI。分布式数据库在收到新上传的图片之后将经纬度信息发送给数据可视化系统对应的模块，数据可视化系统利用该经纬度信息计算得到一个 POI ID 并将该 ID 返回给分布式数据库，然后分布式数据库将该信息写入数据库所对应的属性下。同时，我们采集到的 PM 2.5 值、光照、噪声等数据只是采集点的数值，而要得到 POI 的对应值，要得到该 POI 下所有点（或者下级 POI）的值，利用特定的算法计算得到 POI 点对应的 PM 2.5、光照、噪声值。该过程中数据可视化系统的服务器端的处理模块与数据库的交互如图 3-12 所示：

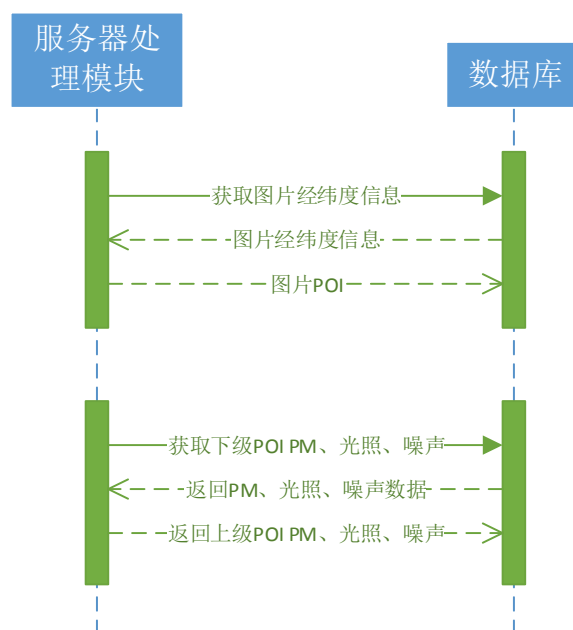


图 3-12 服务器端的处理模块与数据库的交互流程

第四章 数据可视化系统的详细设计与实现

结合数据可视化系统的需求分析及其总体结构的设计,我们可以将数据可视化系统细分为两个模块分别进行实现: web 前端展示模块以及服务器端的处理模块。本章将根据数据可视化系统需求分析的结果阐述 web 前端展示模块和服务器端的处理模块的详细设计及具体实现。同时,由于数据可视化系统的数据展示需求对数据库存储的数据格式有特定的要求,本章也将对数据可视化系统需要的数据库表的设计进行介绍。

4.1 数据库设计

数据可视化系统的主要工作是对客户端收集的数据进行可视化展示,同时负责移动协作感知平台的用户注册/登录以及用户个人信息的管理工作。为了便于数据库对数据的管理以及数据可视化系统在展示时对所需数据的获取,我们需要对数据库中数据的存储格式进行定义。其中,数据可视化系统需要的数据主要有用户个人相关信息,用户上传数据信息,POI 相关的信息,POI 的 PM 2.5、光照强度、噪声大小等信息,以及图片相关的信息,本节将对这些数据在数据库中的存储格式进行介绍。

4.1.1 用户信息表

移动协作感知平台的用户可以在网站进行注册,然后可以通过移动客户端上传自己采集的感知数据,了解自己日常生活区域的环境状况。用户信息表会存储用户注册时提供的基本信息,以及用户完成数据采集任务所获取的激励值。则可以得到表 4-1 所示的数据库表。

表 4-1 用户信息表

编号	用户名	密码	性别	邮箱	激励值
int	string	string	string	string	int

编号: 编号字段为整数值,是用户信息表主键,值用于标示用户数,随着用户增加自增。

用户名：字符串值，每个用户唯一，用于标示不同用户。

密码：密码字段为字符串值，用于存储经过加密的用户密码。

性别：性别字段为字符串值，用于存储用户的性别信息。

邮箱：邮箱字段为字符串值，用于存储用户注册时提供的邮箱信息，每个用户的邮箱信息也是唯一的。

激励值：激励值字段为整数值，用于存储用户通过完成数据采集任务而获得的奖励。

4.1.2 POI 表

我们在地图上进行 POI 的展示需要知道 POI 的经纬度信息，同时由于我们需要根据地图的缩放级别对 POI 进行聚合展示，所以需要有一个能保存 POI 经纬度信息以及缩放级别的数据库表，格式如表 4-2 所示：

表 4-2 POI 数据库表

POI ID	经度	纬度	缩放级别 1	缩放级别 2
string	double	double	int	int

POI ID：字符串值，用于标记聚合后的 POI 点，以及标记上传的图片所属的 POI，用于在点击 POI 时显示该 POI 所对应的图片。POI ID 按照行政区划以及算法计算的结果进行编码。对于聚合程度高的 POI，表示国家、省市、区县这些级别的 POI 的 ID 直接进行指定，字符串不同的位置段用来区分不同行政级别地点的 POI。对于表示区县级别以下的地理位置的 POI，表示 POI ID 的字符串的前半部分和区县的值相同，字符串的后半部分由算法计算得出。

经度、纬度：用于在地图上标记 POI 位置。对于聚合程度高的 POI，表示国家、省市区县这些级别的 POI 的经纬度值可以直接进行指定。对于表示区县级别以下的地理位置的 POI，POI 的经纬度值可以根据被聚合的 POI 或者数据采集点的经纬度值由算法计算得出。

缩放级别 1、2：表示 POI 应该被显示的地图的缩放级别，当地图的缩放级别在缩放级别 1 和缩放级别 2 之间时，应该在地图上显示该 POI 点。缩放级别的值对于不同聚合程度的 POI 来说是不同的，而同一聚合程度的 POI 是一样的，只需要在添加新的 POI 时写入数据库即可。

4.1.3 POI PM 表

每个 POI 点代表的区域在每一天会有一条对应的 PM 2.5、光照强度和噪声

大小的数据，我们在显示 POI 点 PM 2.5、光照强度或者噪声大小的变化曲线时会用到该项数据，故我们需要一个表 4-3 所示的 POI PM 2.5（POI 光照强度以及 POI 噪声大小）的数据库表：

表 4-3 POI PM 数据库表

POI ID	日期	PM 2.5
string	YYYYMMDD	float

POI ID: POI ID 即 POI 表中的 POI ID，每当 POI 表中增加一个新的 POI 时，POI PM 表就会存储该 POI 以后每天的 PM 2.5 值（光照强度以及噪声大小）。

日期: 用于表明 POI 的某一个 PM 2.5 值（光照强度以及噪声大小）所对应日期。该值由系统自动生成，每天自动加一。

PM 2.5: 每个 POI 某一天的 PM 2.5 值（光照强度以及噪声大小）可以根据特定的算法利用被聚合 POI 的当天的对应数据计算得出。

4.1.4 图片信息表

关于图片在展示时我们需要知道一些信息，图片存储在服务器上的位置用于在展示图片时获取图片，图片所属于的 POI 的 ID 用于在地图上点击 POI 时显示该 POI 下的所有图片，而图片的 PM 2.5 值以及 id 用于在柱状图中显示对应图片的 PM 2.5 信息，我们可以使用表 4-4 所示的数据库表来存储我们需要的上述信息：

表 4-4 图片信息数据库表

经度	纬度	图片位置	图片 id	图片 PM 2.5	POI ID
double	double	string	string	int	string

经度、纬度: 表示采集图片时采集者所处位置的经纬度值，该值由客户端从移动终端传感器获得并上传至服务端。用于计算图片所对应的 POI。

图片位置: 即图片在服务器上的存储位置的路径，由移动协作感知平台服务器端的网络接口模块在接收到图片并存储时自动生成存储路径并写入数据库表中。该路径用于前端展示图片时浏览器从服务器上获取图片。

图片 id: 图片 id 即图片名，由移动协作感知平台服务器端的网络接口模块在接收到图片并存储时根据上传图片的用户的用户名自动生成图片 id 并写入数据库表。

图片 PM 2.5: 客户端采集到图片之后由 PM 分析模块分析计算得出，并作

为图片的一个属性跟图片一起上传到服务器端。

POI ID: POI ID 由服务器端的 POI 计算部分根据图片的经纬度数据计算得出，应用于点击地图 POI 点时显示该 POI 点下的图片。

4.1.5 数据库接口设计

数据可视化系统中的数据库接口主要用于服务器端的处理模块与分布式数据库的交互。服务器端的处理模块在接收并响应前端界面的数据请求时要调用数据库接口向数据库请求数据并返回，同时，在进行图片对应的 POI 的计算和 POI 的 PM 2.5、光照、噪声等数值的计算时也需要通过数据库接口向数据库请求所需要的数据并将计算出的结果写入数据库。根据上述需求，我们设计的数据库接口如表 4-5 所示：

表 4-5 数据库接口

函数名	参数	返回值	功能说明
computePOI	double lon: 图片经度 double lng: 图片纬度	图片对应的 POI 的 ID	根据图片的经度和纬度计算图片所对应的 POI
getPOIsInfo	无	所有 POI 的信息	返回数据库中所有 POI 的信息用于地图显示
setDayFpm	string poi: POI ID Date timestamp: 日期 int fpm: PM 2.5 值	无	将 POI ID 为 poi 的 POI timestamp 日期的 PM 设置为 fpm
getDayFpm	string poi: POI ID Date timestamp: 日期	POI 的 PM 2.5 值	返回 POI ID 为 poi 的 POI timestamp 日期的 PM 值
getPicinPOI	string poi: POI ID Date timestamp: 日期	POI 某天的图片	返回 POI 某个日期的图片，若 timestamp 为 NULL，则返回该 POI 的所有图片
userRegister	String userName: 用户名 String email: 用户邮箱	0 或者 1	用户名和邮箱唯一，返回 1，表示注册成功；否则返回 0，注册失败

表 4-5 数据库接口（续上表）

函数名	参数	返回值	功能说明
userLogin	String userName: 用户名 String passwd: 密码	0 或者 1	用户名和密码都匹配，返回 1，表示登录成功；否则返回 0，登录失败

4.2 服务器端的处理模块

数据可视化系统的服务器端的处理模块位于移动协作感知平台的服务器端。我们利用 Java 中的 servlet 技术将数据可视化系统的服务器端的处理模块作为一个单独的服务部署在 Tomcat 服务器上，这样可以做到与移动协作感知平台服务器端的其他模块如数据融合、轨迹预测或者激励模块相互独立，而只通过数据库接口与分布式数据库进行交互。这样在除了分布式数据库模块之外的其他某一个模块出现问题时，不会影响剩余模块的正常工作。

数据可视化系统的服务器端的处理模块主要处理两部分工作，包括计算新上传的图片所对应的 POI 以及接收并响应前端页面的数据请求。而这两部分工作是独立并行运行的，不会产生任何的数据或者其他信息的交互。

4.2.1 POI 计算部分

服务器端的处理模块的 POI 计算部分在处理模块在服务器上部署完成之后会一直运行，并定期轮询分布式数据库模块，查询是否有新的图片上传，如果有新上传的图片，则通过分布式数据库的接口得到新上传的图片的经纬度值，并根据经纬度值计算新上传的图片所对应的 POI 的 ID，然后将计算得到的 POI ID 返回给分布式数据库，存储在图片对应的数据库表中。

在地图的缩放级别比较小时，地图上所显示的 POI 点代表不同级别的行政区划，如国家、省市或者区县等地点，这些 POI 点的位置是一定的且聚合的规则可以按照行政区划的划分来聚合，则他们的 POI ID 不用计算，由系统直接指定即可。在地图缩放级别较大时，地图上所显示的 POI 点代表区县级别以下的地点，这些 POI 点事先是不确定的，而是在数据采集的过程中逐渐确定下来的。

对于区县级别以下地点 POI 的聚合，我们将其分为两个聚合级别：首先将采集的数据所在的点聚合到聚合度为 1（定义为 JH1）的 POI 点，然后将聚合度为 1 的 POI 点聚合到聚合度为 2（定义为 JH2）的 POI 点，聚合度为 2 的 POI 会直接聚合到表示区县级别地点的 POI。而对于图片所属于的区县的信息，我们可以通过 google 地图的反向地址编码服务得到。我们可以得到上传图片的经纬度值，

然后利用 google 地图的反向地址编码服务接口，传入图片经纬度值，得到图片的具体地址信息，包括国家、省市以及区县信息，从而可以得到图片所属 POIID 的前半部分编码。由于我们可以得到图片所属 POI ID 的前半部分编码，所以我们的计算主要是要得到图片所属 POI ID 的后半部分编码。

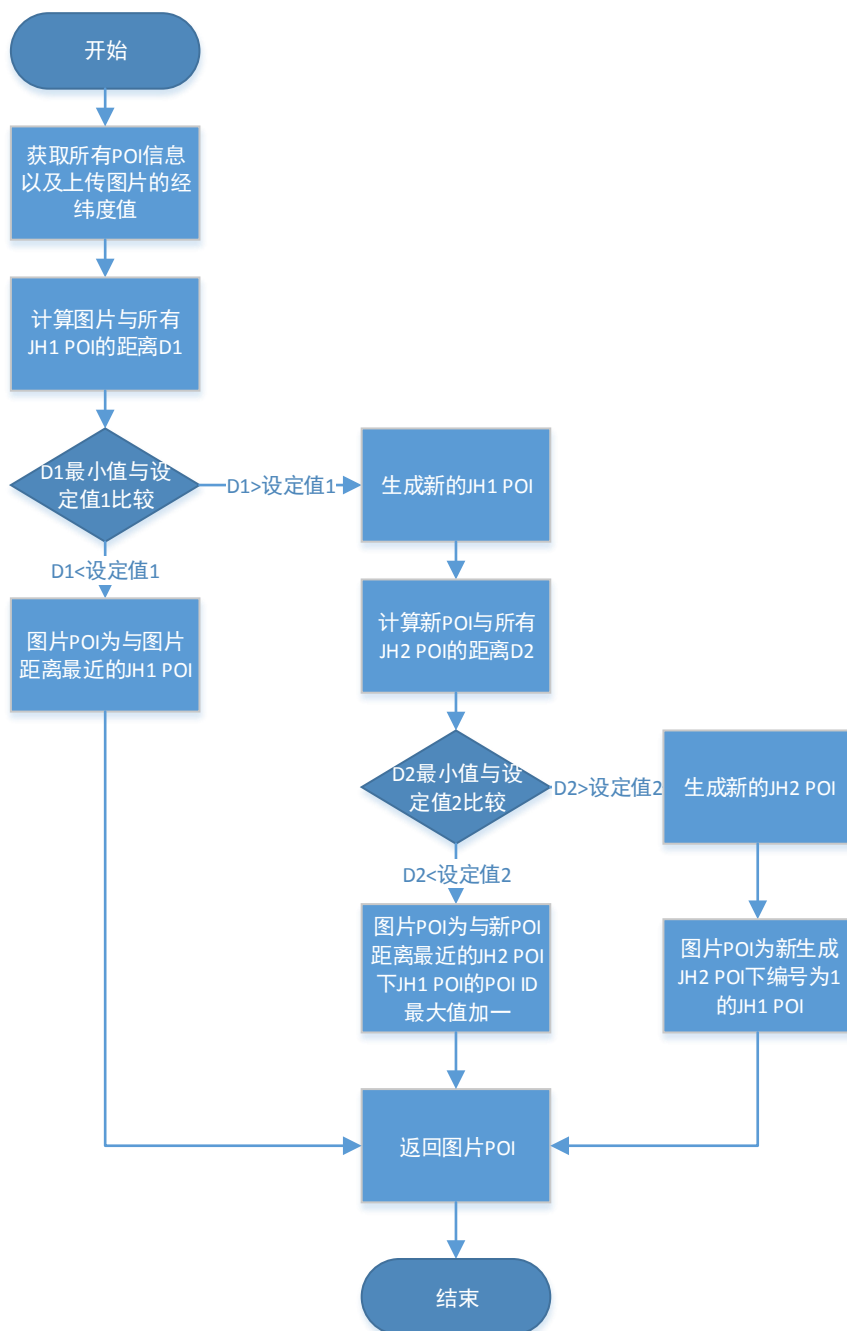


图 4-1 图片对应 POIID 的计算流程图

对于一张新上传的图片，我们可以通过 google 地图的反向地址编码服务得到其所属于的区县信息，然后我们可以获取该区县下所有的 JH1 和 JH2 的 POI。我们通过计算图片的与所有 JH1 的 POI 之间的距离，然后取其中的最小值，判断该最小值与我们设定的边界值的大小，从而确定该图片是否属于已存在的某一

个 JH1 的 POI, 如果该最小值小于边界值, 则该图片属于与图片距离最近的 POI, 得到图片所属 POI ID; 如果该最小值大于边界值, 则生成一个新的 JH1 的 POI, 其经纬度信息即为新上传的图片的经纬度值, 再计算新生成的 JH1 的 POI 与所有的 JH2 的 POI 之间的距离, 然后取其中的最小值, 判断该最小值与我们设定的边界值的大小, 从而确定新生成的 JH1 的 POI 是否属于已存在的某一个 JH2 的 POI, 如果该最小值小于边界值, 则新生成的 JH1 的 POI 属于与该 POI 距离最近的 JH2 的 POI, 则该 JH1 POI 的 POI ID 为 JH2 的 POI 下的 JH1 的 POI ID 的最大值加一, 而图片所属于的 POI 即为新生成的 JH1 的 POI; 如果该最小值大于边界值, 则生成一个新的 JH2 的 POI, 其经纬度信息也为新上传的图片的经纬度值, 并且该新生成的 JH2 的 POI 的 ID 为该区下所有 JH2 的 POI 的 ID 的最大值加一, 新生成的 JH1 的 POI 的 ID 为新生成的 JH2 的 POI 下编号为 1 的 ID 号, 图片所属于的 POI 为新生成的 JH1 的 POI, 完成新上传的图片对应的 POI ID 的计算。上述计算新上传的图片对应的 POI ID 的流程如图 4-1 所示。

4.2.2 POI PM 计算

我们通过收集到的图片分析出来的 PM 2.5 数据 (以及采集到的光照、噪声等数据) 只能表示图片拍摄地点一定范围之内的数据, 而在对地图进行缩小操作时, 地图上的 POI 会聚合, 那么代表更大范围的 POI 的 PM 2.5 等数据无法直接获得, 则我们可以利用特定的算法 (例如, PM 2.5 数据可以通过下一级的 POI 的 PM 数据求平均值获得) 得到聚合级别较高的 POI 的 PM 2.5 等数据。

我们要计算一个 POI 某天的 PM 2.5 等数据, 可以先通过数据库接口 `getDayFpm()` 获得聚合到该 POI 的所有下一级 POI 当天的 PM 2.5 等数据, 然后对所获得的所有数值求平均值 (或者其他适合特定类型数据的算法), 并将该值作为所求 POI 当天的 PM 2.5 值写入数据库作数据展示之用。

我们从数据库获得的数据的格式为如图 4-2 所示的 JSON 对象数组:

```
[
  - {
    "poi_pm_id": "10010117",
    "poi_pm_date": "2014-03-26",
    "poi_pm_value": 132
  },
  + { ... },
  + { ... },
  + { ... }
]
```

图 4-2 POI 的 PM 数据的 JSON 格式

4.2.3 响应前端请求部分

在服务器上部署的数据可视化系统的处理模块接收前端页面的 HTTP 请求，并根据请求类型及其他参数调用不同的分布式数据库接口向数据库请求数据，然后将数据打包成特定的数据格式返回给前端页面。上述接受请求并响应的整个流程如图 4-3 所示：

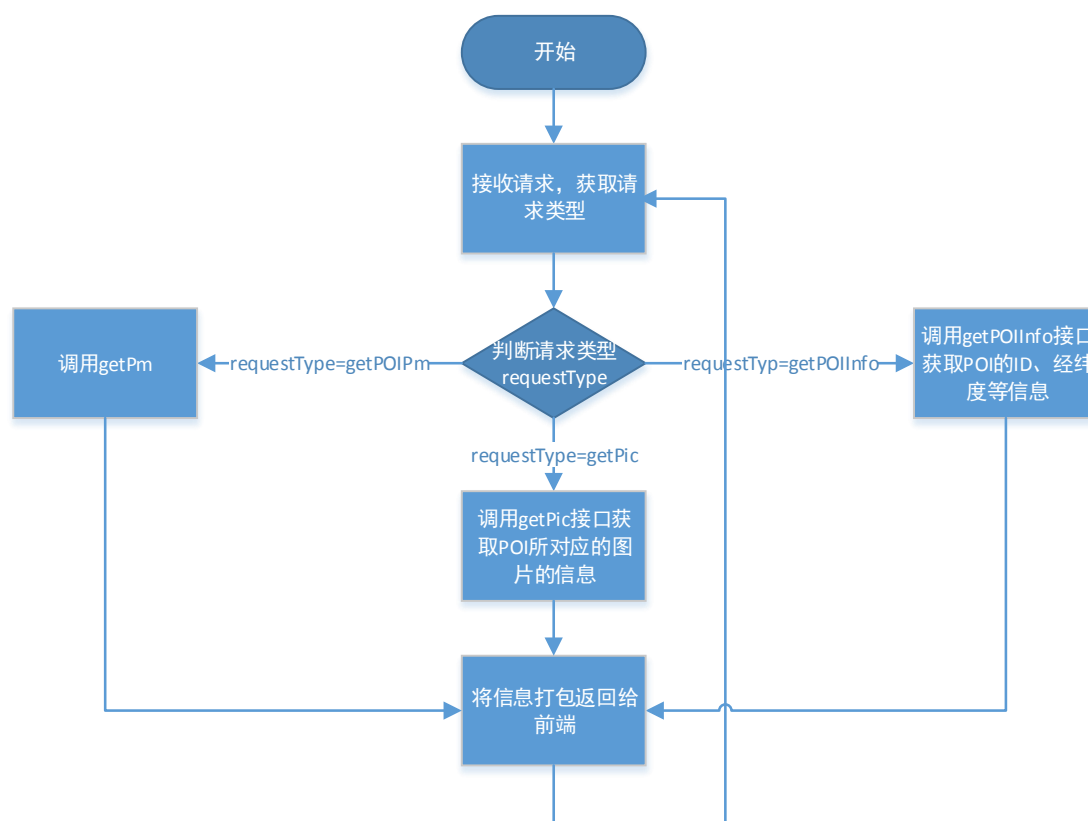


图 4-3 服务端接收请求并响应流程

对服务端接收前端请求并进行响应的流程的实现主要依靠 servlet 中的 doGet 函数，实现的核心代码如下：

```

protected void doGet (HttpServletRequest request, HttpServletResponse response)
{
    String queryType = request.getParameter("queryType"); //获取请求类型

    If (queryType.equals("getallpoiinfo")) { //请求 POI 信息
        Set<JSONObject> allPOIInfo = ww.getPOIsInfo();//获取 POI 信息

        response.setContentType("application/json");//返回 POI 信息
        PrintWriter out = response.getWriter();
    }
}

```

```

        out.println(allPOIInfo);
        out.flush();
    }
    else if (queryType.equals("getpoipm")) { //请求 POI PM 信息
        //获取并返回 POI PM 信息
    }
    else if (queryType.equals("getpoiimg")) { //请求单个 POI 的图片信息
        //获取 POI 图片信息并返回
    }
    else if (queryType.equals("getviewimg")) { //请求多个 POI 的图片信息
        //获取 POI 图片信息并返回
    }
}

```

4.2.4 用户注册登录处理

服务器端的处理模块还要负责接收前端页面的注册和登录请求，并同数据库进行交互从而确定注册和登录是否成功，并向前端页面返回响应结果。

对于没有移动协作感知平台账号的用户，需要进行账号的注册，而在账号注册过程中我们需要确保账号的唯一性，同时采集用户的性别、邮箱等基本信息。账号注册的流程如图 4-4 所示。

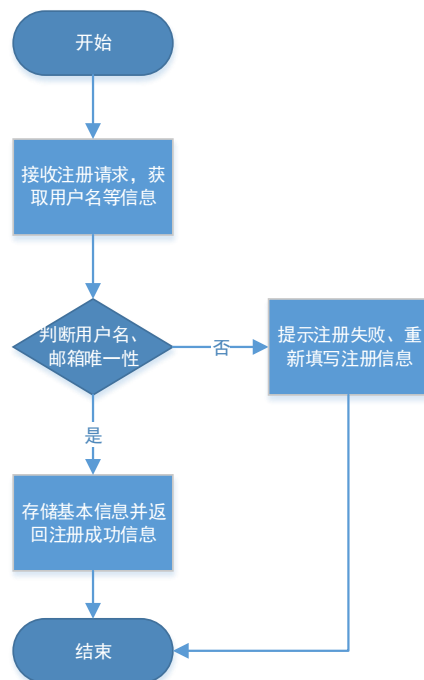


图 4-4 服务器处理用户注册流程

对于已有账号的用户，在登录移动协作感知平台网站时要对账号密码进行验证，在验证成功之后会获取该用户的基本信息以及上传的感知数据的信息，并将该信息返回给网页进行展示。服务器端处理用户登录的流程如图 4-5 所示。

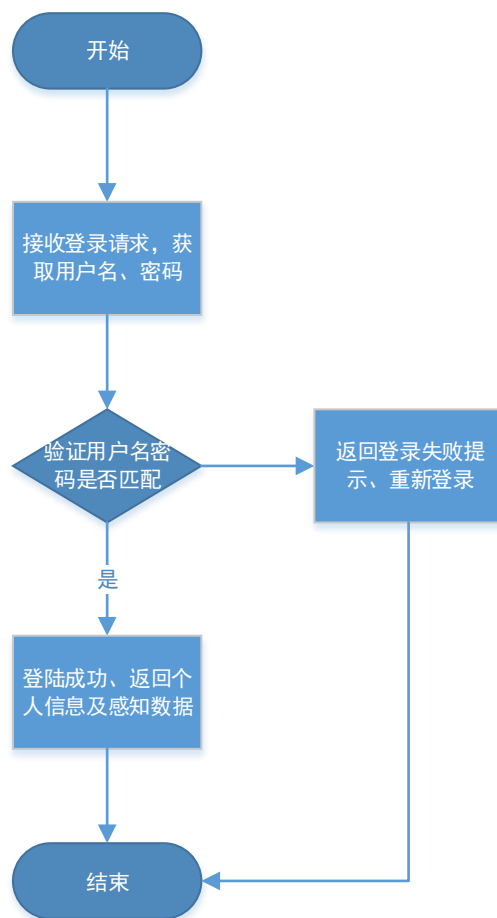


图 4-5 服务器处理用户登录流程

4.3 web 前端展示模块

结合数据可视化系统的需求分析和总体架构，我们知道 web 前端展示模块主要负责表现层的工作，包括地图显示及地图中 POI 的聚合展示，地图可视范围中 POI 一段时间的 PM 2.5 值、光照强度、噪声等数据的展示，地图可视范围中 POI 所对应的图片的展示，以及每张图片所对应的 PM 2.5 值的展示。由此我们可以将 web 前端的展示效果划分为四个部分，分别展示地图，POI 对应的相关数据的变化曲线，图片以及表示图片对应的相关数据的数值的柱状图。网页的基本布局如图 4-6 所示。

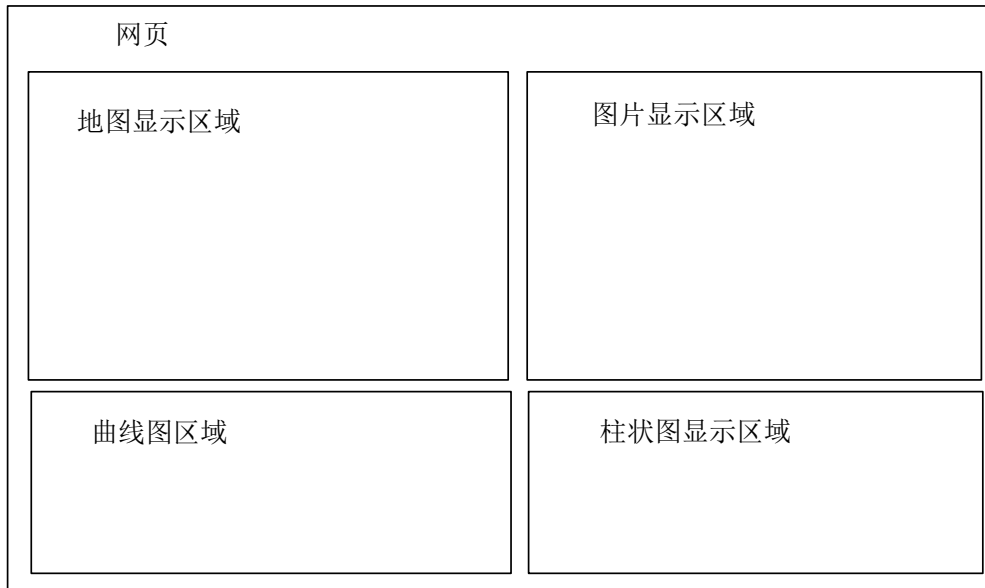


图 4-6 web 前端页面布局

我们可以通过如下的 HTML 网页代码以及用于控制各区域大小和位置的 CSS 代码来实现图 4-6 所示的页面布局：

```
<div id="container" style="width: 100%; height: 100%">
  <div id="topHalfContainer" style="width: 100%; height: 65%;">
    <div id="map"> </div>
    <div id="blueimp-gallery-carousel"> </div>
  </div>
  <div id="bottomHalfContainer" style="width: 100%; height: 35%;">
    <div id="lineContainer"> </div>
    <div id="columnContainer"> </div>
  </div>
</div>
```

其中 id 为 map 的 div 用于地图的显示，id 为 blueimp-gallery-carousel 的 div 用于图片的幻灯片展示，id 为 lineContainer 的 div 用于曲线图的显示，id 为 columnContainer 的 div 用于柱状图的显示。

Web 前端界面的基本布局如图 4-6 所示，根据需求分析的结果可得 web 前端展示模块中地图显示区域将作为整个前端界面展示效果改变的触发点，地图区域的变化带动其他模块变化。前端界面的基本控制逻辑为：地图中展示的是有数据展示需求的 POI 点，在对地图进行缩放、拖动等操作从而改变地图可视范围时，地图上显示的 POI 会发生变化，地图上显示的 POI 的改变导致页面产生新的数据请求。一方面根据地图上显示的 POI 利用 Ajax 技术向服务器请求 POI 一段时间的 PM 2.5 值改变曲线图区域的显示效果，另一方面根据地图上显示的 POI

利用 Ajax 技术向服务器请求 POI 对应的图片的信息，主要是用于改变图片展示效果的图片在服务器上的路径信息和用于改变柱状图展示效果的图片 PM 2.5 值以及图片名等信息。地图上的另一个操作是点击 POI，在 POI 被点击时，我们可以获得被点击 POI 的 POIID，然后根据 POIID 利用 Ajax 技术向服务器请求 POI 对应的图片的信息，主要是用于改变图片展示效果的图片在服务器上的路径信息和用于改变柱状图展示效果的图片 PM 2.5 值以及图片名等信息。综合上述的分析，我们可以得到 web 前端展示模块的主要流程如图 4-7 所示。

由 web 前端展示模块的流程图可以知道，地图缩放级别和可视范围改变这个事件以及对地图上的 POI 的点击事件是前端界面展示效果改变的触发事件。我们对地图缩放级别和可视范围改变事件以及对地图上的 POI 的点击事件的监控通过谷歌地图的事件监听函数来实现。

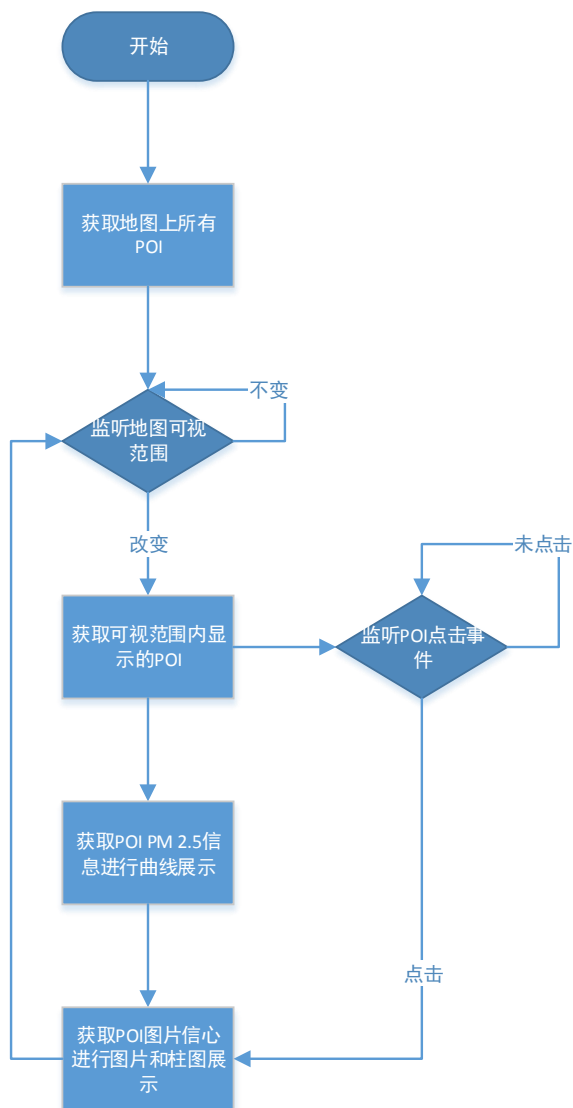


图 4-7 web 前端展示模块流程图

我们对地图可视范围改变事件的监听实现如下：

```

google.maps.event.addListener(map, 'bounds_changed', debounce(function (event)
{ //监听地图边界的变化
    //获取可视范围中的 POI，在地图上进行展示
    //获取 POI 的 PM 2.5 值用于曲线图区域的变化曲线的展示
    //获取 POI 所对应的图片信息用于图片以及图片 PM 2.5 值柱状图的展示
}, 300));

```

其中的 `debounce()` 函数用于可视范围改变这个高频事件的防抖，因为在缩放移动或者移动地图的过程中，可视范围变化的频率非常高，每一次改变都会导致 POI 相关信息的请求，影响 web 页面的性能和展示效果，使用 `debounce()` 函数可以延迟地图可视范围的变化引起的数据请求，有效提高 web 界面的展示效果。

我们对 POI 点击事件的监听实现如下：

```

google.maps.event.addListener (clickMarker, 'click', function() { //监听被 POI 点
击，其中 clickMarker 表示被监听的 POI，click 表示点击事件
    //获取被点击 POI 所对应的图片的相关信息，用于图片区域以及图片 PM
2.5 值柱状图的展示
});

```

4.3.1 地图显示区域

由需求分析结果可知，地图显示区域所展示的主要内容就是在地图上有数据展示需求的 POI 进行展示，以及在我们对地图进行缩放操作时对 POI 进行聚合和分散的展示。该区域展示的主要流程如图 4-8 所示：

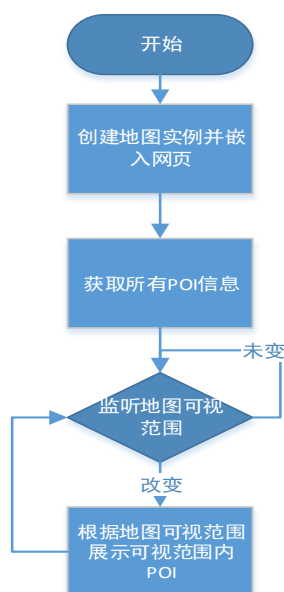


图 4-8 地图展示区域流程图

由上述流程图可知, 在我们进行地图以及 POI 的展示时, 首先需要创建一个实例并在网页上显示出来, 接下来利用 Ajax 技术向服务器发送 HTTP 请求, 获得需要展示的所有 POI 点的数据并根据地图的缩放级别以及 POI 的最大最小缩放级别数据, 确定需要显示的 POI 并将其在地图上显示出来。

对于地图显示区域的展示, 我们采用了第二章介绍的谷歌地图进行实现, 利用谷歌地图简洁而丰富的 API, 我们可以很容易实现想要的功能。

创建地图并将地图嵌入 web 页面的主要代码如下:

```
var myLatLng = new google.maps.LatLng (39.5209922936,116.6802978516);
var options = {
    zoom: 4, //地图初始缩放级别
    maxZoom: 18, //地图最大缩放级别
    center: myLatLng, //地图初始中心点
    mapTypeControl: false,
    mapTypeId: google.maps.MapTypeId.ROADMAP //地图类型
};
var map = new google.maps.Map (document.getElementById ('map'), options);//将
地图嵌入 web 页面 id 为 map 的 div 区域
```

而对 POI 的聚合显示, 我们需要先获取所有的 POI, 并按 POI 的最大最小缩放级别用数组分别存储, 再根据地图缩放级别确定要显示的 POI 从而达到 POI 的聚合效果, 主要实现过程如下:

```
google.maps.event.addListener (mgr, 'loaded', function () { //监控地图缩放级别
    mgr.addMarkers (zoom_1_8_Markers, 1, 8); //地图缩放级别为 1-8 时, 显示
    最小缩放级别为 1, 最大缩放级别为 8 的 POI, 存储在 zoom_1_8_Markers 中。
    mgr.addMarkers (zoom_4_8_Markers, 4, 8);
    mgr.addMarkers (zoom_5_8_Markers, 5, 8);
    mgr.addMarkers (zoom_9_11_Markers, 9, 11);
    mgr.addMarkers (zoom_12_14_Markers, 12, 14),
    mgr.addMarkers (zoom_15_18_Markers, 15, 18);
    mgr.refresh ();
});
```

4.3.2 图片显示区域

由需求分析结果可知, 图片显示区域展示的主要内容是对地图上显示的 POI 所对应的图片进行幻灯片展示, 并在图片上标记图片所对应的 POI 在地图上的

图标以表明图片与 POI 的对应关系。该区域进行展示的流程如图 4-9 所示。

由 4-9 所示的流程图可知，在我们进行图片展示时，首先要获取地图上显示的 POI，然后根据获取的 POI 信息利用 Ajax 技术向服务器发送 HTTP 请求，获取这些 POI 所对应的图片的信息，然后将图片在服务器上存储路径写入网页，浏览器即可获取图片进行展示。

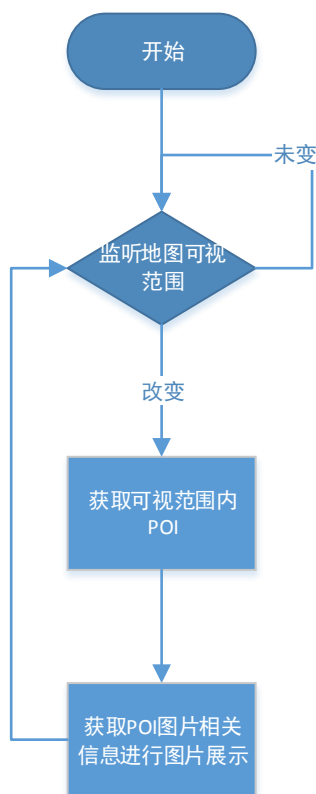


图 4-9 图片展示区域流程图

浏览器从服务器获取的图片相关信息会在服务器端被组织成 JSON 格式的字符串，到达浏览器后由负责图片展示的函数 `showViewPOIImg()` 对 JSON 字符串进行解析并得到所需要的信息。服务器返回的 JSON 字符串格式如图 4-10 所示：

```
[
  - {
    "pic": "ST_1414641774149.jpg",
    "fpm": 1024,
    "poi": "10010117",
    "date": 1414598400000
  },
  + { ... },
  + { ... },
]
```

图 4-10 服务器返回 JSON 格式图片信息

该 JSON 对象数组会包含 POI 的多张图片，每张图片的信息如图所示。其中

pic 表示图片的路径和名称，用于图片的展示；fpm 表示图片的 PM 2.5 值，后续将被用于图片 PM 2.5 值的柱状图展示；poi 和 date 分别表示图片对应的 POI 以及图片的拍摄日期。

4.3.3 曲线图区域

由需求分析结果可知，曲线图区域展示的主要内容是对地图上显示的 POI 在一段时间内的 PM 2.5 值的变化趋势进行展示，并确保每条表示 POI PM 2.5 变化的曲线的颜色与 POI 在地图中的图标的颜色保持一致。曲线图区域进行展示的流程如图 4-11 所示：

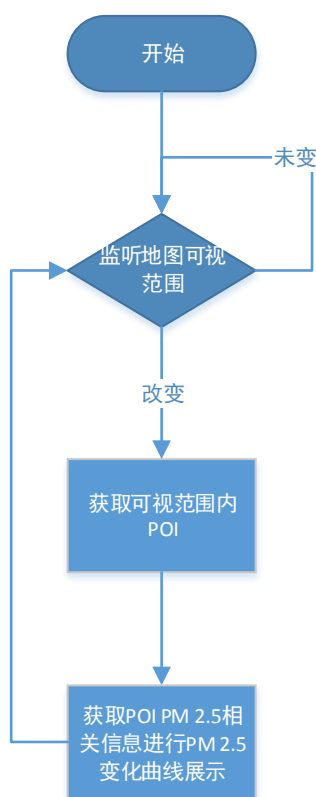


图 4-11 曲线图展示区域流程图

由上述流程图可知，我们在进行 POI PM 2.5 值的变化的展示时，首先要获取地图上显示的 POI，然后根据获取的 POI 信息利用 Ajax 技术向服务器发送 HTTP 请求，获取这些 POI 一段时间的 PM 2.5 值，然后利用 Highcharts 进行曲线展示。

对于曲线图区域的曲线图的实现，我们采用了第二章介绍的 Highcharts 开源图表库进行实现。在 web 展示页面按要求引入其所需要的文件之后，只需要将获取的 POI 的 PM 2.5 值的信息传入接口函数即可实现 POI PM 2.5 值的变化趋势的展示。

浏览器从服务器端获取的 POI PM 2.5 信息会在服务器端被组织成 JSON 格式的字符串，到达浏览器后由负责曲线图展示的函数 `showViewPOIPM()` 对 JSON 字符串进行解析并得到所需要的信息。服务器返回的 JSON 字符串格式如图 4-12 所示：

```
[
  - {
    "poi_pm_id": "10010111",
    "poi_pm_date": "2014-03-26",
    "poi_pm_value": 39
  },
  - {
    "poi_pm_id": "10010111",
    "poi_pm_date": "2014-03-27",
    "poi_pm_value": 56
  },
  + { ... },
  + { ... }
]
```

图 4-12 服务器返回 JSON 格式图片信息

该 JSON 对象数组包含一个 POI 多个日期的 PM 2.5 值，其中的 `poi_pm_value` 即为 PM 2.5 的值；`poi_pm_id` 和 `poi_pm_date` 分别表示 POI 的 ID 以及该 PM 值对应的日期。

4.3.4 柱状图区域

由需求分析结果可知，柱状图展示区域展示的主要内容是图片展示区域所展示的每张图片的 PM 2.5 值的柱状图表示，并确保柱状图的颜色与图片所对应的 POI 在地图中的图标的颜色保持一致。柱状图展示区域进行展示的流程如图 4-13 所示。

由图 4-13 所示的流程图可知，我们在进行图片 PM 2.5 值的柱状图展示时，首先要获取地图上显示的 POI，然后根据获取的 POI 信息利用 Ajax 技术向服务器发送 HTTP 请求，获取这些 POI 所对应的图片的信息，然后利用 Highcharts 进行柱状图展示。

对于柱状图展示区域柱状图的实现，我们采用了第二章介绍的 Highcharts 开源图表库进行实现。在 web 展示页面按要求引入其所需要的文件之后，只需要将获取的图片的 PM 2.5 值的信息传入接口函数即可实现图片 PM 2.5 值的柱状图显示。

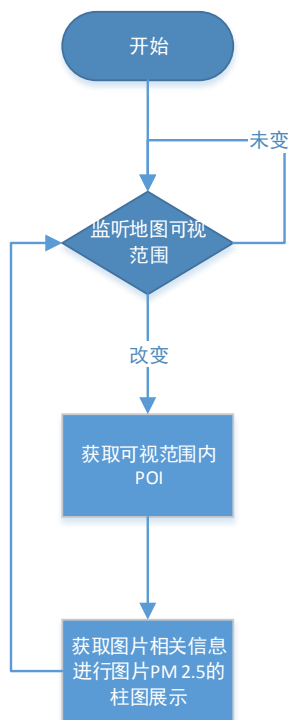


图 4-13 柱状图展示区域流程图

浏览器从服务器获取的图片相关信息会在服务器端被组织成 JSON 格式的字符串，到达浏览器后由负责图片展示的函数 `showViewPOIImg()` 对 JSON 字符串进行解析并得到所需要的信息。服务器返回的 JSON 字符串格式如图 4-14 所示：

```
[
  - {
    "pic": "ST_1414641774149.jpg",
    "fpm": 1024,
    "poi": "10010117",
    "date": 1414598400000
  },
  + { ... },
  + { ... },
]
```

图 4-14 服务器返回 JSON 格式图片信息

该 JSON 对象数组会包含 POI 的多张图片，每张图片的信息如图所示。其中 `pic` 表示图片的路径和名称，用于图片的展示；`fpm` 表示图片的 PM 2.5 值，被用于图片 PM 2.5 值的柱状图展示；`poi` 和 `date` 分别表示图片对应的 POI 以及图片的拍摄日期。

第五章 数据可视化系统的测试

前面两章我们对移动协作感知平台的数据可视化系统进行了详细的分析和设计，并对其中的关键部分的进行了实现。在此基础上，本章我们将对我们实现的基于移动协作感知平台的数据可视化系统进行功能测试，以验证我们的设计和实现可以满足数据可视化的要求。

5.1 测试环境

数据可视化系统基于移动协作感知平台来实现并成为移动协作感知平台的一部分，我们的测试也将基于已搭建的移动协作感知平台。由于数据可视化系统的展示界面是 web 端网页，则我们只需要一台连接到服务器的 PC 客户端，通过浏览器便可查看可视化界面，对可视化效果进行测试。测试拓扑如图 5-1 所示：

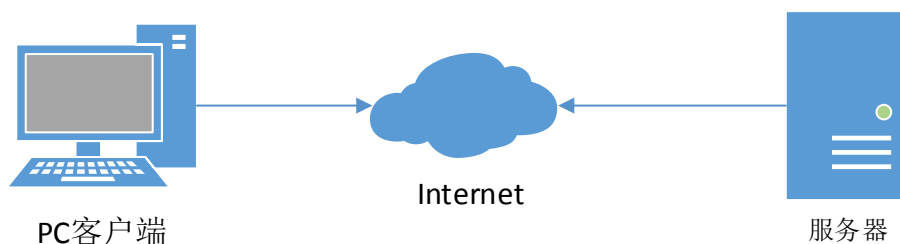


图 5-1 测试环境拓扑图

在我们的测试环境中，PC 端操作系统为 Windows 7 旗舰版 SP1，web 展示采用 Chrome 35.0 浏览器，而在服务器端，采用 Tomcat 7.0 服务器软件进行服务请求处理，数据存储采用基于 Hadoop 的分布式数据库 HBase，版本号为 0.94.19。

5.2 测试目的

对数据可视化系统的测试的目的是检验我们实现的基于移动协作感知平台的数据可视化系统的相关功能能否运行正常，是否能够达到我们所设计的要求和展示效果。

由需求分析的结果可知，数据可视化系统包含服务器端和 web 前端两个部分的功能。服务器端响应浏览器的数据请求，同时要负责采集图片所对应的 POI 的计算；而 web 前端主要负责对服务器端经过处理的数据进行可视化展示，主要

包括通过在线地图对展示数据的 POI 点的展示；在我们移动或者缩放地图时对 POI 的聚合展示；对地图可视范围中的 POI 一段时间的 PM 2.5 值（光照强度以及噪声大小）的展示；对地图可视范围中的 POI 一段时间所采集到的图片的幻灯片展示以及当地图中 POI 被点击时对单个 POI 一段时间所采集到的图片的幻灯片展示；跟随图片的幻灯片切换，对每张图片的 PM 2.5 值的展示。

5.3 测试方案及过程

由于条件的限制，在本次数据可视化系统的测试中，测试数据绝大部分来源于移动协作感知项目组的同学在北京邮电大学校园内通过移动客户端收集所得，其余部分数据由移动协作感知平台根据系统设定默认填充或者手动填写以确保测试效果的完整体现。

5.3.1 服务器端的处理模块测试

服务器端的处理模块的主要工作是对新上传的图片的 POI 进行计算以及对前端页面的请求进行响应并返回所需的数据，所以对服务器端的处理模块的测试也主要针对这两个方面。而对前端页面的请求的处理的功能的测试由 web 前端展示模块的测试可以得出结果，因为前端展示的正常显示说明服务器对前端请求的处理是正常的。所以本节将只对服务器端计算新上传的图片对应的 POI 的功能进行测试。

1、新上传的图片属于已存在的某个 POI

1) 测试内容：在某个 POI 的范围内采集一张图片并上传，通过图片的经纬度值计算该图片所对应的 POI ID。为了简化步骤我们可以取某一 POI 范围内的一对经纬度值，传入 POI ID 的计算函数，查看返回的 POI ID 是否与我们选择的 POI ID 一致。

2) 测试步骤：在地图上获取北邮学五的一对经纬度值，并将经纬度值传入 POI ID 的计算函数，运行程序，查看返回的结果。

3) 预期结果：在我们的数据库中学五所对应的 POI 的 ID 为 10010111，则 POI ID 的计算函数返回的结果为 10010111。

4) 测试结果：实际测试结果如图 5-2 所示：

```
2014-11-14 14:18:03,003 [Timer-0] 116.35607147216797;39.96360778808594
2014-11-14 14:18:03,003 [Timer-0] minPOI=10010111; min=26.04515142148072
2014-11-14 14:18:03,003 [Timer-0] 10010111
2014-11-14 14:18:03,003 [Timer-0] compute run
```

图 5-2 图片属于已存在的 POI 计算结果

5) 结果分析: 分析图 5-2 可知, 由于我们输入的经纬度值在 POI ID 为 10010111 的 POI 的范围之内, 则 POI 计算函数根据输入的经纬度值返回的 POI ID 为 10010111, 符合我们对测试结果的预期。

2、新上传的图片不属于已存在的任意 POI

1) 测试内容: 在所有 POI 的范围之外采集一张图片并上传, 通过图片的经纬度值计算该图片所对应的 POI ID。为了简化步骤我们可以取所有 POI 范围之外的一对经纬度值, 传入 POI ID 的计算函数, 查看返回的 POI ID 是否与我们选择的 POI ID 一致。

2) 测试步骤: 在地图上获取北邮足球场的一对经纬度值, 并将经纬度值传入 POI ID 的计算函数, 运行程序, 查看返回的结果。

3) 预期结果: 在我们的数据库中北邮校内所有 POI 的最大 POI ID 为 10010118, 则 POI ID 的计算函数返回的结果为 10010119。

4) 测试结果: 实际测试结果如图 5-3 所示:

```
2014-11-16 11:48:40,040 [Timer-0] 116.35577392578125;39.96292495727539
2014-11-16 11:48:40,040 [Timer-0] minPOI=10010113; min=83.201965427417119
2014-11-16 11:48:40,040 [Timer-0] 10010119
```

图 5-3 图片不属于已存在的 POI 计算结果

5) 结果分析: 分析图 5-3 可知, 由于我们输入的经纬度值不在所有 POI 的范围之内, 距离图片最近的 POI 是 10010113, 距离为 83.201965427 米, 大于我们设定的边界值, 则 POI 计算函数根据输入的经纬度值返回的 POI ID 为当前 POI ID 的最大值加一, 10010119, 符合我们对测试结果的预期。

5.3.2 web 前端展示模块测试

1、POI 展示及聚合测试

1) 测试内容: POI 展示及聚合测试主要测试谷歌地图对展示数据的 POI 的展示, 以及我们在对地图进行缩放操作时 POI 的聚合与分散。

2) 测试步骤: 打开浏览器, 输入 web 前端展示页面在服务器上的网址, 查看地图上 POI 的显示; 在地图界面滚动鼠标滑轮或者点击地图边栏的 zoom 标记 (“+”、“-”) 从而改变地图缩放级别, 查看地图上 POI 的显示。

3) 预期结果: 初始打开 web 展示页面, 地图上显示一个表示北邮的 POI, 对地图进行放大操作到一定缩放级别时, 地图上会显示分表表示北邮校内多个地点的 POI, 再将地图的缩放级别调小到初始值, 地图上又只显示一个表示北邮的 POI。

4) 测试结果：实际测试结果如下图所示，图 5-4 是页面初始加载时的 POI 显示，图 5-5 是对地图进行放大操作时地图上 POI 的显示，再对地图进行缩小操作后 POI 显示同图 5-4。

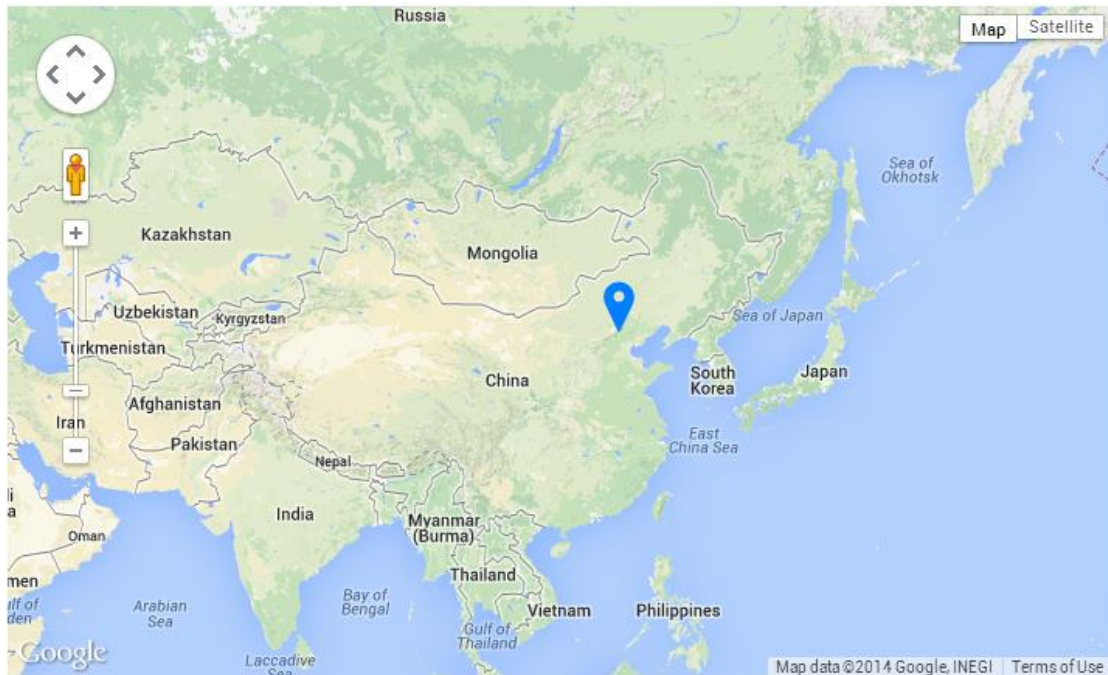


图 5-4 初始地图 POI 显示

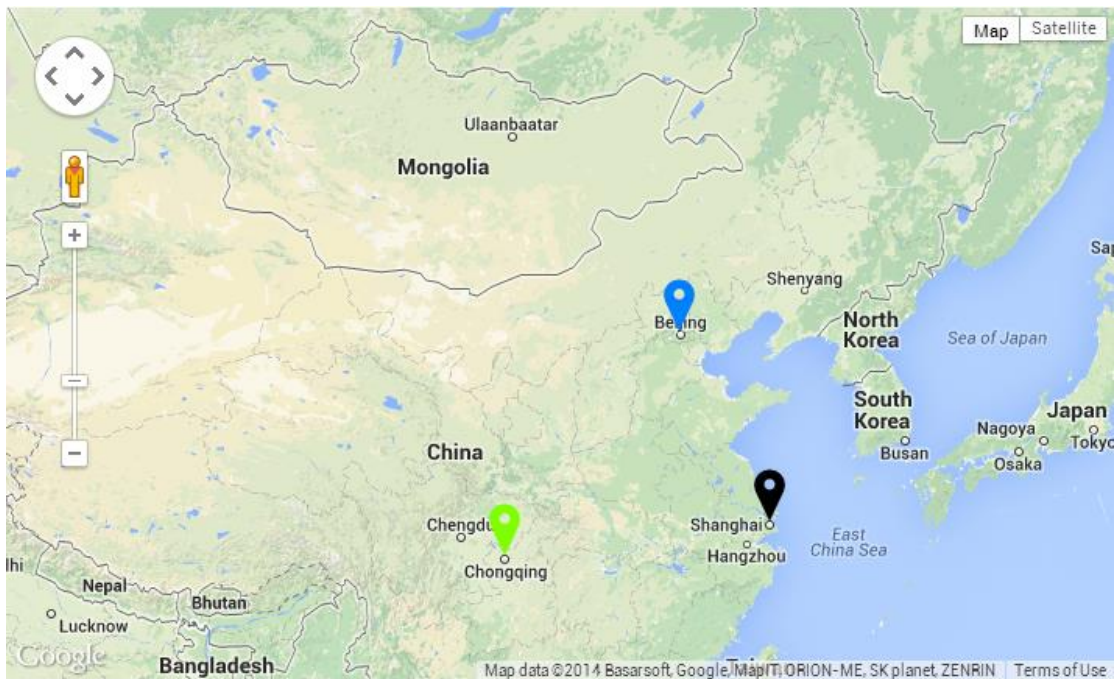


图 5-5 地图放大时 POI 显示

5) 结果分析：分析图 5-4、5-5 可知，地图在初始加载时能够正常地图上的 POI，而在我们对地图进行缩放操作时，地图上的 POI 也能够正常的进行聚合和分散，符合我们对测试结果的预期。

2、PM 2.5 变化曲线随 POI 变化测试

1) 测试内容: PM 2.5 变化曲线随 POI 变化测试主要测试当我们通过缩放或者移动地图的操作来改变地图上显示的 POI 时, 曲线图区域表示 PM 2.5 值变化的曲线会跟随改变, 显示地图上所有 POI 的 PM 2.5 值的变化曲线。

2) 测试步骤: 打开 web 前端展示页面, 查看地图上的 POI 展示以及曲线图区域的 PM 2.5 变化曲线; 在地图界面通过滚动鼠标滑轮或者点击地图 zoom 标记(“+”、“-”)的操作改变地图缩放级别或者拖动地图从而改变地图上显示的 POI 点, 再次观察地图上显示的 POI 以及曲线图区域的 PM 2.5 变化曲线。

3) 预期结果: 初始时地图上显示一个 POI 并且曲线图区域有一条表示地图上 POI PM 2.5 一段时间变化趋势的曲线, 在地图上进行操作改变地图上显示的 POI 之后, POI PM 2.5 变化曲线跟随地图上 POI 的改变而改变, 动态展示在地图上显示的 POI 点一段时间的 PM 2.5 值的变化趋势, 并且 PM 2.5 变化曲线的颜色与地图上 POI 图标的颜色一致。

4) 测试结果: 实际测试结果如下图所示, 图 5-6 是初始时地图上 POI 以及曲线图区域的 PM 2.5 变化曲线的展示, 图 5-7 是进行地图操作改变地图上显示的 POI 之后, 地图上 POI 以及曲线图区域的 PM 2.5 变化曲线的展示。

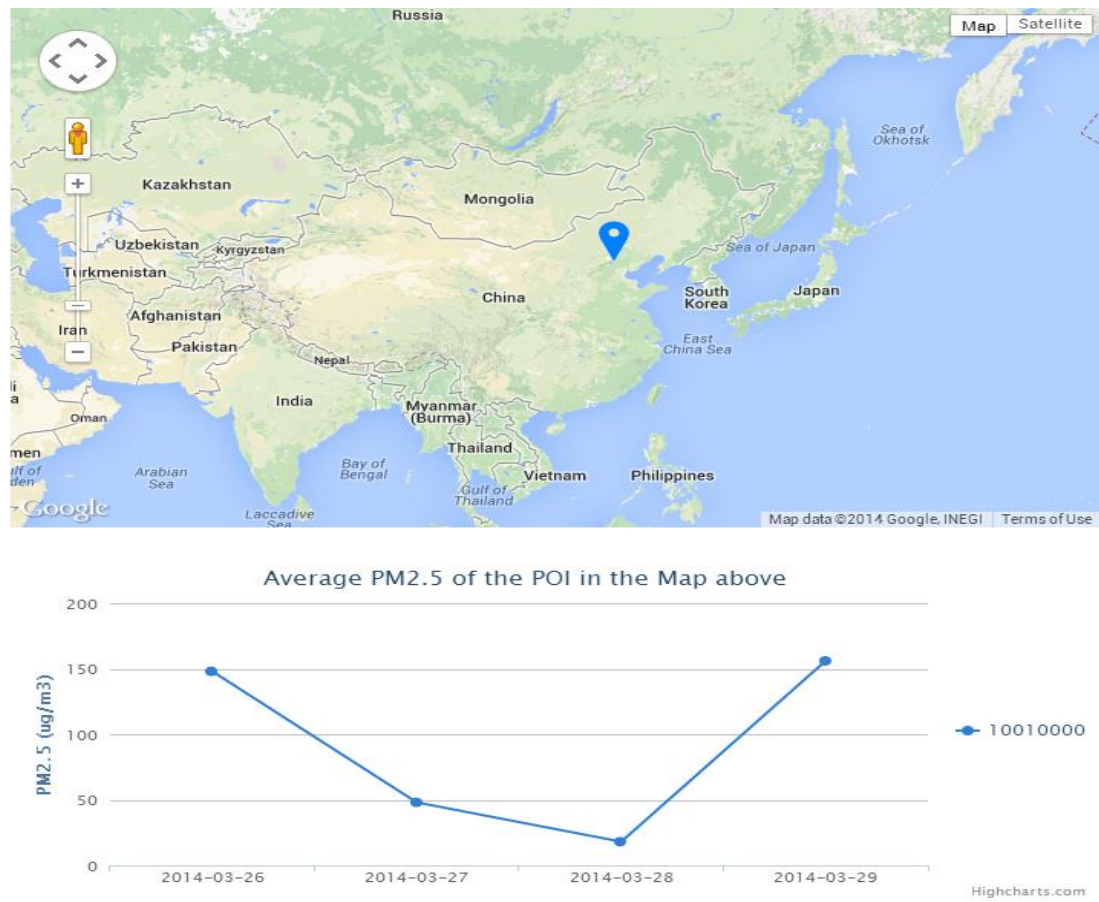


图 5-6 初始 POI 及 PM 2.5 变化曲线展示



图 5-7 地图操作之后 POI 及 PM 2.5 变化曲线展示

5) 结果分析: 分析图 5-6、5-7 可知, 初始地图上只有一个 POI 时, 曲线图区域的 PM 2.5 变化曲线可以正常显示地图上 POI 一段时间的 PM 2.5 值的变化趋势, 在通过地图操作改变地图上显示的 POI 之后, 曲线图区域的 PM 2.5 变化曲线仍然可以正常显示地图上 POI 一段时间的 PM 2.5 值的变化趋势, 并且在地图上 POI 动态变化过程中, PM 2.5 曲线也跟随动态变化, 同时表示 POI PM 2.5 变化趋势的曲线颜色始终同地图上 POI 图标的颜色保持一致, 符合我们对测试结果的预期。

3、图片展示区域随 POI 变化测试

1) 测试内容: 图片展示区域跟随 POI 变化的测试主要测试当我们通过缩放或者移动地图的操作来改变地图上显示的 POI 时, 图片展示区域显示的地图上 POI 对应的图片会跟随改变, 显示地图上所有 POI 的所对应的图片。

2) 测试步骤: 打开 web 前端展示页面, 查看地图上的 POI 展示以及图片展示区域所显示的图片; 在地图界面通过滚动鼠标滑轮或者点击地图 zoom 标记 (“+”、“-”) 的操作改变地图缩放级别或者拖动地图从而改变地图上显示的 POI 点, 再次观察地图上显示的 POI 以及图片展示区域所显示的所有图片。

3) 预期结果: 初始时地图上显示一个 POI 并且图片展示区域只显示该 POI 所对应的图片, 在地图上进行操作改变地图上显示的 POI 之后, 图片展示区域显

示的图片跟随地图上 POI 的改变而改变，动态展示在地图上显示的所有 POI 点所对应的图片，并且图片左上角表示图片所属的 POI 的图标颜色与地图上 POI 图标的颜色一致。

4) 测试结果：实际测试结果如下图所示，图 5-8 显示的是页面初始加载时地图上显示的 POI 以及图片展示区域所显示的图片，图 5-9 显示的是通过地图缩放或者移动操作改变地图上显示的 POI 之后，地图上的 POI 以及图片展示区域的图片展示。

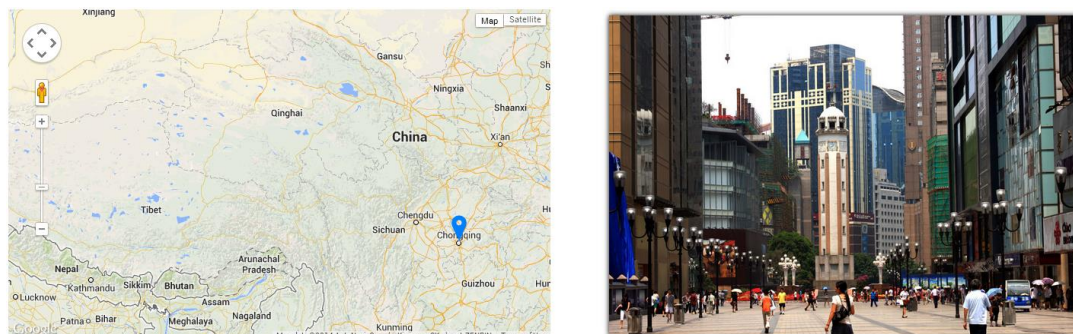


图 5-8 初始 POI 及图片展示

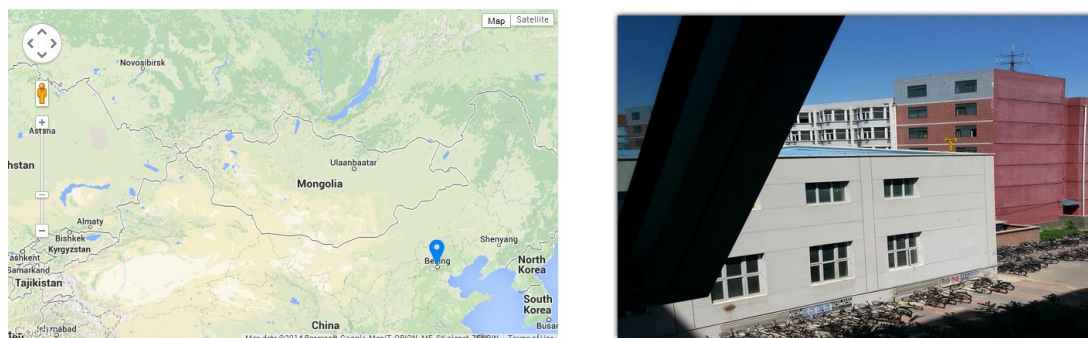


图 5-9 地图操作之后 POI 及图片展示

5) 结果分析：分析图 5-8、5-9 可知，初始地图上只有一个表示重庆的 POI 时，图片展示区域显示的重庆的图片，在通过地图的拖动操作将地图上显示的 POI 变成北京之后，图片展示区域所显示的图片便变成北京的图片，并且在地图上 POI 动态变化过程中，图片展示区域的图片始终跟随地图上显示的 POI 动态变化，同时图片左上角表示图片所属 POI 的图标的颜色始终同地图上 POI 图标的颜色保持一致，符合我们对测试结果的预期。

4、点击 POI 显示 POI 对应图片测试

1) 测试内容：点击 POI 显示 POI 对应图片测试主要测试当我们点击地图上任意一个 POI 时，图片显示区域会显示我们点击的 POI 所对应的图片，并且图片左上角表示图片所属 POI 的图标的颜色同我们在地图上点击的 POI 图标的颜色保持一致。

2) 测试步骤：打开 web 前端展示页面，查看图片展示区域显示的图片以及

图片左上角表示图片所属 POI 的图标的颜色；点击地图上显示的任意一个 POI，再次查看图片展示区域显示的图片以及图片左上角表示图片所属 POI 的图标的颜色。

3) 预期结果：打开 web 前端展示页面时，图片展示区域会随机显示地图上所有 POI 所对应的图片并且图片左上角表示图片所属 POI 的图标的颜色与图片对应的 POI 在地图上显示的图片颜色一致；点击地图上任意一个 POI 之后，图片展示区域只显示被点击 POI 所对应的图片并且图片左上角表示图片所属 POI 的图标的颜色同我们在地图上点击的 POI 图标的颜色保持一致。

4) 测试结果：实际测试结果如下图所示，图 5-10 显示的是初始加载页面时地图上 POI 以及图片展示区域图片的显示，图 5-11 显示的是点击地图上一个 POI 之后图片显示区域所显示的图片。

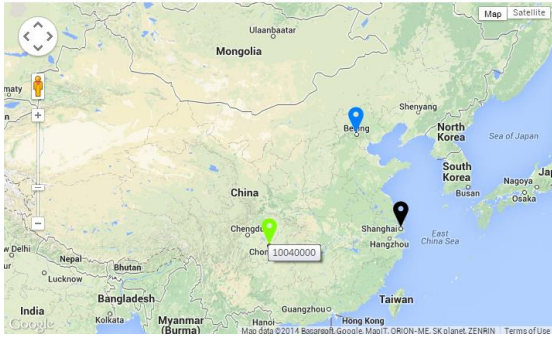


图 5-10 初始 POI 及图片展示区域图片显示

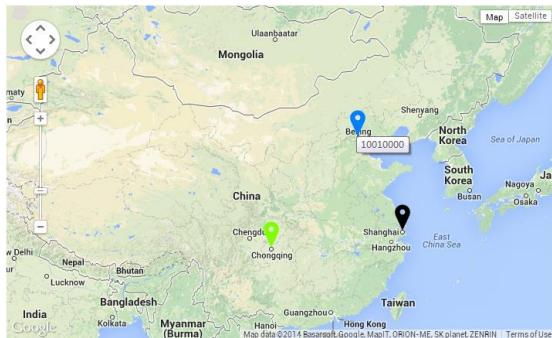


图 5-11 点击某个 POI 后图片展示区域图片显示

5) 结果分析：分析图 5-10、5-11 可知，初始加载 web 前端展示页面时，图片展示区域会随机显示地图上所有 POI 所对应的图片并且图片左上角表示图片所属 POI 的图标的颜色与图片对应的 POI 在地图上显示的图片颜色一致；点击地图上任意一个 POI 之后，图片展示区域只显示被点击 POI 所对应的图片并且图片左上角表示图片所属 POI 的图标的颜色同我们在地图上点击的 POI 图标的颜色保持一致，符合我们对测试结果的预期。

5、图片切换时柱状图跟随显示图片 PM 2.5

1) 测试内容：图片切换时柱状图跟随显示图片 PM 2.5 主要测试当图片进行

幻灯片切换时，表示图片 PM 2.5 值的柱状图跟随图片动态切换，并且柱状图的颜色始终与图片所对应的 POI 在地图上图标的颜色保持一致。

2) 测试步骤：打开 web 前端展示页面，移动地图使地图可视范围内有 POI 显示，查看图片展示区域图片的切换与柱状图显示区域柱状图的切换以及柱状图的颜色。

3) 预期结果：当地图上有 POI 显示之后，图片展示区域会显示地图上 POI 所对应的图片并会进行幻灯片形式的切换，同时柱状图显示区域的柱状图会跟随图片的切换变换高度以表示对应图片的 PM 2.5 值的大小，并且柱状图的颜色始终与图片所对应的 POI 在地图上的图标的颜色保持一致。

4) 测试结果：实际测试结果如下图所示，图 5-12、5-13 分别表示柱状图展示两个不同 POI 的图片时的结果。

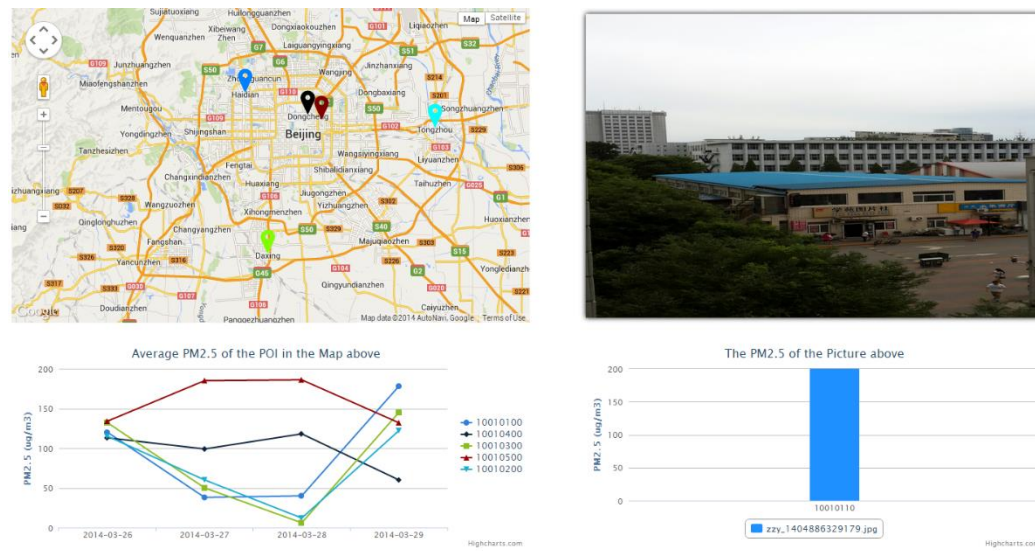


图 5-12 柱状图展示海淀区 POI 对应图片的结果

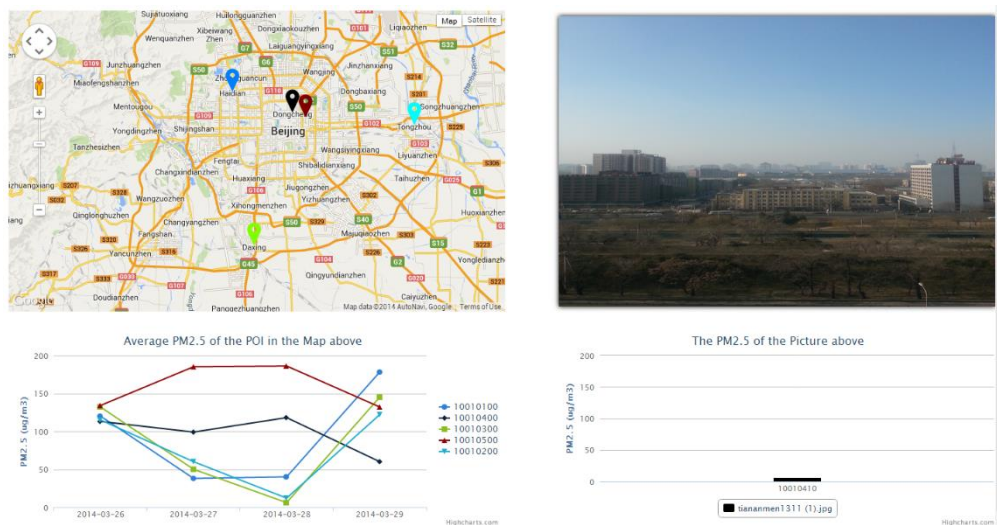


图 5-13 柱状图展示东城区 POI 对应图片的结果

5) 结果分析: 分析图 5-12、5-13 展示的结果可知, 当图片展示区域展示的图片在进行幻灯片形式的切换展示不同 POI 的图片时, 柱状图显示区域的柱状图会跟随图片的切换动态改变从而显示图片的 PM 2.5 值, 并且柱状图的颜色始终与图片所对应的 POI 在地图上图标的颜色保持一致, 符合我们对测试结果的预期。

6、前端界面整体关联测试

1) 测试内容: 前端界面整体关联测试主要测试整个前端界面的整体演示效果, 即以地图作为整个前端界面演示效果改变的触发点, 对地图显示区域、POI PM 2.5 变化曲线显示区域、图片显示区域以及图片 PM 2.5 值柱状图显示区域的整体联动效果进行测试。具体内容就是我们通过移动或者缩放地图的操作, 改变地图上显示的 POI, 然后查看地图上 POI 的聚合情况、POI PM 2.5 变化曲线的变化情况、图片显示区域显示图片的变化情况以及图片 PM 2.5 值柱状图显示区域的变化情况。

2) 测试步骤: 打开 web 前端展示页面, 查看地图上 POI 的显示、曲线图区域的 PM 2.5 变化曲线、图片展示区域所显示的图片以及柱状图显示区域柱状图的切换以及柱状图的颜色改变; 在地图界面滚动鼠标滑轮或者点击地图边栏的 zoom 标记(“+”、“-”)从而改变地图缩放级别或者移动地图改变地图显示的 POI, 再次查看地图上 POI 的显示、曲线图区域的 PM 2.5 变化曲线、图片展示区域所显示的图片以及柱状图显示区域柱状图的切换以及柱状图的颜色改变。

3) 预期结果: 初始打开 web 展示页面, 地图上显示一个表示北邮校园的 POI, 曲线图区域显示一条表示北邮校园一段时间 PM 2.5 变化趋势的曲线并且曲线的颜色与地图上表示北邮的 POI 的图标的颜色一致, 图片展示区域以幻灯片形式显示北邮的图片并且图片左上角表示图片所属的 POI 的图标颜色与地图上 POI 图标的颜色一致, 柱状图区域显示表示每张图片 PM 2.5 值的柱状图并跟随图片的切换而切换同时柱状图的颜色始终与图片所对应的 POI 在地图上的图标的颜色保持一致; 通过地图缩放或者移动操作改变地图显示的 POI 之后, 地图上出现六个表示北邮校内不同地点的 POI, 曲线图区域显示六条表示北邮校内六个地点一段时间 PM 2.5 变化趋势的曲线并且曲线的颜色与地图上表示校内地点的 POI 的图标的颜色一致, 图片展示区域以幻灯片形式随机显示校内六个地点的图片并且图片左上角表示图片所属的 POI 的图标颜色与地图上各个 POI 图标的颜色一致, 柱状图区域显示表示每张图片 PM 2.5 值的柱状图并跟随图片的切换而切换同时柱状图的颜色始终与图片所对应的 POI 在地图上的图标的颜色保持一致。

4) 测试结果: 实际测试结果如下图所示, 图 5-14 显示的是页面初始加载时、

曲线图区域 POI PM 2.5 变化曲线的情况、图片展示区域所显示的图片以及柱状图区域表示图片 PM 2.5 值的柱状图的显示；图 5-15 显示的是通过地图缩放操作改变地图上显示的 POI 之后地图上 POI 的显示情况、曲线图区域 POI PM 2.5 变化曲线的情况、图片展示区域所显示的图片以及柱状图区域表示图片 PM 2.5 值的柱状图的显示情况。

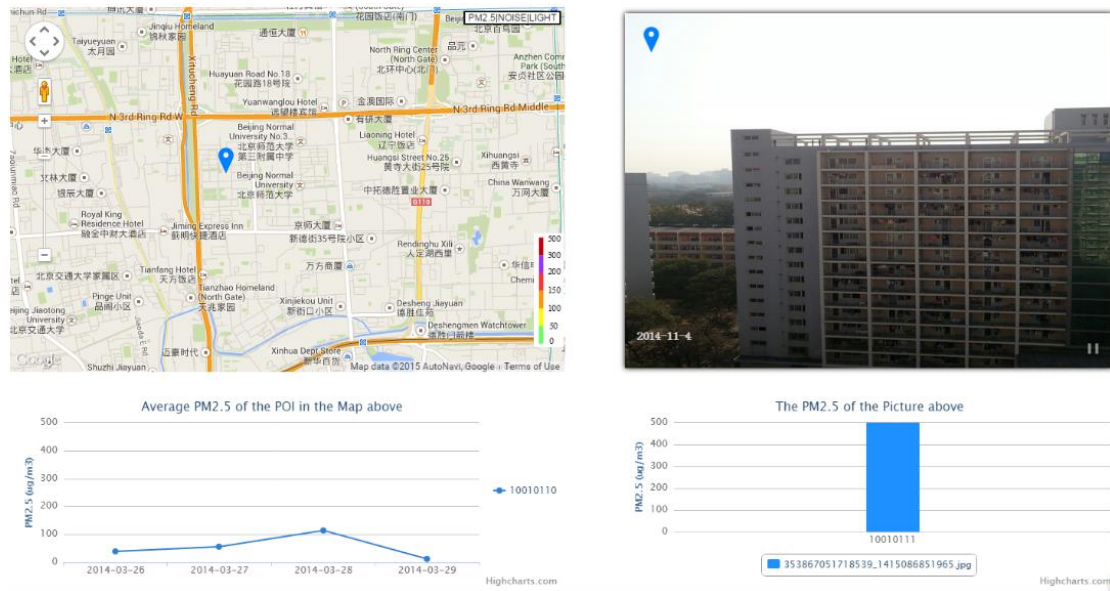


图 5-14 初始时 web 展示界面

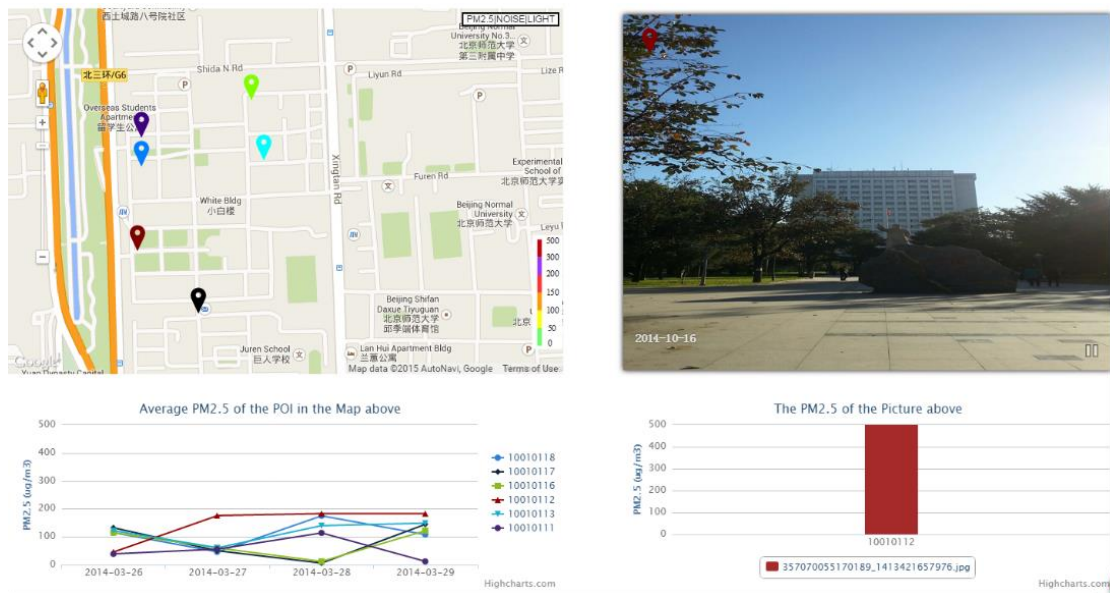


图 5-15 地图操作之后 web 展示界面

5) 结果分析：分析图 5-14、5-15 显示的结果可知，在我们通过移动或者缩放地图的操作改变地图上显示的 POI 时，地图上显示的 POI 会跟随地图的缩放进行聚合和分散；曲线图变化区域 POI PM 2.5 的变化曲线跟随地图上 POI 的改变而动态改变并且 PM 2.5 变化曲线的颜色始终与地图上 POI 的图标的颜色保持一致；图片显示区域所显示图片始终是地图上 POI 所对应的图片并且图片左上

角表示图片所属的 POI 的图标颜色与地图上 POI 图标的颜色一致；表示图片 PM 2.5 值的柱状图显示区域同时跟随图片的切换而切换并且柱状图的颜色始终与图片所对应的 POI 在地图上的图标的颜色保持一致。符合我们对测试结果的预期。

5.4 测试总结

通过上述的测试，验证了我们对数据可视化系统的设计和实现的正确性。虽然在 web 前端展示模块的测试过程中，中间の詳細变化效果无法通过图片进行展示，但是最终的图片结果如实的反应了我们进行测试所期望的结果。

第六章 结束语

本章作为全文的结束,将在前面数据可视化系统的背景介绍、相关技术分析、总体设计、详细设计与实现以及系统测试五章的基础上,对全文进行总结。同时,对本文工作中的不足以及下一步的改进工作进行阐述。最后,对作者在研究生阶段的科研工作整理说明。

6.1 全文总结

通信、微电子、和软件等技术的飞速发展,使得智能移动终端得到了迅猛的普及,而智能移动终端丰富的计算、感知和通信能力使得以移动协作感知技术进行低成本大范围的感知数据收集成为可能。而数据可视化技术和工具的发展使得我们可以对收集的感知数据进行直观、形象的展示,并可以实现用户与数据的交互,增强用户对数据的控制。本文利用 web 前端的相关技术和工具,实现了对移动协作感知平台收集数据的可视化工作。

本文在对移动协作感知背景和数据可视化技术分析的基础上,对移动协作感知平台进行需求分析和总体设计,并在移动协作感知平台上设计实现一个由地图作为关键控制元素的数据可视化界面,对移动协作感知平台的客户端收集的感知数据进行图片和图表形式的可视化,最后对我们实现的数据可视化系统进行测试,验证我们实现的系统的正确性。

6.2 不足和下一步工作

由于本文作者水平有限以及对数据可视化相关技术的研究和理解的不足,本文实现的数据可视化系统有很大的改进空间。

1) 数据可视化系统展示效果比较单一。目前的数据可视化系统在展示效果方面还有很多可以改进的地方,在地图上可以根据 POI 的 PM_{2.5} 的值对 POI 代表的范围进行染色,通过颜色的深浅来表示空气质量的好坏;在曲线图区域可以动态设置变化曲线表示的 PM_{2.5} 的变化时间段,而不是现在只能显示固定时间段的 PM_{2.5} 的变化;图片展示区域对图片的展示形式可以做一些改进,比如多

张图片的对比等形式；柱状图显示区域柱状图的也可以显示多张图片的 PM 2.5 值，然后跟随图片进行切换等等。

2) 后续可以考虑加入基于所有数据的分析结果的展示。随着收集数据的数量和种类的增多，我们可以对收集到的数据进行一些大数据的分析，例如可以根据大量 PM 2.5 值的收集分析环境污染的分布和变化规律，对改善环境提出有针对性的建议。

3) 数据可视化系统 web 展示的性能优化。目前实现的数据可视化系统，只是简单的完成了数据可视化的基本功能需求，对 web 前端展示的性能未做过多的考虑，后续可以采取相应的 web 性能优化方案进行优化，从而提高 web 展示界面的使用体验。

6.3 研究生期间的工作

6.3.1 参加的项目

- IPv6 重编号技术研究

项目时间：2013.4 - 2013.7

IPv6 网络中存在两种主机地址自动配置方式，即 DHCPv6 协议和 ND 协议中的 SLAAC 机制。在 IPv6 网络重编号过程中，需要主机在 DHCPv6/SLAAC 两种配置方式之间切换，本项目对 DHCPv6/SLAAC 的协作和切换问题进行了研究和实验测试。

本人任务即是在学习 DHCPv6 和 SLAAC 的基础上，对当前主流桌面操作系统和手机操作系统对 DHCPv6 和 SLAAC 两种地址配置方式的协作进行实验研究，并输出实验文档。

- 自配置演示系统原型开发

项目时间：2013.10 - 2013.12

本项目通过开发一个原型演示系统，对基于自治网络的自配置特性提出的 CNP（Configuration Negotiation Protocol，配置协商协议）协议进行实验验证。

本人通过学习使用 Dribbler 和 Quagga 软件，实现演示环境的 DHCPv6 和路由功能，并通过 CNP 协商 OSPF 参数，完成 CSG 设备的参数自配置功能。

- 参与式协作感知平台开发

项目时间：2014.3 - 2014.12

参与式协作感知平台利用参与者的移动终端上的传感器收集环境相关数据并上传至服务器。在服务器端，对数据经过分析、处理，能够以多种形式展示给

用户，并根据用户所处的环境对用户提供服务。

本人基于上传数据的位置信息对经过处理的数据在 web 端进行展示，包括在地图上对收集数据的 POI 的聚合展示，以及与 POI 对应的图片、PM 2.5、光照和噪声等环境数据的展示。

6.3.2 完成的文档

1. AM flags 与 lifetime 关系实验报告；
2. Android & IOS 终端测试报告；
3. 移动协作感知平台数据可视化模块的需求分析；
4. 移动协作感知平台数据可视化模块的总体设计；
5. 移动协作感知平台数据可视化模块的详细设计；
6. 移动协作感知平台数据可视化模块的测试报告。

参考文献

- [1] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. Srivastava. Participatory Sensing. In Proceedings of the 1st Workshop on World-Sensor-Web (WSW), 2006:1–5.
- [2] K. Shilton, N. Ramanathan, S. Reddy, V. Samanta, J. Burke, D. Estrin, M. Hansen, M. Srivastava. Participatory design of sensing networks: strengths and challenges. In Proceedings of the Tenth Anniversary Conference on Participatory Design 2008:282-285.
- [3] O. Riva, C. Borcea. The Urbanet Revolution: Sensor Power to the People. Pervasive Computing, IEEE, 2007, 6(2):41-49.
- [4] J. Ma, C. Chen, and J.P. Salomaa. mWSN for large scale mobile sensing. J. Signal Process. Syst., 2008, 51:195–206.
- [5] 张浩, 郭灿. 数据可视化技术应用趋势与分类研究[J]. 软件导刊, 2012, 11(5):169-172.
- [6] 李俊受. 中国和韩国移动互联网发展的比较研究[D]. 北京邮电大学, 2010:25-27.
- [7] 杨娟. 基于监测个人交通方式的一种能量有效的移动感知框架[D]. 西安电子科技大学, 2013:19-20.
- [8] S. Liu, J. Yang, B. Li, C. Fu. Volunteer Sensing: The New Paradigm of Social Sensing. In Proceedings of the 17th International Conference on Parallel and Distributed Systems, 2011:982-987.
- [9] D. Philipp, F. Durr, K. Rothermel. A Sensor Network Abstraction for Flexible Public Sensing Systems. In Proceedings of the 8th International Conference on Mobile Adhoc and Sensor Systems (MASS), 2011:460-469.
- [10] D. Cuff, M. Hansen, J. Kang. Urban sensing: out of the woods. ACM Commun, 2008, 51(3): 24-33.
- [11] R.K. Ganti, F. Ye, H. Lei. Mobile crowdsensing: current state and future challenges. Communications Magazine, IEEE, 2011, 49(11):32-39.
- [12] A. Krause, E. Horvitz. A. Kansal, F. Zhao. Toward Community Sensing. In Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN), 2008:481-492.
- [13] Google Map. API 文档 [OL]. <https://developers.google.com/maps/?hl=zh-cn>

- [14] 王芳. Google 地图开发研究[J]. 计算机与数字工程, 2010, 38(3):146-149.
- [15] 王晶, 温向彬. 利用 jQuery 操作 HTML 元素[J]. 农业网络信息, 2008, (4):98-99.
- [16] jQuery. 官方文档 [OL]. <http://jquery.com/>.
- [17] Highcharts. 官方文档 [OL]. <http://www.highcharts.com/>.
- [18] HTTP 协议. 官方文档 [OL]. <http://www.w3.org/Protocols/>.
- [19] 杨国瑞, 张思博. 基于 AJAX 的 Web 应用架构设计[J]. 现代电子技术, 2006, (15):95-98.
- [20] JSON. 官方文档 [OL]. <http://www.json.org/json-zh.html>.

致 谢

首先要感谢我的导师龚向阳老师，龚老师一丝不苟的作风，严谨求实的科研态度，踏踏实实的精神对我产生巨大影响。龚老师不仅授我以知识，更是以身作则，教育我做人，在两年半的研究生生活之中给我终生受益的教诲。对龚老师的感激之情是无法用言语表达的。在此论文完成之际，请让我对您表示衷心的感谢和祝福。

同时我还要感谢王文东老师和阙喜戎老师。王老师在科研过程中给予我的悉心教导，在项目实践中给以及时的批评与指正，当我遇到难题的时候王老师都能够启发我克服困难，这些经历都使我受益匪浅。阙老师在除了科研项目工作给我指导和建议，还在生活上给予我无微不至的关怀和照顾。对两位老师的感谢之情一言难尽。

另外我还要感谢宋铮师兄、张波师兄和高慧师兄，三位博士师兄科研过程中踏实的科研精神，给我树立了学习的榜样，在项目设计阶段，三位师兄提供了很多建设性意见；感谢参与式感知项目组成员赵露名、李莹、徐登佳、刘肖阳等同学，大家在项目过程中相互帮助克服种种困难才保证了项目顺利完成。同时，也要对宿舍的所有同学表达谢意，感谢你们对我生活上的照顾，让我在科研之外的生活丰富多彩，充满欢乐。

最后要感谢我的父母及家人，是家人的关怀和照顾成就了我今天的成绩，感谢你们对我默默地支持和付出。

作者攻读学位期间发表的学术论文目录

- [1] 袁龙运. 基于 EXata 仿真平台的 Re-ECN 机制的实现与仿真, 已被中国科技文在线刊载录用, 属于规定的核心学术期刊, 署名单位为北京邮电大学