

## Divide full data volume into defined sub-volume blocks for label analysis

This Xtra demonstrates how to use the Arithmetic module to divide a whole volume data into defined sub-volume blocks and then perform label analysis on the blocks.

This Xtra contains:

- A PDF tutorial file explaining the workflow
- A project (VolumeBlocks\_40x40x40.hx) reproducing step by step the tutorial
- A recipe (VolumeBlocks\_40x40x40.hxrecipe) automating the workflow

The tutorial data "10mc3\_200.vol.am" is composed of 200x200x200 voxels. In this tutorial, it will be sub-divided into smaller blocks of 40x40x40 voxels.

Note: isotropic sub-volume blocks (NxNxN voxels, where N=40) are generated in this example but the same technique can be used to generate anisotropic sub-volume blocks of MxNxP voxels. We will come back to this remark when presenting the arithmetic expression used for volume division.

### 1. Generate the sub-volume blocks

We create an Arithmetic module attached to the input data. To obtain a label field as output, we set the *Result Channels* port to 1 *value (label)*. We then type the following *Expression*:

$$((i\%40)\!=39) \&\& ((j\%40)\!=39) \&\& ((k\%40)\!=39)$$

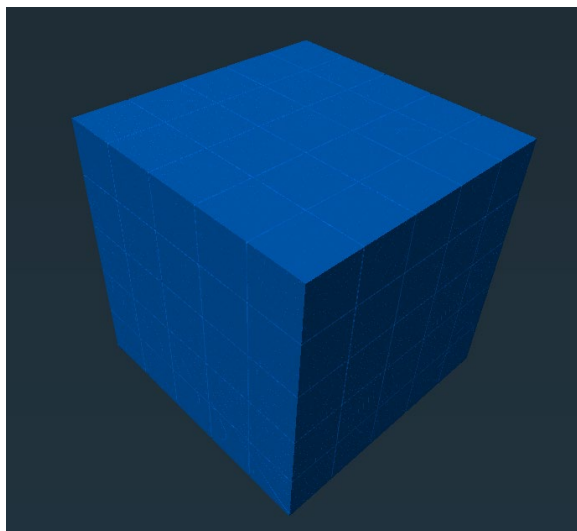
Note: for a NxNxN voxels isotropic sub-division of the input volume, the generic expression is:

$$((i\%N)\!=N-1) \&\& ((j\%N)\!=N-1) \&\& ((k\%N)\!=N-1)$$

And for a MxNxP voxels anisotropic sub-division of the input volume, the generic expression is:

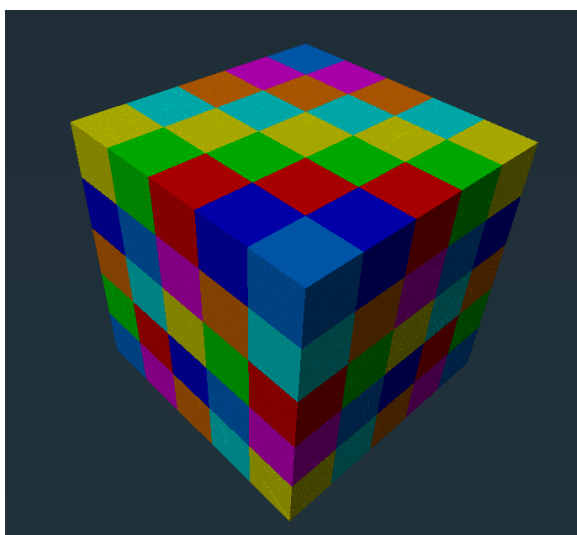
$$((i\%M)\!=M-1) \&\& ((j\%N)\!=N-1) \&\& ((k\%P)\!=P-1)$$

We press Apply and rename the result of Arithmetic into *Blocks*.



Separated sub-volume blocks

We then attach a Labeling module, with *Interpretation* set to *3D* and *Neighborhood* set to *26*, and an Expand Labels module with the same settings and *Type* set to *Fill All Image*. This generates a volume sub-divided into blocks that are each assigned a label.



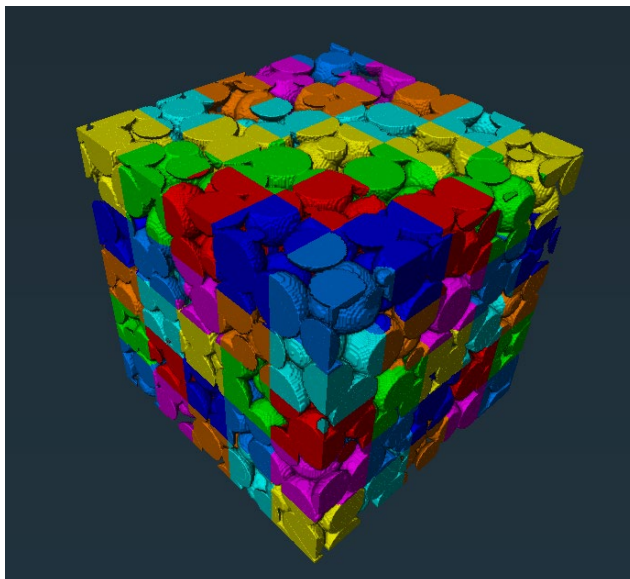
Labeled sub-volume blocks

## 2. Create the labels per blocks and perform the label analysis

Coming back to the input volume data, we attach an Auto Thresholding module to binarize it.

Then we create a second Arithmetic module that we connect to the thresholded data (as *Input A*) and to the volume sub-divided by blocks Blocks.expanded (as *Input B*). We then type the following *Expression*:  $(A > 0) * B$ .

This will assign to the thresholded voxels (i.e. voxels where  $A > 0$ ) the label value (B) corresponding to the block it belongs to.



Thresholded data sub-divided in labeled blocks.

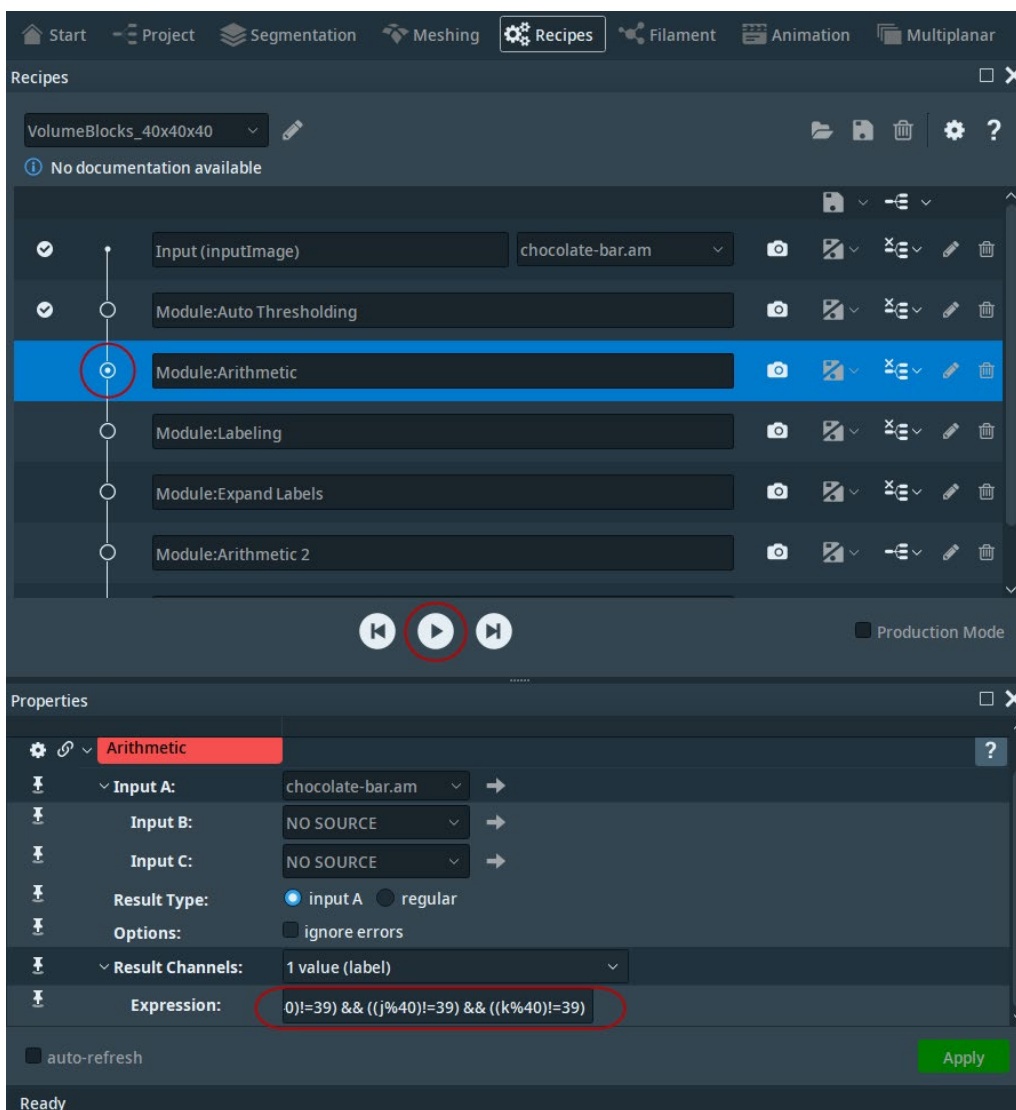
We can then perform a Label Analysis and compare results blocks by blocks.

Open VolumeBlocks\_40x40x40.hx to reproduce the workflow.

### 3. Use and edit the recipe

To reproduce the workflow on your data, do the following:

- Empty the Project View (Ctrl+N)
- Load your data
- Right click in the Project View, select Create Object and create a Recipe Player
- Select VolumeBlocks\_40x40x40.hxrecipe as *Recipe*
- Set your data as *Input data*
- If you press Apply, the recipe will run and output the thresholded data sub-divided in labeled 40x40x40 voxels block
- To edit the recipe, click on *Open*
- Add a breakpoint in front of the first Arithmetic module and play the recipe
- Once the Arithmetic module is displayed in the Properties area, you can edit the *Expression* field
- Save the recipe to be able to reuse it



Recipes workroom interface