

## 缩放到图层范围

当一个一组要素共存于矢量地图图层上，例如圆形和多边形，它们共同构成一个范围，这也称为一个**边界框**。此矩形范围可容纳所有几何形状。获取这样一组特征的程度可能很有用，这样我们就可以针对兴趣点以最佳分辨率重新定位地图，这正是我们在本食谱中将要做的。

我们将以低分辨率在地图上放置一些要素，但是首先，我们将地图分辨率配置为从更高开始。刚开始时功能部件组看起来很小，但是我们将提供一个按钮，可根据用户的组合范围方便地平移和缩放用户使其更接近功能部件组。

可以在中找到源代码 `ch03/ch03-zoom-to-extent/` 。我们最终得到的内容类似于以下屏幕截图：

怎么做...

发现你 可以使用以下说明将图层上的某些要素缩放到一定程度：

- 1 创建一个具有OpenLayers依赖项的HTML文件，一个 `div` 用于保存地图的元素以及一个用于容纳按钮的面板：

```
<div id="js-map"></div>
<div>
  <button id="js-zoom">Zoom to extent</button>
</div>
```

复制

- 2 创建一个自定义JavaScript文件，并为背景映射创建一个栅格图层：

复制

```
var rasterLayer = new ol.layer.Tile({
  source: new ol.source.OSM({
    attributions: [
      new ol.Attribution({
        html: 'Tiles courtesy of ' +
          '<a href="http://www.thunderforest.com">Andy Allan</a>'
      }),
      ol.source.OSM.ATTRIBUTION
    ],
    url: 'http://{a-c}.tile.thunderforest.com/cycle/' +
      '{z}/{x}/{y}.png'
  })
});
```

### 3 创建具有某些几何形状的矢量层，如下所示：

复制

```
var vectorLayer = new ol.layer.Vector({
  source: new ol.source.Vector({
    features: [
      new ol.Feature(new ol.geom.Circle([-376645, 7762876], 200)),
      new ol.Feature(new ol.geom.Circle([-375955, 7762195], 200)),
      new ol.Feature(new ol.geom.Circle([-376953, 7761632], 200))
    ]
  })
});
```

### 4 实例化一个新的 map：

复制

```
var map = new ol.Map({
  view: new ol.View({
    zoom: 11,
    center: [-372592, 7763536]
  }),
  target: 'js-map',
  layers: [rasterLayer, vectorLayer]
});
```

### 5 最后，添加按钮 click 处理程序和逻辑以更新地图，如下所示：

复制

```
document.getElementById('js-zoom')
  .addEventListener('click', function() {
    map.beforeRender(
      ol.animation.pan({
        source: map.getView().getCenter(),
        duration: 150
      }),
      ol.animation.zoom({
        resolution: map.getView().getResolution(),
        duration: 500,
        easing: ol.easing.easeIn
      })
    );
    map.getView().fit(
      vectorLayer.getSource().getExtent(), map.getSize()
    );
  });
```

怎么运行的...

让我们集中讨论单击按钮时发生的情况，因为这是利用OpenLayers代码重新定位地图的地方：

复制

```
map.beforeRender(
  ol.animation.pan({
    source: map.getView().getCenter(),
    duration: 150
  }),
  ol.animation.zoom({
    resolution: map.getView().getResolution(),
    duration: 500,
    easing: ol.easing.easeIn
  })
);
```

点击内处理程序，我们首先添加两个动画预渲染函数（`ol.animation.pan` 和 `ol.animation.zoom`）。我们暂时将根据地图项的范围来重新定位地图视图的中心位置和缩放级别，并且我们希望通过平滑过渡来执行此更改。这些类型的过渡行为以及它们如何工作的细节先前已经涵盖了**移动周围的地图视图**在配方第1章 ([/book/web\\_development/9781785287756/1](/book/web_development/9781785287756/1))，**Web制图基础知识**。

复制

```
map.getView().fit(
  vectorLayer.getSource().getExtent(), map.getSize()
);
```

地图视图（通过 `getView` `map`方法检索）包含一个称为的方法 `fit`。的目的 `fit` 是将给定的几何形状或范围容纳到自定义区域（通常与地图视口大小相同，但不一定如此）。

该 `fit` 方法期望第一个参数是两种类型之一，我们对type感兴趣 `ol.Extent` 。此类型只是构成我们要素边界框的一组坐标。向量源有一个称为的方法 `getExtent` ，使我们能够检索此信息。我们传入结果 `ol.Extent` 数组作为第一个参数。

的第二个参数 `fit` 期望类型为 `ol.Size` ，它只是一个包含框的宽度和高度（以像素为单位）的数组。我们希望要素的范围尽可能地适合地图的视口，因此我们使用 `getSize` `map`方法，该方法以所需格式返回大小。

该 `fit` 方法可以选择使用第三个参数-一个具有可配置属性的对象，例如 `minResolution` ，可用于确定调用此函数时要查看功能范围的最小分辨率。

该 `fit` 方法会以可用的最佳分辨率自动调整地图视图，以便显示所有功能。

---

◀ 上一节 (/book/web\_development/9781785287756/3/ch03lvl1sec32/removing-or-cloning-features)

下一节 ▶ (/book/web\_development/9781785287756/3/ch03lvl1sec34/adding-text-labels-to-geometries)

---

