From scan to 3D print

Method note

MCT-081

Innovation with Integrity

microCT

1. Introduction

It is not hard to see why 3D printing is a natural exponent to 3D imaging. In the latest decades 3D printing has become a mainstream technology. Its applications are numerous and include

- Customization (e.g. implants Bruker microCT User Meeting, 2010, Verschueren).
- Research and educational purposes (Bruker microCT User Meeting, 2014, Kleinteich)
- Rapid prototyping and production

In this method note we go through the key steps going from scanning to printing. As a plethora of hard- and software solutions exist in this field we do not aim to be exhaustive. We focus on the big essentials and illustrate for a specific sample, a snake.

2. From scan to print in 3D

2.1. Scan and reconstruct

These two steps in the process are well known to us all. The normal recommendations apply; here it can be of interest even more than usual not to go for the smallest pixel size but the optimal one for the sample. More pixels do not necessarily lead to better results, on the contrary, but will lead to larger datasets and models.

2.2. Binarize

The first step of interest might be to resize the dataset if the resolution was above required for the part to be printed. Thresholding proceeds in the usual manner, the aid of image filters can help in facilitating this (see MN077 - Image filtering in CTAn). Next a suitable ROI is defined. It is advisable to remove potential noise from the image by means of despeckling and/or morphological operations. This can have a great influence on the size of the eventual 3D model. Sweeping in 3D can ensure that only one object remains in the volume should this be of interest.

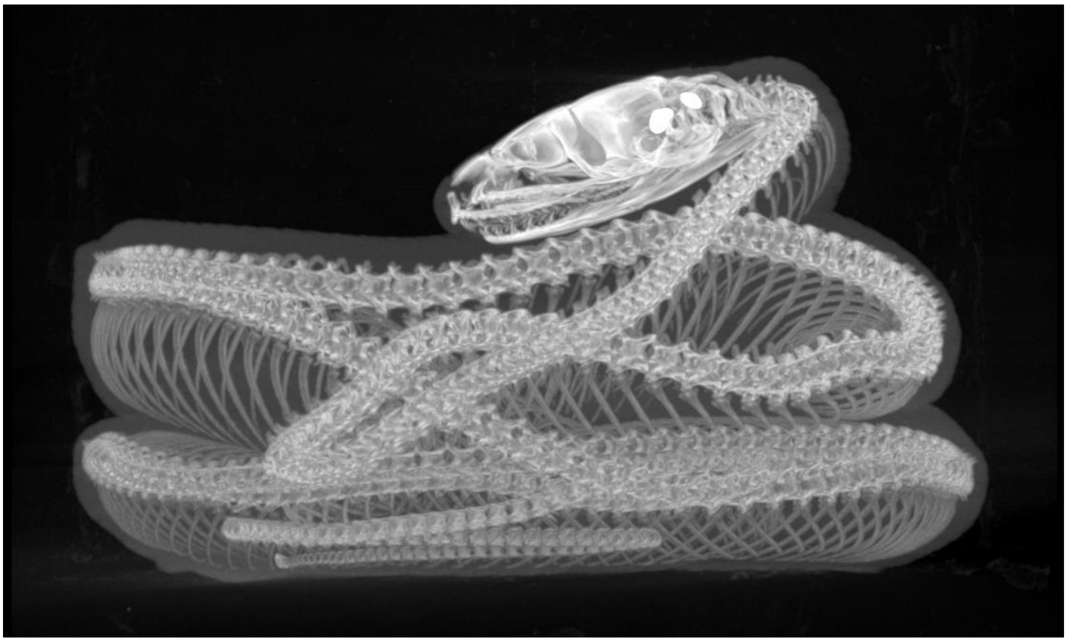

Figure 1. Maximum intensity projection of reconstruction slices of the snake sample. The scan was performed using the SkyScan1173 at an image pixel size of 80µm.

2.3. Meshing

Next a 3D model representing the surface of our binary object needs to be constructed. It consists of polygonal elements (in our case triangles) of which the vertices and normals are saved. The method note "MN017_Basic 3D surface rendering" describes the essential steps in CTAn. The program offers three possible meshing algorithms:

- The marching cubes algorithm (and its extensions) is probably the most popular isosurface meshing algorithm to day. The original paper on it is by Lorensen and Cline [1] and it is well described in literature. The algorithm considers the value of eight neighbouring voxels at a time and determines the polygon(s) needed to represent the surface in this cube. This surface is finally represented by triangles of which the vertices are stored.

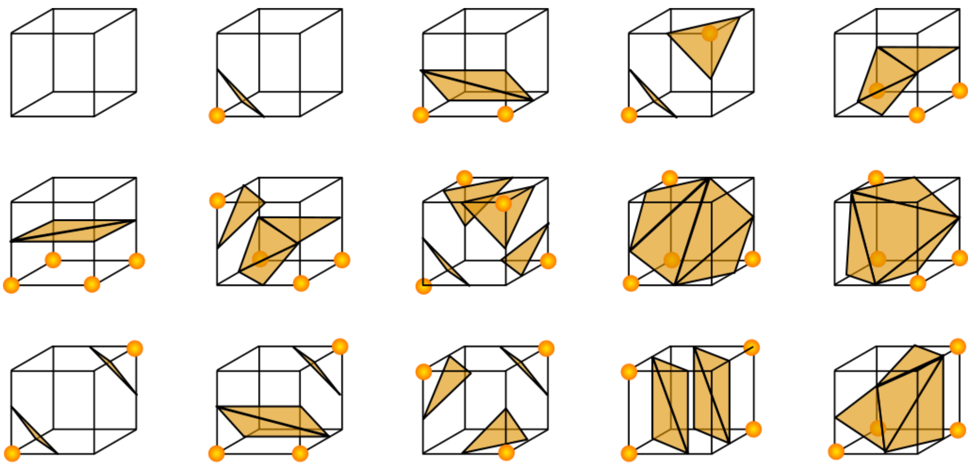

Figure 2. By exploiting rotational and reflective symmetry the marching cubes algorithm reduces the number of possible unique cases. The original 15 cases are shown above, reproduced from [2].

- Incidentally 3D morphometric analysis in CTAn also makes use of the marching cubes algorithm to calculate values for surfaces and volumes. This is because triangulated surface representations give more accurate surface area measurements than simple pixel counting techniques.

- Double time cubes [3] is a computationally less expensive algorithm than marching cubes. It leads to smaller models (approximately half the number of facet triangles as for marching cubes), with a more smoothed surface detail.

- Adaptive rendering is a Bruker microCT proprietary algorithm, the mesh characteristics depend on the selected parameter settings:

  - Locality defines the distance in pixels to the neighbouring point used for finding the object border. It hence takes the form of an integer with the range going from 1 to 100. Increasing the value allows “jumping” through noise on the object border, but small objects (often created by binarization of noise) with size less than this parameter will be lost.

  - Tolerance defines the sub-pixel accuracy with which the object border is delineated. The range goes from 0 to 1 pixel with steps 0.05 pixel. Reducing the number makes the model more accurate, but increases the model file size.

Four output formats are available in CTAn: stl, ply, ctm and p3g. The first two are commonly used file types, the latter two are Bruker microCT proprietary formats intended for our visualization software. Which file type is applicable will depend on the technology used in the printing step (i.e. compatibility). In some cases this implies remeshing and/or repairing the mesh using additional software mainly because of file size and/or compatibility issues. A vast number of software solutions exist with largely varying functionality, both in free (e.g. MeshLab) and in commercial form (e.g. ScanIP from Simpleware or Magics from Materialise). The examples are by no means exhaustive, such a review falls outside the scope of this document.

2.4. Inspection

Prior to printing we inspect the generated mesh, to which end the surface rendering program CTVol is of interest. Some iteration of the binarization and meshing steps might be required before a model accurately representing the object is obtained. Whilst double time cubes and adaptive rendering tend to deliver smaller models, the risk of e.g. preserving thin features less well exists.

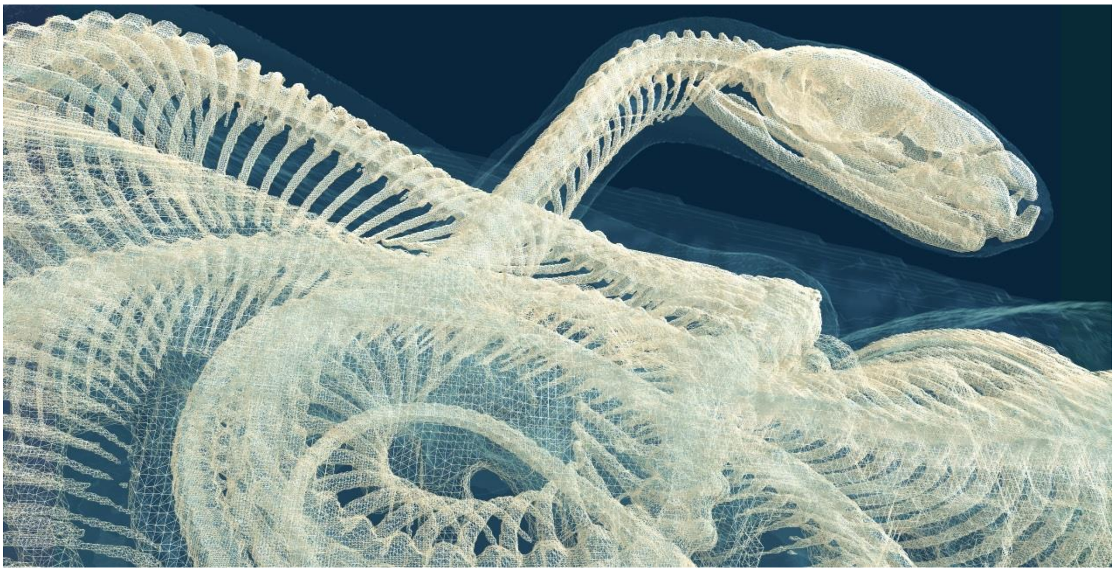

Figure 3. Surface rendering of snake model using CTVol. The model of the bones contains 1588804 facets and 793705 vertices.

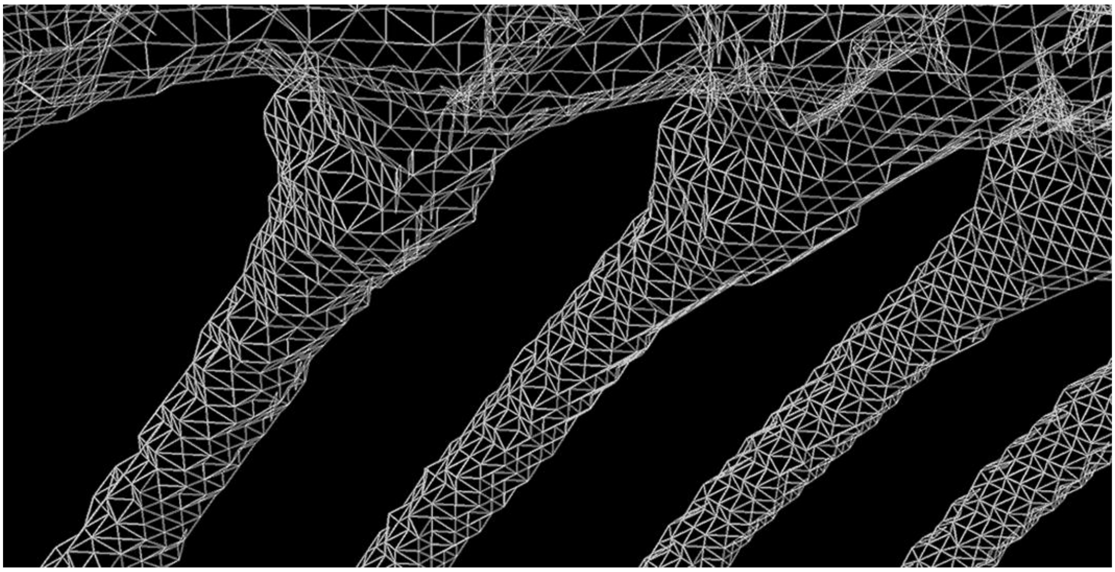

Figure 4. Close-up of the ribs highlighting the triangular nature of the mesh.

2.5. 3D printing

We import our model into the 3D printer software and rescale to the desired size. As a final step, depending on the type of printer, the printed parts can still need cleaning and/or finishing.

For the example of our snake we used a ProJet SD 3500 printer from 3D Systems. The software interface allows direct import of stl files generated by CTAn. A first print-heads delivers an acrylic plastic which will form the object; a second delivers a support material (wax) outside the sample volume. They are deposited and cured using UV light in a line by line manner. After printing the wax is removed in an oven at 65°C and the surface is cleaned in an ultrasonic bath containing water and a small amount of detergent. Bruker microCT offers the possibility to print in 3D using this system on a contract base.

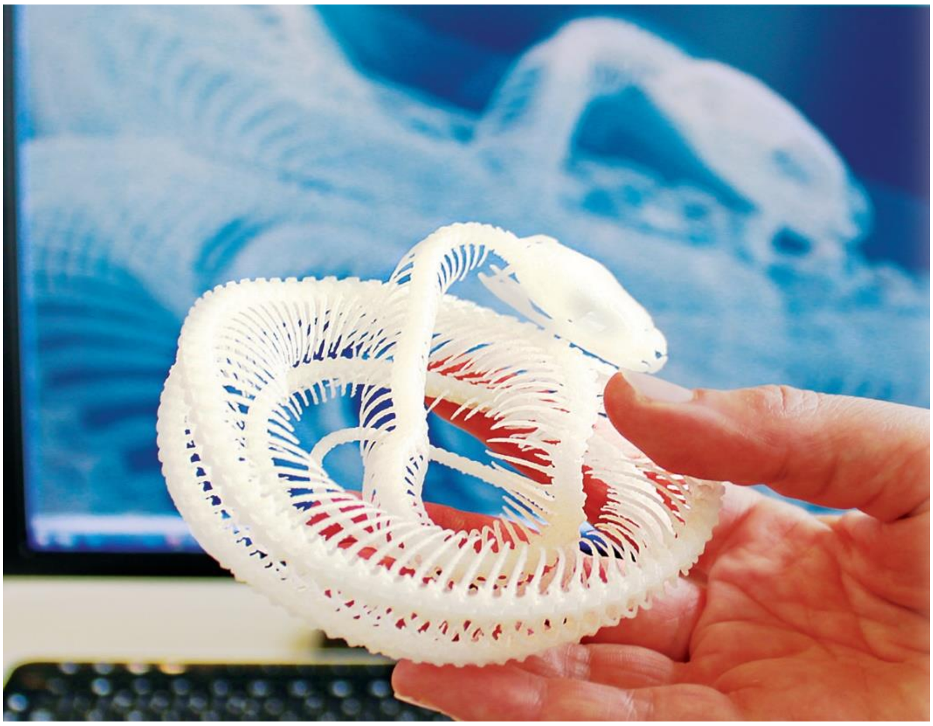

Figure 5. 3D print made of the snake skeleton.

References

1. W.E. Lorensen, H.E. Cline (1987), Marchin Cubes: A High Resolution 3D Surface Construction Algorithm. Computer Graphics, Volume 21, Number 4.

2. Wikipedia, Marching cubes, https://en.wikipedia.org/wiki/Marching_cubes

3. D.J. Bouvier (1997), Double-Time Cubes: A Fast 3D Surface Construction Algorithm for Volume Analysis. International Conference on Imaging Science, Systems and Technology.