ISSN NO: 2770-0003
Date of Publication: 28-02-2022

Representation Forms Of 3d Data- Objects

Beknazarova Saida Safibullayevna[1], Anvarxo'jaev Sarvarxo'ja A'zamxo'ja o'g'li[2]
[1] Professor of Tashkent University of Information Technologies named after Muhammad Khwarizmi, Tashkent, Uzbekistan, saida.beknazarova@gmail.com
[1] master of Tashkent University of Information Technologies named after Muhammad Khwarizmi, Tashkent, Uzbekistan

Annotation: machine learning methods are increasingly resorted to to solve this issue. If we imagine the work of a machine learning algorithm in the form of a "black box", to which data of a given type is fed to the input and the algorithm outputs a prediction in the form of data of a given type, then if data encoding three-dimensional structures are presented at the input and/or output, we talk about the field of machine learning, which is called 3D ML (three dimensional data machine learning problems) or, the term Geometric deep learning is often found when it comes to the use of deep architectures.

Today, machine learning methods are increasingly resorted to to solve this issue. If we imagine the work of a machine learning algorithm in the form of a "black box", to which data of a given type is fed to the input and the algorithm outputs a prediction in the form of data of a given type, then if data encoding three-dimensional structures are presented at the input and/or output, we talk about the field of machine learning, which is called 3D ML (three dimensional data machine learning problems) or, the term Geometric deep learning is often found when it comes to the use of deep architectures.

For anyone who would like to dive deeper into the subject of 3D ML, we recommend that you familiarize yourself with the materials of the site geometricdeeplearning.com . It will be especially useful to read a review article on the entire field "M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, P. Vandergheynst, Geometric deep learning: going beyond Euclidean data, IEEE Signal Processing Magazine 2017".

The types of tasks that need to be solved in the field of 3D ML can be very different:

–classical problems (classification and clustering of 3D data),
–generation of 3D content (generating models)
–restoring a 3D model of an object from one or more RGB images of this object (2D-to-3D),
–static mesh animation and many others.

Fig.1 3D ML IDF0 scheme.

Recently, deep learning methods have become very popular in the field of 3D ML. This can be explained by the fact that most classical machine learning algorithms require analytical functions to manipulate data. For example, in order to build metric classification/clustering algorithms, you need to enter a metric in the space of 3D models. Such metrics can be constructed using three-dimensional Zernike descriptors [2] - spectral decomposition of two-dimensional manifolds, analogous to the Fourier transform.

So, for example, the authors in the work "Characterizing Structural Relationships in Scenes Using Graph Kernels" [3] use three-dimensional Zernike descriptors to build a metric core for clustering complex three-dimensional scenes, and in "Fast human pose estimation using 3D Zernike descriptors" [4] the authors use descriptors to build a fast and stable model for estimating human posture in space. But for most 3D ML tasks, picking up similar functions turns out to be a big test. In this case, the use of neural networks is a more understandable and simple alternative.

ISSN NO: 2770-0003
Date of Publication: 28-02-2022

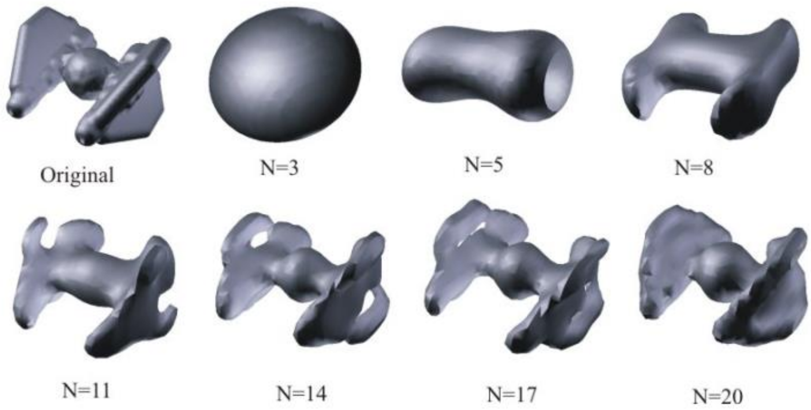

Fig.2 On the top left is the original voxelized 3D model of the spacecraft. The remaining images are a mesh reconstructed from isosurfaces described by Zernike descriptors. The number N denotes the number of different Zernike moments used for recovery.

To date, there are already many different deep learning architectures for solving 3D ML problems, but all of them can be divided into several types, depending on the type of task and data type. In this note, we will consider only one 3DML task, namely the single image 2D-to-3D task.

But before you deal with the problem statement, you need to describe the existing most common ways of representing 3D data and their features.

Data is one of the most important components of machine learning. The more of them, the better we can train our model, the more generalizing ability it will have. And indeed, below we will talk about machine learning algorithms and how they work with data, but first of all this data needs to be extracted from the outside world and prepared for processing. In real-world tasks, data preprocessing and optimization of the data processing pipeline takes most of the time.

An important detail when directly choosing a machine learning model that will solve the task is the choice of the form of data representation. For the same data, their representations may differ greatly, which will affect the choice of the processing model. So, for example, we can represent plain text as a linear sequence of characters, and in this case recurrent architectures are better suited for data processing, and if the text is presented as an image, then convolutional architectures are better suited for data processing.

Which form of data should be used when solving a specific task is determined by what data is already available at the moment, how the data is extracted from the outside world and which aspect of the data is most prioritized in a particular task.

If we talk about the field of image processing, then there is a canonical way to represent information in the form of raster RGB images, i.e., in fact, we store three numeric matrices. Sometimes several more channels may be added to the three main channels, for example, satellite images may also have an infrared channel, an ultraviolet channel, etc. This way of presenting data is both computationally efficient, because there are many ready—made fast processing algorithms for working with matrices and their multidimensional tensor analogues, which also shift well to computing on video cards.

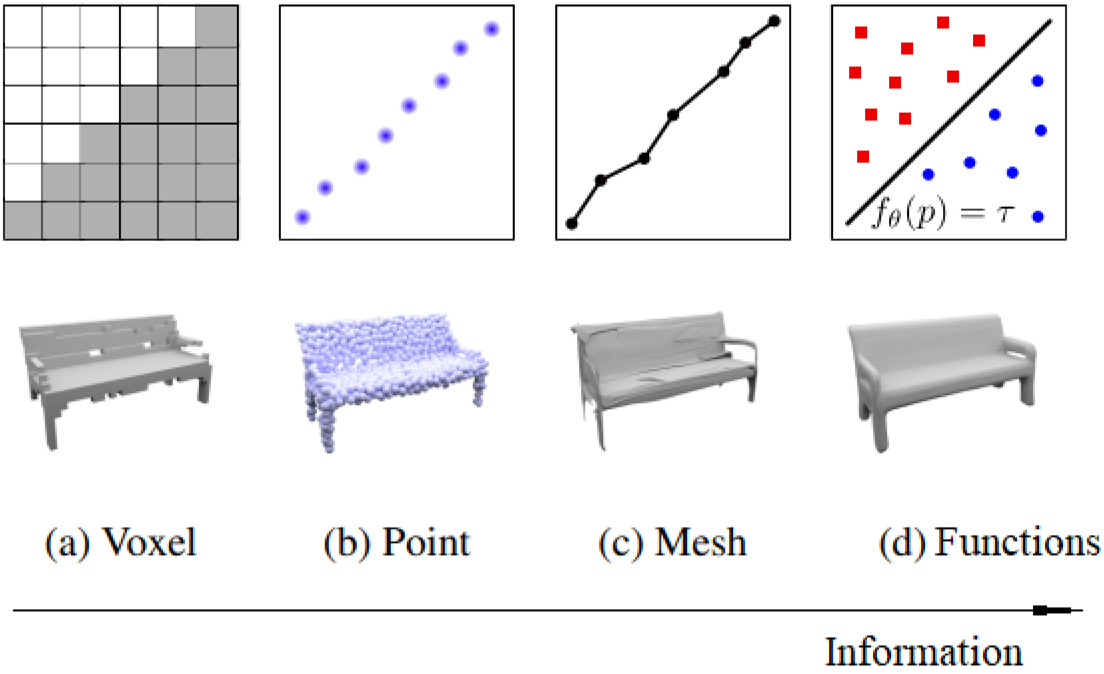

Fig.3 Comparison of 3D data representation forms: the main 4 types in ascending order of spatial informativeness, their 2D counterparts.

For the 3DML domain, there is currently no single form of data representation that is both compact, computationally efficient and easily extracted from real data.

Next, we will consider examples of working with data in Python, using the most popular frameworks for working with 3D:

- trimesh
- pytorch3d

Code examples and test data can be found in the repository on GitHub.

To begin with, we will download all the libraries necessary for operation and, if possible, install a compatible video card as a device for processing pytorch CUDA tensors.

Most often, for example, as in [5], there are four main ways of representing three-dimensional data.

Reference:

1. Russell S. Artificial Intelligence: a Modern Approach, S. Russell, P. Norvig / Williams, Moscow, 2018, 1408 p.
2. Novotni, Marcin and Reinhard Klein. “Shape retrieval using 3D Zernike descriptors.” Computer-Aided Design 36 (2004): 1047-1062. [project page]
3. Fisher, Matthew & Savva, Manolis & Hanrahan, Pat. (2011). Characterizing Structural Relationships in Scenes Using Graph Kernels. ACM Trans. Graph… 30. 34. 10.1145/2010324.1964929.
4. Berjón, Daniel & Morán, Francisco. (2012). Fast human pose estimation using 3D Zernike descriptors. Proceedings of SPIE — The International Society for Optical Engineering. 8290. 19-. 10.1117/12.908963.
5. Mescheder, Lars & Oechsle, Michael & Niemeyer, Michael & Nowozin, Sebastian & Geiger, Andreas. (2018). Occupancy Networks: Learning 3D Reconstruction in Function Space. [code]