

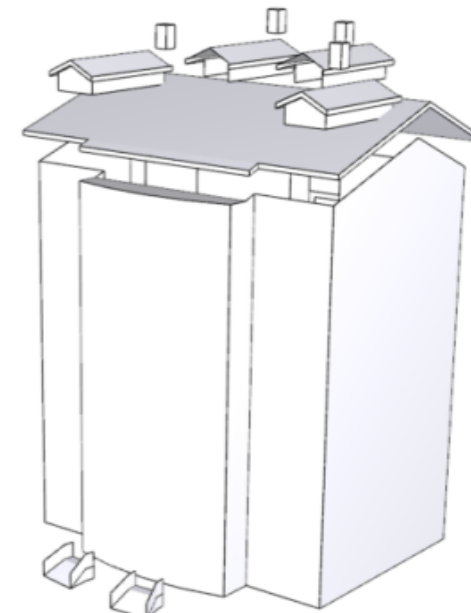
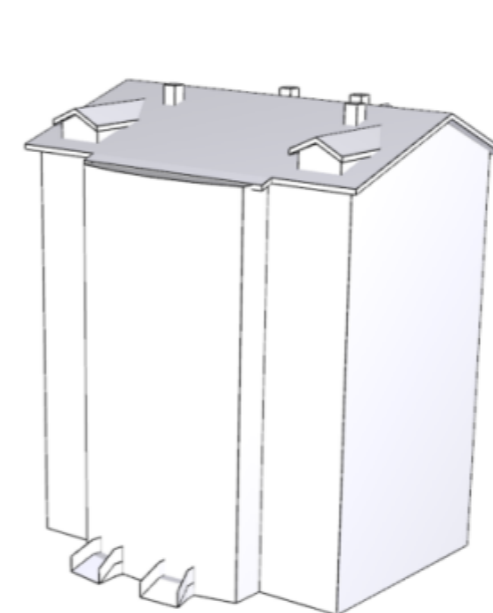
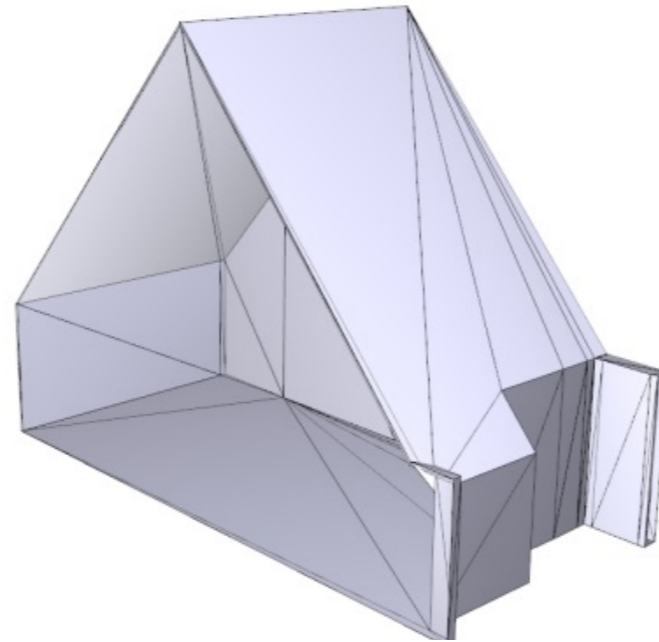
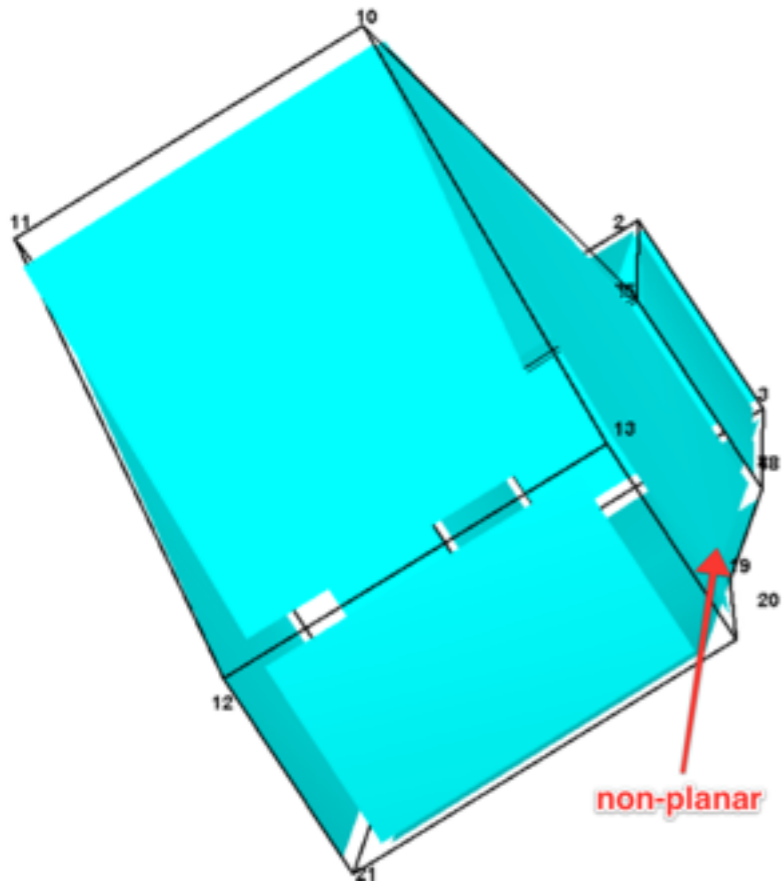
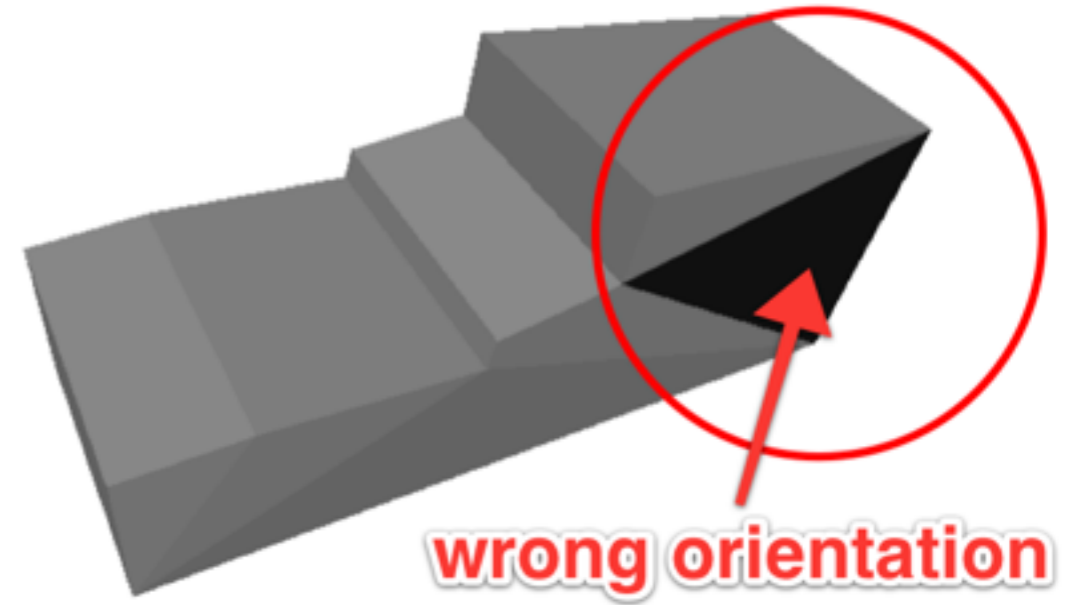
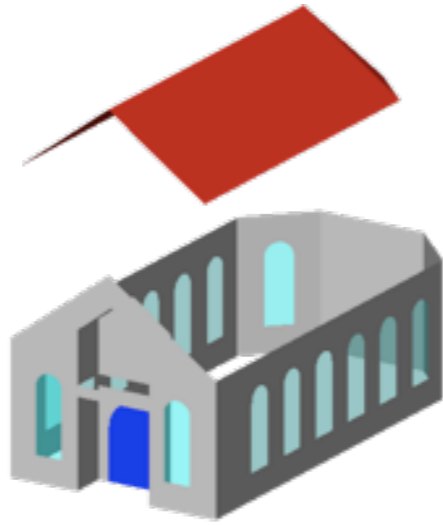
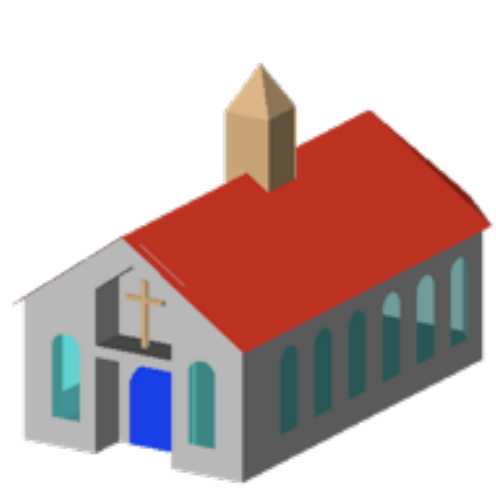
Validation and automatic repair of 3D geometries

Hugo Ledoux

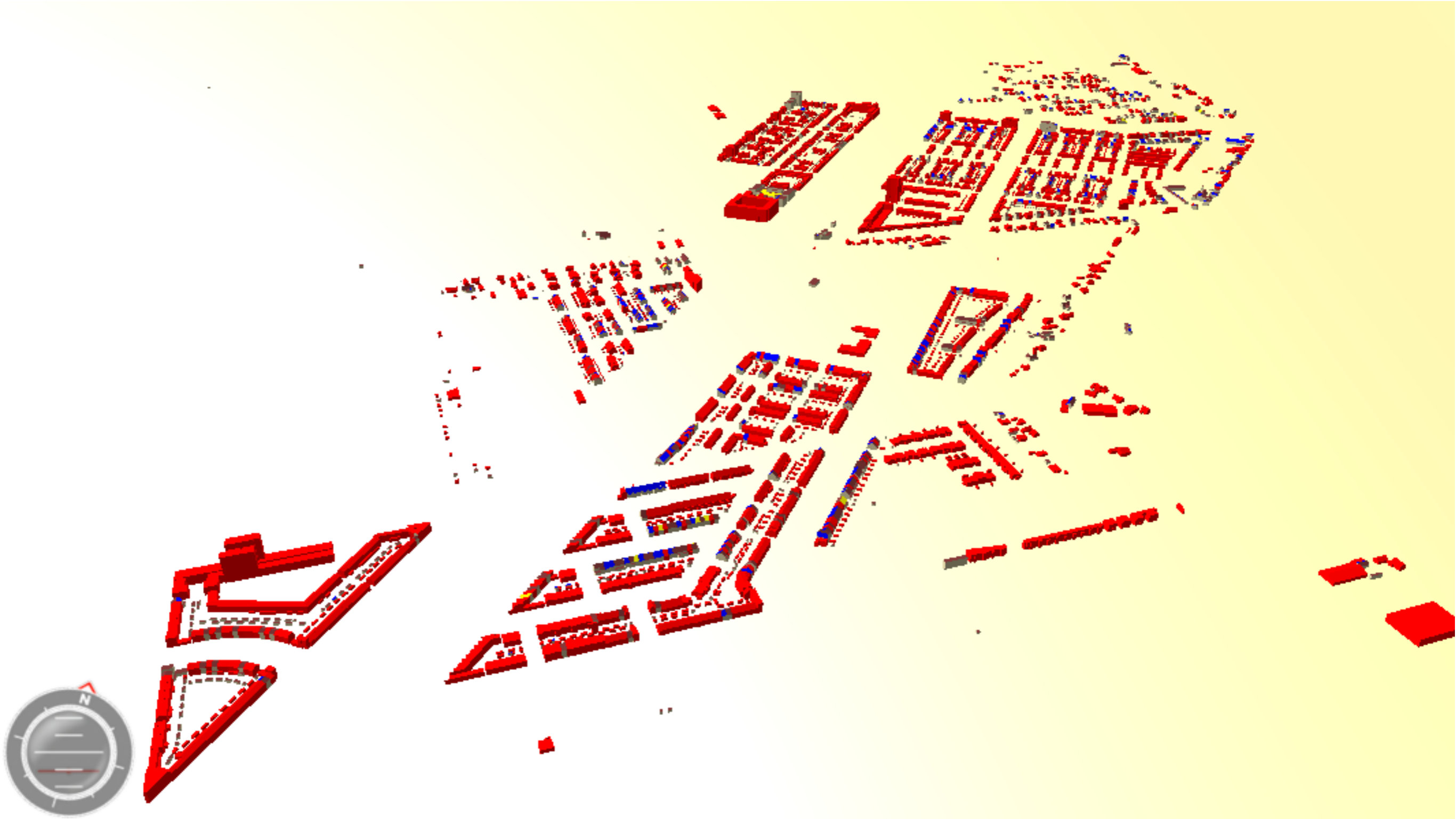
2015-02-06

ITC students – “3D visit”

errors = very common in 3D buildings



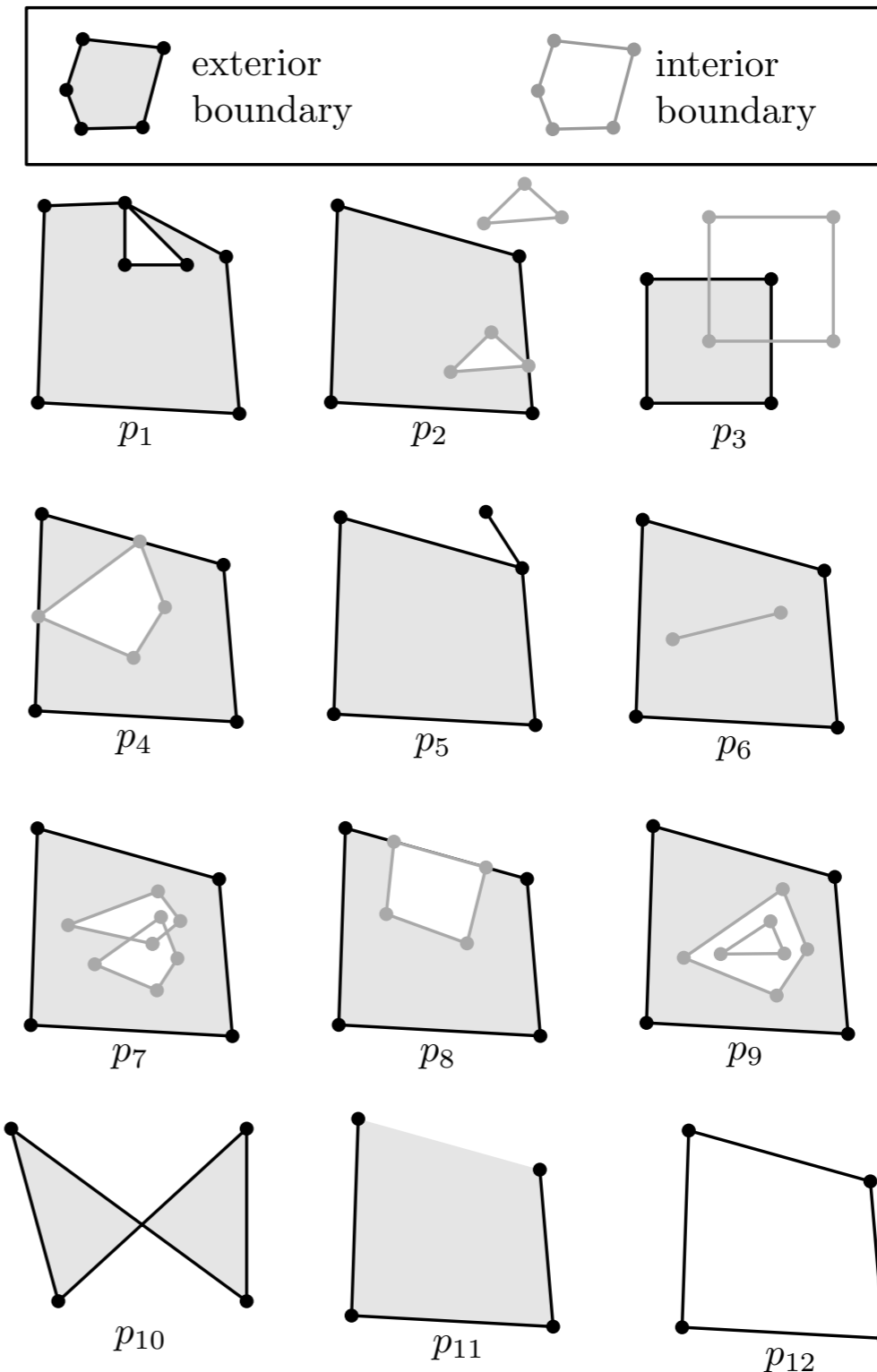
Rotterdam is a nice example... (>80% invalid)



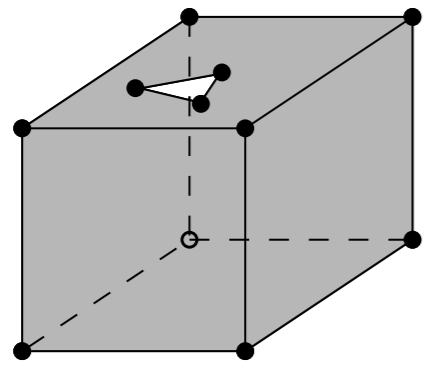
rules for validation of 2D polygons

OGC Simple Features and ISO19107 rules:

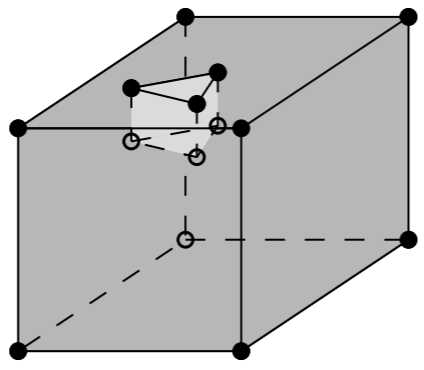
- 1 no self-intersection
- 2 closed boundaries
- 3 rings can touch but not overlap
- 4 no duplicate points
- 5 no dangling edges
- 6 connected interior
- 7 etc



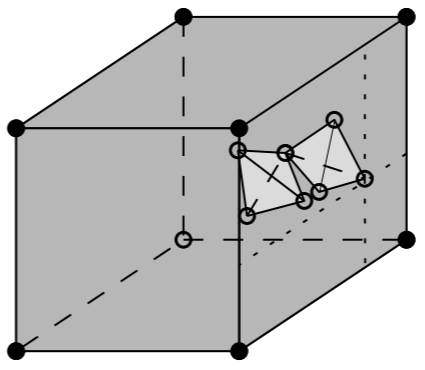
ISO19107 = also in 3D



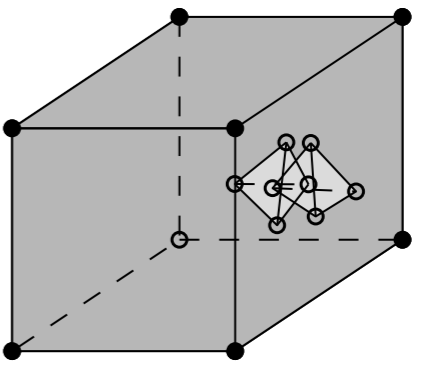
S₁
invalid



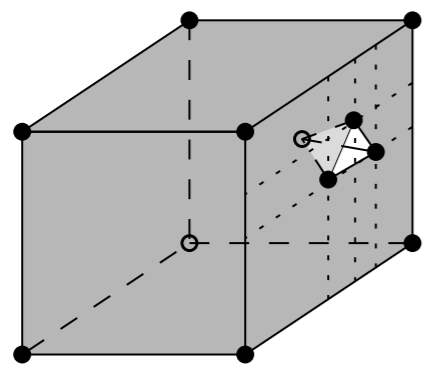
S₂
valid



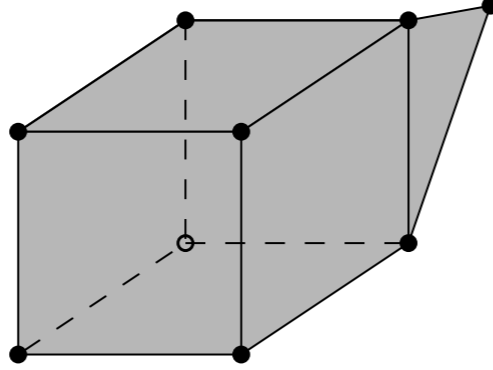
S₃
valid



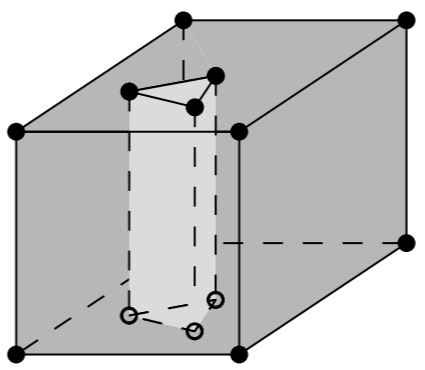
S₄
invalid



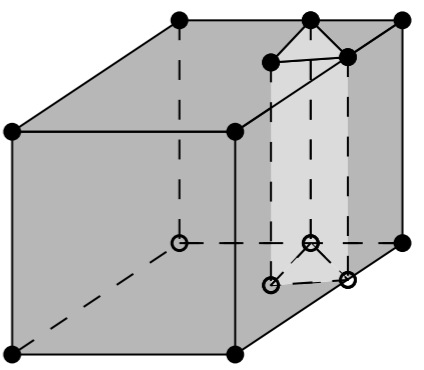
S₅
invalid



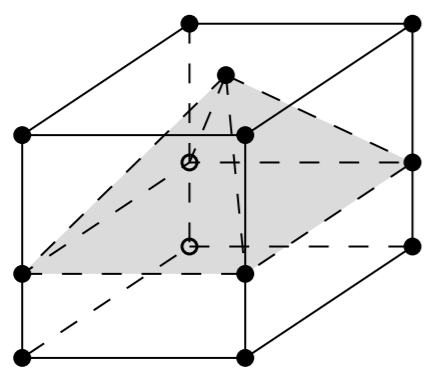
S₆
invalid



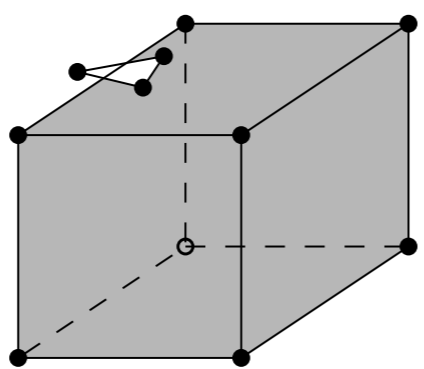
S₇
valid



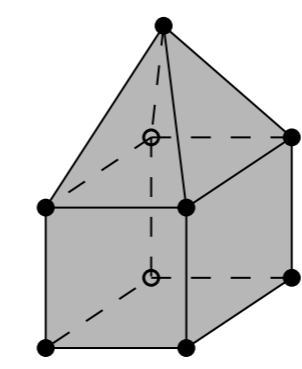
S₈
invalid



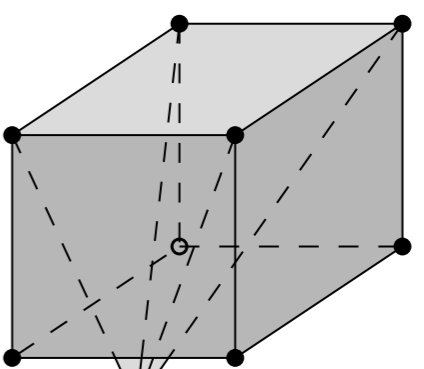
S₉
invalid



S₁₀
invalid



S₁₁
valid



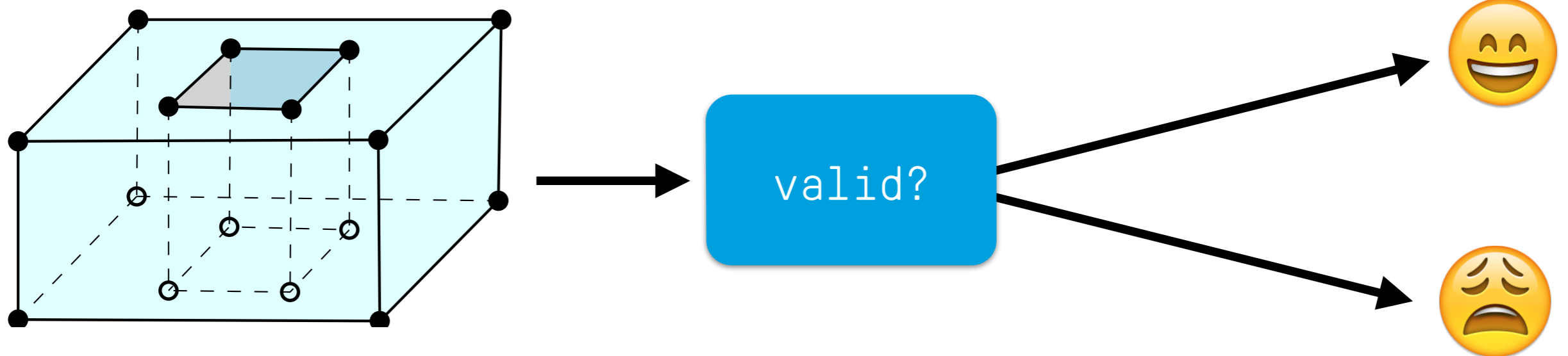
S₁₂
invalid

ISO19107 = also in 3D

- 1 distinct vertex
- 2 closedness of the rings of every surface
- 3 orientation of points within a surface (with inner rings)
- 4 planarity of surfaces
- 5 non-self intersection of surfaces
- 6 non-overlapping inner rings on a surface
- 7 orientation of normal vectors
- 8 “watertightness” of every shell
- 9 “connectedness” of the interior
- 10 how inner/outer shells interact with each others
- 11 ...

my geometric validation software: val3dity

- as ISO 19107 as possible (only linear/planar primitives)
- use of CGAL: robust and fast
- C++
- kind to the user



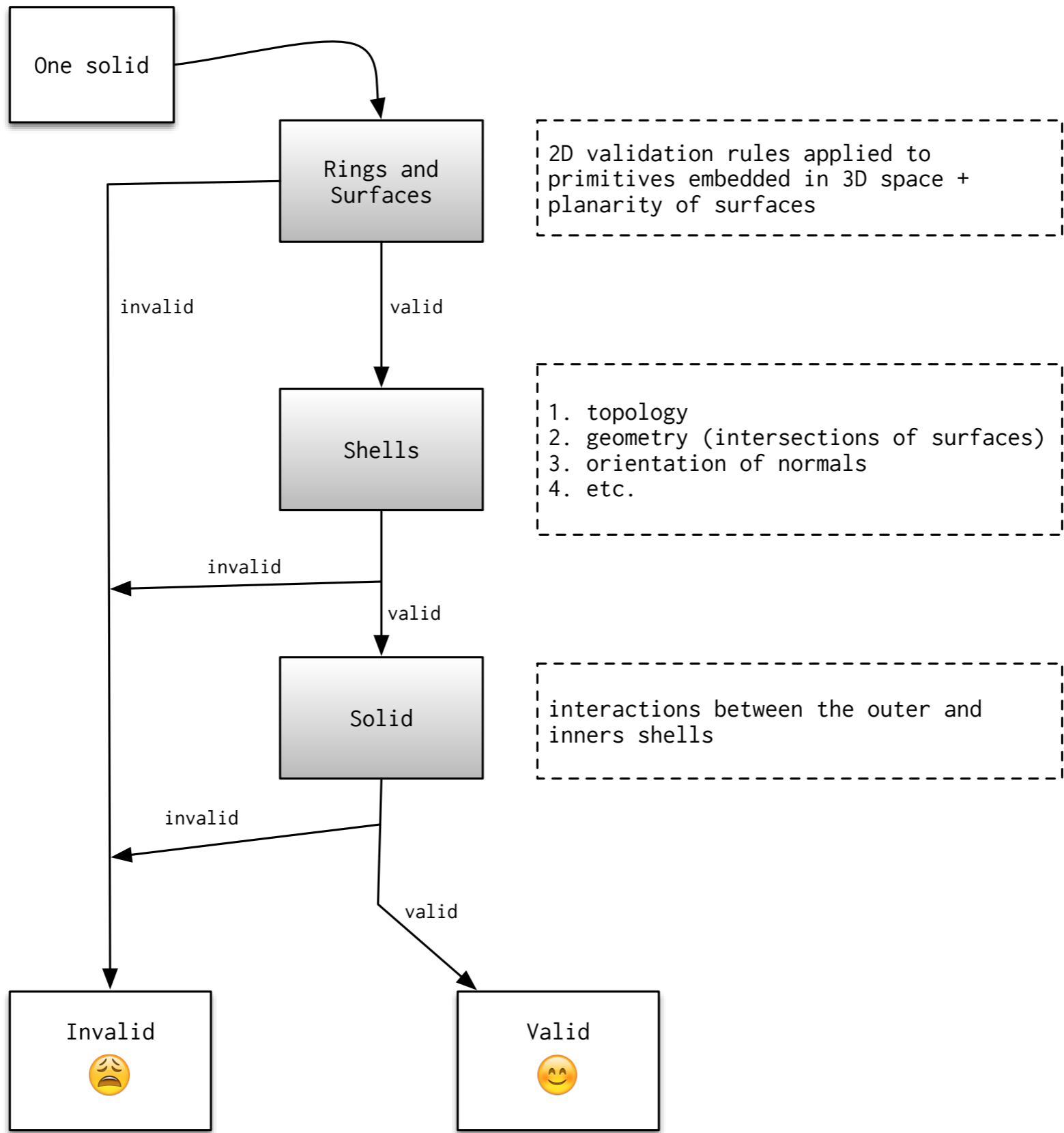
≠ schema validation (*.xsd)

```
<gml:Solid>
  <gml:exterior>
    <gml:CompositeSurface>
      <gml:surfaceMember>
        <!--top surface-->
          <gml:Polygon gml:id="a">
            <gml:exterior>
              <gml:LinearRing>
                <gml:pos>0 0 1</gml:pos>
                <gml:pos>1 0 1</gml:pos>
                ...
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </gml:surfaceMember>
      </gml:CompositeSurface>
    </gml:exterior>
  </gml:Solid>
```

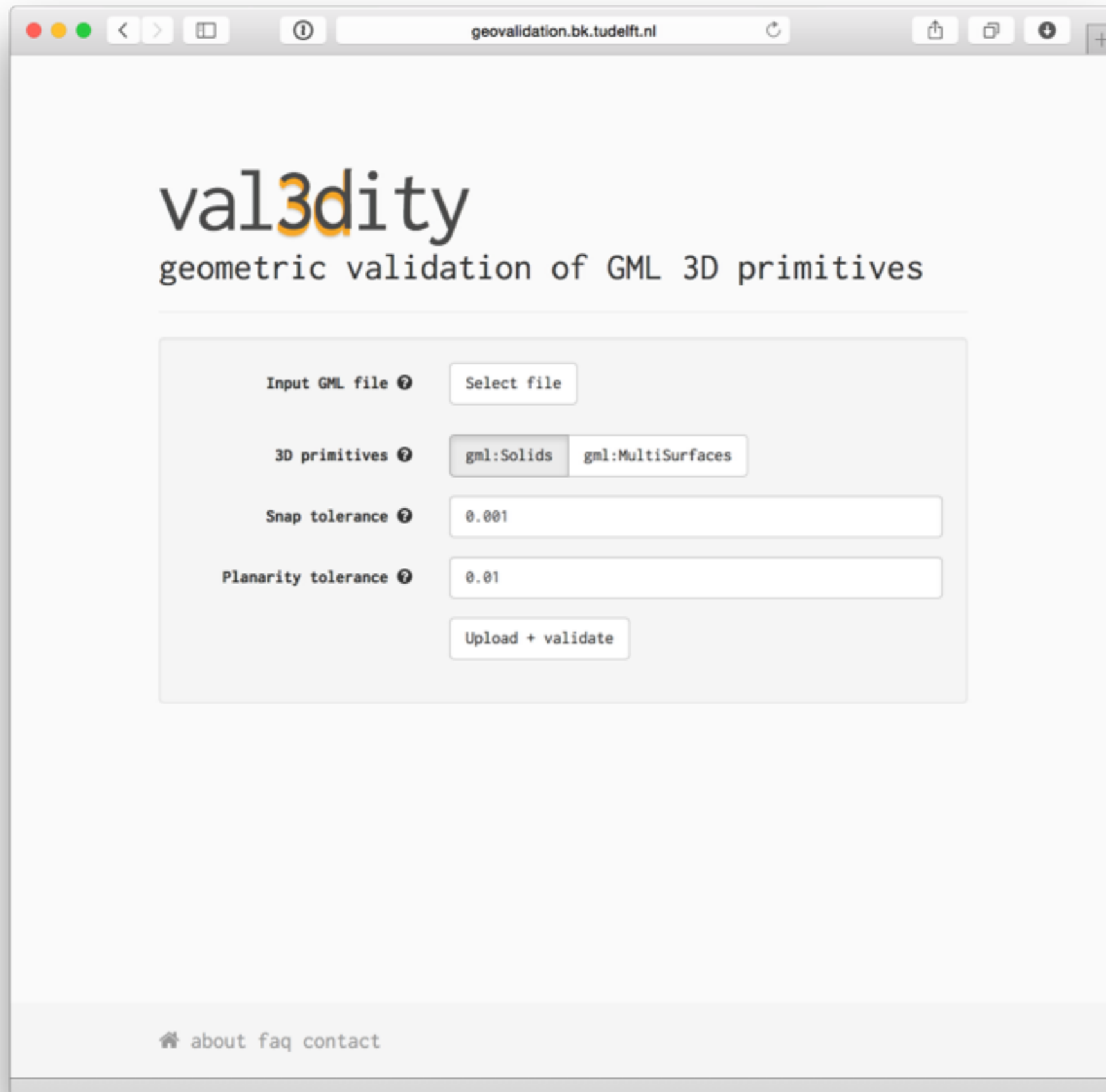
only syntax is checked, eg 3
numbers separated by a space

...

hierarchical validation



val3dity is free & open-source



The screenshot shows a web browser window with the URL `geovalidation.bk.tudelft.nl`. The page features the **val3dity** logo and the subtitle "geometric validation of GML 3D primitives". Below this is a form with the following fields:

- Input GML file**: A button labeled "Select file".
- 3D primitives**: Two radio buttons, with "gml:Solids" selected and "gml:MultiSurfaces" unselected.
- Snap tolerance**: A text input field containing the value "0.001".
- Planarity tolerance**: A text input field containing the value "0.01".

At the bottom of the form is a button labeled "Upload + validate". A footer at the bottom of the page contains a home icon and the text "about faq contact".

val3dity

geometric validation of GML 3D primitives

Input GML file ⓘ

Select file

3D primitives ⓘ

gml:Solids

gml:MultiSurfaces

Snap tolerance ⓘ

0.001

Planarity tolerance ⓘ

0.01

Upload + validate

outputs a report with the errors

```
<val3dity>
  <inputFile>delft.gml</inputFile>
  <snaptolerance>0.001</snaptolerance>
  <time>Tue Apr 22 12:07:11 2014</time>
  <Solid>
    <id>6e359e22-e6d7-41d1-ba8c-91e0068704f7</id>
    <ValidatorMessage>
      <type>ERROR</type>
      <errorCode>210</errorCode>
      <errorType>NON_PLANAR_SURFACE</errorType>
      <shell>1</shell>
      <face>14</face>
    </ValidatorMessage>
  </Solid>
  <Solid>
    <id>59feffb1-604c-4032-b414-1d72f1d2371d</id>
    <ValidatorMessage>
      <type>ERROR</type>
      <errorCode>400</errorCode>
      <errorType>SHELLS_FACE_ADJACENT</errorType>
      <shell>2</shell>
      <face>3</face>
    </ValidatorMessage>
  </Solid>
</val3dity>
```

IDs of gml:Solid used



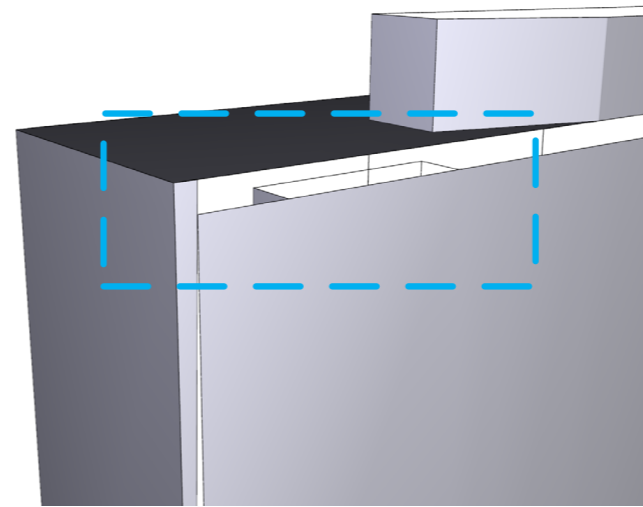
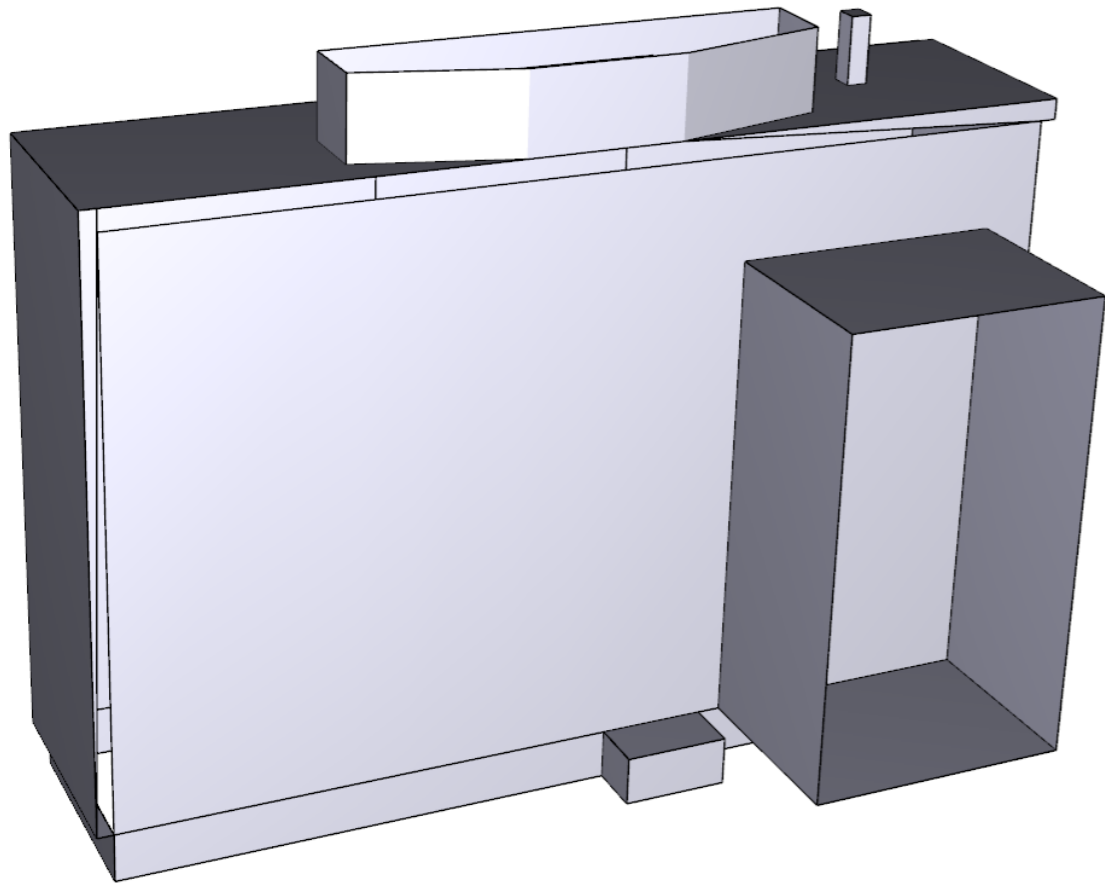
error type



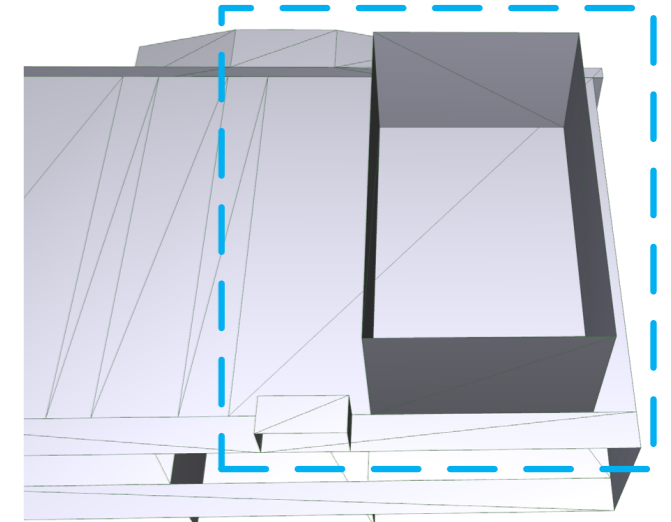
link to specific surface where
the problem is



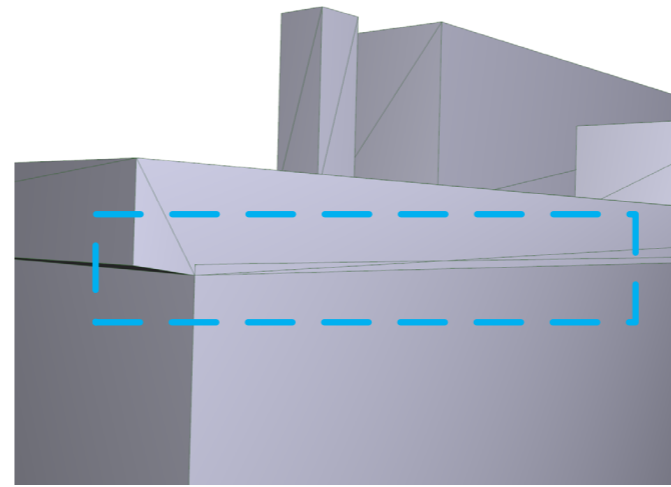
a very nasty building



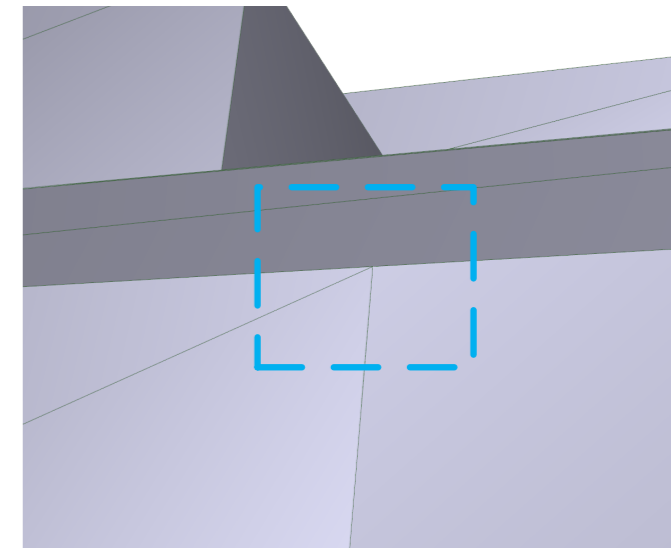
a)



b)

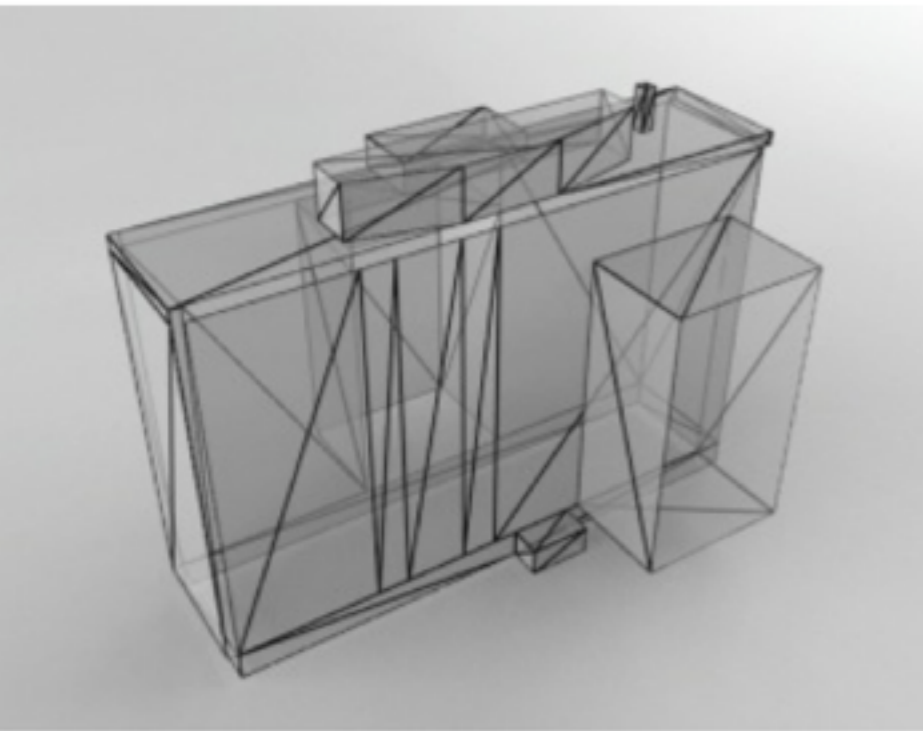


c)

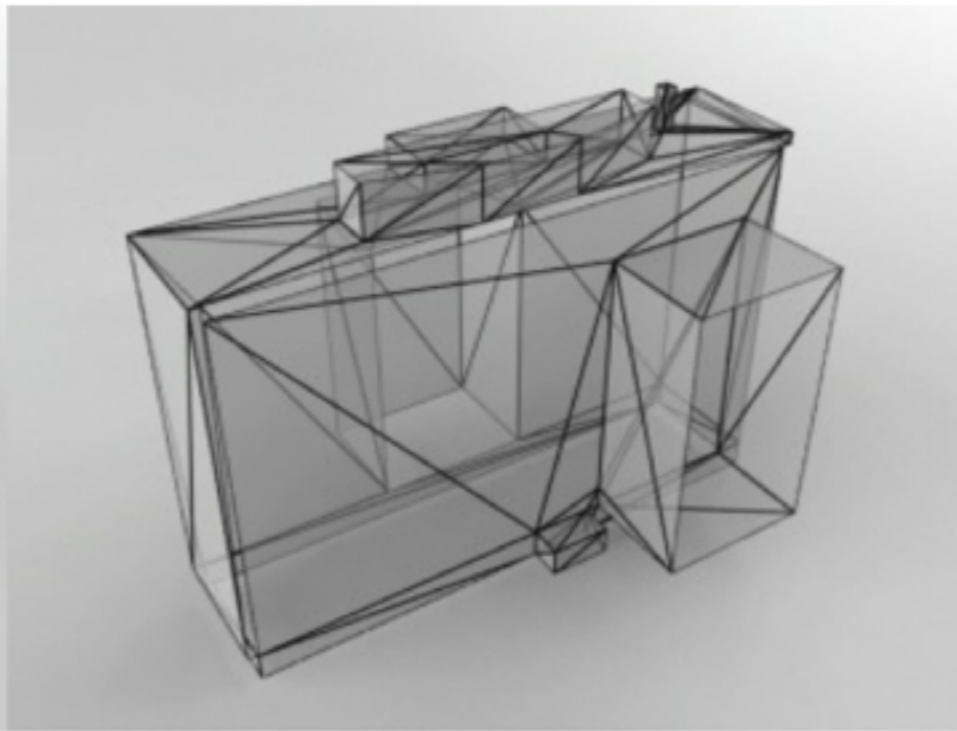


d)

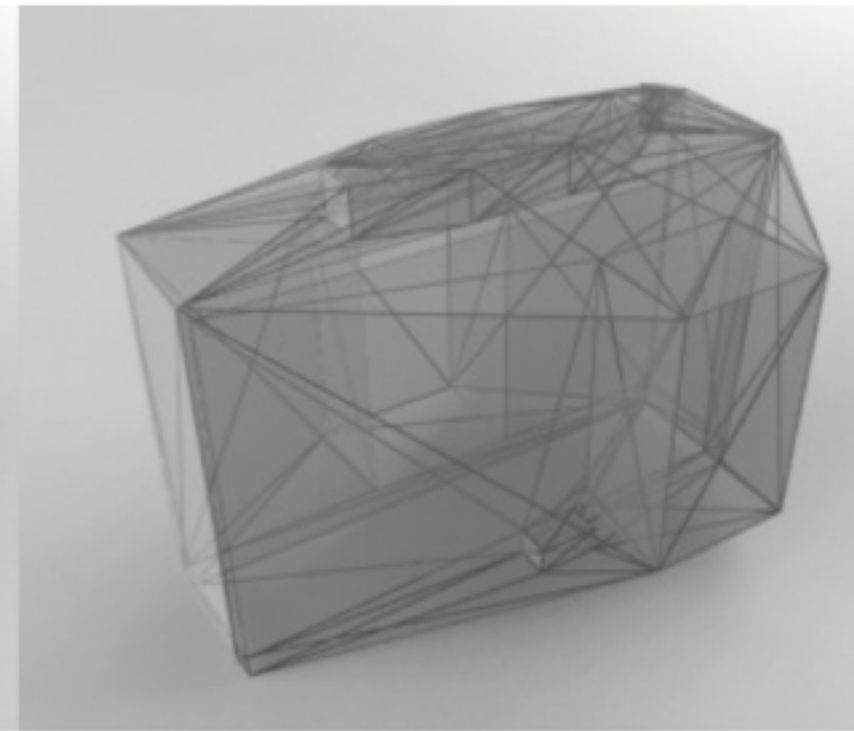
automatic repair of 3D buildings



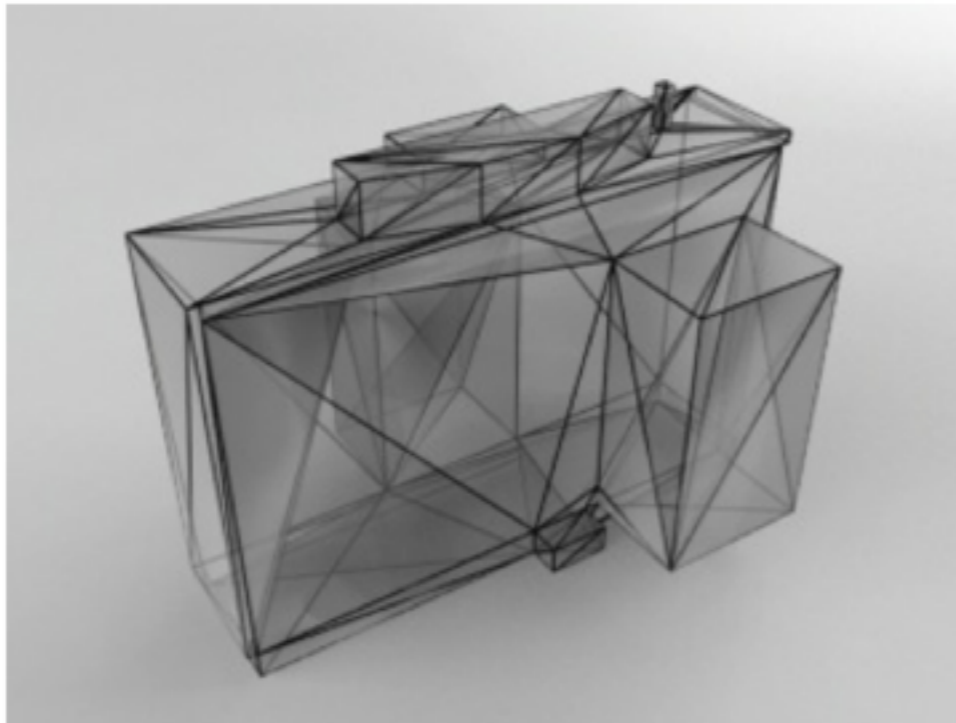
a)



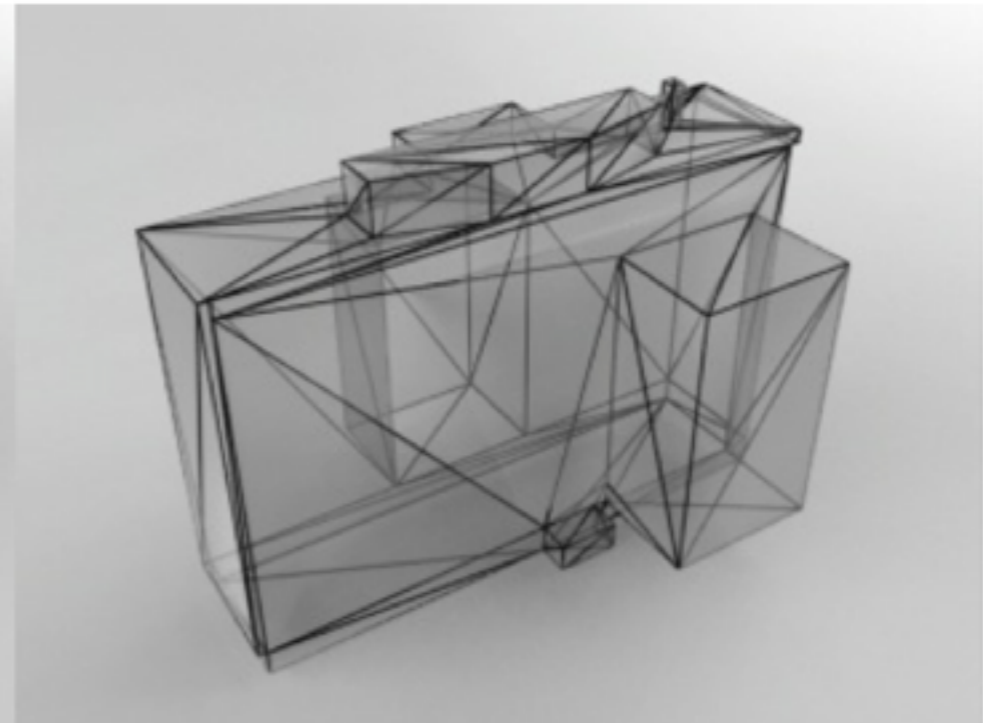
b)



c)



d)



e)

Repair using shrink-wrapping

Junqiao Zhao

Hugo Ledoux

Jantien Stoter

TU Delft 2013

thank you.

Hugo Ledoux

`h.ledoux@tudelft.nl`
`tudelft.nl/hledoux`

→ `geovalidation.bk.tudelft.nl/val3dity`

→ `github.com/tudelft3d/val3dity`