Lesson 4.1
Three-dimensional geometries in geoinformation

GEO1004:
3D modelling of the built environment

https://3d.bk.tudelft.nl/courses/geo1004

3D geoinformation
Department of Urbanism
Faculty of Architecture and the Built Environment
Delft University of Technology

About GIS standards

Standardisation organisations

- **ISO**: International Standardisation Organisation
- **OGC**: Open Geospatial Consortium
- **INSPIRE**: **In**frastructure for **S**patial **I**nfo**r**mation in **E**urope
- **NEN**: *normalisatie en normen* (“Dutch ISO”)

ISO 191xx series of geographic information standards

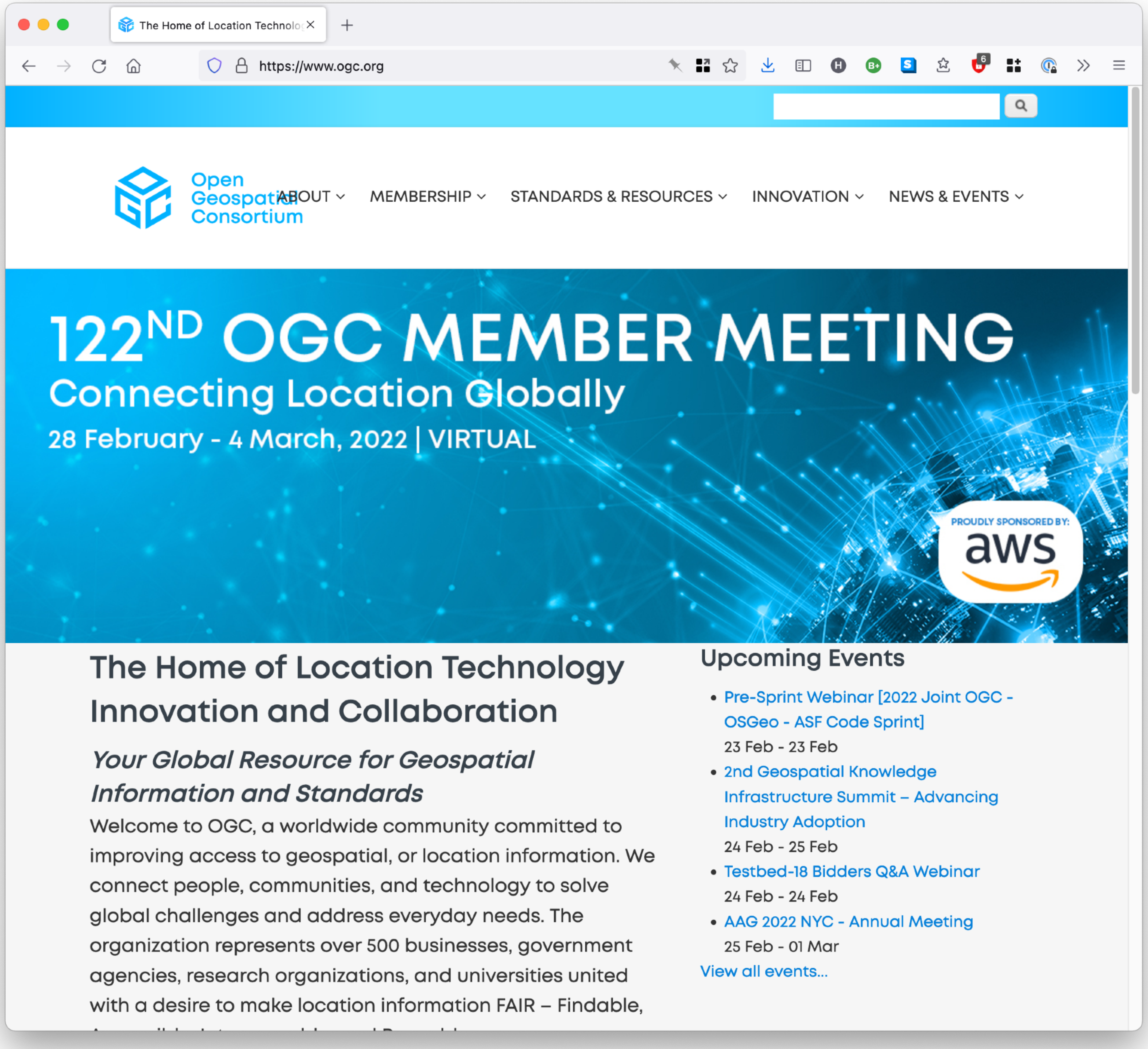
- A main goal of the ISO 191xx series is to facilitate the interoperability of geographic information systems, i.e. ability to discover, access, understand and use the information and tools, independently from the platform supporting them
- Structured set of standards specifying methods, tools and services for the management of geographic information, including:
  - Definition of data,
  - Access to data
  - Presentation of data
  - Transfer between users
- weirdly, and annoyingly, one has to pay to read these 😩 (but free if TUDelft student!)

ISO 191xx series of geographic information standards

6709 - Standard representation of latitude, longitude and altitude for geographic point locations	19122 - Qualifications and Certification of personnel
19101 - Reference model	19123 - Schema for coverage geometry and functions
19101-2 - Reference model - Part 2: Imagery	19124 - Imagery and gridded data components
19103 - Conceptual schema language	19125-1 - Simple feature access - Part 1: Common architecture
19104 - Terminology Introduction	19125-2 - Simple feature access - Part 2: SQL option
19105 - Conformance and testing	19126 - Profile - FACC Data Dictionary
19106 - Profiles	19127 - Geodetic codes and parameters
19107 - Spatial schema	19128 - Web Map server interface
19108 - Temporal schema	19129 - Imagery, gridded and coverage data framework
19109 - Rules for applicaiton schema	19130 - Sensor and data models for imagery and gridded data
19110 - Methodology for feature cataloguing	19131 - Data product specifications
19111 - Spatial referencing by coordinates	19132 - Location based services - Reference model
19112 - Spatial referencing by geographic identifiers	19133 - Location based services - Tracking and navigation
19113 - Quality principles	19134 - Multimodal location based services for routing and navigation
19114 - Quality evaluation procedures	19135 - Procedures for registration of geographical information items
19115 - Metadata	19136 - Geography Markup Language

19115-2 - Metadata - Part 2: Extensions for imagery and gridded data	19137 - Generally used profiles of the spatial schema and of similar important other schemas
19116 - Positioning services	19138 - Data quality measures
19117 - Portrayal	19139 - Metadata - Implementation specification
19118 - Encoding	19140 - Technical amendment to the ISO 191** Geographic information series of standards for harmonization and enhancements
19119 - Services	19141 - Schema for moving features
19120 - Functional standards	
19121 - Imagery and gridded data	

OGC: Open Geospatial Consortium®



122ND OGC MEMBER MEETING

Connecting Location Globally

28 February - 4 March, 2022 | VIRTUAL

The Home of Location Technology Innovation and Collaboration

Your Global Resource for Geospatial Information and Standards

Welcome to OGC, a worldwide community committed to improving access to geospatial, or location information. We connect people, communities, and technology to solve global challenges and address everyday needs. The organization represents over 500 businesses, government agencies, research organizations, and universities united with a desire to make location information FAIR – Findable,

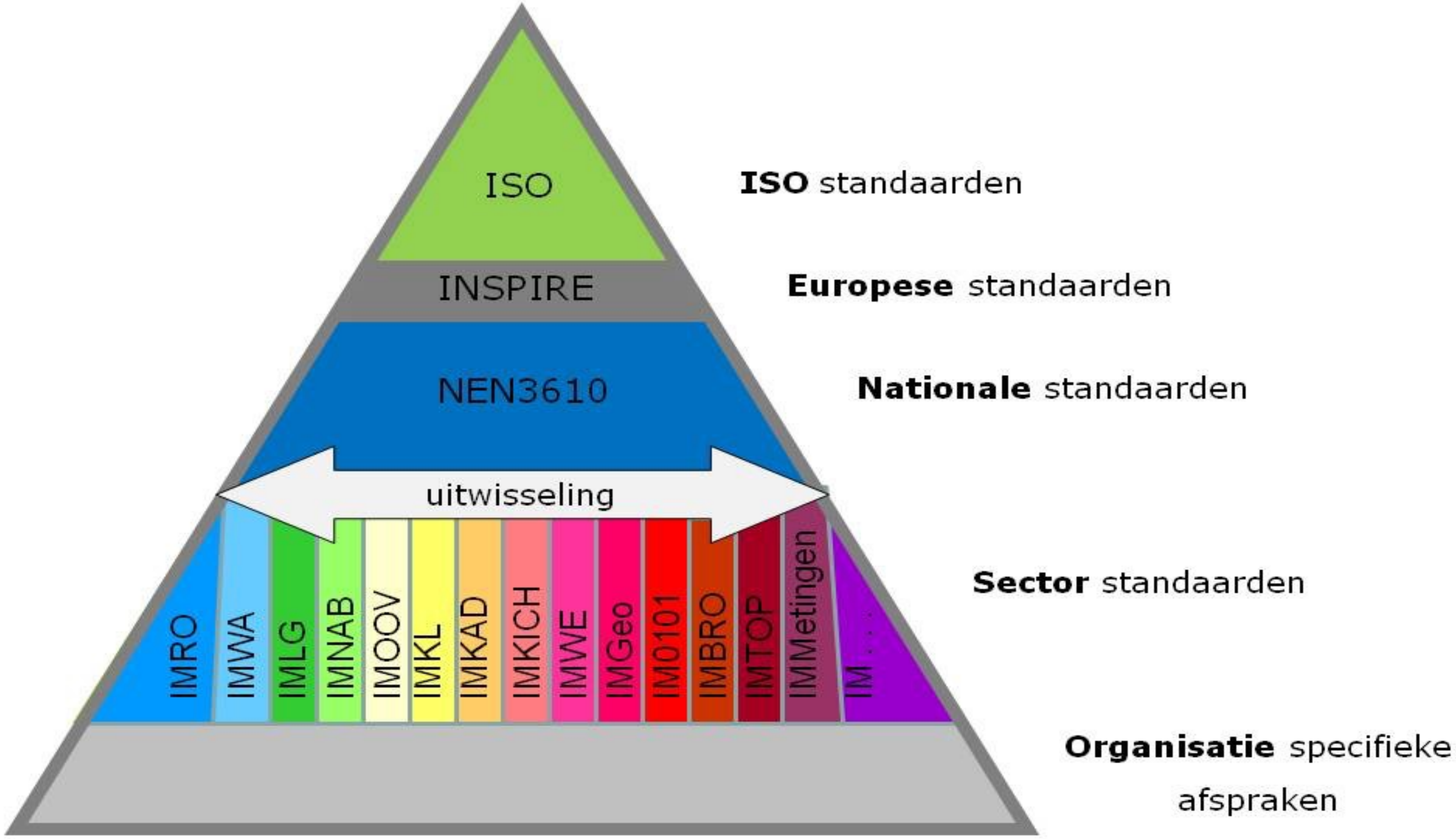
Upcoming Events

- Pre-Sprint Webinar [2022 Joint OGC - OSGeo - ASF Code Sprint]
  23 Feb - 23 Feb
- 2nd Geospatial Knowledge Infrastructure Summit – Advancing Industry Adoption
  24 Feb - 25 Feb
- Testbed-18 Bidders Q&A Webinar
  24 Feb - 24 Feb
- AAG 2022 NYC - Annual Meeting
  25 Feb - 01 Mar

View all events...

- many standards are aligned with ISO's (ie, they are exactly the same)
- OGC usually standards closer to implementations then ISO
  - GML
  - KML
  - netCDF
- abstract specifications = ISO
- implementation specification = OGC
- unlike ISO's, OGC's documents are free for everyone 😊

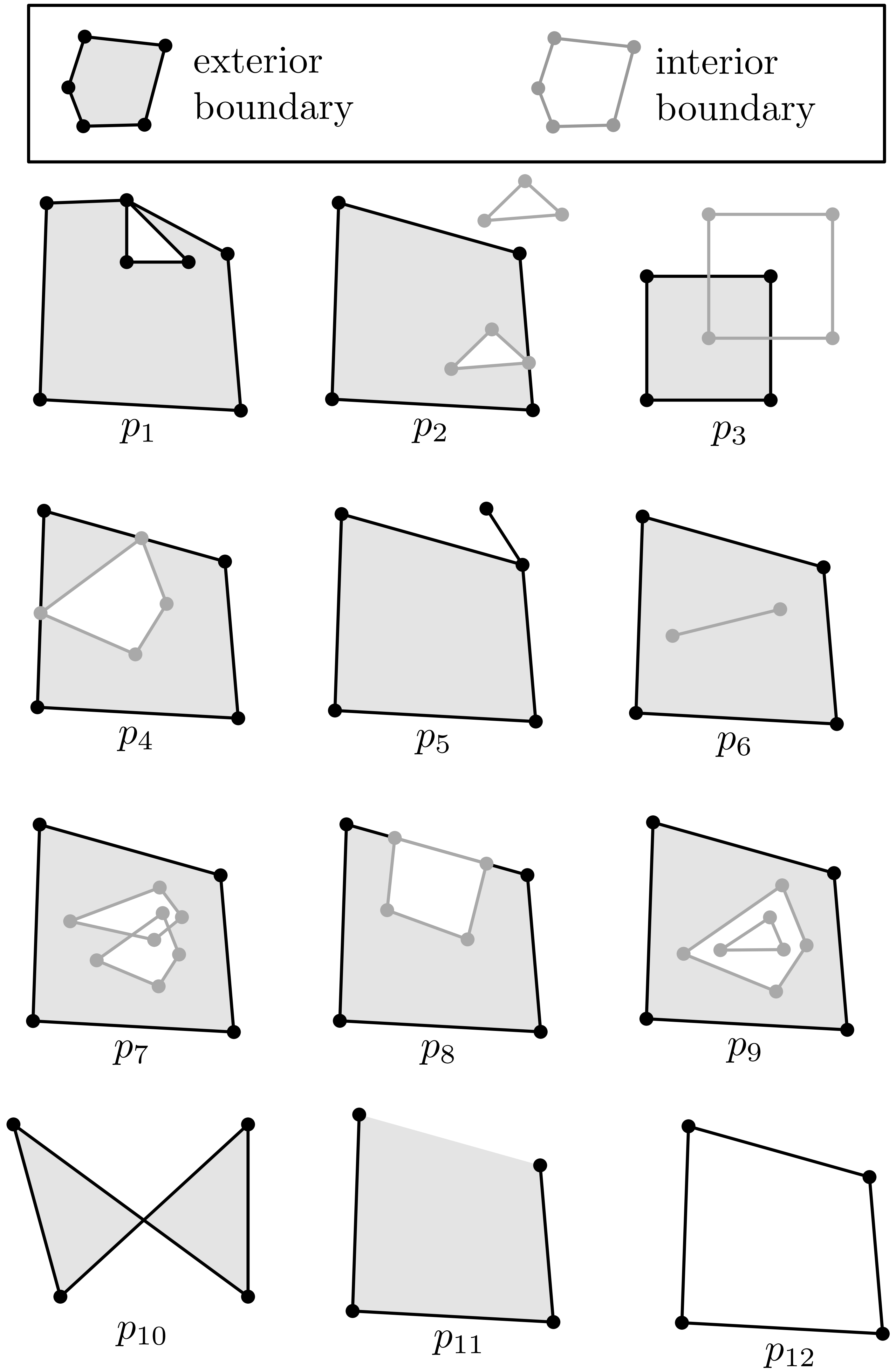
The standards from a Dutch point-of-view



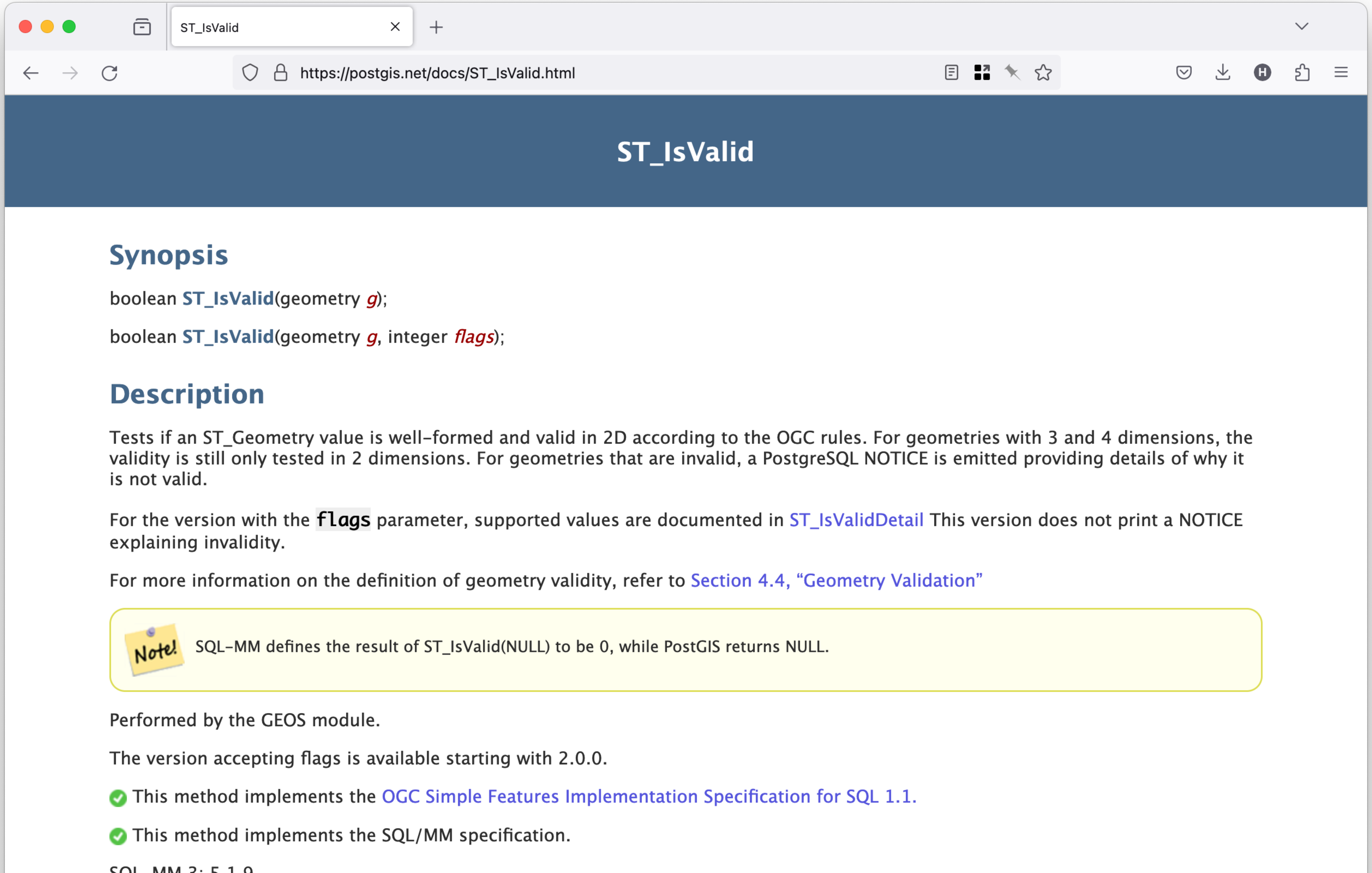
Rules for validation in 2D

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



Rules for validation in 2D: standard implementations available



ST_IsValid

Synopsis

boolean **ST_IsValid**(geometry *g*);

boolean **ST_IsValid**(geometry *g*, integer *flags*);

Description

Tests if an ST_Geometry value is well-formed and valid in 2D according to the OGC rules. For geometries with 3 and 4 dimensions, the validity is still only tested in 2 dimensions. For geometries that are invalid, a PostgreSQL NOTICE is emitted providing details of why it is not valid.

For the version with the `flags` parameter, supported values are documented in ST_IsValidDetail This version does not print a NOTICE explaining invalidity.

For more information on the definition of geometry validity, refer to Section 4.4, “Geometry Validation”

Note! SQL-MM defines the result of ST_IsValid(NULL) to be 0, while PostGIS returns NULL.

Performed by the GEOS module.

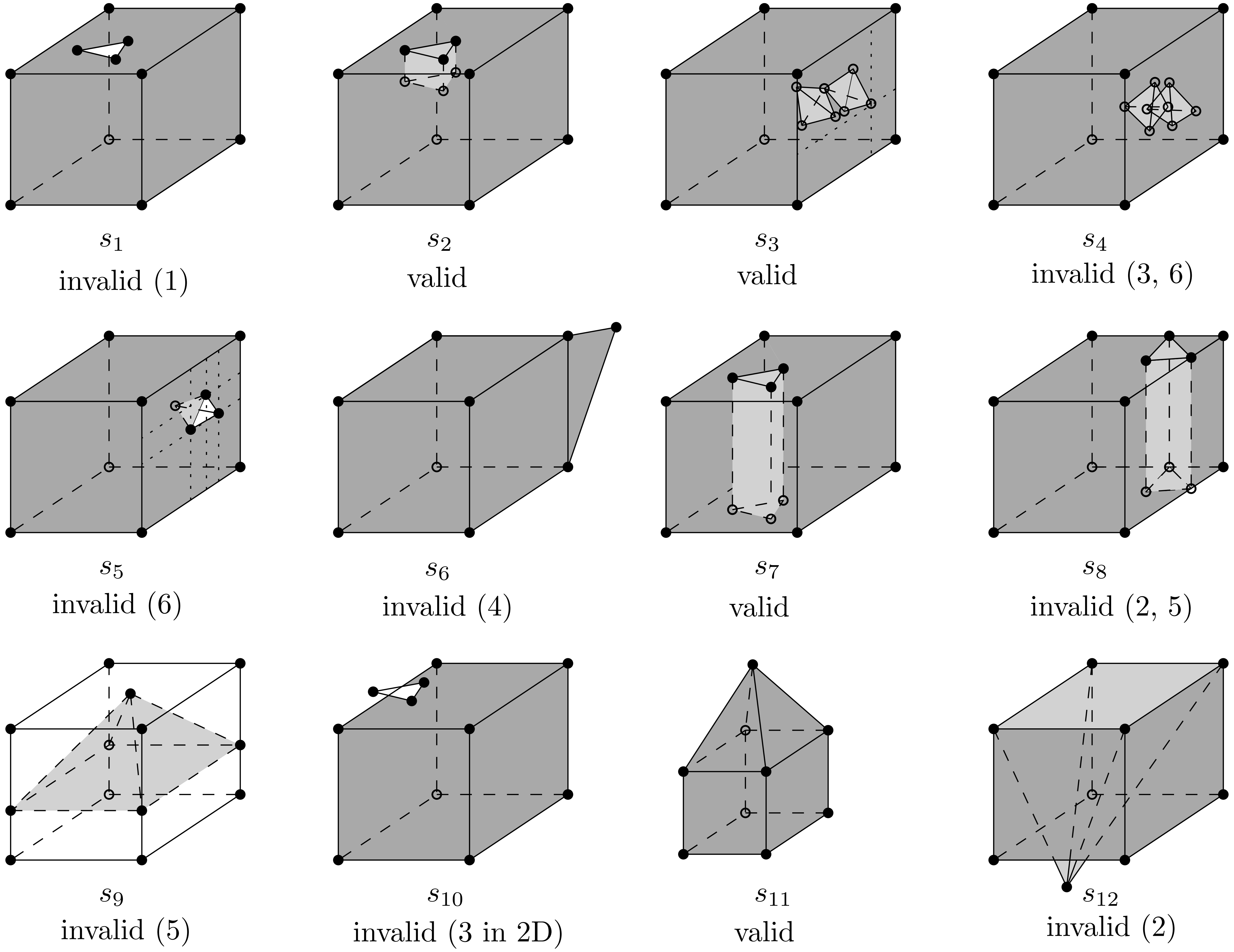
The version accepting flags is available starting with 2.0.0.

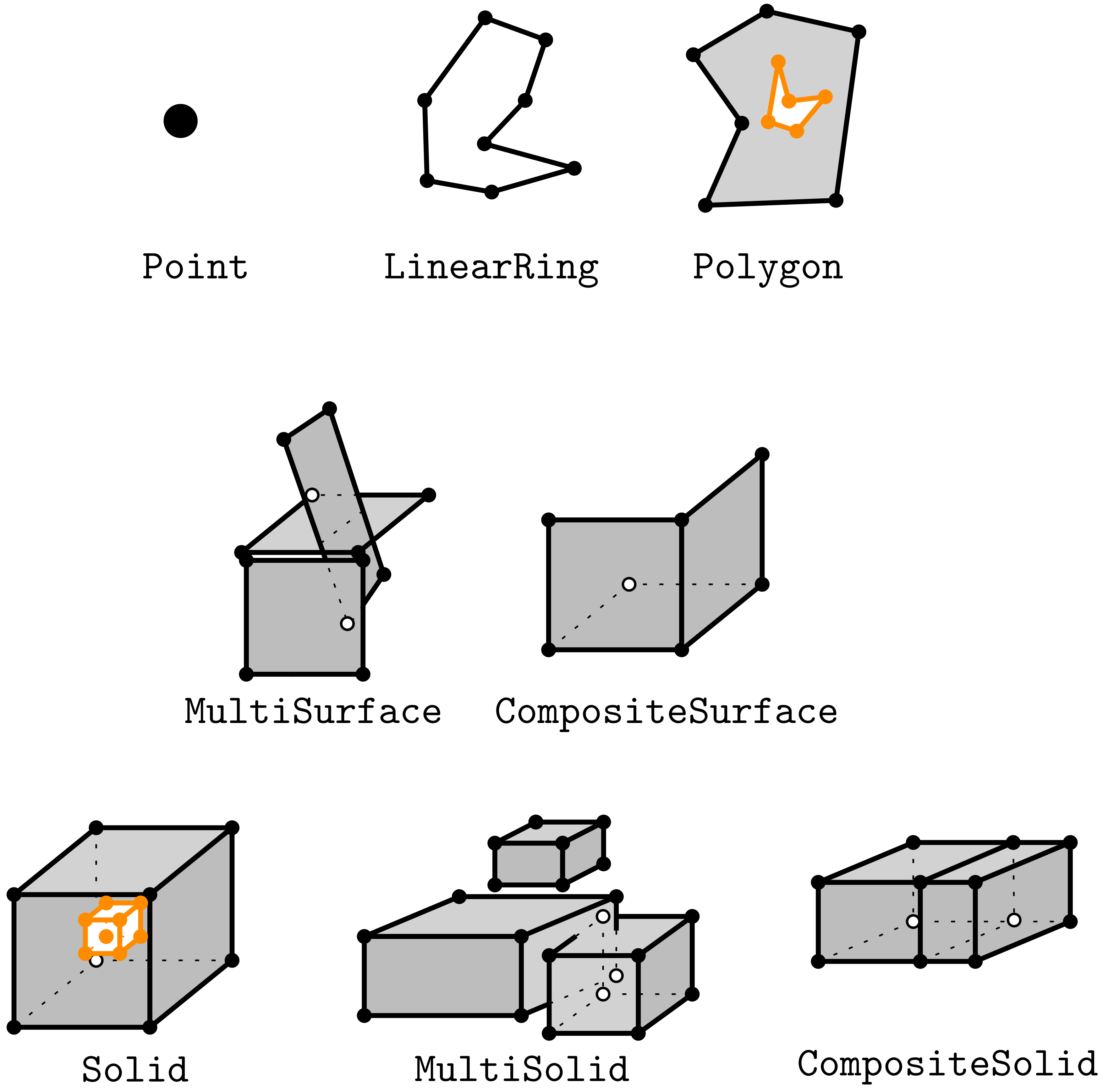
✓ This method implements the OGC Simple Features Implementation Specification for SQL 1.1.

✓ This method implements the SQL/MM specification.

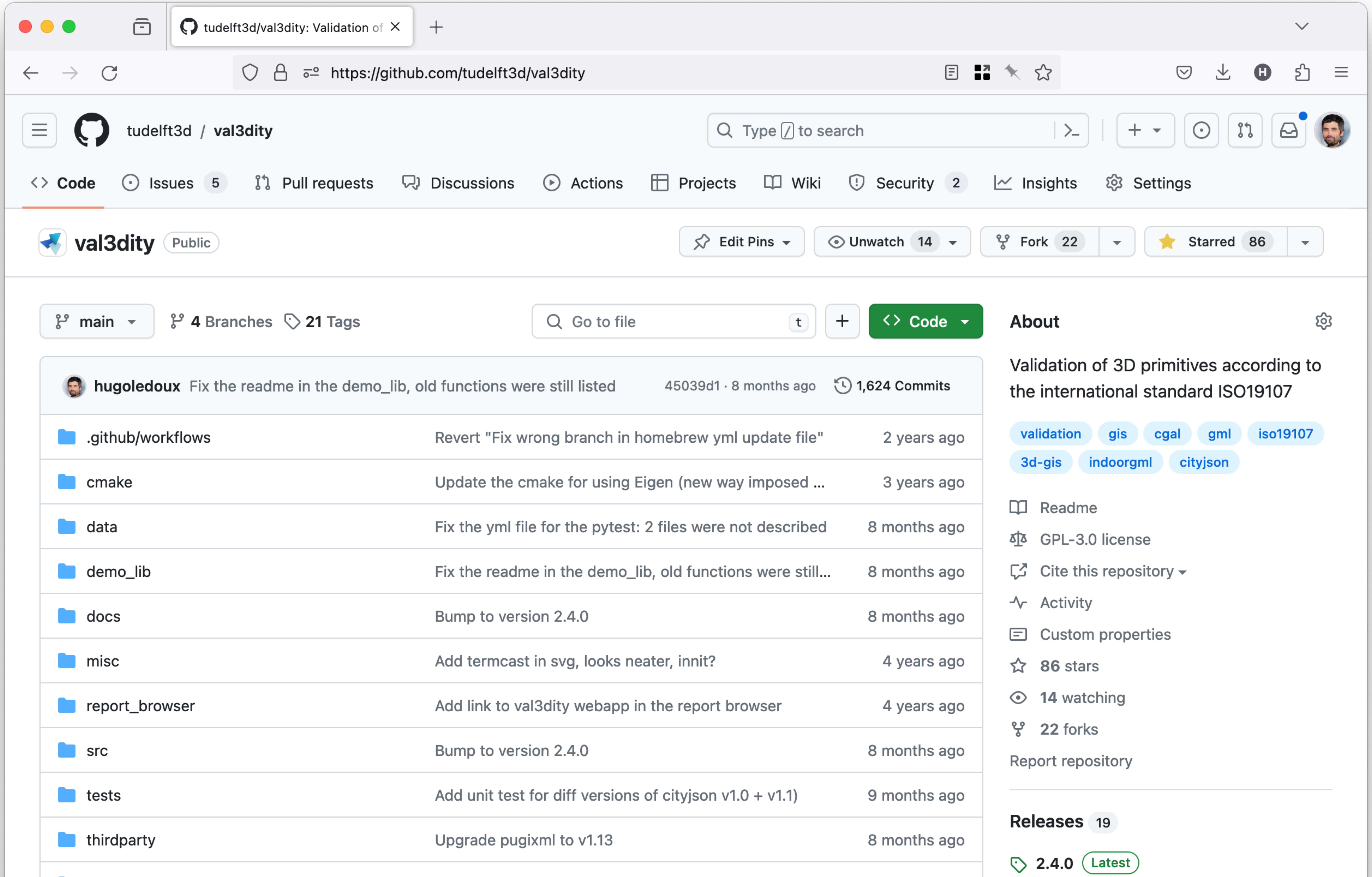
SQL-MM 3: 5.1.9

In 3D it's way way way more complex...





In 3D it's way more complex... and no GEOS/PostGIS implementation



hugoledoux Fix the readme in the demo_lib, old functions were still listed 45039d1 · 8 months ago 1,624 Commits

.github/workflows	Revert "Fix wrong branch in homebrew yml update file"	2 years ago
cmake	Update the cmake for using Eigen (new way imposed ...	3 years ago
data	Fix the yml file for the pytest: 2 files were not described	8 months ago
demo_lib	Fix the readme in the demo_lib, old functions were still...	8 months ago
docs	Bump to version 2.4.0	8 months ago
misc	Add termcast in svg, looks neater, innit?	4 years ago
report_browser	Add link to val3dity webapp in the report browser	4 years ago
src	Bump to version 2.4.0	8 months ago
tests	Add unit test for diff versions of cityjson v1.0 + v1.1)	9 months ago
thirdparty	Upgrade pugixml to v1.13	8 months ago

About

Validation of 3D primitives according to the international standard ISO19107

validation gis cgal gml iso19107 3d-gis indoorgml cityjson

- Readme
- GPL-3.0 license
- Cite this repository
- Activity
- Custom properties
- 86 stars
- 14 watching
- 22 forks

Report repository

Releases 19

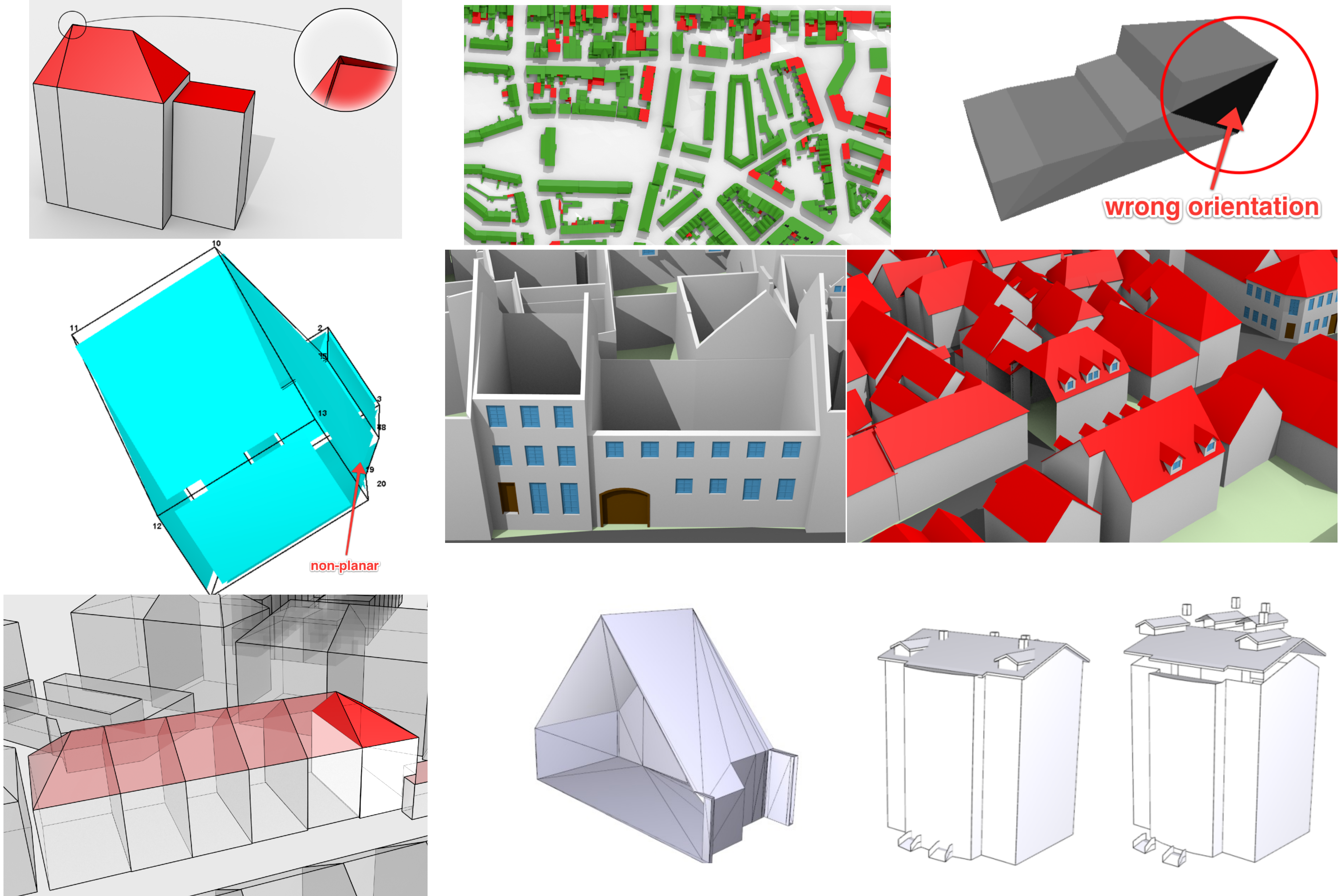
2.4.0 Latest

Quiz time

Do current 3D city models often contain these geometric errors?

tl;dl: **YES.**

Errors are *very* common in 3D city models: my wall of shame



Should we care?

YES!

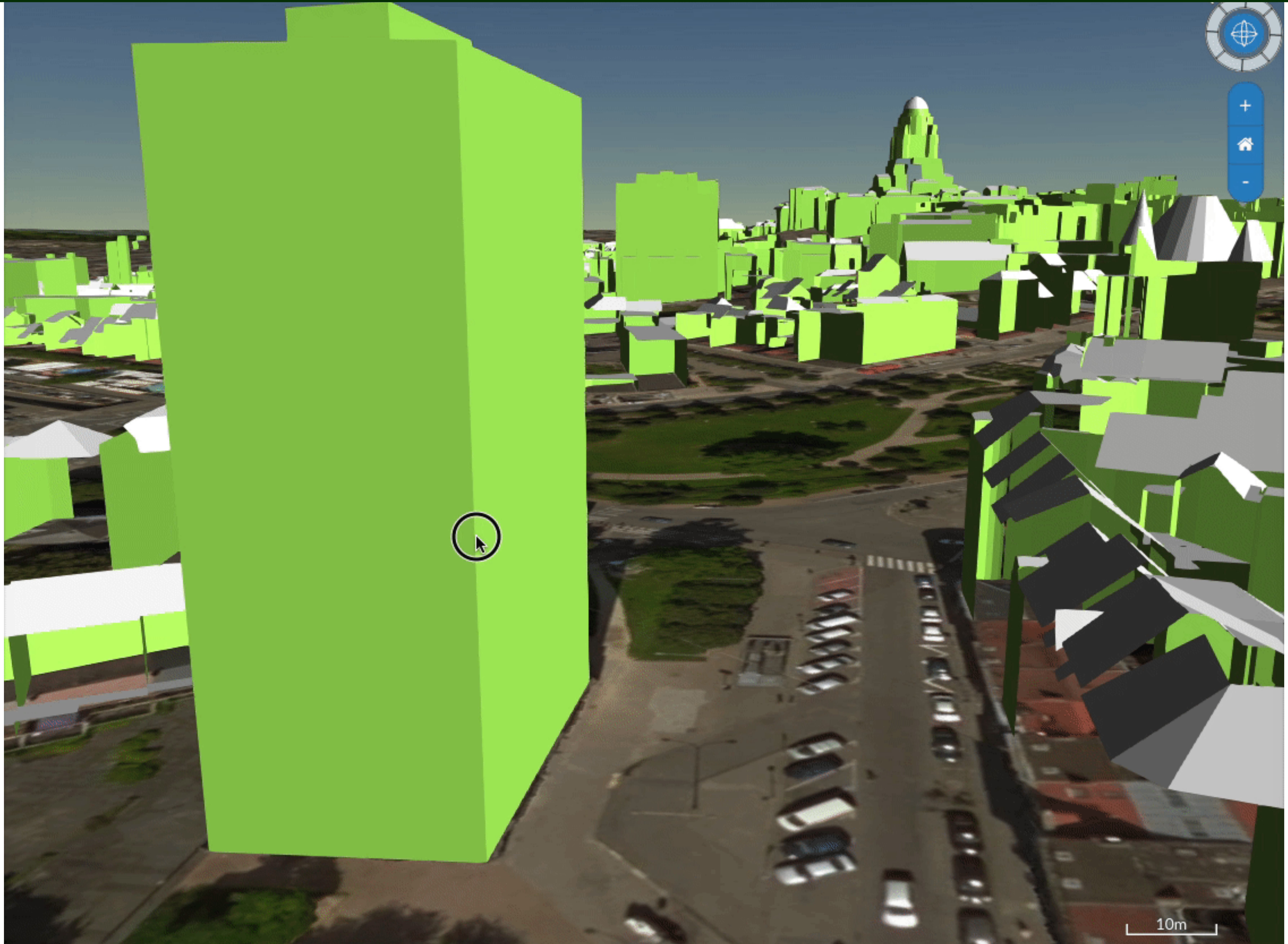
(especially if you want to pass this course 😅)

Visualisation — duplicated surfaces == annoying

Visualisation — duplicated surfaces == distracting

TUDelft 3Dgeoinfo

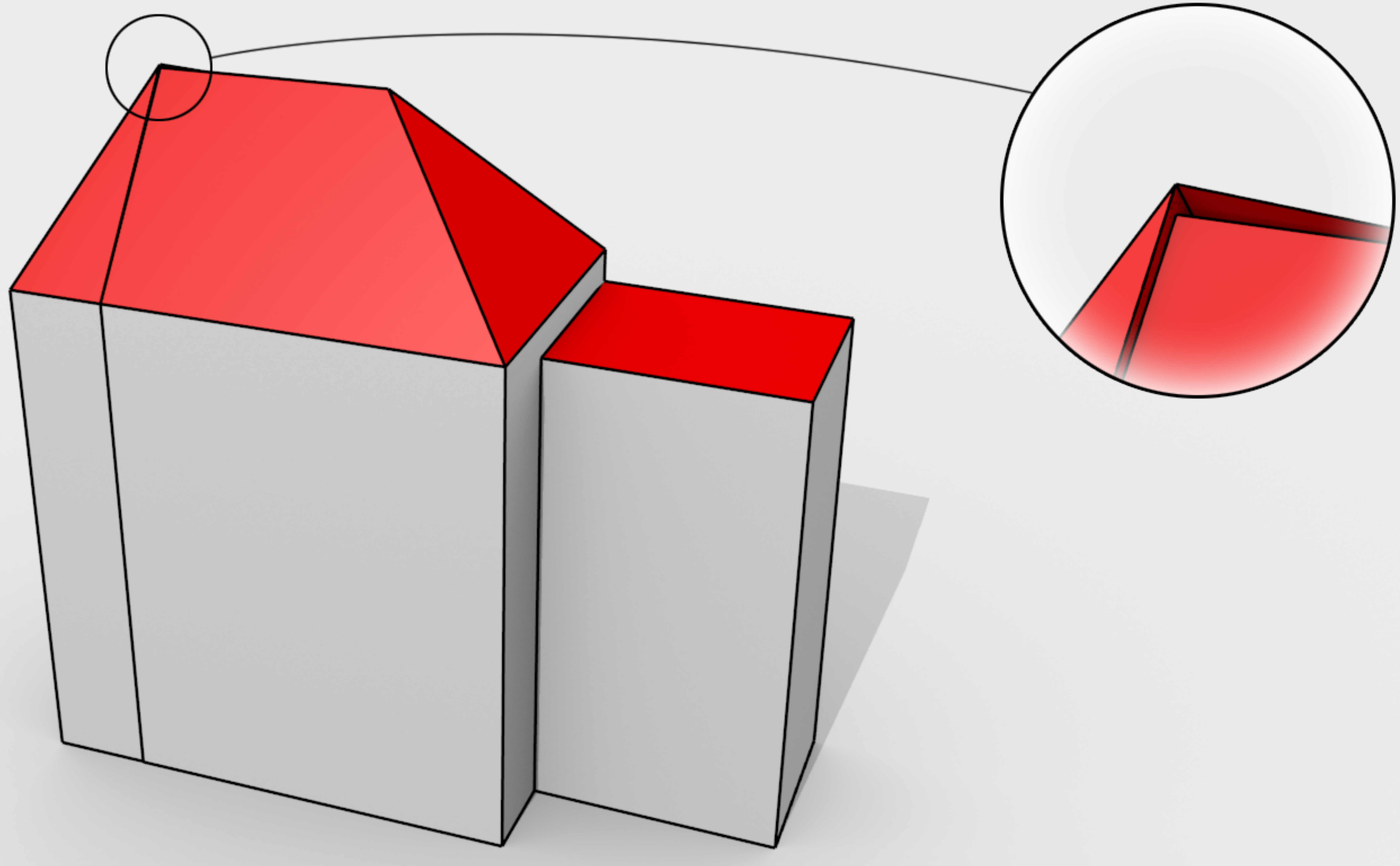
Visualisation — wrong orientation == missing faces



Visualisation — wrong orientation of surfaces (red ones)

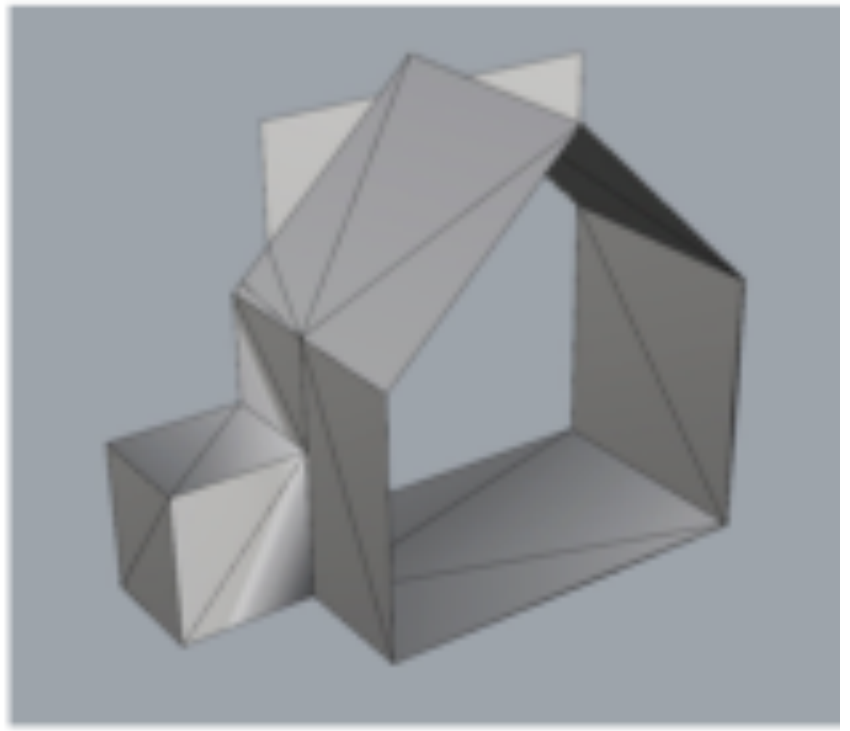
Solar potential — wrong orientation == no potential assigned

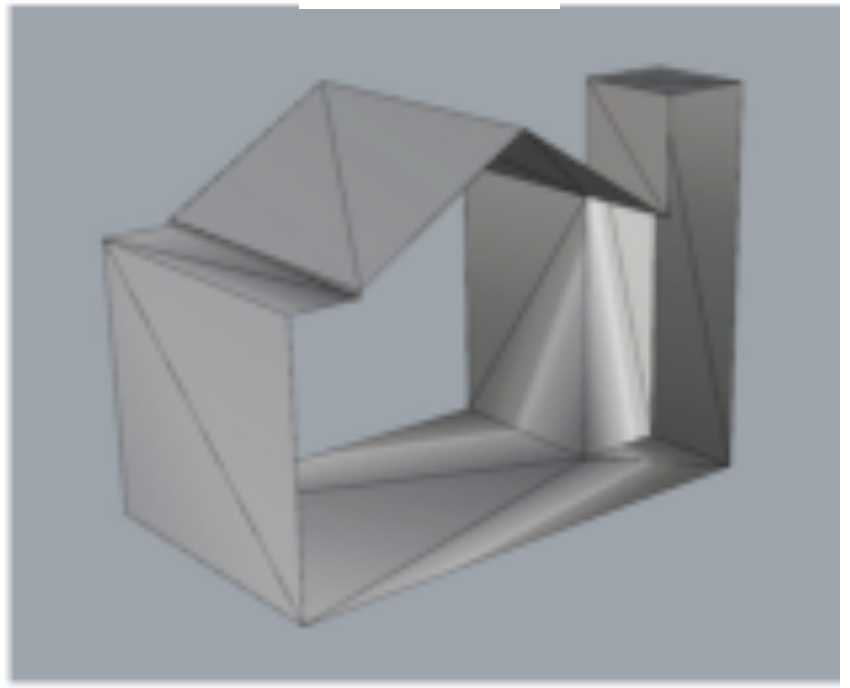
Volume calculation — tiny problems == some methods do not work

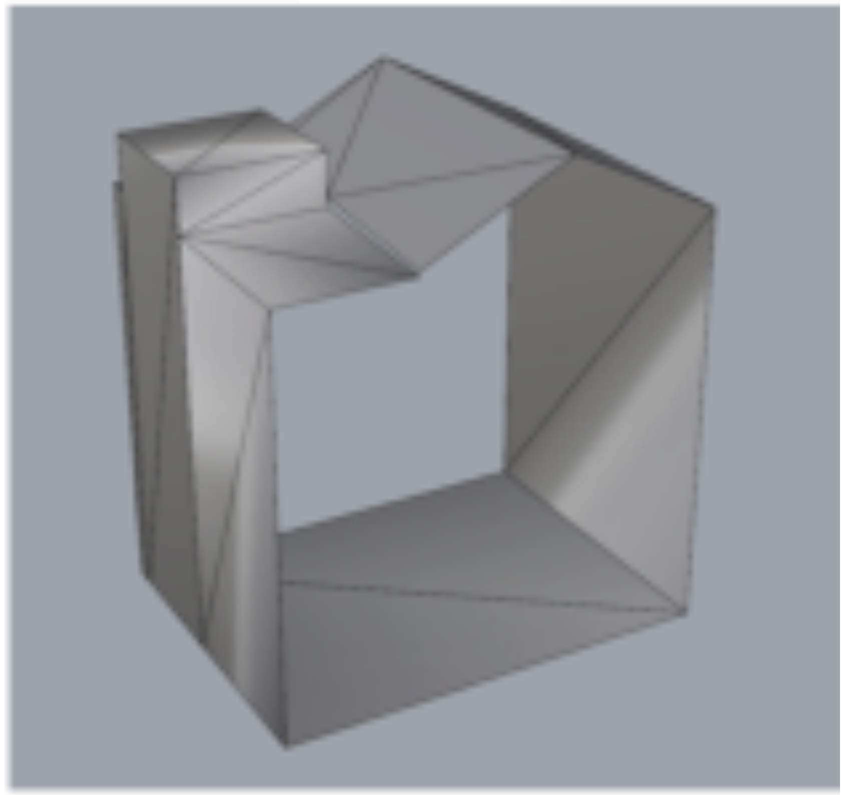


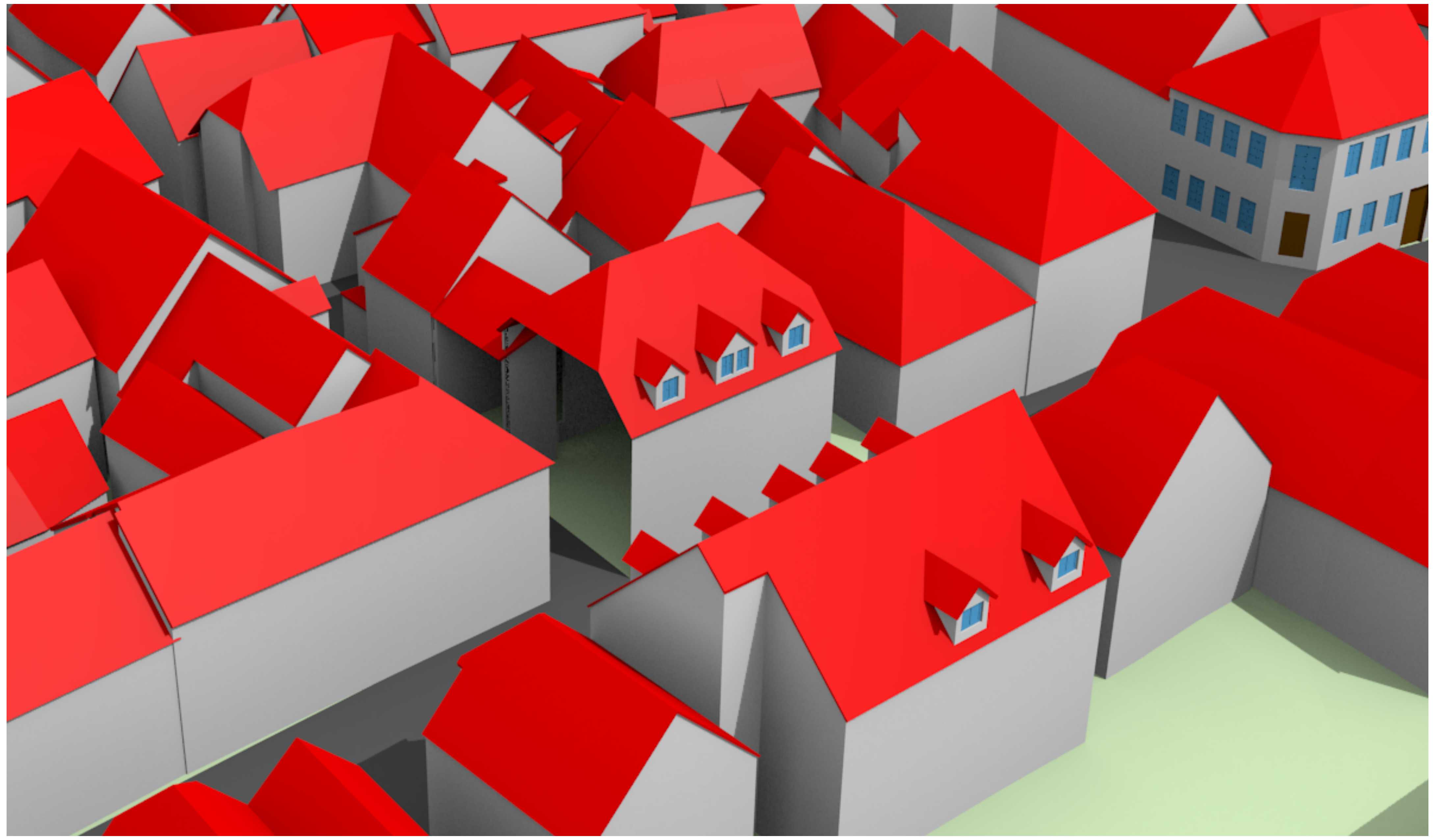
Volume calculation — tiny problems == some methods do no work

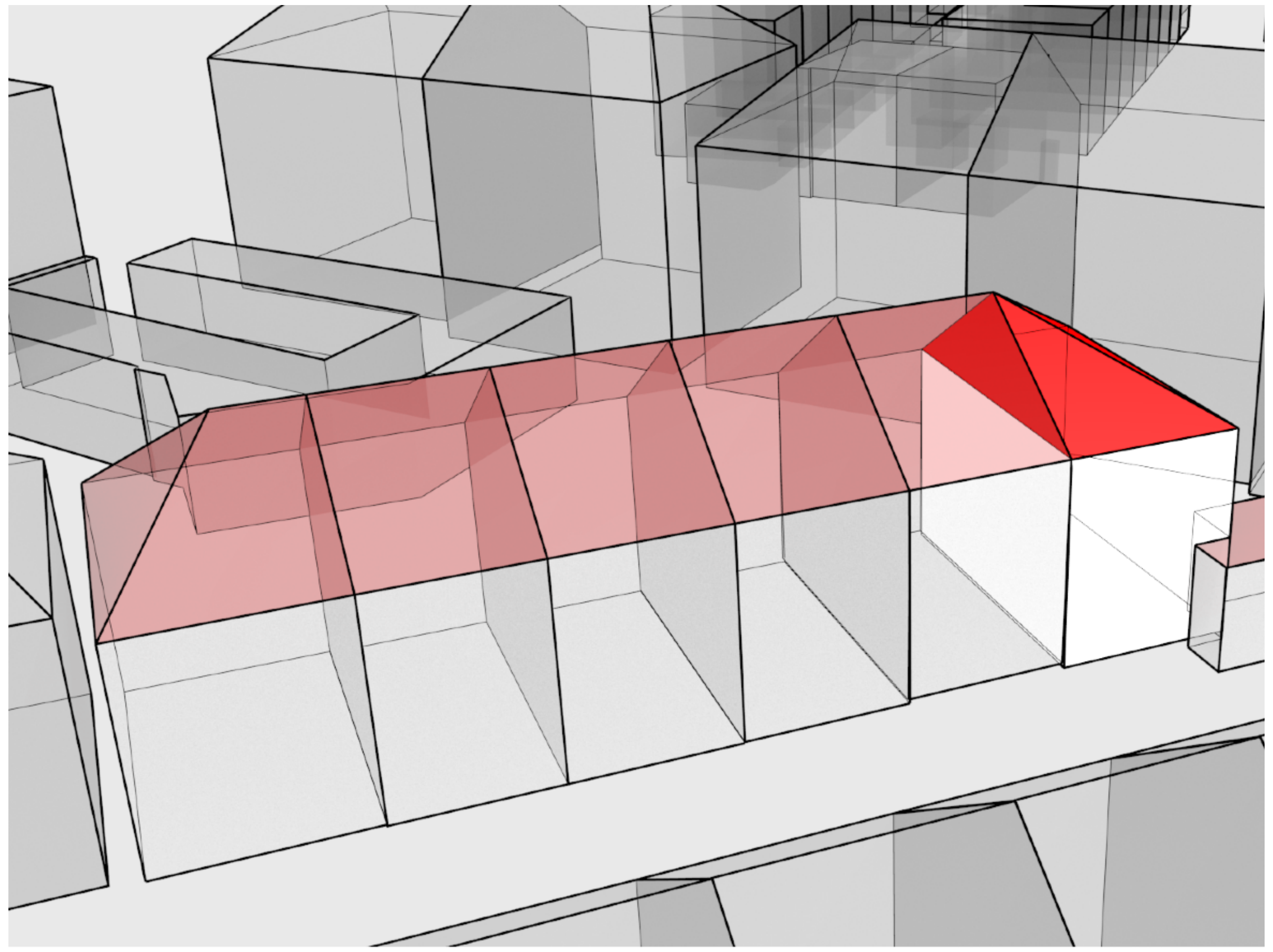
Volume calculation — big problems == some methods do no work



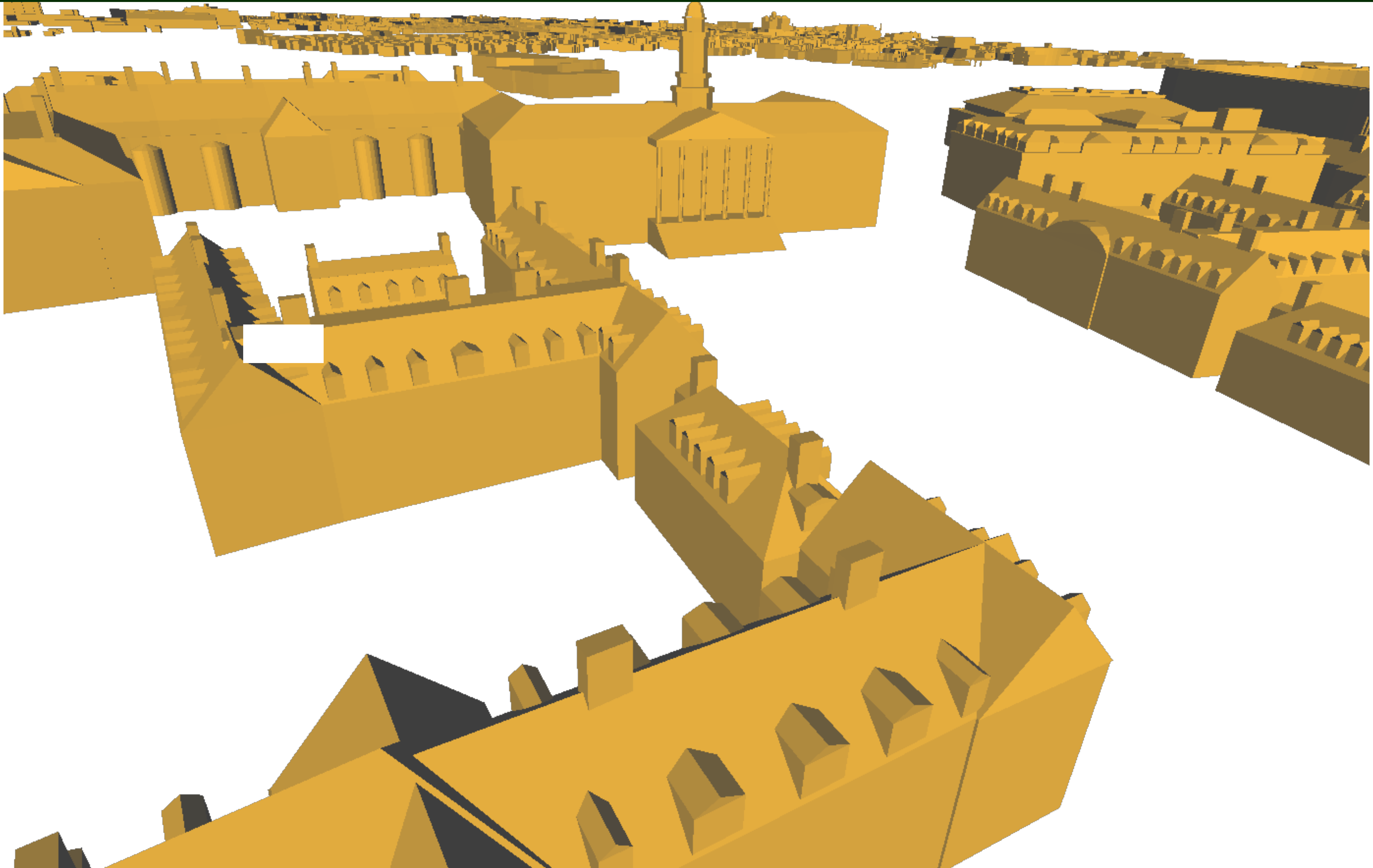




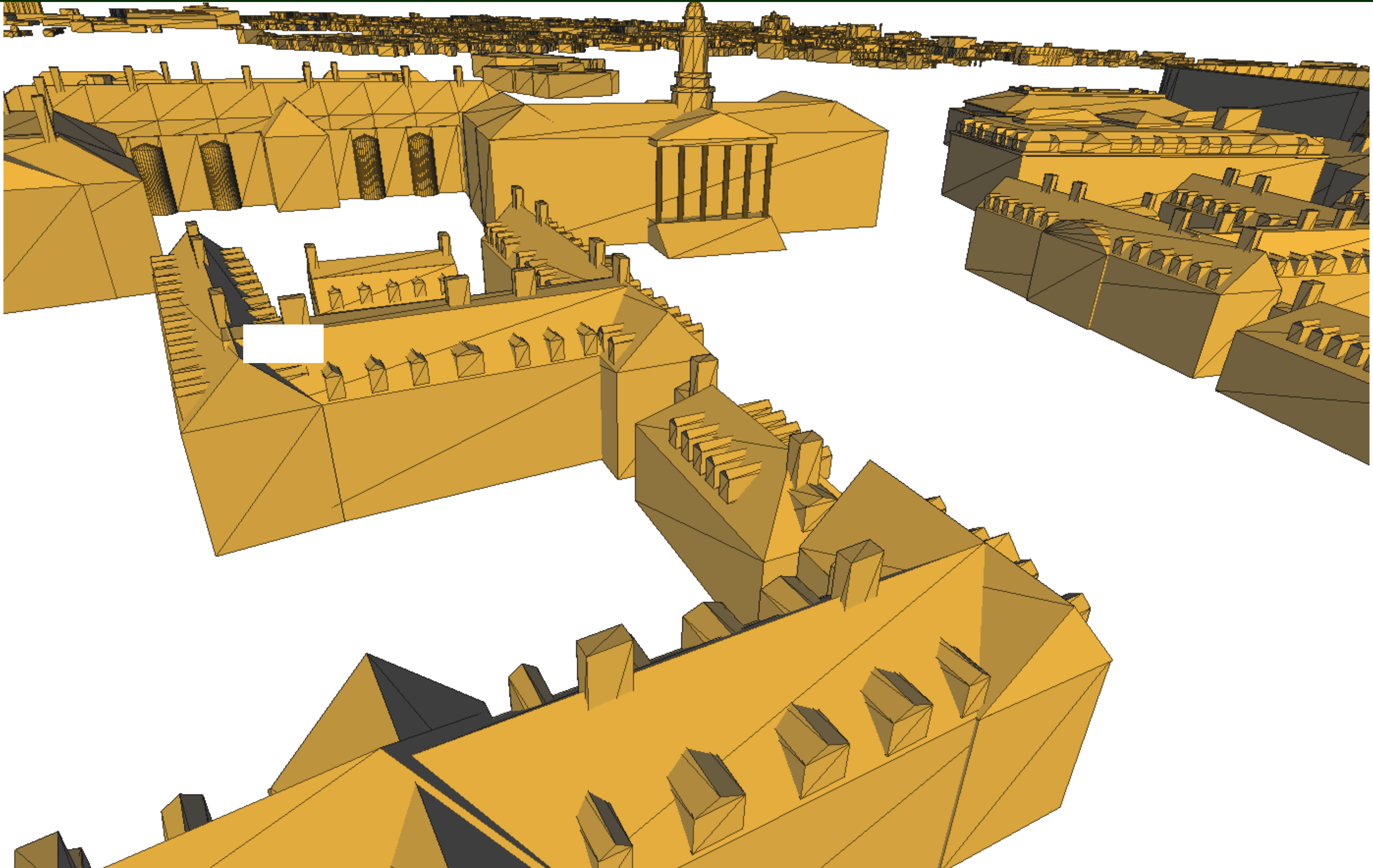




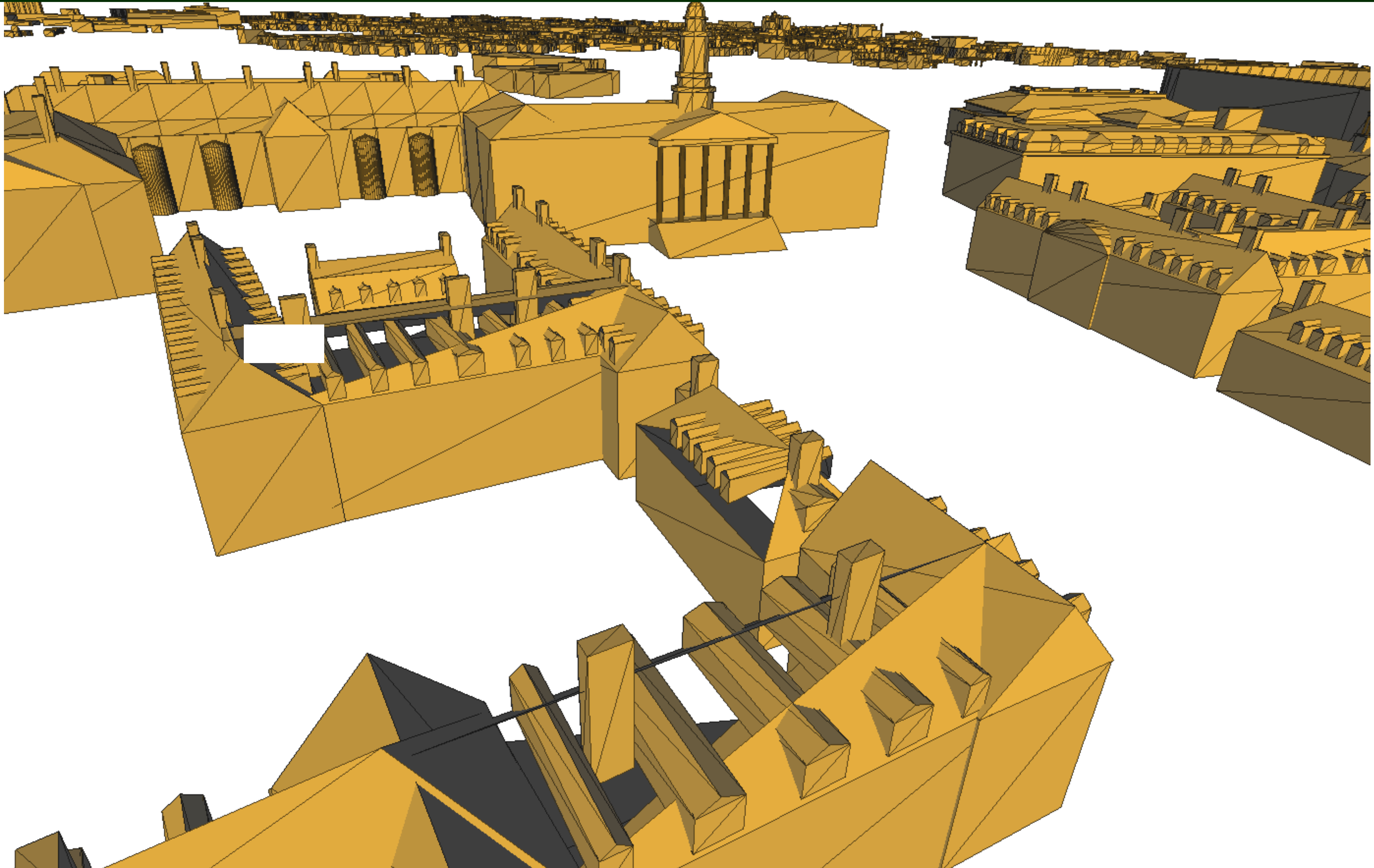
Volume calculation — superstructures not topologically connected



Volume calculation — superstructures not topologically connected



Volume calculation – superstructures not topologically connected

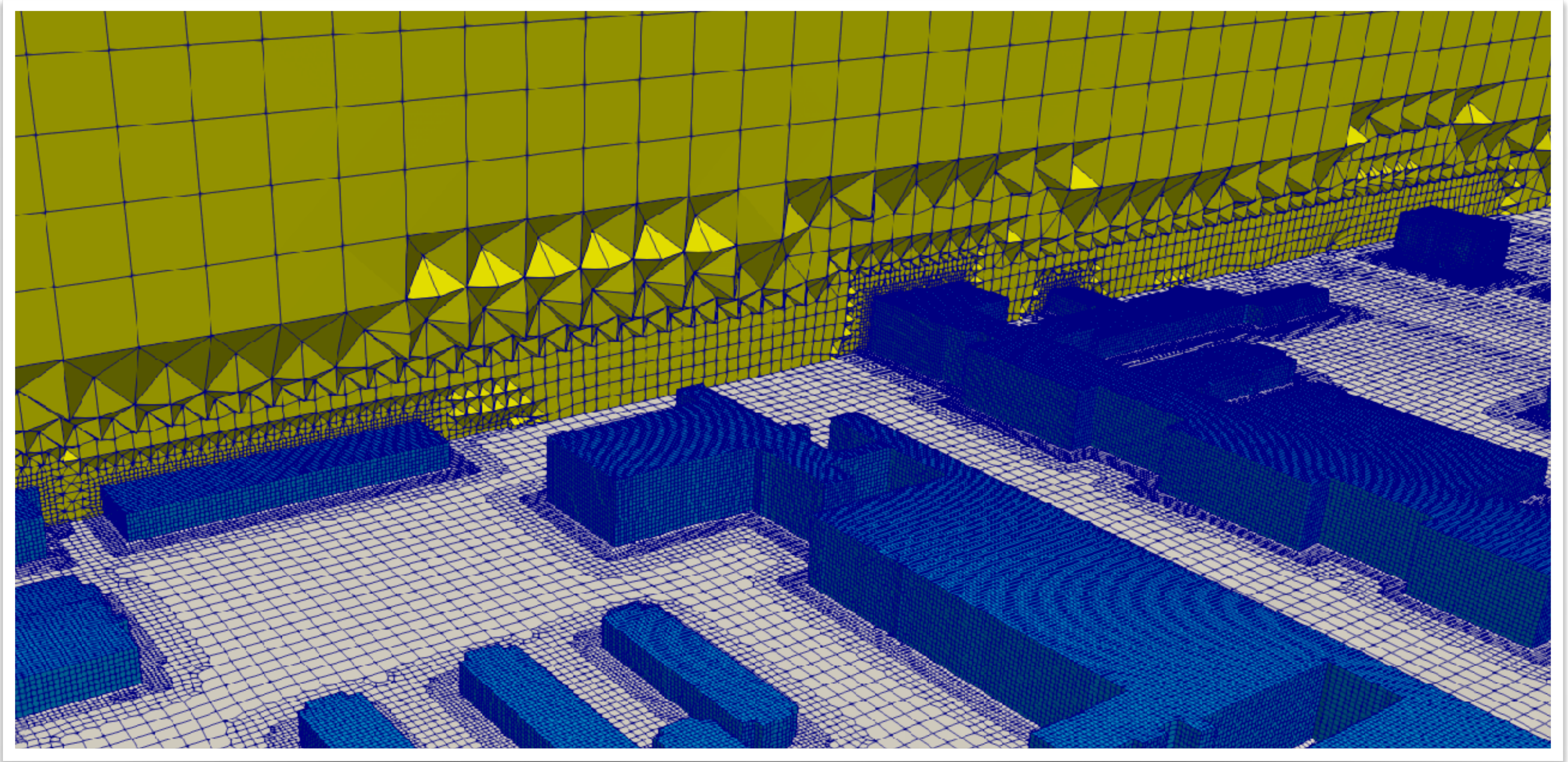


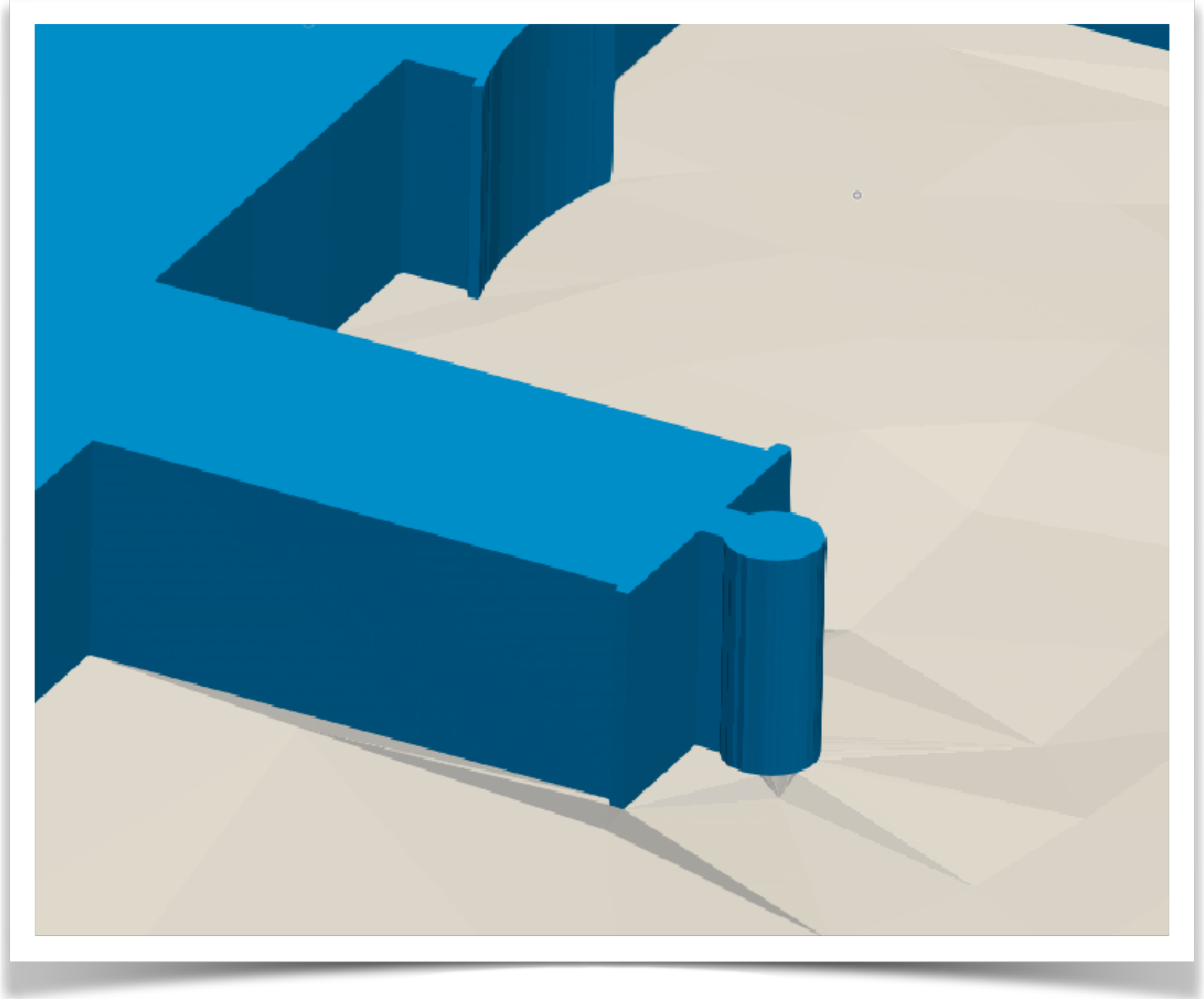
Advanced simulations — wind comfort for pedestrians

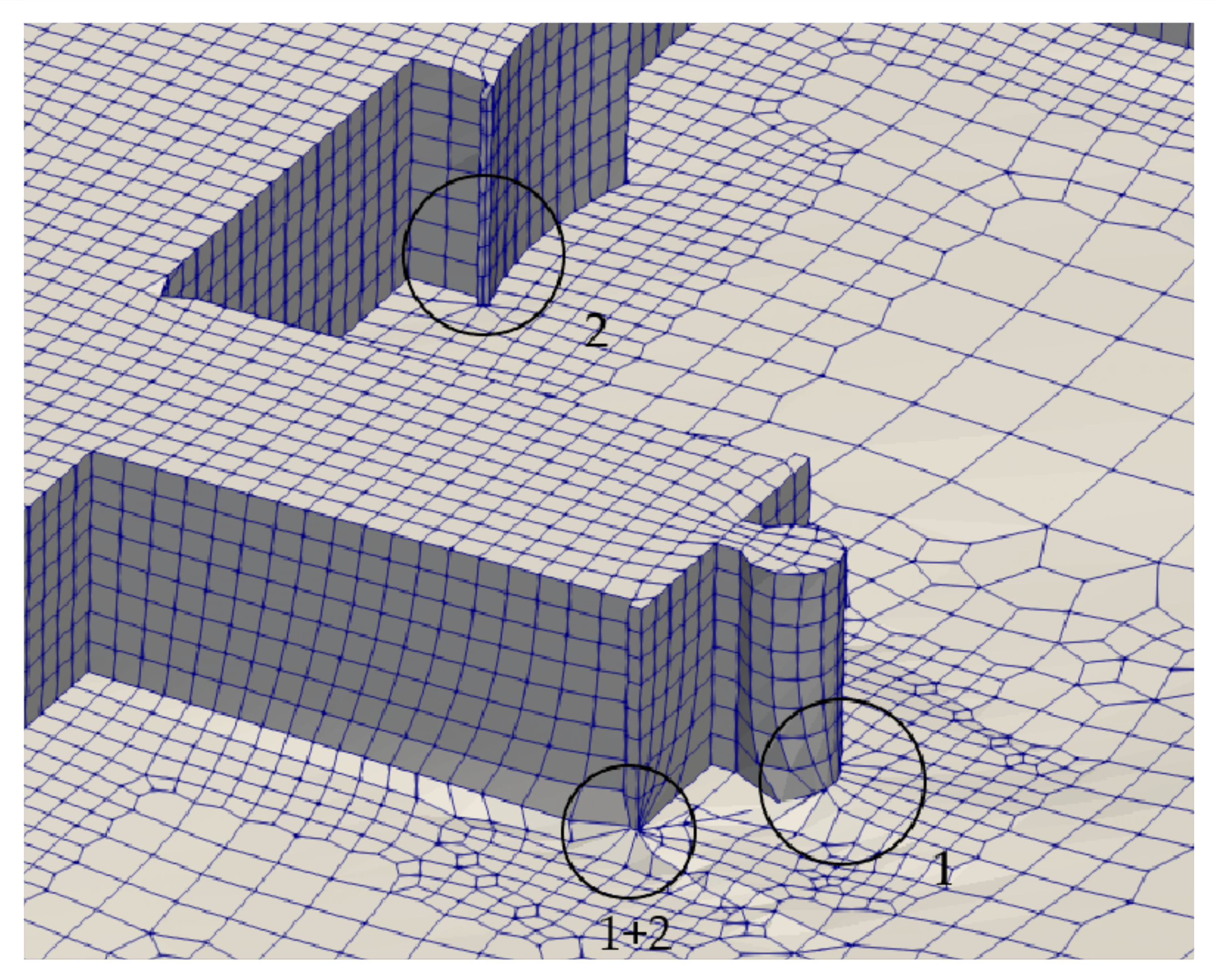
Advanced simulations — CFD methods: subdivision of volume (mesh)

computational fluid dynamics

GEO5015: Modelling wind and dispersion in urban environments (Q4!)



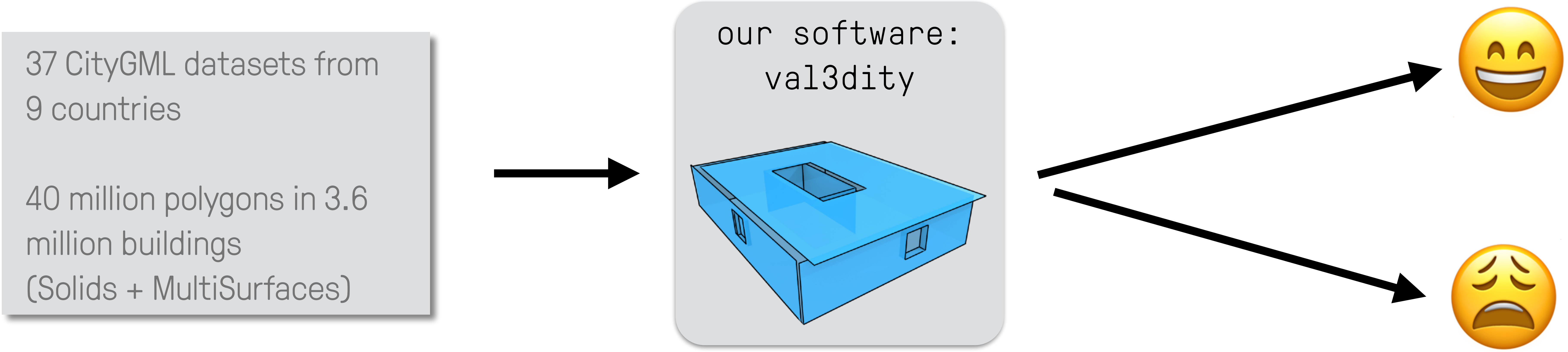




Advanced simulations — very strict input requirements

- no holes
- no intersection
- triangles must have certain shape (no *slivers*)
- a “perfect” model is thus required
- in practice, that usually means days or even weeks of (semi-)manual repair for an area like the TUDelft campus

An old-ish study we made



37 CityGML datasets from 9 countries

40 million polygons in 3.6 million buildings (Solids + MultiSurfaces)

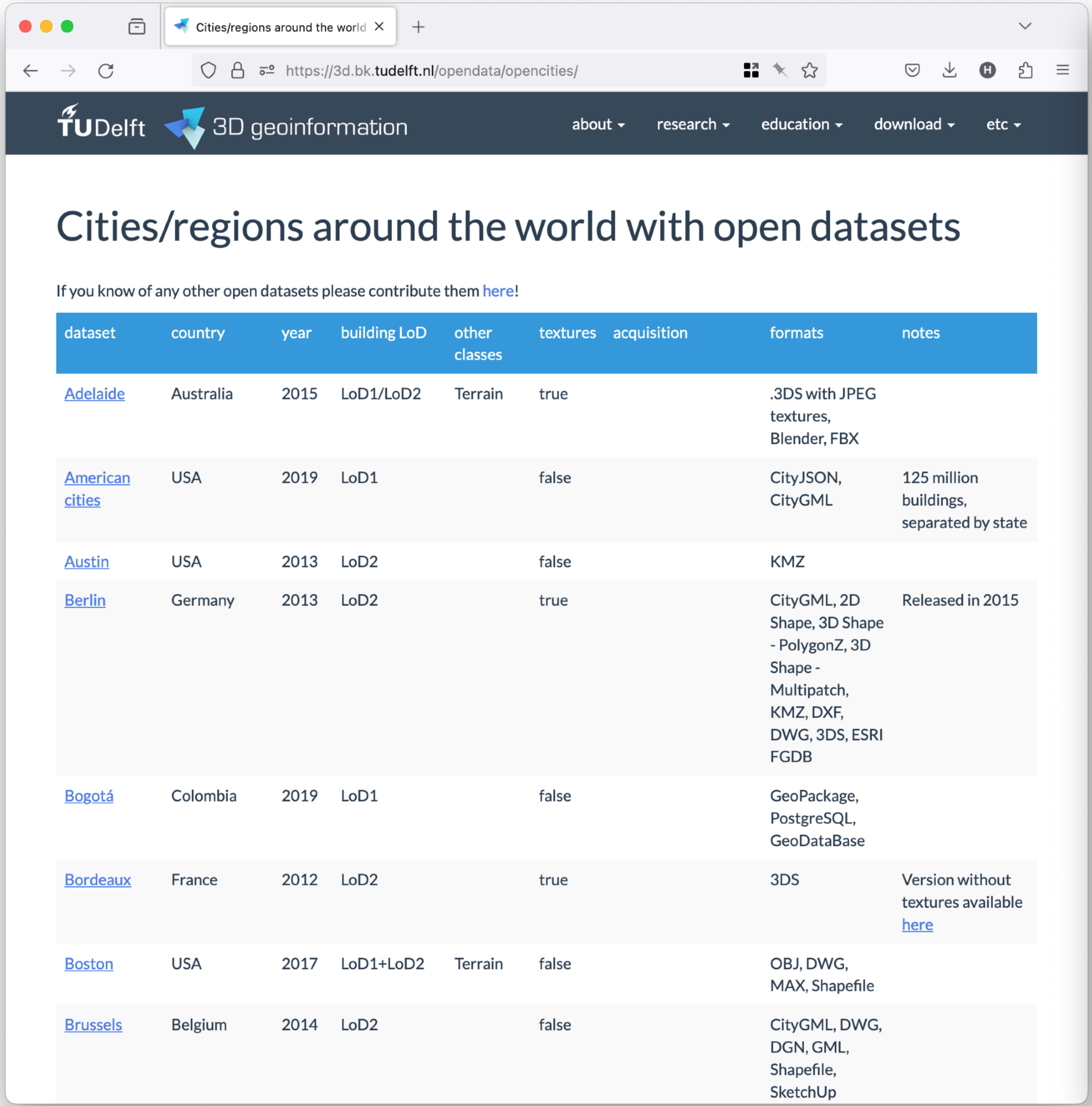
our software: val3dity

😄

😩

- 37 datasets (3D cities) from 9 countries
- 40 million polygons in 3.6 million buildings

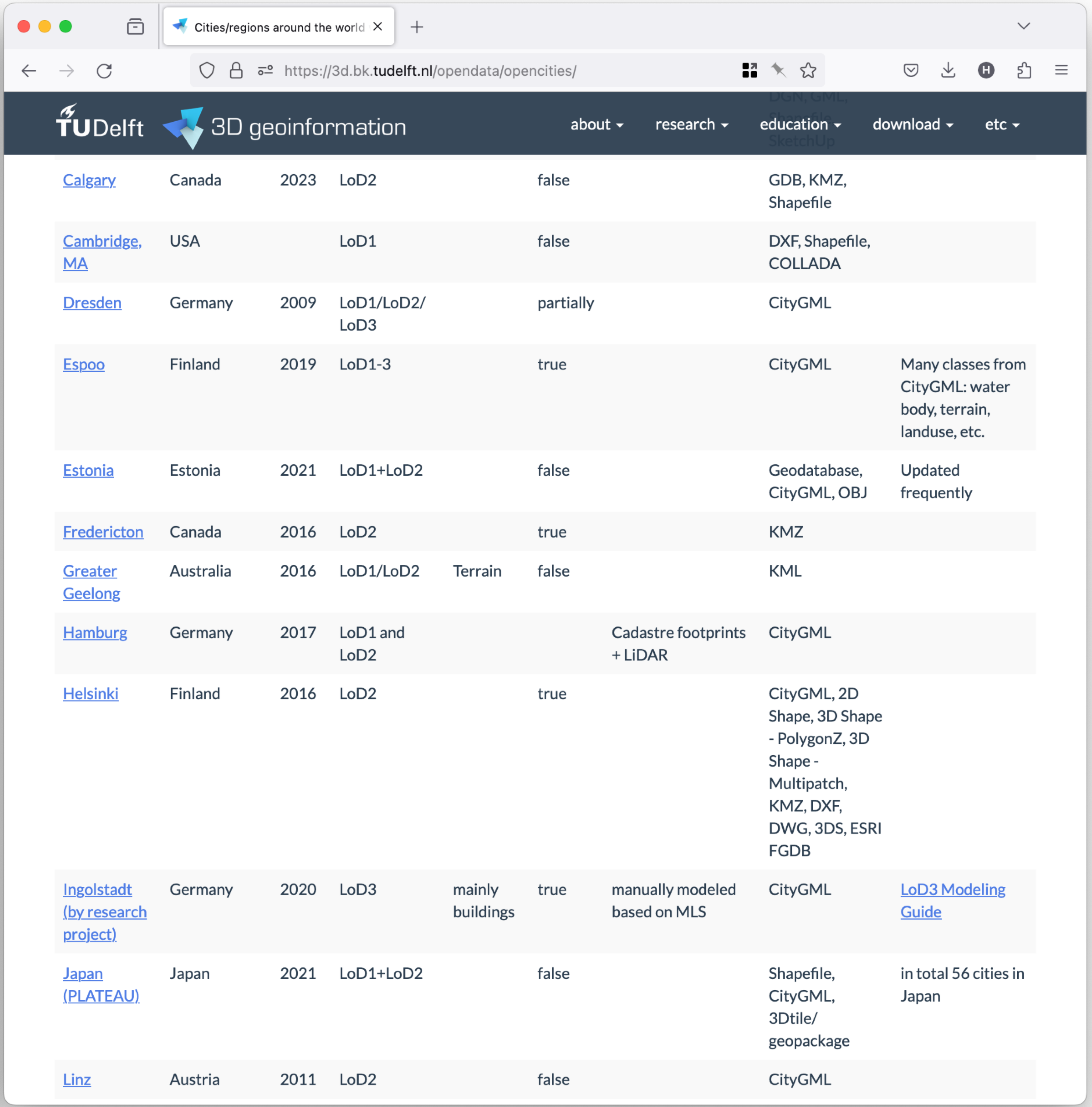
Many cities around the world have 3D models openly available


Cities/regions around the world
https://3d.bk.tudelft.nl/opendata/opencities/
TUDelft
3D geoinformation
about
research
education
download
etc

Cities/regions around the world with open datasets

If you know of any other open datasets please contribute them here!

dataset	country	year	building LoD	other classes	textures	acquisition	formats	notes
Adelaide	Australia	2015	LoD1/LoD2	Terrain	true		.3DS with JPEG textures, Blender, FBX	
American cities	USA	2019	LoD1		false		CityJSON, CityGML	125 million buildings, separated by state
Austin	USA	2013	LoD2		false		KMZ	
Berlin	Germany	2013	LoD2		true		CityGML, 2D Shape, 3D Shape - PolygonZ, 3D Shape - Multipatch, KMZ, DXF, DWG, 3DS, ESRI FGDB	Released in 2015
Bogotá	Colombia	2019	LoD1		false		GeoPackage, PostgreSQL, GeoDataBase	
Bordeaux	France	2012	LoD2		true		3DS	Version without textures available here
Boston	USA	2017	LoD1+LoD2	Terrain	false		OBJ, DWG, MAX, Shapefile	
Brussels	Belgium	2014	LoD2		false		CityGML, DWG, DGN, GML, Shapefile, SketchUp	

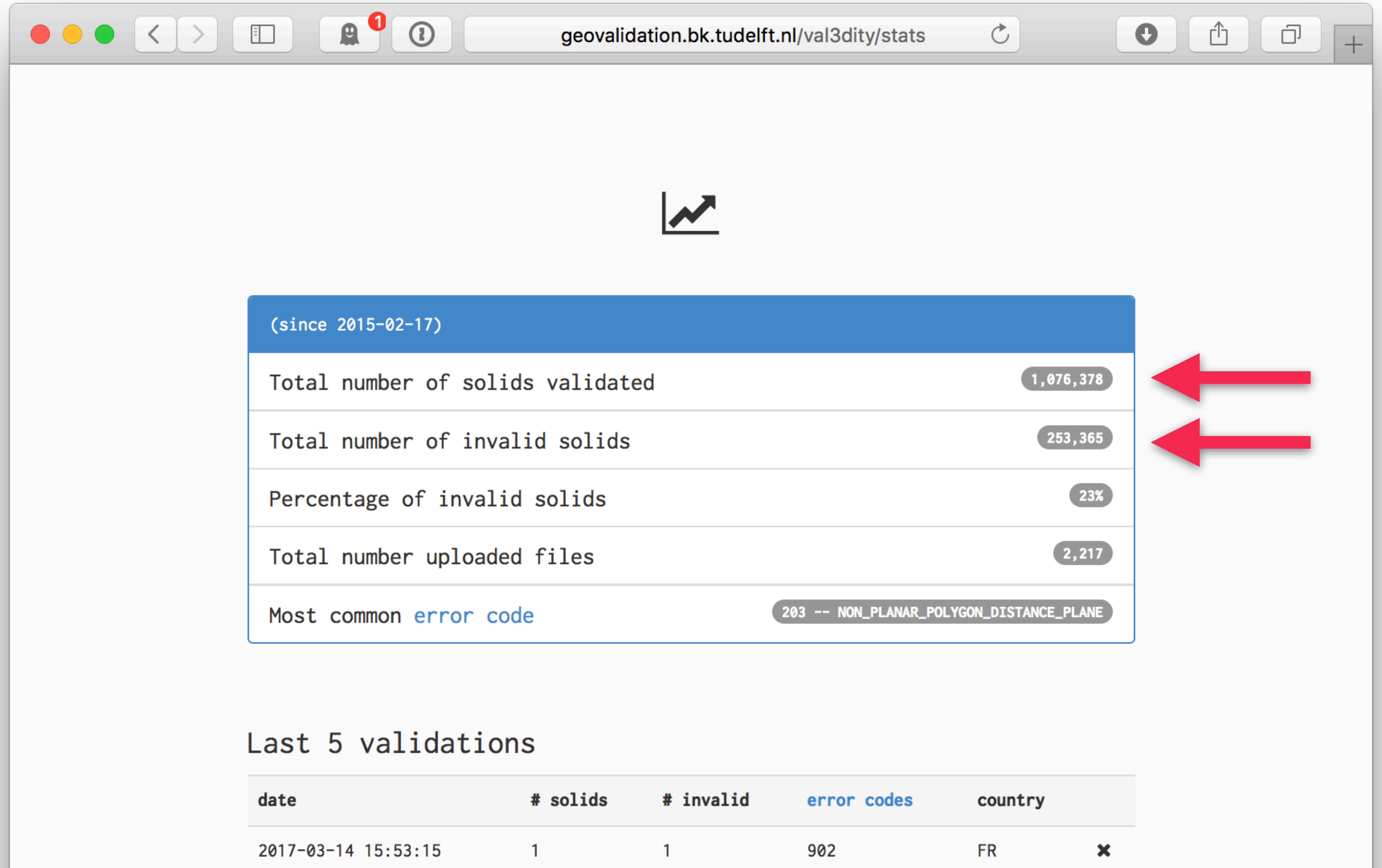

Cities/regions around the world
https://3d.bk.tudelft.nl/opendata/opencities/
TUDelft
3D geoinformation
about
research
education
download
etc

dataset	country	year	building LoD	other classes	textures	acquisition	formats	notes
Calgary	Canada	2023	LoD2		false		GDB, KMZ, Shapefile	
Cambridge, MA	USA		LoD1		false		DXF, Shapefile, COLLADA	
Dresden	Germany	2009	LoD1/LoD2/LoD3		partially		CityGML	
Espoo	Finland	2019	LoD1-3		true		CityGML	Many classes from CityGML: water body, terrain, landuse, etc.
Estonia	Estonia	2021	LoD1+LoD2		false		Geodatabase, CityGML, OBJ	Updated frequently
Fredericton	Canada	2016	LoD2		true		KMZ	
Greater Geelong	Australia	2016	LoD1/LoD2	Terrain	false		KML	
Hamburg	Germany	2017	LoD1 and LoD2			Cadastre footprints + LiDAR	CityGML	
Helsinki	Finland	2016	LoD2		true		CityGML, 2D Shape, 3D Shape - PolygonZ, 3D Shape - Multipatch, KMZ, DXF, DWG, 3DS, ESRI FGDB	
Ingolstadt (by research project)	Germany	2020	LoD3	mainly buildings	true	manually modeled based on MLS	CityGML	LoD3 Modeling Guide
Japan (PLATEAU)	Japan	2021	LoD1+LoD2		false		Shapefile, CityGML, 3Dtile/geopackage	in total 56 cities in Japan
Linz	Austria	2011	LoD2		false		CityGML	

Summary of results

- No 3D city dataset is 100% valid, not even LoD1.2 😩
- Many simple errors, eg:
  - repeated vertices
  - non-planar polygons (most common error!)

Notice that these are often not visible



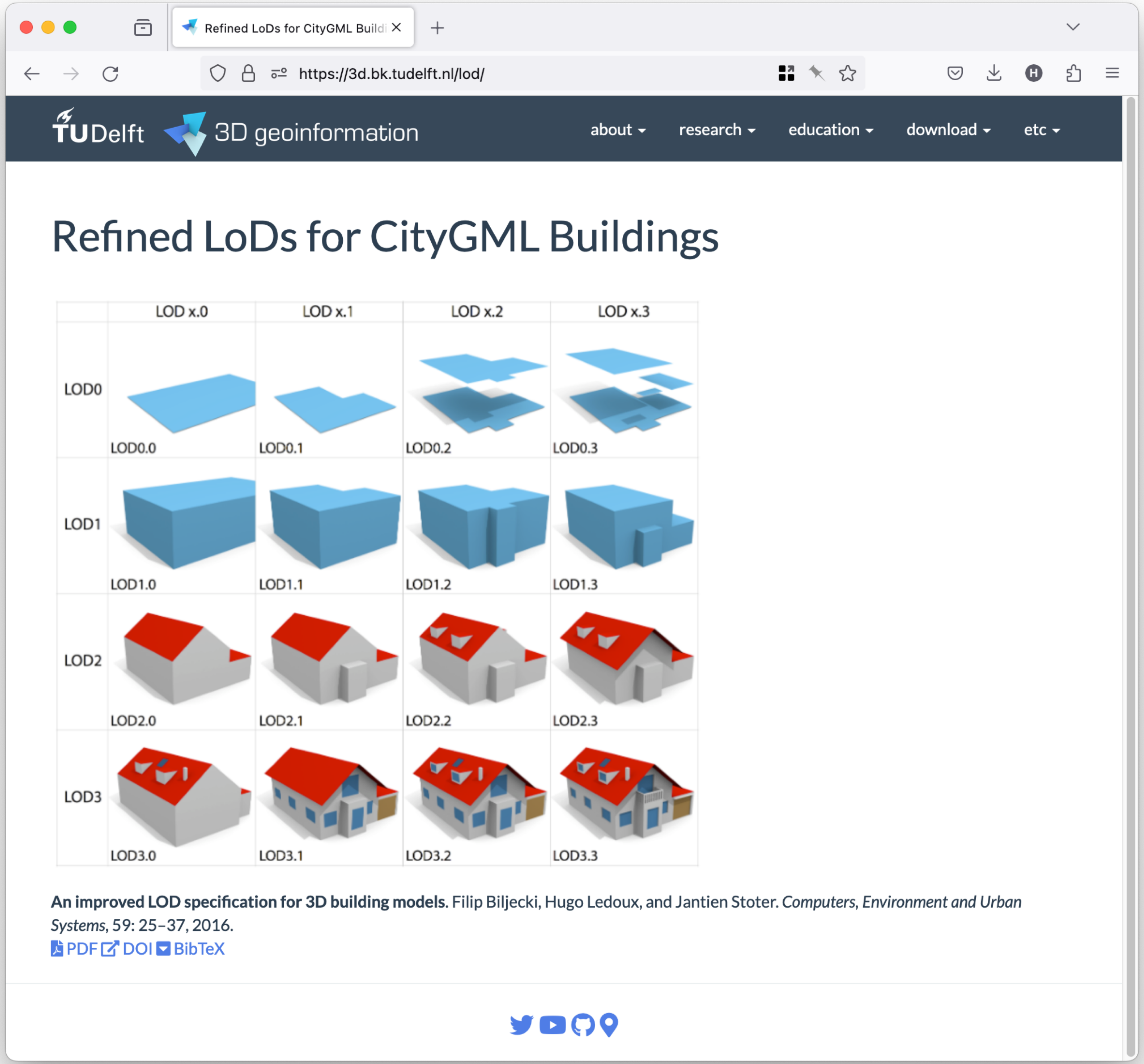
(since 2015-02-17)

Total number of solids validated	1,076,378
Total number of invalid solids	253,365
Percentage of invalid solids	23%
Total number uploaded files	2,217
Most common error code	203 -- NON_PLANAR_POLYGON_DISTANCE_PLANE

Last 5 validations

date	# solids	# invalid	error codes	country	
2017-03-14 15:53:15	1	1	902	FR	✖

The Levels-of-details (LoDs)



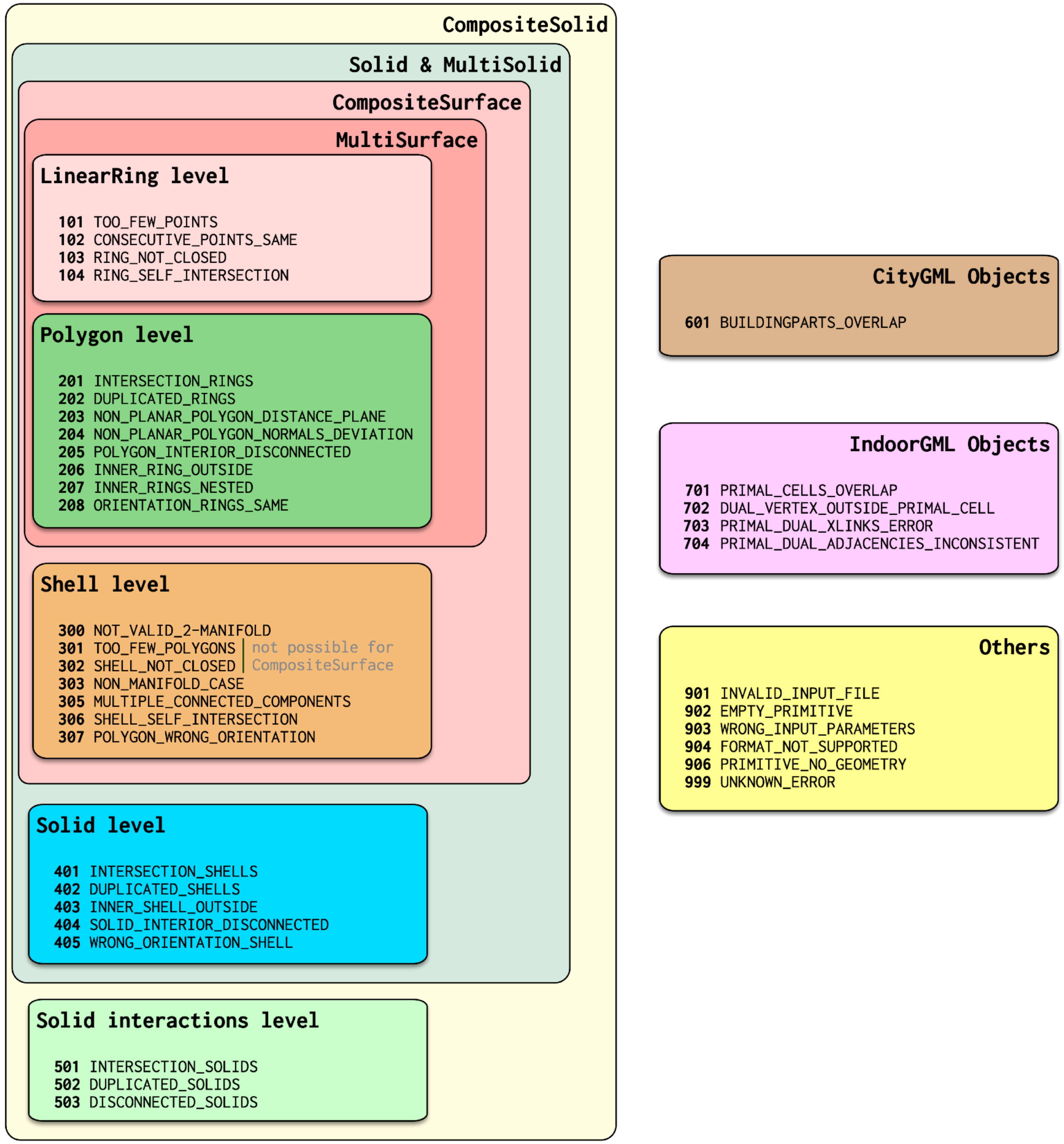
Refined LoDs for CityGML Buildings

	LOD x.0	LOD x.1	LOD x.2	LOD x.3
LOD0	LOD0.0	LOD0.1	LOD0.2	LOD0.3
LOD1	LOD1.0	LOD1.1	LOD1.2	LOD1.3
LOD2	LOD2.0	LOD2.1	LOD2.2	LOD2.3
LOD3	LOD3.0	LOD3.1	LOD3.2	LOD3.3

An improved LOD specification for 3D building models. Filip Biljecki, Hugo Ledoux, and Jantien Stoter. *Computers, Environment and Urban Systems*, 59: 25–37, 2016.

PDF DOI BibTeX

val3dity reports with a specific error code



CompositeSolid

Solid & MultiSolid

CompositeSurface

MultiSurface

LinearRing level

101 TOO_FEW_POINTS
102 CONSECUTIVE_POINTS_SAME
103 RING_NOT_CLOSED
104 RING_SELF_INTERSECTION

Polygon level

201 INTERSECTION_RINGS
202 DUPLICATED_RINGS
203 NON_PLANAR_POLYGON_DISTANCE_PLANE
204 NON_PLANAR_POLYGON_NORMALS_DEVIATION
205 POLYGON_INTERIOR_DISCONNECTED
206 INNER_RING_OUTSIDE
207 INNER_RINGS_NESTED
208 ORIENTATION_RINGS_SAME

Shell level

300 NOT_VALID_2-MANIFOLD
301 TOO_FEW_POLYGONS
302 SHELL_NOT_CLOSED
303 NON_MANIFOLD_CASE
305 MULTIPLE_CONNECTED_COMPONENTS
306 SHELL_SELF_INTERSECTION
307 POLYGON_WRONG_ORIENTATION

not possible for CompositeSurface

Solid level

401 INTERSECTION_SHELLS
402 DUPLICATED_SHELLS
403 INNER_SHELL_OUTSIDE
404 SOLID_INTERIOR_DISCONNECTED
405 WRONG_ORIENTATION_SHELL

Solid interactions level

501 INTERSECTION_SOLIDS
502 DUPLICATED_SOLIDS
503 DISCONNECTED_SOLIDS

CityGML Objects

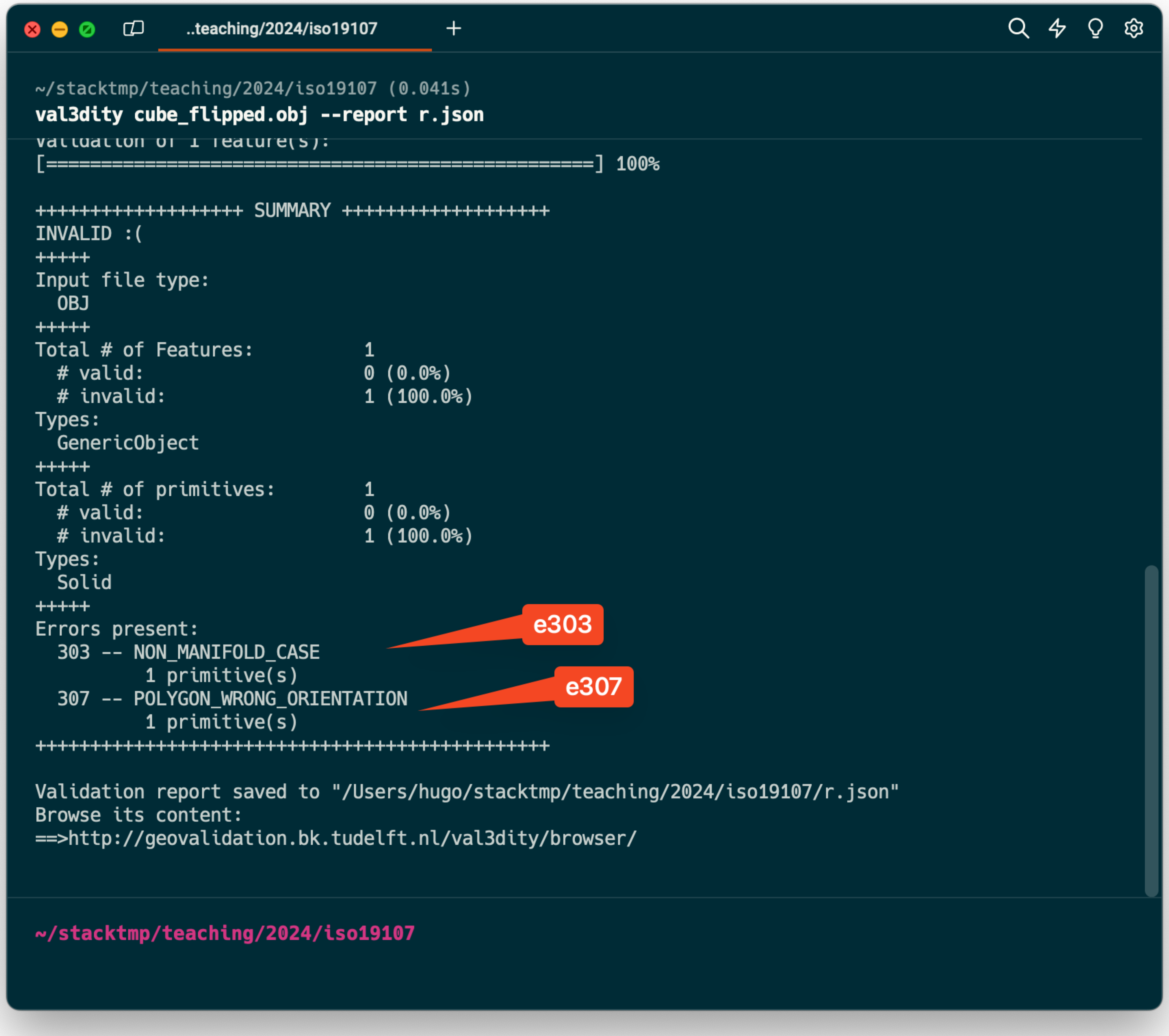
601 BUILDINGPARTS_OVERLAP

IndoorGML Objects

701 PRIMAL_CELLS_OVERLAP
702 DUAL_VERTEX_OUTSIDE_PRIMAL_CELL
703 PRIMAL_DUAL_XLINKS_ERROR
704 PRIMAL_DUAL_ADJACENCIES_INCONSISTENT

Others

901 INVALID_INPUT_FILE
902 EMPTY_PRIMITIVE
903 WRONG_INPUT_PARAMETERS
904 FORMAT_NOT_SUPPORTED
906 PRIMITIVE_NO_GEOMETRY
999 UNKNOWN_ERROR



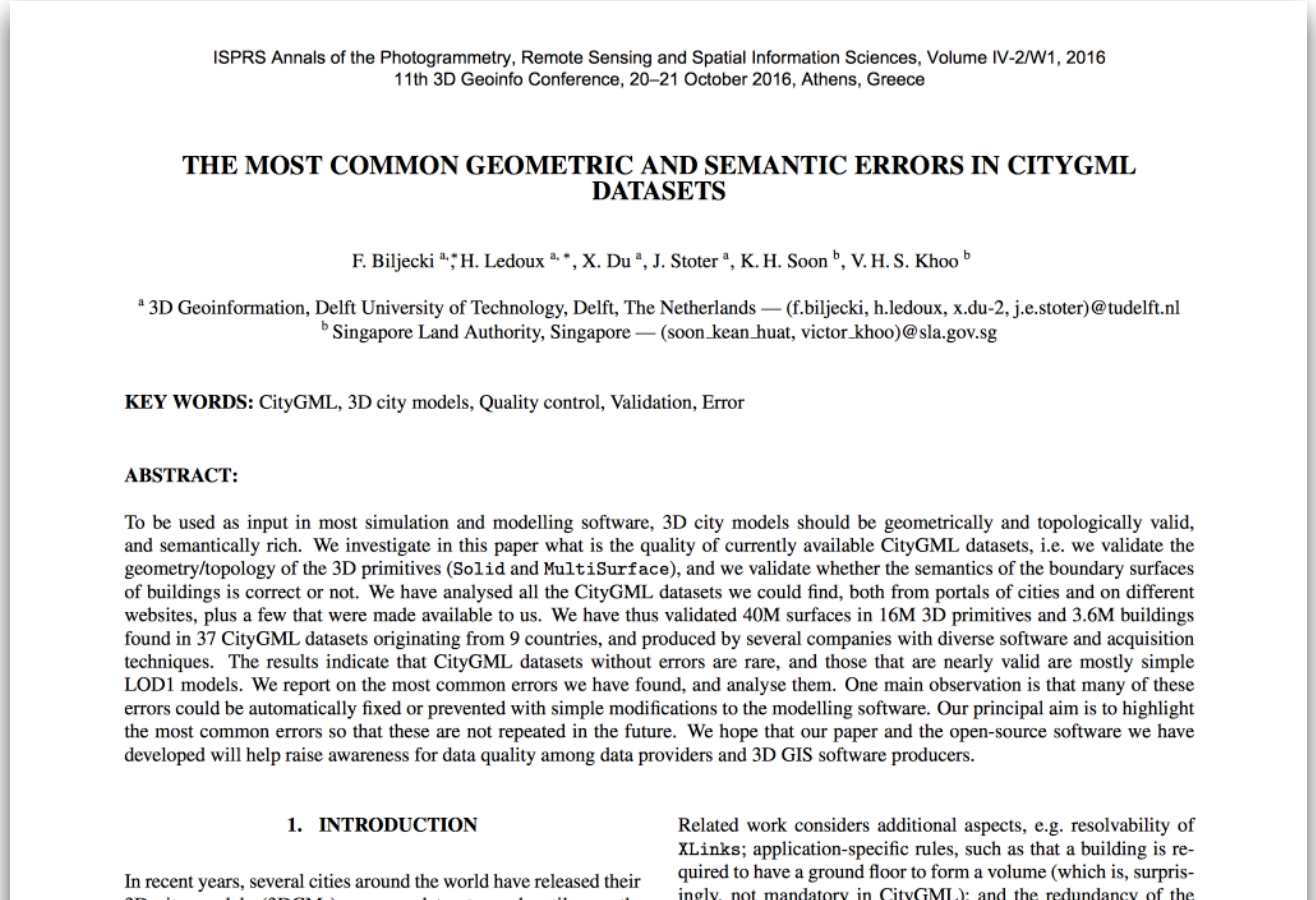
```
~/stacktmp/teaching/2024/iso19107 (0.041s)
val3dity cube_flipped.obj --report r.json
validation of 1 feature(s):
[==================================================] 100%

+++++++++++++++++++ SUMMARY +++++++++++++++++++
INVALID :(
+++++
Input file type:
  OBJ
+++++
Total # of Features:          1
  # valid:                    0 (0.0%)
  # invalid:                  1 (100.0%)
Types:
  GenericObject
+++++
Total # of primitives:        1
  # valid:                    0 (0.0%)
  # invalid:                  1 (100.0%)
Types:
  Solid
+++++
Errors present:
  303 -- NON_MANIFOLD_CASE
          1 primitive(s)
  307 -- POLYGON_WRONG_ORIENTATION
          1 primitive(s)
+++++++++++++++++++++++++++++++++++++++++++++++

Validation report saved to "/Users/hugo/stacktmp/teaching/2024/iso19107/r.json"
Browse its content:
==>http://geovalidation.bk.tudelft.nl/val3dity/browser/

~/stacktmp/teaching/2024/iso19107
```

Summary of results



ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume IV-2/W1, 2016
11th 3D Geoinfo Conference, 20–21 October 2016, Athens, Greece

THE MOST COMMON GEOMETRIC AND SEMANTIC ERRORS IN CITYGML DATASETS

F. Biljecki [a,*], H. Ledoux [a,*], X. Du [a], J. Stoter [a], K. H. Soon [b], V. H. S. Khoo [b]

[a] 3D Geoinformation, Delft University of Technology, Delft, The Netherlands — (f.biljecki, h.ledoux, x.du-2, j.e.stoter)@tudelft.nl
[b] Singapore Land Authority, Singapore — (soon_kean_huat, victor_khoo)@sla.gov.sg

KEY WORDS: CityGML, 3D city models, Quality control, Validation, Error

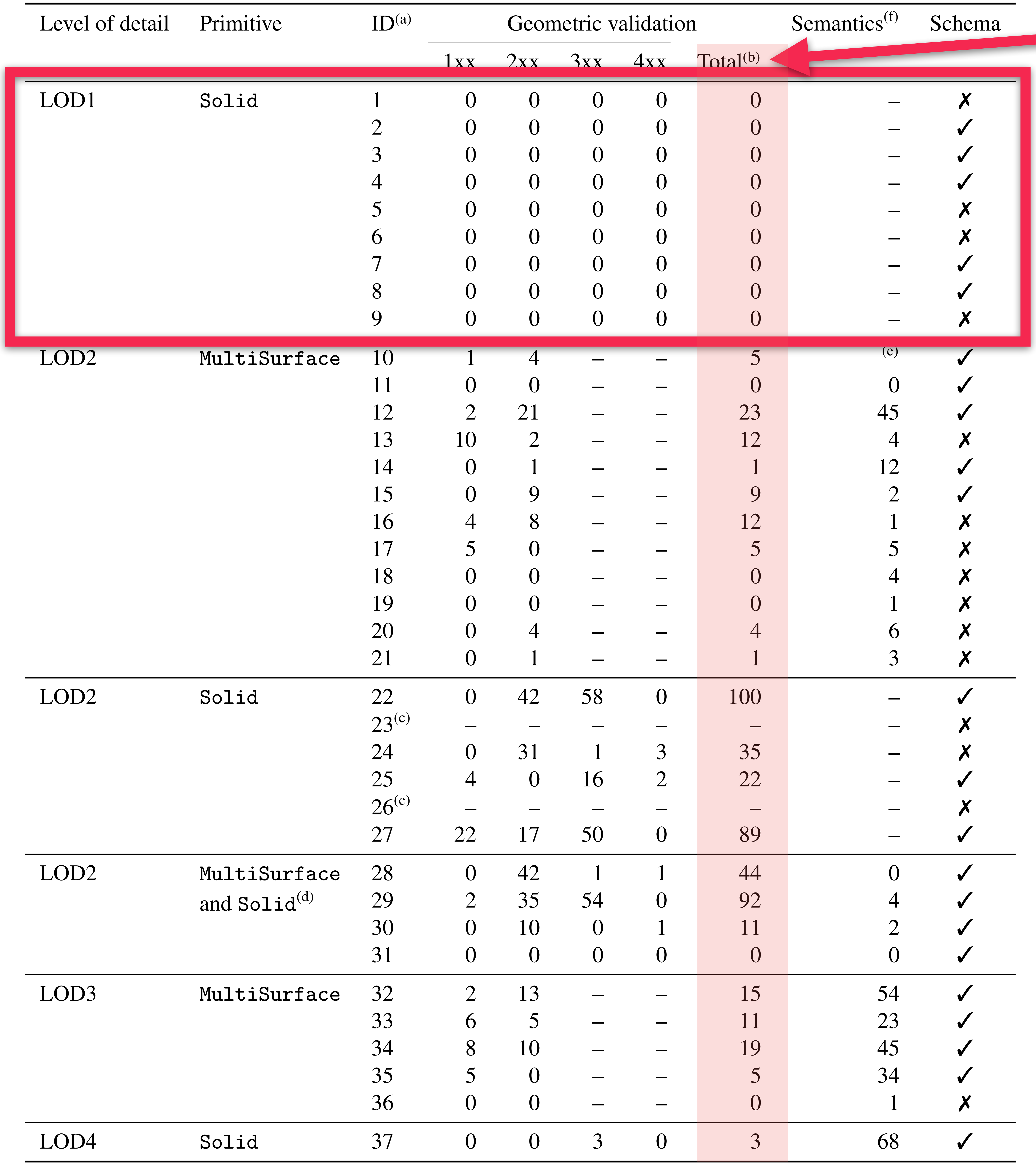
ABSTRACT:

To be used as input in most simulation and modelling software, 3D city models should be geometrically and topologically valid, and semantically rich. We investigate in this paper what is the quality of currently available CityGML datasets, i.e. we validate the geometry/topology of the 3D primitives (`Solid` and `MultiSurface`), and we validate whether the semantics of the boundary surfaces of buildings is correct or not. We have analysed all the CityGML datasets we could find, both from portals of cities and on different websites, plus a few that were made available to us. We have thus validated 40M surfaces in 16M 3D primitives and 3.6M buildings found in 37 CityGML datasets originating from 9 countries, and produced by several companies with diverse software and acquisition techniques. The results indicate that CityGML datasets without errors are rare, and those that are nearly valid are mostly simple LOD1 models. We report on the most common errors we have found, and analyse them. One main observation is that many of these errors could be automatically fixed or prevented with simple modifications to the modelling software. Our principal aim is to highlight the most common errors so that these are not repeated in the future. We hope that our paper and the open-source software we have developed will help raise awareness for data quality among data providers and 3D GIS software producers.

1. INTRODUCTION

In recent years, several cities around the world have released their

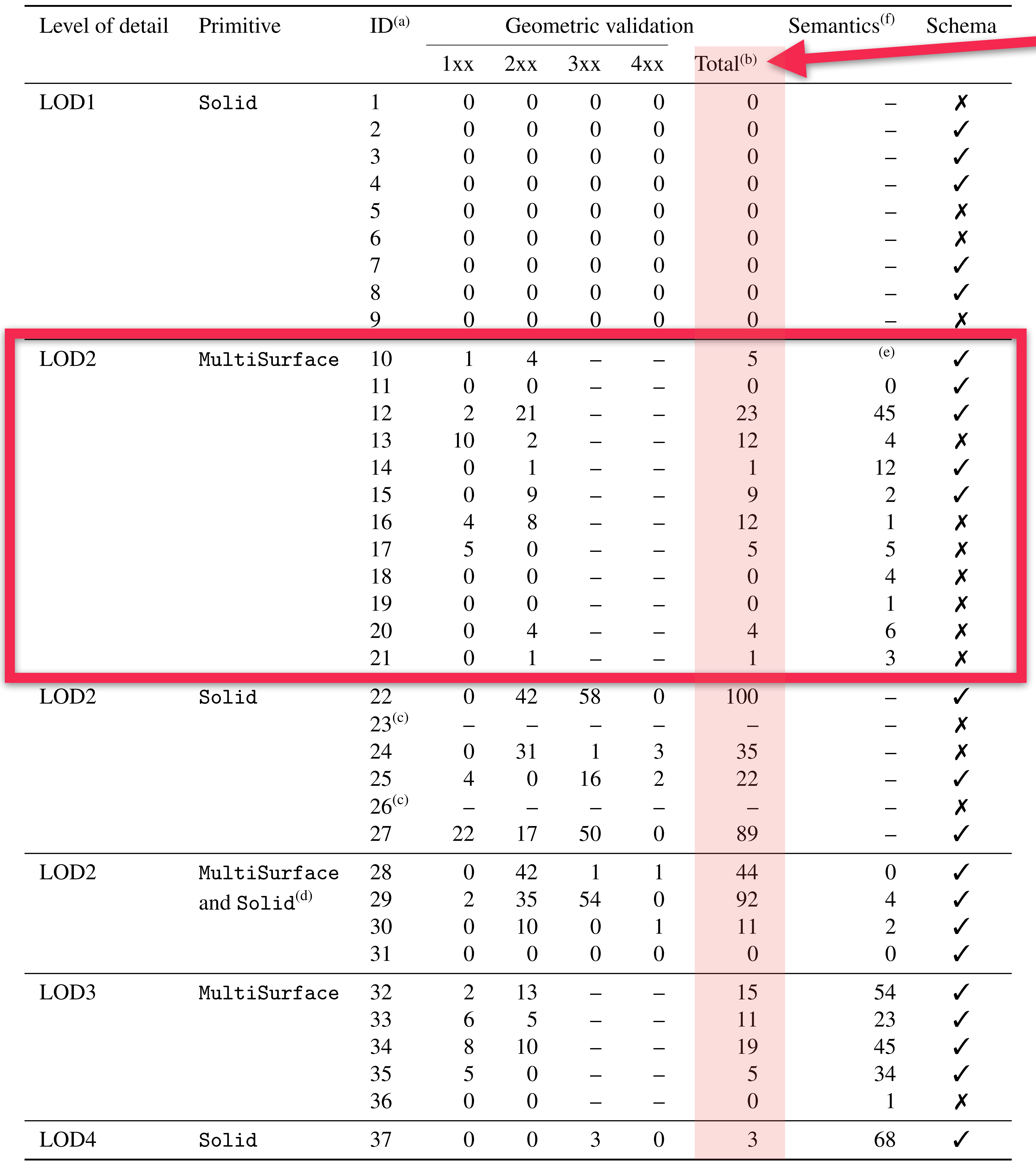
Related work considers additional aspects, e.g. resolvability of `XLinks`; application-specific rules, such as that a building is required to have a ground floor to form a volume (which is, surprisingly, not mandatory in CityGML); and the redundancy of the



Level of detail	Primitive	ID[a]	Geometric validation					Semantics[f]	Schema
			1xx	2xx	3xx	4xx	Total[b]		
LOD1	`Solid`	1	0	0	0	0	0	–	✗
		2	0	0	0	0	0	–	✓
		3	0	0	0	0	0	–	✓
		4	0	0	0	0	0	–	✓
		5	0	0	0	0	0	–	✗
		6	0	0	0	0	0	–	✗
		7	0	0	0	0	0	–	✓
		8	0	0	0	0	0	–	✓
		9	0	0	0	0	0	–	✗
LOD2	`MultiSurface`	10	1	4	–	–	5	[e]	✓
		11	0	0	–	–	0	0	✓
		12	2	21	–	–	23	45	✓
		13	10	2	–	–	12	4	✗
		14	0	1	–	–	1	12	✓
		15	0	9	–	–	9	2	✓
		16	4	8	–	–	12	1	✗
		17	5	0	–	–	5	5	✗
		18	0	0	–	–	0	4	✗
		19	0	0	–	–	0	1	✗
		20	0	4	–	–	4	6	✗
		21	0	1	–	–	1	3	✗
LOD2	`Solid`	22	0	42	58	0	100	–	✓
		23[c]	–	–	–	–	–	–	✗
		24	0	31	1	3	35	–	✗
		25	4	0	16	2	22	–	✓
		26[c]	–	–	–	–	–	–	✗
		27	22	17	50	0	89	–	✓
LOD2	`MultiSurface` and `Solid`[d]	28	0	42	1	1	44	0	✓
		29	2	35	54	0	92	4	✓
		30	0	10	0	1	11	2	✓
		31	0	0	0	0	0	0	✓
LOD3	`MultiSurface`	32	2	13	–	–	15	54	✓
		33	6	5	–	–	11	23	✓
		34	8	10	–	–	19	45	✓
		35	5	0	–	–	5	34	✓
		36	0	0	–	–	0	1	✗
LOD4	`Solid`	37	0	0	3	0	3	68	✓

% of invalid

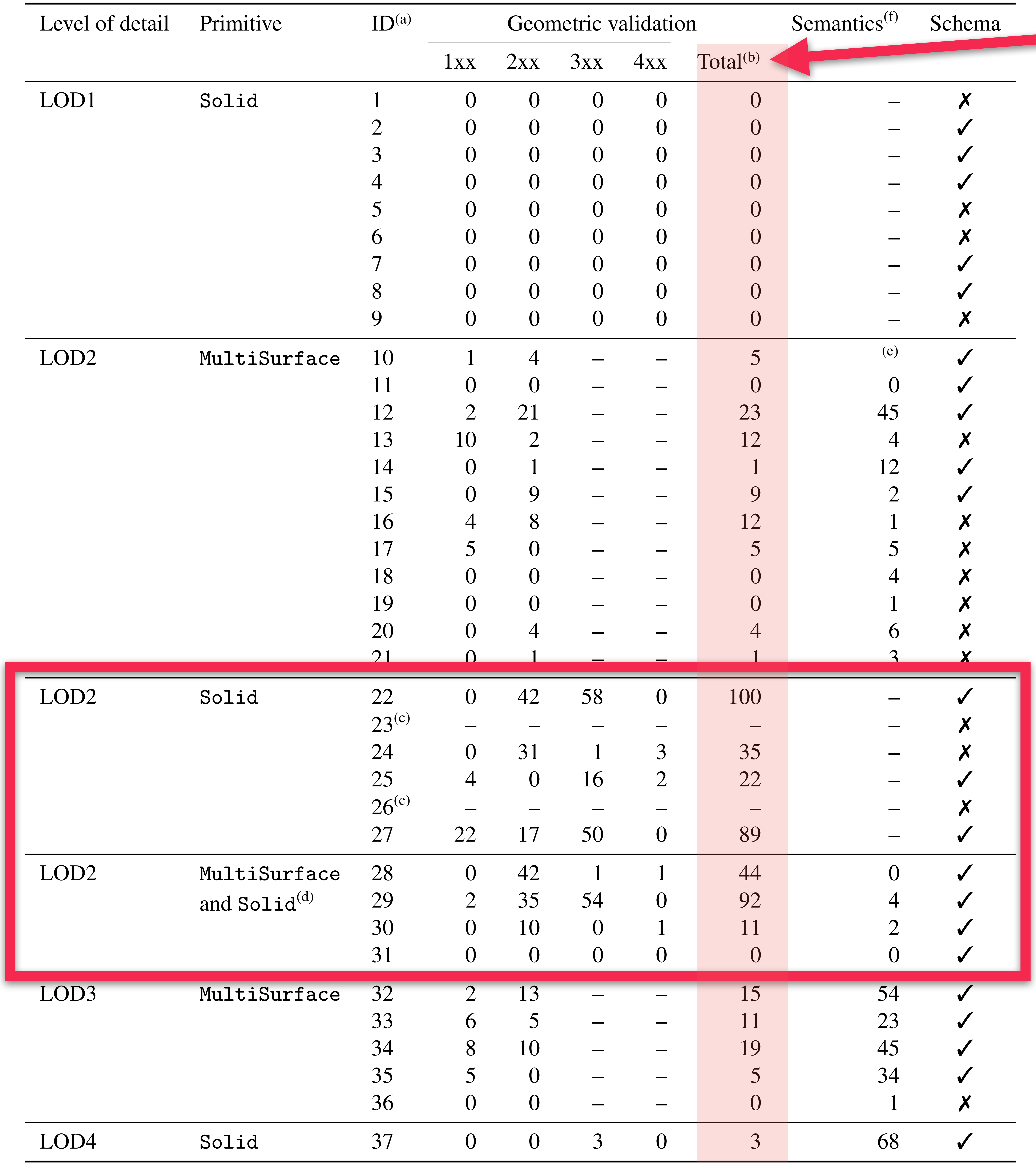
actually all between 99.5% and 99.9%



Level of detail	Primitive	ID[a]	Geometric validation					Semantics[f]	Schema
			1xx	2xx	3xx	4xx	Total[b]		
LOD1	Solid	1	0	0	0	0	0	–	✗
		2	0	0	0	0	0	–	✓
		3	0	0	0	0	0	–	✓
		4	0	0	0	0	0	–	✓
		5	0	0	0	0	0	–	✗
		6	0	0	0	0	0	–	✗
		7	0	0	0	0	0	–	✓
		8	0	0	0	0	0	–	✓
		9	0	0	0	0	0	–	✗
LOD2	MultiSurface	10	1	4	–	–	5	[e]	✓
		11	0	0	–	–	0	0	✓
		12	2	21	–	–	23	45	✓
		13	10	2	–	–	12	4	✗
		14	0	1	–	–	1	12	✓
		15	0	9	–	–	9	2	✓
		16	4	8	–	–	12	1	✗
		17	5	0	–	–	5	5	✗
		18	0	0	–	–	0	4	✗
		19	0	0	–	–	0	1	✗
		20	0	4	–	–	4	6	✗
		21	0	1	–	–	1	3	✗
LOD2	Solid	22	0	42	58	0	100	–	✓
		23[c]	–	–	–	–	–	–	✗
		24	0	31	1	3	35	–	✗
		25	4	0	16	2	22	–	✓
		26[c]	–	–	–	–	–	–	✗
		27	22	17	50	0	89	–	✓
LOD2	MultiSurface and Solid[d]	28	0	42	1	1	44	0	✓
		29	2	35	54	0	92	4	✓
		30	0	10	0	1	11	2	✓
		31	0	0	0	0	0	0	✓
LOD3	MultiSurface	32	2	13	–	–	15	54	✓
		33	6	5	–	–	11	23	✓
		34	8	10	–	–	19	45	✓
		35	5	0	–	–	5	34	✓
		36	0	0	–	–	0	1	✗
LOD4	Solid	37	0	0	3	0	3	68	✓

% of invalid

rather good, but it’s “easy”



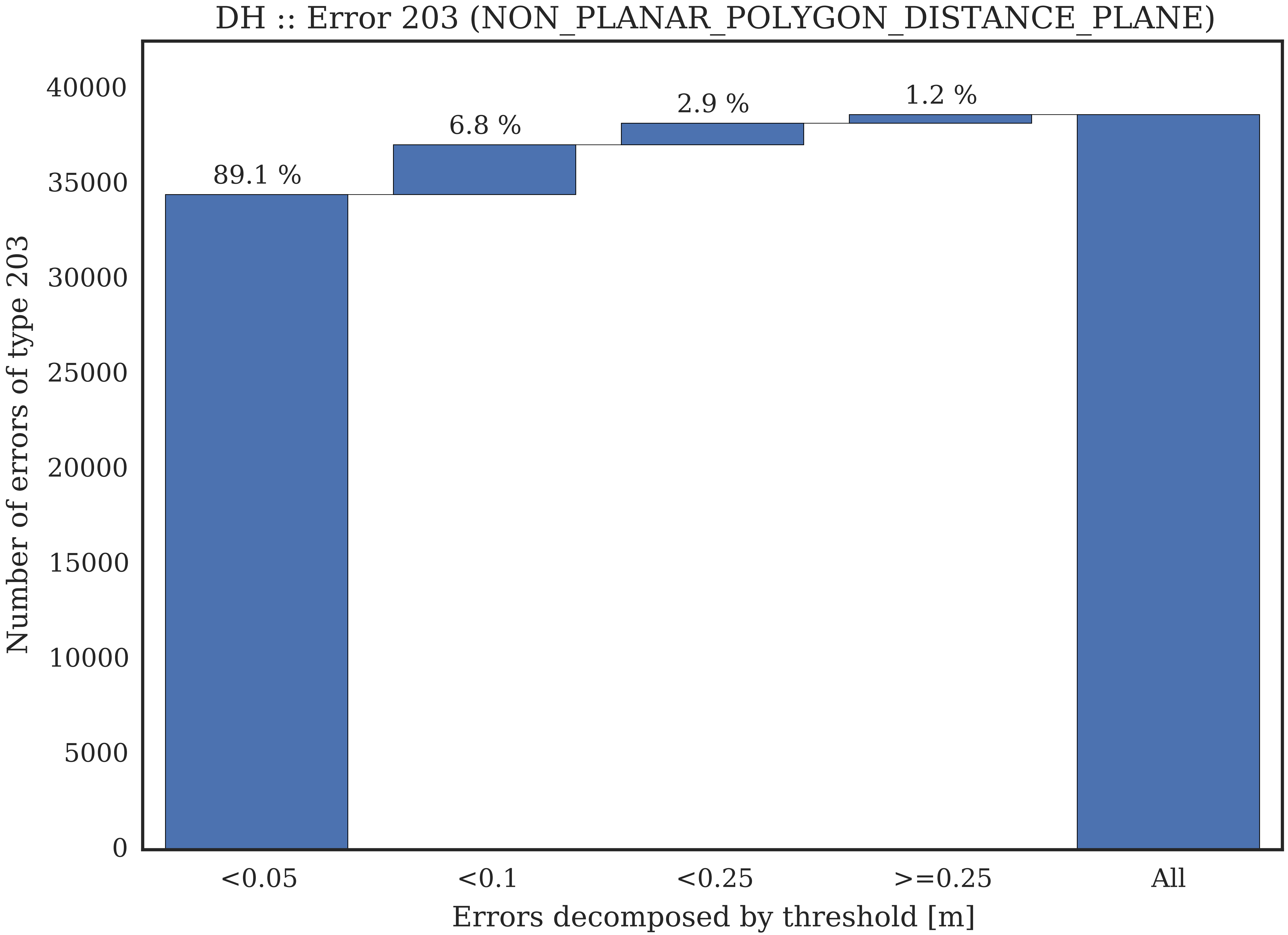
Level of detail	Primitive	ID[a]	Geometric validation					Semantics[f]	Schema
			1xx	2xx	3xx	4xx	Total[b]		
LOD1	Solid	1	0	0	0	0	0	–	✗
		2	0	0	0	0	0	–	✓
		3	0	0	0	0	0	–	✓
		4	0	0	0	0	0	–	✓
		5	0	0	0	0	0	–	✗
		6	0	0	0	0	0	–	✗
		7	0	0	0	0	0	–	✓
		8	0	0	0	0	0	–	✓
		9	0	0	0	0	0	–	✗
LOD2	MultiSurface	10	1	4	–	–	5	[e]	✓
		11	0	0	–	–	0	0	✓
		12	2	21	–	–	23	45	✓
		13	10	2	–	–	12	4	✗
		14	0	1	–	–	1	12	✓
		15	0	9	–	–	9	2	✓
		16	4	8	–	–	12	1	✗
		17	5	0	–	–	5	5	✗
		18	0	0	–	–	0	4	✗
		19	0	0	–	–	0	1	✗
		20	0	4	–	–	4	6	✗
		21	0	1	–	–	1	3	✗
LOD2	Solid	22	0	42	58	0	100	–	✓
		23[c]	–	–	–	–	–	–	✗
		24	0	31	1	3	35	–	✗
		25	4	0	16	2	22	–	✓
		26[c]	–	–	–	–	–	–	✗
		27	22	17	50	0	89	–	✓
LOD2	MultiSurface and Solid[d]	28	0	42	1	1	44	0	✓
		29	2	35	54	0	92	4	✓
		30	0	10	0	1	11	2	✓
		31	0	0	0	0	0	0	✓
LOD3	MultiSurface	32	2	13	–	–	15	54	✓
		33	6	5	–	–	11	23	✓
		34	8	10	–	–	19	45	✓
		35	5	0	–	–	5	34	✓
		36	0	0	–	–	0	1	✗
LOD4	Solid	37	0	0	3	0	3	68	✓

% of invalid

some datasets couldn't be read

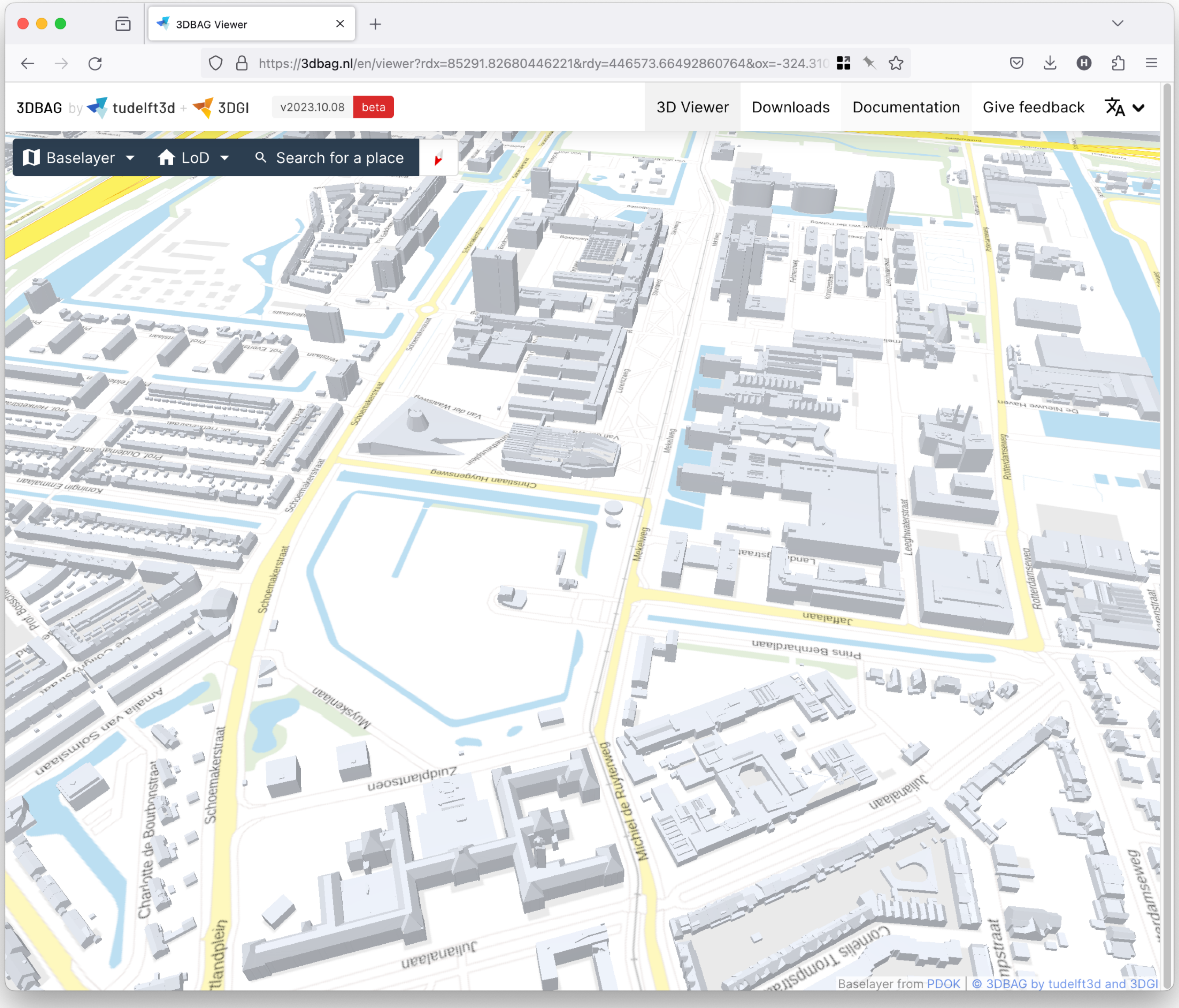
high % of invalid

Most common error: non-planar surface

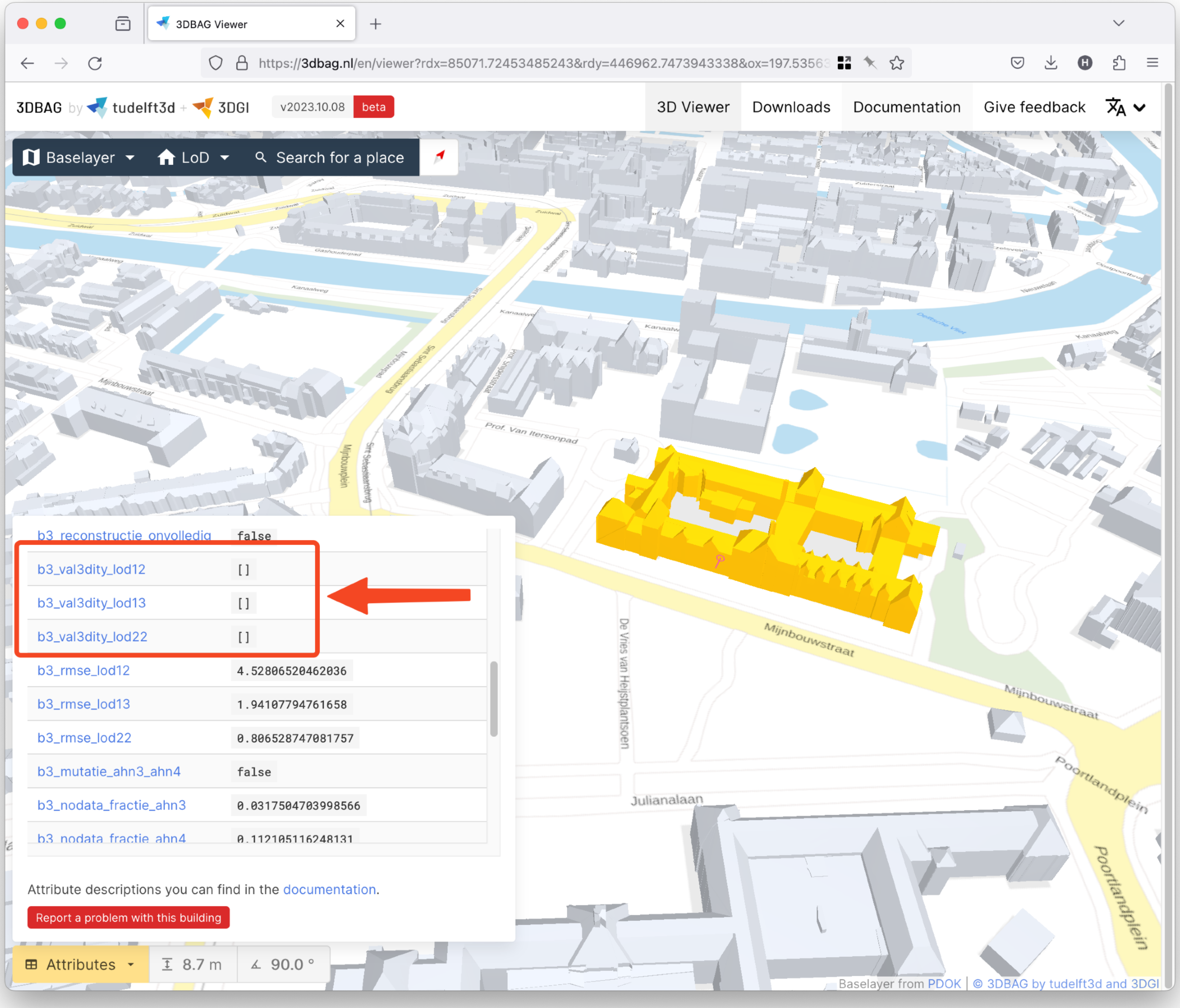


3DBAG is ~99% geometrically valid 💪 🚀

Lesson 4.2
this Wednesday



3DBAG: you can see the validity



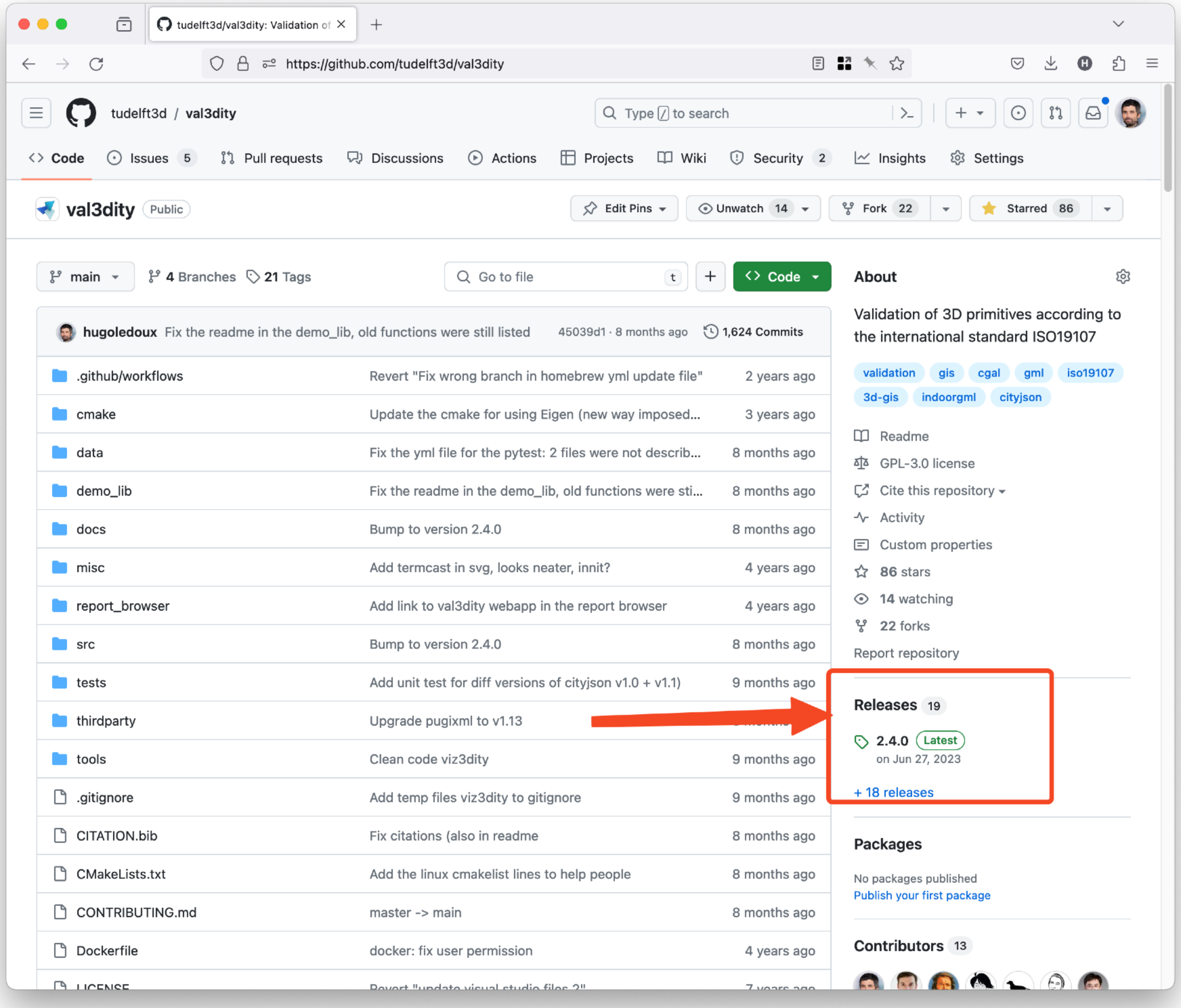
b3_reconstructie_onvolledig	false
b3_val3dity_lod12	[]
b3_val3dity_lod13	[]
b3_val3dity_lod22	[]
b3_rmse_lod12	4.52806520462036
b3_rmse_lod13	1.94107794761658
b3_rmse_lod22	0.806528747081757
b3_mutatie_ahn3_ahn4	false
b3_nodata_fractie_ahn3	0.0317504703998566
b3_nodata_fractie_ahn4	0.112105116248131

Attribute descriptions you can find in the documentation.

Report a problem with this building

val3dity is integrated in the process

How to use val3dity?



hugoledoux Fix the readme in the demo_lib, old functions were still listed 45039d1 · 8 months ago 1,624 Commits

.github/workflows	Revert "Fix wrong branch in homebrew yml update file"	2 years ago
cmake	Update the cmake for using Eigen (new way imposed...	3 years ago
data	Fix the yml file for the pytest: 2 files were not describ...	8 months ago
demo_lib	Fix the readme in the demo_lib, old functions were sti...	8 months ago
docs	Bump to version 2.4.0	8 months ago
misc	Add termcast in svg, looks neater, innit?	4 years ago
report_browser	Add link to val3dity webapp in the report browser	4 years ago
src	Bump to version 2.4.0	8 months ago
tests	Add unit test for diff versions of cityjson v1.0 + v1.1)	9 months ago
thirdparty	Upgrade pugixml to v1.13	
tools	Clean code viz3dity	9 months ago
.gitignore	Add temp files viz3dity to gitignore	9 months ago
CITATION.bib	Fix citations (also in readme	8 months ago
CMakeLists.txt	Add the linux cmakelist lines to help people	8 months ago
CONTRIBUTING.md	master -> main	8 months ago
Dockerfile	docker: fix user permission	4 years ago
LICENSE	Revert "update visual studio files 2"	7 years ago

Validation of 3D primitives according to the international standard ISO19107

validation gis cgal gml iso19107 3d-gis indoorgml cityjson

Readme
GPL-3.0 license
Cite this repository
Activity
Custom properties
86 stars
14 watching
22 forks
Report repository

Releases 19

2.4.0 Latest
on Jun 27, 2023

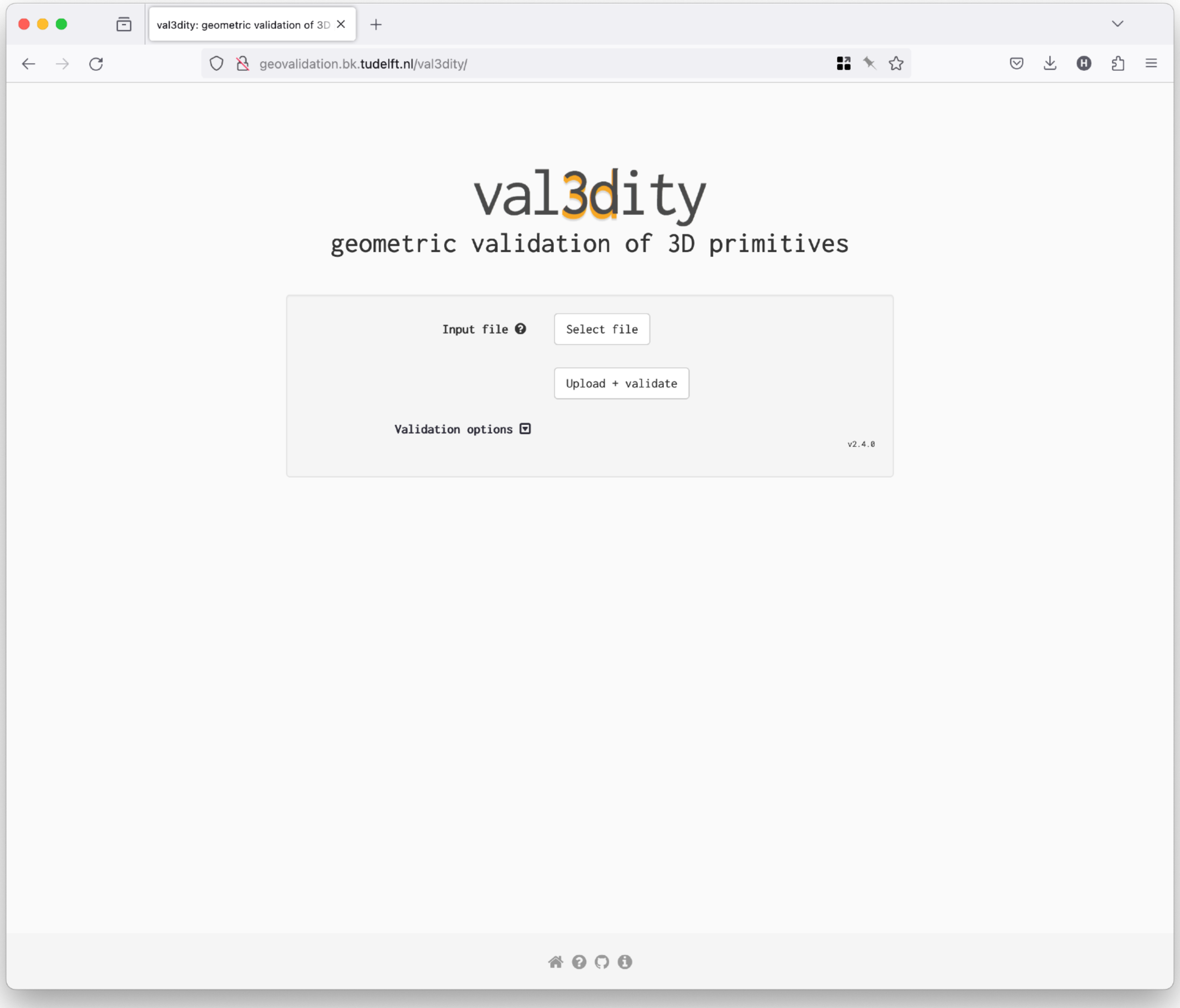
+ 18 releases

Packages

No packages published
Publish your first package

Contributors 13

CLI
(compile or download Window binary)


val3dity: geometric validation of 3D
geovalidation.bk.tudelft.nl/val3dity/
Select file
Upload + validate

val3dity
geometric validation of 3D primitives

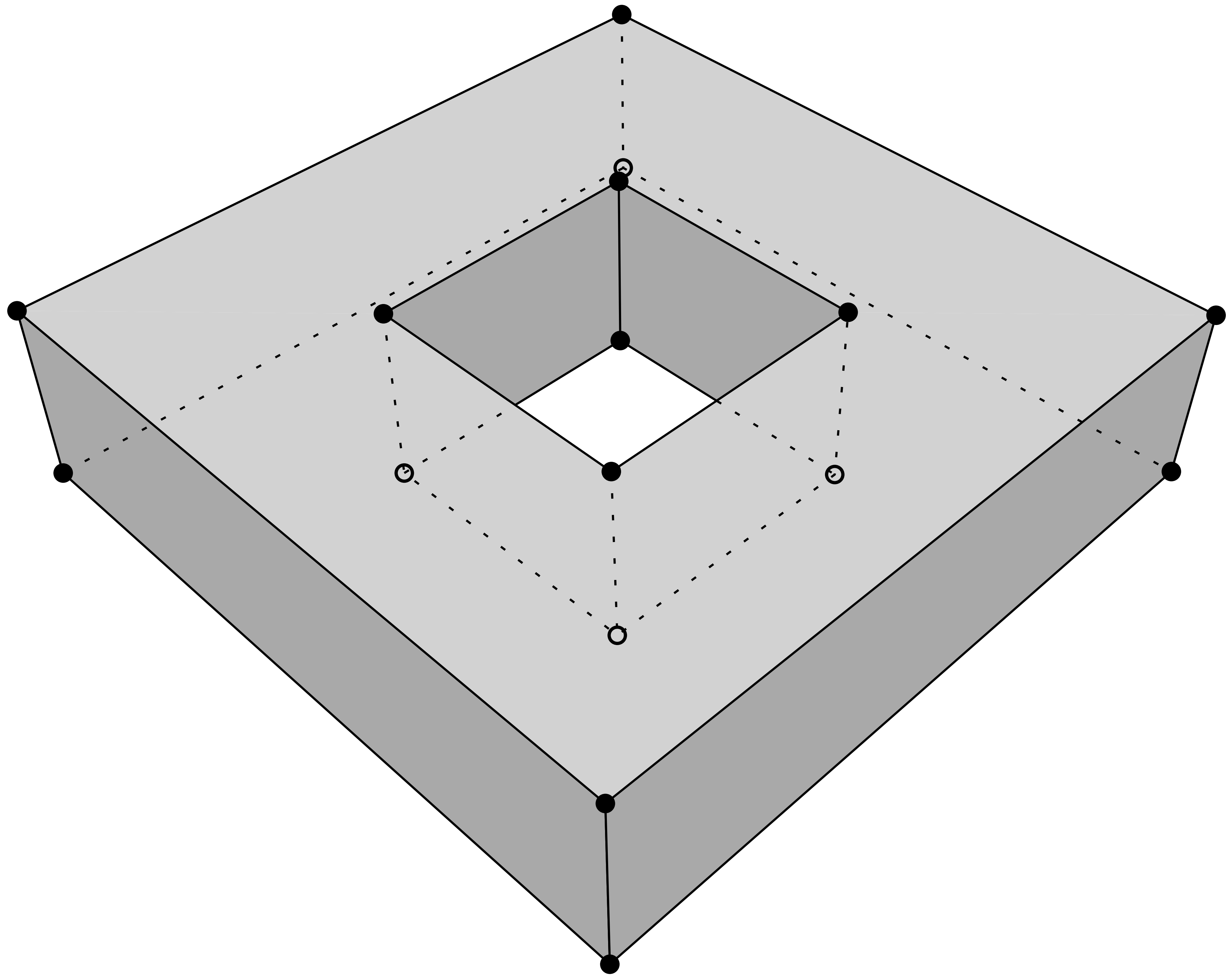
Input file

Validation options

v2.4.0

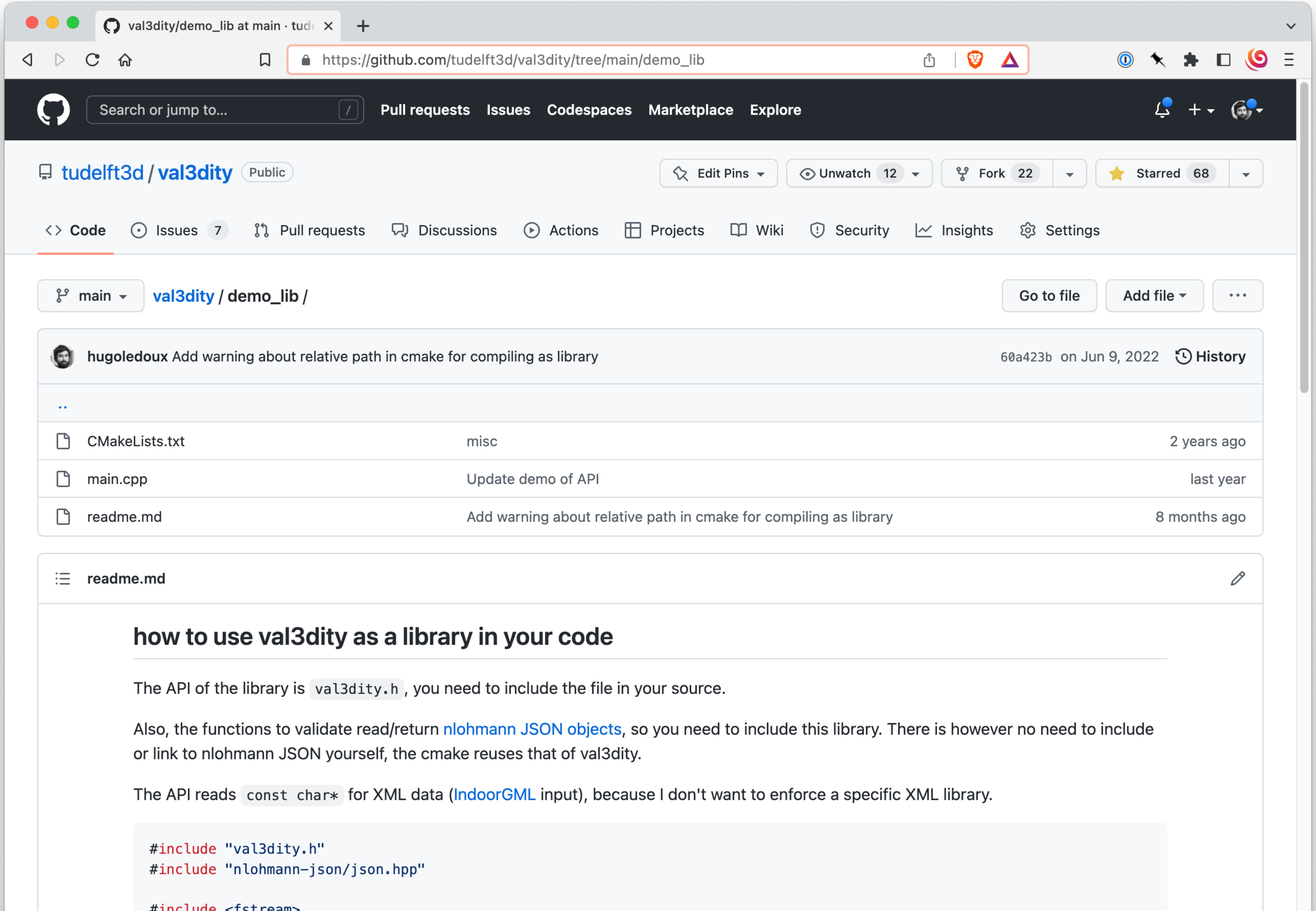
web-application

Quiz time!



1. How many interior shells does this “squared torus” have?
2. How many surfaces form its outer shell?
3. List them

Using val3dity as a library in C++



hugoledoux Add warning about relative path in cmake for compiling as library 60a423b on Jun 9, 2022

..

CMakeLists.txt	misc	2 years ago
main.cpp	Update demo of API	last year
readme.md	Add warning about relative path in cmake for compiling as library	8 months ago

readme.md

how to use val3dity as a library in your code

The API of the library is `val3dity.h`, you need to include the file in your source.

Also, the functions to validate read/return nlohmann JSON objects, so you need to include this library. There is however no need to include or link to nlohmann JSON yourself, the cmake reuses that of val3dity.

The API reads `const char*` for XML data (IndoorGML input), because I don't want to enforce a specific XML library.

```
#include "val3dity.h"
#include "nlohmann-json/json.hpp"

#include <fstream>
```