Height Aware 3D Application Documentation

Submitted by:

Arora Engineers, Inc.

Updated: 9/12/2018

Contents

FIGURE 1 THE DEFAULT APPLICATION VIEW AFTER INITIAL LOADING

Background

The PHL HeightAware 3D application is an update to the original Height Aware application which was rendered within the application in 2D. This revised application provides much of the same functionality for analyzing proposed obstacles with several enhancements and leverages the esri 4.8 api to render the 3d view. The goal of this application is to provide a streamlined and easy-to-use application for analyzing potential obstructions in 3D. A major feature enhancement of this application is that the extent of the underlying elevation model has been greatly increased to enclose all the surfaces in the application.   (*In figure 2, the original 1ft dem is placed at the center of the larger USGS dem and symbolized with its own renderer based on the elevation range within its data to make the roads and bridges visible*)

Elevation Sources

The original dem was built using 1ft contours generated in 2013. In the original application, estimating obstruction heights outside of this dem extent required that users enter the elevation manually before submitting the request. By leveraging the USGS 10m dems and resampling them to 1ft, this updated application provides estimated ground elevations for the entire airspace. The final dem in this 3D application was created from USGS 10m rasters with the time period of the content specified as 2015 and the published data of 3/29/2017.

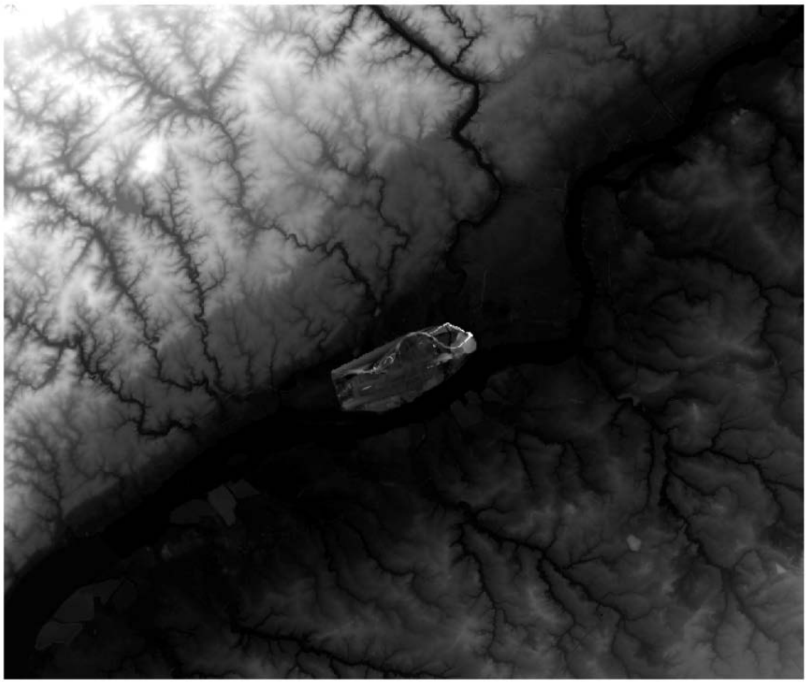


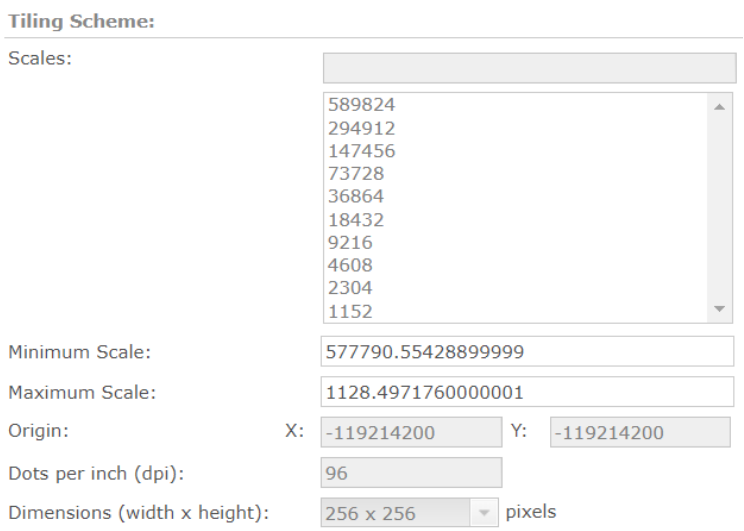
FIGURE 2 THE ORIGINAL 1FT DEM AT THE CENTER OF THE USGS 10 METER DEMS RESAMPLED TO 1FT

FIGURE 3 THE ORIGINAL 1 FT. DEM MERGED WITH THE FINAL USGS RESAMPLED 1FT DEM

In Figure 3, the original 1ft dem has been merged with the USGS dems to create the final dem used in the application. The elevations of the bridges in the original dem are visible as light colored areas to the east of the airport. The USGS dems do not include any bridges or manmade structures.

Elevation Image Service

The ArcGIS Server is utilized to create the tile cache at these scales in the State Plane projection (wkid: 2272) native to the Master Geodatabase:



Tiling Scheme:

Scales:	589824 294912 147456 73728 36864 18432 9216 4608 2304 1152
Minimum Scale:	577790.55428899999
Maximum Scale:	1128.4971760000001
Origin:	X: -119214200 Y: -119214200
Dots per inch (dpi):	96
Dimensions (width x height):	256 x 256 pixels

The Image Service url is passed into the Elevation Layer constructor in the esri jsapi. The Elevation Layer is a tile layer used for rendering elevations in SceneViews. The Elevation Layer is set as the only elevation layer in the array of the elevation layers comprising the ground of the WebScene when the scene is created.

Included Map/Feature Services

To provide the airspace GIS data and querying capability for this obstruction analysis application, there is one map service containing all of the airspace surface polygons and their derived (5' x 5') rasters. Due to the large number of layers in that service, additional GIS data is served through the 'ContextFeatures' FeatureServer.

1. **Surfaces** (MapServer) – Used as the input to the GIS Server Identify Task which performs a server-side query and returns the data that populates the 'Results of the Obstruction Analysis' panel. Both 2D and 3D surfaces are provided. Refer to the diagram below for the schema of the map service.
2. **Surfaces** (FeatureServer) – Used as the input to the Feature Layers with z-values generated in the app. Both 2D and 3D surfaces are provided. Refer to the diagram below for the individual layers on the FeatureServer and their indexes. They symbology of the published surfaces is used in the app rather than creating new renderers for each surface type.
3. **ContextFeatures** (FeatureServer) – Used as the input to the Feature Layers containing structures or other non-airspace surfaces. Tree is a subset of the Obstruction Points created by applying a definition query on the Tree layer in the mxd when it was published. The ABOVEGROUN attribute is used by the Feature Layer renderer to generate 3D trees in the app. Building contains structures with valid structure heights. The colorful symbology for the buildings is controlled by the Feature Layers in the app rather than the map service.

The **Surfaces** (MapService) Schema:

- Part77 3D Surfaces (0)
    - Transitional (1)
    - Approach (2)
    - Horizontal (3)
    - Conical (4)
- Critical 3D Surfaces (5)
    - TERPS (6)
    - DEPARTURE (7)
    - OEI (8)
- Critical 2D Surfaces (9)
    - RunwayHelipadDesignSurface (10)
    - AirOperationsArea (11)
- 3D Surface Rasters (12)
    - 9L / 27R (13)
        - SI34VISAREA_9L_27R_51 (14)
        - SI34VISAREA_9L_27R_77 (15)
        - SI20VISAREA_9L_27R_48 (16)

        - PIRTS77_9R_27L_92 (65)
        - PIRTS77_9R_27L_9 (66)
        - PIRTS77_9R_27L_83 (67)
        - PIRTS77_9R_27L_32 (68)
        - PIRTS77_9R_27L_13 (69)
    - 8 / 26 (70)
        - PIRAS77_8_26_84 (71)
        - PIRPS77_8_26_86 (72)
        - PIRTS77_8_26_42 (73)
        - PIRTS77_8_26_79 (74)
        - VISAS77_8_26_68 (75)
        - TRNS77_8_26_3 (76)
        - TRNS77_8_26_28 (77)
    - 17 / 35 (78)
        - TRNS77_17_35_4 (79)
        - TRNS77_17_35_74 (80)
        - PIRTS77_17_35_94 (81)
        - PIRTS77_17_35_26 (82)
        - PIRPS77_17_35_80 (83)
        - NPIRAS77_17_35_45 (84)
        - PIRAS77_17_35_11 (85)
- RasterGround (86)

The **Surfaces** (FeatureServer) Schema:

- Transitional (1)
- Approach (2)
- Horizontal (3)
- Conical (4)
- TERPS (6)
- DEPARTURE (7)
- OEI (8)
- RunwayHelipadDesignSurface (10)
- AirOperationsArea (11)

The index associated with each layer on the FeatureServer is derived from the corresponding layer index on the Surface (MapServer) map service.

Included Context Services

To provide visual guides for interacting with the data, buildings and trees have been added to the Web Scene. The building polygons are rendered on-the-ground and extruded to a height (feet) specified in the STRUCTHGHT field. The color symbology chosen for the buildings is not controlled by the map service the buildings are published in. Instead, it is created in the app code using the feature layer renderer required to extrude the buildings. Each Terminal Building is shown with a unique color fill.

Included Aerial Imagery Service

The Imagery sources for this application include 2015 NAIP Imagery at the full extent. PHL Aerial Imagery exists around the airport property from 2016. The Imagery has been resampled to 1ft, mosaiced, balanced, and split into several equal sized .tif files that are stored in a folder registered with the GIS Server as a data store. The rasters made available through a Map Service with the tile cache built and stored on the GIS Server.

App functionality

In the upper left corner of the application, there is a menu button which when clicked opens a popup-menu containing the widgets that provide functionality in the app.

- **Runway Selector** – Provides the user with a way to filter which surfaces are contained in the analysis by runway end.
- **Obstruction Analysis** – Opens the Obstruction Analysis Panel allowing users to propose obstacles and view the potential impact with airspace surfaces.
- **Legend** – Provides a small graphic for each feature type along with the layer title to assist with comprehending the surfaces displayed in the app
- **Position** – The Panel for this widget contains information about the position of the user's view in relationship to the ground. Heading, Tilt, Eastings, Northings, and Camera Height are updated in this panel as the user navigates the app. The Position is recorded in feet according to the State Plane Projection.
- **File Loader** – (not functioning) Provides the user with a way to analyze multiple obstacles by uploading them as a csv file with eastings, northings, and proposed height.
- **Measure 3D** – Opens the widget panel and displays and Active Measure button that, when clicked, will enable measuring distances in both horizontal and vertical directions. This tool will display measurements that follow the 3d features as well as the ground surface. The measured values for these multiple dimensions are all displayed and updated as the user moves the mouse location.
- **User Guide** – Opens a new tab and loads the user guide pdf document

Data Analysis Output

When the app loads, the Obstruction Analysis Panel (OAP) is open in the upper right. Clicking 'Activate' will enable a graphic layer that is tracking the mouse location on the map. As the user moves the graphic around the map, the location and ground elevation will update in the OAP. Clicking on the map will place an obstruction with the height specified in the OAP. If the height value is not specified at the time an obstruction is placed, then a value equal to 200 ft. minus the ground elevation is used as the peak height for the obstruction. Modifying the values for the height of the obstruction and the ground elevation then clicking Submit will update the obstacle in the map and regenerate the Obstruction Results. The ground elevation, x-coord, and y-coord are queried from the map as the mouse location changes. Using these values along with the proposed obstruction height, a 3d graphic is added to the map representing the obstacle where the user clicked.

Within the typescript/javascript code in the application, the intersection of the proposed obstacle and each penetrated surface is determined by leveraging the glMatrix library, specifically the mat4 module which contains methods for working with 4x4 matrixes. The intersection point output is used to create features in the 'Surface Intersection Point' feature layer which is added to the scene but made not visible by default.

These same obstruction settings are also submitted to the GIS Server through a query passed to the IdentifyTask, and the results populate the 'Results of Obstruction Analysis' table that appears in the lower right once the operation is completed. All surfaces that intersect the x-y location of the obstacle are returned in the results table, and any surfaces that are penetrated by the obstacle are given a bright red color in the Clearance column signifying a negative penetration value.

The results from the Obstruction Analysis are calculated by subtracting the proposed obstacle height from the pixel values in the associated surface raster(s) located in the **Surface** map service. If the resulting value is negative, then the clearance value is negative, and the table cell is colored red.

Use Case Scenarios

Maintenance and Recovery Plan

This 3D Web Mapping Application requires ArcGIS Server to host the feature server and the image server. At the time of development only one GIS Server is needed. Adding more GIS Servers to the site would allow for a failover in the event that the GIS Server needs to be taken offline. The steps for adding a GIS Server to an existing site are available here https://enterprise.arcgis.com/en/server/latest/install/windows/join-existing-site.htm.

The code base is maintained using GitHub which allows the files to be versioned into development and production branches at a minimum. Deploying the application files onto a new web server is as simple as cloning the GitHub repo and running the build script (grunt –force) which transpiles the typescript into javascript. The /src folder containing the index.html file must be registered as an application on IIS.

A web server is required to host the application files (.js, .css, and .html). We use Windows Server's IIS in development and in the cloud "production" site due to the large percentage of projects where it is the web server. It is possible to deploy this application onto any http web server that has ports 80 and 443 opened.