3D models and reference spaces

3D FOR CULTURAL HERITAGE

MARCO CALLIERI

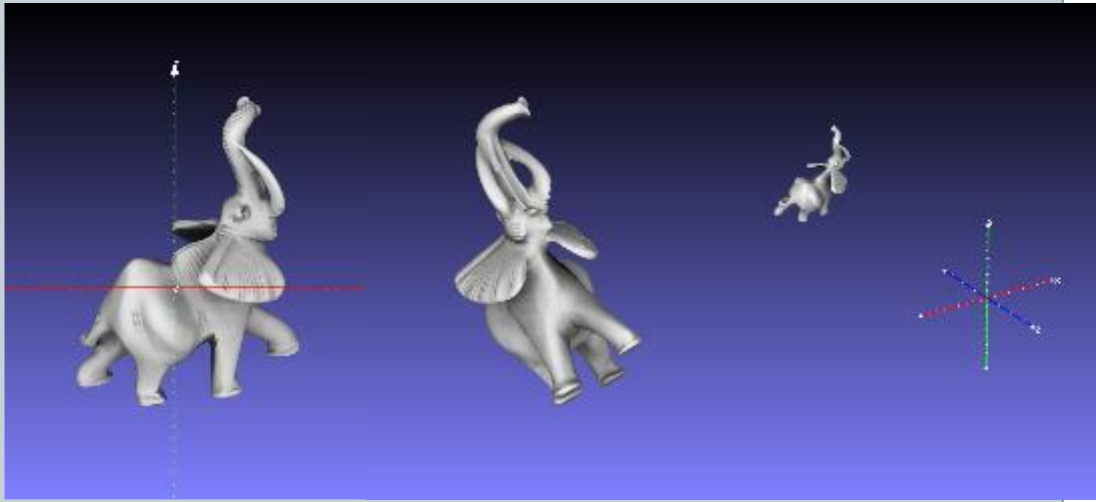
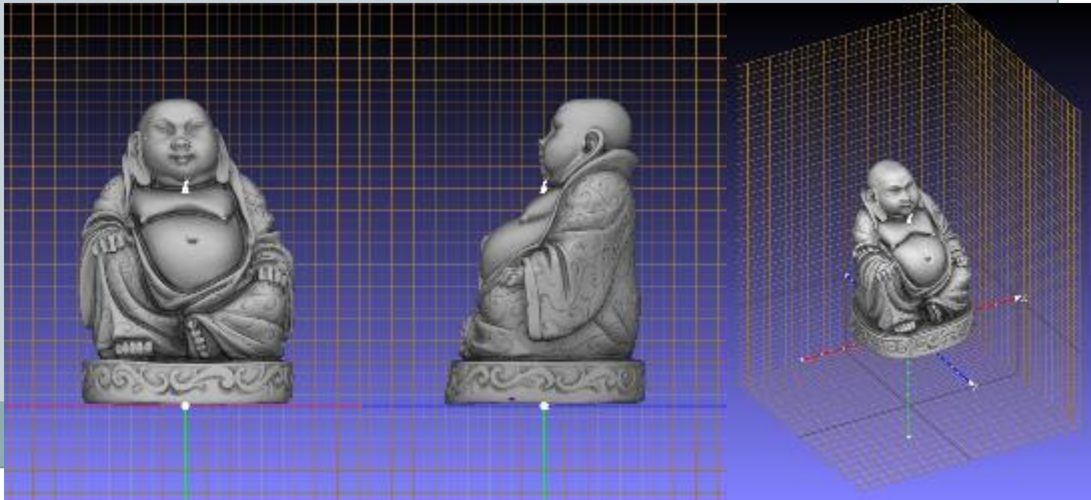
VISUAL COMPUTING LAB

ISTI-CNR PISA, ITALY

Models in space

We have opened 3D models that appear «well» placed and oriented, while other seems a bit off.

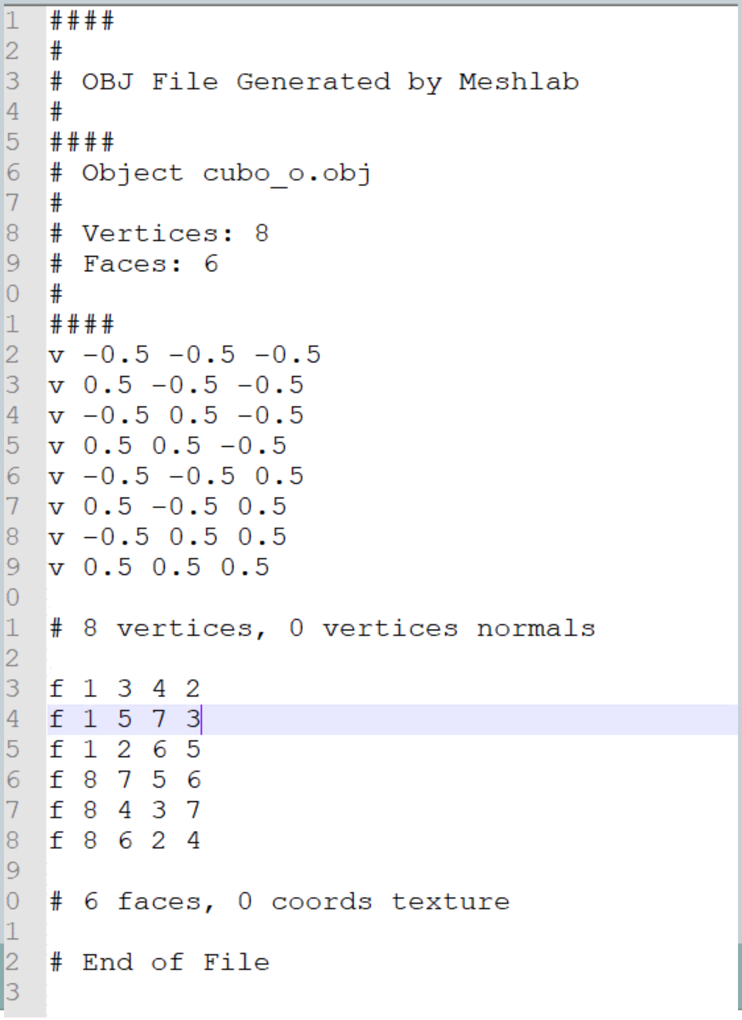
This is due to the data inside the file…

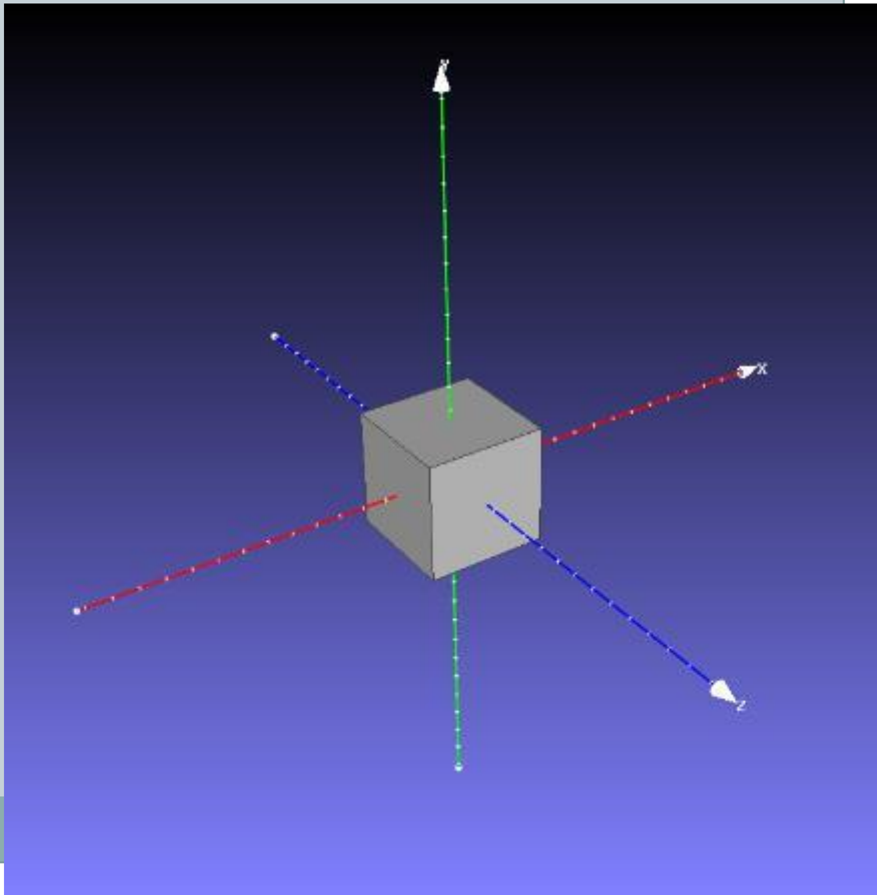
Inside a 3D model

A simple cube

Coordinates & rendering

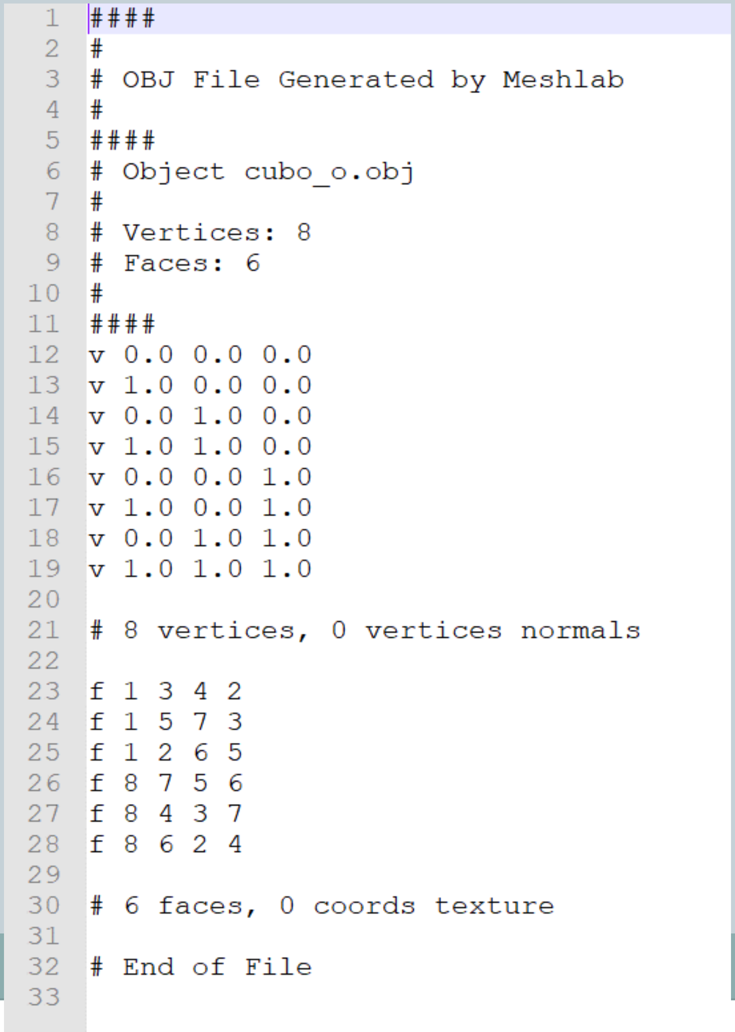


```
####
#
# OBJ File Generated by Meshlab
#
####
# Object cubo_o.obj
#
# Vertices: 8
# Faces: 6
#
####
v -0.5 -0.5 -0.5
v 0.5 -0.5 -0.5
v -0.5 0.5 -0.5
v 0.5 0.5 -0.5
v -0.5 -0.5 0.5
v 0.5 -0.5 0.5
v -0.5 0.5 0.5
v 0.5 0.5 0.5

# 8 vertices, 0 vertices normals

f 1 3 4 2
f 1 5 7 3
f 1 2 6 5
f 8 7 5 6
f 8 4 3 7
f 8 6 2 4

# 6 faces, 0 coords texture

# End of File
```

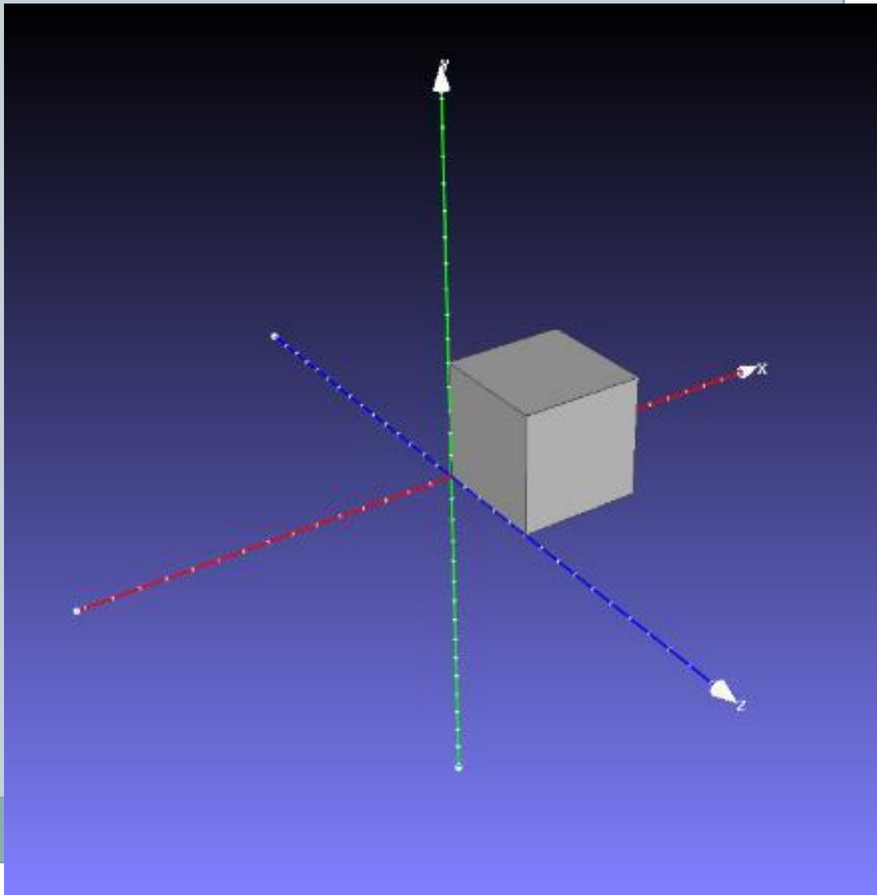


A simple cube

Coordinates & rendering

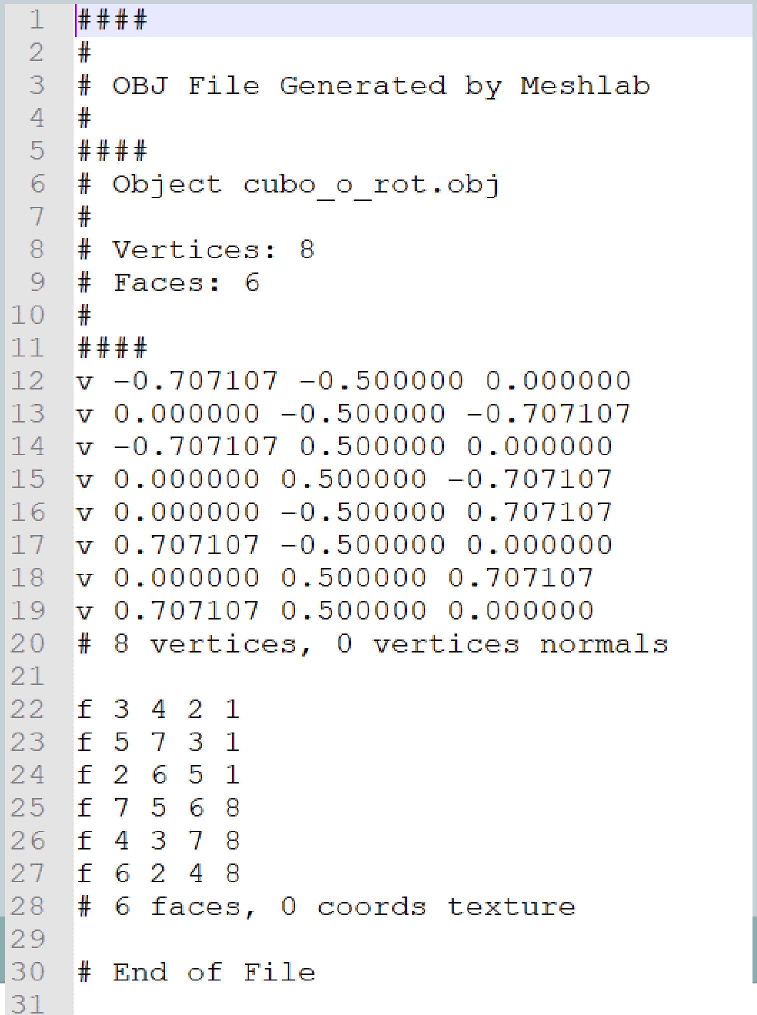


```
####
#
# OBJ File Generated by Meshlab
#
####
# Object cubo_o.obj
#
# Vertices: 8
# Faces: 6
#
####
v 0.0 0.0 0.0
v 1.0 0.0 0.0
v 0.0 1.0 0.0
v 1.0 1.0 0.0
v 0.0 0.0 1.0
v 1.0 0.0 1.0
v 0.0 1.0 1.0
v 1.0 1.0 1.0

# 8 vertices, 0 vertices normals

f 1 3 4 2
f 1 5 7 3
f 1 2 6 5
f 8 7 5 6
f 8 4 3 7
f 8 6 2 4

# 6 faces, 0 coords texture

# End of File

```

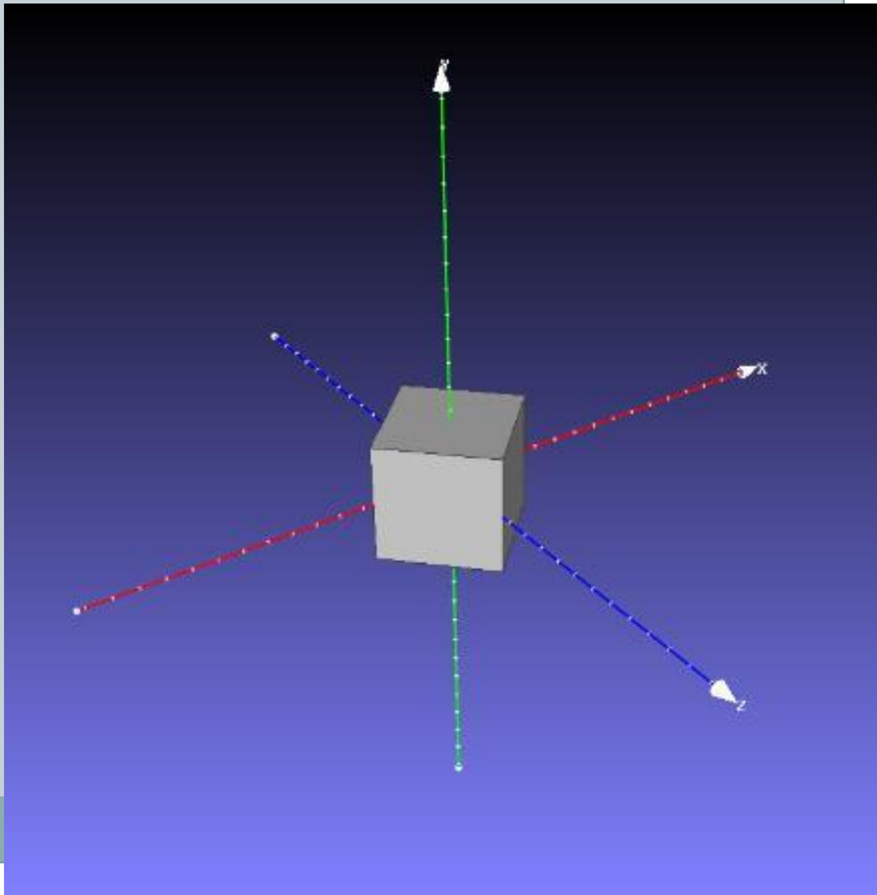


A simple cube

Coordinates & rendering

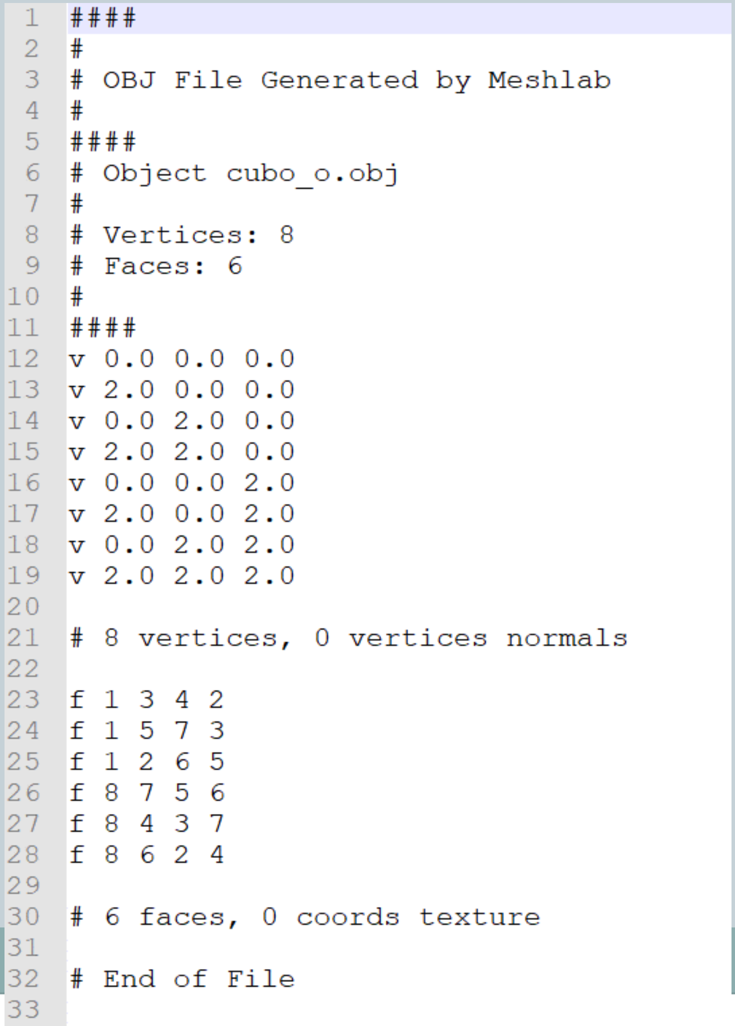


```
####
#
# OBJ File Generated by Meshlab
#
####
# Object cubo_o_rot.obj
#
# Vertices: 8
# Faces: 6
#
####
v -0.707107 -0.500000 0.000000
v 0.000000 -0.500000 -0.707107
v -0.707107 0.500000 0.000000
v 0.000000 0.500000 -0.707107
v 0.000000 -0.500000 0.707107
v 0.707107 -0.500000 0.000000
v 0.000000 0.500000 0.707107
v 0.707107 0.500000 0.000000
# 8 vertices, 0 vertices normals

f 3 4 2 1
f 5 7 3 1
f 2 6 5 1
f 7 5 6 8
f 4 3 7 8
f 6 2 4 8
# 6 faces, 0 coords texture

# End of File

```

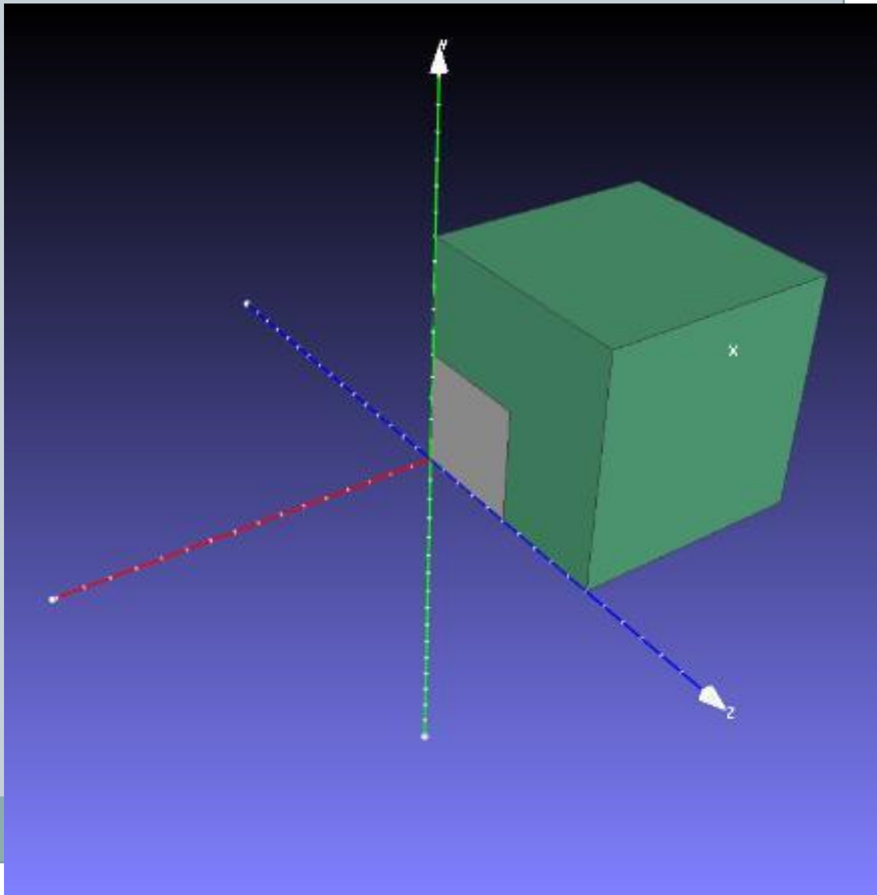


A simple cube

Coordinates & rendering



```
####
#
# OBJ File Generated by Meshlab
#
####
# Object cubo_o.obj
#
# Vertices: 8
# Faces: 6
#
####
v 0.0 0.0 0.0
v 2.0 0.0 0.0
v 0.0 2.0 0.0
v 2.0 2.0 0.0
v 0.0 0.0 2.0
v 2.0 0.0 2.0
v 0.0 2.0 2.0
v 2.0 2.0 2.0

# 8 vertices, 0 vertices normals

f 1 3 4 2
f 1 5 7 3
f 1 2 6 5
f 8 7 5 6
f 8 4 3 7
f 8 6 2 4

# 6 faces, 0 coords texture

# End of File
```



Vertices

Inside a 3D model, the coordinates of the vertices determine:

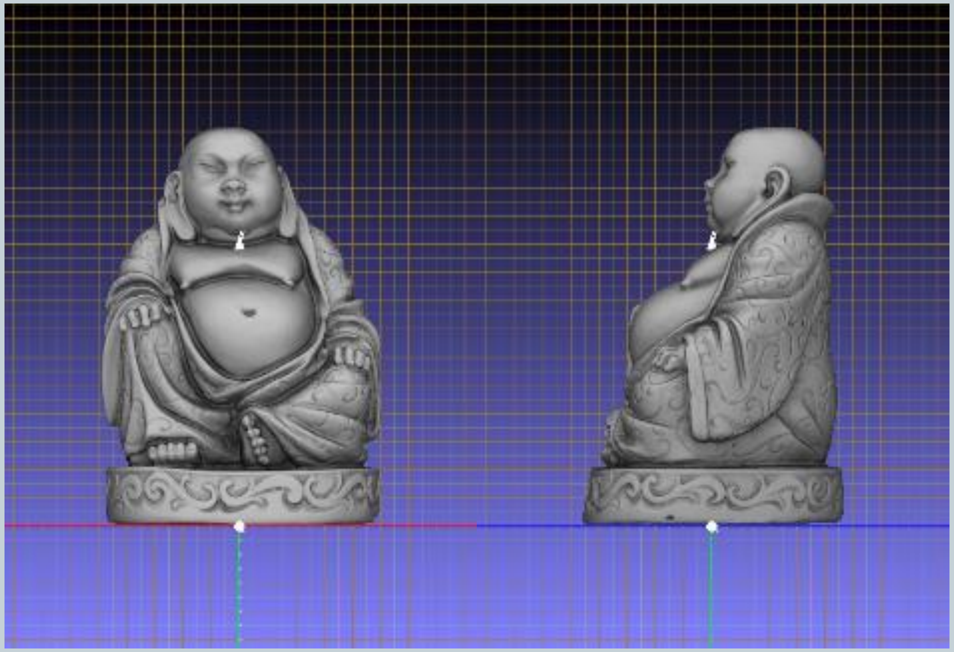
1. Shape and geometric properties of the object
2. Orientation and position in space of the object

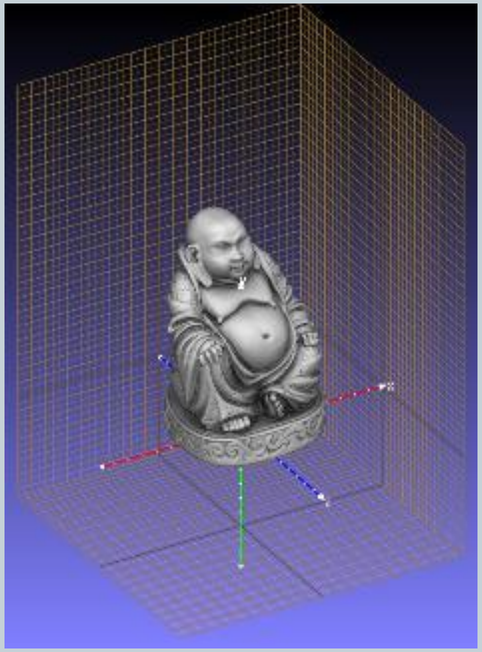
Depending on how I change the coordinates, I can change orientation and position, while keeping the geometric properties.

«well placed» model

Ideally, I want all my models:

- Geometry «straight» with respect to the axis
- Origin somewhere meaningful





TRANSFORMATIONS

- **TRANSLATION**: by adding a vector [tx,ty,tz] to all [xyz] coordinates, I can move the 3D object around
- **ROTATION**: using a set of formulas, I can calculate new coordinates, changing the orientation in space of the object
- **SCALING**: by multiplying all coordinates for a factor S: [s*x, s*y, s*z]: I can change the size of the 3D object, or I can change the measure unit of a 3D model (see later).

All these transformation may be represented as a MATRIX

SCALE

Measure unit

In almost all 3D file formats, there is no way to specify what is the measure unit.

Notable exception: Collada

The 3D model just contains numbers, it is up to you to know which is the measure unit of your 3D data.

From the practical points of view, you'll mostly find data in either **meters** or **millimiters**.

Commonly

Scanners for small objects (triangulation) generally save their data in **millimeters**.

Long-range scanners for building and survey generally save their data in **meters.**

3D data coming from long-range scanning in US *might* be in feet, as the cadastre does require measures in feets by law (although quite uncommon to find).

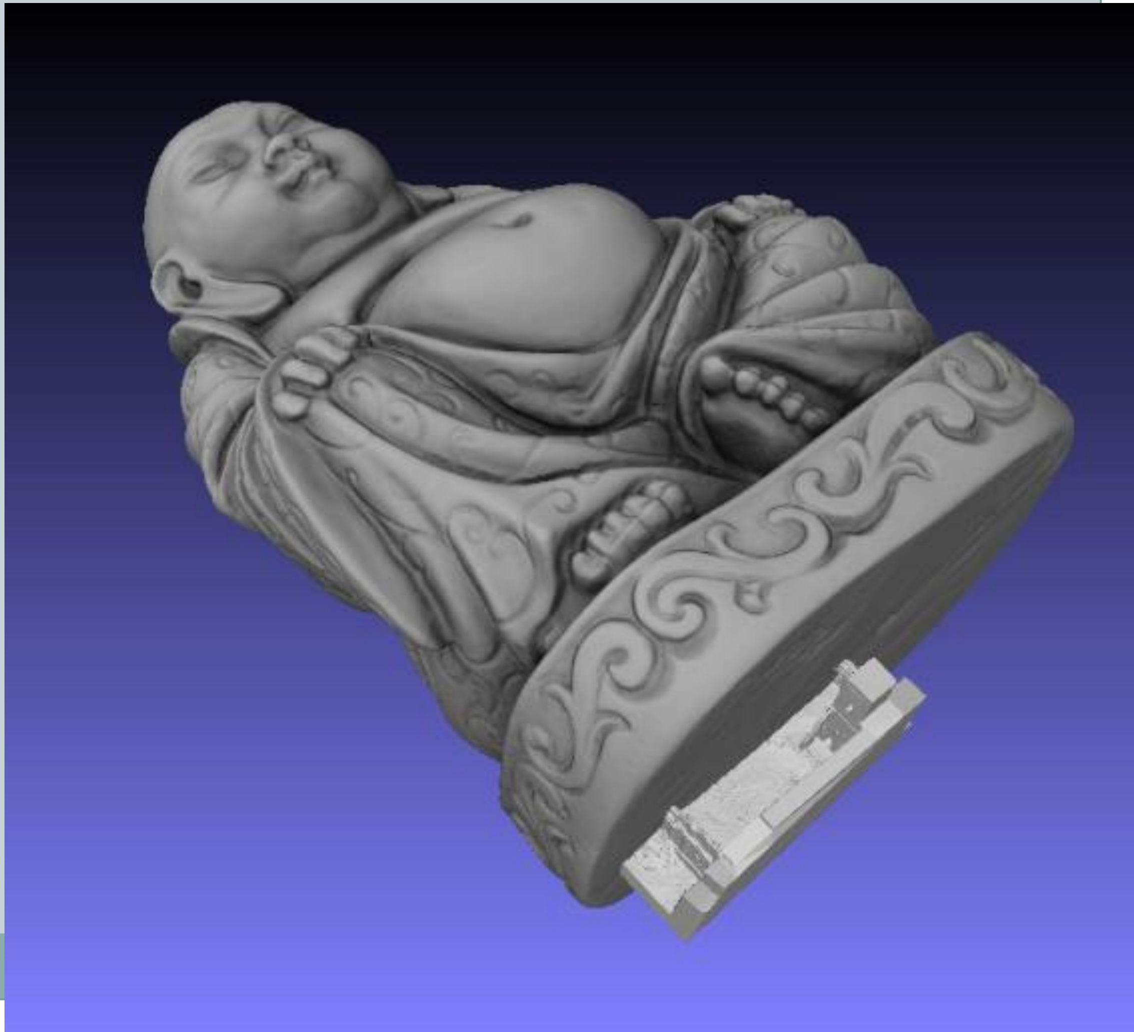
Other fields use other measures (e.g. molecular data is in angstrom)

Commonly

When you load 2 models with a different measure unit….

In most tools, the model is loaded and displayed as it is, resulting in inconsistencies

Some tools assume a specific measure unit, e.g. 1 Blender unit = 1 meter. Anything not in meters may appear too large or too small



Change of scale

Chaging the measure unit it is a matter of SCALING the object, i.e. multiplying all coordinates of an object by a factor.

To go from meters to millimeters, all coordinates must be multiplied by 1000. From millimeters to meters, multiply by 0.001

Setting of scale

3D data generated by photogrammetry / 3D-from-images does not have a scale. The object is perfectly fine, but the scale is unknown.

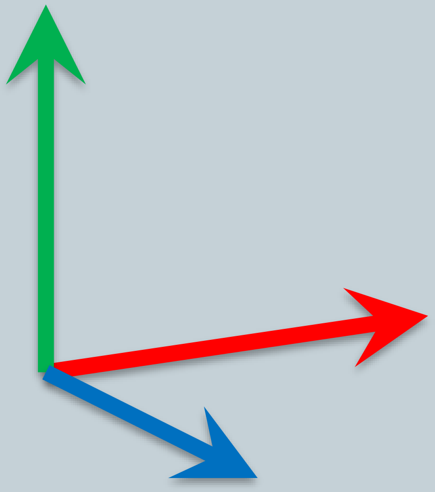
In this case, you need to take a measurement on both the physical object and the 3D model, calculate the scale factor as **physical_dist / model_dist** and then scale the model using this scale factor.

X Y Z

Unfortunately

The 3D space is defined by an ORIGIN and three orthogonal directions we call AXIS.

However, also this simple concept can cause inconsistencies

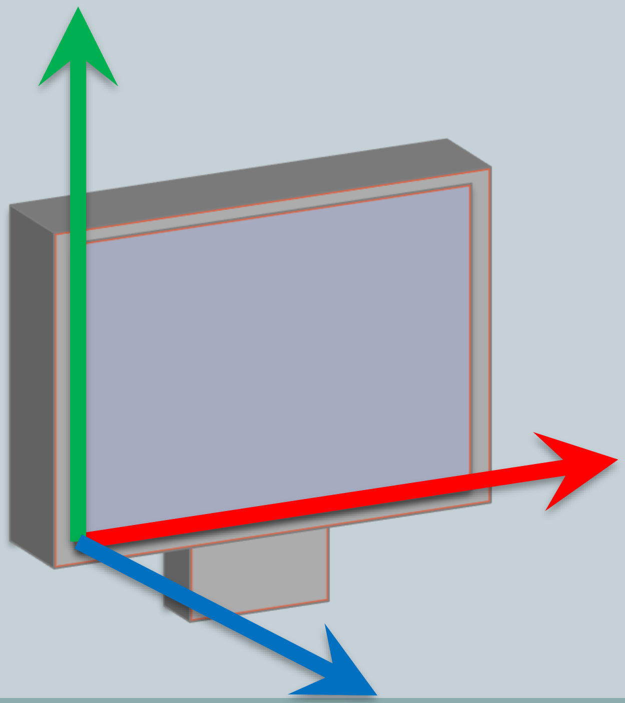


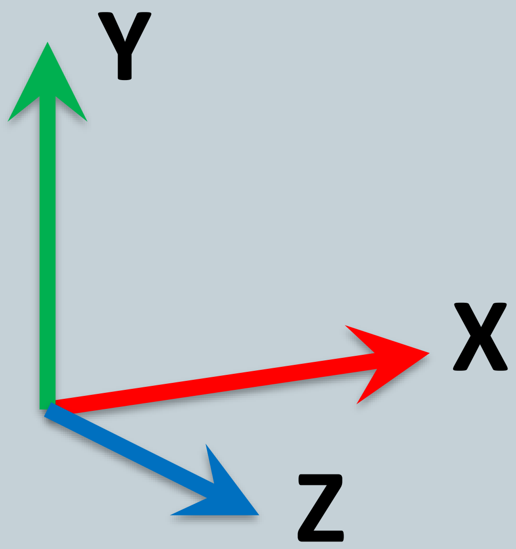
Computer Graphics

X: horizontal axis, grows towards right

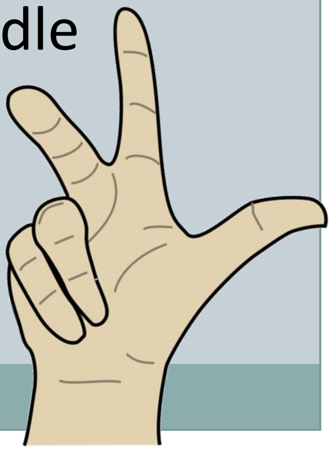
Y: vertical axis, grows towards up

Z: depth axis, grows out of the screen





Right-hand reference system:
XYZ as thumb, index, middle

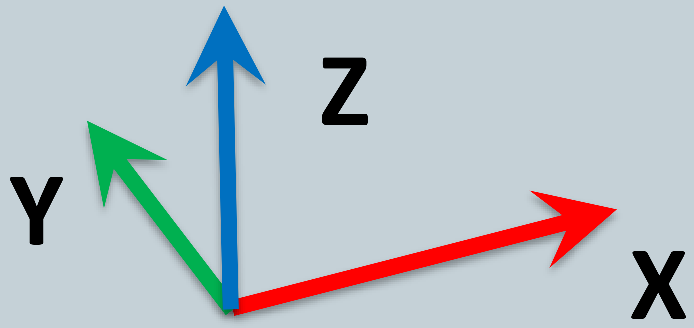


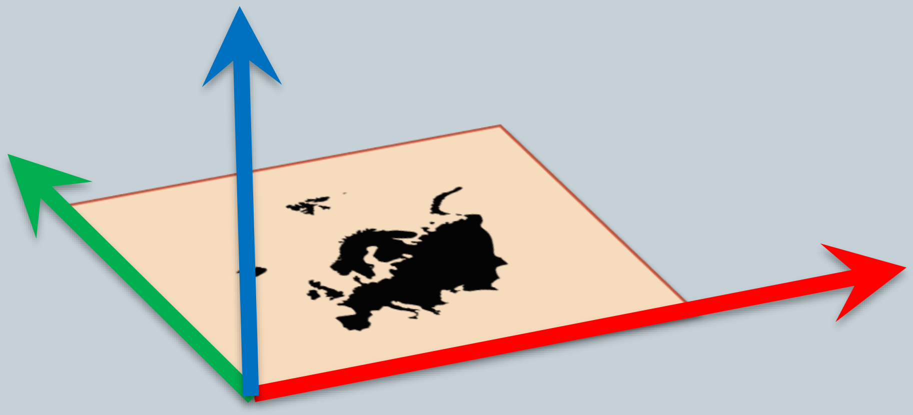
Engineering/survey

X: east-west, grows towards east

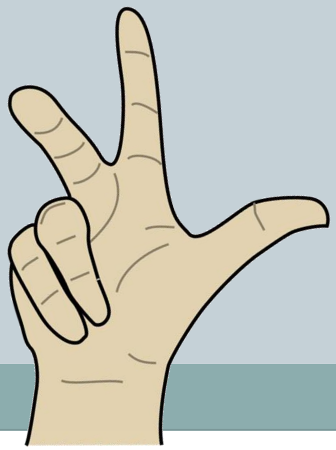
Y: north-south, grows towards north

Z: elevation, grows towards the sky



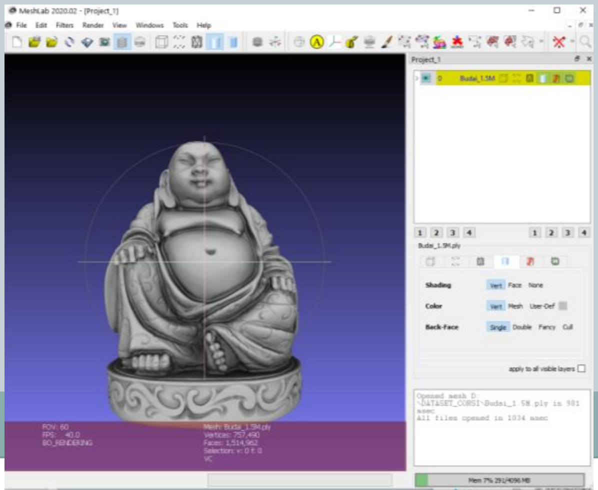


Right-hand reference system:
XYZ as thumb, index, middle

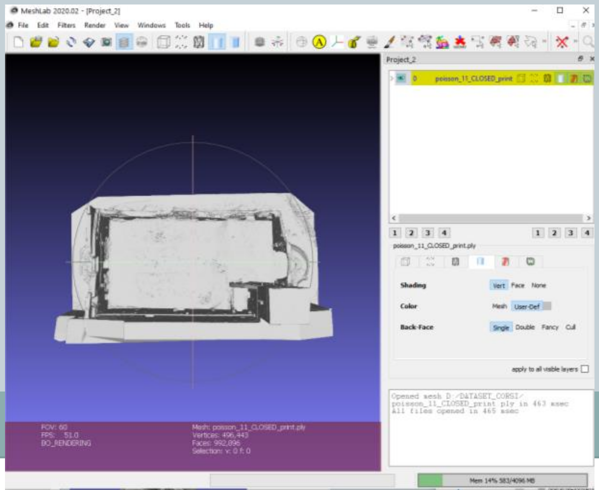


2 models, 2 orientations

- Most models created for CG use, in CG tools, use the CG reference space. Most models of large scale objects, like buildings or terrains use the engineering reference space.
- There are tools using CG reference (MeshLab, Sketchfab, 3DHOP), other use Eng. reference (Blender, CloudCompare).
- Beware, some importer/exporter do try «guessing» the orientation and flip the axis. Just be mindful.


MeshLab 2020.02 - [Project_1]
Budai_1.5M
Shading Vert Face None
Color Vert Mesh User-Def
Back-Face Single Double Fancy Cull
apply to all visible layers
Mem 7% 291/4096 MB

```
Opened mesh D:
\DATASET_CORSI\Budai_1.5M.ply in 901
msec
All files opened in 1034 msec
```


MeshLab 2020.02 - [Project_2]
poisson_11_CLOSED_print
Shading Vert Face None
Color Mesh User-Def
Back-Face Single Double Fancy Cull
apply to all visible layers
Mem 14% 583/4096 MB

```
Opened mesh D:/DATASET_CORSI/
poisson_11_CLOSED_print.ply in 463 msec
All files opened in 465 msec
```

It’s always *RIGHT ;)*

Fortunately, both reference systems are RIGHT hand, they can be converted by a simple rotation around X.

There are also LEFT hand reference system, but they are becoming rare, and not used in the CH world. In that case, the objects will flip their orientation like in a mirror.

Anyway, be careful when moving data from one software to another.

Geography

reference space

The 3D space is defined by an ORIGIN and three orthogonal directions we call AXIS.

Where is the origin? Somewhere meaningful. It is our choice. For small objects, statues, things taht can be moved, we may safely choose our origin and axis orientation. Clearly, we have to take sensible decisions.

Fo buildings and terrains, however, we may want to use a reference space that can be used in a «geographical» sense.

"absolute" reference space

Earth is round (almost) and it is comple

There are MANY geographical reference spaces, each one has a specific origin and axis, plus a lot of other information to determine the real position on the geoid

"absolute" reference space

Problem:

With absolute/geographical reference spaces, the coordinates of the 3D models become very, very large (millions).

This is fine in GIS and dedicated software, but it cause lots of problems in CG tools.

Solution:

use a **local** origin, and store the offset from the original origin

Non cartesian

Beware, some reference spaces are NOT cartesian: a vertex is still 3 coordinates, but are:

- **Latitude (°)**
- **Longitude (°)**
- **Elevation (meters)**

This data can still be saved in a normal 3D file format, but they are almost impossible to use inside “standard” CG tools. They are used in GIS tools and other specific software.