University of Stuttgart
Institute for Visualization and Interactive Systems (VIS)

Prof. Dr. Thomas Ertl *Stuttgart, 08.05.2020*

Scientific Visualization (Assignment 3)

Exercise 3. 1 [2 Points] Cartesian Grids

Cartesian Grids can be useful in spaces with less or more dimensions than three. In the lecture you heard about indexing in these grids. Please provide a formula or code for mapping a 4D point in normalized coordinates $x \in [0 \dots 1)^4$ to a global cell index for a structured Cartesian grid of size $N_x \times N_y \times N_z \times N_w$.

For example, the point $(0.46, 0.57, 0.23, 0.68)$ in a grid of size $128 \times 64 \times 32 \times 16$ should be mapped to index 2683450.

Exercise 3. 2 [8 Points] ParaView Introduction: Point Splatting

In this exercise, we will setup a popular visualization package, ParaView, which will be used during the next assignments. Additionally you will have to adapt a first ParaView filter that manipulates given data. Since you will write custom plugins for ParaView you will need a self-compiled development version of it. As the compilation process of ParaView is quite tedious, we offer a Docker container containing an already compiled ParaView. You can download this pre-compiled container from https://cloud.visus.uni-stuttgart.de/index.php/s/1XGGaPNpZu4S5gx. Please do this in a timely manner as this container is quite large (1.4 GB).

To use the Docker version of ParaView you will need two different programs, Docker itself and a VNC client of your choice. While Docker provides the environment to run the ParaView container, the VNC client will enable you to see the visual output. You can download Docker from https://www.docker.com/get-started as binary for Windows and MacOS, or you can follow the guide at https://docs.docker.com/engine/install/ubuntu/ to set it up for Linux. As VNC viewer we used the one from RealVNC for all our tests. It is available at https://www.realvnc.com/de/connect/download/viewer/.

For most of the steps required for this task, we compiled step-by-step videos that show the necessary procedure. You can find the videos alongside with the assignment sheet in Ilias. Of course you can choose to build ParaView yourself to run it without any virtualization, but be aware that this is a very time-consuming task.

Step 1 - Setup Docker and the ParaView container

The process for installing Docker differs for each operating system. Please follow the aforementioned guides to set it up for your machine. Windows users should select to use Linux containers, as the image we are providing is an Ubuntu container. Once docker is installed and running, it can be used from the terminal of your choice (e.g. bash under Linux or PowerShell under Windows). To set up the *container* used for this assignment sheet, please navigate your terminal into the folder containing the *Dockerfile* shipped with this assignment sheet and execute the following commands.

```
$ docker load -i ./paraview-5.8.0.docker
$ docker build -t scivis-2020:sheet-3 .
$ docker run -p 5000:22 -p 5920:5920 --name scivis_sheet_3 -it
   scivis-2020:sheet-3
```

While the first line loads the ParaView container (you might have to adapt the path to it), the second one prepares all data needed for this assignment sheet. Please note the dot at the end of the second line. It indicates the folder containing the provided *Dockerfile*. The *Dockerfile* contains the instructions to download the source code skeleton of the assignment sheet and set up the build. The last line starts the container. From that point onwards, your terminal will behave like a Linux terminal. Your current location in the container should be the ParaView binary folder. To start ParaView just type

```
$ ./paraview
```

ParaView should now be running but you cannot see any GUI. At any time you can leave the docker console by typing `exit`. You can re-enter the container by typing

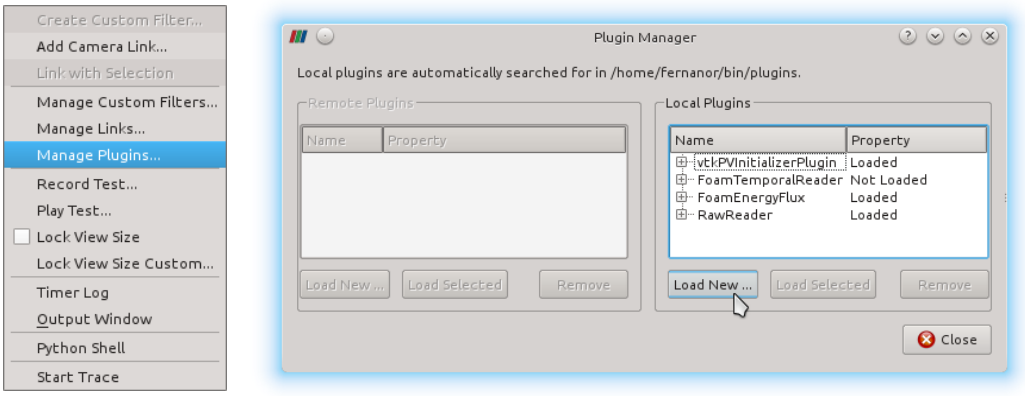
```
$ docker start -ia scivis_sheet_3
```

Further documentation can be found inside the *Dockerfile*. After starting ParaView, you will not be able to see the GUI, as docker is a purely console-based virtualization service. Here, the VNC client comes into play. It opens a connection to the container and enables you to see the visual output. Please use it to connect to `localhost:5920` while ParaView is running in the Docker container. The user name to enter will be `root`, the password `docker`. After doing that, you should see the ParaView window and can start to use it.

Step 2 - Edit and compile the plugin:
When the Docker container is set up, the plugin you will have to adapt is already configured and compiled once. To recompile it, navigate into the folder `~/scivis-2020/build/` and run `make install`. The file you will have to adapt is located in `~/scivis-2020/source/plugins/assignment_03/modules/exercise_02/exercise_02.cxx` The file can be edited by using any terminal-based text editor like nano or vim.

To use the plugin in ParaView, open the `Tools` menu and select `Manage Plugins` (see Figure 1).
Click the button `Load New` and navigate to the *plugin* build folder.
Select the shared object (`.so`) file. Make sure the `Property` field says `Loaded` and close the plugin manager.
In the `Filters/SciVis` menu of the main window there should now be an option `PointSplatter`.
If this is the case, you are ready to fill in the missing TODOs of the plugin.

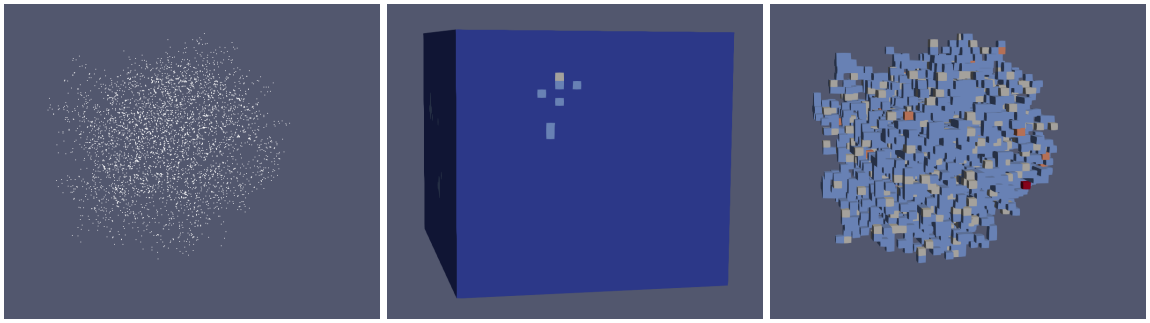
Step 3 - Particle Splatting:
The goal of this task is to finish the provided Particle Splatter plugin for ParaView. It converts the input particle data given in the file 1m40.csv into a rectilinear grid by counting the particles residing in each cell. To make the splatting easier for you, the splatter should



(a) Tools menu

(b) Plugin Manager

Figure 1: Steps in Paraview to load a custom plugin.



(a) After step 2 (b) After step 3 (c) After step 4

Figure 2: Visualization results after applying the filtering steps.

not use any kernel to increase the influence radius of the particles. The process to visualize everything works as follows:

1. Open ParaView and load the 1m40.csv file using the Open File dialog. The file is located in `~/scivis-2020/source/data`. Then, hit "Apply" on the left side. A window containing the data of the file should open on the right.

2. Use the Filter "Table to Points" to convert the incoming table data into point data further processable by ParaView. The filter can be found under `Filters/Alphabetical/TabletoPoints`. Before hitting "Apply" you will have to change some settings so that the correct values will be transferred. The correct values are marked in figure 3. When no particles are visible in the 3D view, click once into the 3D view and then on the eye symbol besides the Table to Points filter in the Pipeline Browser. After these steps are done, the 3D visualization should look like depicted in Figure 2a.

3. Now, the custom filter you have to implement comes into play. Apply it on the now visible point data by selecting `Filters/SciVis/PointSplatter`. The correct settings are given in Figure 4. Please do not forget to hit the "Apply" button.

   If everything was correct, the resulting visualization should look like Figure 2b. The

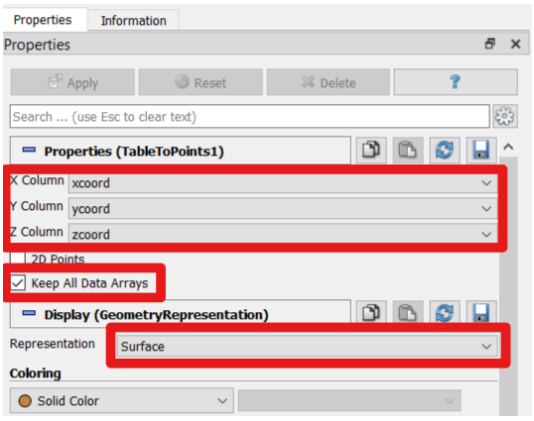


Figure 3: Correct settings for the Table to Points Filter

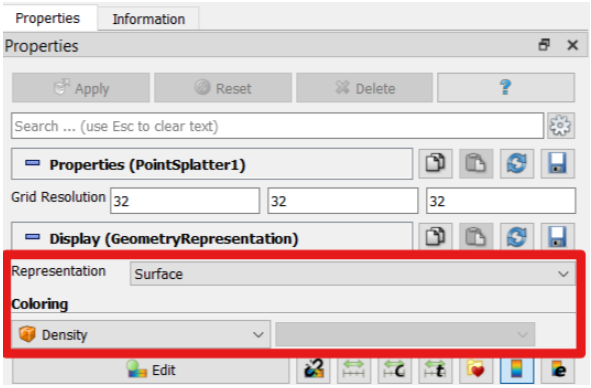


Figure 4: Correct settings for the PointSplatter filter.

shown result only appears if the code given in `exercise_02.cxx` has been properly extended by you. All areas where additions are necessary are marked by TODOs.

4. Now, only the outer cells are visible. Please apply the ”Threshold” filter that can be found in the filters menu on the data of the previous step. This filters out the empty cells leading to a block structure resembling the particles (cf. Figure 2c). Correct Settings for the filter are shown in Figure 5.

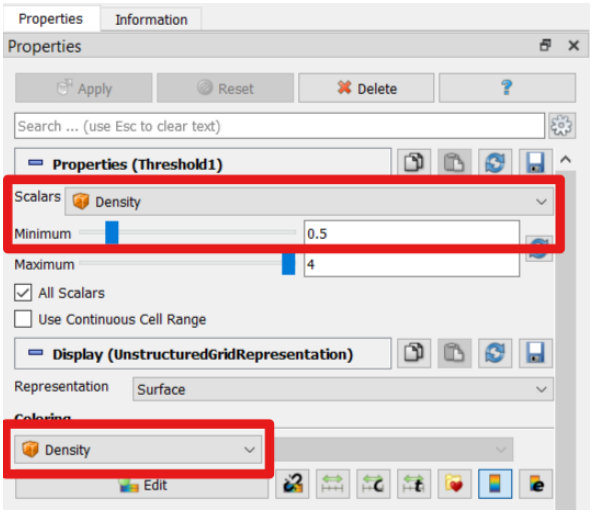


Figure 5: Correct settings for the Threshold filter

Hints: When no cells are visible after completing the last step, some parts of the implementation of the Point Splatter filter could be missing. Please fill them in. If the cells do not resemble the structure of the input data, something might be wrong with the cell index calculation.

Please hand in your edited `exercise_02.cxx` file. If it does not produce correct results as shown in Figure 2c, you can also hand in screenshots to prove the completion of the previous steps. To extract the file from the Docker container, please leave the container by typing `exit`. Then, you can execute

```
$ docker cp scivis_sheet_3:/root/scivis-2020/source/plugins/
    assignment_03/modules/exercise_02/exercise_02.cxx ./
```

This copies the file into your current folder.

Submission Deadline: 15.05.2020, 23:55

please hand in your submission through the ILIAS system.