AUTODESK®
FORGE DevCon

Large scale point cloud visualization in Forge Viewer with Airsquire

Yue You
Airsquire CTO

YouYue123
youyue@airsquare.ai

Michael Beale
Developer Advocate

@micbeale
michael.beale@autodesk.com

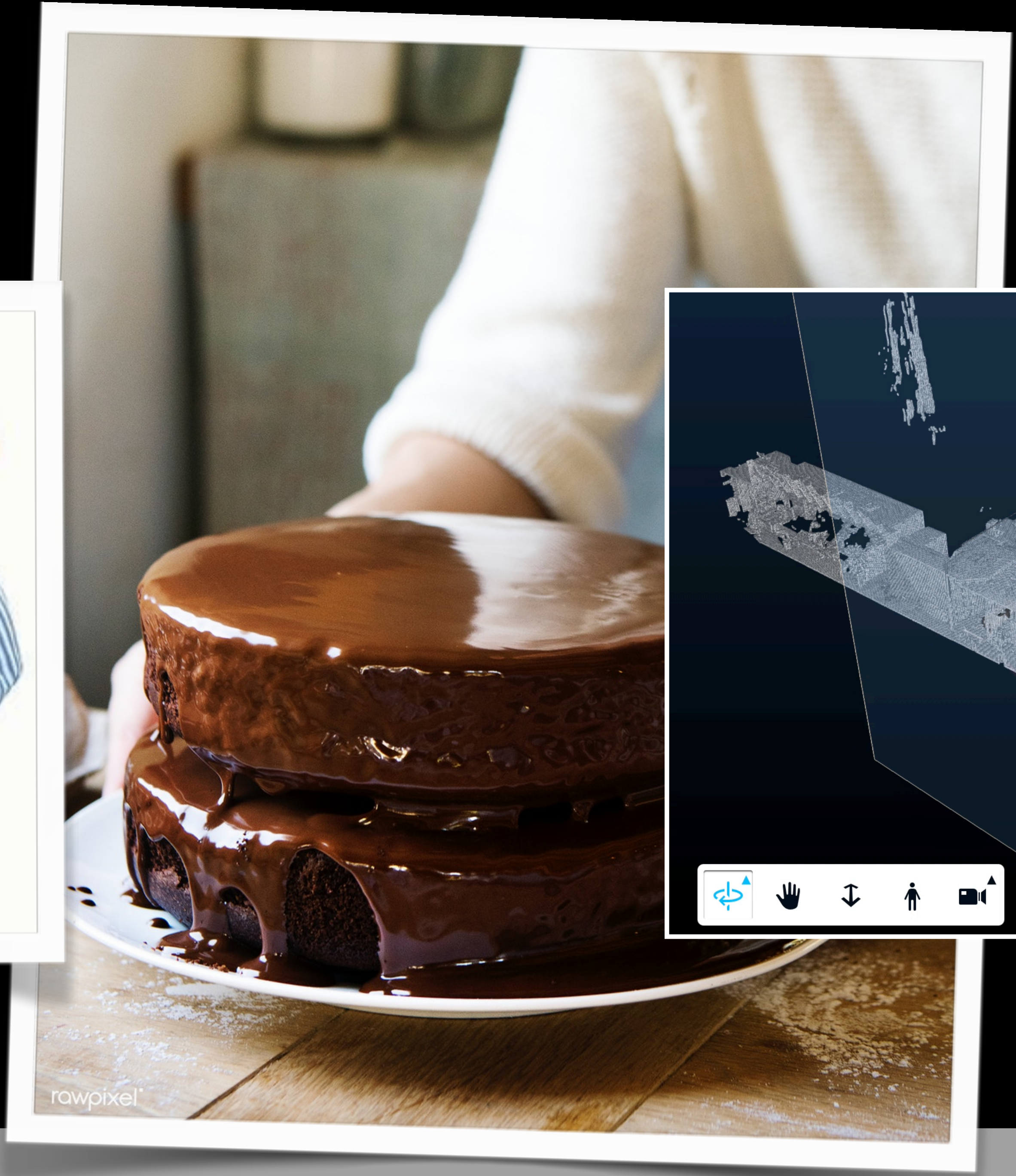
AIRSQUIRE
AUTODESK® FORGE

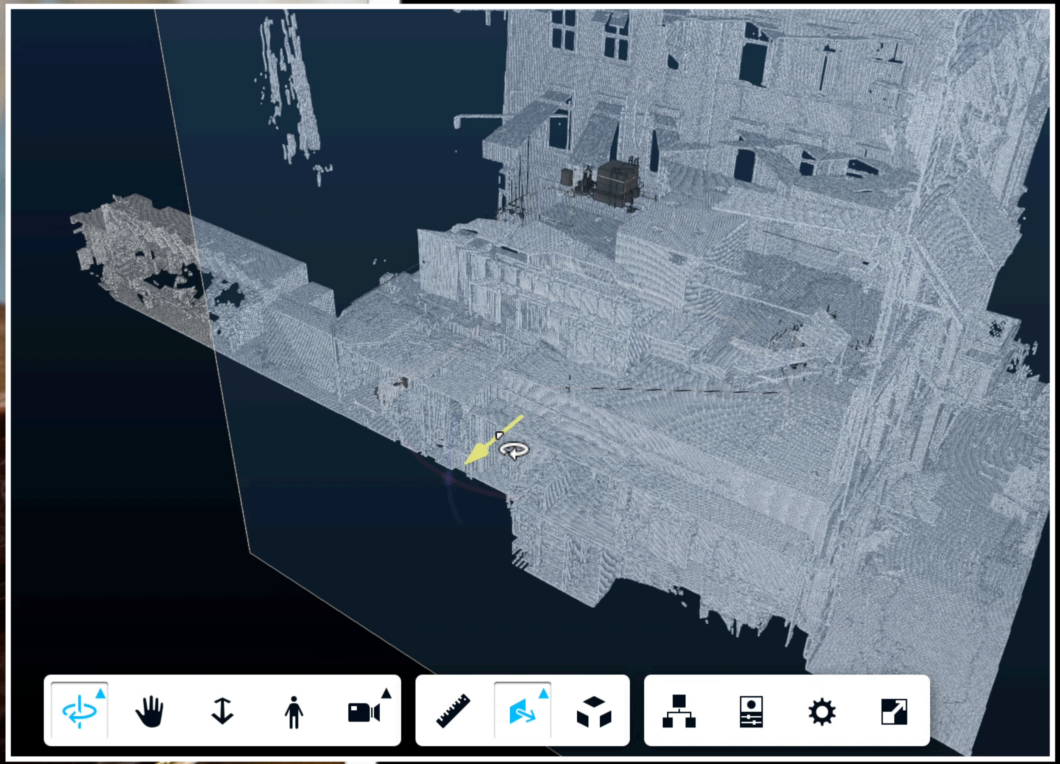
Class Summary

- Industry Trends
- AirSquire Demo
- **Code :>** Revit+Point Cloud Scan
- Advanced Topics: streaming / sectioning / measuring
- Q & A

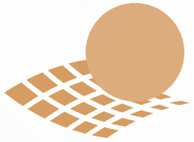
AIRSQUIRE
AUTODESK® FORGE

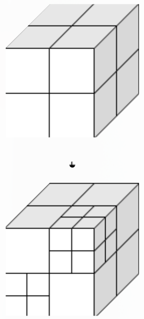
Code :>





Terms



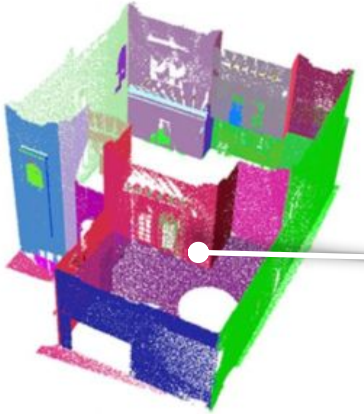


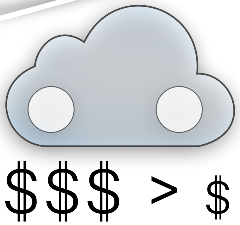
- Point Cloud
- Streaming
- Octree
- LiDAR - Light Detection and Ranging

Questions you may have…

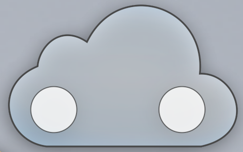
- How do I Access point cloud data?
- How do I Organize point cloud data?
- Why is a web service for point clouds needed?
- What is the expected data volume of point cloud data?
- Do existing technologies already address the challenge?

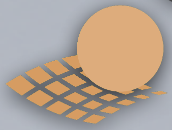
Trends in the Point Cloud industry





- Downstream analysis the “norm” - Segmentation, Classification, Summary attributes
- Compute / Scanner costs dropping, making “measurement activity" more viable/practical
- Scanner density and data volume increasing
- Increasing demand for management & access tools of massive collections
  - Move to Point Cloud Web Services
  - Access Points via Streaming





How it all started…

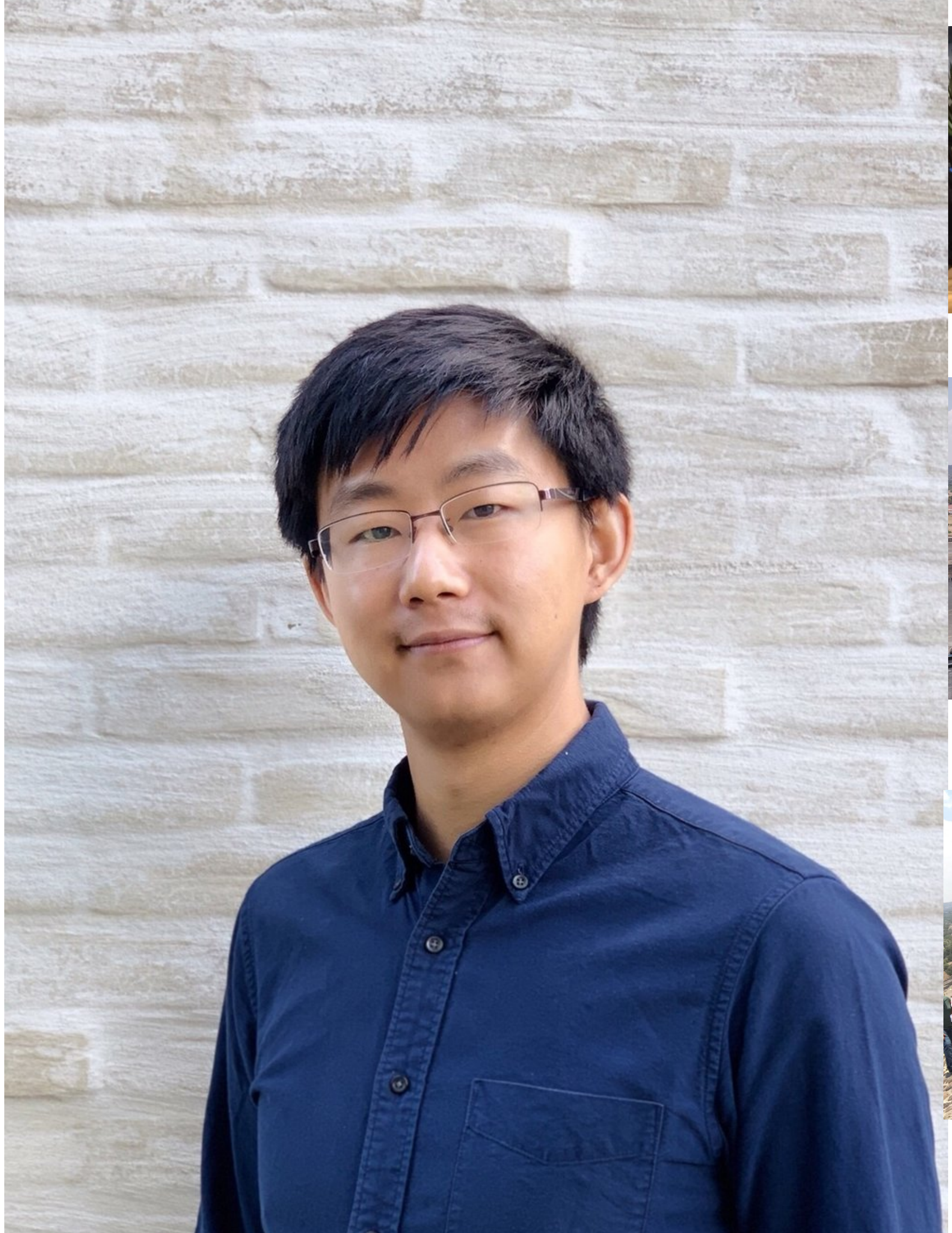
Forge Accelerator’s

http://forge.autodesk.com/accelerator

You Yue

Airsquire CTO

youyue@airsquare.ai



AIRSQUIRE

AUTODESK® FORGE

Who is Airsquire?

Airsquire as-built Visualisation and Verification. Process, interrogate and verify point clouds in hours, not days.

Our proprietary A.I. algorithm compares BIM model against Point Cloud scan to automatically detect progress and deviation.

Use cases: existing verification, progress verification, as-built verification

Current customers includes UK and BeNeLux top Constructors:

MWH TREATMENT

WILLMOTT DIXON SINCE 1852

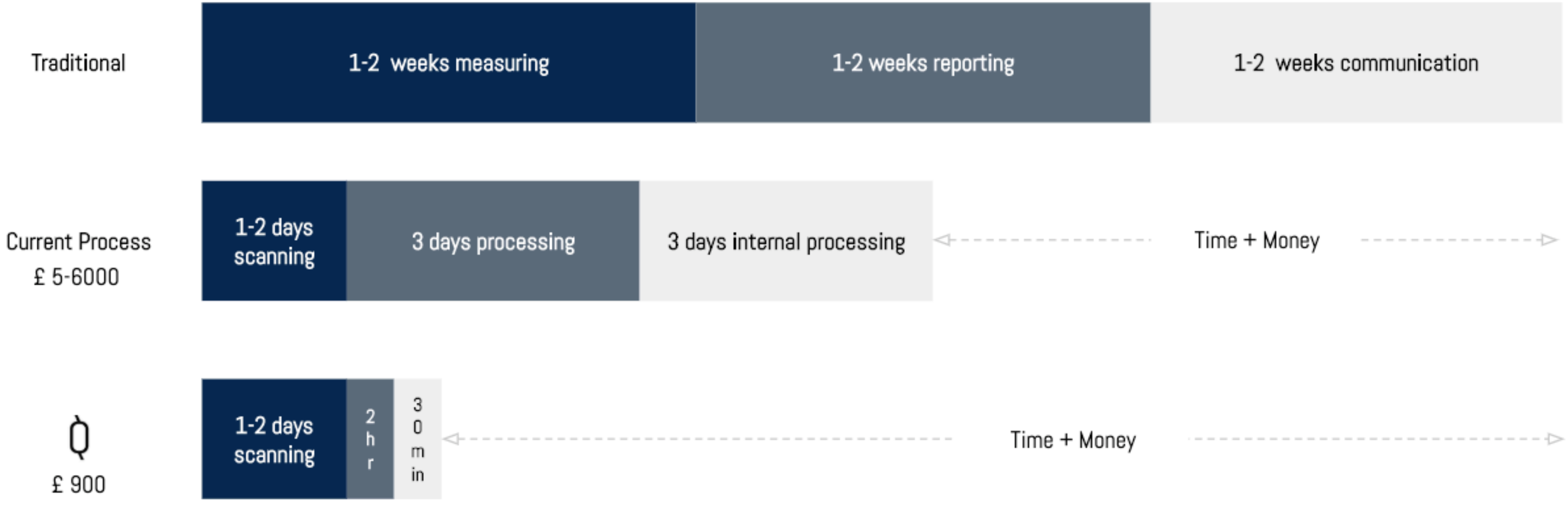
bam

msa msasurvey.com

VolkerWessels

BOUSTEAD PROJECTS

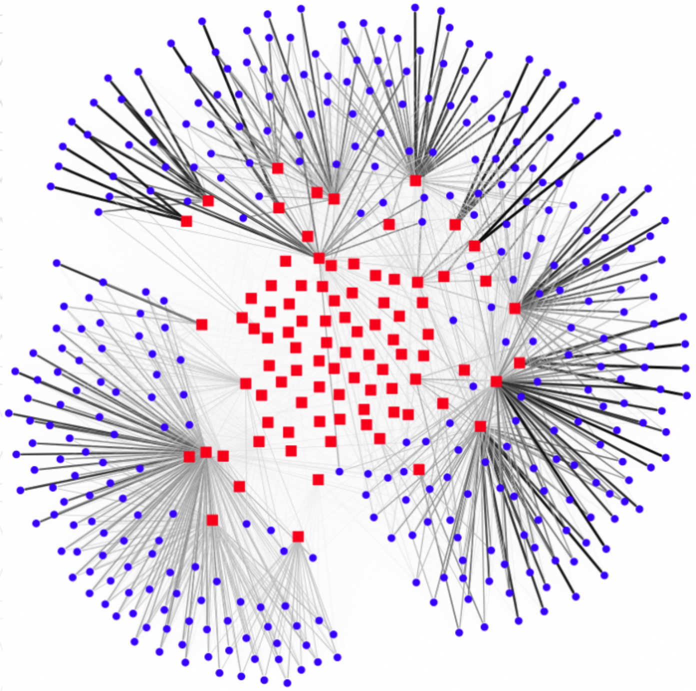
Benefit



AIRSQUIRE
AUTODESK® FORGE

Why large scale **model + points** in the cloud, matters

- In construction management, **Reality** is complex
- Traditional communication is not accurate
- Contextual model and point cloud, **keeps everyone, on the same page**



AIRSQUIRE

AUTODESK® FORGE

AIRSYNC construction verification with AI assistance

Simplest workflow

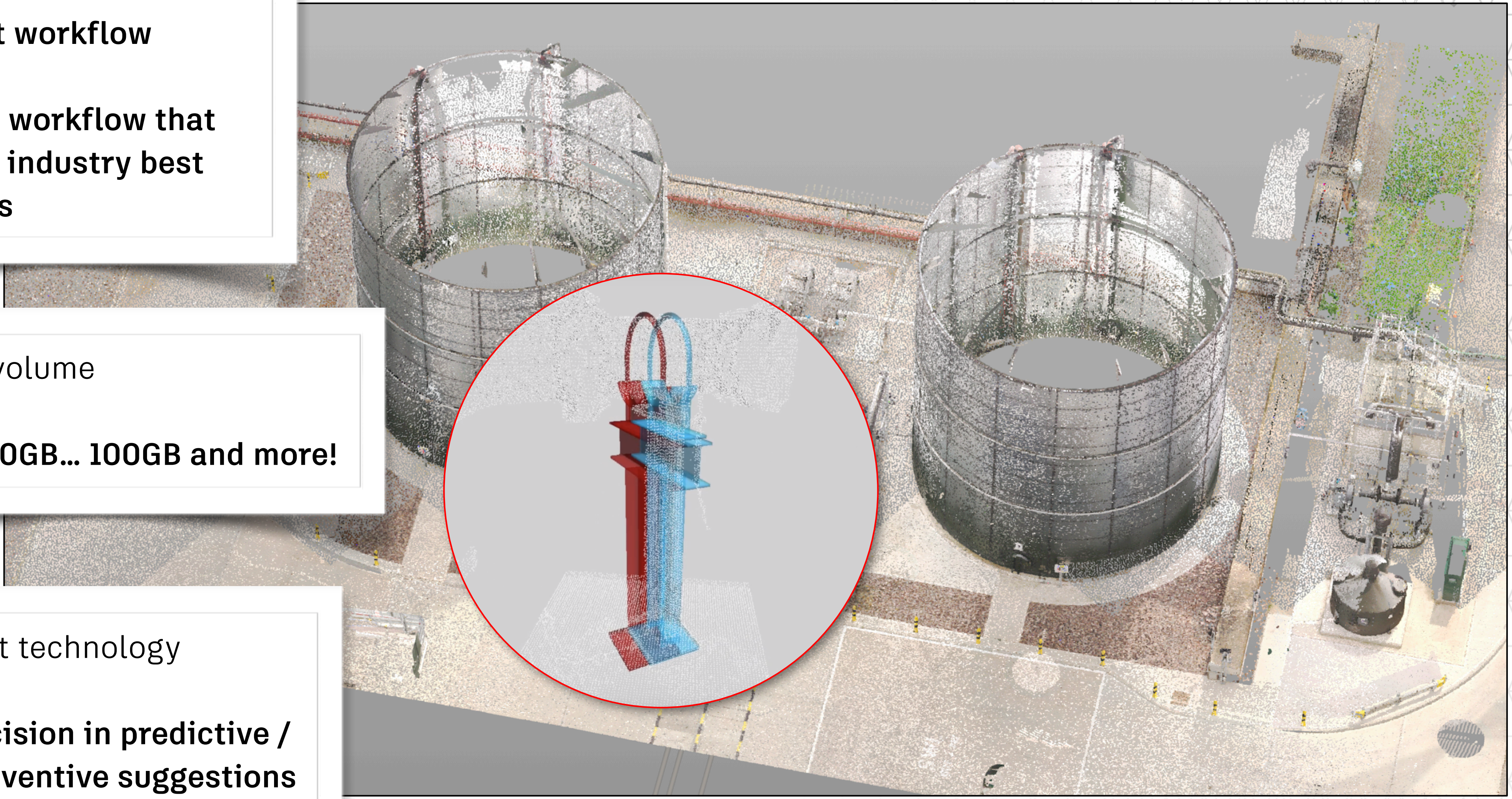
Intuitive workflow that fits with industry best practices

Big data volume

Not just 10GB… 100GB and more!

Smartest technology

mm precision in predictive / cost-preventive suggestions

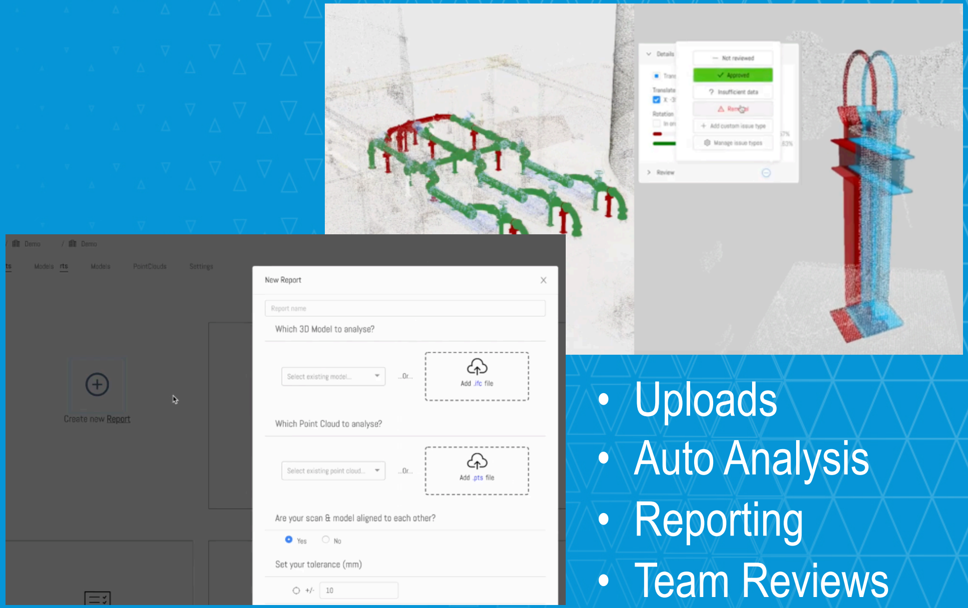


AIRSQUIRE

AUTODESK® FORGE

AIRSQUIRE

Demo



- Uploads
- Auto Analysis
- Reporting
- Team Reviews

https://www.youtube.com/watch?v=wGxr-DLHHxM

AIRSQUIRE

AUTODESK® FORGE

Now, the hard part…

Potree Vs Custom Built

AIRSQUIRE

AUTODESK® FORGE

Potree

- ForgeViewer still THREE.js R71
- Find old R87 Three + Potree
- Must port code from R87 to R71
- Compatibility with Forge WebglRenderer
- Add latest ForgeViewer

Custom Built

- Use point cloud class
- Add simple loading

```
geometry.isPoints = true;
```

- Use Recap.dll for RCP decode
- Use octree index from .RCP
- Connect camera FOV to Octree
- stream in/out

Let’s start with a basic point-cloud…

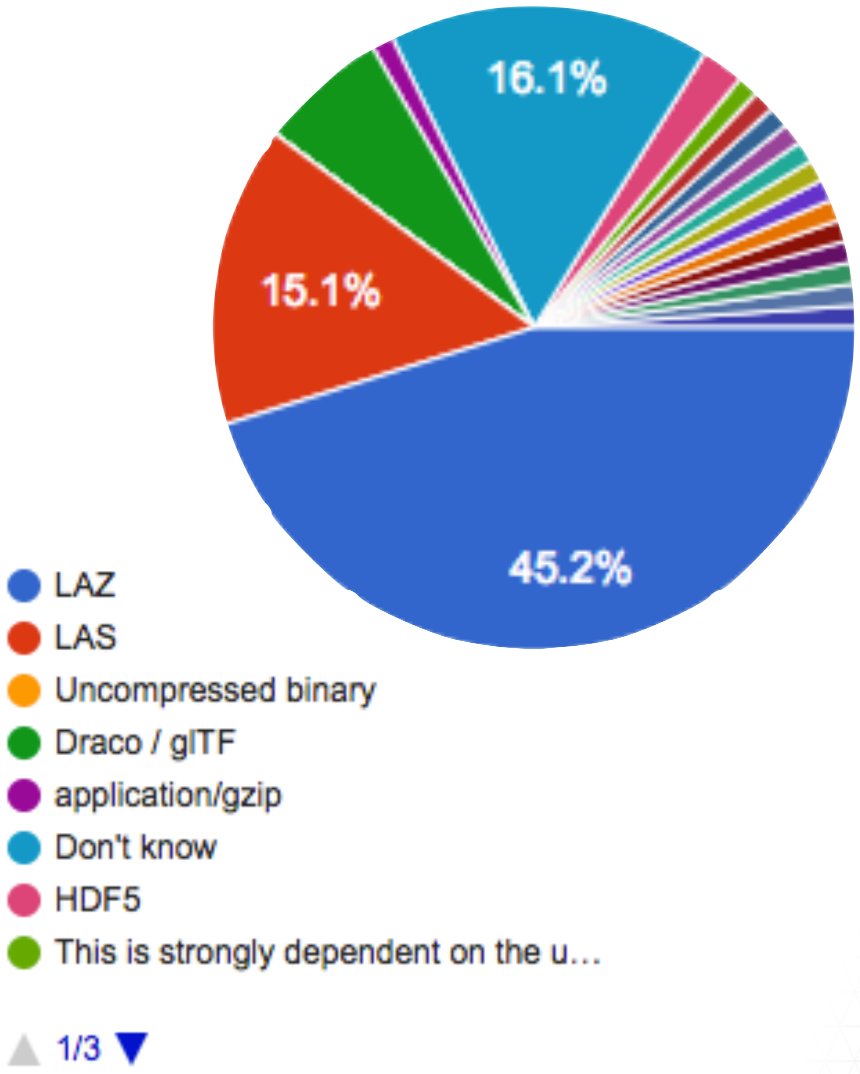
This flag will force Forge Viewer to render the geometry as gl.POINTS

```
const geometry = new THREE.BufferGeometry();
   const numPoints = width * length;
   const positions = new Float32Array(numPoints * 3);
    const colors = new Float32Array(numPoints * 3);
   geometry.addAttribute('position', new THREE.BufferAttribute(positions, 3));
   geometry.addAttribute('color', new THREE.BufferAttribute(colors, 3));
   geometry.computeBoundingBox();

   geometry.isPoints = true;

const material = new THREE.PointCloudMaterial({ size: PointSize,
    vertexColors: THREE.VertexColors })
const pointcloud = new THREE.PointCloud(geometry, material);
this.forgeViewer.impl.createOverlayScene('pointclouds');
this.forgeViewer.impl.addOverlay('pointclouds', this.points)
```

AIRSQUIRE

AUTODESK® FORGE

Let's add Draco compression...

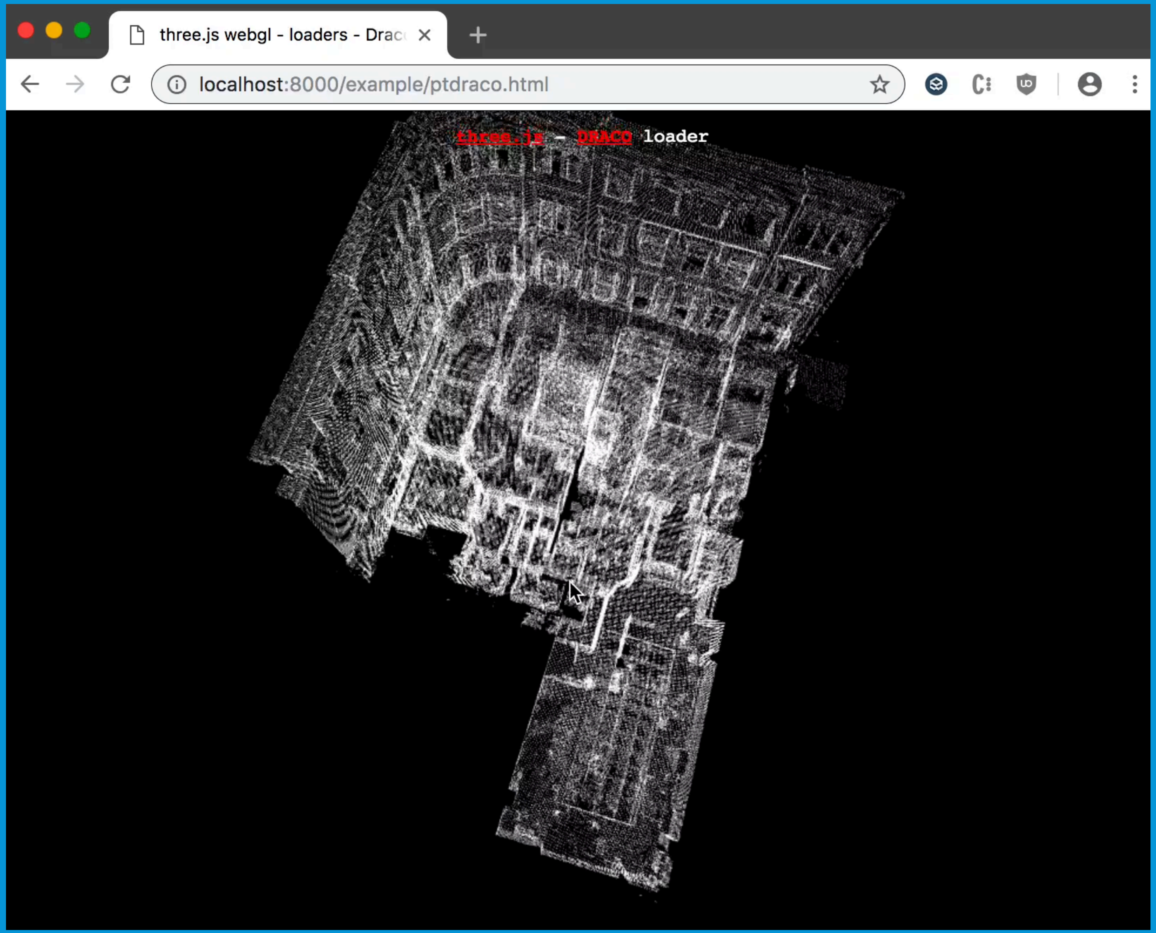


http://docs.opengeospatial.org/per/18-048r1.html

- Highly Compressed
  - Open Source
  - Web Friendly
  - KD Tree (NN-Search)
- Tooling
  - Google (draco-encode)
  - Cesium (PC Tiler)
  - glTF
  - RCP > PLY > .DRC

AIRSQUIRE
AUTODESK® FORGE

Demo - putting it together … a sample scan with Three.js



Sample Scan

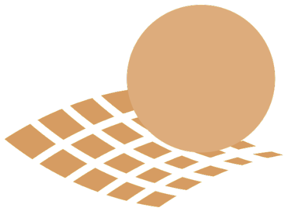
- 2 Million Points
- PLY file: 80MB
- LAZ file: 8MB
- **DRC file: 4MB**

https://github.com/wallabyway/forge-point-clouds

AIRSQUIRE
AUTODESK® FORGE

Let's Code !

But 'Real' point-clouds are big …. Really big !



Examples

- **Floor - 6 scans**
- Factory - 60 scans
- Oil/Gas - 600 scans

So, for one Floor

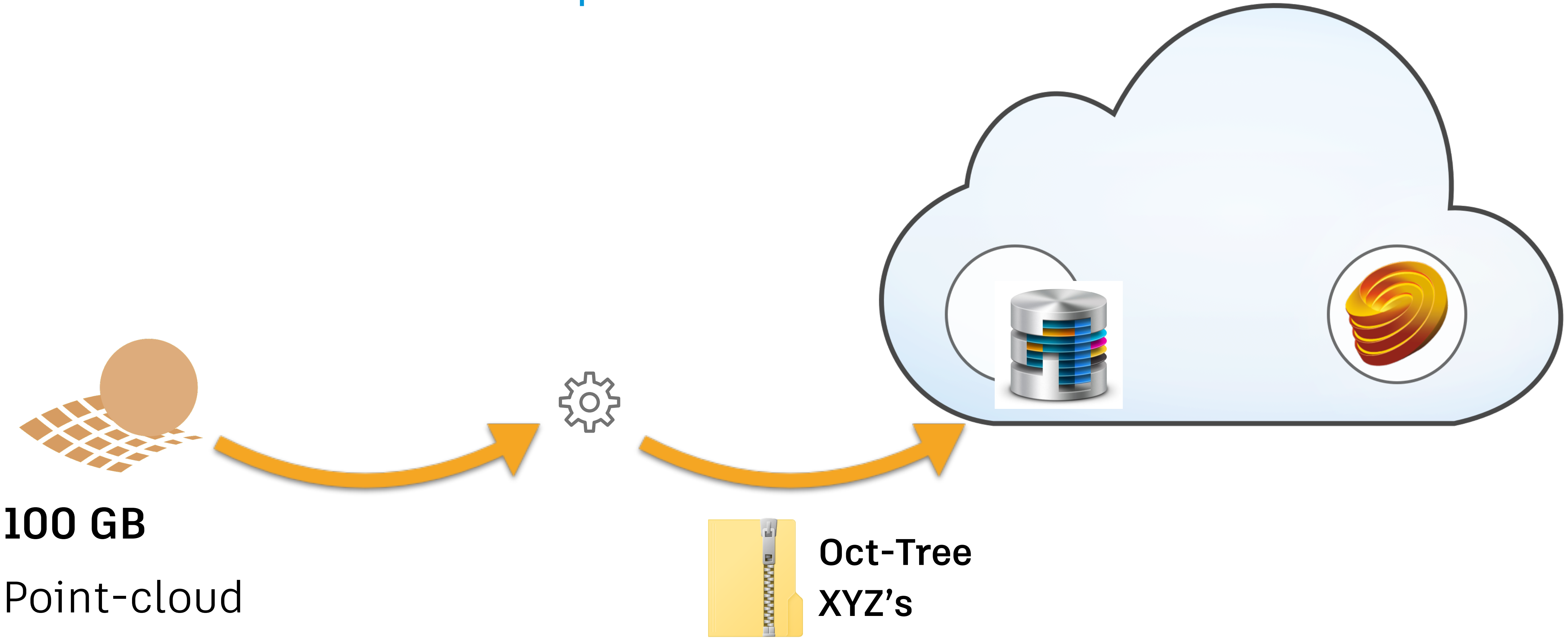
- 100,000 elements
- 100 GB point cloud

AIRSQUIRE

AUTODESK® FORGE

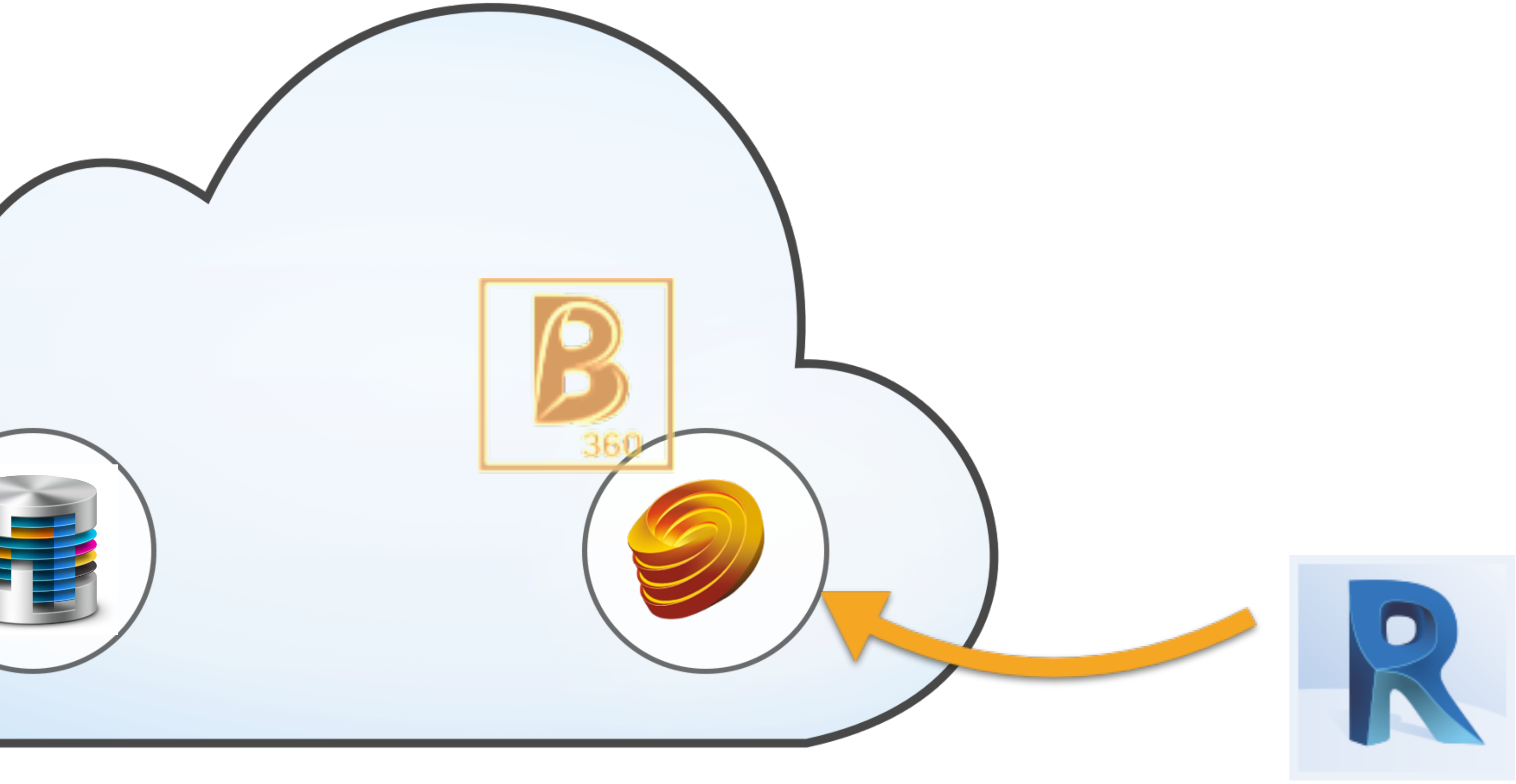

Streaming + Spatial Index
Oct-Tree
XYZ's
1 TB
AIRSQUIRE
AUTODESK® FORGE

Pre-Process Scan's on upload

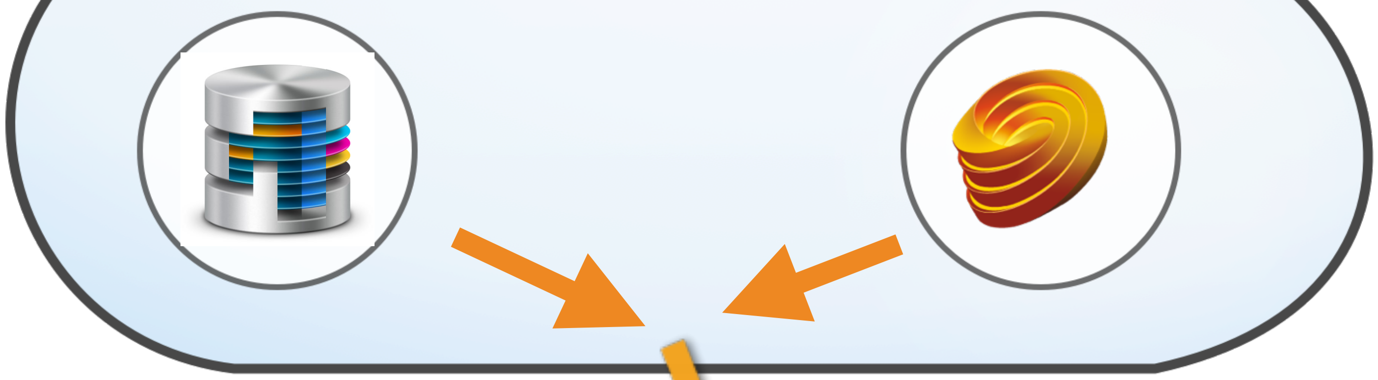


AIRSQUIRE
AUTODESK® FORGE

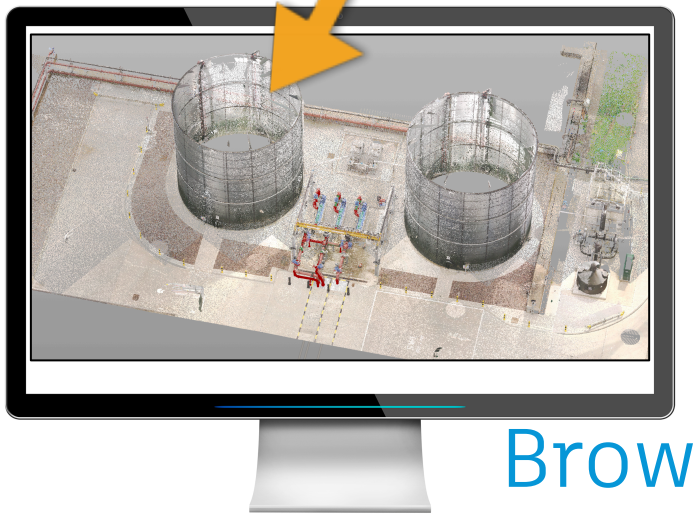
Upload Model to Cloud



Revit / Navis Models



Model + PointCloud:
combined inside ForgeViewer



Browser

AIRSQUIRE
AUTODESK® FORGE

AIRSQUIRE

Advanced Topics

- Spatial indexing
- Level of detail(LOD)
- Shading
- Sectioning
- Measuring

AIRSQUIRE
AUTODESK® FORGE

Scalability of point cloud is insane

- A **Flat** point cloud **data structure** is not scalable
- Render performance drops with the amount of points
- ie. 10 times more data → response 2-3 times slower in rendering
- **100 +GB data is very normal**

**Data is from Netherland escience center 2014 presentation*

Spatial data organization for scalability

Problem:

- How can the “flat data structure” be organized more efficiently

Challenge:

- No assumptions on “how the data is organized”

Answer:

- Spatial clustering (e.g.**octree)**
- as obtained by **Morton code**

**Data is from Netherland escience center 2014 presentation*

Octree

An octree is an object that represents a spacial partitioning. It is made up by a tree data structure in which each node has exactly eight children. Occupied leaf nodes are represented as voxels. Octrees can be used to offer a simplified representation for shapes or point clouds, or can act as an occupancy grid/space:

Octrees are collidable, measurable and detectable objects. This means that octrees:

- can be used in collision detections with other collidable objects.
- can be used in minimum distance calculations with other measurable objects.
- can be detected by proximity sensors.

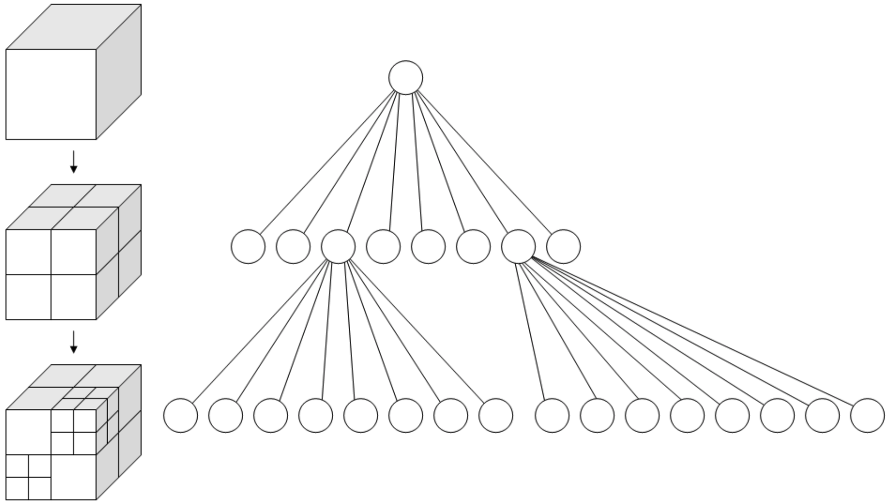


*Figure is from Wikipedia

Morton Code(Z-order code)

Morton encoding is a mapping from a multi-dimensional space to one dimension [32]. When generating a Morton code, first, a bit code is constructed for every node. This node is then converted to an integer, if needed. The nodes, once laid out, follow a Z-order curve, which enhances data-locality.

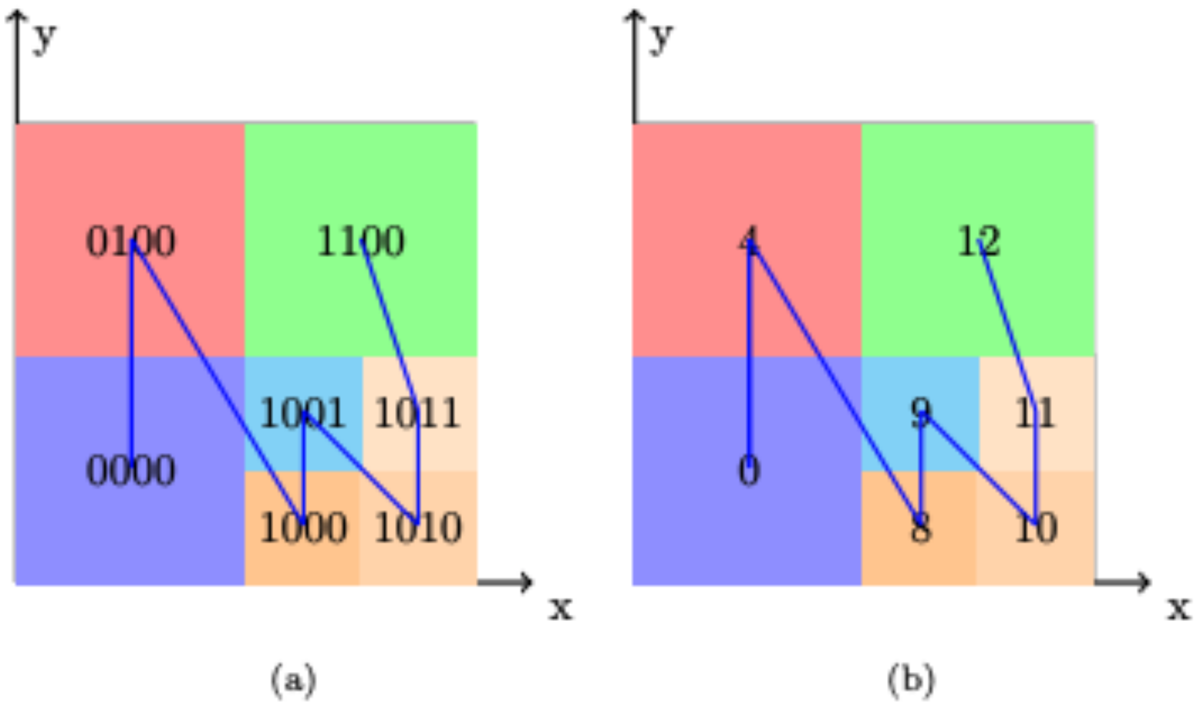


*Figure is Jaber J. Hasbestan Inanc Senocak's paper (Binarized octree generation for Cartesian adaptive mesh refinement around immersed geometries)

3D Morton code to construct Octree

1	1	0	...

Level N-1 of octree

0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	...

Level N of octree

Morton – Encoded Array

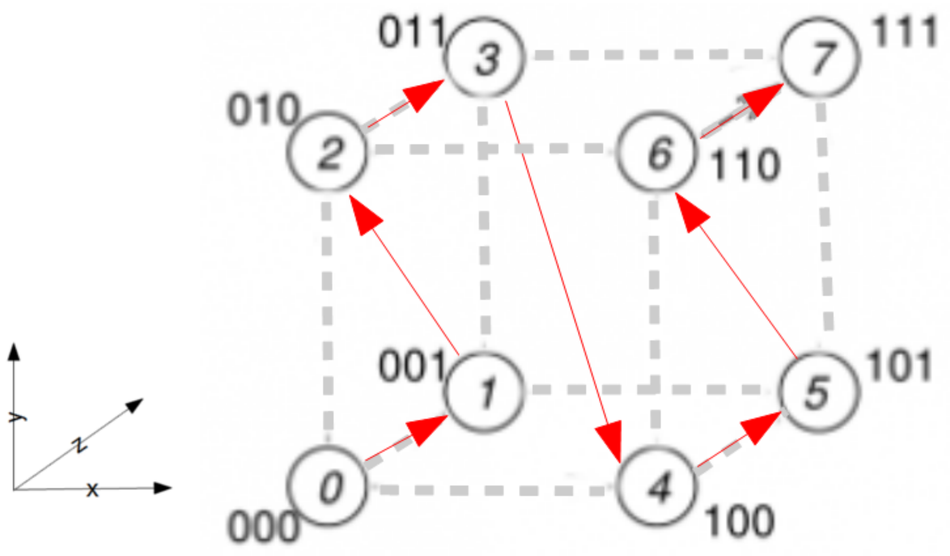


*Figure is from [Jeroen Baert's Blog] Morton encoding/decoding through bit interleaving: Implementations

3D Morton code to construct Octree

Get 3D Morton code based on position

```
uint64_t getSpaceIndex(int x, int y, int z){
    uint64_t answer = 0;
    for (uint64_t i = 0; i < (sizeof(uint64_t)* CHAR_BIT)/3; ++i) {
        index |= ((x & ((uint64_t)1 << i)) << 2*i) |
     ((y & ((uint64_t)1 << i)) << (2*i + 1)) |
     ((z & ((uint64_t)1 << i)) << (2*i + 2));
    }
    return index;
}
```

AIRSQUIRE
AUTODESK® FORGE

LOD - “Level of Detail”

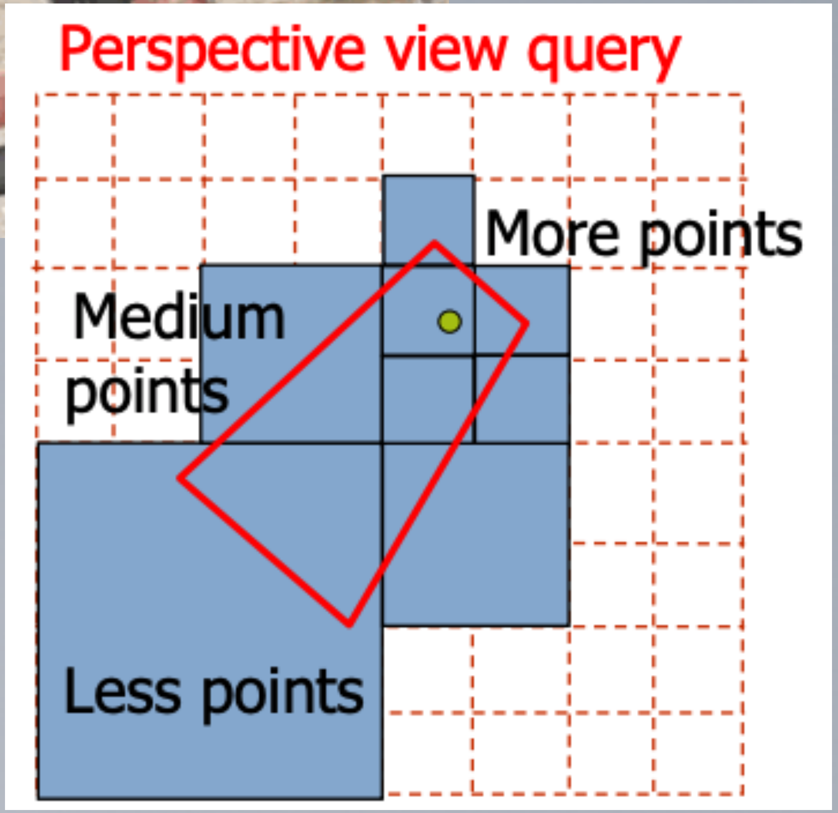
https://www.youtube.com/watch?time_continue=1&v=mJIF4wbnhpc

LOD - Level of Detail

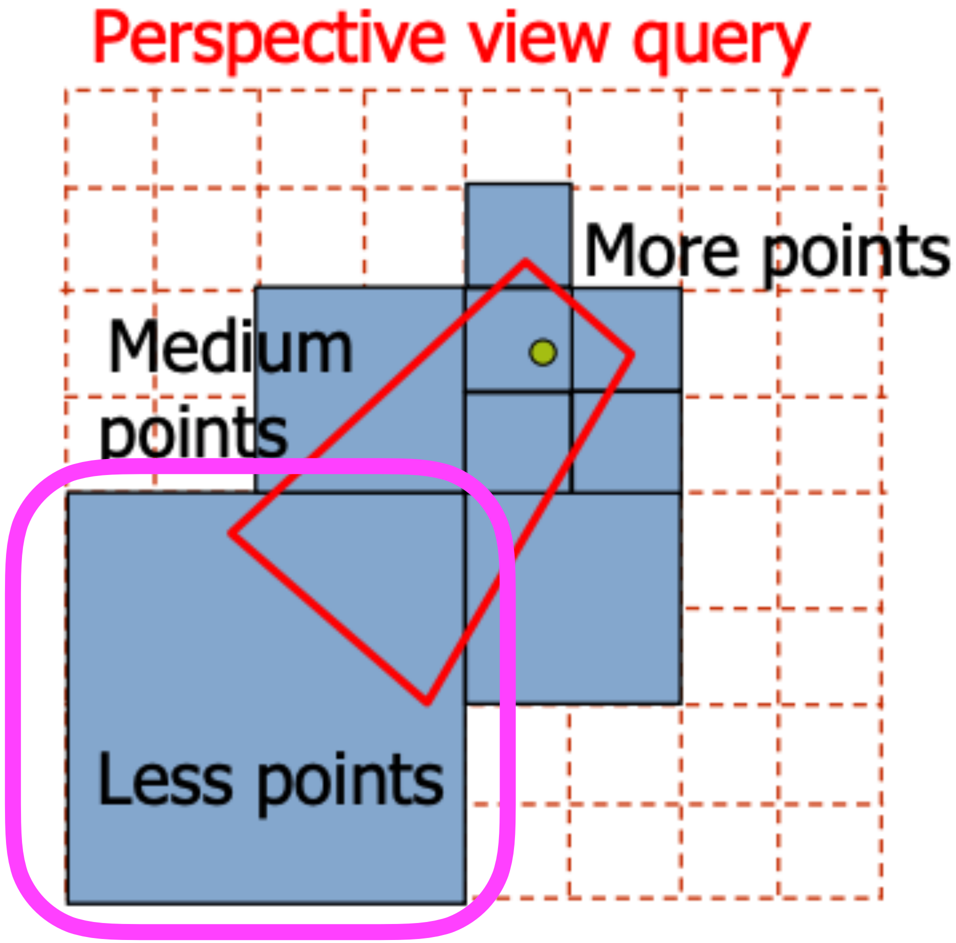
- GPU perf should not be wasted to render points which is not necessary

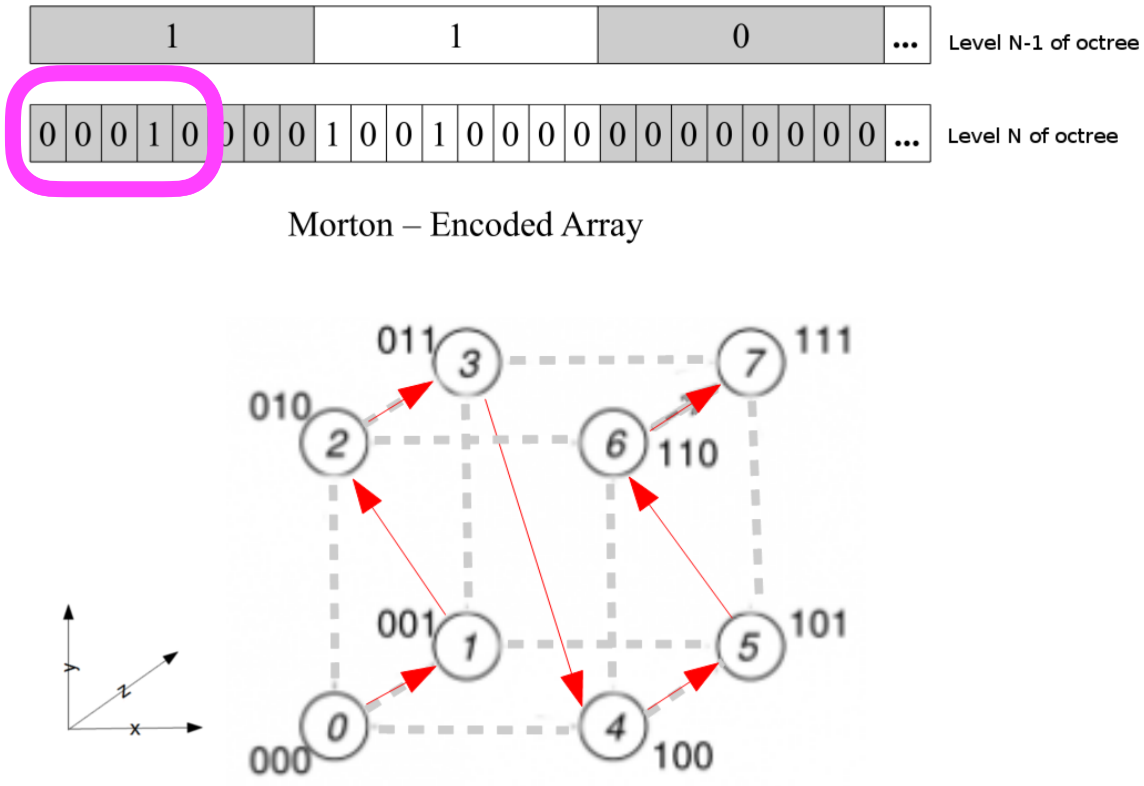
Solution

- Morton code allows fast LOD selection and compact storage with minimum necessary information



Example: Query of Octree with perspective view

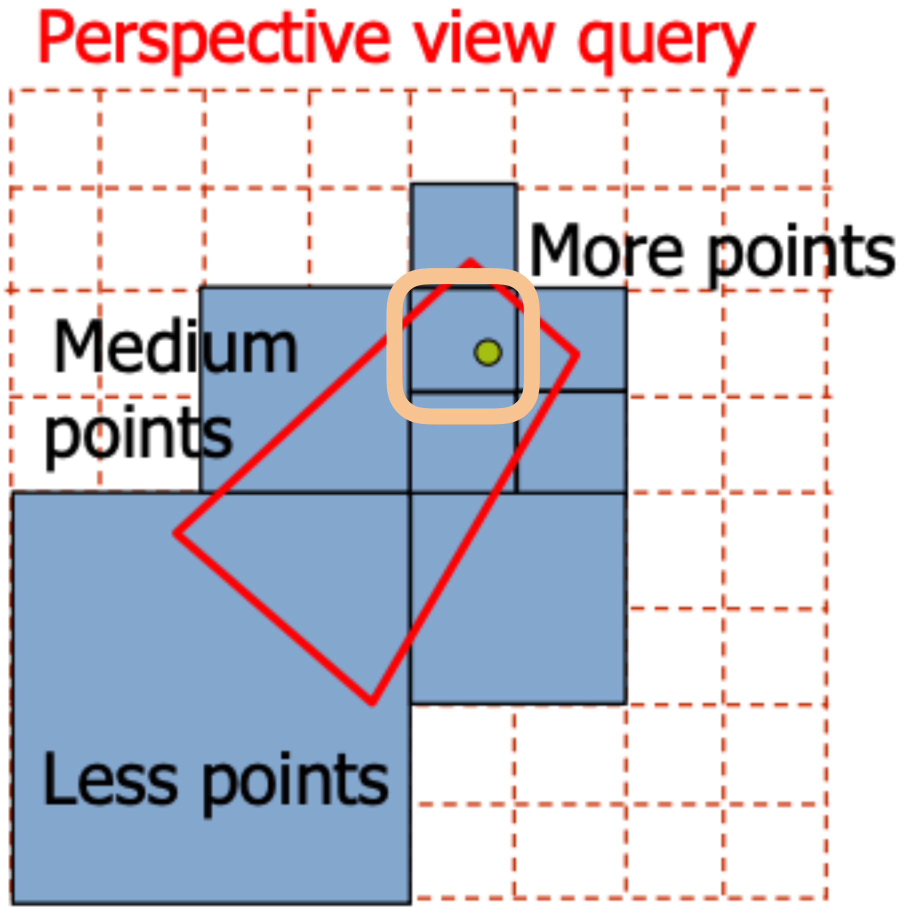


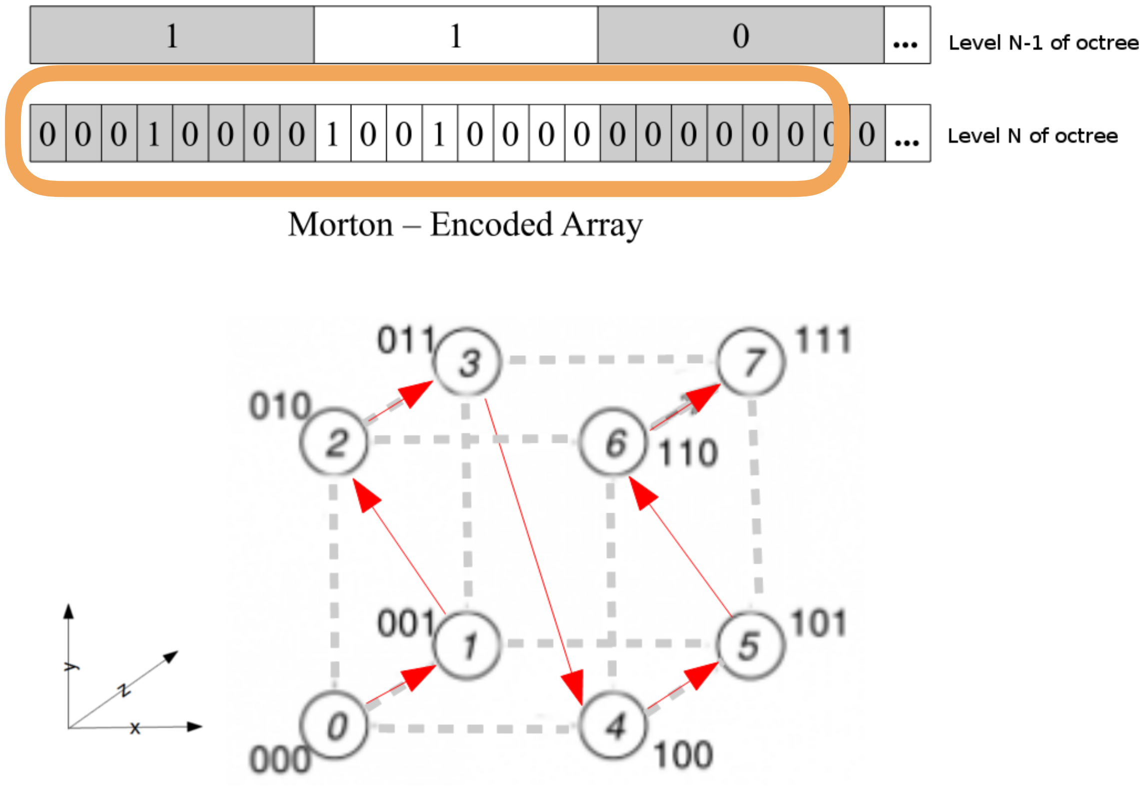


*Data is from Netherland escience center 2014 presentation

AIRSQUIRE AUTODESK

Example: Query of Octree with perspective view

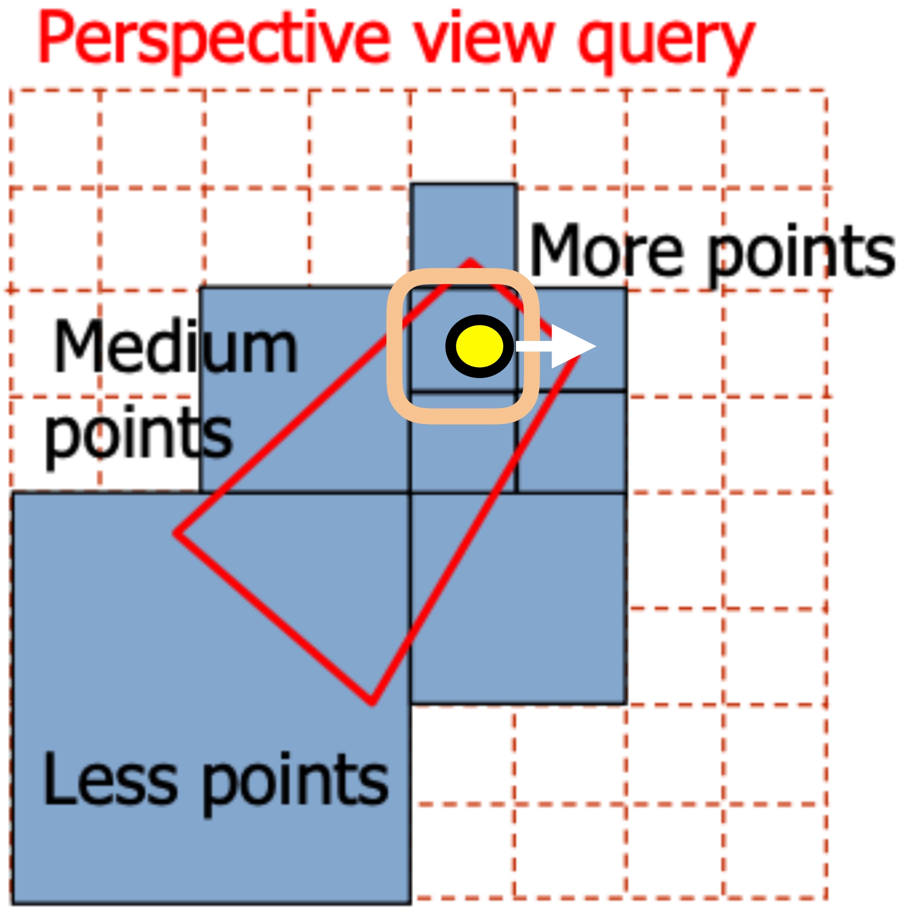


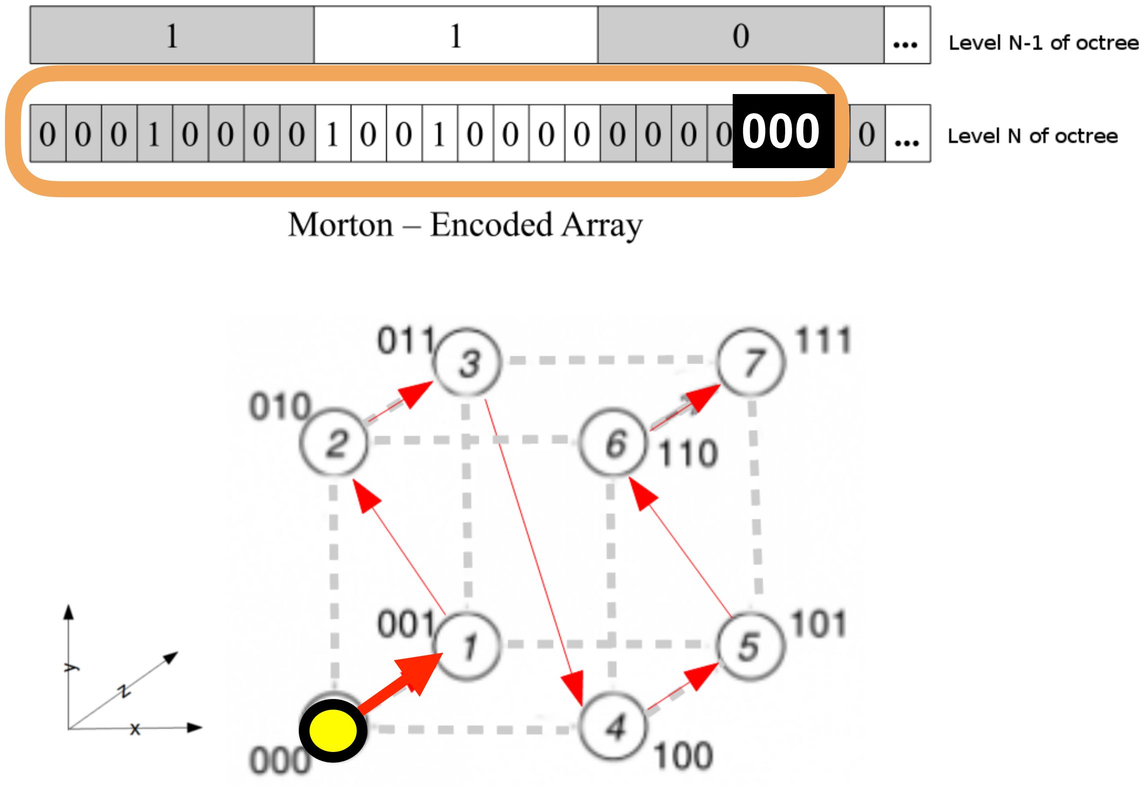


*Data is from Netherland escience center 2014 presentation

AIRSQUIRE
AUTODESK.

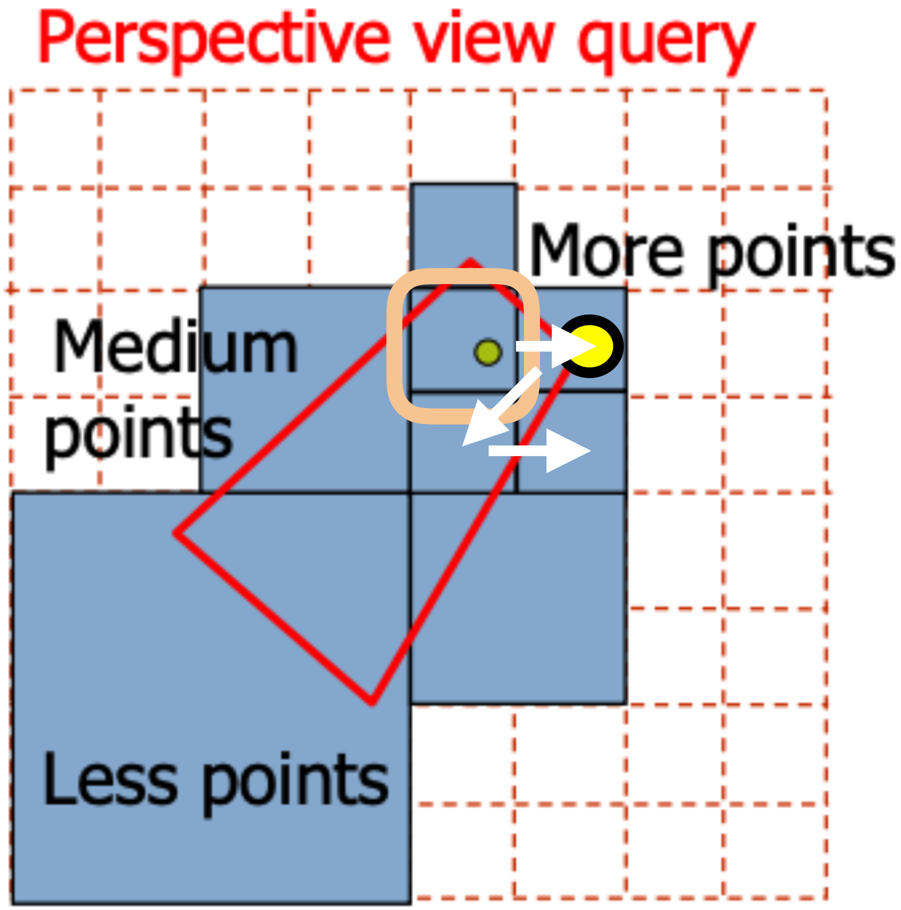
Example: Query of Octree with perspective view

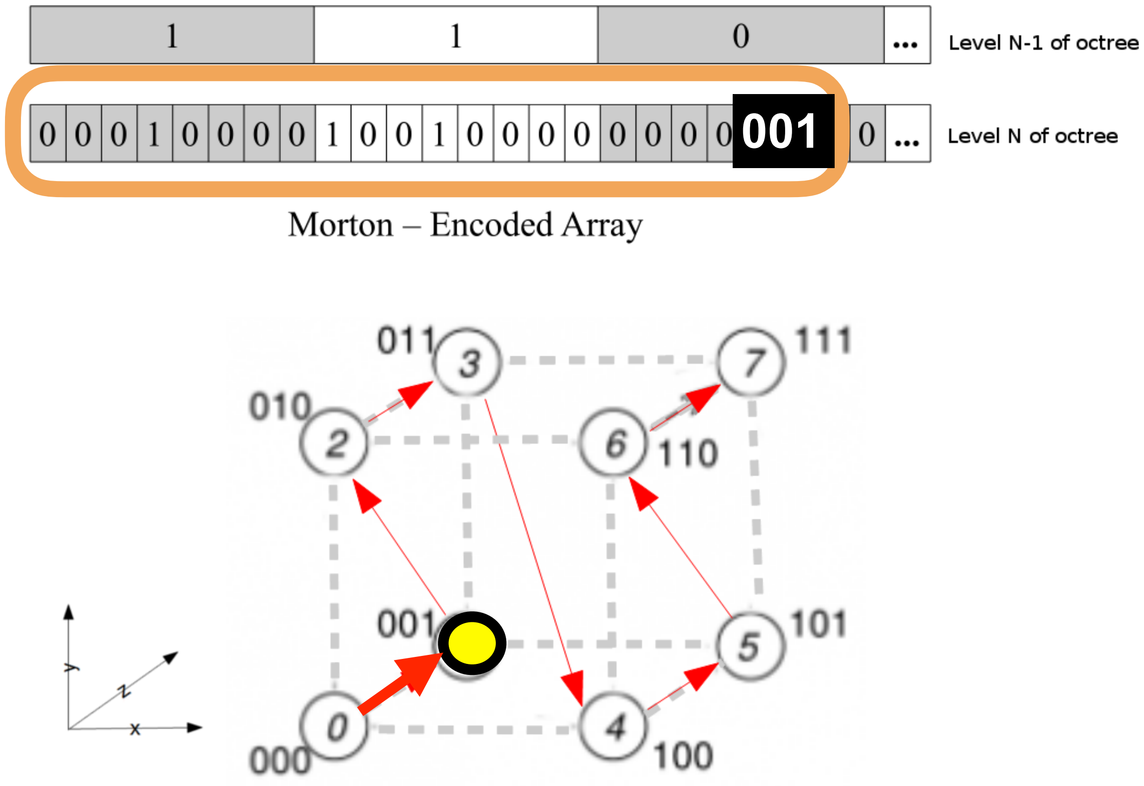




*Data is from Netherland escience center 2014 presentation

Example: Query of Octree with perspective view

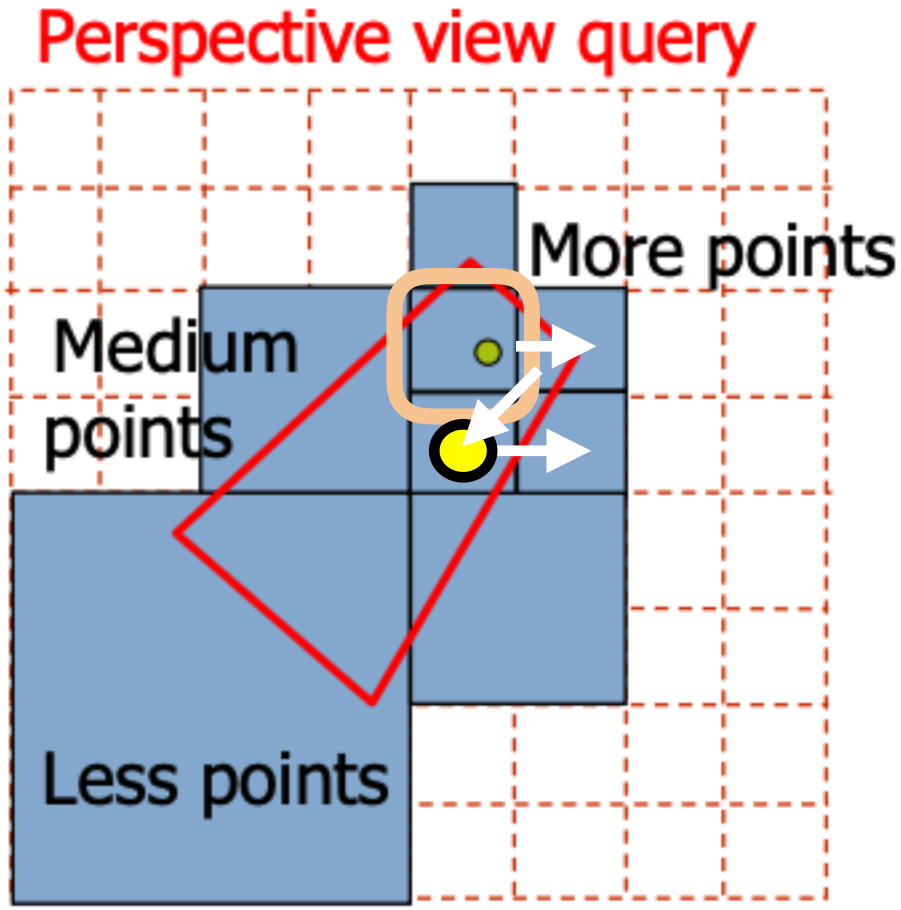


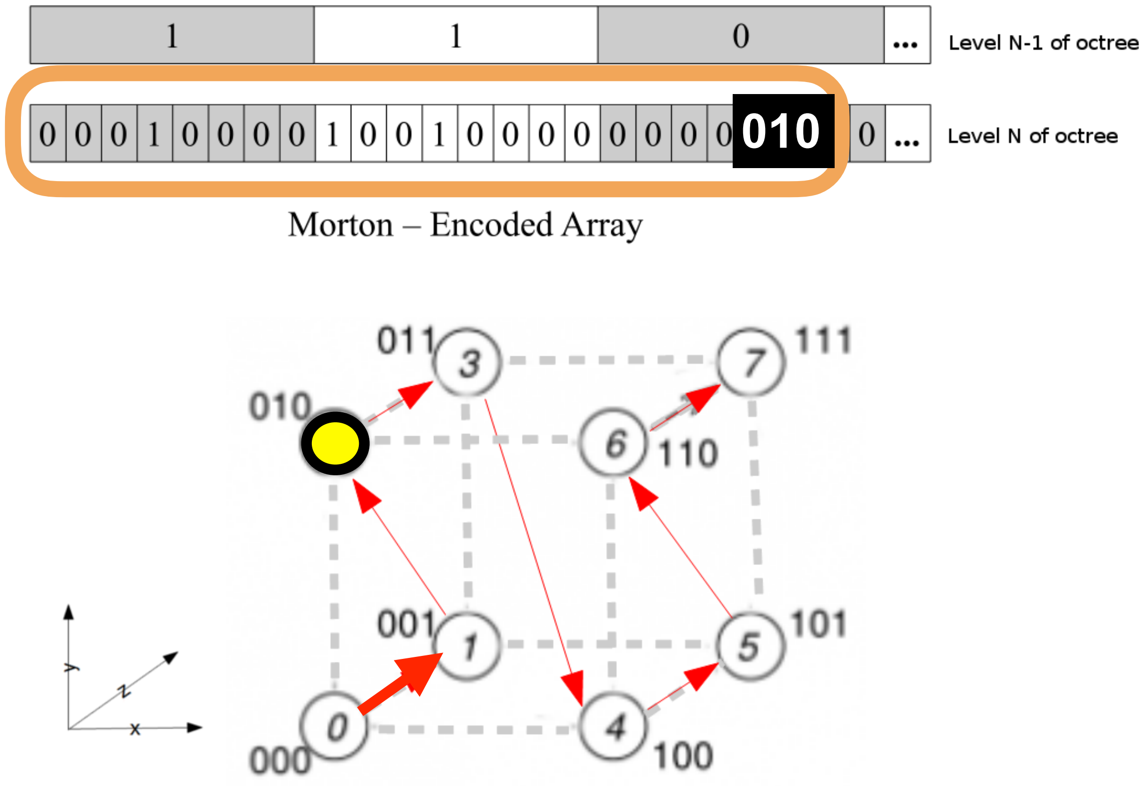


*Data is from Netherland escience center
2014 presentation

AIRSQUIRE
AUTODESK.

Example: Query of Octree with perspective view





*Data is from Netherland escience center 2014 presentation

AIRSQUIRE
AUTODESK.

Point cloud rendering loop

```
const pointCloudRenderingLoop () => {
  requestAnimationFrame(pointCloudRenderingLoop)
  const overlayScene = forgeViewer.impl.overlayScenes[pointCloudOverlayName]
  const result = pointCloud.queryPointBlock(
        forgeViewer.impl.camera,
        forgeViewer.impl.glrenderer()
  )
}

Class PointCloud {
   function queryPointBlock(camera: THREE.Camera, renderer: THREE.WebGLRenderer) {
        updateVisibility(this, camera, renderer)
        postProcessPointCloud(this)
   }
}
```

LOD / Streaming - Summary of workflow

1. Pre-process point cloud in *streaming server*

2. Persist spatial index information

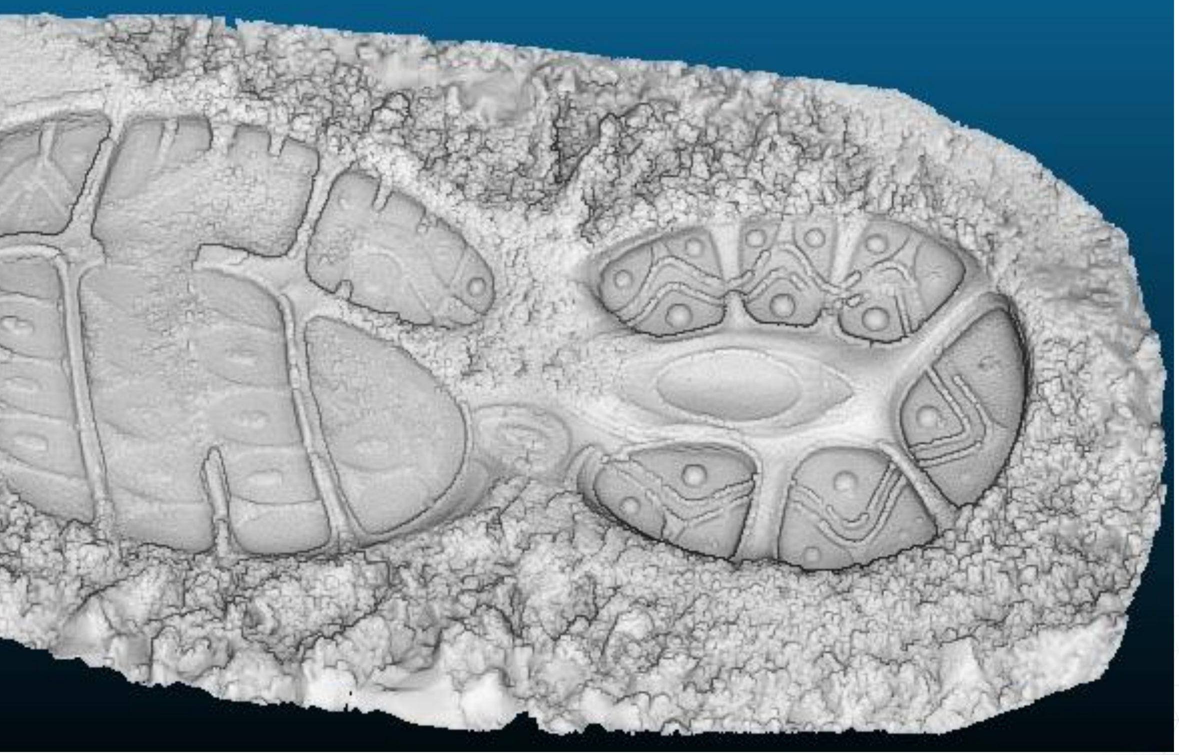
3. Load spatial root in frontend and binding LOD checking in render loop

4. Send data request to streaming server based on checked LOD

AIRSQUIRE

AUTODESK® **FORGE**

Shading



SSAO Shading

faster alternative to normal-based shading

EDL Shading

- doesn't rely on any information apart from the geometry itself
- Eye Dome Lighting (EDL)

AIRSQUIRE

AUTODESK® FORGE

http://docs.opengeospatial.org/per/18-048r1.html

Measurement /Section

AIRSQUIRE

AUTODESK® FORGE

Steps for “Measuring”

To Pick a point

0. Add a toolbar button and panel
1. For model point picking we use Forge VIEWER - “Snapper” Class
2. For point cloud picking we render a “Color Buffer” Target, PointCloud index === color
3. Calculate and ‘display’ the distance

Quick Demo

Let's look at the code:>

Steps for "Sectioning"

0. Use forge built-in section toolkit
1. Bind native forge clip plane change event to change shader uniforms

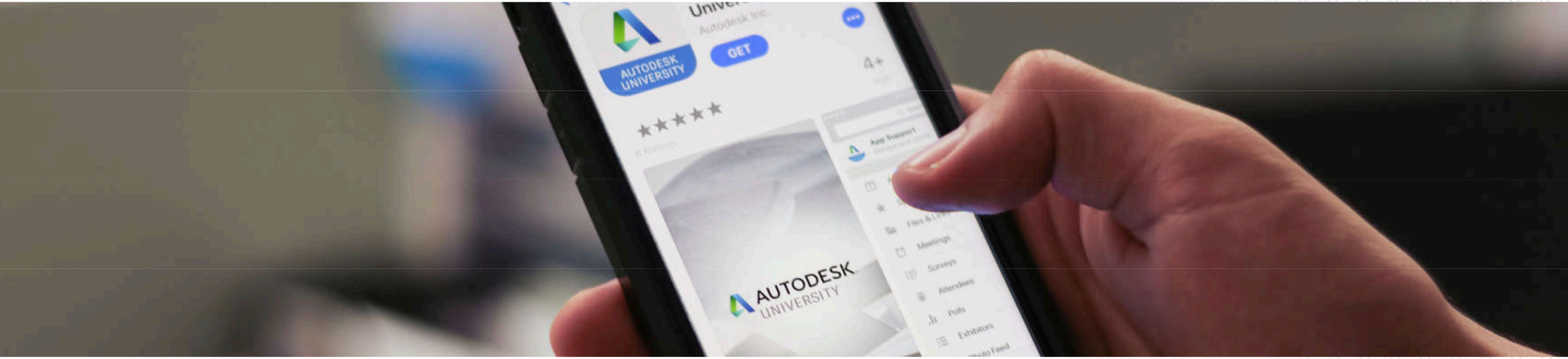
Let's look at the code:>

Summary

- Learnt about a ‘verification’ workflow
- Learnt how to combine **pointcloud+Model** in Forge Viewer
- Learnt about streaming point-clouds
- Learnt how to Measure and section point-clouds in Forge Viewer

Be heard !

Provide feedback in the CLASS SURVEY in the app



AIRSQUIRE

AUTODESK® FORGE

Questions ?

Links

https://github.com/wallabyway/forge-point-clouds

https://www.youtube.com/watch?v=wGxr-DLHHxM

http://forge.autodesk.com/accelerator

Contact

Yue You

www.Airsquire.ai

YouYue123

youyue@airsquire.ai

Michael Beale

Developer Advocate

@micbeale

michael.beale@autodesk.com

AIRSQUIRE

AUTODESK® FORGE