Multi-disciplinary geometry (libraries) in BIM and the IfcOpenShell software library

Thomas Krijnen
Buildings as decompositions of solid volume products

source: (Pauwels, Krijnen, Terkaj & Beetz 2016)
IfcOpenShell: contributors

IfcOpenShell Contributions Over Time by Author

BlenderBIM Add-on
IfcOpenShell: language
The aim: automation on BIM

Derive higher level geometric concepts in BIM

ex: an efficient manifold representation of the facade

A digital twin that can be updated over the operation phase of a building

ex: add a wall that splits a space in two, or the reverse
Multi-disciplinary challenges

Geometric analysis
Solar potential
Building code compliance
Zoning regulations

Efficient visualization

Light weight models for mutation

Simulations
thermal
acoustics
light

Multi-disciplinary challenges

Heat dissipation among spaces with thermal interfaces
Multi-disciplinary challenges

Observable content in geospatial information

Nagel, Stadler, and Kolbe (2009)
Disjoint geometries
Aim: Manifold exterior shell of the building facade
A problem

Data comes from heterogenous sources, cannot rely on a single authoring tool to construct such secondary representations
Aim: Interior space with interior elements imprinted
A problem: geometric imprecision
A problem:

Computational challenges

- ‘Fuzzy’ boolean operations have limited robustness
- ‘Exact’ boolean operations do not ‘fuse’ disjoint elements
Earlier attempt

Donkers, Ledoux, Zhao and Stoter (2013)

Figure 3.9: Visualisation of the exterior shell extraction with closing
Decompose into - and align - halfspaces

\[ a = ax + by + cz + d \]
\[ b = \ldots \]
\[ \ldots \]
\[ S = a \cup b \cup c \cup d \cup e \cup f \]

CGAL - halfspace boolean ops
IfcOpenShell - ifc parsing and mapping to polyhedra
python - glue language
numpy - vector math
scipy - kdtree for spatial lookup
igraph - connected components
C++ - main implementation language

Decompose into - and align - halfspaces

\[ a = ax + by + cz + d \]
\[ b = \ldots \]
\[ \ldots \]
\[ S = a \cup b \cup c \cup d \cup e \cup f \]

CGAL - halfspace boolean ops
IfcOpenShell - ifc parsing and mapping to polyhedra
python - glue language
numpy - vector math
scipy - kdtree for spatial lookup
igraph - connected components
C++ - main implementation language

The result

From spaces to zones using halfspaces

- wall
- space
- opposite halfspace within solid
- boundary halfspace of solid
- opposite halfspaces touching

CGAL – halfspace boolean ops
IfcOpenShell – ifc parsing and mapping to polyhedra
rdflib – RDF querying
python – glue language
C++ – main implementation language
From spaces to zones using halfspaces

```sparql
  ?elem a <http://example.org/classes/Element> .
  ?elem a <http://example.org/classes/ifcType> "IfcWall" .
  ?elem a <http://example.org/classes/LoadBearing> FALSE .
  ?p2 a <http://example.org/classes/touches> ?r .
  ?sp1 a <http://example.org/classes/ifcType> "IfcSpace" .
  ?sp2 a <http://example.org/classes/ifcType> "IfcSpace" .
  ?sp1 a <http://example.org/classes/boundedBy> ?q .
  ?sp2 a <http://example.org/classes/boundedBy> ?r .
  ?q a <http://example.org/classes/hasEquation> ?eq1 .
  ?r a <http://example.org/classes/hasEquation> ?eq2 .
  filter(?p1 != ?p2)
} .
```

Embed runtime-queryable semantic filters
From spaces to zones using halfspaces

Individual spaces

Aggregated interior spaces joined across non-loadbearing partitions
Voxels evacuation analysis

https://github.com/opensourceBIM/voxelization_toolkit

<table>
<thead>
<tr>
<th>C++</th>
<th>- main implementation language</th>
</tr>
</thead>
<tbody>
<tr>
<td>python</td>
<td>- data visualization</td>
</tr>
</tbody>
</table>
Voxels headroom analysis

https://github.com/opensourceBIM/voxelization_toolkit
voxel analysis implementation language

```python
file = parse("*.ifc")
all_surfaces = create_geometry(file, exclude={"IfcSpace", "IfcOpeningElement", ...})
voxels = voxelize(all_surfaces)

stairs = create_geometry(file, include={"IfcStair"})
stair_ids_region = voxelize(stairs, type="uint", method="surface")
stair_ids_empty = constant_like(voxels, 0, type="uint")
stair_ids = union(stair_ids_region, stair_ids_empty)
stair_ids_offset = shift(stair_ids, dx=0, dy=0, dz=1)

stair_voxels_region = voxelize(stairs)
stair_voxels_empty = constant_like(voxels, 0)
stair_voxels = union(stair_voxels_region, stair_voxels_empty)

railings = create_geometry(file, include={"IfcRailing"}, optional=1)
railing_voxels_orig = voxelize(railings)
railing_voxels_down = sweep(railing_voxels_orig, dx=0.0, dy=0.0, dz=-1.0)
stair_voxels_wo_railing = subtract(stair_voxels, railing_voxels_orig)

stair_offset = shift(stair_voxels_wo_railing, dx=0, dy=0, dz=1)
stair_offset_min_1 = subtract(stair_offset, stair_voxels_wo_railing)
stair_offset_min = subtract(stair_offset_min_1, railing_voxels_down)
extrusion = sweep(stair_voxels_wo_railing, dx=0.0, dy=0.0, dz=-0.4)
stair_top = subtract(stair_offset_min, extrusion)

surfaces = create_geometry(file, exclude={"IfcOpeningElement", "IfcDoor", "IfcSpace", ...})
...
voxelization

scalar/vector fields

robust and trivial boolean operations

efficient and trivial distance calculation on superimposed grids

closes gaps due to modelling issues, precision issues or intentional gaps
IfcOpenShell: architecture v0.7

Share partly the same code but different translation unit
CGAL

Predominantly only polyhedra
Exact rational number type
Machine native interval for performance, fallback to exact when uncertain
Friendly documentation, but chaotic packages
Rather academic, not a focus on CAD

Open CASCADE

BRep data model
Cryptic, but consistent API and data model
Tolerance and fuzziness
Many CAD operations implemented
Multi-disciplinary geometry
(libraries) in BIM and the
IfcOpenShell software library

Thomas Krijnen