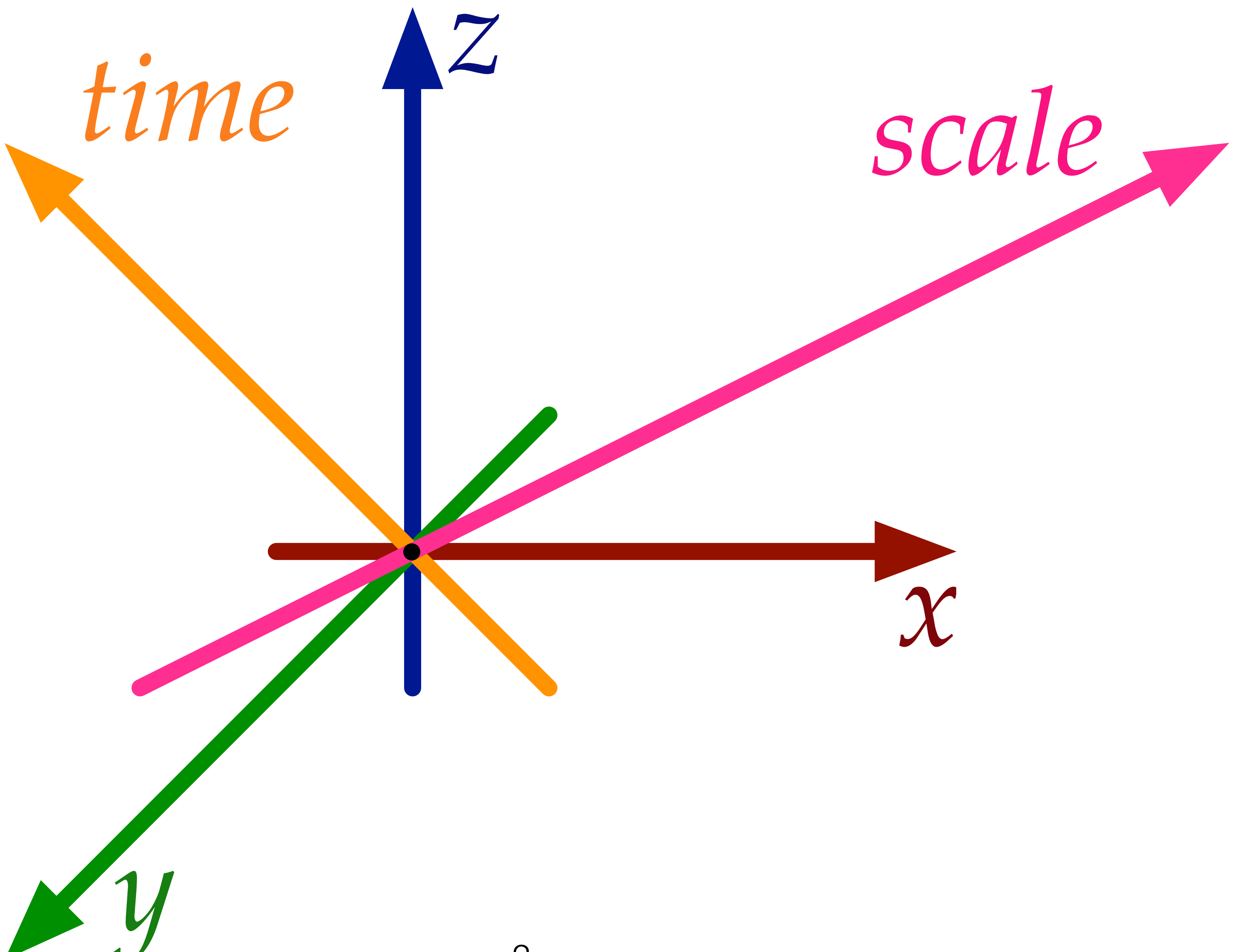
4D visualisation and GIS-BIM integration

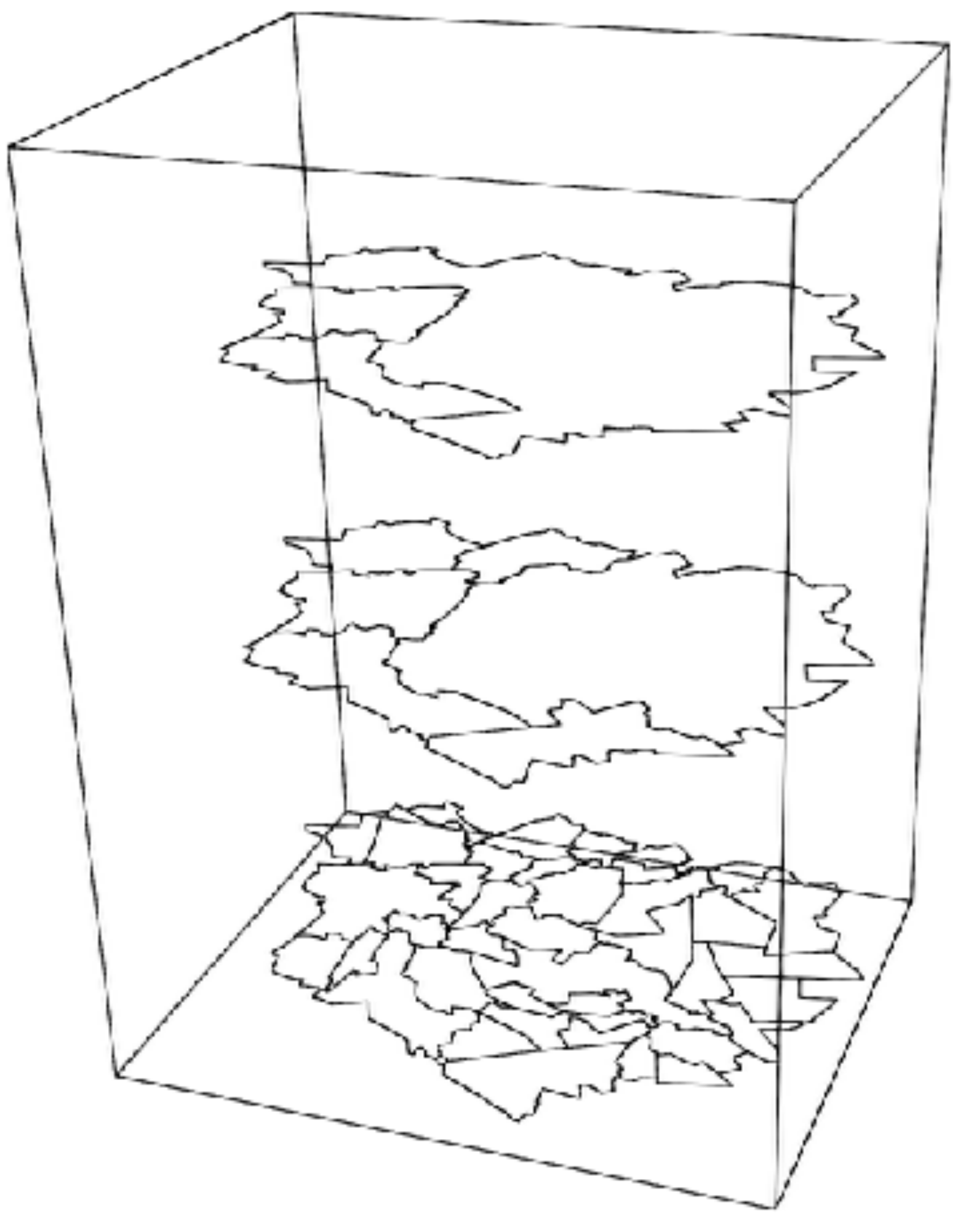
Ken Arroyo Ohori

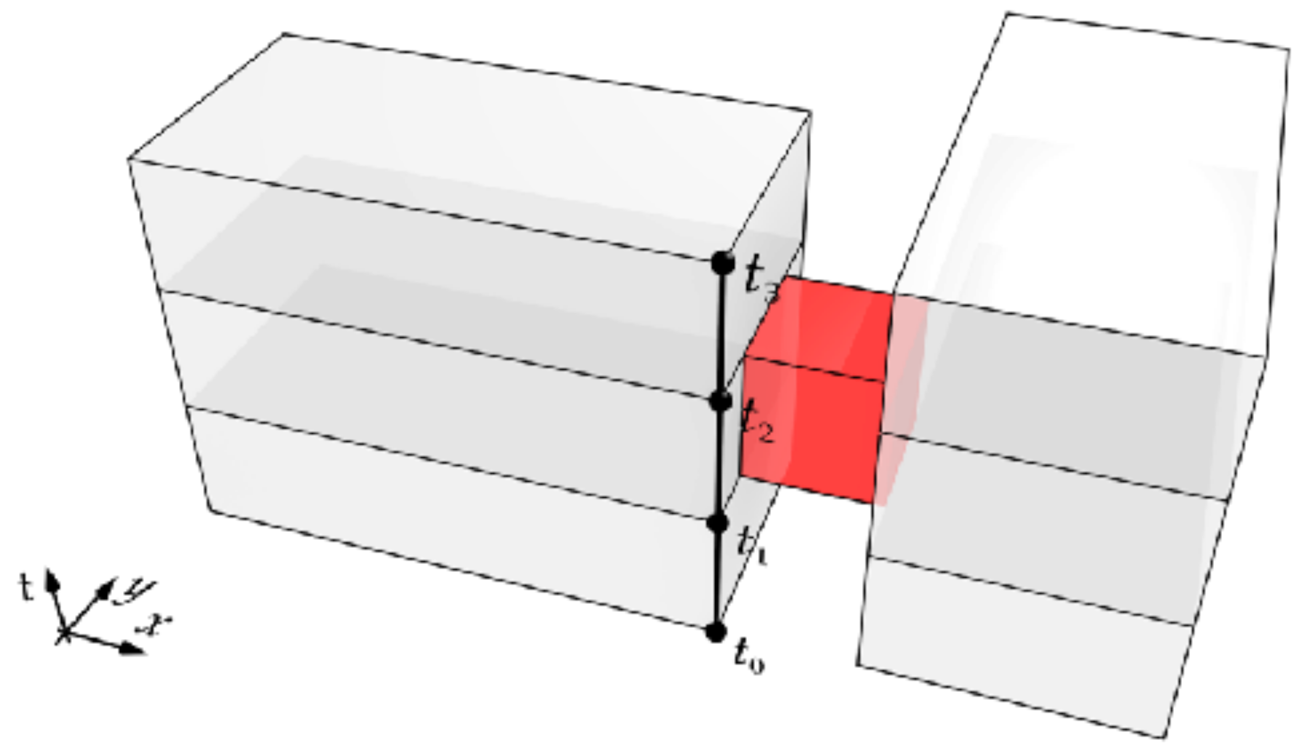
3D talk, 4.5.2017

Higher-dimensional modelling



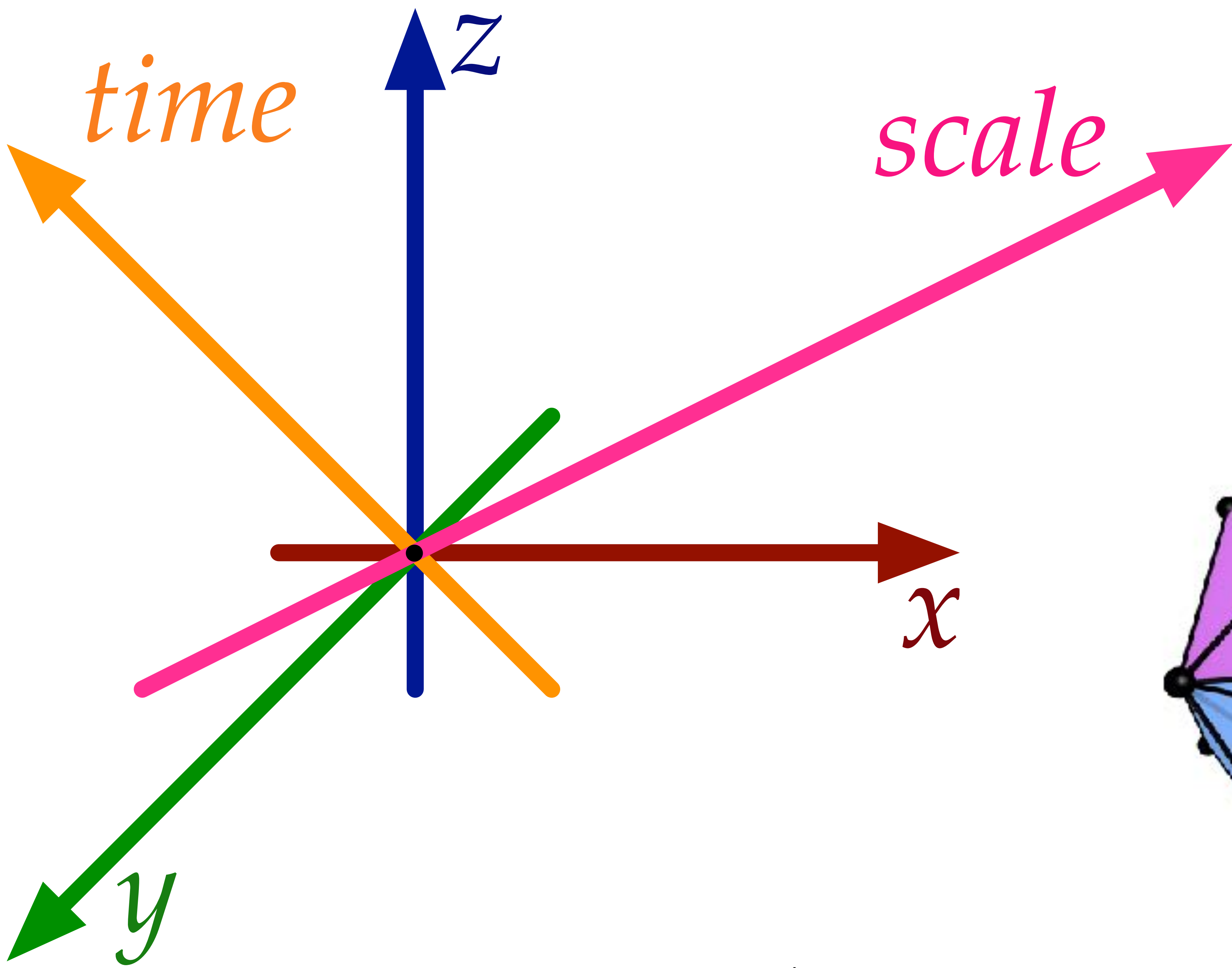
Higher-dimensional modelling

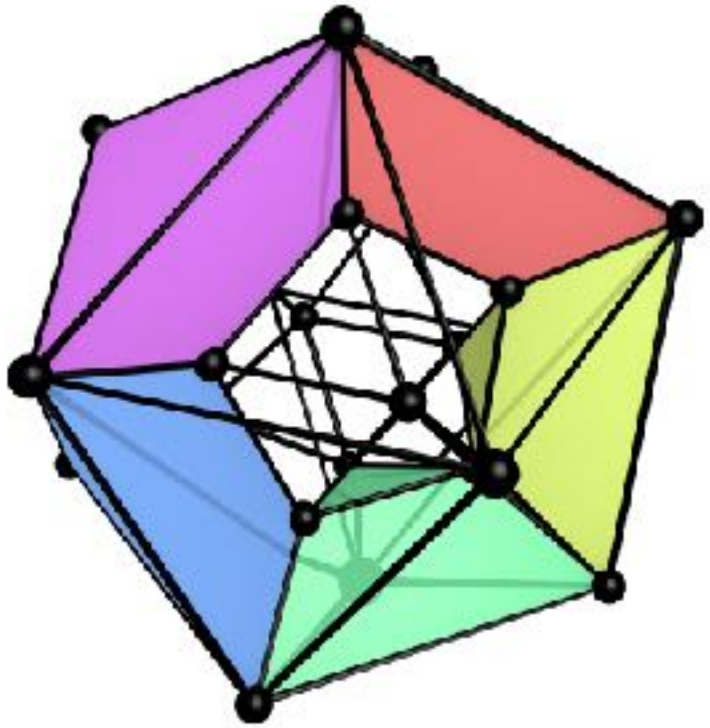




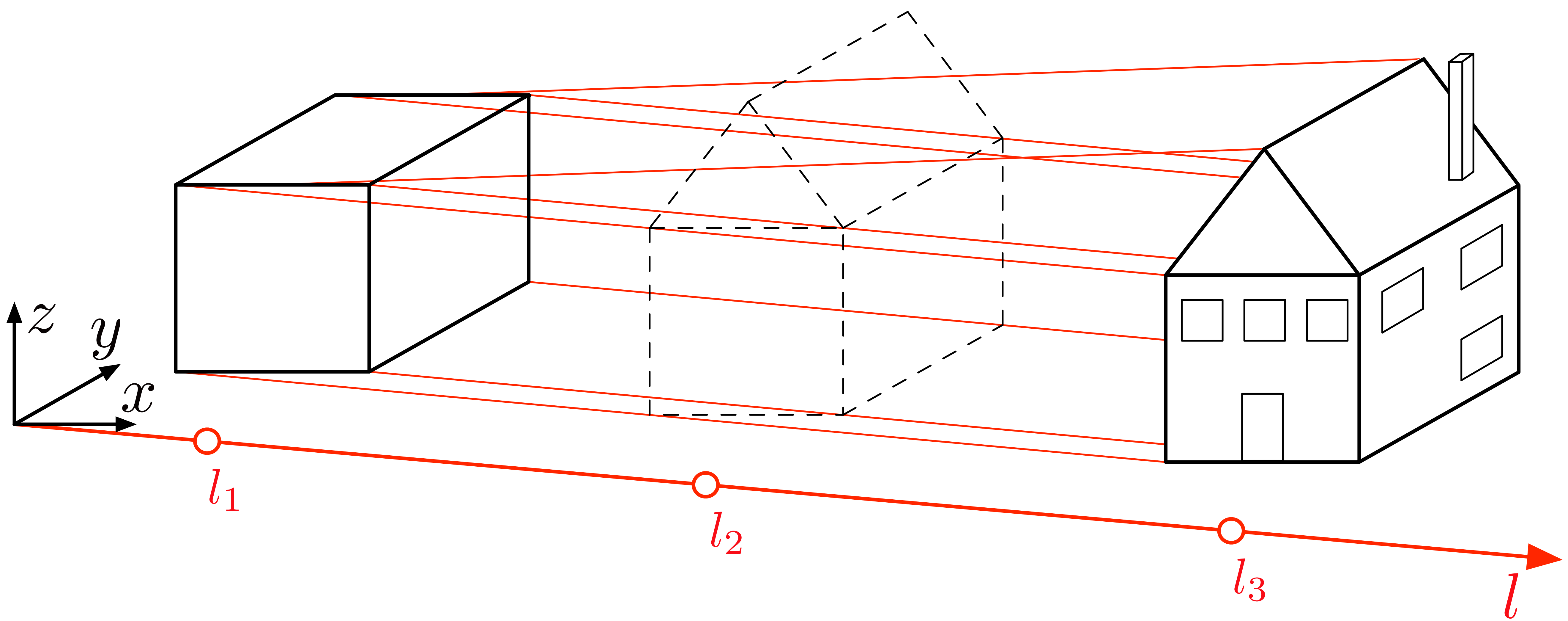
Meijers, 2011

Higher-dimensional modelling

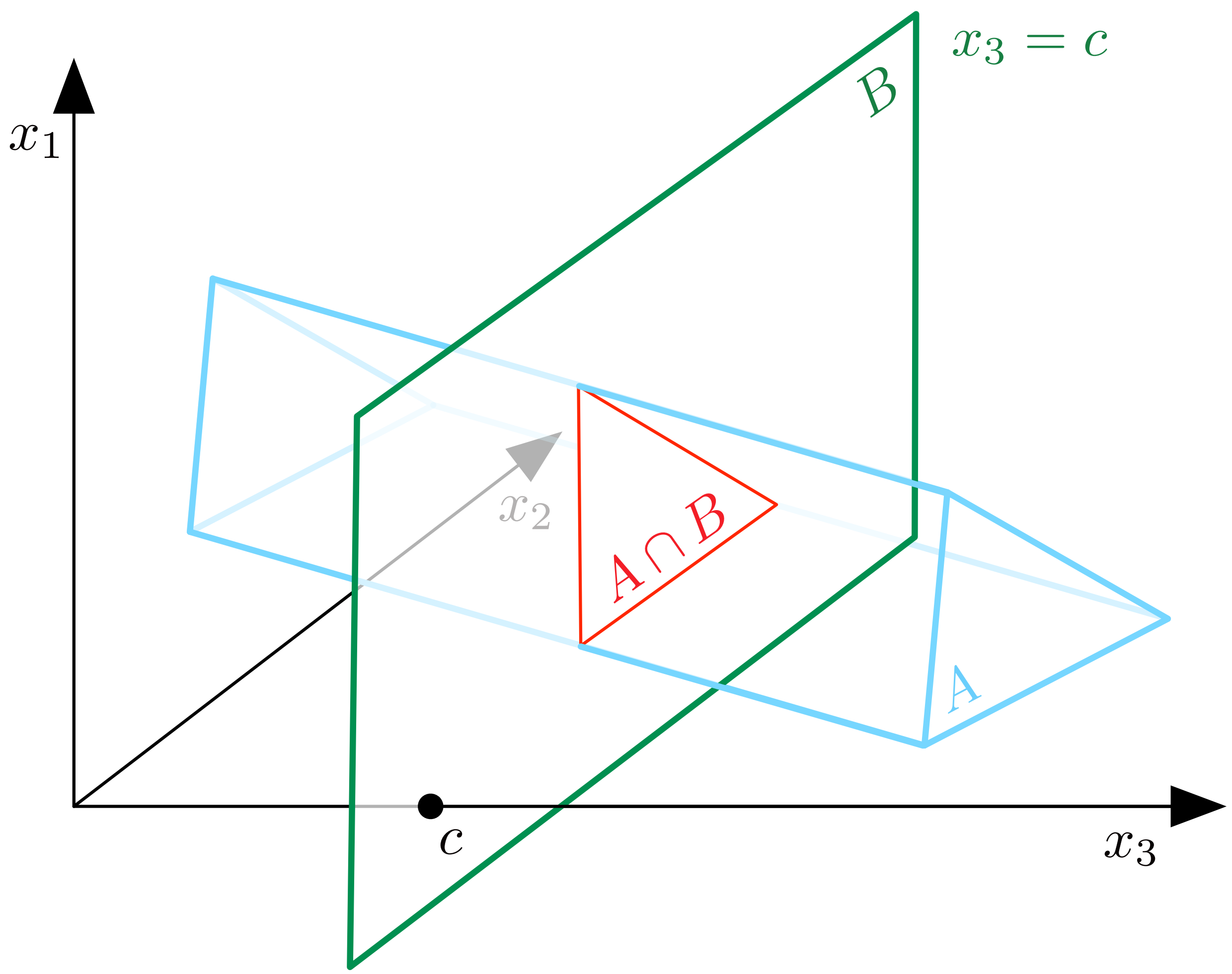


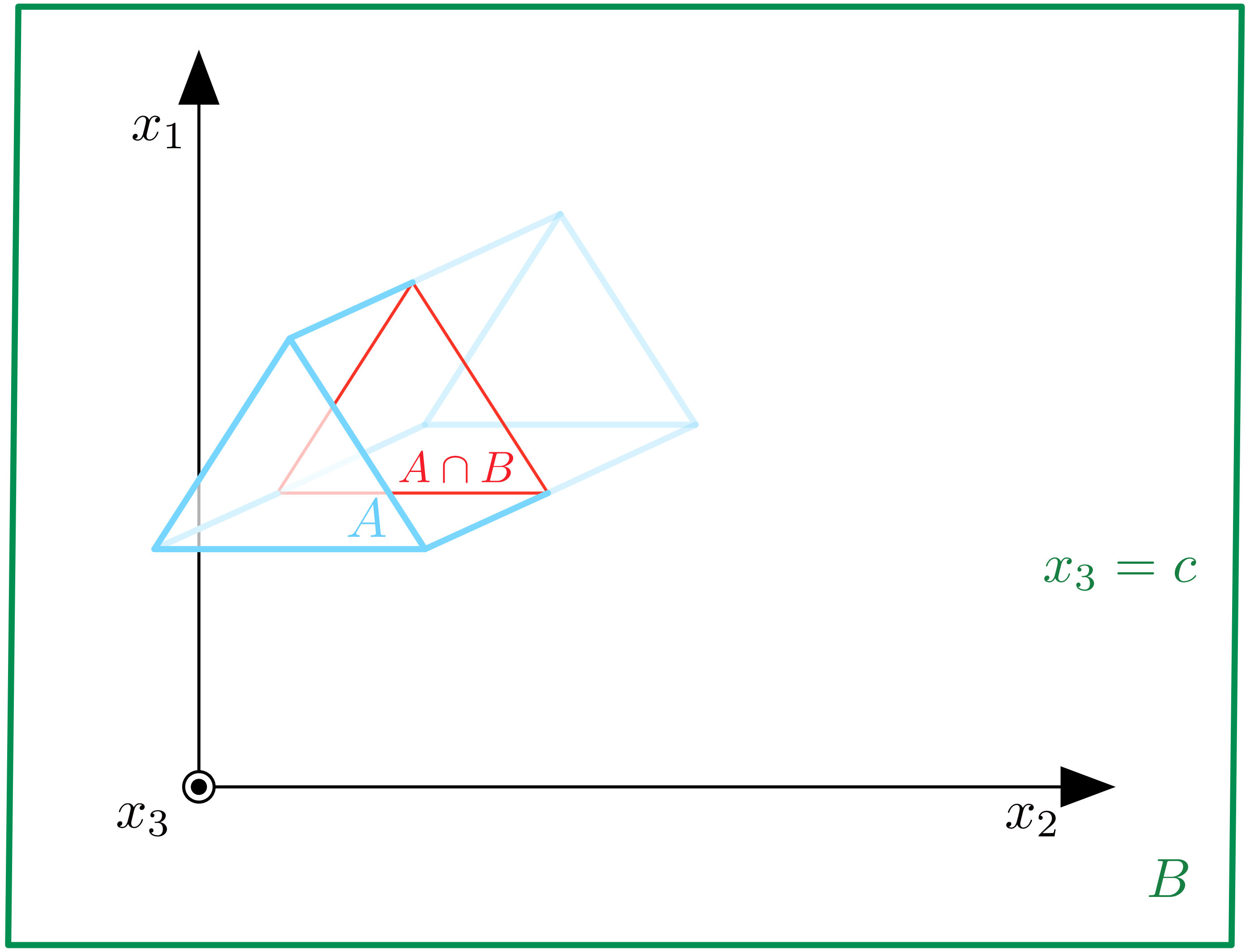


Slicing (conceptually)



Slicing as two problems

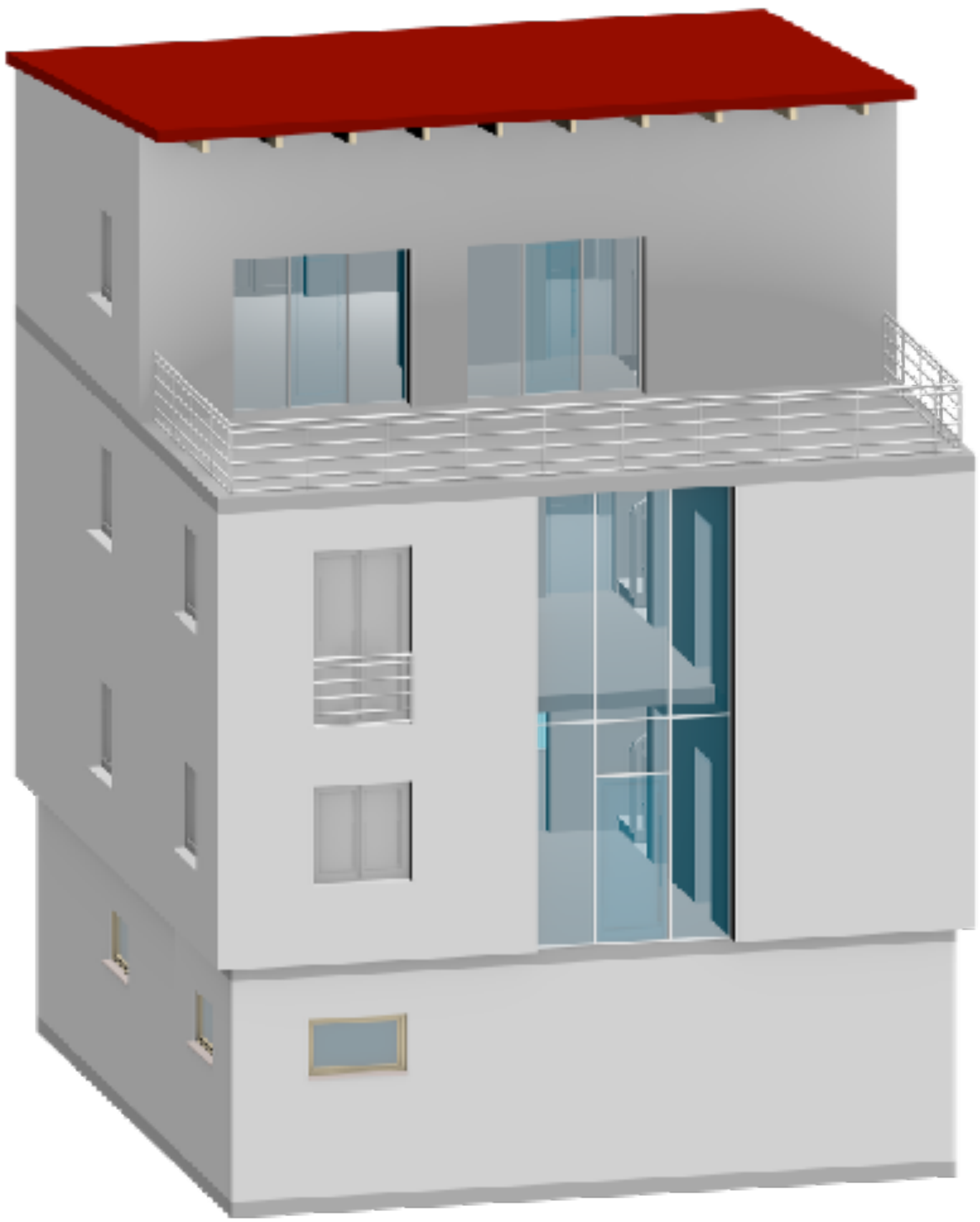


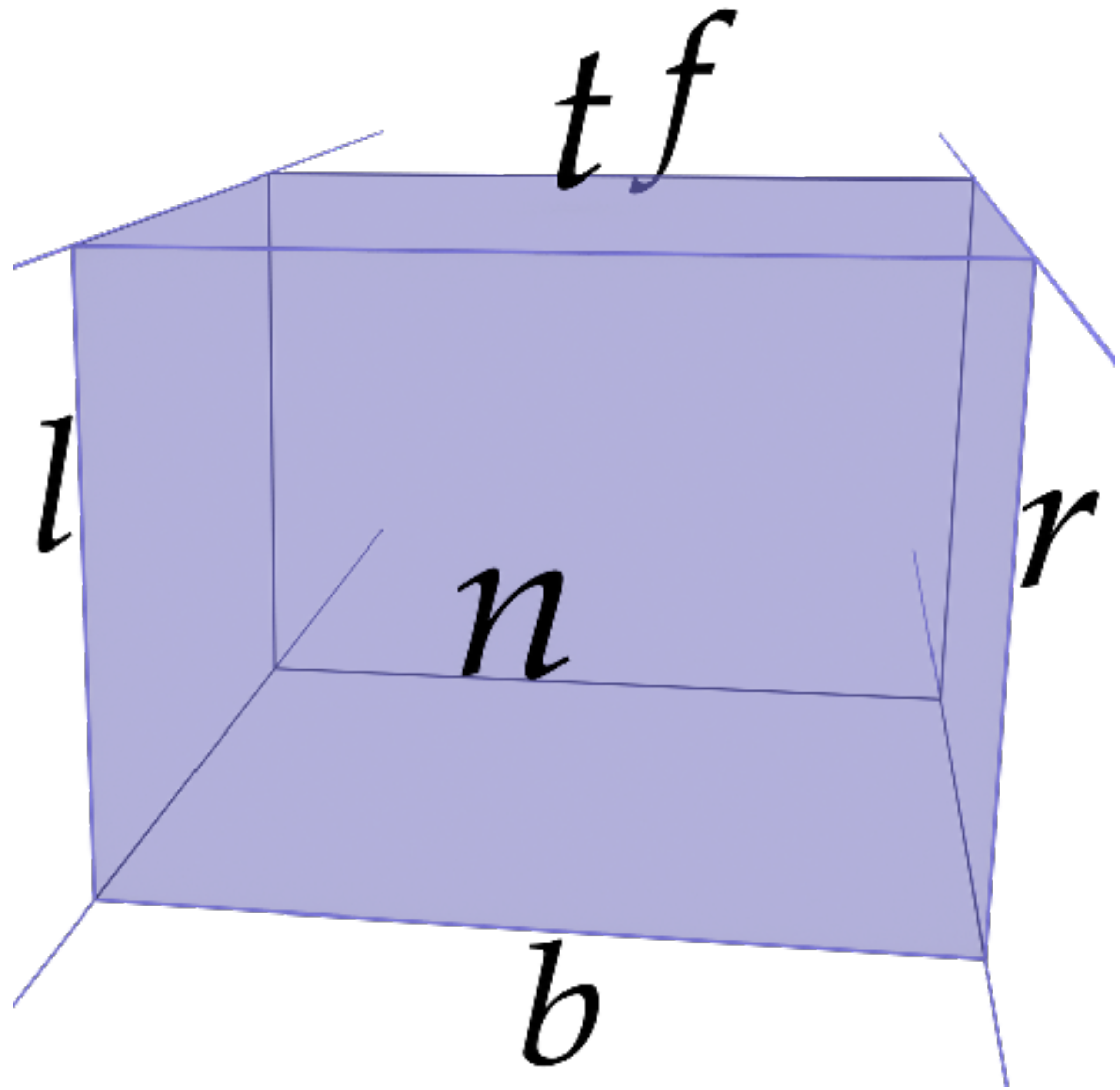


computing intersections: future work from 2018

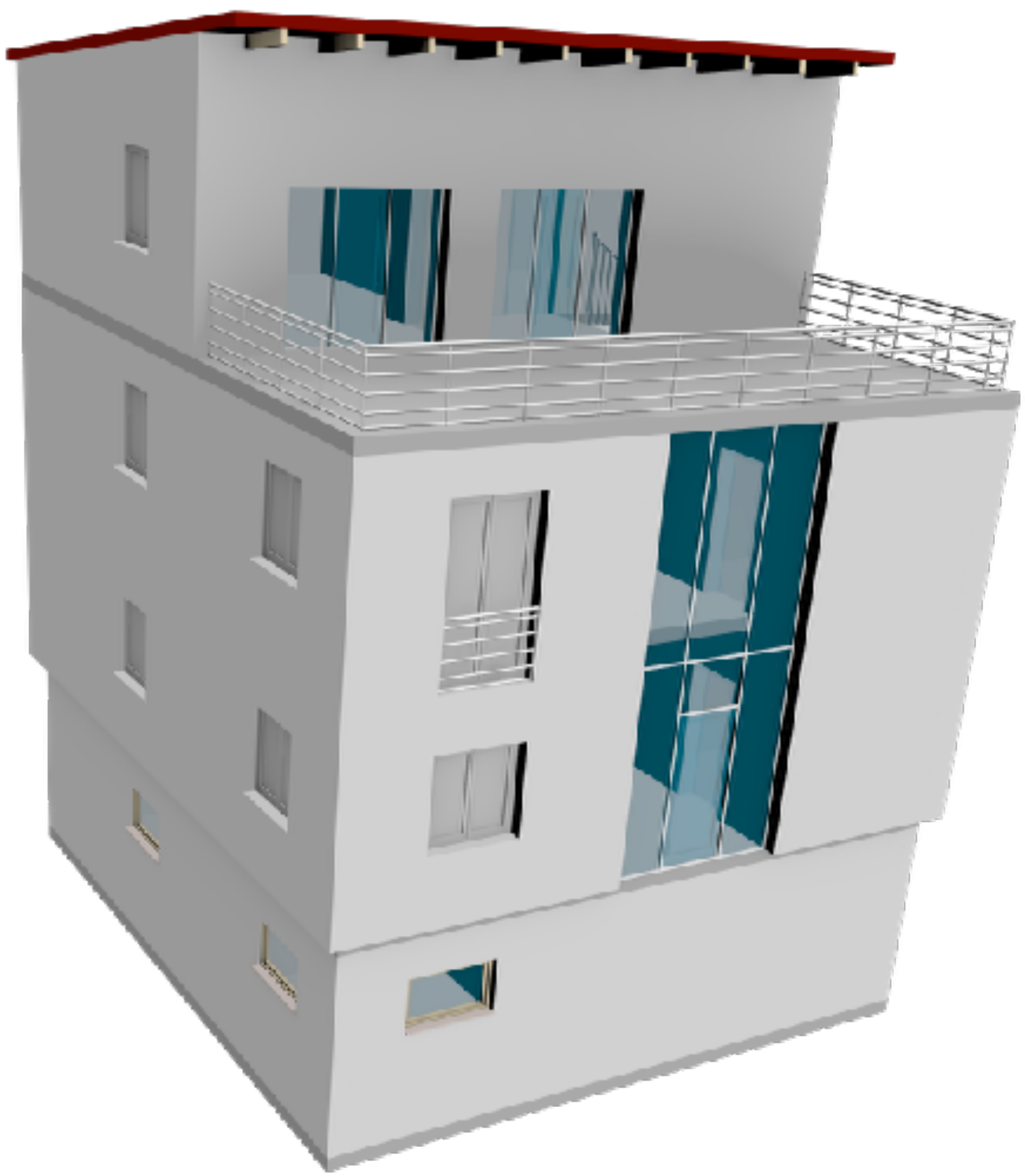
projections to 2D/3D: this presentation

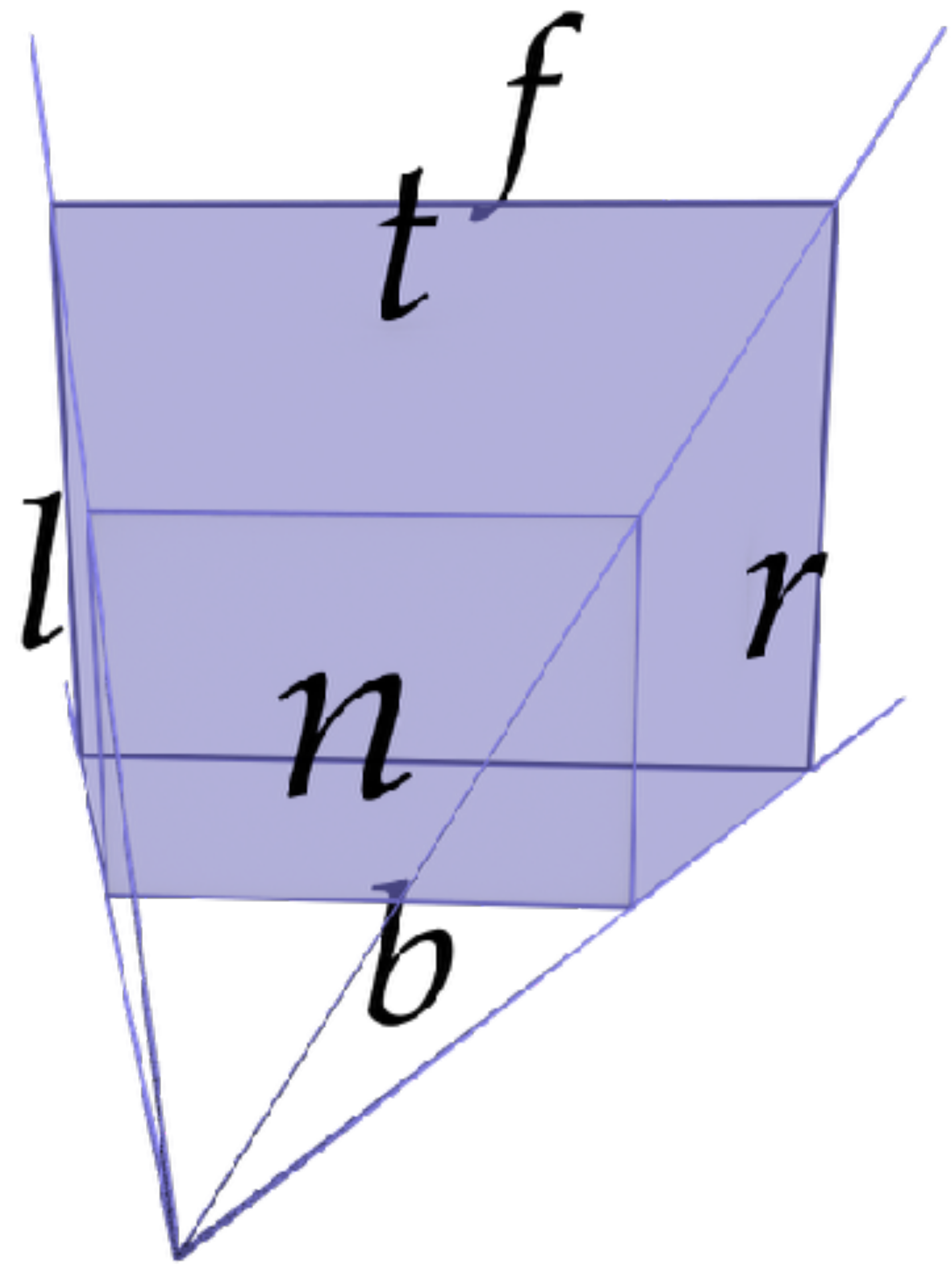
Projections from 3D to 2D: orthographic



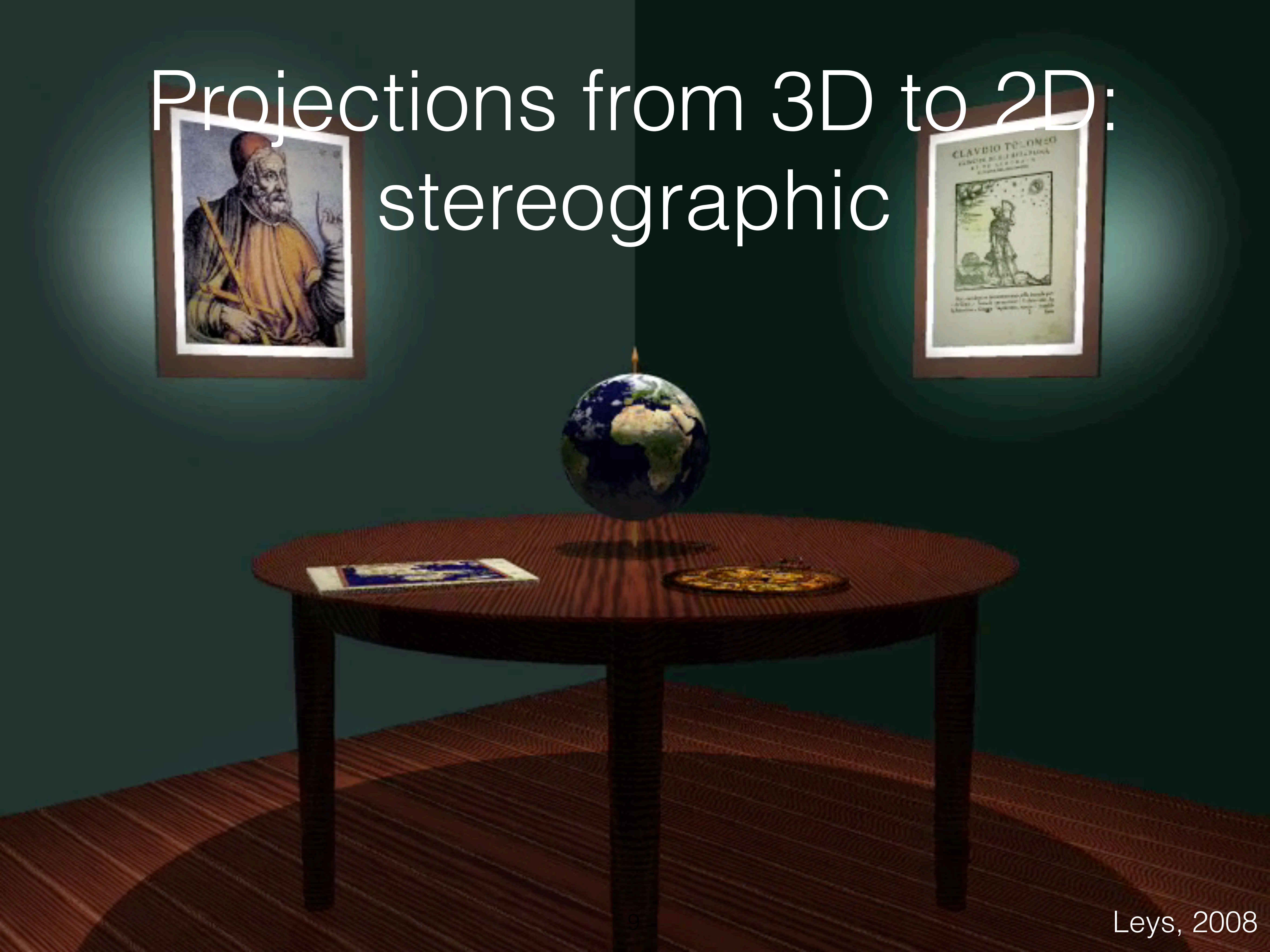


Projections from 3D to 2D: perspective



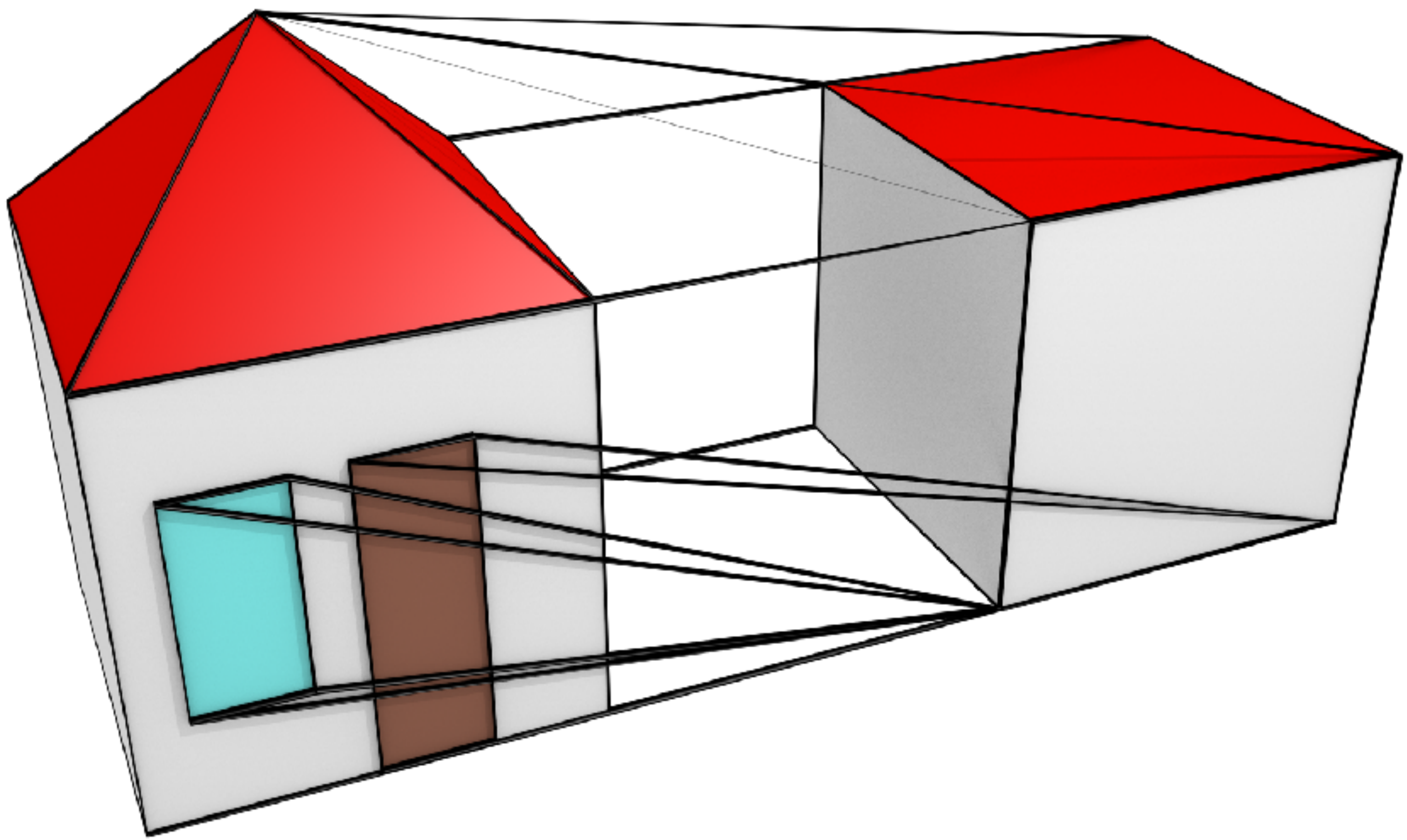


Projections from 3D to 2D: stereographic

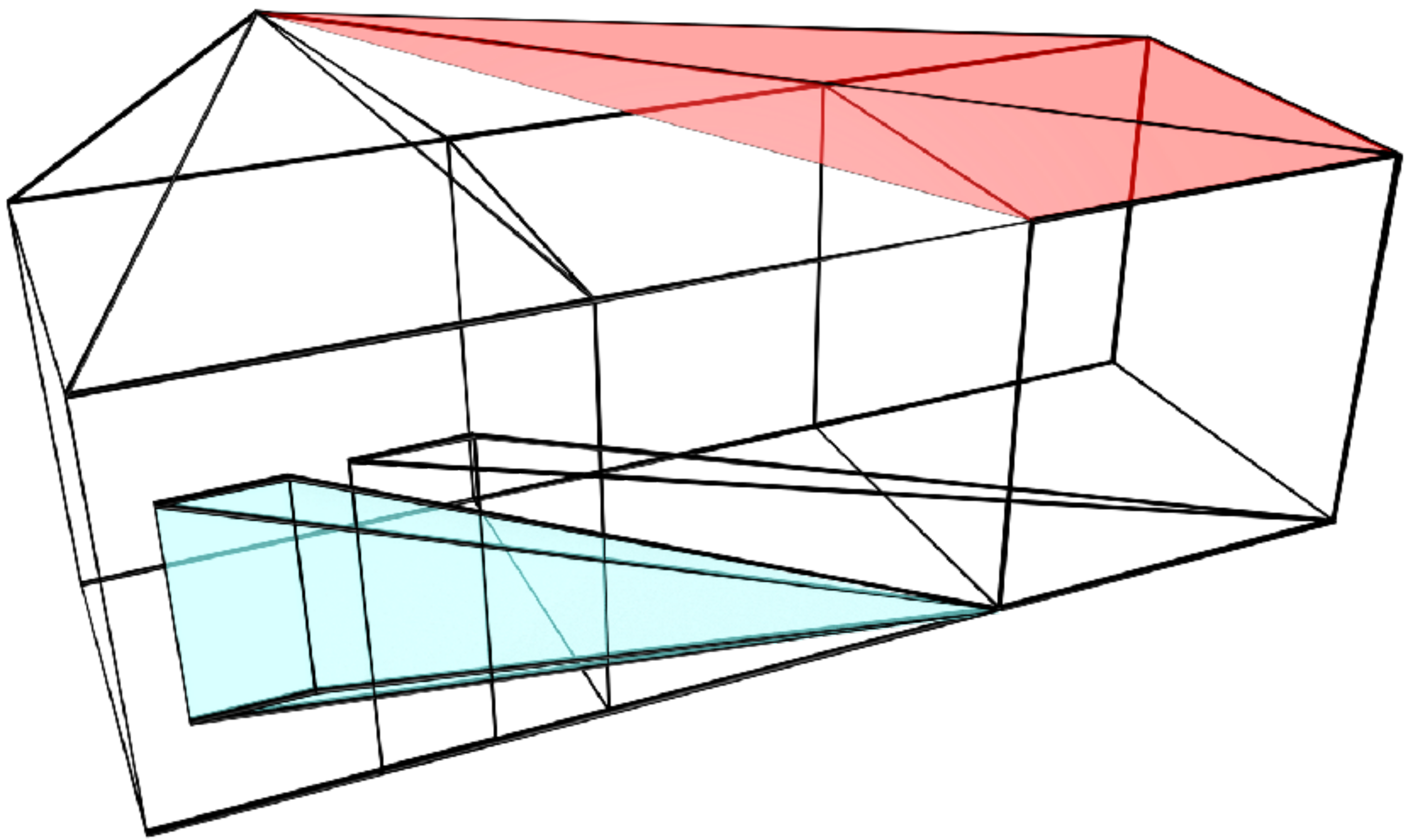


Leys, 2008

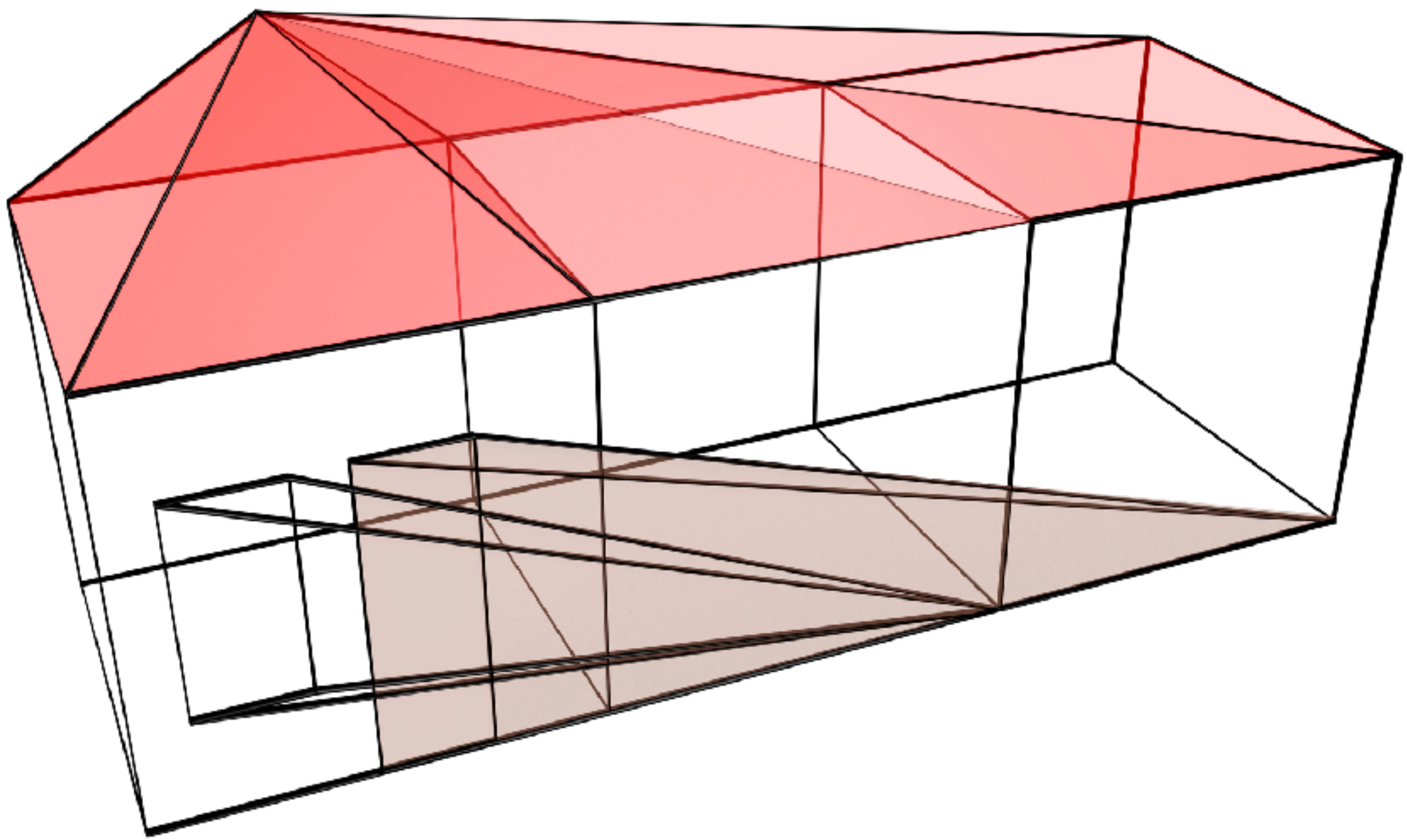
The 4D house



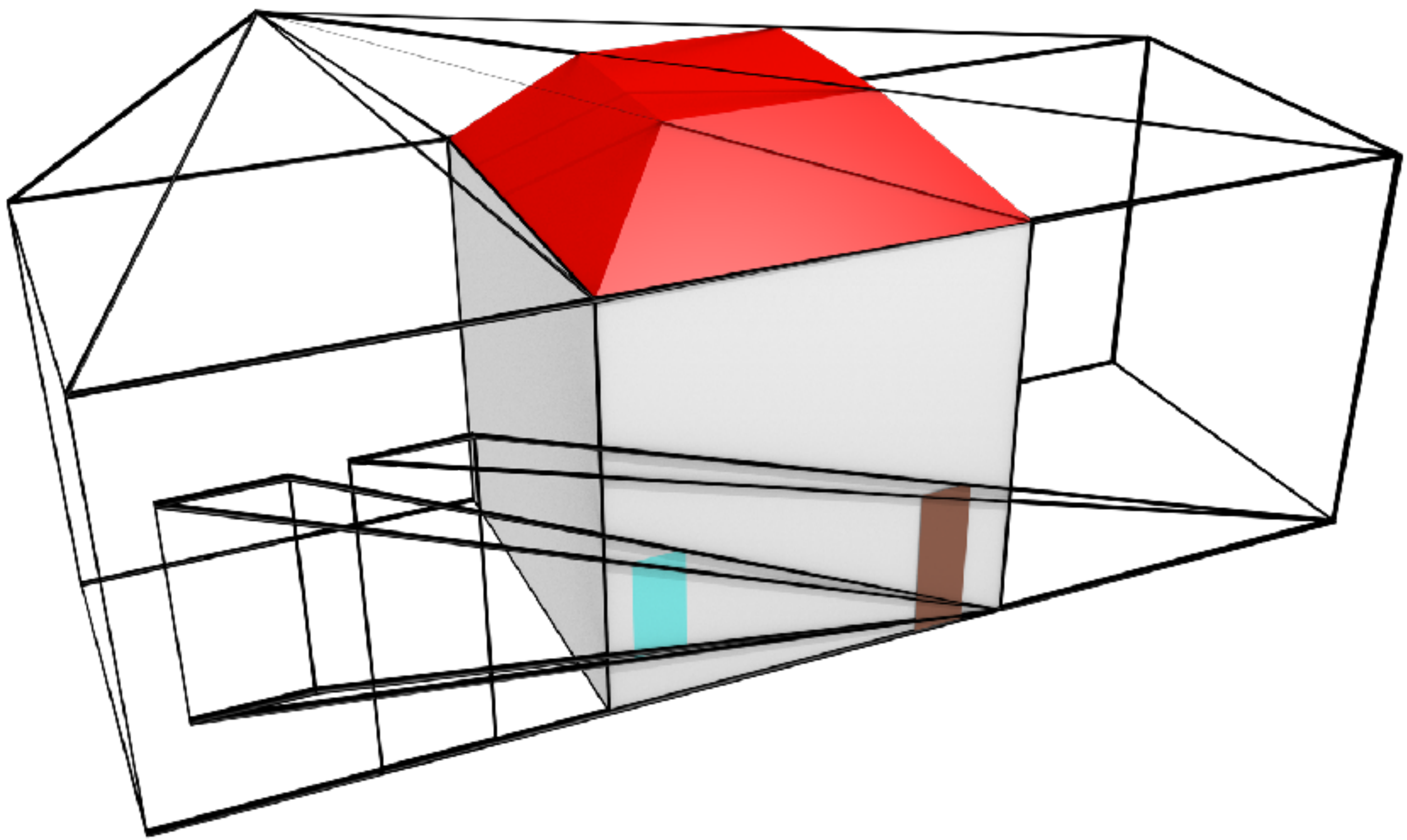
The 4D house



The 4D house



The 4D house



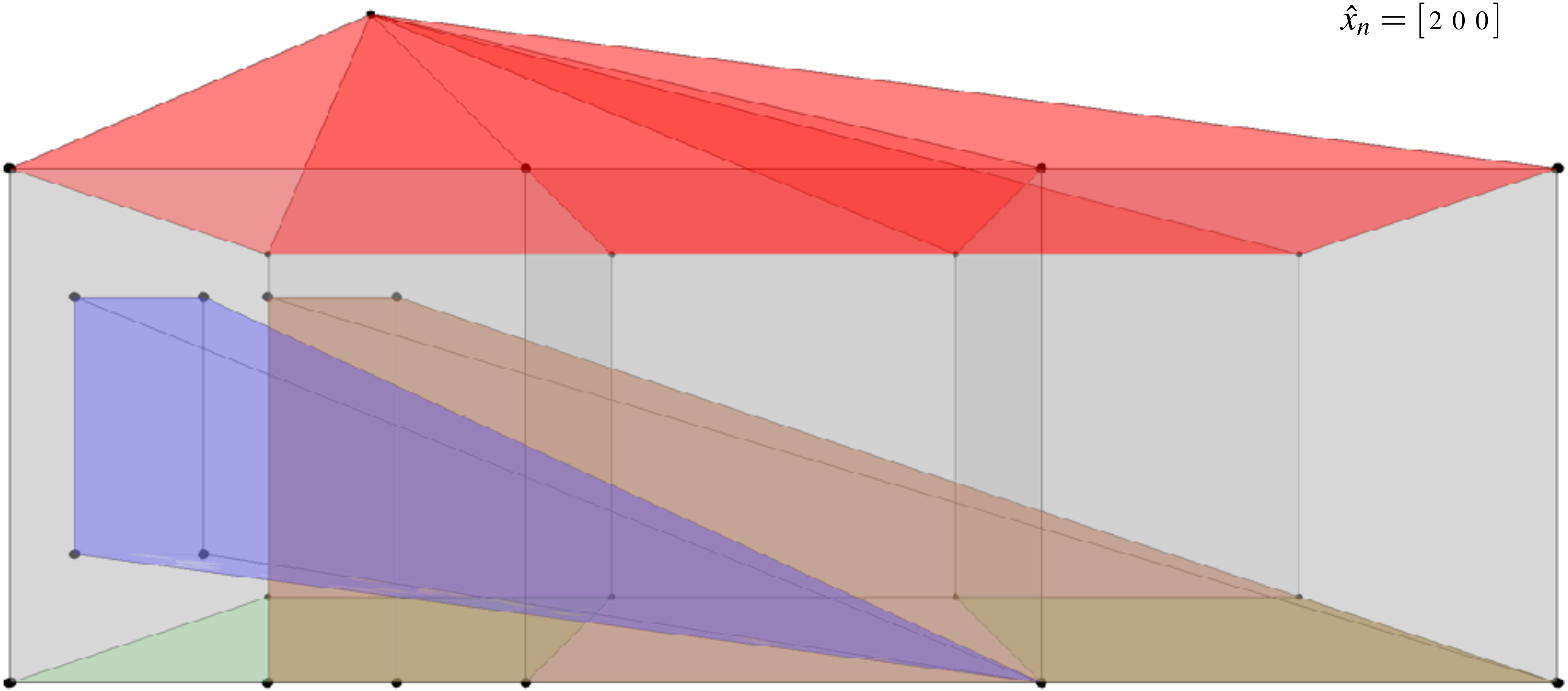
4D to 3D projections

- ‘Long axis’: more formal recreation of typical 4D diagrams
- Orthographic and perspective
- R^4 in/out to S^3, then stereographically to R^3

‘Long axis’ projection

$$P' = P[\, I \; \hat{x}_n \,].$$

$$\hat{x}_n = [\, 2 \;\; 0 \;\; 0 \,]$$



4D projections based on unit vectors

Orthogonal projection

$$\hat{d} = \frac{to - from}{\left\| to - from \right\|}$$

$$\hat{a} = \frac{up \times over \times \hat{d}}{\left\| up \times over \times \hat{d} \right\|}$$

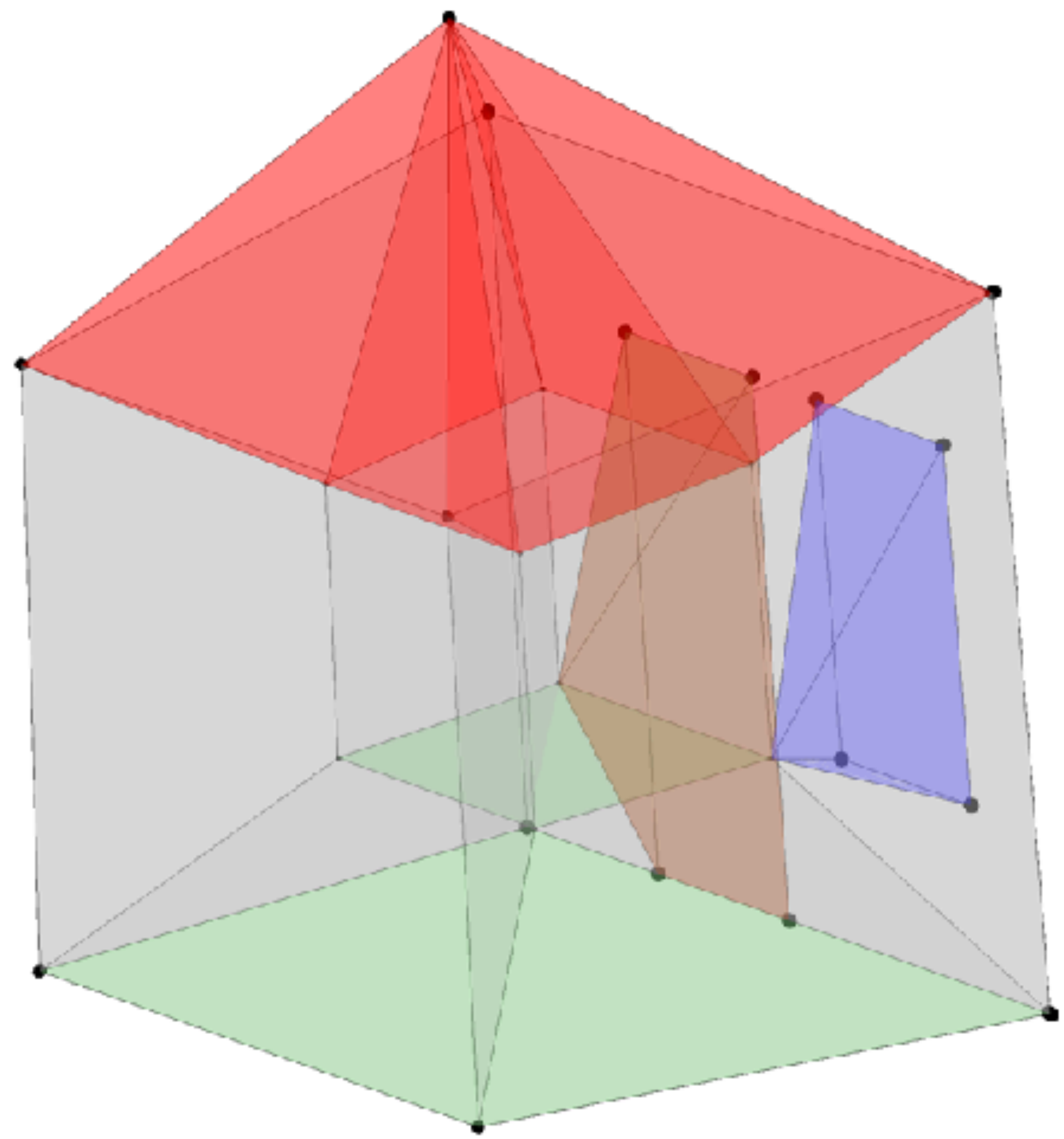
$$\hat{b} = \frac{over \times \hat{d} \times \hat{a}}{\left\| over \times \hat{d} \times \hat{a} \right\|}$$

$$\hat{c} = \hat{d} \times \hat{a} \times \hat{b}$$

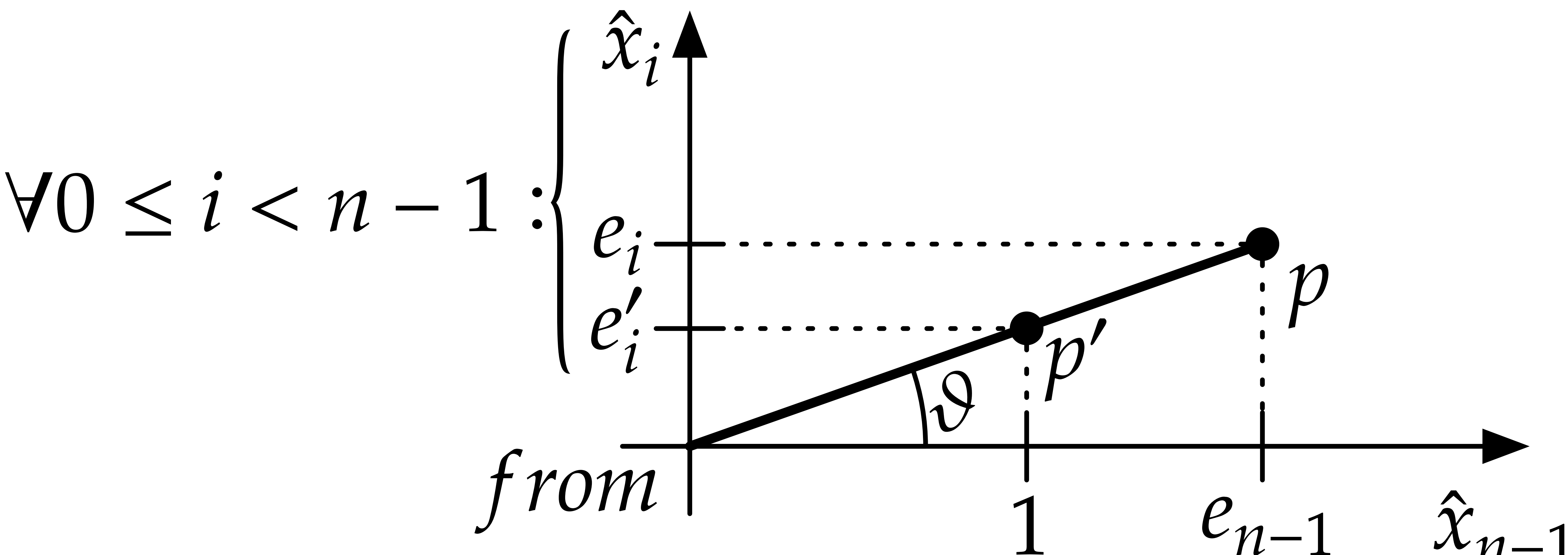
$$E = \left[P - from \right] \left[\hat{a} \quad \hat{b} \quad \hat{c} \quad \hat{d} \right]$$

Based on Hollasch, 1991

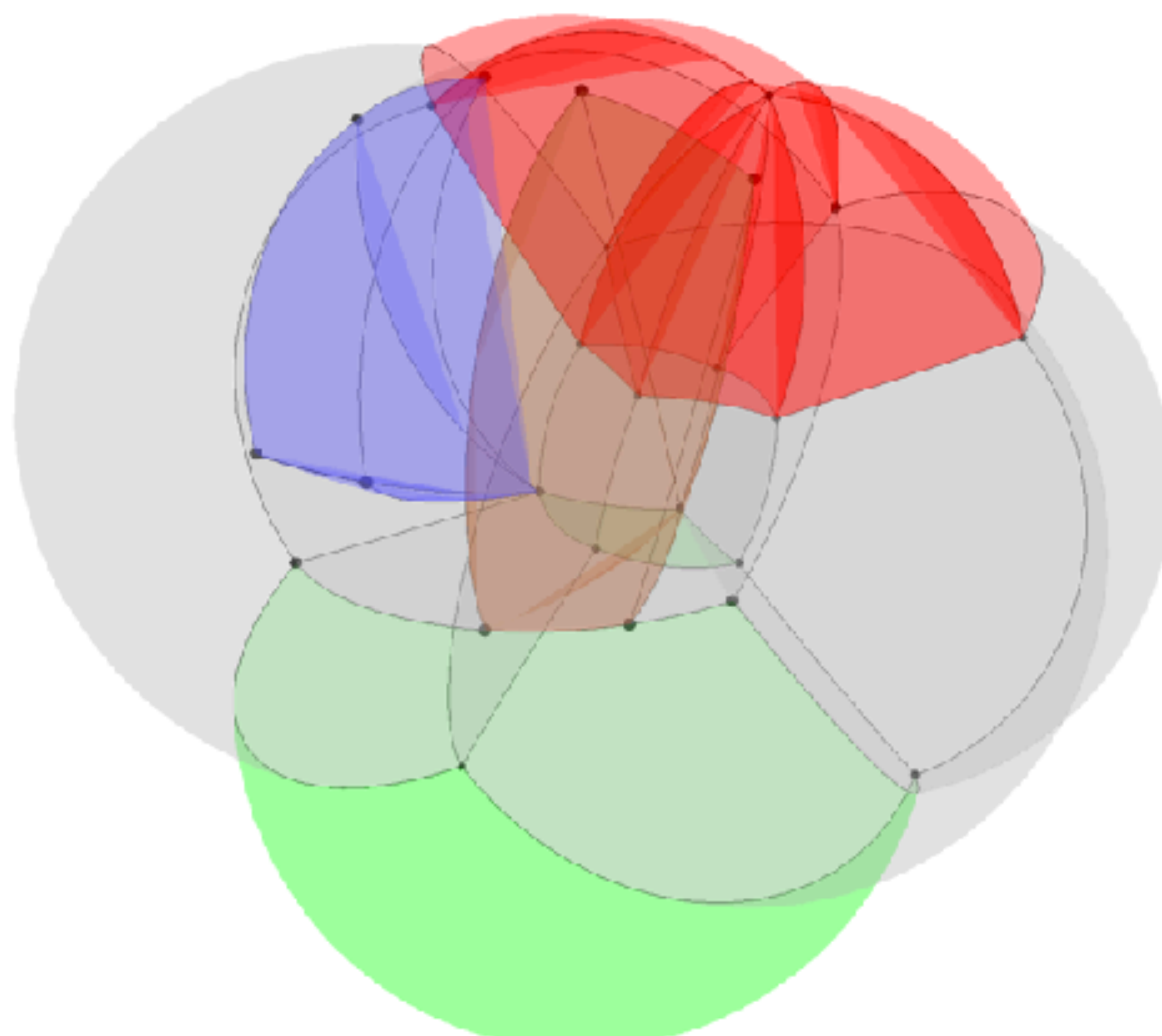
Orthographic and perspective projections



Orthographic and perspective projections



R^4 to S^3 to R^3



To spherical coordinates

$$
\begin{aligned}
r &= \sqrt{x_0^2 + \cdots + x_{n-1}^2} \\
\vartheta_i &= \cos^{-1}\left(\frac{x_i}{\sqrt{r^2 - \sum_{j=0}^{i-1} x_j^2}}\right), \quad \text{for } 0 \leq i < n-2 \\
\vartheta_{n-2} &= \tan^{-1}\left(\frac{x_{n-1}}{x_{n-2}}\right)
\end{aligned}
$$

Back to cartesian with r=1

$$x_i = r \cos \vartheta_i \prod_{j=0}^{i-1} \sin \vartheta_j, \quad \text{for } 0 \leq i < n-2$$

$$x_{n-1} = r \prod_{j=0}^{n-2} \sin \vartheta_j$$

Stereographically to S^3

$$x'_i = \frac{x_i}{x_n - 1}, \quad \text{for } 0 \leq i < n$$

Paper

Submitted to PeerJ Computer Science

minor revision

Visualising higher-dimensional space-time and space-scale objects as projections to $\mathbb{R}^3$

Ken Arroyo Ohori[1], Hugo Ledoux[2], and Jantien Stoter[3]

[1–3]**3D Geoinformation, Delft University of Technology, Delft, Netherlands**

Corresponding author:
First Author[1]

Email address: g.a.k.arroyoohori@tudelft.nl

ABSTRACT

Objects of more than three dimensions can be used to model geographic phenomena that occur in space, time and scale. For instance, a single 4D object can be used to represent the changes in a 3D object's shape across time or all its optimal representations at various levels of detail. In this paper, we look at how such higher-dimensional space-time and space-scale objects can be visualised as projections from $\mathbb{R}^4$ to $\mathbb{R}^3$. We present three projections that we believe are particularly intuitive for this purpose: (i) a simple 'long axis' projection that puts 3D objects side by side; (ii) the well-known orthographic and perspective projections; and (iii) a projection to a 3-sphere (S^3) followed by a stereographic projection to $\mathbb{R}^3$, which results in an inwards-outwards fourth axis. Our focus is in using these projections from $\mathbb{R}^4$ to $\mathbb{R}^3$, but they are formulated from $\mathbb{R}^n$ to $\mathbb{R}^{n-1}$ so as to be easily extensible and to incorporate other non-spatial characteristics. We present a prototype interactive visualiser that applies these projections from 4D to 3D in real-time using the programmable pipeline and compute shaders of the Metal graphics API.

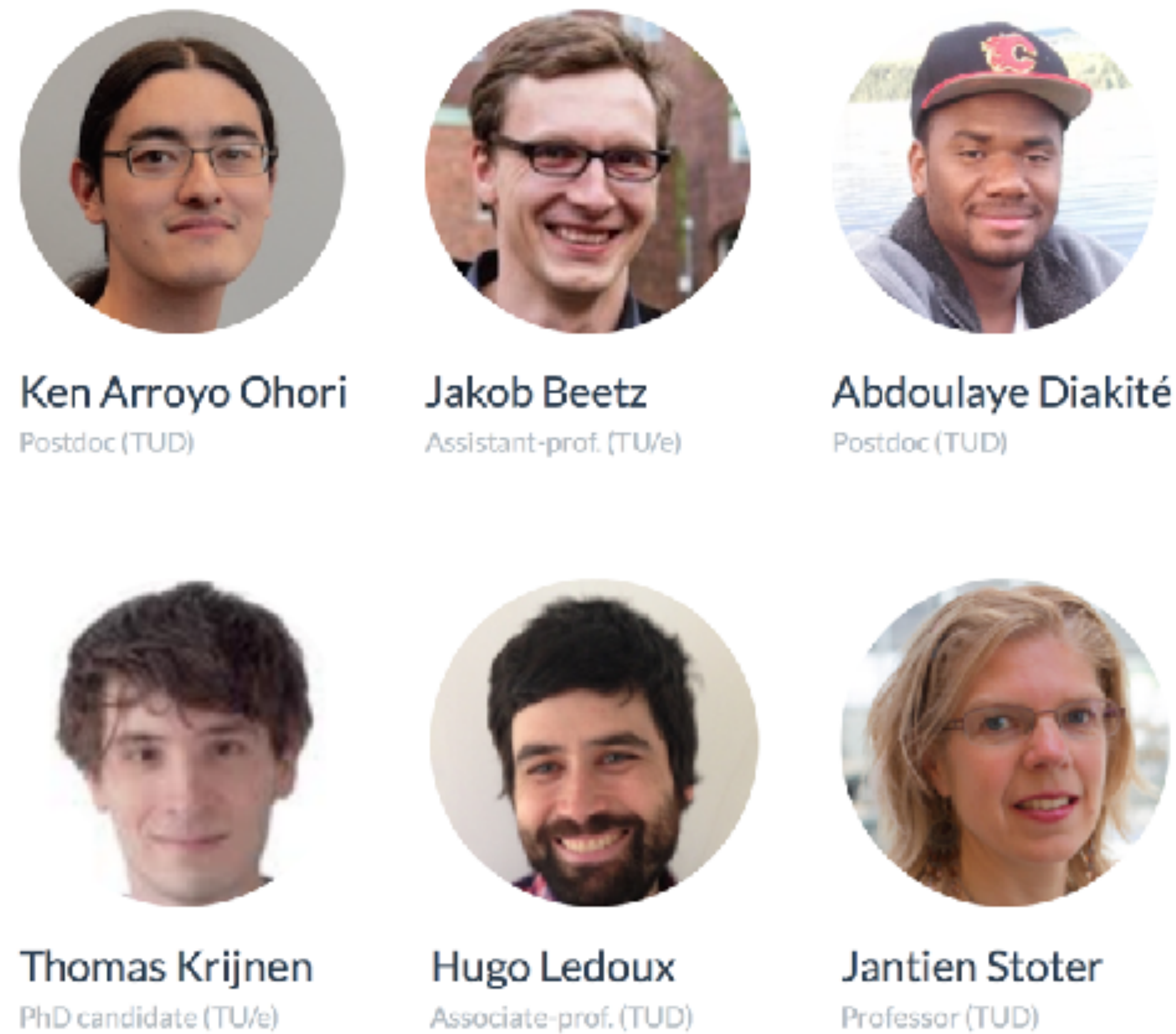
BACKGROUND

Projecting the 3D nature of the world down to two dimensions is one of the most common problems at the juncture of geographic information and computer graphics, whether as the map projections in both paper and digital maps (Snyder, 1987; Grafarend and You, 2014) or as part of an interactive visualisation of a 3D city model on a computer screen (Foley and Nielson, 1992; Shreiner et al., 2013). However, geographic information is not inherently limited to objects of three dimensions. Non-spatial characteristics such as time (Hägerstrand, 1970; Güting et al., 2000; Hornsby and Egenhofer, 2002; Kraak, 2003) and scale (Meijers, 2011a) are often conceived and modelled as additional dimensions, and objects of three or more dimensions can be used to model objects in 2D or 3D space that also have changing geometries along these non-spatial characteristics (van Oosterom and Stoter, 2010; Arroyo Ohori, 2016). For example, a single 4D object can be used to represent the changes in a 3D object's shape across time or all the best representations of a 3D object at various levels of detail (van Oosterom and Meijers, 2014; Arroyo Ohori et al., 2015a,c).

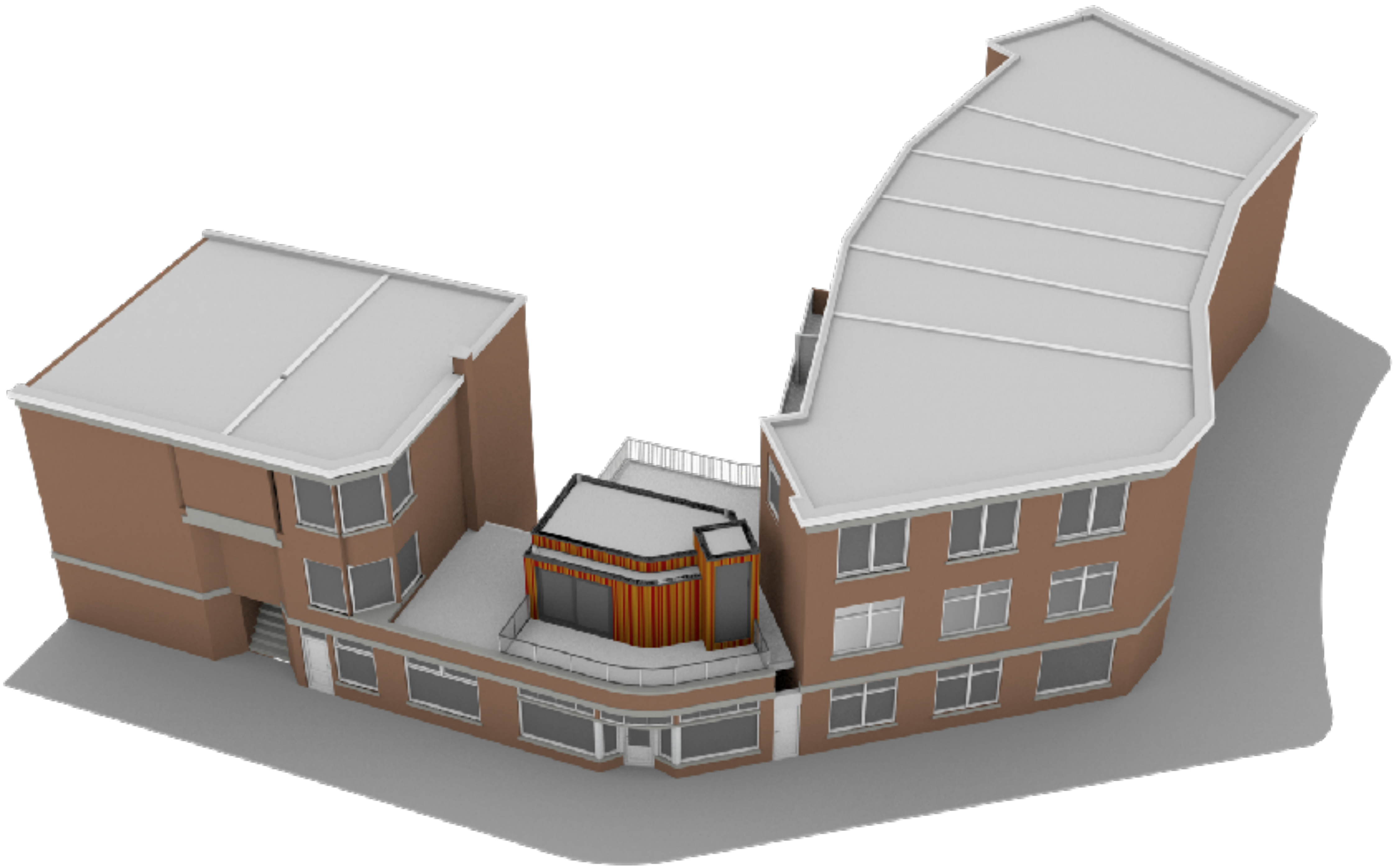
Objects of more than three dimensions can be however unintuitive (Noll, 1967; Frank, 2014), and visualising them is a challenge. While some operations on a higher-dimensional object can be achieved by running automated methods (e.g. certain validation tests or area/volume computations) or by visualising only a chosen 2D or 3D subset (e.g. some of its bounding faces or a cross-section), sometimes there is no substitute for being able to view a complete *n*D object—much like viewing floor or façade plans is often no substitute for interactively viewing the complete 3D model of a building. By viewing a complete model, one can see at once the 3D objects embedded in the model at every point in time or scale as well as the equivalences and topological relationships between their constituting elements. More directly, it also makes it possible to get an intuitive understanding of the complexity of a given 4D model.

For instance, in Fig. 1 we show an example of a 4D model representing a house at two different levels of detail and all the equivalences its composing elements. It forms a valid manifold 4-cell (Arroyo Ohori demo

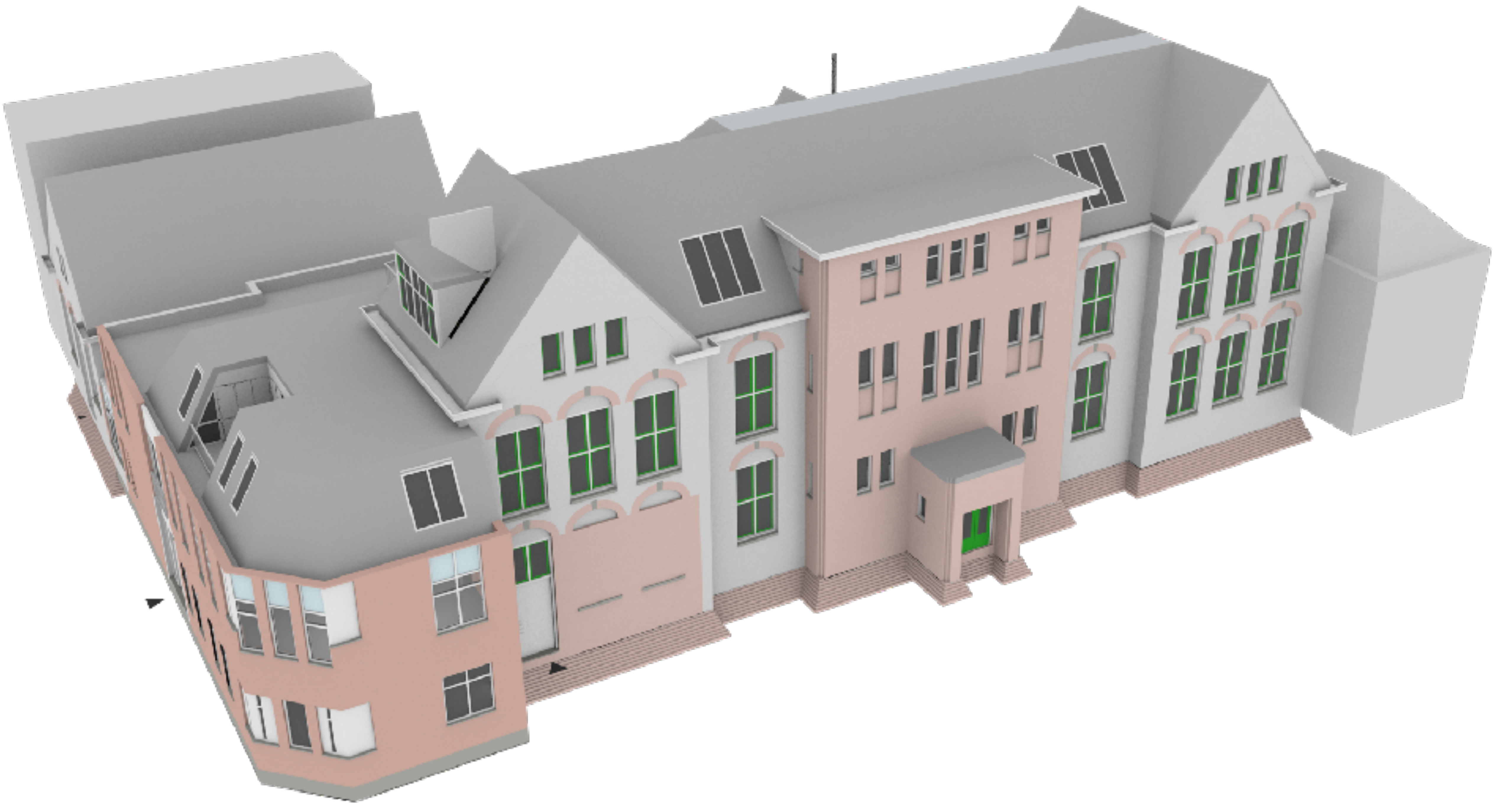
BIM-GIS integration: GeoBIM project



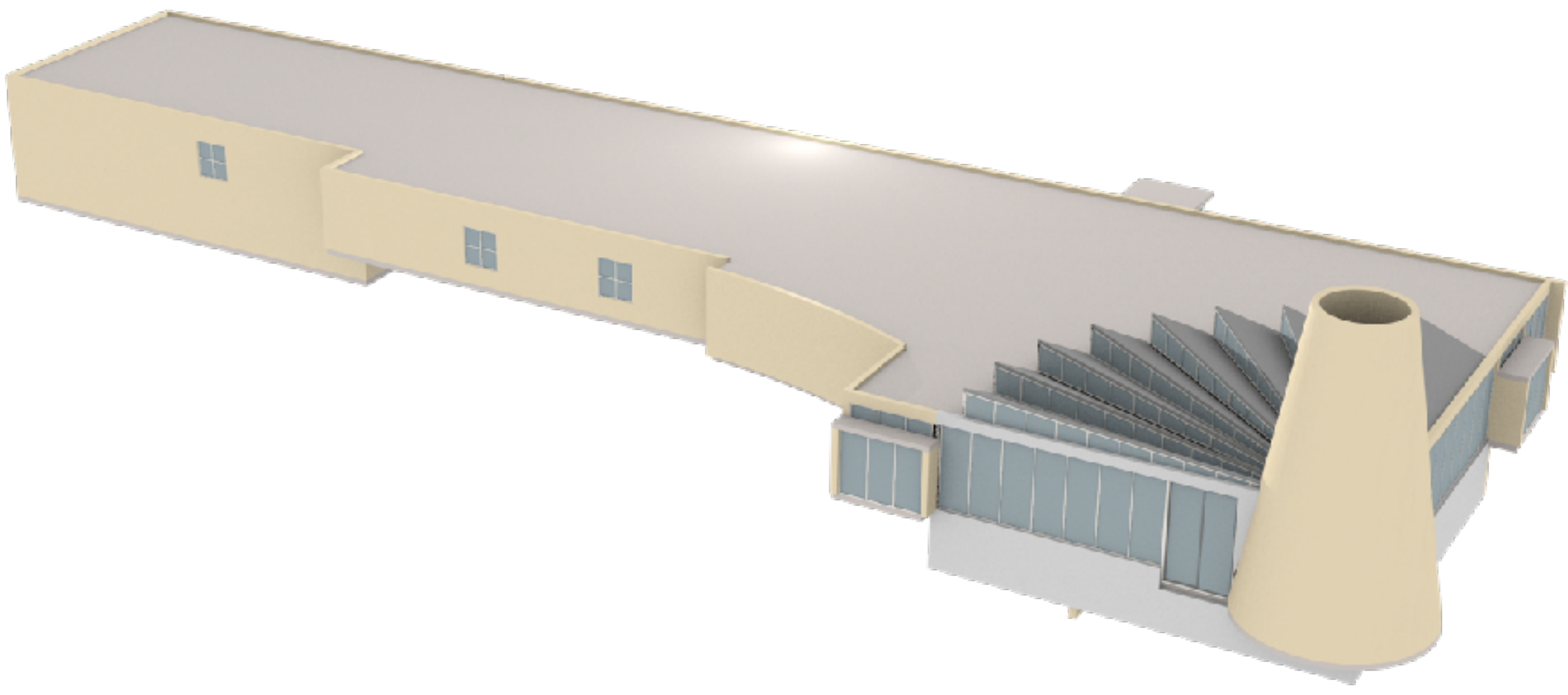
Beautiful IFC models



Beautiful IFC models



Beautiful IFC models



Our goal

- Develop an **interface between CityGML and IFC** to prepare for a fundamental solution to bridge the gap between Geo and BIM.
- open-source API to represent IFC + CityGML with the same data structure
- recommendations for future integration

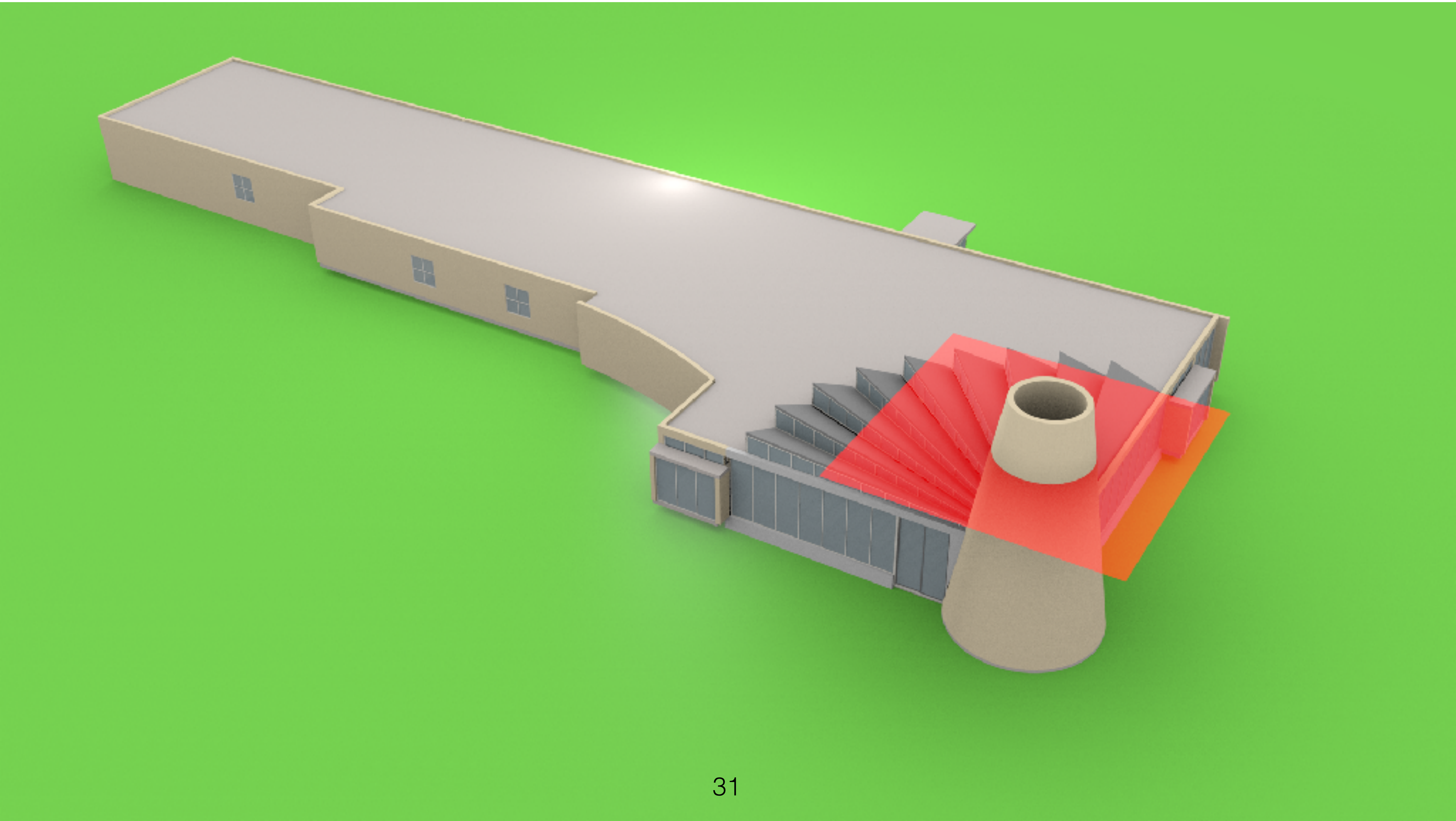
GeoBIM project use cases

- **Use case 1**: The process of submitting an IFC model to a building permit-application portal by citizens and companies; checking the IFC design against the existing physical world (represented in a 3D city model) and against a 3D zoning plan; and finally updating the 3D city model by integrating the 3D building model.

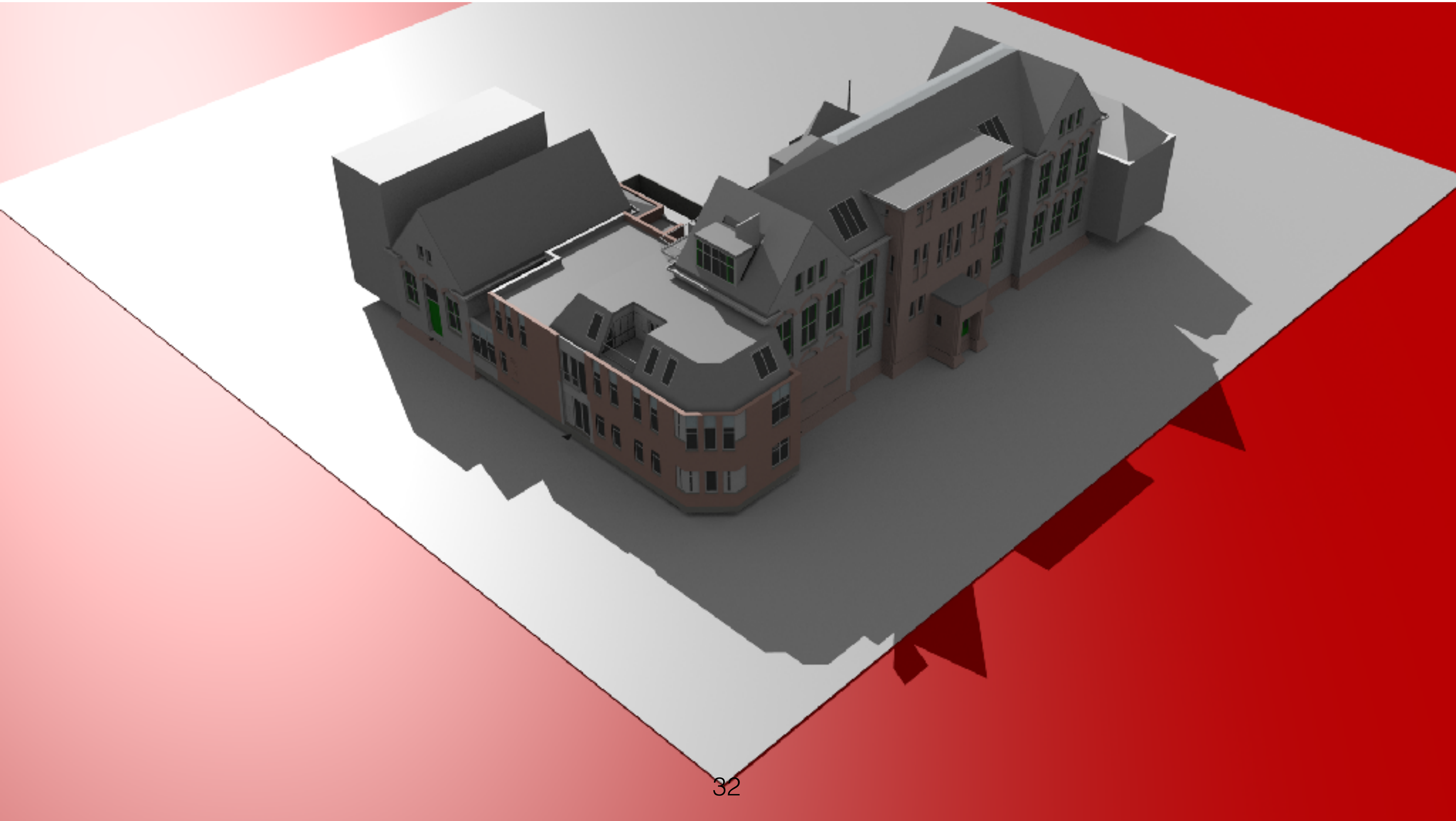
- **Use case 2**: The process of supporting the life-cycle of objects with a continuous information chain: using information about complete urban areas in the design process (i.e. using geo-information in a BIM application) and, at a later stage, converting plan, design and construction data to maintenance data. The focus of this second use case is on large infrastructure projects.

- **Use case 3**: Integration of sub-soil information in the BIM design process (Abdou).

Zoning plans (height)



Shadow tests



Methodology

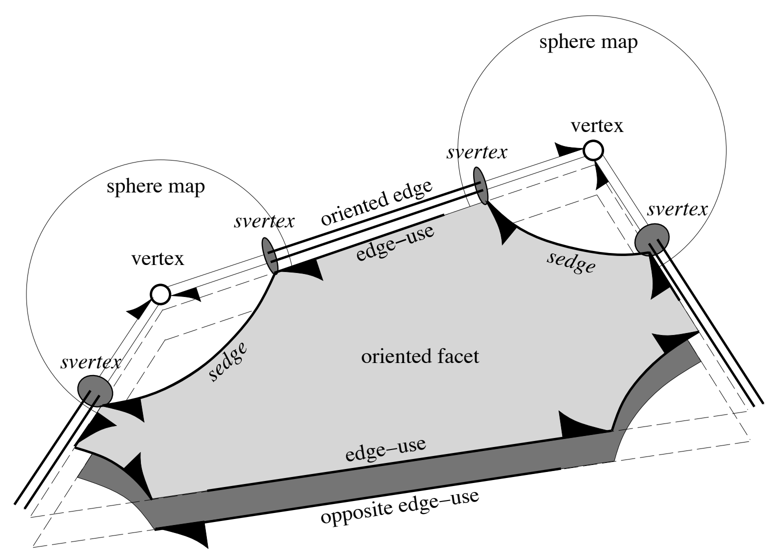
IFC

IfcOpenShell
the open source Ifc toolkit and geometry engine

CGAL

.obj

CityGML



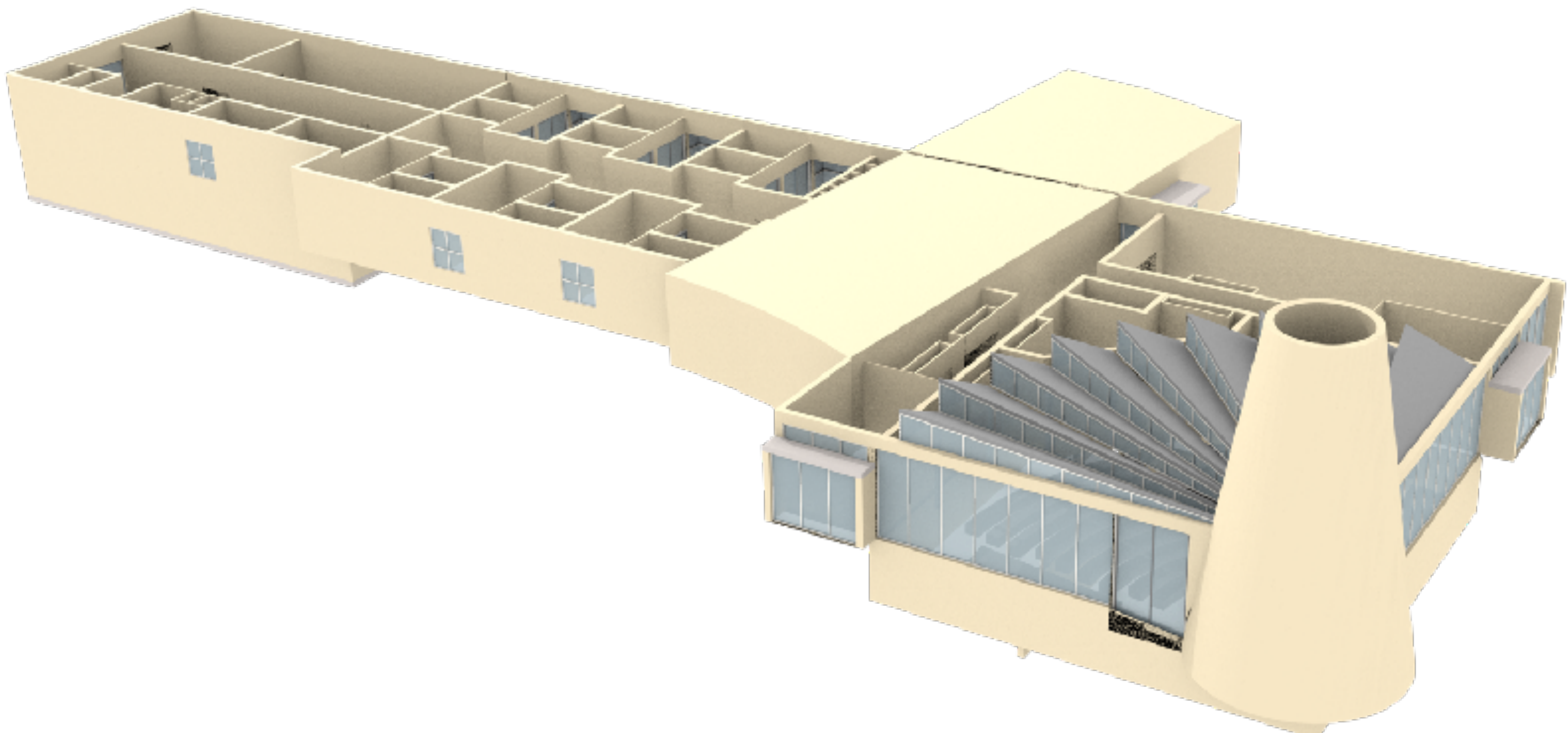
Hachenberger, 2006

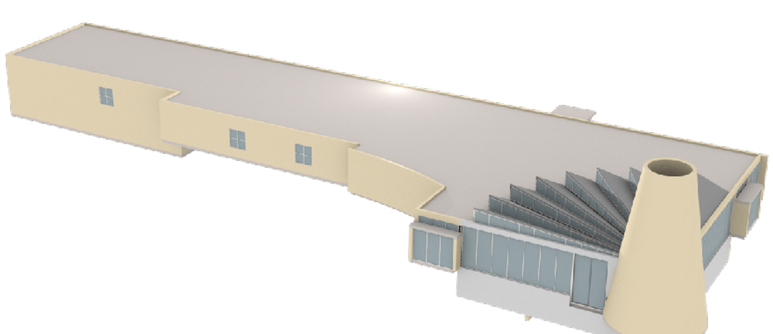
Nef polyhedron per volumetric object
Boolean set operations to solve use cases

Prototype status

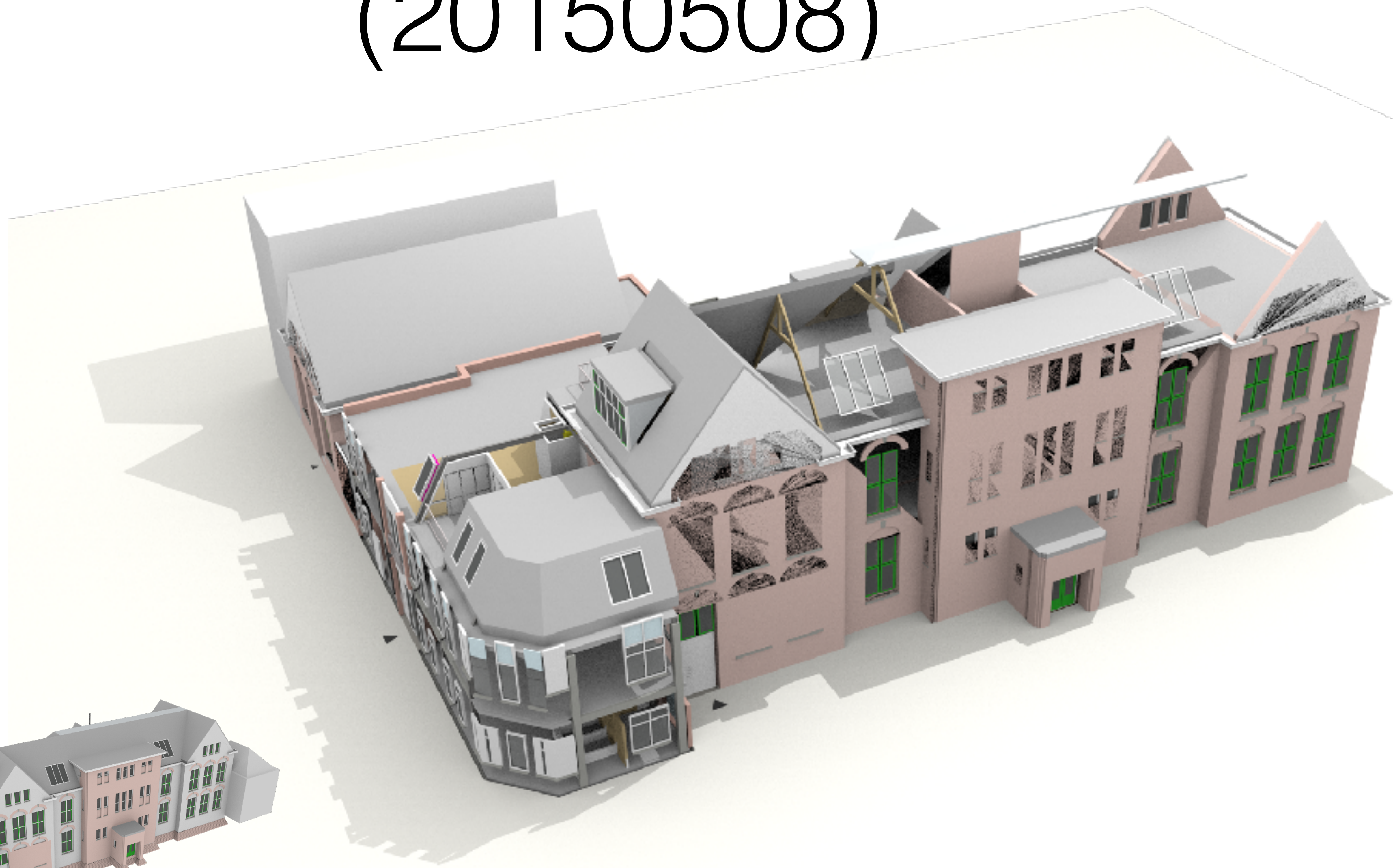
- Parsing ~60 geometric and all (~700) mostly semantic classes in Ifc2x3 and Ifc4
- Generating CGAL Polyhedra_3 and Nef_polyhedra_3 from every object
- Output as .obj with correct materials
- Output separately objects that fail necessary validation tests
- Catch errors caused by bad data

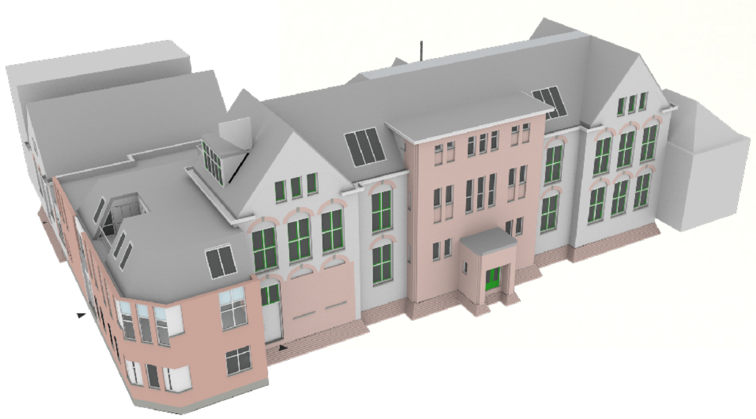
CUVO Ockenburghstraat KOW



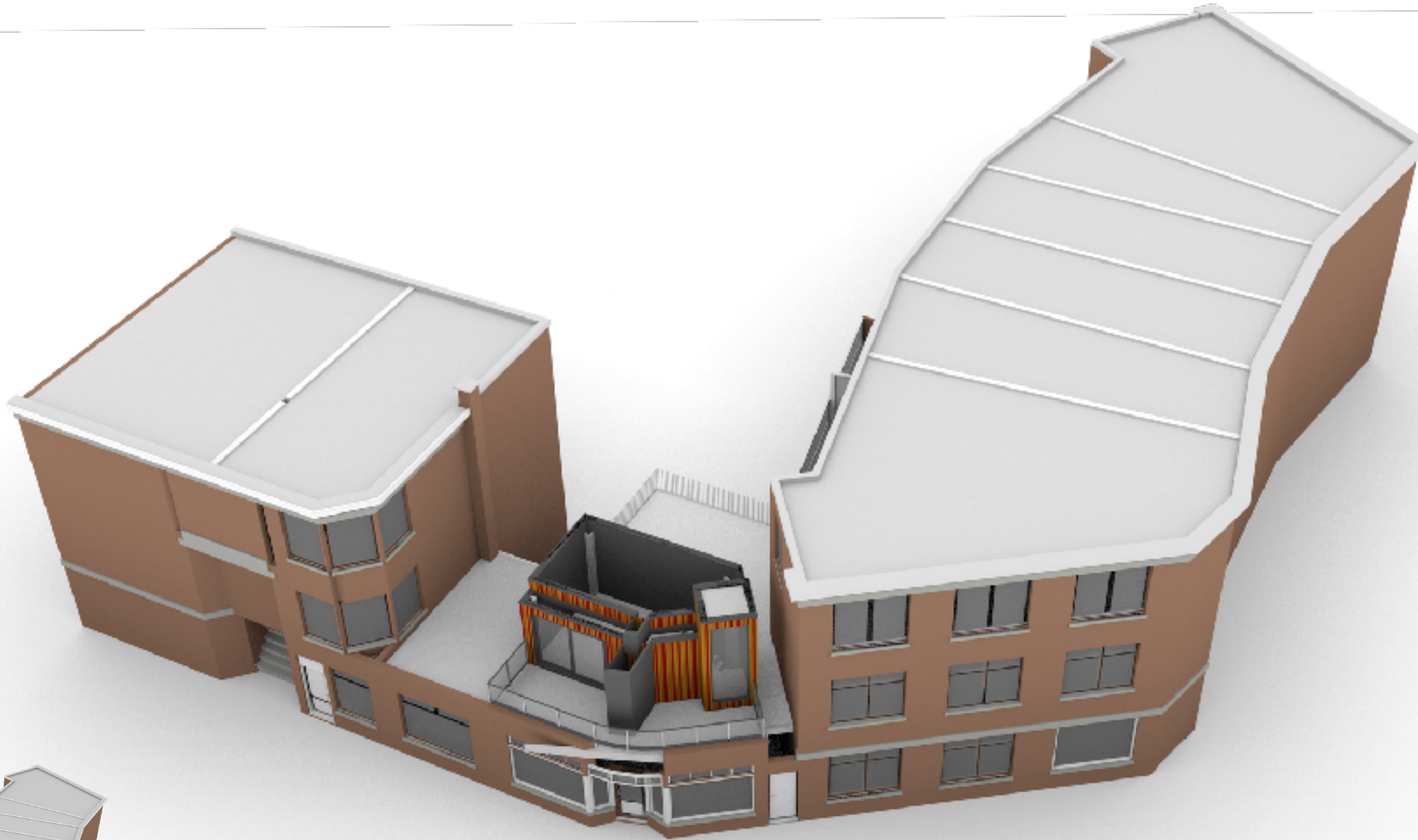


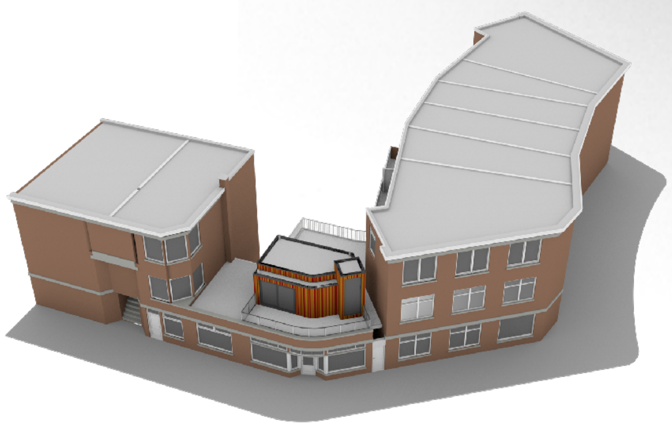
Witte_de_Withstraat (20150508)





Rabarberstraat144



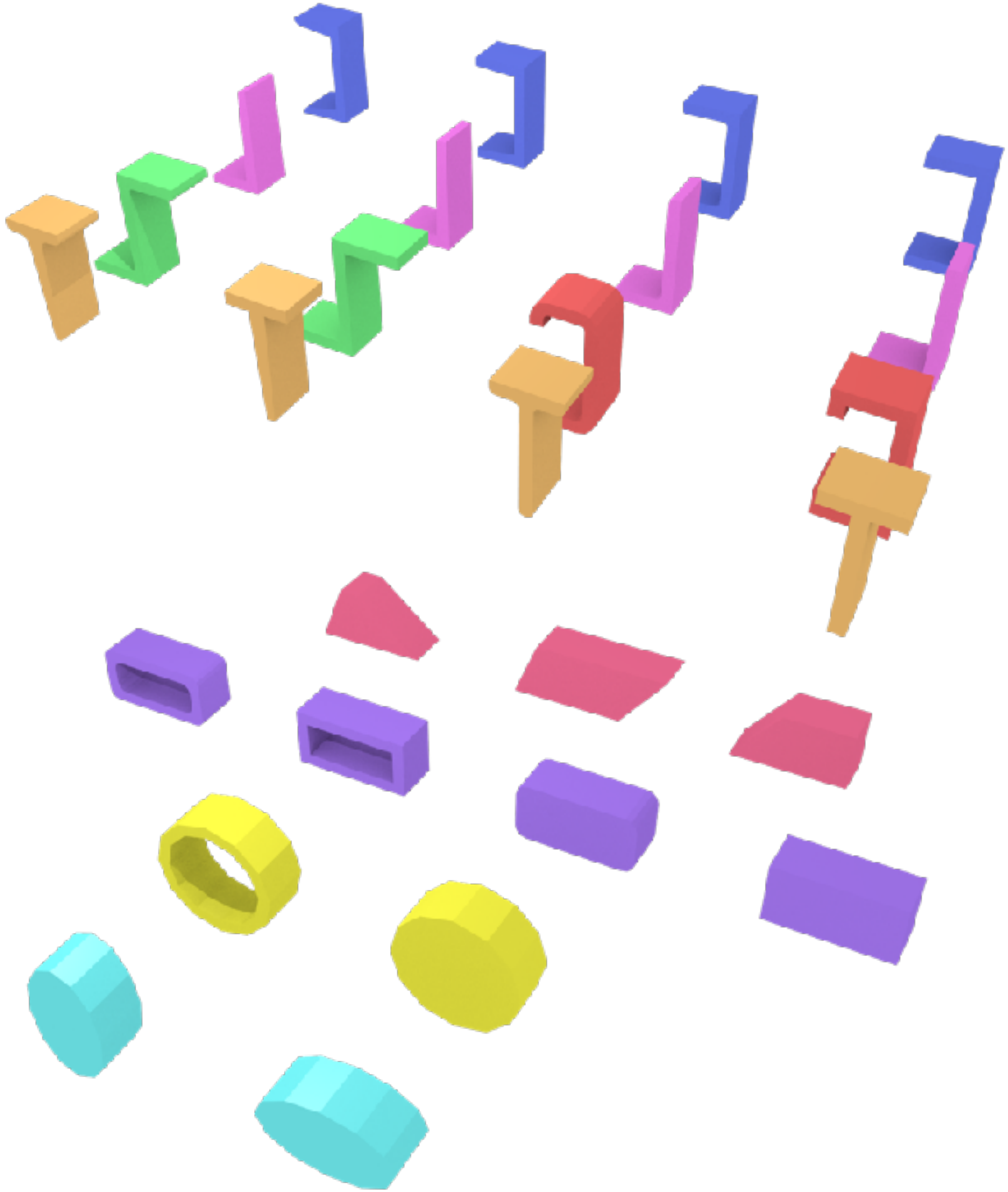


To do

- Fix more geometric issues
- Add more validation tests
- CityGML output
- Work on use cases using: zoning plans, shadows, noise and/or solar potential
- Use CityGML in design software?
- Test simplebim CityGML add-on?

(Unexpected) problems

- Many parametric shapes
- Many types of transformations, some bugs remain
- **Very** bad data
- Need to discretise smooth curves and surfaces
- Ifc schema pieces that are hard to implement in CGAL
- Minimalist approach in GIS vs. maximalist approach in BIM (12 modules in CityGML 2 vs. 768 entities in Ifc4Add1). Not counting selects, functions, etc.
- BIM concepts that differ or don't exist in GIS software (topological faces, wires)
- Georeferencing issues



Transformations / placements

- IfcAxis1Placement
- IfcAxis2Placement2D
- IfcAxis2Placement3D
- IfcCartesianTransformationOperator2D
- IfcCartesianTransformationOperator2DnonUniform
- IfcCartesianTransformationOperator3D
- IfcCartesianTransformationOperator3DnonUniform
- IfcLocalPlacement

some errors

thanks!

References

- **Martijn Meijers**. Variable-scale Geo-information. *PhD thesis, Delft University of Technology*, December 2011.
- **Jos Leys and Étienne Ghys and Aurelién Alvarez**. Dimensions: une promenade mathématique. 2008. *Available at http://www.dimensions-math.org/Dim_E.htm*.
- **Steven Richard Hollasch**. Four-space visualization of 4D objects. *Master's thesis, Arizona State University*, 1991.
- **Peter Hachenberger**. Boolean Operations on 3D Selective Nef Complexes Data Structure, Algorithms, Optimized Implementation, Experiments and Applications. *PhD thesis, Saarland University*, 2006.