

UTFGrid: растровые тайлы с атрибутикой

[Обсудить в форуме](#) Комментариев — 0

Эта страница опубликована в основном списке статей сайта по адресу <http://gis-lab.info/qa/utfgrid.html>

Приведено описание стандарта UTFGrid и рассмотрен пример его использования.

Содержание

- [1 Введение](#)
- [2 Структура UTFGrid-файла](#)
- [3 Как это работает](#)
- [4 Преимущества и недостатки](#)
- [5 Заключение](#)

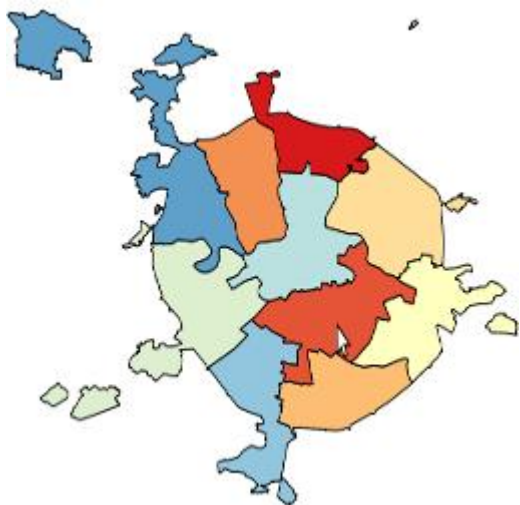
Введение

Основное назначение любого картографического web-приложения - это предоставление клиенту пространственной информации, включающей геометрическую и атрибутивную составляющие. Поэтому, в любом случае, на начальном этапе каждый разработчик должен ответить на вопрос: "Какой формат передачи данных следует использовать в конкретном случае?" Как известно, все форматы передачи пространственной информации можно разделить на векторные и растровые. Использование векторных форматов (KML, GML, GeoJSON и т.д.) с одной стороны очень удобно: в клиентском приложении сразу становятся доступными и геометрическая, и атрибутивная составляющие, что обеспечивает простоту добавления приложению интерактивности. С другой же - область применения векторных форматов довольно ограничена и определяется главным образом объёмом передаваемых данных. Например, если векторный слой содержит несколько миллионов объектов, то во-первых, клиенту придется ждать пока эти данные будут загружены браузером, а во-вторых пока они будут отрисованы. При этом попытка изменения масштабного уровня вновь потребует перерисовки объектов, что также может занять продолжительное время. Естественно, что ни о каком комфорте при работе с таким приложением речи не идёт, что делает использование векторных форматов недопустимым. В этом случае более подходящим решением является использование растров.

Однако как быть с атрибутикой? Допустим, что пользователю нужна возможность получения информации об объекте, на который указывает указатель мыши. Конечно, можно реализовать серверный скрипт, которому будут передаваться текущие координаты, а в ответ возвращаться нужная информация (в случае WMS слоёв можно воспользоваться стандартной возможностью любого WMS-сервера - [поддержкой запросов GetFeatureInfo](#)). Однако в данной статье мы рассмотрим другой способ решения этой задачи, предложенный компанией [MapBox](#), известный под именем UTFGrid ([спецификация](#)).

UTFGrid представляет собой стандарт, описывающий эффективный с точки зрения объёма передаваемых данных, способ кодирования атрибутивной информации объектов, представленных на растре. Данный стандарт был специально разработан для использования в Веб-браузерах и может использоваться, например, для вывода всплывающих подсказок при перемещении указателя мыши (например как [здесь](#)). В качестве контейнера UTFGrid использует формат JSON.

Рассмотрим пример данных, закодированных при помощи UTFGrid. Предположим, у нас есть некоторый растровый слой:



Пример исходного растра

тогда его представление при помощи UTFGrid будет выглядеть следующим образом:

```

1.      {
2.          "data": {
3.              "BOG": {
4.                  "name": "район Богородское"
5.              },
6.              "VNUK": {
7.                  "name": "район Внуково"
8.              },
9.              "ALT": {
10.                 "name": "Алтуфьевский район"
11.             },
12.             "SIL": {
13.                 "name": "район Силино"
14.             },
15.             "DOR": {
16.                 "name": "район Дорогомилово"
17.             },
18.             "ZUZ": {
19.                 "name": "район Зюзино"
20.             },
21.             "AIR": {
22.                 "name": "район Аэропорт"
23.             },
24.             "BIR": {
25.                 "name": "район Бирюлево Восточное"
26.             },
27.             "VESH": {
28.                 "name": "район Вешняки"
29.             },
30.             "LEF": {
31.                 "name": "район Лефортово"
32.             }
33.         },
34.         "grid": [
35.             "      !",
36.             " !!!!!",
37.             " !!!!!!!",
38.             " !!!!!!! !",
39.             " !!!!!!! !!",
40.             " !!!!! !!! !!!!!",
41.             " !!!!! ! !!!!!",
42.             " !!!!! !",
43.             "      !!!      !!!      $$$$ ",
44.             "      !!!!!      $$$ "

```

```
45.      "      !!!!!      $$$$      "
46.      "      !!!!!      $$$      "
47.      "      !!      $$      "
48.      "      !!      $$ $$      "
49.      "      !!      %$$$$$$$      "
50.      "      !!!      %%%%%%%%%$      "
51.      "      !!!      %%%%%%%%%$$$$$$$$$      "
52.      "      !!!!!%%%%%%%%%$$$$$$$$$$$$$      "
53.      "      !!!!!%%%%%%%%%$$$$$$$$$$$$###      "
54.      "      !!!!!%%%%%%%%%$$$$$$$$$$$$###      "
55.      "      !!!!!!!%%%%%%%%%$$$$$$$$$$$$$#####      "
56.      "      !!!!!!!%%%%%%%%%&&$$$$$$$$$#####      "
57.      "      !!!!!!!%%%%%%%%%&&&&&&#####      "
58.      "      !!!!!!!%%%%%%%%%&&&&&&#####      "
59.      "      !!!!!!!%%%%%%%%%&&&&&&##### ###      "
60.      "      '!!!!!!%%%%%%%%%&&&&&&#####      "
61.      "      !!!!!!!%%%%%%%%%&&&&&&#####      "
62.      "      !!!!!!!%%%%%%%%%&&&&&&#####      "
63.      "      '!!!!!!%%%%%%%%%&&&&&&#####      "
64.      "      '!!!!!!%%%%%%%%%&&&&&&#####      "
65.      "      ' ' ' ' ' ' ' ' ' ' ' ' %&&&&&&&&&&&&#####      "
66.      "      ' ' ' ' ' ' ' ' ' ' ' ' %&&&&&&&&&&&& ( ( #####      "
67.      "      ' ' ' ' ' ' ' ' ' ' ' ' &&&&&&&&&&&& ( ( ##### ) #      "
68.      "      ' ' ' ' ' ' ' ' ' ' ' ' &&&&&&&&&&&& ( ( ##### ) ) # ) )      "
69.      "      ' ' ' ' ' ' ' ' ' ' ' ' &&&&&&&&&&&& ( ( ( # ) ) ) ) ) )      "
70.      "      ' ' ' ' ' ' ' ' && ' &&&&&&&&&&&&&&&&&&&& ( ( ( ( ( ) ) ) ) ) ) )      "
71.      "      ' ' ' ' ' ' ' ' ' ' &&&&&&&&& ( ( ( ( ( ( ( ( ( ) ) ) ) ) ) ) ) ) )      "
72.      "      ' ' ' ' ' ' ' ' ' ' &&&& ( ( ( ( ( ( ( ( ( ( ) ) ) ) ) ) ) ) ) )      "
73.      "      ' ' ' ' ' ' ' ' ' ' ( ( ( ( ( ( ( ( ( ( ( ( ) ) ) ) ) ) ) ) ) )      "
74.      "      ' ' ' ' ' ' ' ' ' ' ( ( ( ( ( ( ( ( ( ( ( ( ( ) ) ) ) ) ) ) ) ) )      "
75.      "      ' ' ' ' ' ' ' ' ( ( ( ( ( ( ( ( ( ( ( ( ( ) ) ) ) ) ) ) ) ) )      "
76.      "      ' ' ' ' ' ' ' ' * * * ( ( ( ( ( ( ( ( ( ( ( ( ) ) ) ) ) ) ) ) ) )      "
77.      "      ' ' ' ' ' ' ' ' * * * * ( ( ( ( ( ( ( ( ( ( ( ( ) ) ) ) ) ) ) ) ) )      "
78.      "      ' ' ' ' ' ' ' ' * * * * * * * ( ( ( ( ( ) ) ) ) ) ) ) ) ) )      "
79.      "      ' ' ' ' ' ' ' ' * * * * * * * * ( + + ( ( ) ) ) ) ) ) ) )      "
80.      "      ' ' ' ' ' ' ' ' ' ' * * * * * * * * ( + + + + + ( ) ) ) ) ) ) ) )      "
81.      "      ' ' ' ' ' ' ' ' ' ' * * * * * * * * ( ( + + + + + + + + + + ) )      "
82.      "      ' ' ' ' ' ' ' ' * * * * * * * * ( ( + + + + + + + + + + )      "
83.      "      '      * * * * * * * * + + + + + + + + + + + + + + + +      "
84.      "      ' ' ' ' ' ' ' ' ' ' * * * * * * * * + + + + + + + + + + + +      "
85.      "      ' ' ' ' ' ' ' ' ' ' * * * * * * * * + + + + + + + + + + + +      "
86.      "      ' ' ' ' ' ' ' ' ' ' * * * * * * * * + + + + + + + + + + + +      "
87.      "      ' ' ' ' ' ' ' ' ' ' * * * * + + + + + + + + + + + +      "
88.      "      ' ' ' ' ' ' ' ' ' ' * * * + + + + + + + + + + + +      "
89.      "      ' ' ' ' ' ' ' ' ' ' * * * * + + + + + + + +      "
90.      "      '      * * * *      "
91.      "      '      * * *      "
92.      "      '      * * * *      "
93.      "      '      * * * * * * *      "
94.      "      '      * * * * * * *      "
95.      "      '      * * * * *      "
96.      "      '      * * * * *      "
97.      "      '      * *      "
98.      "      '      * * *      "
99.      ],
100.     "keys": [
101.         " ",
102.         "SIL",
103.         "BOG",
104.         "ALT",
105.         "AIR",
106.         "DOR",
107.         "VNUK",
108.         "LEF",
109.         "VESH",
```

```
110.         "ZUZ" ,
111.         "BIR"
112.     ]
113. }
```

Структура UTFGrid-файла

Представленный файл содержит три основных свойства: `data`, `grid`, `keys`. Свойство `keys` содержит список идентификаторов всех объектов тайла, в `data` представлены атрибутивные значения этих объектов. Свойство `grid` содержит текстовое представление растрового тайла. В нашем примере `grid` содержит 64 строки по 64 символа в каждой, то есть каждый такой символ соответствует объекту размером 4x4 пиксела в оригинальном растре (согласно спецификации UTFGrid по умолчанию использует шаг дискретизации 2x2 пиксела, но этот размер можно изменять, что соответственным образом скажется на размере UTFGrid-файла).

Как это работает

- Когда пользователь осуществляет некоторые действия (перемещает указатель мыши или кликает по карте) над тайлом, то происходит загрузка UTFGrid-файла, соответствующего этому тайлу;
- Определяется положение указателя мыши внутри тайла и этому положению ставится в соответствие символ из свойства `grid` загруженного UTFGrid-файла. Предположим, указатель мыши наведён на объект "район Аэропорт". Соответствующий символ грида - "%";
- Согласно [спецификации](#) вычисляется порядковый номер идентификатора в списке `keys`: находим [десятичный код](#) символа грида в нашем случае он равен 37, вычитаем из него 1 (так как $37 > 35$) и вычитаем 32. Получаем 4 (в Python для выполнения этой операции можно воспользоваться командой `ord`, обратная операция выполняется при помощи `unichr`);
- В списке `keys` находим идентификатор, расположенный по указанному индексу (нумерация начинается с 0). В нашем случае `keys4="AIR"`;
- Внутри объекта `data` по полученному идентификатору находим атрибутивную информацию.

UTFGrid используется, в частности, в таком популярном продукте как [TileMil](#), [например](#). Поддержка UTFGrid также реализована в библиотеке [OpenLayers](#). Ознакомиться в интерактивном режиме с UTFGrid можно на сайте [visible map](#).

Преимущества и недостатки

С одной стороны, при небольшом количестве кликов на тайл, количество запросов будет сопоставимо с `GetFeatureInfo` для WMS-слоя. Однако объем переданных клиенту данных будет больше, так как веб-браузеру будет "в нагрузку" передана информация о тех объектах, которые не попадают в область его интересов. Так же клиенту будет передана "лишняя" информация в случае если один элемент будет размещен на нескольких тайлах, в том числе на тайлах разного уровня. С другой стороны, задача раздачи статического контента, как правило, куда менее затратна для веб-сервера с точки зрения потребляемых ресурсов и проще поддается масштабированию при росте нагрузки на сервер.

По этой причине, вряд ли будет рационально использовать этот механизм для векторного слоя состоящего из миллионов объектов, в случае если им пользуется полтора пользователя. В проектах же с большой аудиторией, где регулярно происходит обращение к одним и тем же данным, использование UTFGrid видится вполне целесообразным.

Так же, к недостаткам можно отнести то факт, что если растровый тайл построен на базе векторных данных, содержащих перекрывающиеся объекты, то с помощью UTFGrid мы не сможем закодировать информацию о таких объектах, то есть элементы списка `keys` сами по себе не могут быть списками.

Заключение

Мы рассмотрели лишь описание стандарта UTFGrid, пример создание полноценного сервиса с поддержкой UTFGrid будет рассмотрен в одной из следующих статей. Пример UTFGrid, представленный в статье, был сгенерирован при помощи [ogr_renderer.py](#).

Последнее обновление: 2014-05-15 01:44

Дата создания: 01.09.2012

Автор(ы): [Денис Рыков](#)