Bringing Your Data to Life in the ArcGIS API for JavaScript: 3D Integrated Mesh & Point Cloud

Sean Morrish, Johannes Schmid

Point Cloud Layers in the ArcGIS API for JavaScript

- `esri/layers/PointCloudLayer`
  - Usable only in 3D at the moment

- Supported functionality
  - Display and explore
  - Data-driven coloring and other visualization options
  - Simple user interaction with click event

Renderers

- PointCloudRGBRenderer
- PointCloudStretchRenderer
- PointCloudUniqueValueRenderer
- PointCloudClassBreaksRenderer

Renderers: PointCloudRGBRenderer

- Use attribute value directly as color for point rendering
- Only useful when data set contains RGB values in attributes
- All you need to do is specify the field name:

```
layer.renderer = new PointCloudRGBRenderer({
  field: "RGB"
});
```

Renderers: PointCloudStretchRenderer

- Define a color ramp, driven by a specific attribute
- For example:
  - Color by height
  - Color by return intensity value
  - ...
- Note: ELEVATION is a special field

```
new PointCloudStretchRenderer({
  field: "ELEVATION",
  stops: [{
    value: -0.78,
    color: [61, 51, 158]
  }, {
    value: 1,
    color: [73, 196, 196]
  }, {
    value: 4,
    color: [235, 162, 84]
  }, {
    value: 12,
    color: [235, 84, 84]
  }, {
    value: 20,
    color: [100, 100, 100]
  }]
});
```

Renderers: PointCloudUniqueValueRenderer

- **Color by classification that is already present in a field**

```
new PointCloudUniqueValueRenderer({
  field: "CLASS_CODE",
  colorUniqueValueInfos: [
    {
      values: [2],
      label: "Ground",
      color: [222, 184, 135]
    }, {
      values: [3, 4, 5],
      label: "Vegetation",
      color: [200, 232, 171]
    }, {
      values: [6],
      label: "Building",
      color: [158, 40, 17]
    }, {
      values: [7, 8, 9, 10, 11, 12],
      label: "Other",
      color: [50, 50, 50]
    }
  ]
});
```

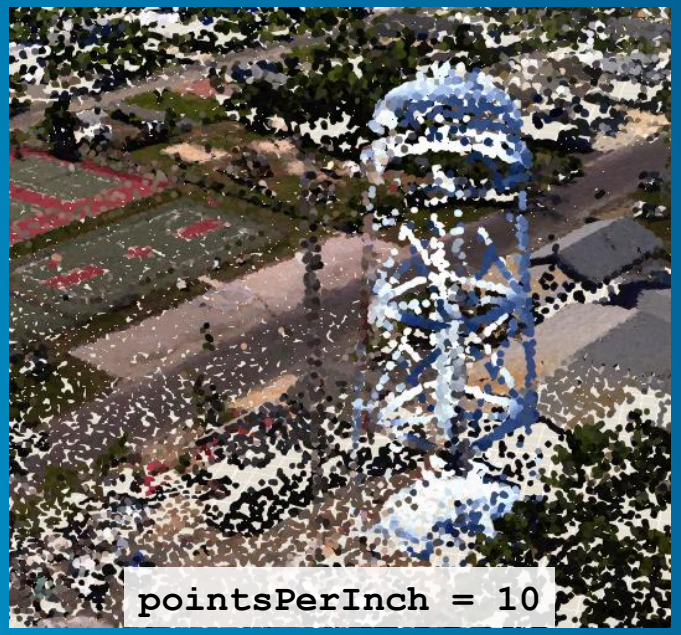
Renderers: PointCloudClassBreaksRenderer

- **Define value ranges that map to specific colors**

```
new PointCloudClassBreaksRenderer({
  field: "INTENSITY",
  colorClassBreakInfos: [
    {
      minValue: 0,
      maxValue: 30000,
      color: [255, 255, 255]
    }, {
      minValue: 30000,
      maxValue: 50000,
      color: [100, 100, 100]
    }, {
      minValue: 50000,
      maxValue: 65700,
      color: [50, 50, 50]
    }]
});
```

Renderers: Common Properties

- `pointsPerInch`
  - Lets you configure the density of points on screen
  - Never exceeds a hard cap on the total number of points



Renderers: Common Properties

- `pointSizeAlgorithm`
  - Two modes: `splat` and `fixed-size`
  - Splat automatically picks a size based on point density
    - Can be tweaked with `scaleFactor` and `minSize`
    - Default

```
new PointCloudRGBRenderer({
  field: "RGB",
  pointSizeAlgorithm: {
    type: "splat",
    scaleFactor: 3,
    minSize: 2
  }
});
```

  - Fixed size uses a constant size for all splats
    - Either in screen units (pixels) or map units (meters)

```
new PointCloudRGBRenderer({
  field: "RGB",
  pointsPerInch: 40,
  pointSizeAlgorithm: {
    type: "fixed-size",
    useRealWorldSymbolSizes: false,
    size: 1
  }
});
```

User Interaction

- `view.hitTest()` returns point location

```
view.on("click", function(screenPoint)
{
  view.hitTest(screenPoint)
    .then(function(hit) {
      hit.results.forEach(result => {
        console.log(result.mapPoint);
        console.log(result.graphic);
      });
    });
});
```

  - `result.mapPoint`: location on “disk” under the cursor
  - `result.graphic.geometry`: actual point location (center)


esri
THE
SCIENCE
OF
WHERE