

# 制图综合(Map Generalization)

---

## 比例(Scale)

- Geographic scale
  - 指一般的景观的尺度
- Cartographic scale
  - 指地图的比例，小比例意味着大的规模
- Data Scale
  - 数据收集的分辨率，这对显示地图的比例尺有影响

## Map Generalization(制图综合)

- 变换地图特征/要素，使得匹配地图的比例尺和用途
- 所有地图都包含着几个尺度的概括(generalization)，尤其随着交互式多尺度地图的出现

## Generalization operators

1. Selection:选择哪些要素被地图包括
2. Simplification:减少要素中点的数量
3. Smoothing：使尖锐的角度变为平滑的曲线
4. Aggregation:使点分组称为一块区域
5. Amalgamation:使几块区域变为一块更大的区域
6. Collapse：使得一块具体的要素变成一个象征点
7. Merge：对线要素进行分组
8. Refinement:选择一部分显示
9. Exaggeration：为清楚起见，放大要素的一部分
10. Enhancement：添加可视以增加细节
11. Displacement：为清楚起见分离要素

# WebMapping

---

## WebMap历史

- 在HTML页面上Post地图静态图像
- 动态绘制图像:Map Server 和 Esri ArcIMS->问题在于速度和可扩展性
- 预生成缓存中的瓦片地图图片：Google Maps,Bing Map,OSM
- 更强的互动性：使用SVG和WebGL绘图；HTML5 Adobe Flash和Microsoft Silverlight（Web呈现技术）
- 移动设备增加了Web map的需求

## Web Service

1. 客户端(Client)发生网络请求(web request)和参数(parameters)
2. 网络服务(Web service)绘制地图
3. 服务端(Server)发送响应(response)至客户端

## web map的体系(system architecture)

从外往内

- web client app
- web server
- geospatial server(类似ArcSDE)
- file server 和 database server 后三者+admin and internal client app属于Internal network

## web map的element

- Basemaps(底图)
  - 提供地理内容
  - 比如矢量路线图和遥感图
  - 一般作为瓦片地图
- 专题图层(Thematic layers)
  - 在底图之上(on top of the basemap)
  - 有可能以瓦片地图展示
  - 常用WMS以临时绘制地图，反映数据的急剧变化(rapidly change)
  - 查询服务器获取数据并使用浏览器绘制
- 交互元素(Interactive elements)
  - 弹出窗口(Popup) · 图表(chart)和曲线图(graph) · 时间滑块(slider bars of time)

## Map Request

<http://mesonet.agron.iastate.edu/cgi-bin/wms/nexrad/n0r.cgi?>

SERVICE=WMS&REQUEST=GetMap&FORMAT=image/png&TRANSPARENT=TRUE&STYLES=&VERSION=1.3.0  
&LAYERS=nexrad-

n0r&WIDTH=877&HEIGHT=276&CRS=EPSG:900913&BBOX=-15252263.28954773,2902486.4758432545,-667  
1748.242369267,5602853.811101243

- SERVICE = WMS
- REQUEST = GetMap
- FORMAT = image
- 主要是会读Request <http://a.tile.openstreetmap.org/15/11068/19742.png>
- tile map时的Request
- Z/X/Y--15/11068/19742

## Web Map Service

---

### type

1. 服务端(server)发送预绘制地图图像(predrawn map images(images tiles))到浏览器(browser)/客户端。
2. 服务端发送一串基于文本的地理坐标和属性(a bunch of text-based geographic coordinates and attributes) · 由浏览器绘制。
3. 在请求时在服务端绘制地图，向浏览器发生画好的图像。

## 动态绘制地图服务(dynamically drawn map service)

## 优点

1. 反映数据的最近状态
2. 对于大尺度/大规模的地图有利->瓦片地图的生成·储存或维护困难
3. WMS允许多种符号和样式(SLDs Styled Layer Description)

## 缺点

1. 如果有大量图层需要渲染·运行速度慢
2. 当大量用户同时看地图时容易过载

## Service

GeoServer/QGIS Server/Map Server/deegree

## WMS(Web Map Service)

- 由服务器返回地图的光栅化图片(rasterized image)
- 由OGC(开源地理空间协会 Open Geospatial Consortium)提出(develop)
  - WFS(Feature):允许空间分析和编辑的矢量几何图形和属性。
  - WCS(Coverage):通常用于栅格数据的显示和分析的称为覆盖的数据块。
  - WPS(Processing):允许用户调用服务器上的地理处理工具
- WMS最新版本是1.3.0 2004年发布

## WMS基本规范

- GetMap Operation:检索指定区域和内容的地图图像。
- GetCapabilities Operation:返回有关服务的元数据(支持的操作和参数等)·在发出其他类型的请求时可以使用该元数据作为指南
- GetFeatureInfo operation:允许用户在特定位置查询 WMS 层的属性数据(几何数据和属性值)
  - QUERY\_LAYERS=MAGIC:Connecticut\_Towns\_WGS84&srs=EPSG:4326&INFO\_FORMAT=text/plain&X=400&Y=300
- DescribeLayer|DescribeLayer (有操作但老师没讲)

## SLD

- 描述应该在 WMS 中应用的所有符号大小、颜色和标记
- 两种使用方法：
  - 将 SLD 放在 Web 服务器上的某个文件中·并在发出 GetMap 请求时引用该文件的 URL。
  - 另一种方法是使用 GetMap 请求的可选 SLD\_BODY 参数·并在请求的 URL 中直接提供相关的 XML 块。
- 生成SLD
  - 现有案例中的SLD
  - 从QGIS中导入SLD

## Example

```

<FeatureTypeStyle>
  <Rule>
    <PointSymbolizer>
      <Graphic>
        <Mark>
          <WellKownName>
            circle
          </WellKownName>
          <Fill>
            <CssParameter name='fill'>#FF0000</CssParameter>
          </Fill>
        </Mark>
      </Graphic>
    </PointSymbolizer>
  </Rule>
</FeatureTypeStyle>

```

## Tiled Maps(瓦片地图)

---

### 简介

- WMS 的符号集和标签选择是有限的、缓慢的和不可伸缩的
- 可以不担心性能问题
- 瓦片地图是向大量用户提供复杂网络地图的唯一模型
- 瓦片缓存表示创建缓存时的数据快照
- 需要定期更新，并且只更新有更新的区域
- 制作自己的瓦片地图需要消耗大量的资源
- 现有的瓦片地图:Google,Microsoft,or Esri;CloudMade ,MapBox(using OSM)

### Projection(投影)

- 经过改进的球面墨卡托投影，该投影是为了方便将世界装配到一组方形瓦片上而创建的。
- EPSG:3857,900913(Google)
- 不用于测量(measurement)
  - 在执行任何测量之前投影到更局部的坐标系中

### Scale(比例)

- 推导方式：把整个世界放在一个2x2的瓦片网络上开始，从数学上推导出了比例
  - 用简单的数字来表示级别
  - 没有好的四舍五入的数字？(Not nice well-rounded numbers)
- $\$map\ width = map\ height = 256 * 2^{\{level\}\ pixels}$

### 瓦片坐标和快捷键(Quadkey)

- The rendered map is cut into tiles of 256 x 256 pixels each

- 计算公式：X, Y转为2进制，交错放一起再转回4进制(XY均从0开始)  $XY = 3 = 011, Y = 5 = 101$   
quadkey=1(Y)0(X)0(Y)0(X)1(Y)1(X)=213 base-4

## 瓦片地图服务

- 将瓦片地图放在网络服务端，让人们可以直接爬取他们
  - `http://{s}.somedomain.com/blabla/{z}/{x}/{y}.png`
- 通过Web Service使得瓦片可正常显示；比如Geoserver的WMTS

## 创建瓦片地图

- 在大的（放大的）尺度上，要缓存的瓦片数量可能是巨大的，特别是如果覆盖的区域很广。有大量的瓦片地图包含了很少的信息。（应该也算缺点）
- 按需要创建

## 矢量瓦片地图

- 保持数据样式与数据坐标和属性分离
- 客户端使用预定义的一组样式规则来绘制由服务器实时发送的原始矢量坐标和属性数据的瓦片
- Mapbox Studio

## 矢量瓦片优点

1. 由客户端渲染
2. 矢量瓦片大小小于图片瓦片
3. GeoWebCache 只需要为所有不同的样式存储一个瓦片
4. 无需相应增加带宽，即可绘制高分辨率地图。
5. 客户端具有对实际功能信息的本地访问权限（发送了地理信息）

# Leaflet API

---

## web mapping API

- API是一个可用于编写程序的框架(framework)
- 常见的API:.Net Framework
- Web mapping API:Openlayers,leaflet,google Maps API,ArcGIS API for JS

## leaflet API

- 一个年轻的FOSS web mapping API
- 轻量级，移动友好型，易上手
- 强调使用瓦片地图和GeoJson绘制地图

## 对JS的引用

- HTML 页面需要包含一个指向 web 映射 API 的 JavaScript 文件的 `< script >` 标记
  - 在你自己的服务器上下载并托管 API
    - 最小化载入时间并允许你自定义 API
  - 在别人的服务器上引用 API

- 内容交付网络(CDN)专门托管常用的 API

## 对样式表的引用

### 类似JS的引用

- 许多wm API提供了一些样式表(stylesheets)帮助制作好看的应用
- 本地使用或使用CDN

## 使用leaflet

- Create a map div作为页面主体

```
<div id = "mapid"></div>
```

- Create a Leaflet Map Object and relate it to the div

```
var mymap;  
mymap = new L.map('mapid');
```

- 一个层(layer)可以表示一个 Web 服务，如 WMS 或Tiled服务，它本身包含许多底层数据层。
- 也可能是引用其他文件的层(layer),比如KML和GeoJSON

```
var layer = L.tileLayer.wms("url",{option});  
layer.addTo(mymap);
```

- 图层的样式
  - Tiled map和WMS已经被服务端渲染过了
  - 定义图层怎样被浏览器渲染，比如GeoJSON
- 交互元素
  - 展示位置当鼠标点击或者悬停
  - 窗口弹出
  - 地图查询
  - 打开图层或关闭(Leaflet提供了Layers Control)

## 客户端绘制矢量图层

- 向客户端发送一些包含特征坐标和属性的文本，允许客户端绘制图层。
- 快速
  - 不需要像服务器再次请求，例如弹出示例等
- 可交互
  - 交互图层不需要往返于服务器和客户端

## 挑战

- 大量的要素和复杂的多边形
- label也是浏览器绘制图像的挑战
- web浏览器提供的符号选择比较简单

## 从服务器检索数据的方式

### GeoJSON

- 一种常用的数据格式，用来在网络地图中显示矢量。
- 矢量数据和属性数据用JavaScript object来表示
- GeoJSON不一定包含样式数据
  - 需要在客户端定义
- 简单、快速地加载
- 使用时一般将GeoJSON放在单独的文件中，用JS去调用

```
var geojsonlayer = L.geoJSON(HZBoundJson,{style:Style});
```