Quick Guide to Tensor Voting using GPU (CUDA)

Ming LIU
liu.ming.prc@gmail.com

March 22, 2013

This document is a **very brief** introduction to the usage of the GPU library for 3D tensor voting. So don't hesitate to ask me for further details and help if you need.

1 Pre-requirement for installation

1.1 CUDA and NVIDIA

I think the main issue for most users maybe the installation of CUDA. Other than that, everything should be OK. If you have problem in compilation the .cu files, even with cuda ¿3.0 installed, it means that your hardware may not compatible. Update your NVIDIA driver to the latest always helps.

Hint:

- `$ nvcc --version`
  to check the CUDA SDK version.
- it is a pity that the `gpgpu` ros package is not updating anymore. In general, the official SDK from NVIDIA is enough.

  `https://developer.nvidia.com/cuda-downloads`

1.2 Other dependent packages

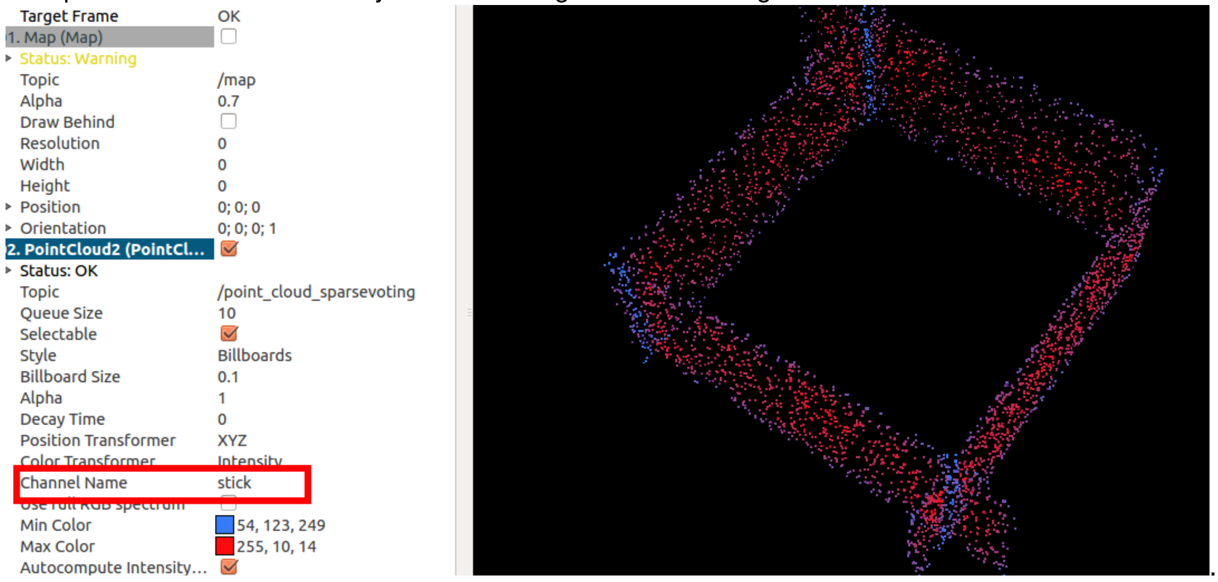
The code is based on ETH-ASL libpointmatcher and libpointmatcher-ros library:

If you checked out the latest nifti-mapping ros-stack, it should be ok. Otherwise, get it from GitHub: `https://github.com/ethz-asl`

2 Quick Tutorial after compilation

1. "rosmake" in tensorvoting package
   Usually it should work if the CUDA SDK is configured. Error may occur if the libpointmatcher is not properly installed.
2. You will get two executables in bin/ : `demo_dense_gpu` and `demo_sparse_gpu`. They are corresponding to the concept of dense voting and sparse voting respectively [2, 3]. To try these two samples, the launch files in launch/ can be used.
3. *sparse voting*:
   use `demo_sparse_gpu.launch`. It will load (by default) the dataset data/planes4.vtk then extract sparse tensor from it. (.vtk files are supported by paraview. please refer to the next section) It will keep doing that in loop, unless a ctrl-c kill to the processes. An rviz will be opened and showing

the sparse information. You may see something like the following:



Target Frame	OK
1. Map (Map)	☐
▸ Status: Warning	
Topic	/map
Alpha	0.7
Draw Behind	☐
Resolution	0
Width	0
Height	0
▸ Position	0; 0; 0
▸ Orientation	0; 0; 0; 1
2. PointCloud2 (PointCl…	✓
▸ Status: OK	
Topic	/point_cloud_sparsevoting
Queue Size	10
Selectable	✓
Style	Billboards
Billboard Size	0.1
Alpha	1
Decay Time	0
Position Transformer	XYZ
Color Transformer	Intensity
Channel Name	stick
Use rainbow/Full RGB spectrum	☐
Min Color	🟦 54, 123, 249
Max Color	🟥 255, 10, 14
Autocompute Intensity…	✓

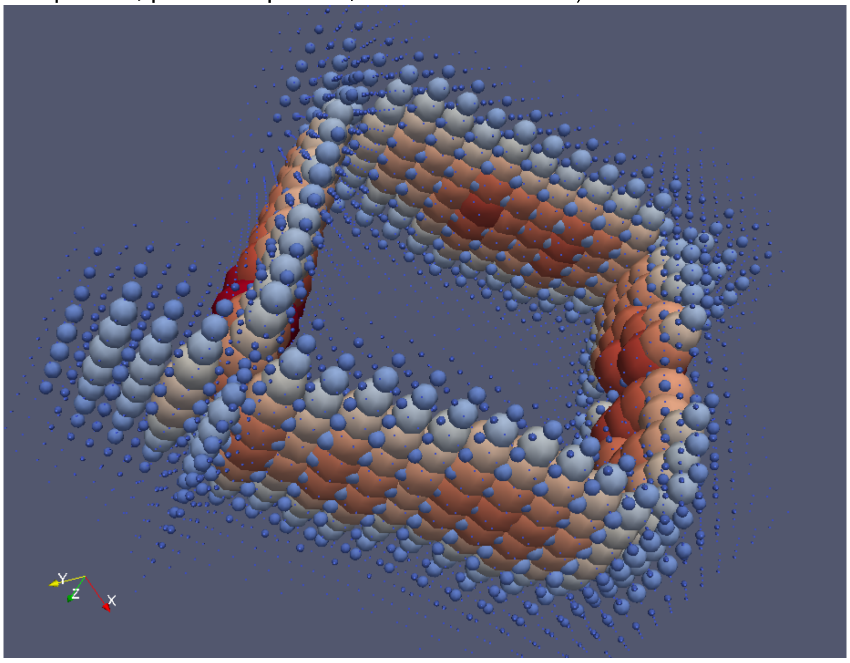
Note that, you can change the channel on the left (highlighted by red rectangle) to switch the representation (stick saliency, plate, ball, x, y, z etc.) In this sample, you see the red-ish color show the magnitude to be considered as planes (stick saliency).

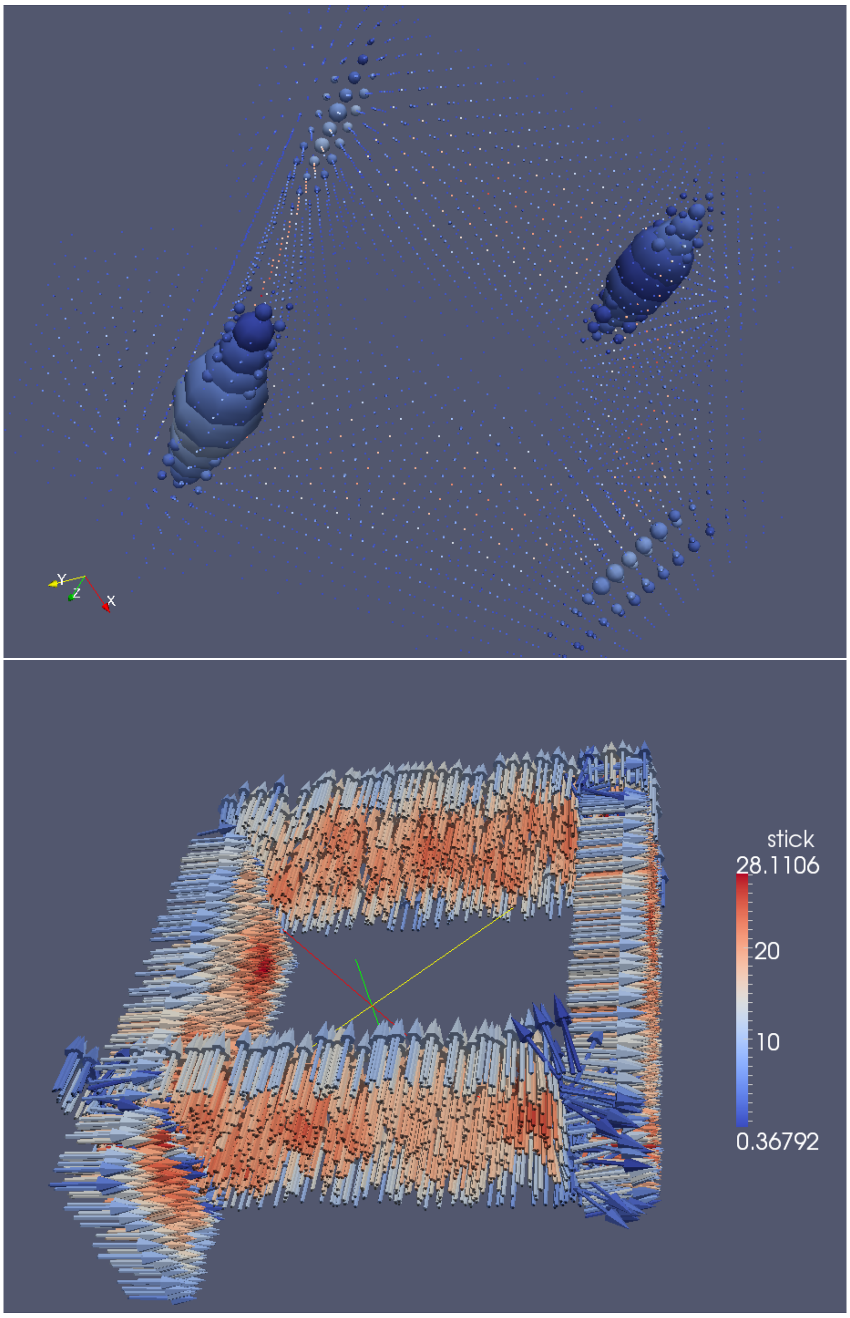
4. *dense voting*:
   similar thing. However, it is not recommended to use rviz to show the result, as explained in the next section.

5. The parameters can be modified from the launch files to tune different datasets. e.g. `sigma` for the kernel size.

3 vtk files and dense voting result inspection

We mainly use paraview (`apt-get install paraview`) to inspect the result of point-clouds. This GPU implementation allows to save .vtk files for further review. It can be turned on by setting the parameter in .launch `savevtk`. The following examples show how they may look like in paraview (stick component, plate component, normal estimation):





For further information, please refer to: [1]

BTW: if you need an omnidirectional camera image dataset with vicon ground truth (used for visual homing, visual odometry etc), please visit my website: http://www.asl.ethz.ch/people/lium/personal

References

[1] Ming Liu, Francois Pomerleau, Francis Colas, and Roland Siegwart. Normal Estimation for Point-cloud using GPU based Sparse Tensor Voting. In *IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, 2012.

[2] P. Mordohai and G. Medioni. Tensor voting: a perceptual organization approach to computer vision and machine learning. *Synthesis Lectures on Image, Video, and Multimedia Processing*, 2(1):1–136, 2006.

[3] H.F. Schuster. Segmentation of lidar data using the tensor voting framework. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 35:1073–1078, 2004.