

# 3D models preparation

Single-res, multi-res, meshes, point-clouds

<http://3dhop.net> 02/2023

# Supported Models

Which 3D formats, how to use them

# 3DHOP and 3D models

---

3DHOP can manage three types of geometries:

- ◆ **Single resolution 3D models**

- Triangular meshes or point-clouds, ideally <1MB, PLY file format, whole file downloaded from remote and directly rendered without optimization.

- ◆ **Multiresolution 3D models**

- Triangular meshes, large to huge sizes are supported (1-100 million triangles), texture is supported. They are converted in pre-processing to the Nexus multiresolution format, progressively streamed from remote and adaptively rendered.
- Point-clouds, colors and normals are supported, large to huge sizes are supported (1-100 million points). They are converted in pre-processing to the Nexus multiresolution format, progressively streamed from remote and adaptively rendered.

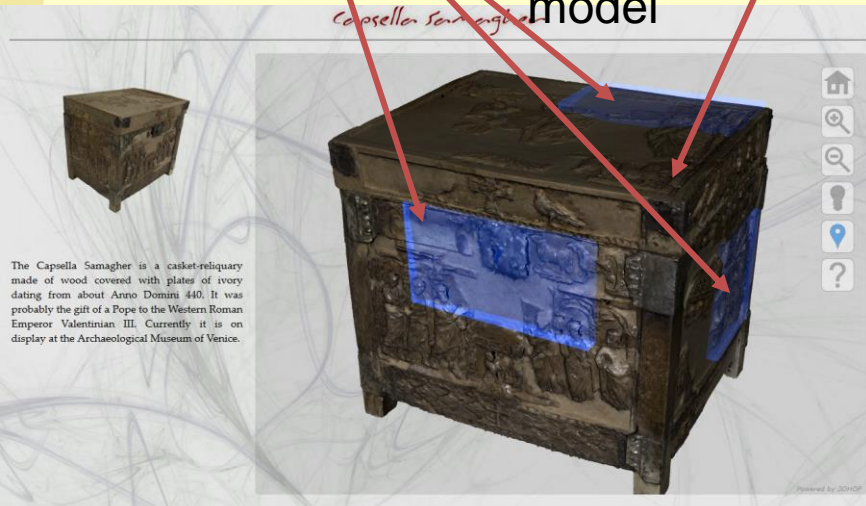
# Instances & hotspots

Any geometry (single-res, multi-res, mesh, point-cloud) may be used for both INSTANCES and clickable HOTSPOTS.

Geometries may be replicated in the scene (many instances or spots using the same geometry).

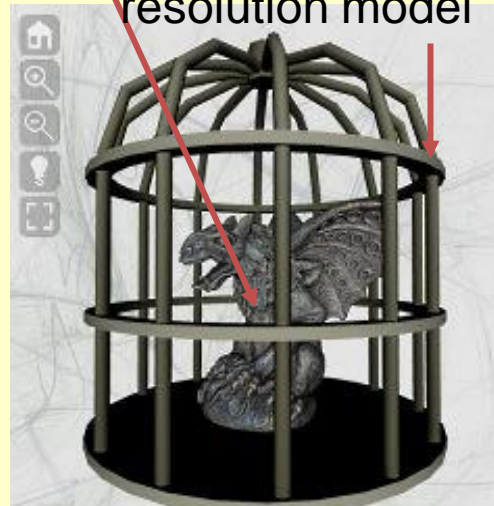
Same single resolution model used in multiple hotspots

Multiresolution model



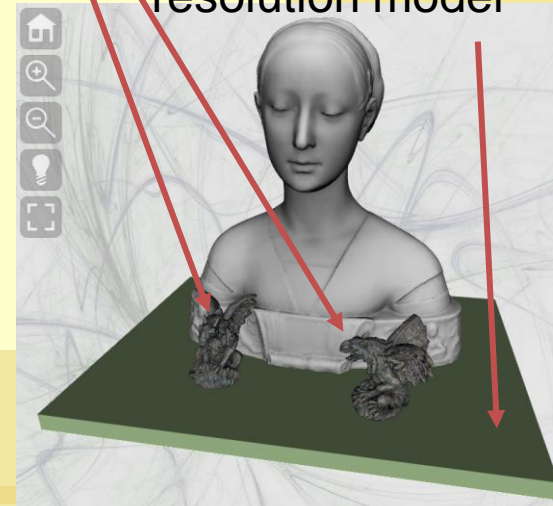
Multiresolution model

Small (86KB) single resolution model

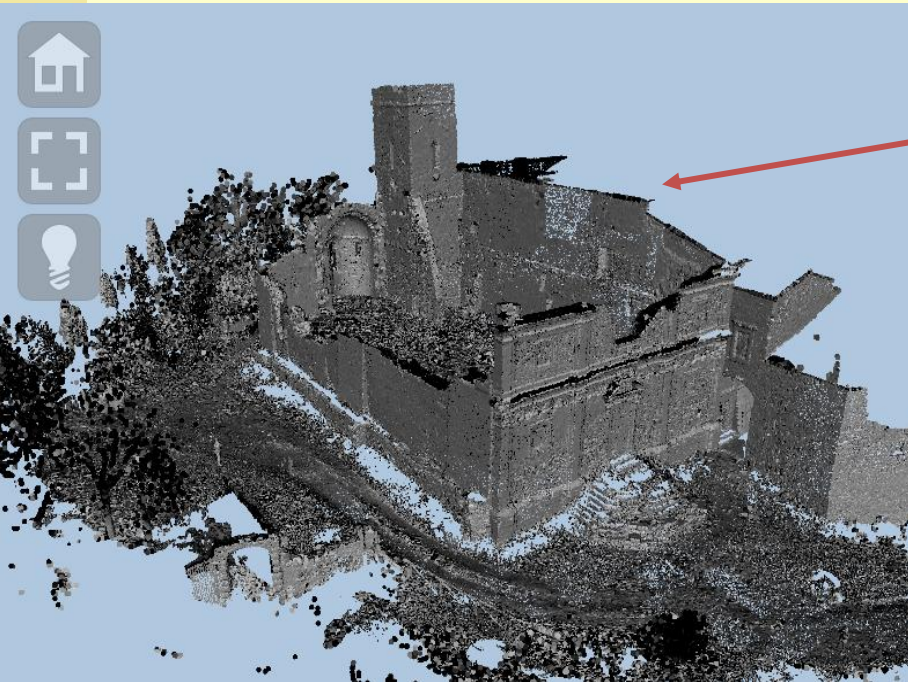


Same multiresolution model instanced 2 times

Small (2KB) single resolution model



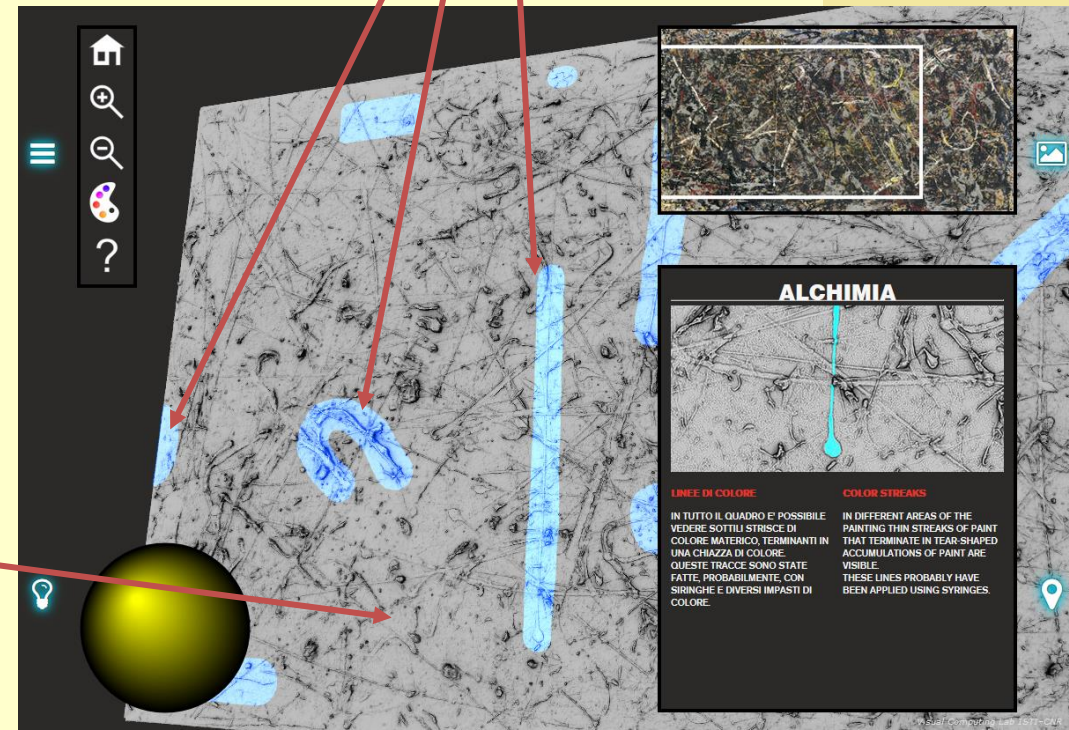
# Instances & hotspots



Multiresolution model  
(point-cloud with normal)

Multiresolution models (1M faces each)  
used as clickable hotspots

Huge multiresolution model  
(triangular mesh, 80M faces)



# Single resolution

Triangular meshes or point-clouds

Small geometries

# Single resolution

---

Standard 3D model: a mesh of triangles, or a set of points. The file is downloaded from remote in a *single chunk* and rendered inside 3DHOP.

Per-vertex color is supported, texture is supported. **Texture mapping should be per-vertex**; there is a filter in MeshLab to convert per-wedge UV to per-vertex UV (filters -> texture -> convert PerWedge UV into PerVertex UV).

Vertex normals **must** be included in the file.

# Small geometries, please

---

Single-resolution models are read from the server as a single file and have to be **COMPLETELY DOWNLOADED** before they appear on the webpage.

NO optimization is used during rendering, everything is drawn as it is.

Any model that is above the 1 MB limit **should be ideally** converted into the multiresolution format !



# Model preparation

---

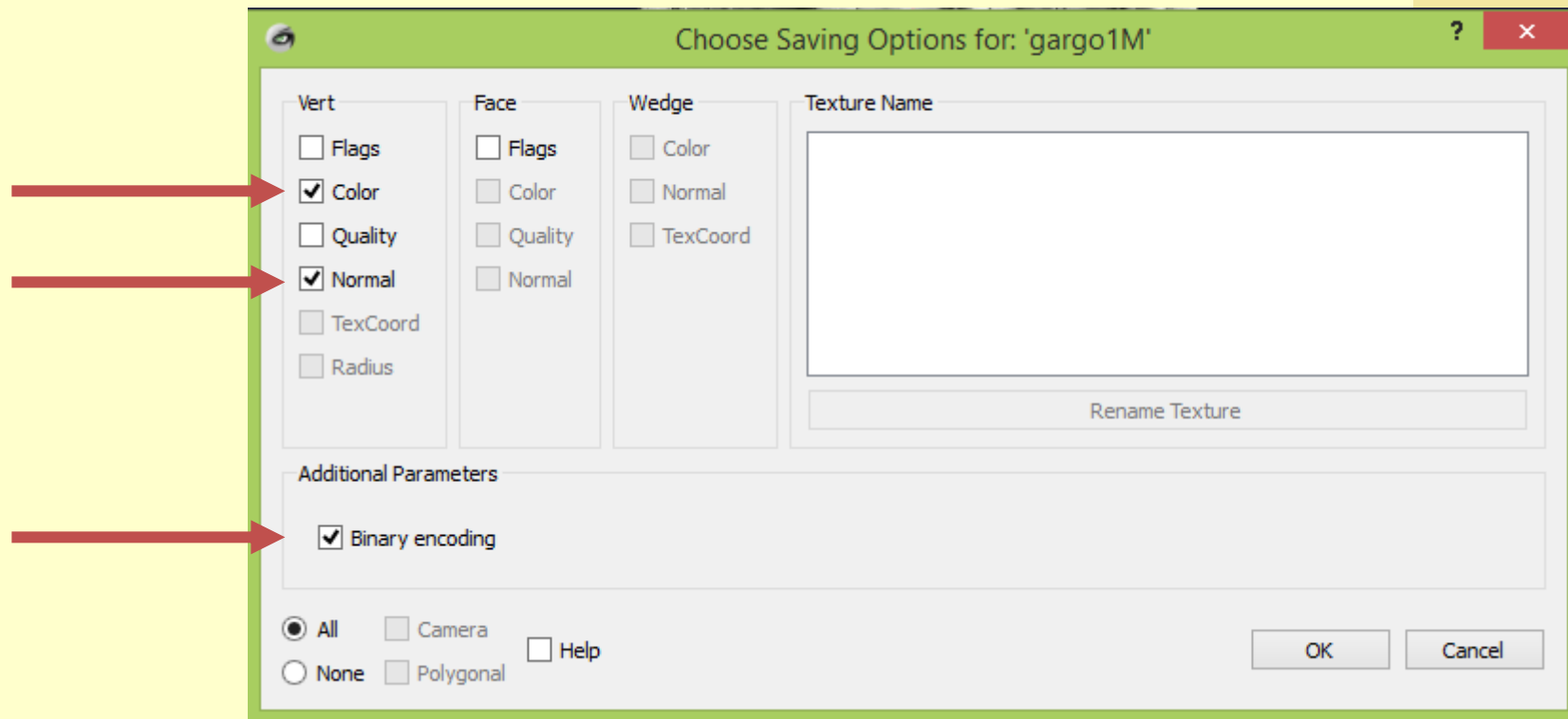
PLY file format is used for single resolution.

Use MeshLab ([www.meshlab.net](http://www.meshlab.net)) to convert your 3D model in PLY from other formats.

Some PLY exporters available in different tools save in ASCII format (much larger) or may have small incompatibilities, loading the file in MeshLab and then exporting it does generally solve the problem.

# Model preparation (no texture)

When exporting, remember to check “normal” for vertices.  
If the mesh has vertex color, check “color” for vertices.  
Save in binary format to save space.  
For point-clouds, do exactly the same.

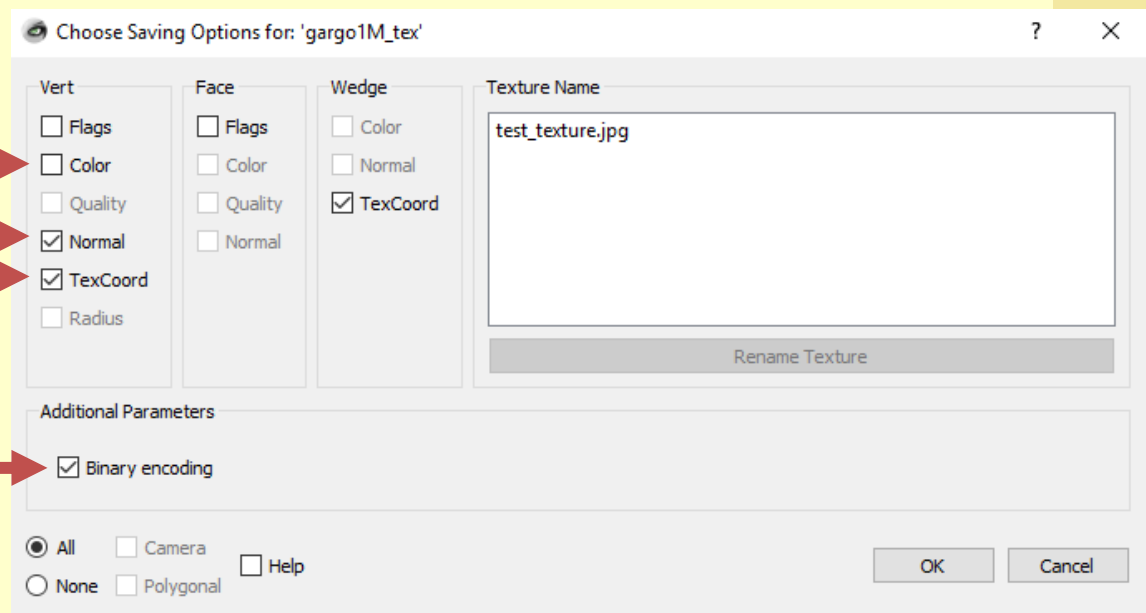


# Model preparation (with texture)

When exporting, remember to check “normal” for vertices. **Uncheck** “color” and be sure “TexCoord” is checked. Save in binary format to save space.

**Both** the PLY file **and** the texture image file have to be put online

Texture image has to be in a **single file with resolution square powers of two: 128x128, 256x256, 512x512, 1024x1024, etc.**



# Multiresolution

Triangular meshes

Lots of geometry

# High-res geometry

---

Multiresolution is used to manage high-resolution triangular meshes inside 3DHOP.

Multiresolution models are streamed from remote and adaptively rendered and optimized according to viewpoint and view distance.

Very large 3D models (1-10-100 millions of triangles) can be efficiently rendered.

3DHOP uses the NEXUS multiresolution format (.nxs or .nxz) (more info at [vcg.isti.cnr.it/nexus](http://vcg.isti.cnr.it/nexus)).

# High-res geometry

---

Per-vertex color is supported

Texture mapping is supported:

- Texture image(s) should be PNG or JPG
- **Texture mapping should be per-wedge;** there is a filter in MeshLab to convert per-vertex UV to per-wedge UV  
(filters->texture->convert PerVertex UV into PerWedge UV)

# Nexus

---

A **NXS** file is created starting from a standard single-resolution model, using a conversion tool.

NXS may be compressed into **NXZ** for smaller server occupation and better network performances.

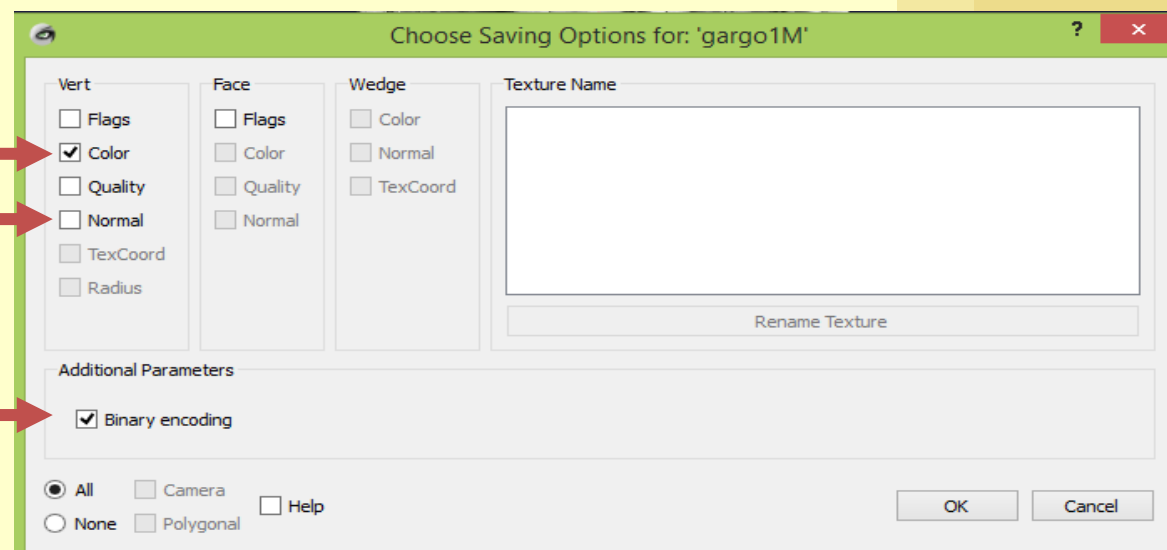
Compression is slightly lossy, but greatly increases network performances.

For local, non-network use (e.g. local kiosks), uncompressed NXS may have better performances.

# Model preparation (no texture)

Conversion to multiresolution requires PLY file format. Use MeshLab to convert from other formats to PLY. Some PLY exporters available in different tools save in ASCII format (much larger) or may have small incompatibilities, loading the file in MeshLab and then exporting it is safe. Save in binary format, if the mesh has vertex colors, check “color” for vertices when exporting.

**Vertex normals are not required** (since Nexus re-calculates normals during the processing), so uncheck “normal” when exporting.



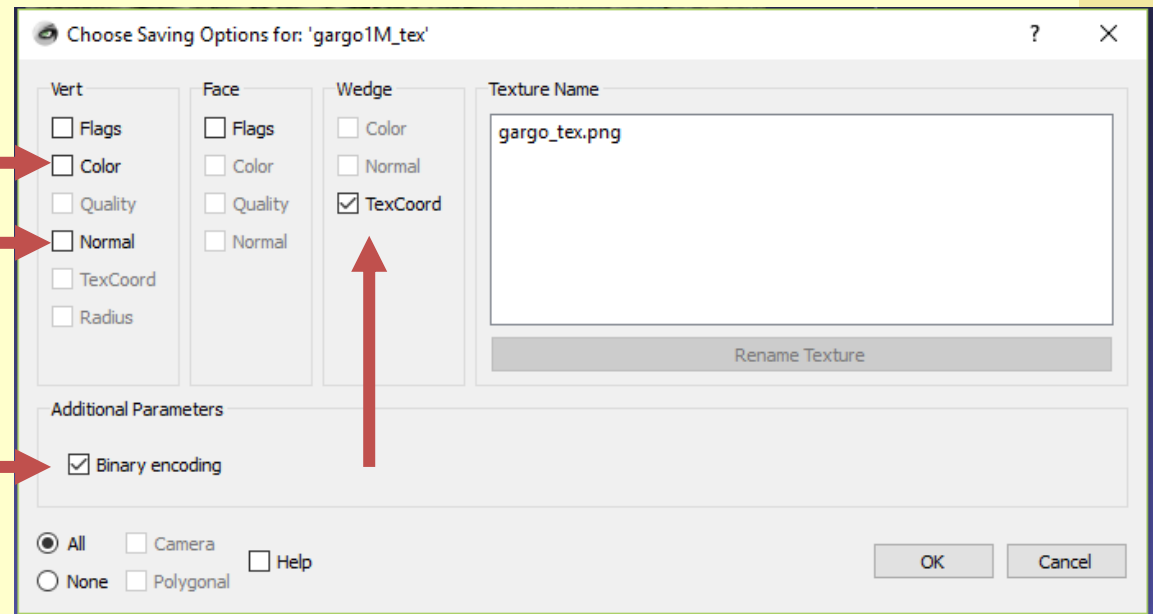


# Model preparation (texture)

If the mesh has a texture, check "TexCoord" under the WEDGE tab. Be sure the texture file name is correct.

Uncheck "color" under the VERT tab.

**Vertex normals are not required** (since Nexus re-calculates normals during the processing), so uncheck "normal" when exporting.



# PLY to Nexus conversion

---

Conversion tool is available from the download page of the 3DHOP website: [3dhop.net/download/](http://3dhop.net/download/)

Download and unzip in a folder.

Inside the folder you will find:

- *readme.txt*, with instructions
- *nxsbuild.exe*, for the PLY→NXS conversion
- *nxscompress.exe*, for NXS →NXZ compression
- *nxsview.exe*, for the NXS files visualization
- A batch file (*Build\_Nexus.bat*), for easier batch conversion
- A batch file (*Compress\_Nexus.bat*), for easier batch compression
- A PLY example file (*gargo.ply*), to test if everything works
- DLLs and support files

# Conversion tool

---

The conversion tool is available, at the moment, only for the Windows platform.

Mac and Linux users may use a windows emulator, or build from the source code available from:

[vcg.isti.cnr.it/nexus](http://vcg.isti.cnr.it/nexus)

3D models may be converted in different ways, absolutely equivalent:

- **Drag & Drop conversion (simpler option)**
- Using the .bat file for a batched conversion
- Directly launching the executable from command line

# Drag & drop conversion / compression

---

- Put the **PLY** to be converted in the folder where the NEXUS tool is located.
- Drag your input PLY file on the **nxsbuild.exe** executable. A command line window will open, showing the process; when the program terminates, the conversion is done. The result has the same name of the ply file, but has **NXS** as extension.
- You may then compress your NXS file: drag the NXS file on the **nxscompress.exe** executable. A command line window will open; when the program terminates, the conversion is done. The result has the same name of the NXS file, but now has **NXZ** as extension.

# .BAT batch conversion

---

This method is useful when you need to convert multiple files:

- Put the PLY(s) to be converted in the folder where the NEXUS tool is located.
- Open the **Build\_Nexus.bat** file with a text editor.
- Change the name of the source and target file, from this:

```
nxsbuild gargo.ply -o output.nxs
```

To this:

```
nxsbuild YOURMODELNAME.ply -o YOURMODELNAME.nxs
```

- Save and close
- Double-click on Build\_Nexus.bat. A command line window will appear; when the window closes, the conversion is finished

# .BAT batch conversion

---

To convert multiple files, just add other lines in the .bat files, like this:

```
nxsbuild MODEL1.ply -o MODEL1.nxs  
nxsbuild MODEL2.ply -o MODEL2.nxs  
nxsbuild MODEL3.ply -o MODEL3.nxs
```

The batch will convert one model at a time

# .BAT batch compression

---

Compression has to be applied *after* conversion in NXS. Again, it is possible to use the provided .bat file, by changing **`nxscompress gargo.nxs -o gargo.nxz`**

To  
**`nxscompress YOURMODELNAME.nxs -o  
YOURMODELNAME.nxz`**

Saving and double-clicking it. A command line window will appear; when the window closes, the compression is finished

The new .NXZ file is now compressed and may be used normally inside 3DHOP.

# .BAT batch compression

---

To convert multiple files, just add other lines in the .bat files, like this:

```
nxscompress MODEL1.nxs -o MODEL1.nxz  
nxscompress MODEL2.nxs -o MODEL2.nxz  
nxscompress MODEL3.nxs -o MODEL3.nxz
```

The batch will compress one model at a time



# Command line conversion/compression

---

It is also possible to directly execute the converter and compressor from command line

- Put the PLY to be converted in the NEXUS tool folder.
- Open a command prompt (cmd.exe), then go in the NEXUS folder.
- On the command prompt write this command line:  
**nxsbuid YOURMODELNAME.ply -o YOURMODELNAME.nxs**
- When the program terminates, the conversion is done.
- Then, to compress the NXS file, use this command line:  
**nxsccompress YOURMODELNAME.nxs -o  
YOURMODELNAME.nxz**
- When the program terminates, the conversion is done

# Troubleshooting

---

- ◆ Drag the .NXS file on nxsview.exe: if the model is visible, the conversion ended ok.
- ◆ Try converting the example PLY (gargo.ply) and check the result. If you receive no error, the converter works on your machine.
- ◆ If your PLY file makes the converter crash, try opening it and saving it again with MeshLab after cleaning (see MeshLab videotutorials <https://www.youtube.com/playlist?list=PLBBF41579E4B65566>).
- ◆ If using drag & drop or batch conversion the process prematurely quits, **try command line conversion for getting details about the error.**

# Multiresolution

Point-clouds

Lots of points

# Point-clouds

---

A **point-cloud** is an unstructured set of vertices.

Point color is supported, point normal is supported.

Point-clouds are managed inside 3DHOP using the multiresolution format, they are streamed from remote and adaptively rendered and optimized according to viewpoint and view distance.

Very large clouds (10-100 millions of vertices) can be efficiently rendered.

# Importing clouds

---

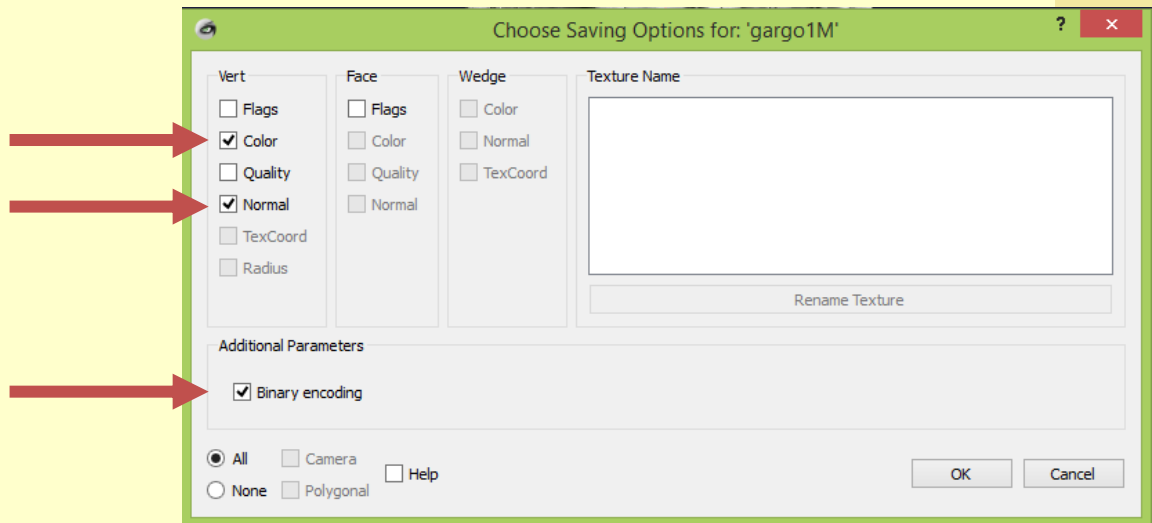
Point-clouds has to be transformed in PLY file format, and then converted in the NXS multiresolution format using the same tool used for triangular meshes.

**MeshLab** does import different formats of point-clouds, which can be later exported as PLY:

- ◆ .PTX (normals are automatically calculated)
- ◆ .PTS
- ◆ .XYZ
- ◆ .txt (generic ASCII list of points)

# Importing clouds

If the cloud has normals, when exporting, remember to check “normal”. If the cloud has color, remember to check “color”. In MeshLab, there is a tool to calculate normals for pointclouds; it may not work on your cloud, depending on its shape, but it is worth a try. Save in PLY binary format to save space.



# Conversion

---

Conversion is done in the same way it has been described for the multiresolution 3D models, using the Nexus converter drag& drop, .bat file or command line.

Just follow the same instructions detailed in the previous section.

Multiresolution point-clouds may also be compressed, again following the instructions already detailed in the previous section. Compression for point-clouds is, however, slightly more lossy than a triangulated mesh.

# Straightening

Correctly oriented geometries



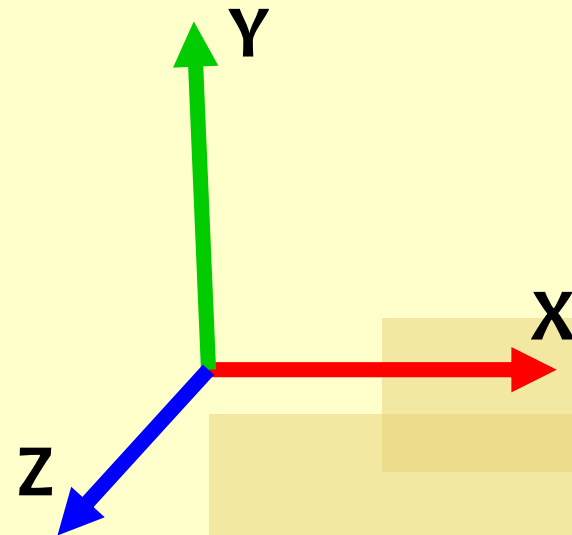
# 3DHOP axis

---

3DHOP uses the standard computer graphics reference system (right hand, Y is up).

Looking at the screen:

- **X** is the horizontal axis, growing towards right
- **Y** is the vertical axis, growing in the upward direction
- **Z** is the depth axis, growing towards you



# Model orientation

---

Unless a transformation is specified for the mesh or the instance, 3DHOP renders the geometries *as they are loaded*. If your object is not straight, or has a different up vector (architectural models generally uses Z as vertical axis), it may be better to put it in the correct reference space *before* loading it inside 3DHOP.

You may use MeshLab to rotate/translate/scale the model.

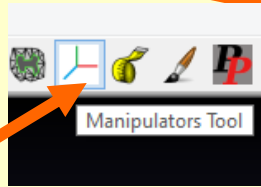
MeshLab uses the same reference system as 3DHOP, if it is straight in MeshLab, it is straight in 3DHOP

# Model orientation

A model may be rotated/translated/scaled using the filters under :

Filters->Normals, Curvatures and Orientation-> Transform: XXXXXX

Or by using manipulators:



<https://www.youtube.com/watch?v=FGaNv23Xvtw>

It is a good idea to have the world axis visualized to see how the model is moving

Remember to freeze matrix before exporting  
Filters->Normals, Curvatures and Orientation->Freeze Current Matrix

