

# osgEarth

یک sdk جغرافیای است. یک engine از OpenSceneGraph است

ویژگی ها :

1. برنامه سه بعدی جغرافیایی
2. تجسم مدل و تصاویر را مستقیماً در حد امکان فراهم می کند
3. با استاندارد ها داده های نقشه ای را فراهم میکند

نیازمندی ها

1. نقشه پایه آماده باشد
2. به خدمات استاندارد wms یا Tms دسترسی داشته باشد
3. داده های ذخیره شده محلی با تصاویر مبتنی در وب باید یکپارچه سازی شود
4. در زمان اجرا لایه های جدید اضافه شود
5. باید با داده های که در طول زمان تغییر می کند مقابله کرد.

## ۱-تکنولوژی مورد استفاده

- برنامه های این sdk را با استفاده از 11 ++ c Api با امکانات کامل توسعه داده شده است
- امکانات بستر اینترنت و local از پشتیبانی نیز می کند.
- با استفاده از یک فایل XML ساده، یک نقشه دو بعدی یا سه بعدی را به سرعت و به راحتی اجرا کنید
- نقشه دو بعدی یا سه بعدی را با استفاده از OPENGL طراحی میکنند
- استفاده از کتابخانه gdal برای ترجمه فرمت های جغرافیایی (شطرنجی و برداری) است

## ۲-راهنمای استفاده

### ۱-۲ ابزار های osgearth

1-osgearth\_viewer

برای نمایش و خواندن فایل نقشه : این دستور یک فایل earth. می خواند و یکسری آرگومان می گیرد

osgearth version

ورژن استفاده شده را برمیگرداند

osgearth\_cache

برای مدیریت کش استفاده می شود و اصولاً غیر تعاملی می باشد که با آرگومان seed می آید

osgearthh\_conv

محتویات یک tilesource در یک پکیج دیگری کپی می کند.

osgEarth\_package

یک بسته مبتنی بر tms قابل توزیع مجدد از یک فایل earth تولید میکند منظور از tms همان tile map service میباشد

osgEarth\_tfs

یک مجموعه داده tfs از منبع ویژگی مانند Shapefile می سازد .با پیش پردازش ویژگی های خود در ساختار شبکه ای ارائه شده توسط tfs می توانید عملکرد مجموعه داده های بزرگ را به طور قابل توجهی افزایش دهید. علاوه بر این بسته tfs تولید شده می تواند توسط هر وب سرور استاندارد ا ارائه شود

## osgearth\_backfill

یک ابزار تخصصی برا پردازش داده ها **tms** می باشد. برخی از خدمات نقشه برداری وب از مجموعه داده های کاملاً متفاوتی در سطوح بزرگنمایی مختلف استفاده می کنند. به ما این امکان را می دهد که علاوه بر ساخت فایل **tms** با روش های معمول با این ابزار می شود میزان بزرگنمایی و لود کردن را تغییر داد.

## osgearth\_overlayviewer

سبب نشان دادن دو نقشه یکی همان نقشه معمولی و دیگری فرستوم های مرزی را برای همپوشانی نشان می دهد.

## osgearth\_boundarygen

ساختن یک مدل هندسی برای اضافه کردن به بیس نقشه استفاده می شود.

## ۲-۲ earth file

یک توصیف **xml** از یک نقشه است این روش ساده و سریعترین راه برای راه اندازی یک نقشه می باشد. این فایل با پسوند

**osg** می باشد

امکانات **earthfile**

- نوع نقشه را مشخص میکند
- منابع تصویر، ارتفاع، بردار و مدل برای استفاده
- در حافظه کش ذخیره میکند

### ۲-۲-۱ تعریف نقشه در earthfile

```
<map name="MyMap">
  <image name="bluemarble" driver="gdal">
    <url>world.tif</url>
  </image>
</map>
```

### ۲-۲-۲ تعریف چند لایه از نقشه

```
<map name="Transportation">

  <!--Add a base map of the blue marble data-->
  <image name="bluemarble" driver="gdal">
    <url>c:/data/bluemarble.tif</url>
  </image>

  <!--Add a high resolution inset of Washington, DC-->
  <image name="dc" driver="gdal">
    <url>c:/data/dc_high_res.tif</url>
  </image>

</map>
```

### ۲-۲-۳ Elevation اضافه کردن لایه ارتفاع با استفاده از

```
<map name="Elevation">

  <!--Add a base map of the blue marble data-->
  <image name="bluemarble" driver="gdal">
    <url>c:/data/bluemarble.tif</url>
  </image>

  <!--Add SRTM data-->
  <elevation name="srtm" driver="gdal">
    <url>c:/data/SRTM.tif</url>
  </elevation>

</map>
```

### ۲-۲-۳ استفاده از caching

به جای اینکه داده ها را دوباره پردازش کند آنها را در cache قرار می دهد

```
<map name="TMS Example">

  <image name="metacarta blue marble" driver="tms">
    <url>http://readymap.org/readymap/tiles/1.0.0/7/</url>
  </image>

  <options>
    <!--Specify where to cache the data-->
    <cache type="filesystem">
      <path>c:/osgearth_cache</path>
    </cache>
  </options>

</map>
```

### ۲-۳ حافظه cache

بسته به ماهیت داده های منبع، osgEarth ممکن است قبل از اینکه تبدیل به یک کاشی زمین شود، باید برخی از پردازش ها را روی آن انجام دهد. این ممکن است شامل دانلود، بازپرداخت، برش، تکه کردن، یا ترکیب کردن باشد. این عملیات می تواند گران شود. با راه اندازی حافظه cache می توانید osgEarth را به گونه ای تنظیم نمود تا نتیجه پردازش را ذخیره کند تا دفعه بعد که به همان کاشی مورد نیاز است، دوباره آن را تکرار نکند.

```
<map>
  <options>
    <cache type="filesystem">
      <path>folder_name</path>
    </cache>
```

## spatial Refrence 2-4

مجموعه ای از مختصات را به یک مکان واقعی متناظر روی زمین نگاشت می کند.  
نمونه coordinates

(-121.5, 36.8, 2000.0)

Coordinate System Type: Geographic  
Units: Degrees  
Horizontal datum: WGS84  
Vertical datum: EGM96

که به چهار روش مقدار دهی می شود

- [Coordinate System Type](#)
- [Horizontal Datum](#)
- [Vertical Datum](#)
- [Projection](#)

## ۲-۴-۱ سیستم coordinate

- ☐ Geographic (longitude and latitude).
- ☐ Projected (X,Y)
- ☐ ECEF (X,Y,Z)

## ۲-۴-۲ داده های افقی

داده های رایج مورد استفاده عبارتند از WGS84 و NAD83 در آمریکای شمالی و ETR89 در اروپا.

## ۲-۴-۳ داده های عمودی

داده های رایج مورد استفاده عبارتند از Geodetic و EGM84 geoid و EGM96 geoid و EGM2008 geoid

## ۲-۴-۲ داده های Projection

در این روش با استفاده از یک فرمول ریاضی و همچنین با استفاده از gdal میتوان مختصات مختلفی را داد

روش های زیادی نیز برای این نوع تعریف وجود دارد که در ادامه نمونه هایی از آن را می بینیم

```
PROJCS["NAD_1983_UTM_Zone_15N",  
  GEOGCS["GCS_North_American_1983",  
    DATUM["D_North_American_1983",  
      SPHEROID["GRS_1980",6378137.0,298.257222101]],  
    PRIMEM["Greenwich",0.0],  
    UNIT["Degree",0.0174532925199433]],  
  PROJECTION["Transverse_Mercator"],  
  PARAMETER["False_Easting",500000.0],  
  PARAMETER["False_Northing",0.0],  
  PARAMETER["Central_Meridian",-93.0],  
  PARAMETER["Scale_Factor",0.9996],  
  PARAMETER["Latitude_Of_Origin",0.0],  
  UNIT["Meter",1.0]]
```

+proj=utm +zone=15 +ellps=GRS80 +units=m +no\_defs

## ۵-۲ ویژگی ها و نماد ها

ویژگی های هندسی برداری هستند. بر خلاف تصاویر و داده های ارتفاعی (که شطرنجی هستند)، ویژگی وضوح نمایشگر گسسته ای ندارد. osgearth میتواند در هر سطحی آنها را نمایش دهد.

ویژگی ترکیبی از سه جز می باشد

- ترکیبی از vector geometry
- ویژگی ها مثل نام ها و مقادیر
- مرجع فضای(مختصات)

### ۱-۵-۲ ایجاد ویژگی لایه

مدل لایه های ما به دو صورت می باشد

- شطرنجی شده به عنوان یک لایه تصویر
- یک لایه مدل

شطرنجی کردن لایه بسیار ساده است و سعی می شود تصاویر را به صورت کاشی در بیاورند و از آن کاشی تصویر لایه معمولی به وجود آورند.

```
<image name="my layer" driver="agglite">
  <features name="states" driver="ogr">
    <url>states.shp</url>
  </features>
  <styles>
    <style type="text/css">
      states {
        stroke:      #ffff00;
        stroke-width: 2.0;
      }
    </style>
  </styles>
</image>
```

در روش دوم از طریق یک فرایند مدل را به نقاط و اشکال هندسی تبدیل میشود این روش در osgEarth بیشتر کاربرد دارد.

```
<feature_model name="boundaries">
  <features name="states" driver="ogr">
    <url>states.shp</url>
  </features>
  <styles>
    <style type="text/css">
      states {
        stroke:      #ffff00;
        stroke-width: 2.0;
      }
    </style>
  </styles>
</feature_model>
```

در این روش می شود از رفرنس دهی استفاده کرد تا بتوان از در چند لایه به کار برد.

```
<feature_source name="data_layer" driver="ogr">
  <url>states.shp</url>
</feature_source>

<feature_model name="boundaries" feature_source="data_layer">
  <styles>
    <style type="text/css">
      states {
        stroke:      #ffff00;
        stroke-width: 2.0;
      }
    </style>
  </styles>
</feature_model>
```

## ۲-۵-۲ ایجاد style

ممکن است برای یک لایه به ویژگی های احتیاج داشته باشید که به صورت کامپوننت استفاده شود .

یک نمونه از کامپوننت style می باشد که به صورت CSS می توان به آن مقداردهی کرد

که به دو روش Expressions , stylesheet انجام میشود

```
<styles>
  <style type="text/css">
    buildings {
      altitude-clamping: terrain;
      extrusion-height: 15;
      extrusion-flatten: true;
      fill:              #ff7f2f;
    }
  </style>
</styles>
```

```

<feature_model name="test">
  <features driver="ogr">
    <geometry>POLYGON( (0 0, 1 0, 1 1, 0 1) )</geometry>
    <profile>global-geodetic</profile>
  </features>
  <styles>
    <style type="text/css">
      default {
        fill:          #ff7f009f;
        stroke:         #ffffff;
        stroke-width:   2.0;
        altitude-clamping: terrain;
        altitude-technique: drape;
        render-lighting: false;
      }
    </style>
  </styles>
</feature_model>

```

### ۳-۵-۲ ویژگی های زمین

یک سری از ویژگی ها با زمین نقشه ارتباط دارند از جمله مواردی که میتوان درباره زمین نام برد:

- خیابان ها که از خطوط زمین پیروی میکنند
- درختان زمین
- و نقشه برداری موضوعی مانند رنگ آمیزی منطقه یک کشور بر اساس جمعیت آن

این ابزار انواع مختلفی از زمین را ارائه می دهد.

### بسته نقشه map clamping

این روش ساده تر از بقیه روش ها می باشد هنگام تهیه ویژگی های صفحه نقشه لایه های ارتفاع را در نقشه نمونه می کند

این روش سبب بهبود کیفیت می شود :

- بسته به وضوح انتخاب شده ، می تواند نمونه ای از داده های ارتفاع در نقشه باشد. برای تعداد زیادی از ویژگی ها ، می تواند CPU فشرده و وقت گیر باشد.
- نمونه برداری دقیق است و برای هر نقطه از هندسه انجام می شود. برای بهبود سرعت تدوین می توانید در سانسروئید هر ویژگی نمونه برداری کنید. بسته به وضوح هندسه ویژگی ، ممکن است برای دستیابی به کیفیت بهتر نیاز به کاهش داده های خود داشته باشید.
- کیفیت رندر در مقایسه با سایر روشها خوب است.

با دستور زیر فعال می شود

```

altitude-clamping: terrain; // terrain-following on
altitude-technique: map;    // clamp features to the map data
altitude-resolution: 0.005; // [optional] resolution of map data to clamp to

```

## روش Draping

این روش زمانی استفاده می شود که زمین های ناهمواری در نقشه وجود داشته باشد و osgearth این بافت را به عنوان یک ویژگی را به یک بافت RTT تبدیل می کند و سپس آن را روی زمین قرار می دهد.

این روش باعث بهبود عملکرد و کیفیت می شود.

- ویژگی هایی را کاملاً با زمین مطابقت خواهد داد. هیچ نگرانی در مورد وضوح وجود ندارد.
- ممکن است هنگام ارائه خطوط یا لبه های چند ضلعی ، آثار باستانی را بدست آورید. بافت پیش بینی شده از نظر اندازه محدود است و هرچه بیشتر منطقه ای که باید آن را بپوشاند ، وضوح تصویر پیش بینی شده کمتر است. این بدان معنی است که در عمل کشیدن برای چند ضلعی ها برای خطوط مفیدتر است.
- مصنوعات مخلوط غیر منتظره ممکن است ناشی از کشیدن بسیاری از هندسه های شفاف در بالای یکدیگر باشد.

فعال کردن این روش

```
altitude-clamping: terrain;          // terrain-following on
altitude-technique: drape;           // drape features with a projective texture
```

## روش gpu clamping

این روش از یک تکنیک دو فاز استفاده می کند: ابتدا از نمونه برداری عمق استفاده می کند تا هر راس را به پوست زمین در یک سطح راس ببندد. ثانياً ، این یک الگوریتم عمیق در سایه قطعه قطعه برای کاهش z-fighting اعمال می شود

- این برای خطوط مناسب است ، ا
- سریع است ، کاملاً در زمان اجرا اتفاق می افتد و از پردازش gpu یز استفاده میکند
- هیچ اثر لبه ای وجود ندارد

## ۴-۵-۲ داده های بزرگ

زمانی که یک سری ویژگی برای نقشه بیش اندازه بزرگ باشند سبب می شود منابع عملکردشان ضعیف شود. راه حل برای آن کاشی کاری و صفحه بندی است که روش های مختلفی دارد که با هم مورد بررسی قرار می دهیم

```
<feature_model name="shapes">
  <features name="data" driver="ogr">
    <url>data.shp</url>
  </features>
  <styles>
    data {
      fill: #ffff00;
    }
  </styles>
</feature_model>
```

## ویژگی لایه ای

در این روش از یک تگ layout استفاده می شود



```

<feature_model name="shapes">
  <features name="data" driver="ogr">
    <url>data.shp</url>
  </features>

  <layout>
    <tile_size>250000</tile_size>
    <level name="data" max_range="100000"/>
  </layout>

  <styles>
    data {
      fill: #ffff00;
    }
  </styles>
</feature_model>

```

می توان با استفاده از **cropping features** مقدار مورد نیاز از کاشی را برداشت و ویژگی را به آن نسبت داد.

```

<feature_model name="roads">
  <features name="roads" driver="ogr" build_spatial_index="true">
    <url>roads.shp</url>
  </features>

  <layout crop_features="true" tile_size="1000">
    <level max_range="5000"/>
  </layout>

  <styles>
    <style type="text/css">
      roads {
        stroke: #ffff7f7f;
      }
    </style>
  </styles>
</feature_model>

```

و با استفاده از **level** میشود سطح های مختلف را مشخص نمود

۳- مراجع

<https://docs.osgearth.org/en/2.10/about.html>