Name : PALLAPOTHU VAISHNAV

ID : 20171cse0480

Sec : 6CSE07

GitHub Link : https://github.com/20171CSE0480/ML-assigment.git

Project Report



[3D MNIST](https://www.kaggle.com/daavoo/3d-mnist)

A 3D version of the MNIST database of handwritten digits

Last Updated: 3 years ago (Version 13 of 3)

About this Dataset

**Context**

The aim of this dataset is to provide a simple way to get started with 3D computer vision problems such as 3D shape recognition.

Accurate [3D point clouds](https://en.wikipedia.org/wiki/Point_cloud) can (easily and cheaply) be adquired nowdays from different sources:

* RGB-D devices: [Google Tango](http://get.google.com/tango/), [Microsoft Kinect](https://developer.microsoft.com/en-us/windows/kinect), etc.
* [Lidar](https://en.wikipedia.org/wiki/Lidar).
* [3D reconstruction from multiple images](https://en.wikipedia.org/wiki/3D_reconstruction_from_multiple_images).

However there is a lack of large 3D datasets (you can find a [good one here](http://shapenet.cs.stanford.edu/) based on triangular meshes); it's especially hard to find datasets based on point clouds (wich is the raw output from every 3D sensing device).

This dataset contains 3D point clouds generated from the original images of the MNIST dataset to bring a familiar introduction to 3D to people used to work with 2D datasets (images).

In the [3D\_from\_2D notebook](http://nbviewer.jupyter.org/github/daavoo/3DMNIST/blob/master/3D_from_2D.ipynb) you can find the code used to generate the dataset.

You can use the code in the notebook to generate a bigger 3D dataset from the original.

**Content**

**full\_dataset\_vectors.h5**

The entire dataset stored as 4096-D vectors obtained from the voxelization (x:16, y:16, z:16) of all the 3D point clouds.

In adition to the original point clouds, it contains randomly rotated copies with noise.

The full dataset is splitted into arrays:

* X\_train (10000, 4096)
* y\_train (10000)
* X\_test(2000, 4096)
* y\_test (2000)

Example python code reading the full dataset:

with h5py.File("../input/train\_point\_clouds.h5", "r") as hf:

X\_train = hf["X\_train"][:]

y\_train = hf["y\_train"][:]

X\_test = hf["X\_test"][:]

y\_test = hf["y\_test"][:]

**train\_point\_clouds.h5 & test\_point\_clouds.h5**

5000 (train), and 1000 (test) [3D point clouds](https://en.wikipedia.org/wiki/Point_cloud) stored in [HDF5 file format](https://support.hdfgroup.org/HDF5/whatishdf5.html). The point clouds have zero mean and a maximum dimension range of 1.

Each file is divided into [HDF5 groups](https://support.hdfgroup.org/HDF5/Tutor/fileorg.html)

Each group is named as its corresponding array index in the original mnist dataset and it contains:

* "points" dataset: x, y, z coordinates of each 3D point in the point cloud.
* "normals" dataset: nx, ny, nz components of the unit normal associate to each point.
* "img" dataset: the original mnist image.
* "label" attribute: the original mnist label.

Example python code reading 2 digits and storing some of the group content in tuples:

with h5py.File("../input/train\_point\_clouds.h5", "r") as hf:

a = hf["0"]

b = hf["1"]

digit\_a = (a["img"][:], a["points"][:], a.attrs["label"])

digit\_b = (b["img"][:], b["points"][:], b.attrs["label"])

**voxelgrid.py**

Simple Python class that generates a grid of voxels from the 3D point cloud. Check kernel for use.

**plot3D.py**

Module with functions to plot point clouds and voxelgrid inside jupyter notebook. You have to run this locally due to Kaggle's notebook lack of support to rendering Iframes. [See github issue here](https://github.com/Kaggle/docker-python/issues/36)

Functions included:

* array\_to\_color Converts 1D array to rgb values use as kwarg color in plot\_points()
* plot\_points(xyz, colors=None, size=0.1, axis=False)
* plot\_voxelgrid(v\_grid, cmap="Oranges", axis=False)

**Acknowledgements**

* Website of the [original MNIST dataset](http://yann.lecun.com/exdb/mnist/)
* Website of the [3D MNIST dataset](https://github.com/daavoo/3DMNIST)