



UNIVERSITY OF PENNSYLVANIA

---

# **Classification of Digitally Reconstructed Neurons with Machine Learning Techniques**

---

*Author:*

Xiaochuan QIN  
xqin@seas.upenn.edu

*Sponsor:*

Dr. Shing Chun Benny LAM  
shinglam@mail.med.upenn.edu

March 14, 2011

# 1 Specific Aims

This project is to propose a machine learning approach which aims at classifying digitally reconstructed neurons. Our algorithm<sup>1</sup> will be design and implemented in two steps: 1) to develop an efficient feature descriptor which is able to measure similarities between 3D neuron models, and 2) Use the descriptor we developed to implement a classifier to sort query neuron models into existing neuron categories, or into the unknown group if no reliable decisions can be made.

Experiments will be conducted to verify the feasibility and to test the performance of our approach. In order to shorten the development cycles, a relatively small set of neuron models will be used. A well tuned system is expected to perform effectively on the groups of neurons which have sufficient data. One can also expect a gradual improvement in classification accuracy during the parameter learning process. It is more important than statistical results that through analyzing the learning process, we can better understand the spatial structures of the neurons and then design more efficient feature descriptors and classifiers.

## 2 Research Strategy

### 2.1 Significance

Databases of digital reconstructions of neuronal morphology have been built and widely used for years in mammal brain research area, such as anatomical investigation and biophysics modeling [2]. However, while the expanding volume of these digitalized 3D neuron models generates unlimited analytical power for neuronal science, the matching of newly discovered neurons with existing ones, and the identifying of new structural types becomes an increasingly difficult task. Matching and classifying algorithms have not been well studied nor applied in this field. We will address this issue by not only proposing solutions, but also constructing an organized and extensible framework for future research.

### 2.2 Innovation

We generalized the concept of the Shape Context Descriptor, which is for quantifying local features in 2D images, into a 3D space. By centering this 3D shape context descriptor at the neuron nucleus, this can be used as a whole cell shape descriptor. More innovatively, we figure out a method to determine the zenith direction of a cell's coordinate system by finding an eigenvector for a 3-by-3 matrix. This significantly saves computational effort of comparisons and thus makes the 3D shape context descriptor practicle.

### 2.3 Approach

#### 2.3.1 Preliminary Studies

Ohbuchi and Takei [7] summarized four major steps in querying a 3D model from a shape-based model database: 1) Query formation, 2) Feature extraction, 3) Dissimilarity computation, and 4) Retrieval. We focus on step 2) and 3), which is the critical part of our algorithm, in this literature review.

Ankerst et al. [1] proposed one of the earliest 3D shape similarity matching algorithms. Their objective is to compare and retrieve 3D molecular models as collections of atoms. Although their algorithm can not be applied to 3D models based on polygons that are commonly used, it is well applicable to our problem. In their algorithm, each atom is modeled as a point labeled by its mass or an "is-a-hydrogen" tag, which is very similar to our cases. Their shape descriptor is in the form of layered concentric shells in a parameterized 3D space. By Subdividing the space and comparing mass weight across shells makes their descriptor rotation invariant. Funkhouser et al [9] extended Ankerst et al's method. A polygon based 3D model is converted into a voxel image, and then the

---

<sup>1</sup>NeuroMorpho.Org hosts the largest public accessible database of digitally reconstructed neurons [2][10]. The latest release of their inventory contains 5793 digital reconstructions of mammal neuron cells contributed by more than 40 laboratories worldwide. The collections of this neuron model library are still increasing. Besides popularity, their data quality - accuracy and organization - inspires us to design algorithms and conduct experiments aiming at NeuroMorpho's database.

distributions of the voxels are analyzed in Ankerst et al's concentric shells. Though their algorithm is aimed at polygonal models, their idea of voxelizing the image fits in our neuron models.

Ankerst et al and Funkhouser et al's methods inspire us to build a shape context descriptor. Belongie et al [3] proposed the shape context descriptor and then made it rotation invariant. They sample on edges in a 2D image, and then center a circle at each sampled point. The circle is divided into several regions according to angle and radius ranges. Each region is assigned a bin in a histogram, in which bar heights are the numbers of sampled points falling in the corresponding regions. By aligning the starting angle along the gradient of the pixel intensity at the circle center, they make the descriptor rotation invariant [4].

In 3D shape comparison, not only the mass of sample points are considered, distances between angles and directions can also be used in building histograms. For example, Ohbuchi et al [6] computed Angle Distance and Absolute Angle Distance. In Ohbuchi and Takei [7], more sophisticated distance measurer called Alpha Multiresolution AAD is computed. These directional distance functions are defined based on the normal directions of polygons in 3D models. Although this is not directly applicable to our model, it is worth of pointing out that directional features may improve our descriptors.

### 2.3.2 Data Abstraction

A typical NeuroMorpho model contains a list of points sampled from a neuron cell. The policy of sampling varies across models. Each sample point is described by a row of data with seven information fields: ID, type, three coordinates (x, y, z) in a Cartesian coordinate system, diameter (d), and the ID of its parent sample point. The coordinates can also be viewed as voxel indices in a typical medical image. The first row is always for the nucleus, which can be positioned at the origin after normalization. The type field indicates which part a sample belongs to (e.g. soma, axon and dendrite etc.). However, many samples are unclassified, and thus no reliable type information can be used in comparison and classification. Since each point has a unique parent (except the nucleus, which has none), a neuron model can be viewed as a tree rooted at the origin. It is convenient to use terminologies in graph theory to presenting our ideas - Nodes are sample points, and edges connect any two adjacent nodes, in which the one closer to the nucleus is the parent node. Weights can be assigned to nodes, and length to edges. To describe the direction of an edge, we also associated it with an "edge vector" which points from a parent node to its descendant .

### 2.3.3 Feature Description

Our method is to generate a 3D shape context descriptor whose center is at the root of a neuron tree. A sphere surrounding its center will be divided into a certain number of regions according to the inclination angle, azimuth angle, and distance from the center. Weighted sample points will be credited to each region (bin) in a histogram graph. Differences between each corresponding pair of bins will be weighted and accumulated. The summation measures the dissimilarity between two neuron models. In order to make two neuron model's shape context descriptors comparable, two issues must be addressed: 1) how does one fairly distribute and weigh sample points, and 2) how does one make the descriptors invariant to rotations, i.e., fix the azimuth axis and zenith axis of the spherical coordinate system of the descriptors.

The first problem is relatively easy. One of the features of NeuroMorpho's dataset is that all the numerical values are measured in a universal unit. Re-sampling the tree along all edges and interpolating diameters of newly sampled points guarantee the even distributions. In terms of weights, various forms of weighing functions can be defined on diameters to emphasize different features.

The second issue has no solutions in literature suitable to our cases. We propose to solve this problem in three steps. 1) Find a unit vector ( $z$ ) which minimizes the total angles deviated from it by all edges vectors (before re-sampling) on the tree. This problem can be formalized as the problem of minimizing  $f(z) = z' A' A z$ , subject to  $z' z = 1$ . Where  $A$  is a matrix contains all the edge (row) vectors. If let  $M = A' A$ , the problem is actually to find a normalized eigenvector for  $M$ . 2) Choose the direction along or opposite to that of  $z$  as the zenith direction. One can compute the summed weights of the re-sampled points in the two hemisphere cut by the equator plane, and the hemisphere with lower total weight will be defined as the upper half. 3) It is more difficult to define an azimuth axis analytically. Instead of, we will select one direction at the comparison step by brute force computation. That is, use relatively small ranges to divide the azimuth angle, and use region boarder blurring make the descriptor

less sensitive to the starting position of the azimuth axis. Then when two models are compared, we arbitrarily fix the azimuth axis for one model, and rotate the other to find the smallest dissimilarity.

### **2.3.4 Classification**

A  $k$  Nearest Neighbor ( $k$ -NN) search algorithm will be used in classification. Neurons have already been classified will be used as the training set. Given a query neuron whose type is unknown, we search the training data to find the top  $k$  neurons with the lowest dissimilarities. These  $k$  neurons vote the type of the query neuron. Votes will be weighted by similarity for better results. If no class wins votes above a certain criterion, "unknown" should be labeled. After voting, successfully labeled neuron models will join the training set in the next query. In the search process, we will simply do a linear search. We chose this naive approach for two reasons. First, at current stage we have limited understandings to the spatial structure of the neuron models. Second, for higher dimensional spaces, a linear search algorithm often outperforms space partitioning approaches [8].

### **2.3.5 Experiments**

To test our approach, we begin with a relatively small data set (totally about 250 models). Firstly, construct a set of pyramidal, a set of granule cells, and a small set of other types of hippocampus cells. Randomly select a small portion of neuron models from these three sets to form a testing set; The remaining neurons will form our training set. Secondly, using the training set, cross validation simulations will be run over all parameter options to find the configuration minimizing the mis-prediction rate. Finally, feed neuron models in the test set to the classifier with the optimal configuration just identified, collect results and report statistics.

## **2.4 Timeline**

### **2.4.1 File I/O and Reformat**

Suitable file format Design; Reformatting procedure Implementation. 5 hours.(5 total)

### **2.4.2 Data Preprocessing**

Remove duplicated nodes; Resampling; Store preprocessed data. 5 hours. (10 total)

### **2.4.3 Descriptor Implementation**

Development and debug. 10 hours. (20 total)

### **2.4.4 Arbitrary Rotation of Descriptor**

Figure out efficient memory organization; Function interface design for machine learning approach; Functionality implementation. 10 hours. (30 total)

### **2.4.5 Measuring Data Generation**

Histogram generation; Store to hard drives. 5 hours. (35 total)

### **2.4.6 k-NN Search Algorithm Implementation**

Framework design; Development and Debug. 15 hours (50 total)

### **2.4.7 Parameter Learning**

Initial parameter selection; Search step tuning; Optimal parameter searching. 20 hours (70 total)

### 2.4.8 Final Experiment

Final experiment on test data set; Statistical results collection. 5 hours (75 total)

### 2.4.9 Documentation and Report

Final report writings. 15 hours (90 total)

## References

- [1] M. Ankerst, G. Kastenmuller, H-P Krigel, T. Seidl, *3D Shape Histogram for Similarity Search and Classification in Spatial Databases*, Proceedings of International Symposium Spatial Databases, Hong Kong, China, July 1999.
- [2] Giorgio A. Ascoli, Duncan E. Donohue, and Maryam Halavi, *NeuroMorpho.Org: A Central Resource for Neuronal Morphologies*, The journal of Neuroscience, August, 2007.
- [3] S. Belongie and J. Malik, *Matching with Shape Contexts*, IEEE Workshop on Content Based Access of Image and Video Libraries, 2000.
- [4] S. Belongie, J. Malik, and J. Puzicha, *Shape Matching and Object Recognition Using Shape Contexts*, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (24): 509 - 521, April 2002.
- [5] Ameesh makadia and Kostas Daniilidis, *Spherical Correlation of Visual Representations for 3D Model Retrieval*, International Journal of Computer Vision 89 (2-3): 193 - 210.
- [6] R. Ohbuchi, T. Minamitani, and T. Tsuyoshi Takei, *Shape-Similarity Search of 3D Models by using Enhanced Shape Functions*, Proceedings, Theory and Practice of Computer Graphics, pp. 97 - 104, Birmingham, UK, 2003.
- [7] Ryutarou Ohbuchi and Tsuyoshi Takei, *Shape-Similarity Comparison of 3D Models using Alpha Shapes*, IEEE Proceedings of The 11th Pacific Conference on Computer Graphics and Applications, 2003.
- [8] Roger Weber, Hans-J. Schek and Stephen Blott, *A Quantitative Analysis and Performance Study for Similarity-Search methods in High-Dimensional Spaces*, Proceedings of the 24th VLDB Conference, New York, 1998.
- [9] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, D. Jacobs, *A search Engine for 3D Models*, ACM TOG, 22(1), pp. 83-105, January, 2003.
- [10] Homepage of NeuroMorpho.Org, <http://neuromorpho.org/neuroMorpho/index.jsp>.