

# Extending IFC with point cloud data

## eg-ice 2015

Thomas Krijnen,  
Jakob Beetz

Eindhoven University  
of Technology,  
The Netherlands

Sebastian Ochmann,  
Richard Vock,  
**Raoul Wessel**  
Institute of Computer  
Science II,  
University of Bonn,  
Germany



**TU/e**

Technische Universiteit  
**Eindhoven**  
University of Technology

Where innovation starts



# Aims

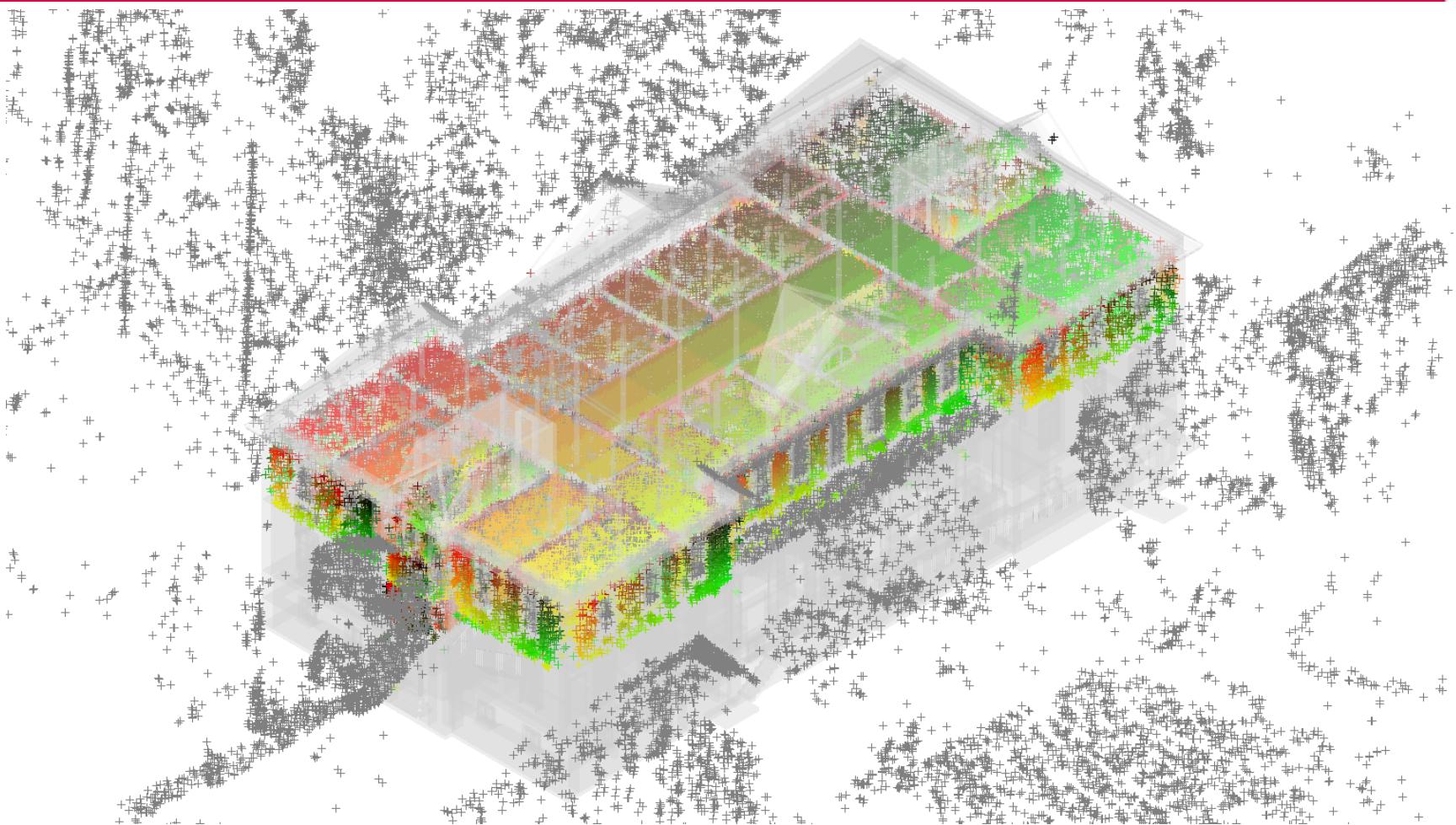
**Unite both data carriers**

**In a harmonized storage format**

**By extending the IFC schema**

**Introduce a binary serialization format for IFC**

# Aims



# IFC and point clouds anno 2014

#1=IFCCARTESIANPOINTLIST3D((121.76123809814453,-78.45201110839844,-13.620319366455078),  
(121.80341339111328,-78.57342529296875,-12.835342407226562), (121.7803726196289,-78.34296417236328,-  
13.846855163574219), (121.78942108154297,-78.3430404663086,-13.734870910644531), (121.78864288330078,-  
78.33686065673828,-13.621849060058594), (121.79190826416016,-78.33324432373047,-13.50927734375),  
(121.80005645751953,-78.33274841308594,-13.397228240966797), (121.82268524169922,-78.3357925415039,-  
13.173797607421875), (122.14295196533203,-78.85536193847656,1.7720565795898438), (122.14188385009766,-  
78.8489990234375,1.8842010498046875), (122.1812515258789,-78.74574279785156,1.8790130615234375),  
(2.644031524658203,-14.406140327453613,1.1417999267578125), (2.628744125366211,-  
14.345318794250488,1.1577377319335938), (3.482269287109375,-17.248559951782227,-1.4286651611328125),  
(3.32574462890625,-17.086864471435547,-1.4688568115234375), (3.4234485626220703,-15.845314025878906,-  
1.2468643188476562), (3.449533462524414,-15.883966445922852,-1.211181640625), (3.5097713470458984,-  
15.58512020111084,-1.1186370849609375), (4.405935287475586,-14.289386749267578,-1.1261444091796875),  
(4.405914306640625,-14.280056953430176,-1.0434646606445312), (4.378328323364258,-14.243566513061523,-  
1.0649948120117188), (5.349382400512695,-14.975531578063965,-0.90240478515625), (5.226930618286133,-  
14.915581703186035,-0.9336929321289062), (2.618467330932617,-10.692724227905273,-1.2475967407226562),  
(2.5753402709960938,-10.5256929397583,-1.2720489501953125), (2.642486572265625,-10.534729957580566,-  
1.2680206298828125), (2.5160579681396484,-10.395642280578613,-1.3473358154296875), (2.638456344604492,-  
10.48128890991211,-1.3255081176757812), (2.763673782348633,-10.58719539642334,-1.269317626953125),  
(2.4005050659179688,-10.004755973815918,-1.3596649169921875), (2.8383045196533203,-10.220603942871094,-  
1.3677139282226562), (2.854219436645508,-10.277368545532227,-1.3037109375), (2.8745193481445312,-  
10.280534744262695,-1.2966690063476562), (2.851787567138672,-10.229779243469238,-1.3521499633789062),  
(2.974489212036133,-10.0681791305542,-1.234466552734375), (1.8945045471191406,-10.313359260559082

# IFC and point clouds anno 2014

## Points for improvement:

No per-point attributes, such as colors, etc.

No level of detail

No explicit means for decomposition

No metadata, such as scanner model, etc.

No way to extract localized subsets

Slow to parse and leading to exorbitant data sizes

# Proposed structure

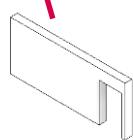
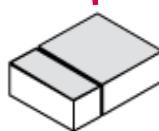
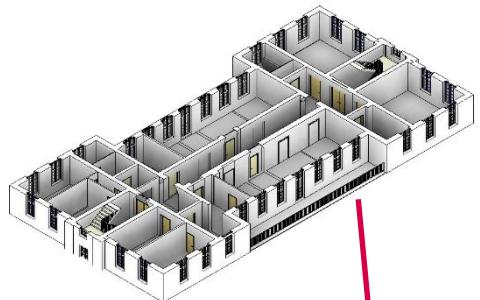
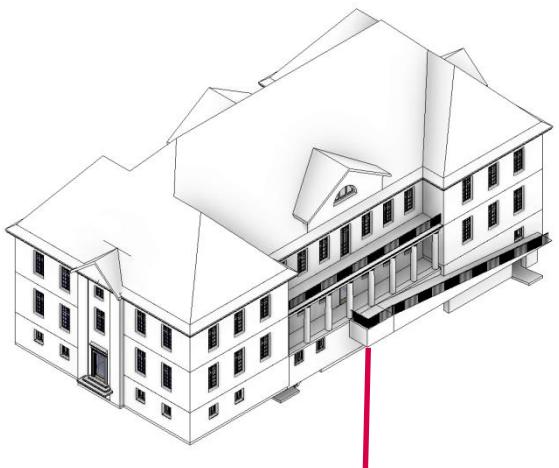
A **schema extension**  
with **point cloud compression techniques**  
that deepens the **semantic relation** between points and  
building elements.

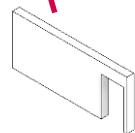
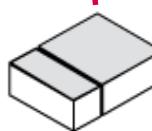
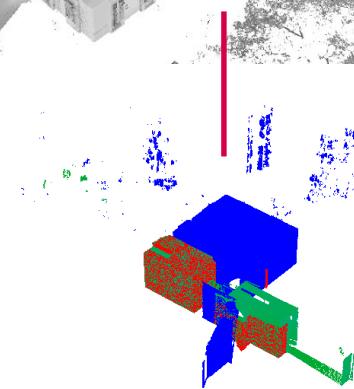
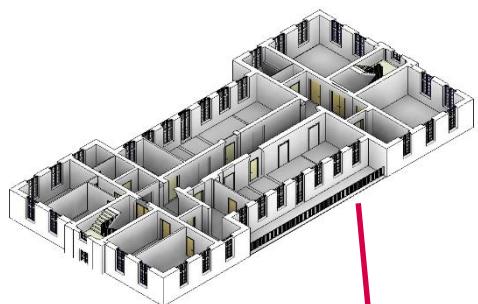
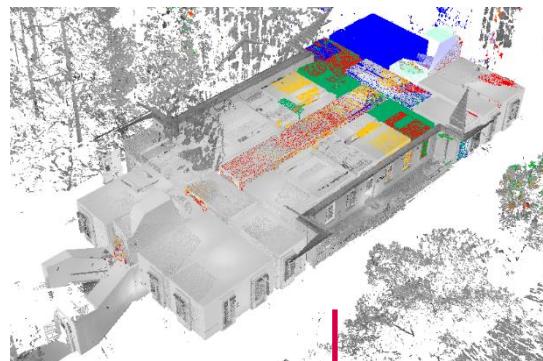
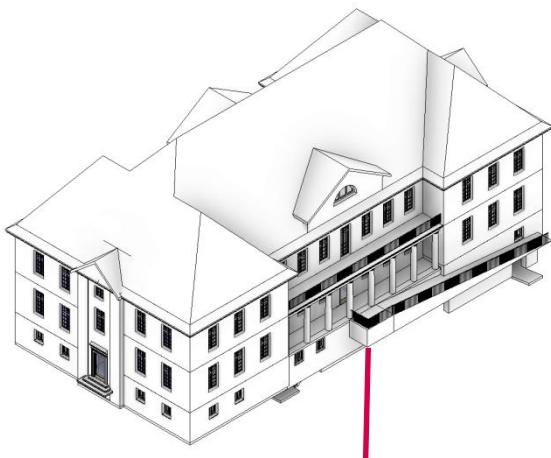
# Proposed structure

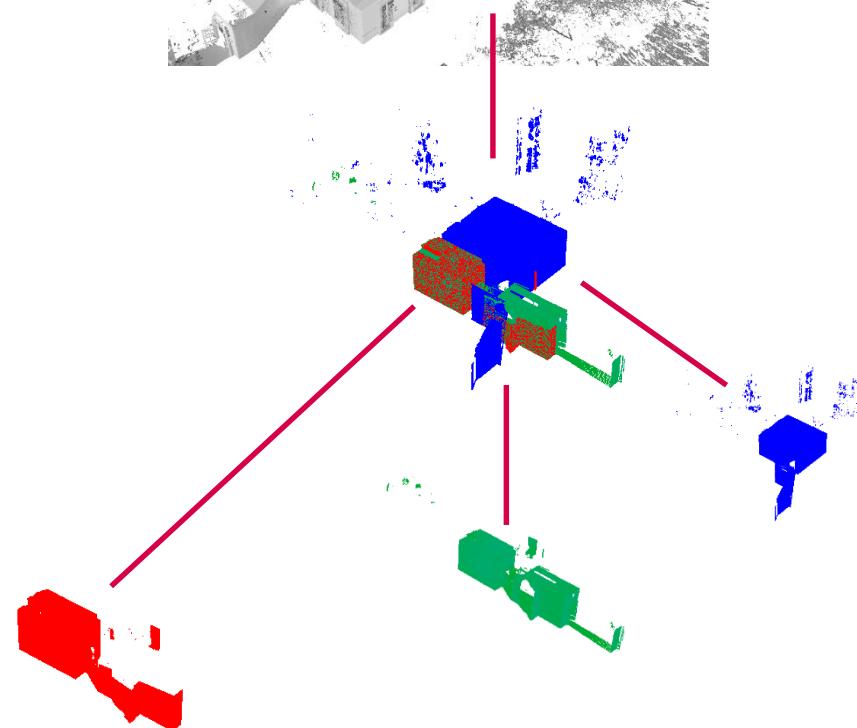
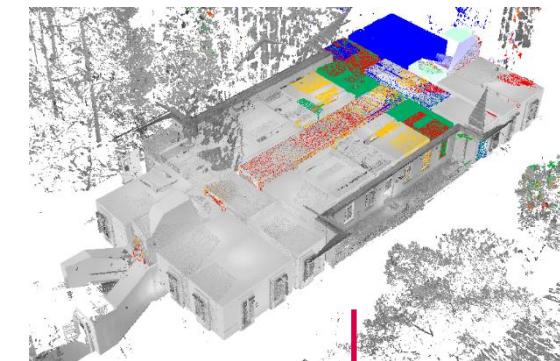
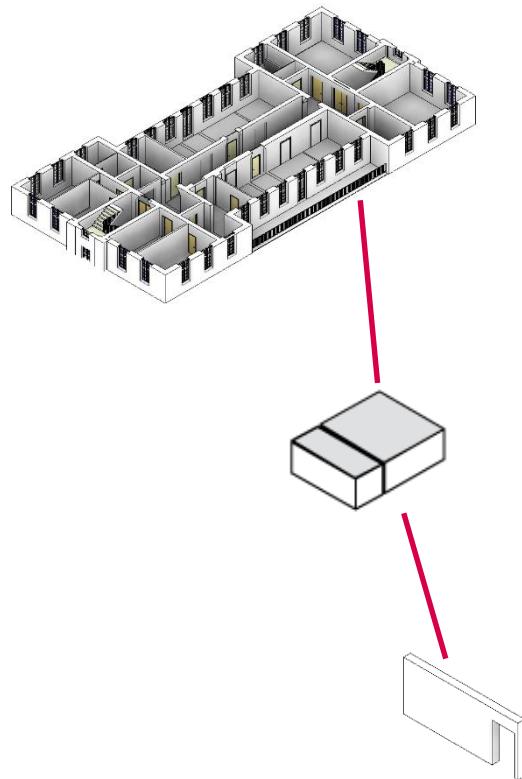
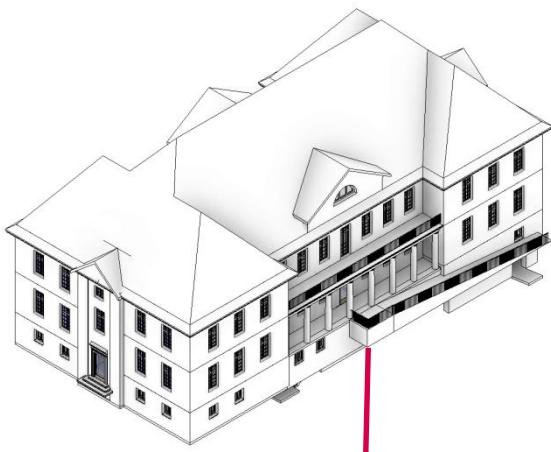
Three layers of compression:

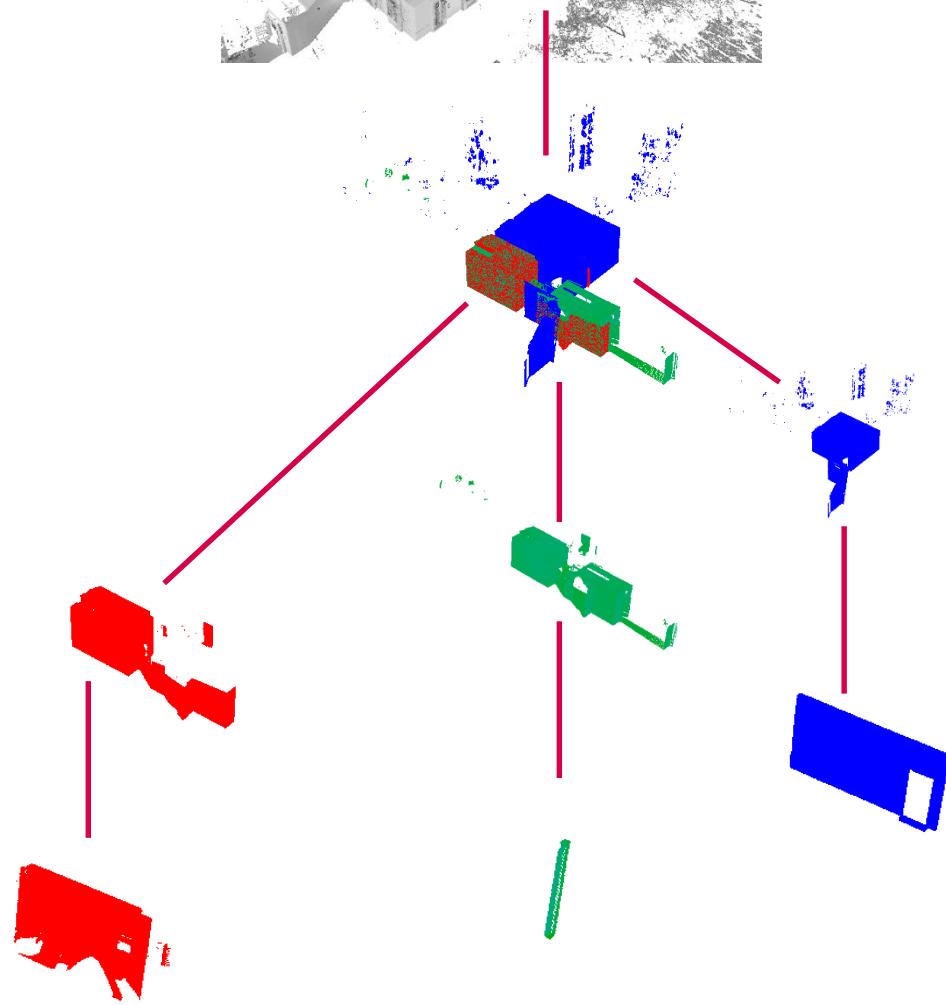
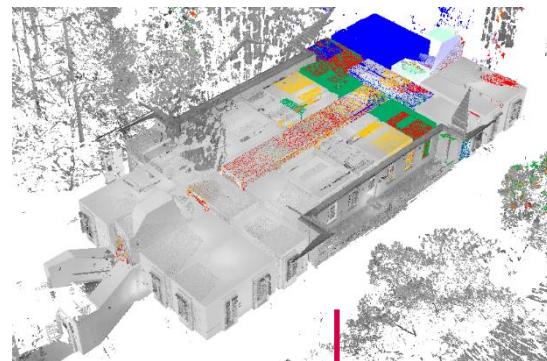
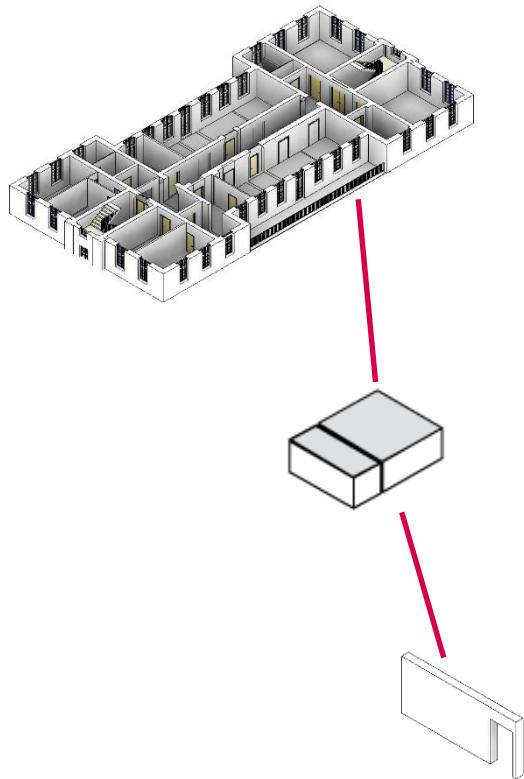
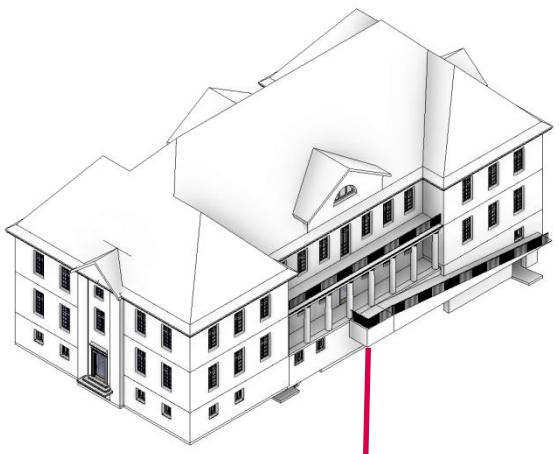
- Project points into **bounded parametric space**
- Reduce the dimensionality of points by **grid discretization**
- **Transparent compression** on a file format level (HDF5)

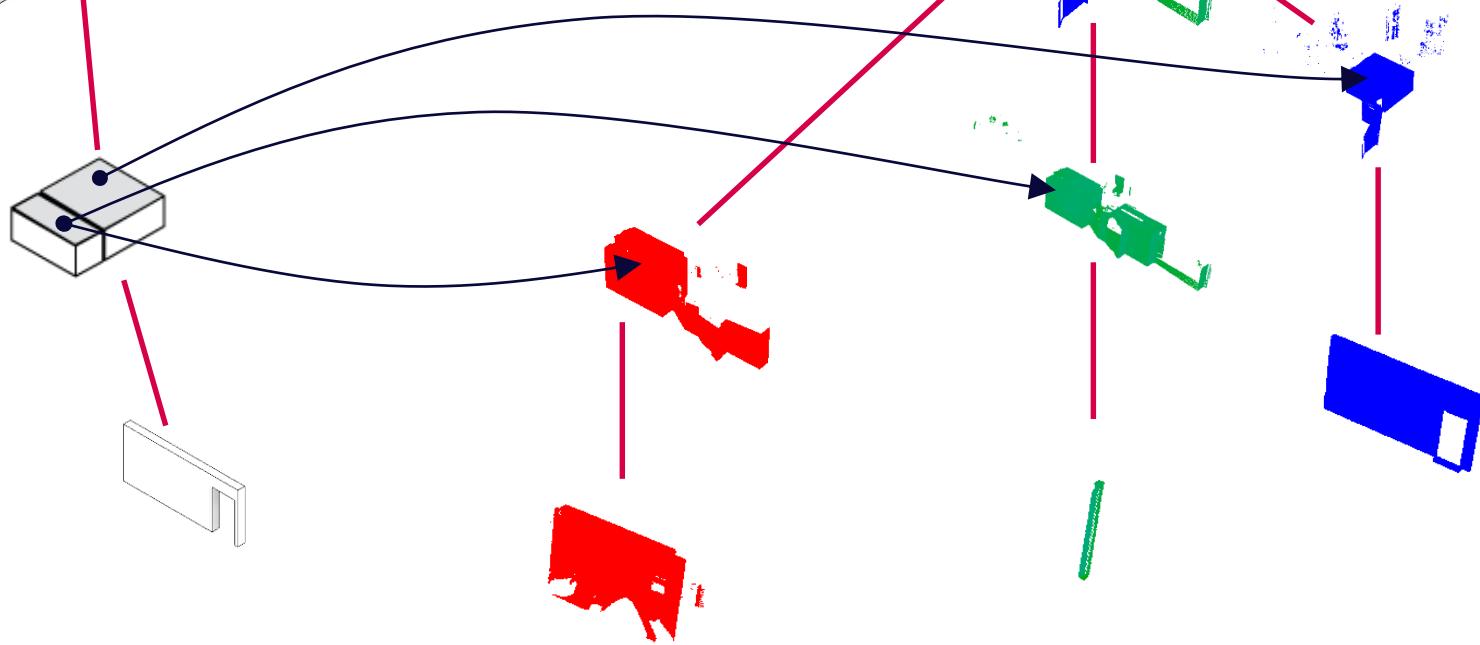
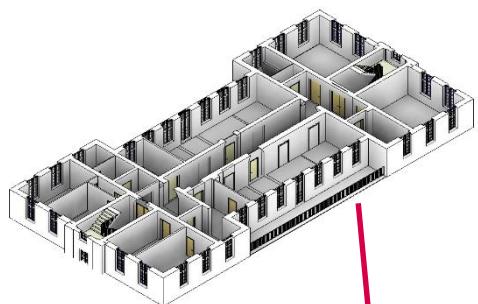
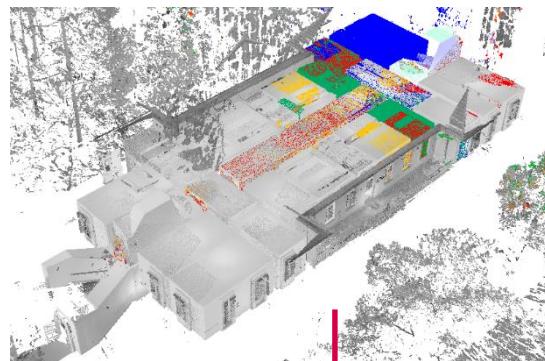
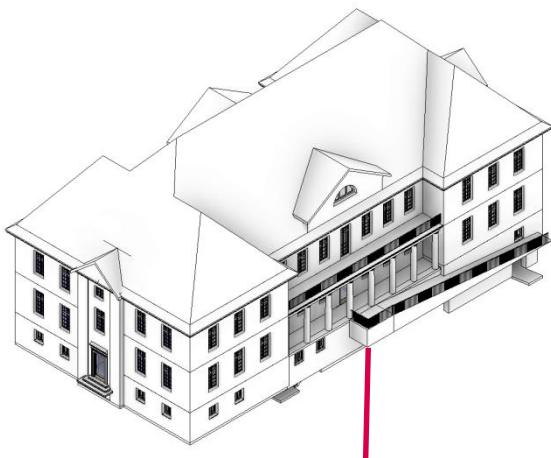


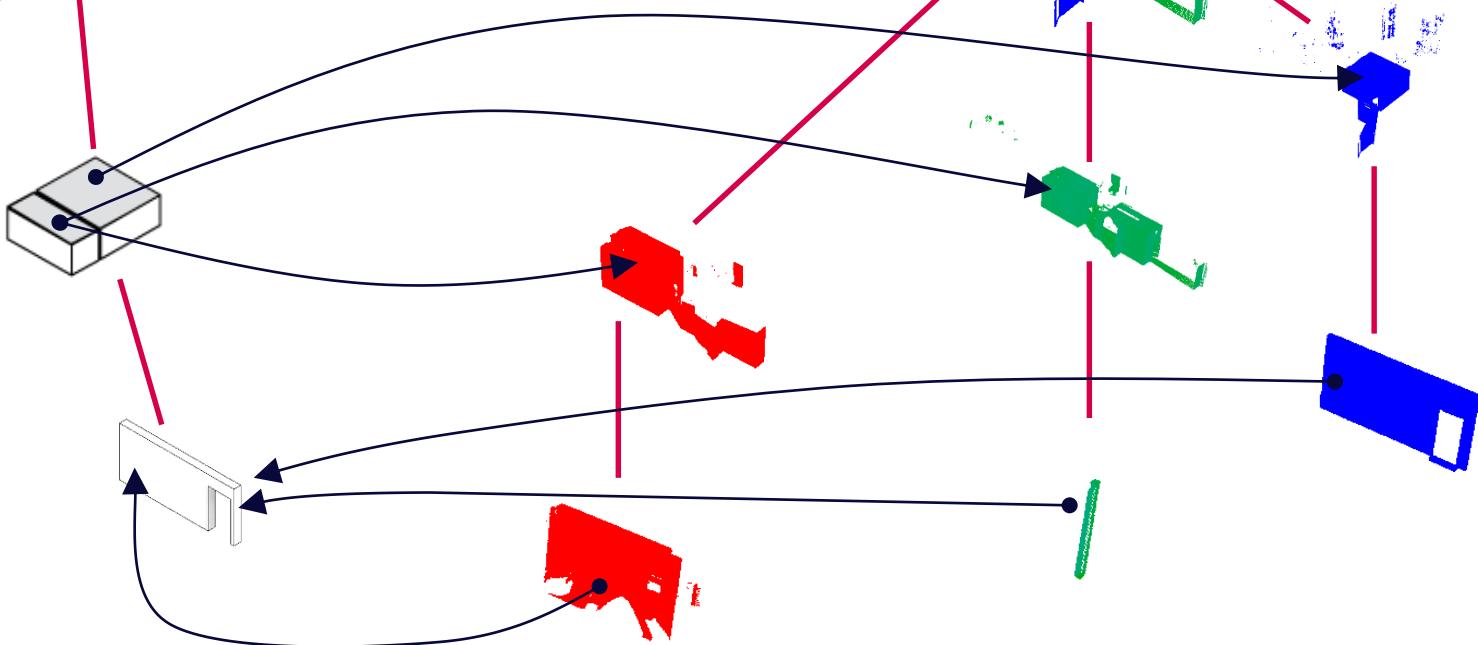
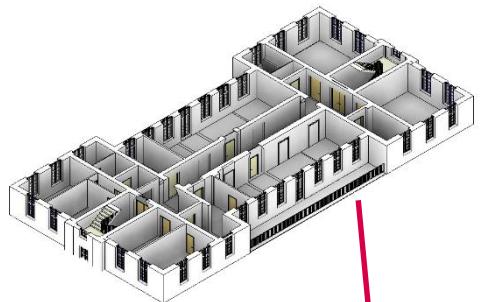
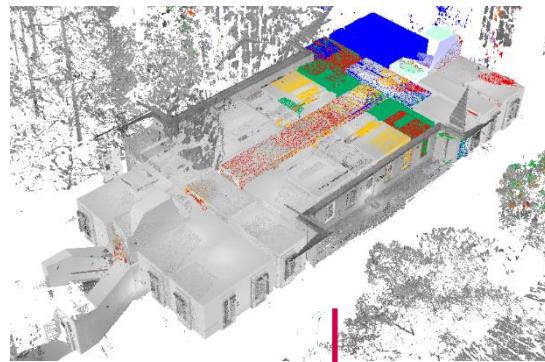
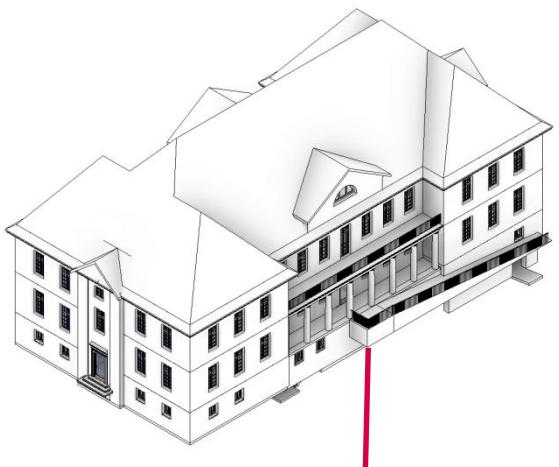






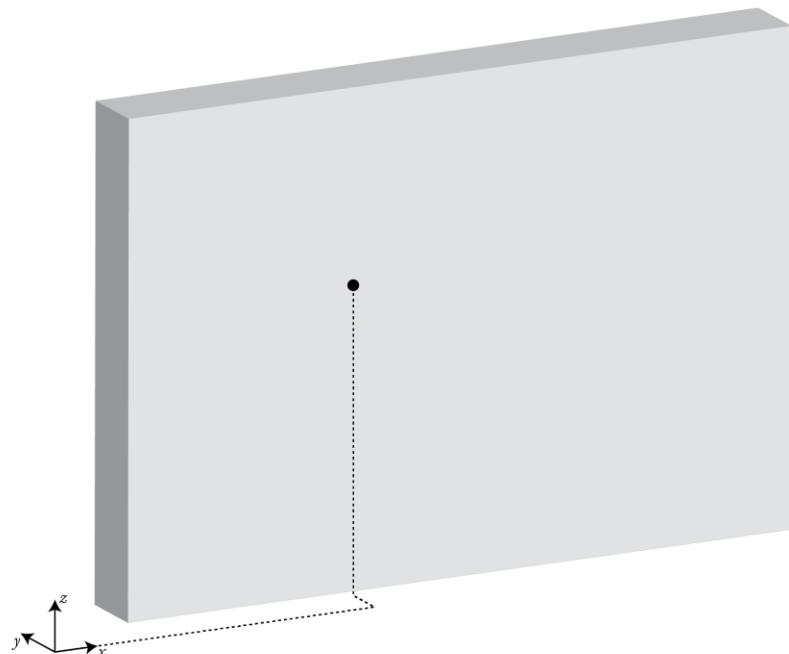






# Proposed structure

Cartesian coordinates offer:

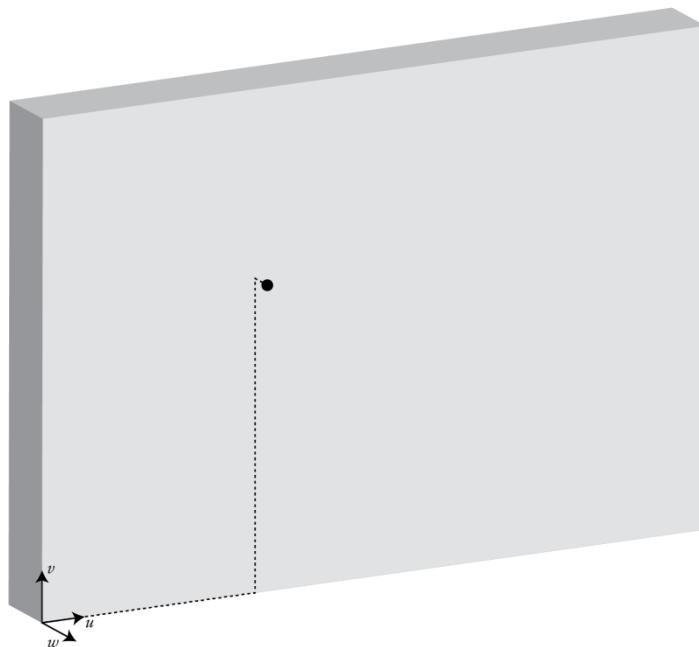


An **easy and intuitive** way to store point cloud data.

Intended for **geographic features** or **unassociated points** unrelated to building element.

# Proposed structure

## Parametrization

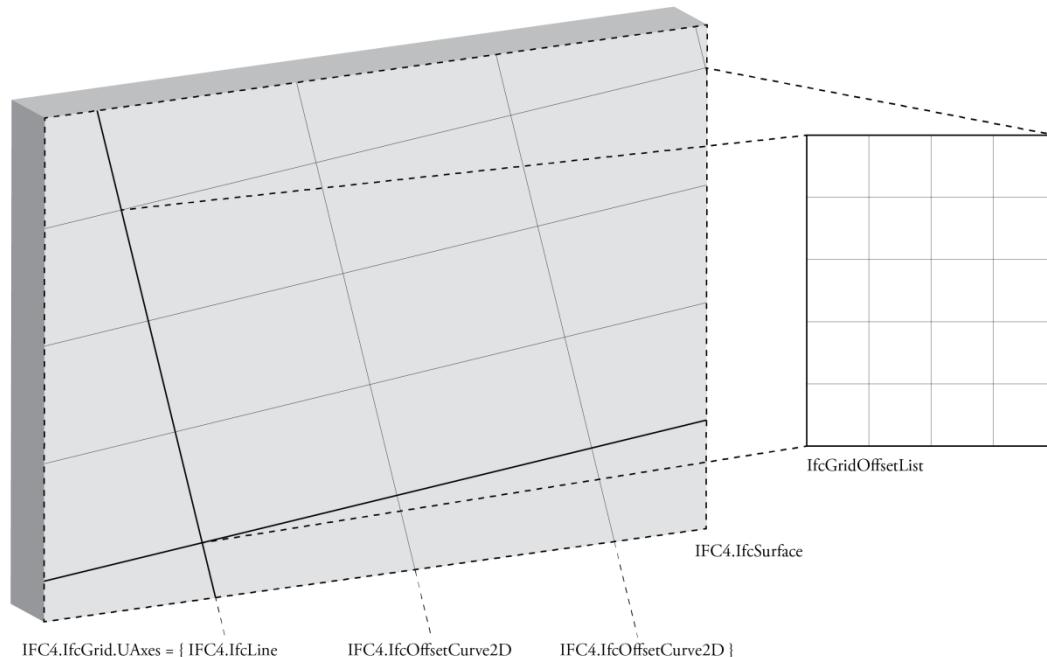


Associated points described in the **parametric** space of building surfaces.

Three components  $\{u, v, w\}$  required, but the range of values is **bounded** and can be **more efficiently encoded**.

# Proposed structure

## Height fields



{u,v} components defined as the intersection of **grid axes**.

Per point only **one component** needs to be stored.

# Proposed structure

## Floating point discretization

(0.8482145585275755, 0.4089384818729891, 0.8027061702482456, 0.11449717768247669)

# Proposed structure

## Floating point discretization

(0.8482145585275755, 0.4089384818729891, 0.8027061702482456, 0.11449717768247669)

≈	≈	≈	≈
(55587,	26799,	52605,	7503)
-----	-----	-----	-----
65535	65535	65535	65535

# Proposed structure

## Floating point discretization

(0.8482145585275755, 0.4089384818729891, 0.8027061702482456, 0.11449717768247669)

≈	≈	≈	≈
(55587,	26799,	52605,	7503)
-----	-----	-----	-----
65535	65535	65535	65535

((55587, 26799, 52605, 7503), 65535)

# Proposed structure

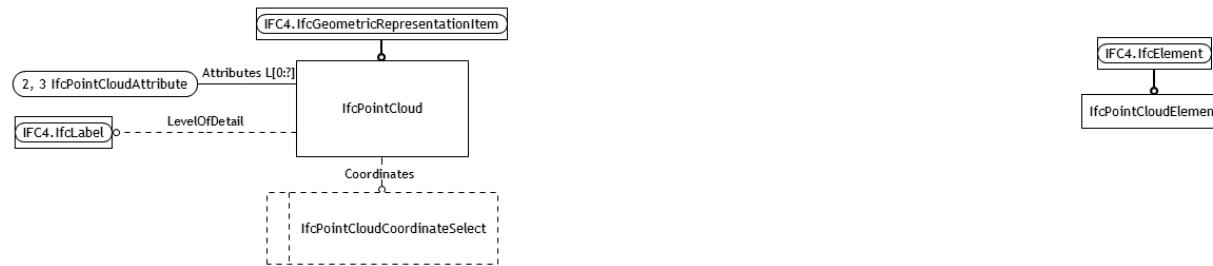
## Floating point discretization

(0.8482145585275755, 0.4089384818729891, 0.8027061702482456, 0.11449717768247669)

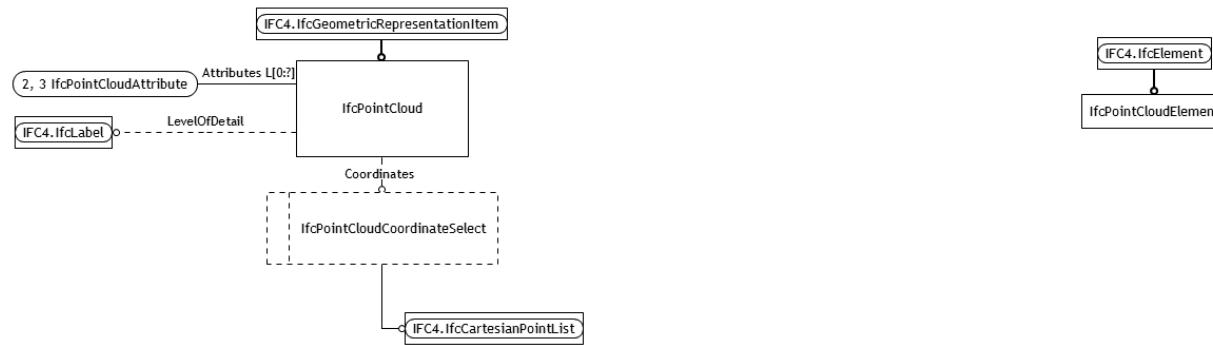
≈	≈	≈	≈
(55587,	26799,	52605,	7503)
-----	-----	-----	-----
65535	65535	65535	65535

((55587, 26799, 52605, 7503), 65536)

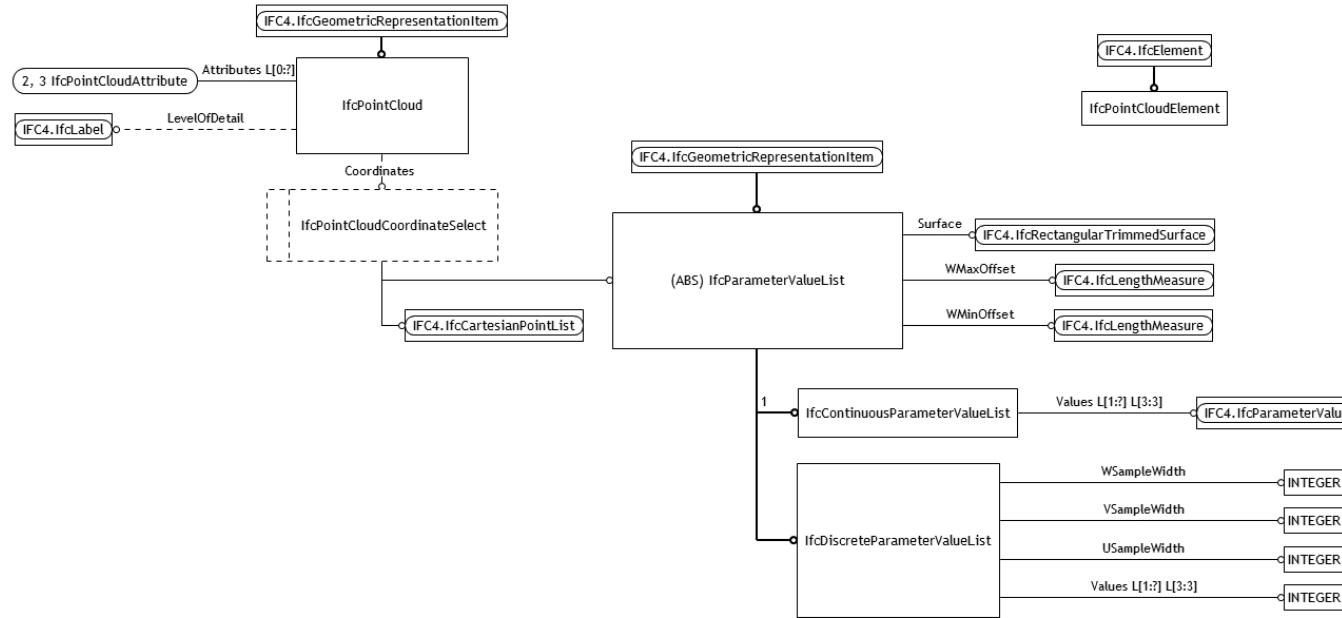
# Proposed structure



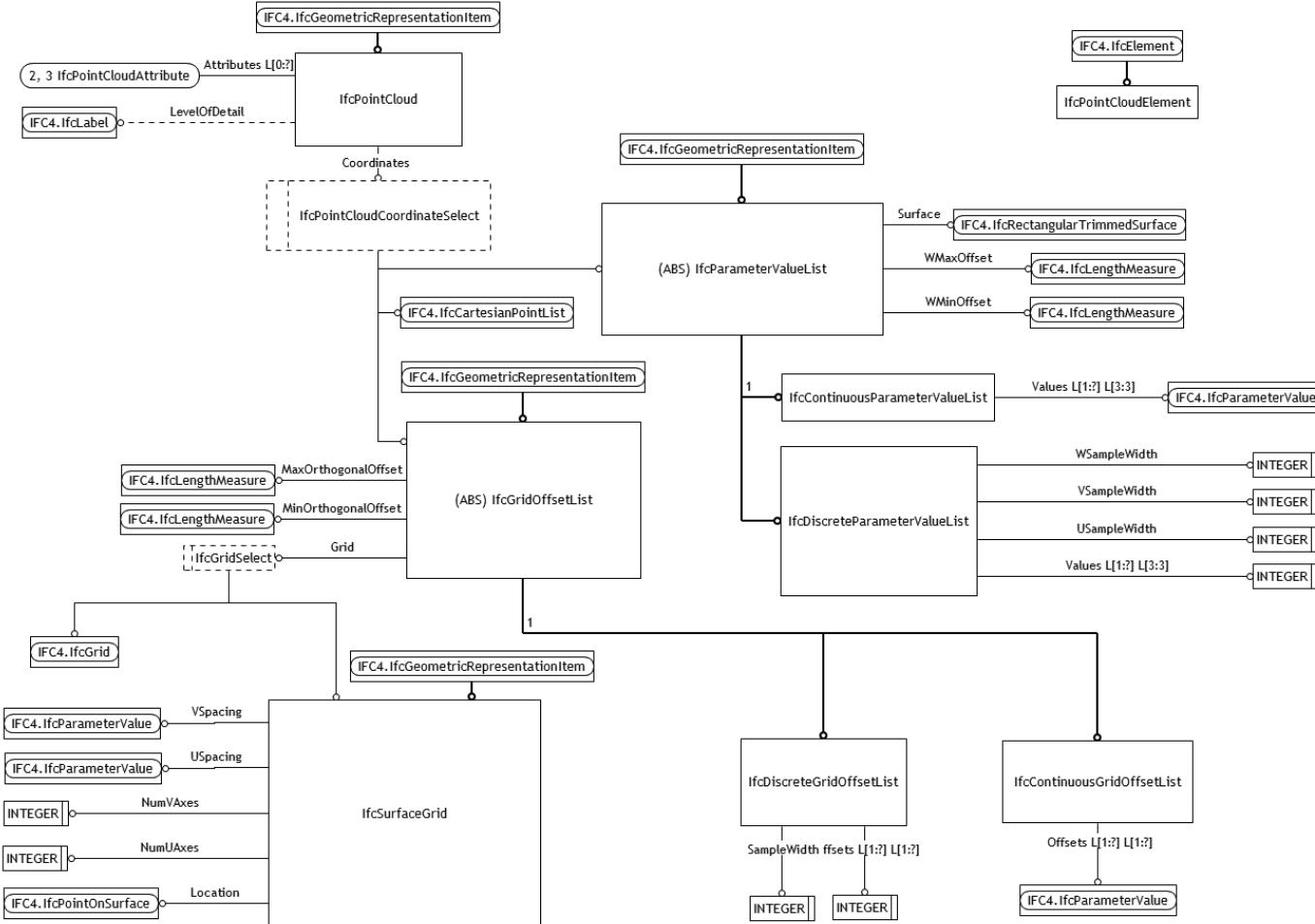
# Proposed structure



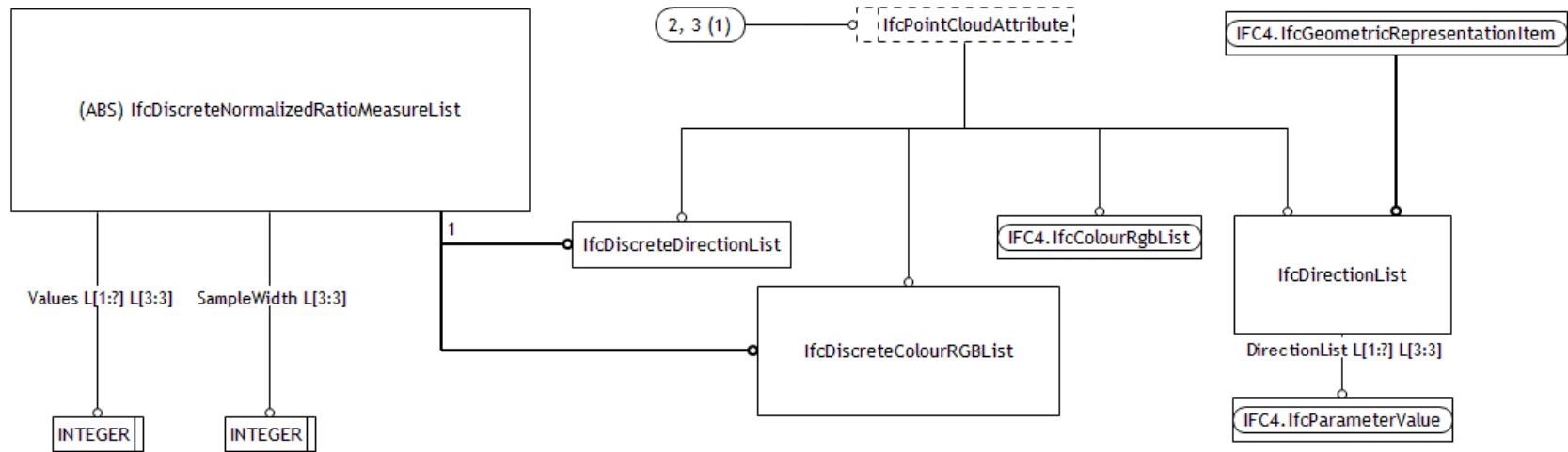
# Proposed structure



# Proposed structure



# Proposed structure



# Serialization in HDF5

HDF5 is a:  
**hierarchical, binary** data format  
for **heterogeneous** data  
stored according to **self-documenting** data types  
with **b-tree indexes** and **transparent compression**.  
It is an **open standard** and an  
**EXPRESS-HDF5 mapping is standardized as ISO 10303-26.**

# Serialization in HDF5

HDFView 2.31

File Window Tools Help

Recent Files J:\src\ifcpc\Plan3D\_Haus30\_PREVIEW\_NEW\_plus\_pc.ifc.hdf

Aggr\_CoordList\_640482 at /Haus30\_population/IfcCartesianPointList3D\_obj

Table

	0	1	2
0	-1.62448185667157		
1	-1.68...	14.30...	7.215...
2	-1.58...	14.36...	7.193...
3	-1.51...	14.33...	7.212...
4	-1.44...	14.31...	7.214...
5	-1.57...	14.47...	3.875...
6	-1.51...	14.40...	7.169...
7	-1.34...	14.29...	7.213...
8	-1.47...	14.47...	3.879...
9	-1.36...	14.34...	7.205...
10	-1.61...	14.51...	7.104...
11	-1.65...	14.56...	7.079...
12	-1.61...	14.53...	7.096...
13	1.30	14.32	7.211

IfcBuildingStorey\_instances at /Haus30\_population/IfcBuildingStorey\_objects/ [Plan3D\_Haus30\_PREVIEW\_NEW...]

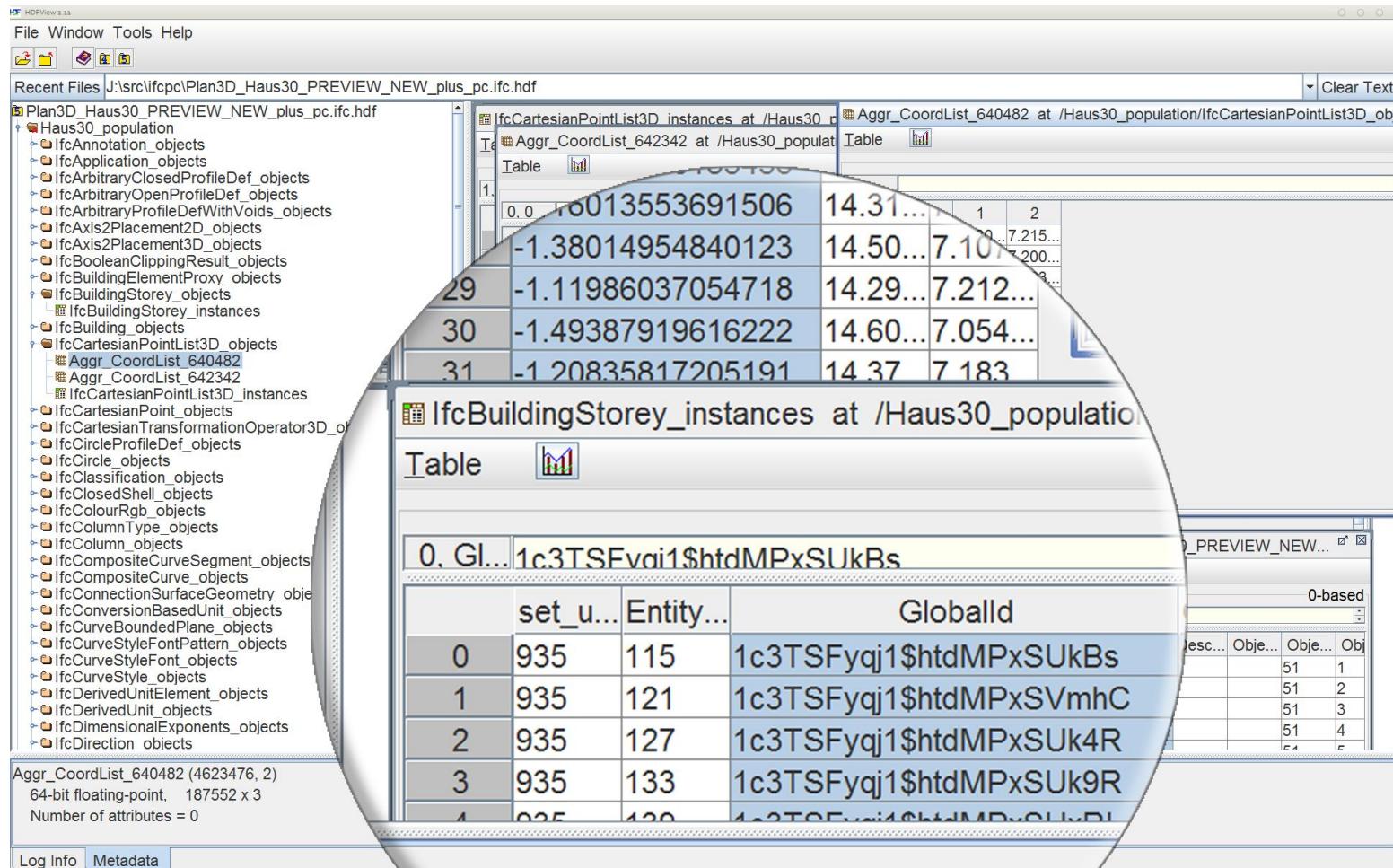
Table

	Entity...	GlobalId	Own...	Own...	Name	Desc...	Obj...	Obj...	Obj...
0	935	115	1c3TSFyqj1\$htdMPxSUKBs	63	0	Ebene -1		51	1
1	935	121	1c3TSFyqj1\$htdMPxSVmhC	63	0	Eingang		51	2
2	935	127	1c3TSFyqj1\$htdMPxSUK4R	63	0	Ebene 0		51	3
3	935	133	1c3TSFyqj1\$htdMPxSUK9R	63	0	Ebene 1		51	4
4	935	139	1c3TSFyqj1\$htdMPxSUK4L	63	0	Ebene 2		51	5

Aggr\_CoordList\_640482 (4623476, 2)  
64-bit floating-point, 187552 x 3  
Number of attributes = 0

Log Info Metadata

# Serialization in HDF5



# Serialization in HDF5

The screenshot shows the HDFView 2.31 interface with the following details:

- File Menu:** File, Window, Tools, Help.
- Recent Files:** J:\src\ifcpc\Plan3D\_Haus30\_PREVIEW\_NEW\_plus\_pc.ifc.hdf
- Left Panel (Tree View):** Shows the hierarchical structure of the HDF5 file, including groups like Plan3D\_Haus30\_PREVIEW\_NEW\_plus\_pc.ifc.hdf, Haus30\_population, IfcBuildingStorey\_objects, IfcBuilding\_instances, IfcCartesianPointList3D\_objects, Aggr\_CoordList\_640482, Aggr\_CoordList\_642342, IfcCartesianPointList3D\_instances, IfcCartesianPoint\_objects, IfcCircleProfileDef\_objects, IfcCircle\_objects, Classification\_objects, IfcCurveBoundedPlane, IfcCurveStyleFontPattern\_objects, IfcCurveStyleFont\_objects, IfcCurveStyle\_objects, IfcDerivedUnitElement\_objects, IfcDerivedUnit\_objects, IfcDimensionalExponents\_objects, and IfcDirection\_objects.
- Central Panels (Tables):**
  - IfcCartesianPointList3D\_instances at /Haus30\_population/IfcCartesianPointList3D\_objects/Aggr\_CoordList\_640482:** A table showing a list of coordinates. The first few rows are:

0	1	2
-1.62448185667157		
-1.59010970850125	14.57...	7.009...
-1.64800431001544	14.62...	7.041...
-1.34185869274497	14.46...	3.887...
  - Aggr\_CoordList\_640482 at /Haus30\_population/IfcCartesianPointList3D\_objects/Aggr\_CoordList\_640482:** A table showing a list of coordinates. The first few rows are:

0	1	2
-1.62...	14.30...	7.215...
-1.68...	14.35...	7.200...
-1.58...	14.36...	7.193...
-1.51...	14.33...	7.212...
  - IfcBuildingStorey\_instances at /Haus30\_population/IfcBuildingStorey\_objects/ [Plan3D\_Haus30\_PREVIEW\_NEW...]**: A table showing building storey instances. The first few rows are:

0	1	2
1c3TSFyqj1\$htdMPxSUkBs	GlobalId	Owner...
935	115	63
935	121	63
935	127	63
- Bottom Panel (Text):** Aggr\_CoordList\_640482 (4623476, 2)  
64-bit floating-point, 187552 x 3  
Number of attributes = 0
- Bottom Navigation:** Log Info, Metadata.

# Implementation

A **prototypical implementation** is provided using only **open source components**, such as:

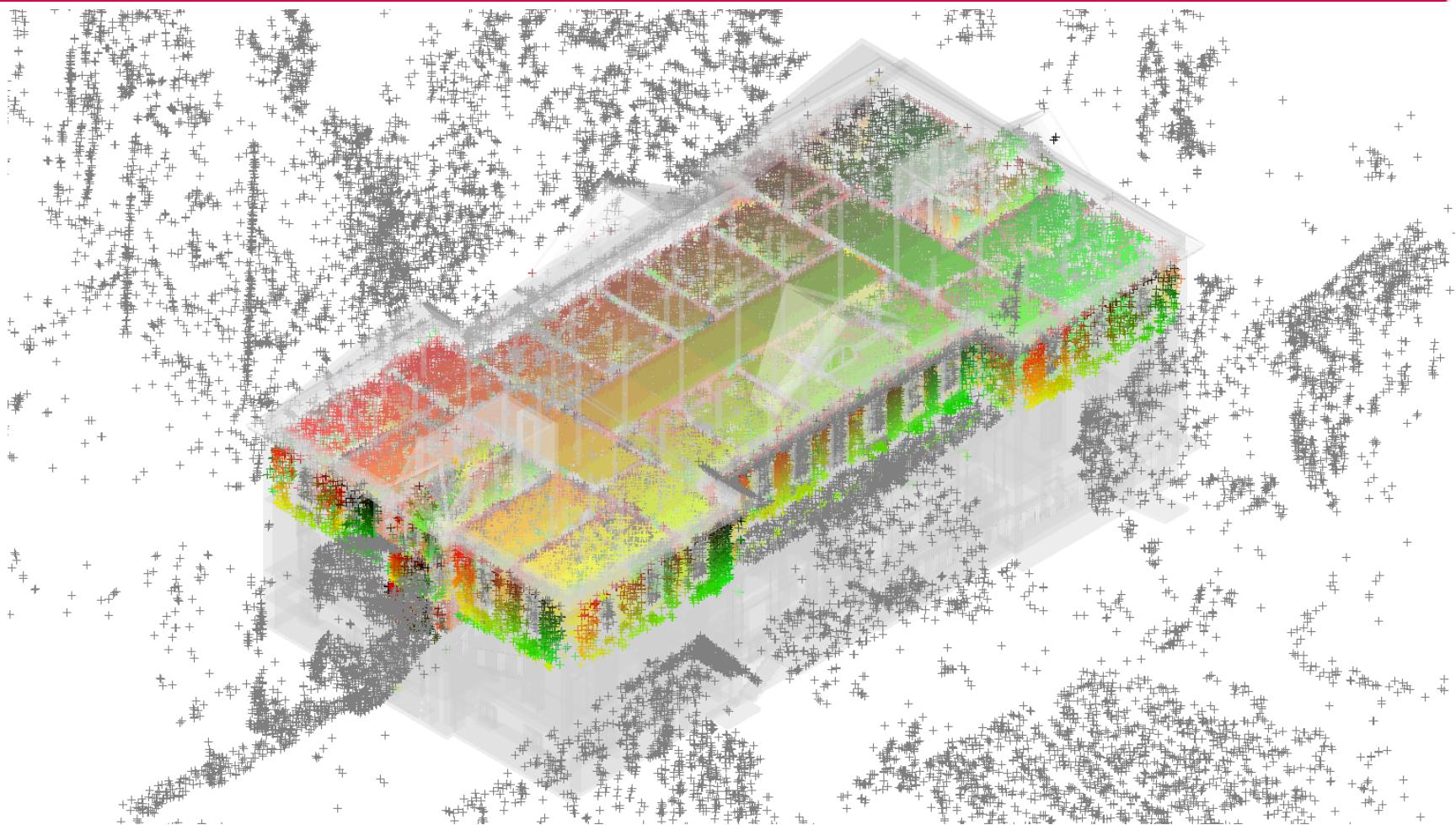
- IfcOpenShell,
- h5py,
- Point Cloud Library (PCL),
- e57lib,
- NumPy,
- Scipy and
- OpenCascade

# Results

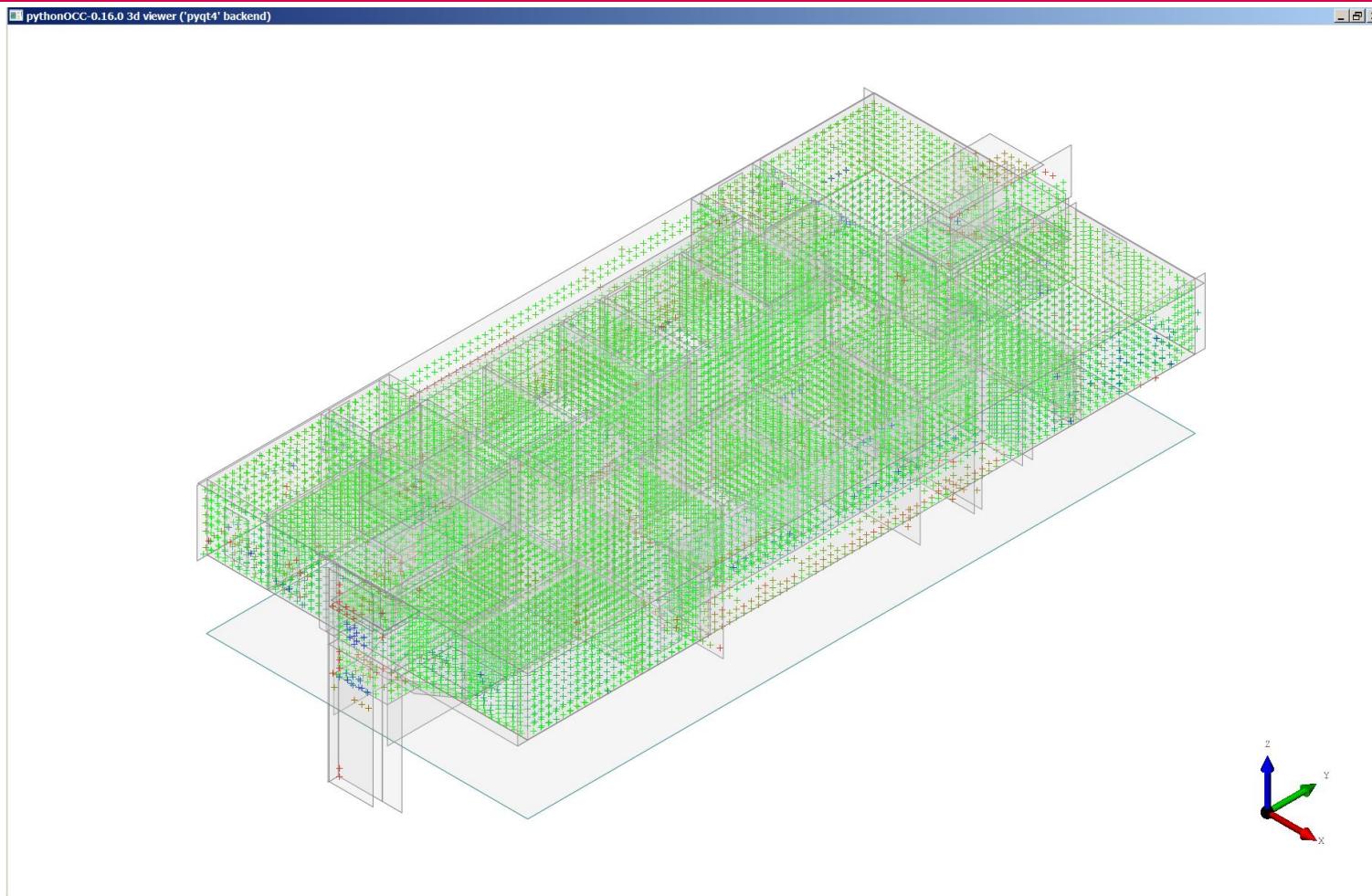
The dataset for this experiment consists of a professionally modelled **IFC file** and subsampled point cloud scan of roughly **15 million points**.



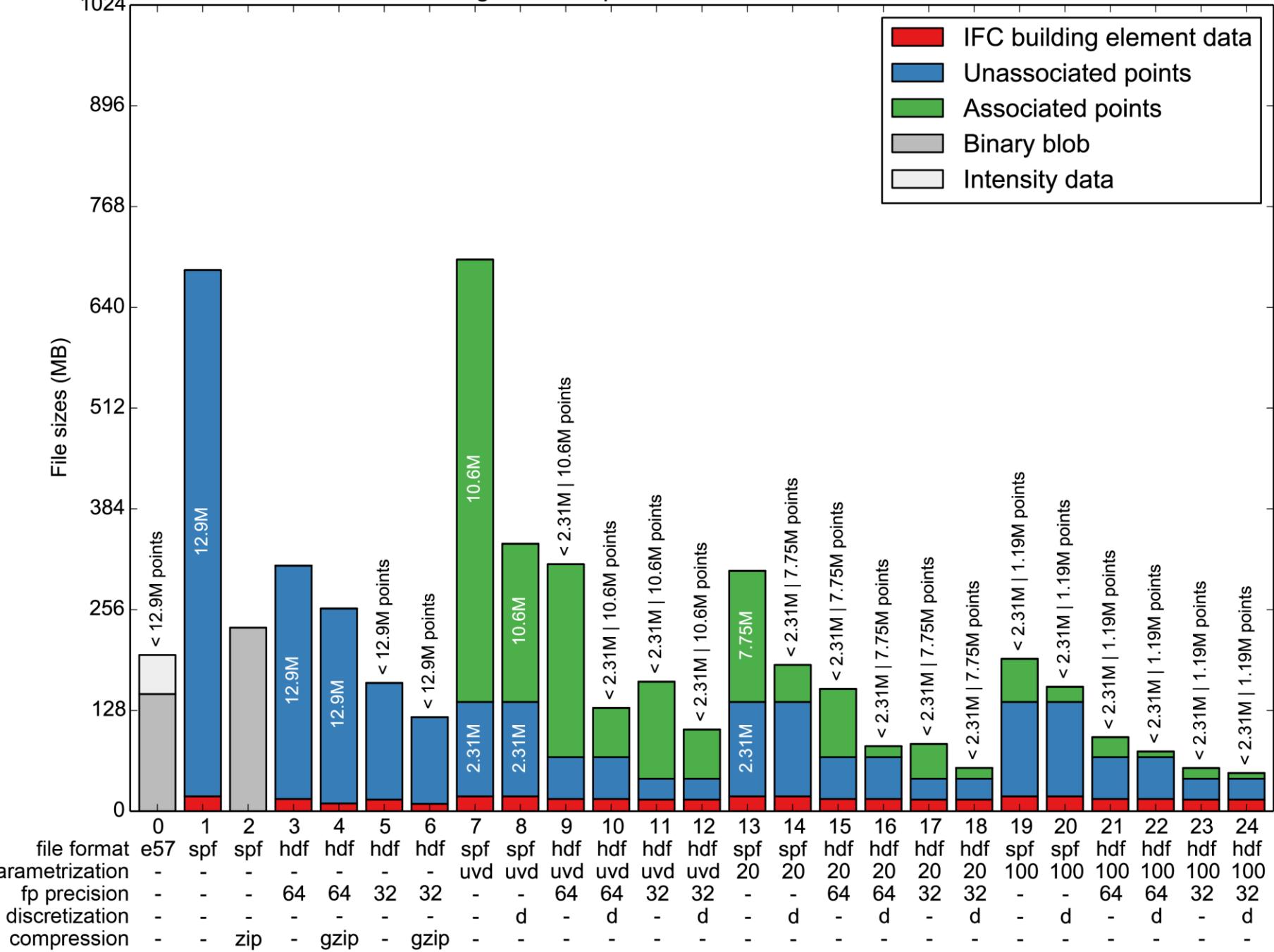
# Results



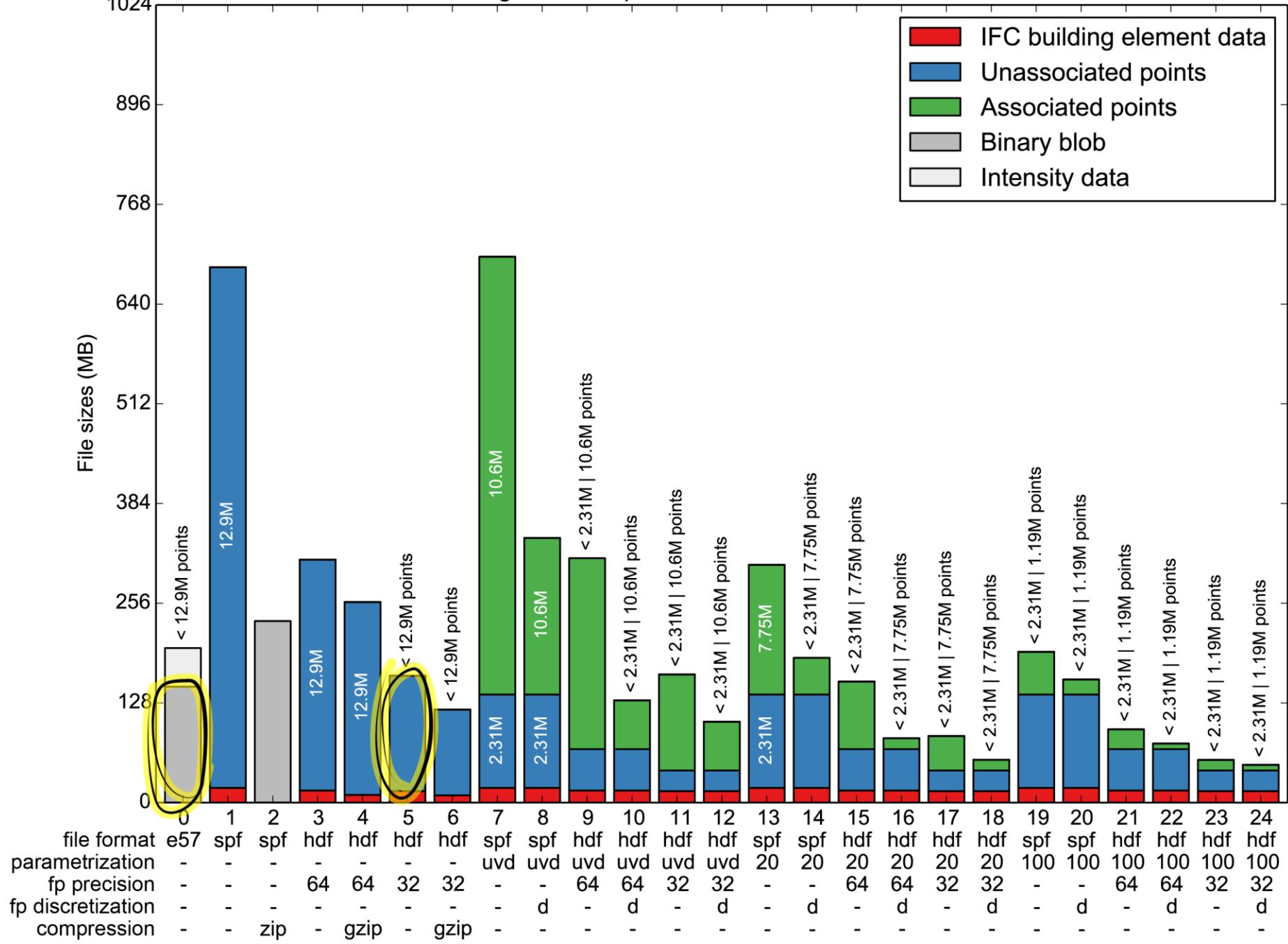
# Results



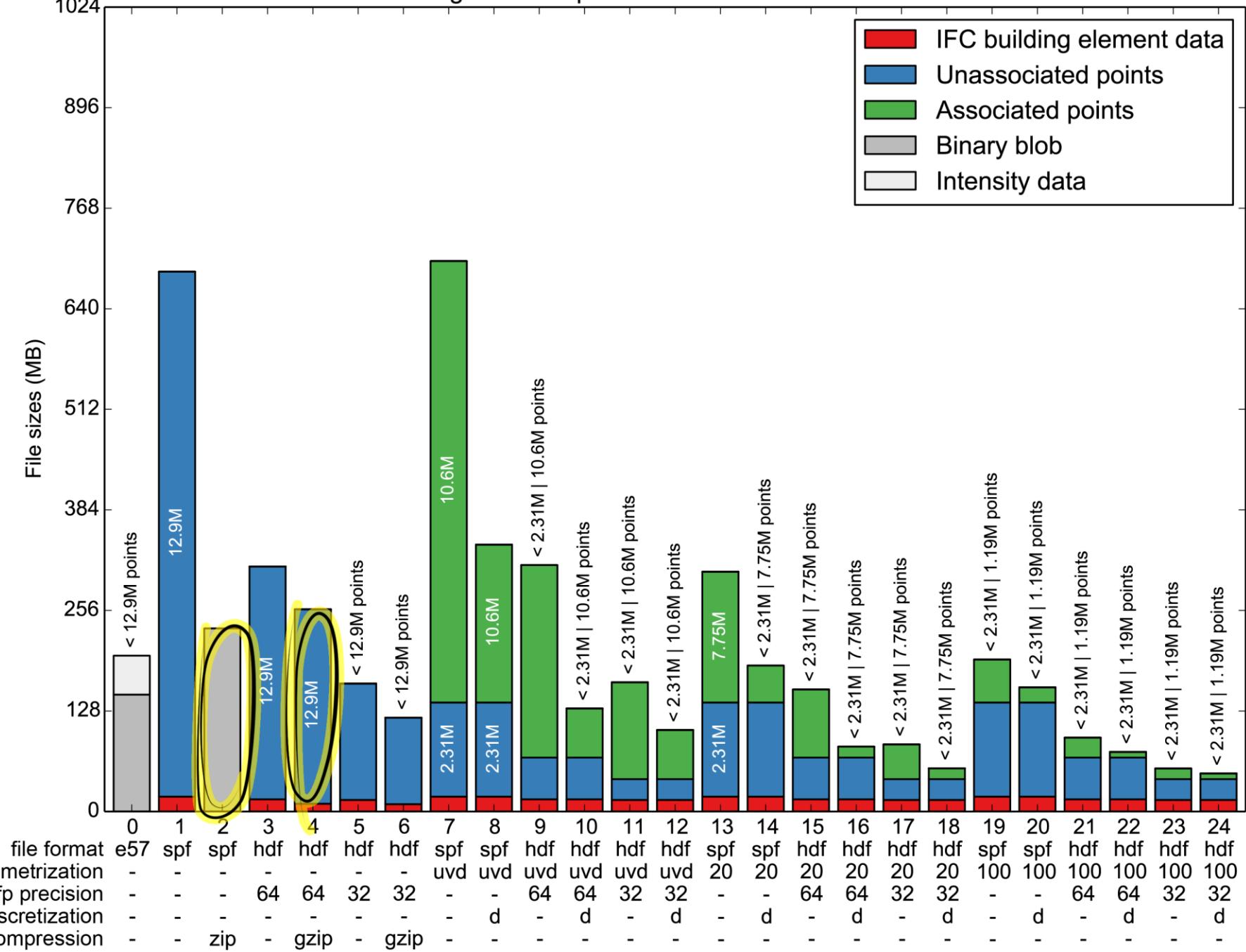
IFC building data and point clouds serialized as SPF and HDF



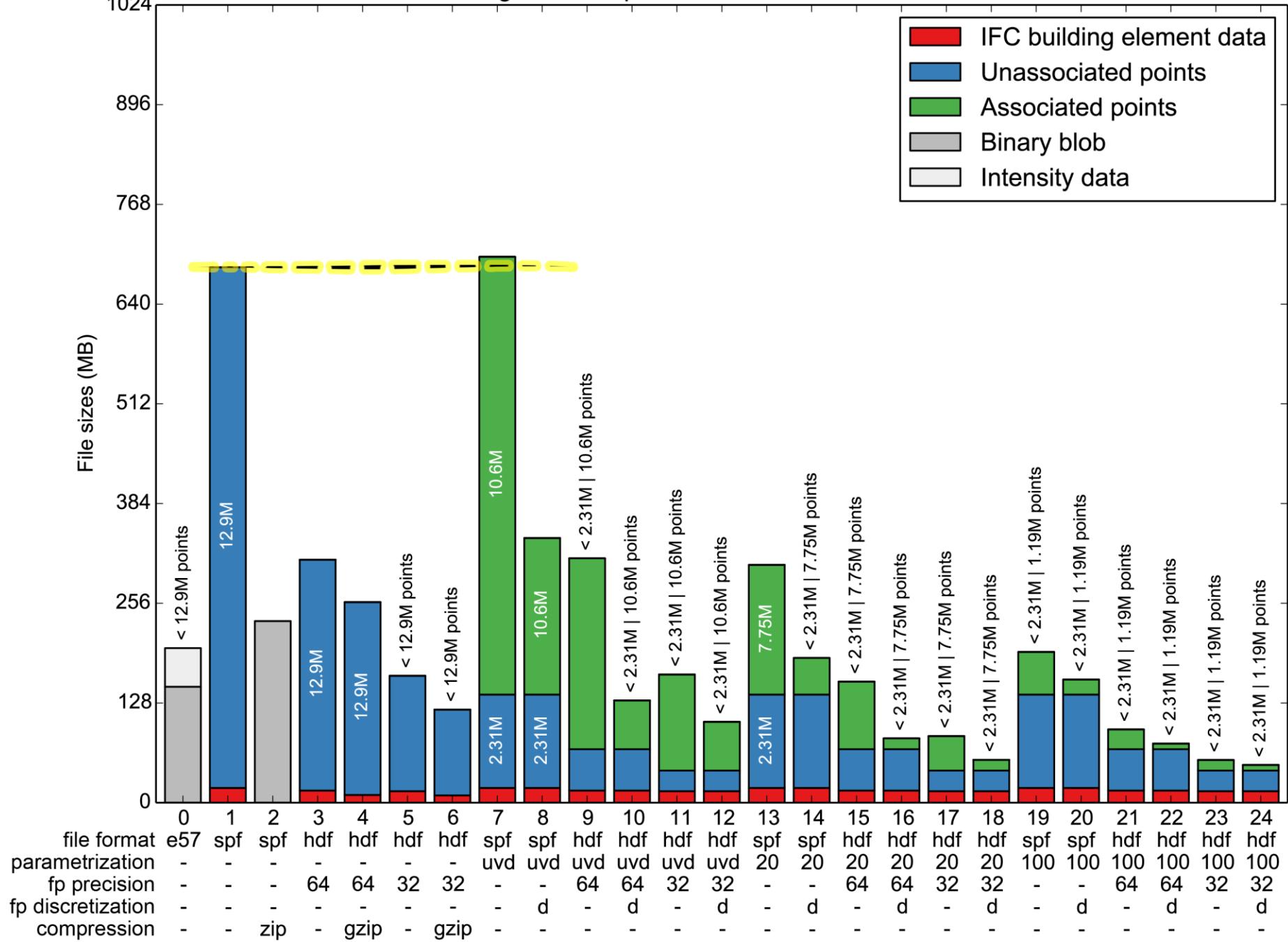
IFC building data and point clouds serialized as SPF and HDF



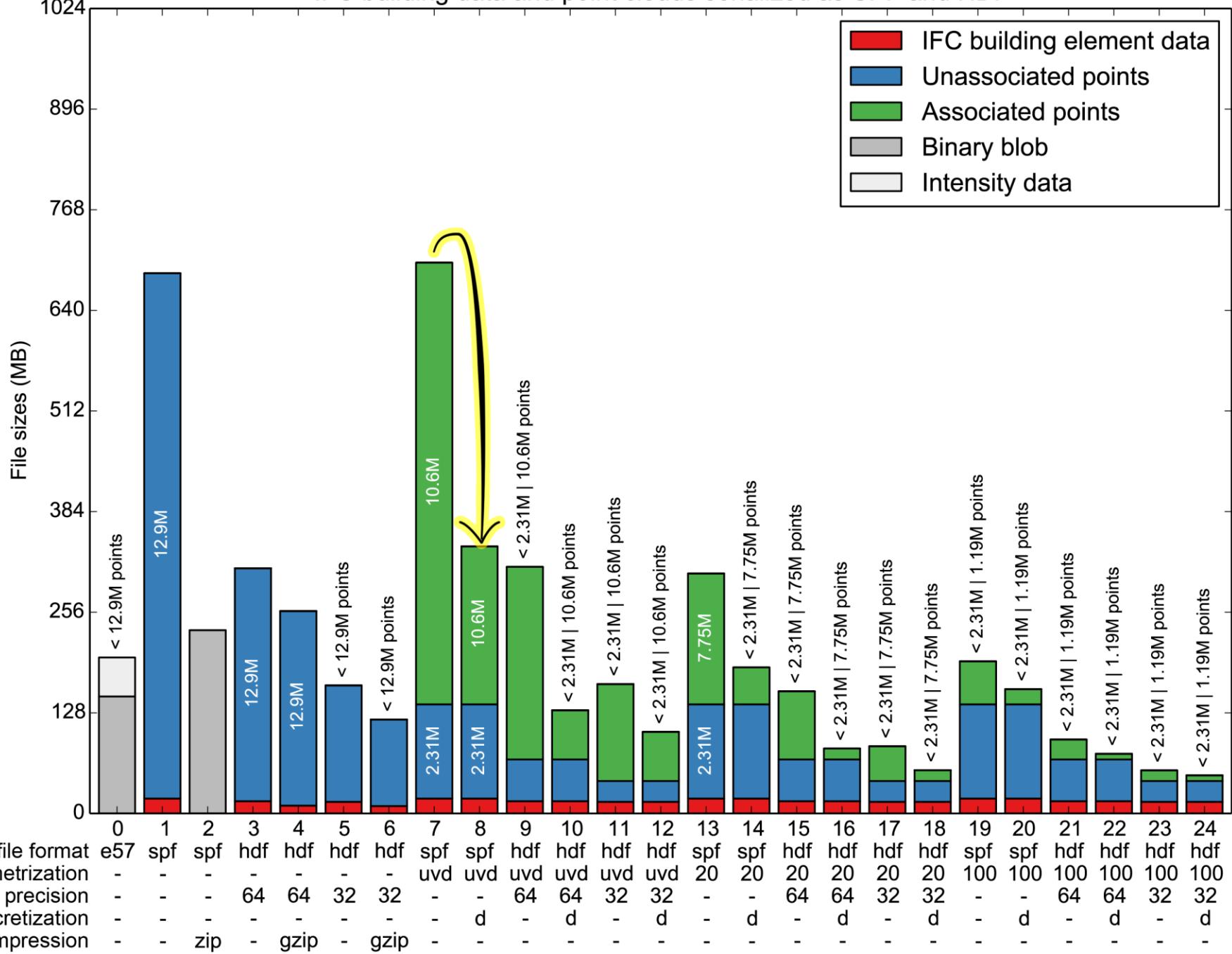
# IFC building data and point clouds serialized as SPF and HDF



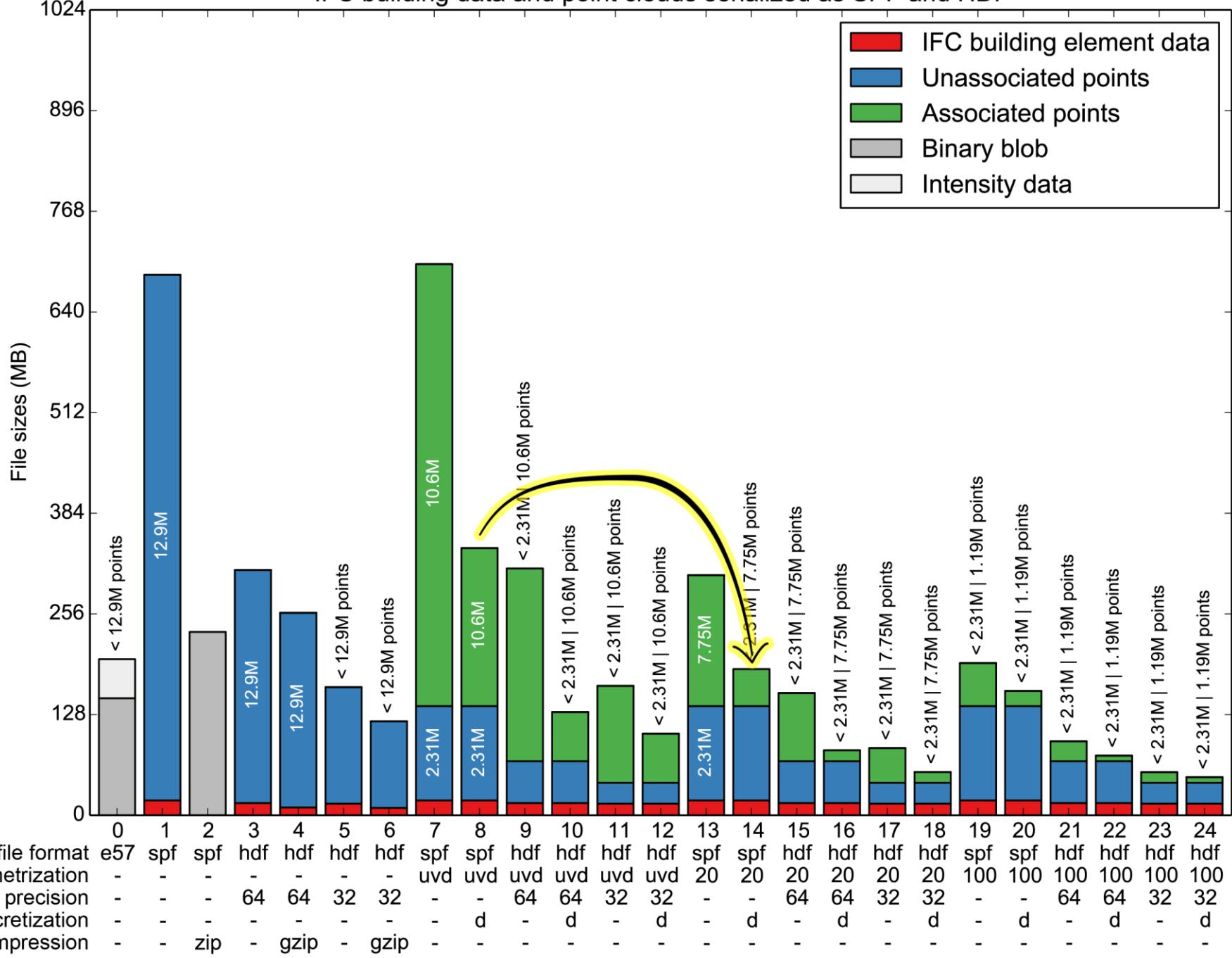
IFC building data and point clouds serialized as SPF and HDF



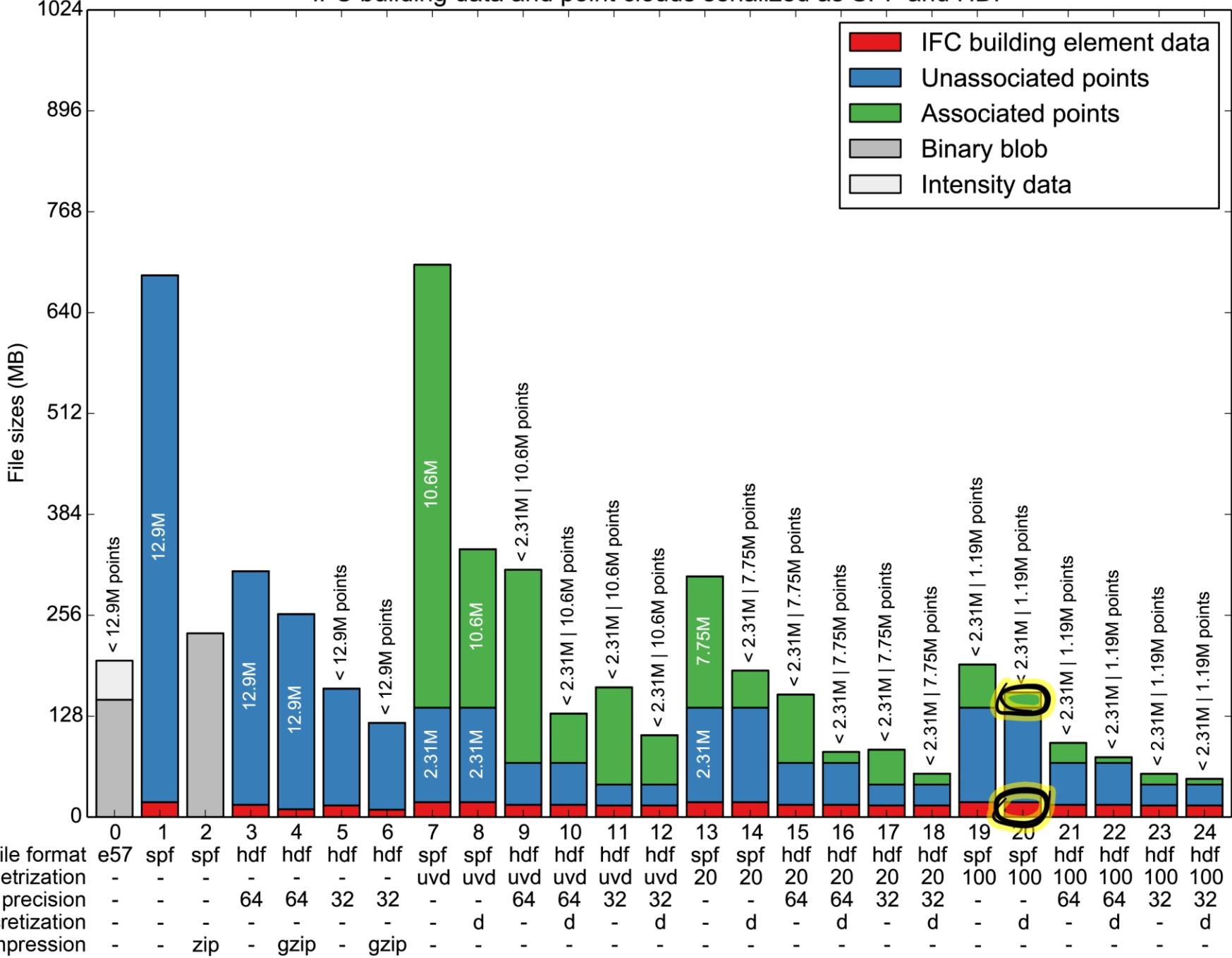
IFC building data and point clouds serialized as SPF and HDF



IFC building data and point clouds serialized as SPF and HDF



# IFC building data and point clouds serialized as SPF and HDF



# IFC and point clouds anno 2014

## Points for improvement:

No per-point attributes, such as colors, etc.

No level of detail

No explicit means for decomposition

No metadata, such as scanner model, etc.

No way to extract localized subsets

Slow to parse and leading to exorbitant data sizes

# IFC and point clouds anno 2015

## Points of improvement:

No per-point attributes, such as colors, etc.

No level of detail

No explicit means for decomposition

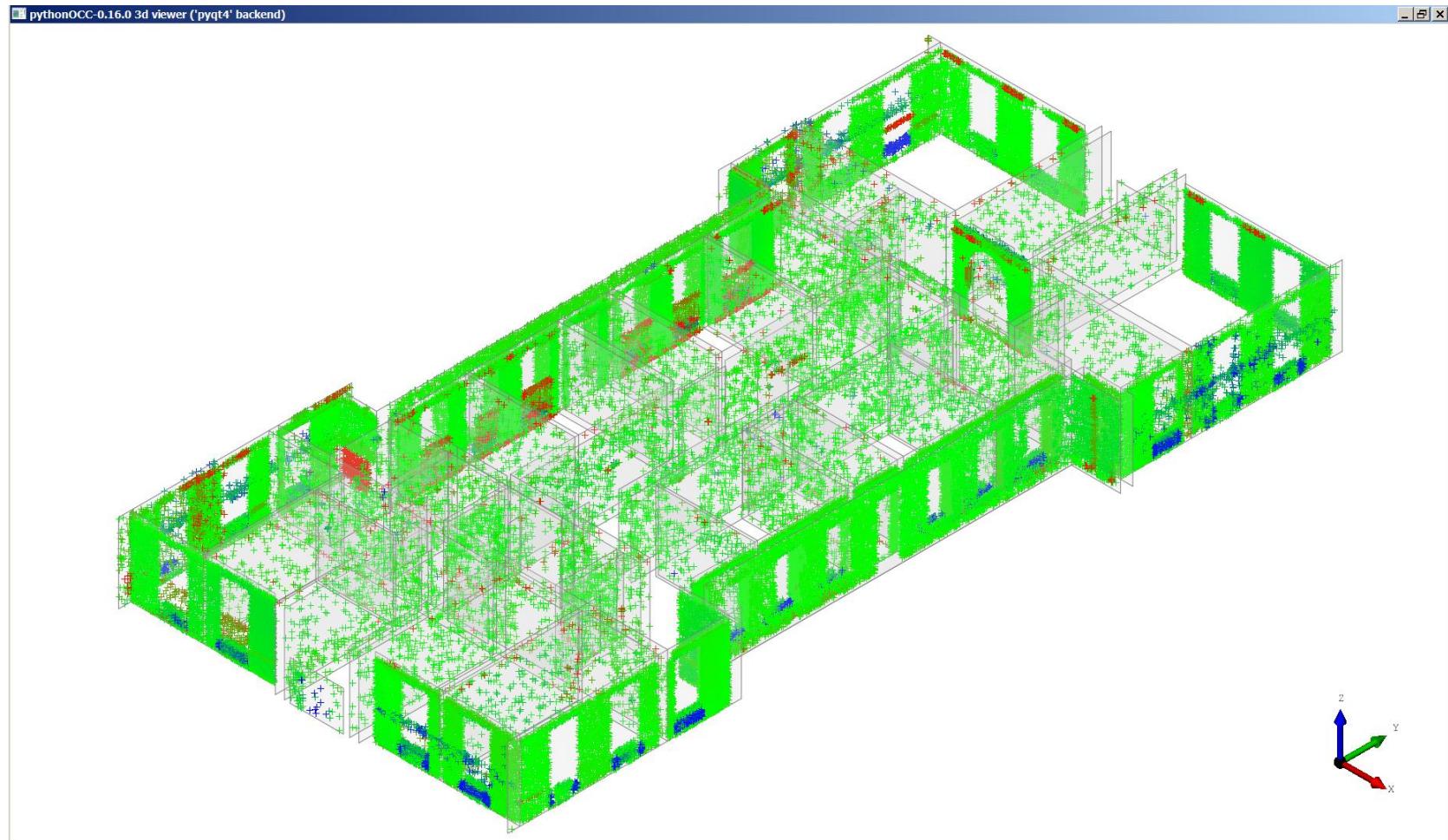
No metadata, such as scanner model, etc.

No way to extract localized subsets

Slow to parse and leading to exorbitant data sizes

Efficient to parse and file sizes under industry standard point cloud formats.

# IFC and point clouds anno 2015



# IFC and point clouds anno 2015

<https://github.com/DesignSystemsEindhoven/IFCPointCloud>

# Extending IFC with point cloud data

## eg-ice 2015

Thomas Krijnen,  
Jakob Beetz

Eindhoven University  
of Technology,  
The Netherlands

Sebastian Ochmann,  
Richard Vock,  
**Raoul Wessel**  
Institute of Computer  
Science II,  
University of Bonn,  
Germany



**TU/e**

Technische Universiteit  
**Eindhoven**  
University of Technology

Where innovation starts