Pattern Recognition 63 (2017) 680–688

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Indexing and mining large-scale neuron databases using maximum inner product search

Zhongyu Li[a], Ruogu Fang[b], Fumin Shen[c], Amin Katouzian[d], Shaoting Zhang[a,*]

[a] Department of Computer Science, University of North Carolina at Charlotte, USA
[b] School of Computing and Information Sciences, Florida International University, USA
[c] University of Electronic Science and Technology of China, China
[d] IBM, Almaden Research Center, San Jose, CA, USA

ARTICLE INFO

Keywords:
Neuron morphology
Large-scale retrieval
Binary coding

ABSTRACT

Morphological retrieval is an effective approach to explore large-scale neuronal databases, as the morphology is correlated with neuronal types, regions, functions, etc. In this paper, we focus on the neuron identification and analysis via morphological retrieval. In our proposed framework, multiple features are extracted to represent 3D neuron data. Because each feature reflects different levels of similarity between neurons, we group features into different hierarchies to compute the similarity matrix. Then, compact binary codes are generated from hierarchical features for efficient similarity search. Since neuronal cells usually have tree-topology structure, it is hard to distinguish different types of neurons simply via traditional binary coding or hashing methods based on Euclidean distance metric and/or linear hyperplanes. Therefore, we employ an asymmetric binary coding strategy based on the maximum inner product search (MIPS), which not only makes it easier to learn the binary coding functions, but also preserves the non-linear characteristics of the neuron morphological data. We evaluate the proposed method on more than 17,000 neurons, by validating the retrieved neurons with associated cell types and brain regions. Experimental results show the superiority of our approach in neuron morphological retrieval compared with other state-of-the-art methods. Moreover, we demonstrate its potential use cases in the identification and analysis of neuron characteristics from large neuron databases.

1. Introduction

How the brain works is one of the most challenging issues in neuroscience. As neurons are the basic elements of the brain, understanding their properties and network connectivity is a key step to tackle this challenging problem. There are approximately 86 billion neurons in the human brain and no two neurons are exactly the same. Figuring out each neuron's properties is difficult. Generally, neurons tend to express distinct morphologies based on their cell types, brain regions and functions. Therefore, it is reasonable to explore the neuronal properties through their morphologies. Recent development in visualization and image processing techniques [12,22,26,31,32] enabled accurate segmentation, tracing and reconstruction of 3D neuronal models from microscopic images. Meanwhile, the fast growing 3D neuron image databases such as NeuroMorpho [1,28] provide a public platform to associate neuronal properties with morphologies. Therefore, morphology-based neuron retrieval becomes an effective way to assist neuroscientists to identify unknown neurons and discover the relationship between the neuronal morphology and the property.

Morphology-based neuron retrieval is made possible because of the recent rapid advancements in neuron tracing techniques [4,10,25,45]. Costa et al. [5] proposed the concept of neuromorphological space, which analyzed the tree-like shape and designed quantified measurements of neuron cell. Wan et al. [40] designed *BlastNeuron* for automated comparison, retrieval and clustering of 3D neuron morphologies. In the retrieval stage, *BlastNeuron* searches for similar neurons via the normalization of rank scores in terms of the similarity of feature vectors. Despite its high accuracy, this method could be inefficient when handling a large-scale neuron database. Therefore, Mesbah et al. [24] proposed a data-driven hashing scheme, i.e., hashing forest, to search among large neuron databases. By establishing multiple unsupervised random forests, 128 or more binary bits are generated to represent morphological features. Hashing forest algorithm has achieved efficient and accurate results in neuron retrieval. Nonetheless, it usually needs a large number of bits (e.g., larger than 128), so its efficiency can be further improved with shorter binary codes. More importantly, the encoding process relies on the embedding of the Euclidean distance, which may not be a suitable similarity

* Corresponding author at: UNC Charlotte, Computer Science, 28223 Charlotte, NC, United States.
E-mail address: szhang16@uncc.edu (S. Zhang).

http://dx.doi.org/10.1016/j.patcog.2016.09.041
Received 30 December 2015; Received in revised form 3 June 2016; Accepted 21 September 2016
Available online 03 October 2016

measure for neuron retrieval issue, as features of neuron data usually lay in complex feature spaces that may not be linearly separable. Therefore, advanced hashing algorithms are important to solve these challenges for efficient and precise retrieval.

As described in Ref. [24], binary coding and hashing techniques have achieved great success in efficient retrieval among large-scale databases, with many methods proposed in recent years, including, but not limited to, Spectral Hashing (SH) [42], Anchor Graph Hashing (AGH) [20], Iterative Quantization (ITQ) [9], and others [11,36,37,50]. However, they may not be directly applicable to the neuron retrieval problem, as the features of 3D neuron morphological data are dramatically different from 2D natural images, and different features usually reflect different level of neuron similarity. For example, the tree-like structure imposes a challenge to differentiate neuron types, since treating all features with equal importance may lead to inaccurate search results. In addition, although supervised binary coding and hashing methods have already been investigated in medical image analysis [19,48,49], it is preferred to employ unsupervised methods for neuron retrieval, since the annotations for neurons may be incomplete.

Although neuromorphology and binary coding are both well-studied in recent years, how to combine them for neuron retrieval remains a challenging problem. Specifically, there are three challenges in binary coding based neuron morphological retrieval:

1. The feature vectors of each neuron are much shorter 30–50 dimensions) compared with traditional 2D images' feature vectors 100–10000 dimensions). Therefore, only much shorter bits of binary codes can guarantee the retrieval efficiency. Employing much shorter binary codes to represent large-scale neuron data is a great challenge;
2. Despite the limited length of neuron feature vectors, each type of feature has their specific meaning, e.g., branch number reflects the connection of neuron cell. Bipolar neurons have two branches, while multipolar neurons have three or more branches connected with other neurons. Currently, most image retrieval methods are either addressing the single feature's binary coding or fusing multiple features in different retrieval stage [3,21,44,46,47]. However, in neuron retrieval problem, each single feature is too short to obtain reliable retrieval results, and fusing multiple features is usually time-consuming. Thus, the specific biological indication and the computational complexity in neuronal feature representation need to be considered.
3. As each neuromorphological feature is extracted based on the tree-like structure, this limitation of feature extraction may cause a tough question, in which the tree-like structure will lead to similar features extracted from different types of neurons, e.g., some unrelated neurons express similar feature vectors. How to differentiate them in non-linear space is a hard problem.

In this paper, we design a binary coding framework to effectively and efficiently analyze large neuron databases. This framework is based on the recent progress of the maximum inner product search, which was proposed for image retrieval [35]. Specifically, we employ the method in Ref. [35] as the baseline, and then adapt it to handle multiple features or feature hierarchies, which is necessary to achieve high precision in this neuron retrieval. We validate the efficacy of the proposed method in the neuron retrieval problem with a large-scale database, and it outperforms several other binary coding or hashing methods. In addition, according to the neuron information provided by NeuroMorpho [28], our proposed method can retrieve similar neurons in terms of the morphology, cell types and brain regions.

The remaining paper is organized as follows: Section 2 briefly reviews the work related to 3D neuron morphology and binary coding methods. Section 3 provides the details of the proposed MIPS based binary coding with feature hierarchy for neuron retrieval system, followed by experiment results and discuss its potential use case in Section 4. Finally, Section 5 concludes the paper and presents future work.

2. Related work

2.1. Neuron tracing for 3d neuron morphology

Neuron tracing aims to manually or automatically reconstruct 3D neuron morphology from fluorescence or electron microscopy images. Compared with 2D neuron image, 3D morphological data reflect spatial structure of the 3D neuron cell with more comprehensive information [51]. From the original microscopy images to the 3D neuron morphological data, neuron tracing consists a number of processing step, including image preprocessing (e.g., noise reduction, deconvolution, mosaicking), segmentation (e.g., soma, dendritic trees, spines, axons segmentation), reconstruction and connection [6,7,23,39,52]. In recent years, there are many tracing and reconstruction software released which make the 3D neuron morphological data easier to acquire. Fig. 1 illustrates a microscopy image from neuron slices [26] and its corresponding 3D morphological data through *Vaa3D* [32]. As shown in Fig. 1(b), morphological data provides more precise and quantitative measurements for neuron cells which facilitate features extraction for further retrieval and analysis..

Benefited from algorithms and software for neuron tracing, more and more 3D morphological databases are released in recent years. Unlike 2D medical images which can extract features with many well-studied algorithms, how to extract features from 3D neuron data is still an unsolved problem. For neuron cells, from axon to soma and then to dendrite, they usually express a tree-like structure. In Refs. [5,43], the authors introduced many quantitative measurements to analysis the tree-like structure of neuron cells.

Therefore, we can utilize these quantitative measurements as neuron morphological features. Specifically, we calculate three levels of measurements in order to reflect neuron morphology more comprehensive:

1. Global measurements, such as neuron's total height, depth, volume, etc. This level of features can express the holistic information of neuron cells;
2. Branch measurements, such as the Euclidean distance from compartments to somas, branch length, etc. This level of features denotes the information of neuron branches that are directly connected to the soma;
3. Bifurcation measurements, such as the angle between two terminal branches, etc. This level of features reflects the bifurcation's information of branches not directly connected to the soma.

In this paper, we calculate in total 38 measurements at the above three levels. Then we assemble them as morphological features to represent each neuron cell for further retrieval and analysis.

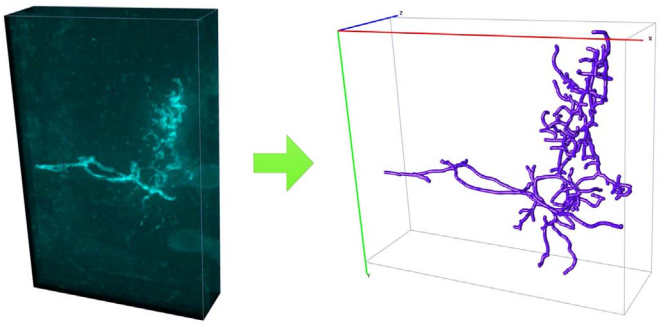


Fig. 1. From original microscopy neuron image to 3D morphological data: (a) original microscopy slices; (b) 3D neuron morphology with quantitative measures.

2.2. Image retrieval via binary coding

In recent years, binary coding and hashing have been widely used in machine learning, computer vision and related areas. By compressing long feature vectors into short binary codes, similarity search will be much more efficient in binary Hamming space compared with high dimensional feature space. The key question is how to obtain binary coding or hashing functions which can not only split feature vectors via binary codes but also keep similarities among original data. Recently, Wang etc. [41] presents a comprehensive survey with various types of hashing methods, including data-independent and data-dependent, supervised and unsupervised, linear and nonlinear, etc.

Data-independent methods usually design generalized binary coding or hashing functions to compact any given datasets. Locality-Sensitive Hashing (LSH) and its variants [8,15,33] are one of the most representative data-independent methods. This type of methods ensures the data similarity with long binary hash bits and multiple hash tables. However, these methods may not suit neuron retrieval problem because of the tree-like structure of neurons and their non-separability in Euclidean space. Another category is the data-dependent methods, whose their binary coding functions are obtained through learning from the given datasets. A large number of learning-based methods are proposed in recent years, including Iterative Quantization (ITQ) [9], Isotropic Hashing (IsoHash) [14], Minimal Loss Hashing (MLH) [29,30], FastHash [17], etc. Some of them are supervised methods which have already achieved high performance in large-scale retrieval. But current neuron database such as NeuroMorpho [28] lack enough normative annotations for every neuron. Therefore, unsupervised binary coding is a better choice for neuron retrieval. In addition, as mentioned before, the difference of some unrelated neurons can be subtle, which is hard to distinguish in linear space. Compared with linear binary coding/hashing algorithms, nonlinear algorithms usually generate more sensitive binary codes to divide data in nonlinear space. Representative methods such as Kernel-Based Supervised Hashing (KSH) [19], Spectral Hashing [42], Anchor Graph Hashing (AGH) [20], Inductive Manifold Hashing (IMH) [20], etc., they construct coding functions based on nonlinear kernel matrix or manifold structure. However, one disadvantage of the above mentioned nonlinear methods is that they fail to consider the diversity among different features when learning binary codes.

Different from the previous hashing methods, we will first fuse features into similarity matrix at different hierarchies. Then we will learn two asymmetric binary coding functions based on the maximum inner product search (MIPS) for query neuron and neuron database respectively. This unsupervised strategy considers the feature diversity and also split data into highly nonlinear space, which shows superior performance for neuron morphological retrieval task.

3. Methodology

In this section, we present the theoretical and technical details of our large-scale neuron morphological retrieval system, including the MIPS notation, feature hierarchy and asymmetric optimization.

3.1. Overview

In our framework, we compute morphological measurements (described in Section 2.1) as features to represent each neuron data. Although directly measuring the similarity between feature vectors offers an accurate solution, the computational efficiency is an issue, especially when searching in a large-scale database with tens of thousands of neurons. Therefore, we focus on learning coding functions to transform morphological features into binary codes. Fig. 2 shows the overview of our proposed framework. In the training phrase, after feature extraction, we group different features into several hierarchies to compute the similarity matrix. Then, we learn binary coding functions which can maximize inner product between two training data sets. Particularly, for optimization convenience, we jointly maximize two asymmetric coding functions $h(\cdot)$ and $z(\cdot)$ for the neuron database and the query neuron respectively. With these coding functions, in the query phase, the features of query neuron and all neurons in the database are compressed into short binary codes. Then, their inner product can be calculated and ranked in descending order. By selecting neurons in the database with top-K largest inner product, the characteristics of query neuron can be identified based on the retrieved neurons..

3.2. MIPS based binary coding with feature hierarchy

Background of MIPS: Let's denote the training neuron data set as $\mathbf{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_i, \ldots, \mathbf{a}_n\} \subset \mathbb{R}^{n \times d}$, which include n neurons, and each neuron has d dimension of features. From each neuron M types of morphological features are extracted, denoted as $\mathbf{a}_i = [\mathbf{a}_i^{(1)}, \ldots, \mathbf{a}_i^{(j)}, \ldots \mathbf{a}_i^{(M)}] \in \mathbb{R}^{1 \times d}$, where $d = \sum_{j=1}^{M} d_j$. Assuming the query neuron is $\mathbf{q} \in \mathbb{R}^{1 \times d}$, the MIPS problem can be defined as:

$$\mathbf{p} = \arg \max_{\mathbf{a}_i \in \mathbf{A}} \mathbf{a}_i \mathbf{q}^{\mathrm{T}} \tag{1}$$

which finds the largest inner product between $\mathbf{q}$ and each element in $\mathbf{A}$. As demonstrated in Ref. [19], the Hamming distance and the code inner product have a one-to-one correspondence. To accelerate com-

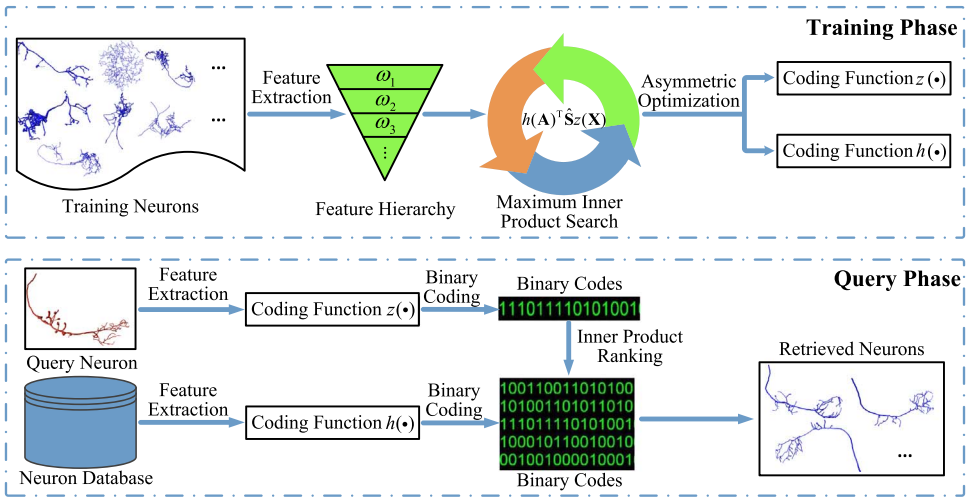


Fig. 2. Overview of the proposed neuron morphological retrieval framework.

putation and save storage, it is practical to employ binary coding method to tackle MIPS problem. A coding function h is learned to map the original feature vectors to bits of binary codes. Thus, problem (1) is reformulated as:

$$\mathbf{p} = \arg\max_{\mathbf{a}_i \in \mathbf{A}} h(\mathbf{a}_i) h(\mathbf{q})^{\mathrm{T}} \tag{2}$$

Compared with common binary coding methods based on Hamming distance minimization, h is likely to be a non-linear function through MIPS, which is more suitable for the neuron retrieval database that is linearly inseparable.

In Ref. [38], Shrivastava and Li proposed the Asymmetric Locality Sensitive Hashing (ALSH) method, proving that it is impossible to inherit the high collision probability guarantee of MIPS problem under current LSH framework. In fact, by adopting two different binary coding functions $h(\cdot)$ and $z(\cdot)$ to compute the inner product of the database and query, the MIPS can be converted as the standard L_2 nearest neighbor search problem [35,38]. Accordingly, to generate more effective binary codes, we also adopt two coding functions for the MIPS problem:

$$\mathbf{p} = \arg\max_{\mathbf{a}_i \in \mathbf{A}} h(\mathbf{a}_i) z(\mathbf{q})^{\mathrm{T}} \tag{3}$$

The remaining issue is how to learn two coding functions $h(\cdot)$ and $z(\cdot)$, which can generate effective binary codes to make the query neuron finding the corresponding similar neurons in database.

Binary Coding for Neuron Retrieval: For the training neuron data set $\mathbf{A}$, we assemble another neuron set $\mathbf{X} \subset \mathbb{R}^{m \times d}$ which is randomly sampled from the training set. We denote matrix $\mathbf{S} \in \mathbb{R}^{n \times m}$ which reflect the similarity of each neuron between $\mathbf{A}$ and $\mathbf{X}$. The idea of MIPS based binary coding is to learn coding functions which can make the inner product of $\mathbf{A}$ and $\mathbf{X}$ to approximate with $\mathbf{S}$ in the form of binary codes:

$$\min_{h,z} \|h(\mathbf{A}) z(\mathbf{X})^{\mathrm{T}} - \hat{\mathbf{S}}\|^2 \tag{4}$$

where $\hat{\mathbf{S}}$ is the binarization form of $\mathbf{S}$ by its mean value. Instead of directly solving this challenging problem, we discard its quadratic part after expansion and only focus on the correlation between similarity matrix $\hat{\mathbf{S}}$ and $h(\mathbf{A}) z(\mathbf{X})^{\mathrm{T}}$. Since the discarded quadratic part is the regularization term, which does not include any ground truth information. Thereby, Eq. (4) can be re-defined as:

$$\max_{h,z} \operatorname{trace}(h(\mathbf{A})^{\mathrm{T}} \hat{\mathbf{S}} z(\mathbf{X})) \tag{5}$$

In practice, we find that omitting the quadratic part does not affect the binary coding performance. It also makes the problem easier to optimize, because the similarity term is more efficient to solve for the non-linear differentiation of neuron morphologies. Subsequently, we define two binary coding matrices $\mathbf{W}, \mathbf{R} \in \mathbb{R}^{d \times r}$ to substitute the coding functions, where $h(\mathbf{A}) = \operatorname{sgn}(\mathbf{AW})$, $z(\mathbf{X}) = \operatorname{sgn}(\mathbf{XR})$, and r is the bits of binary codes for each neuron. Generally, $\mathbf{W}$ and $\mathbf{R}$ are initialized by Principal Component Analysis (PCA) projections or random generation. Then, we obtain Eq. (5) in a new form:

$$\max_{\mathbf{W},\mathbf{R}} \operatorname{trace}(\operatorname{sgn}(\mathbf{AW})^{\mathrm{T}} \hat{\mathbf{S}} \operatorname{sgn}(\mathbf{XR})) \tag{6}$$

In this objective function, we need to optimize $\mathbf{W}$ and $\mathbf{R}$ from the training neurons $\mathbf{A}$, $\mathbf{X}$ and their similarity matrix $\mathbf{S}$. Generally, $\mathbf{S}$ is computed by inner product, $\mathbf{S} = \mathbf{AX}^{\mathrm{T}}$. However, in this work, the extracted morphological features usually express different representative of similarity for neuron cells. Simply aligning these features together to compute similarity matrix may generate ineffective binary codes. Therefore, we first consider feature diversity to compute a more suitable similarity matrix, then optimize $\mathbf{W}$ and $\mathbf{R}$ for the Eq. (6).

Feature Hierarchy: For the single type of neuron feature, e.g., j^{th} feature, its similarity matrix can be obtained by the corresponding inner product of $\mathbf{A}^{(j)}$ and $\mathbf{X}^{(j)}$:

$$\mathbf{S}^{(j)} = \mathbf{A}^{(j)} (\mathbf{X}^{(j)})^{\mathrm{T}} \tag{7}$$

Many articles [13,21] treat the above $\mathbf{S}^{(j)}$ as feature kernels and fuse multiple kernels together with different weights to compute the similarity matrix:

$$\mathbf{S} = \sum_{j=1}^{M} \mu_j \mathbf{S}^{(j)} \tag{8}$$

where μ_j is the similarity weight of j^{th} feature. In most case, there are either few types of features or few numbers of training data, so the computational complexity of $\mathbf{S}$ is acceptable. However, in large-scale neuron retrieval, many features' similarity matrices need to be calculated (38 features in this paper), and usually thousands of neurons in database should be set as the training data to ensure the retrieval precision. The computational complexity of the similarity matrix is an issue in the neuron retrieval task.

Since neuronal features are extracted from the tree-like structure, neuron retrieval can also benefit from being treated as the similarity search of the tree-like structures. Despite the fact that each type of feature has its specific meaning, they can be grouped into hierarchies according to their different levels of representation. The similarity levels can also be computed hierarchically, e.g., the measures of soma (tree's root) and branches (tree's vertices and edges) at the global level (i.e., height, width of the whole neuron cell), features measured in the branches directly connect with soma at first level branches, etc. Assuming there are L hierarchies for all types of features; the similarity matrix can be re-calculated as:

$$\mathbf{S} = \sum_{l=1}^{L} \omega_l [\mathbf{A}^{(l)} (\mathbf{Z}^{(l)})^{\mathrm{T}}] \tag{9}$$

where $A^{(l)} = [A^{(j_1)}, \ldots, A^{(j_l)}]$ means that features $j_1, \ldots, j_l$ are grouped together and they all belong to the l^{th} hierarchy. Consequently, the computation efficiency will have great improvement via the feature hierarchy process ($L \ll M$). More importantly, each hierarchical weight ω_l is much easier to acquire compared with each feature's weight μ_j. In practice, hierarchical weights are determined by the neuronal tree-like structure, and we will discuss it the experiment.

Asymmetric Optimization: After computing the similarity matrix via feature hierarchy, we adopt an asymmetric strategy to solve the optimization problem of Eq. (6).

As both $\mathbf{W}$ and $\mathbf{R}$ are constrained by the sign function, it is hard to simultaneous optimize them together. Instead, we first assume that the right part of Eq. (6) is fixed as a constant matrix $\mathbf{Z} = \operatorname{sgn}(\mathbf{XR})$, and then we consider the following sub-problem with variable $\mathbf{W}$:

$$\max_{\mathbf{W}} \operatorname{trace}(\operatorname{sgn}(\mathbf{AW})^{\mathrm{T}} \hat{\mathbf{S}} \mathbf{Z}) \tag{10}$$

In the same way, fix the left part $\mathbf{H} = \operatorname{sgn}(\mathbf{AW})$, we can obtain the sub-problem with variable $\mathbf{R}$:

$$\max_{\mathbf{R}} \operatorname{trace}(\mathbf{H}^{\mathrm{T}} \hat{\mathbf{S}} \operatorname{sgn}(\mathbf{XR})) \tag{11}$$

Compared with Eq. (6), only one sign function and coding matrix are included in each sub-problem. Based on this asymmetric design, if we can solve the two sub-problems, then optimal $\mathbf{W}$ and $\mathbf{R}$ for the whole problem can also be obtained by several alternative iterations between (10) and (11).

For the sub-problem (10), despite that only one sign function remains, it is still a discrete optimization issue. To solve this, we introduce an auxiliary variable $\mathbf{B} \in \{-1, 1\}^{n \times r}$ as the binary codes of $\mathbf{A}$ to replace the discrete part $\operatorname{sgn}(\mathbf{AW})$, and the sub-problem (10) can be separated into two terms:

$$\max_{\mathbf{B},\mathbf{W}} \operatorname{trace}[(\mathbf{B}^{\mathrm{T}} \hat{\mathbf{S}} \mathbf{Z}) - \lambda \|\mathbf{B} - \mathbf{AW}\|^2] \tag{12}$$

The first term maximizes inner product via the learned binary codes, and the second term ensures that $\mathbf{AW}$ can approximate with the target

binary codes $\mathbf{B}$. λ is denoted as a trade-off parameter between these two terms. Subsequently, $\mathbf{W}$ can be optimized by several alternative iterations with $\mathbf{B}$:

$$\begin{cases}\mathbf{B} = \mathrm{sgn}(\hat{\mathbf{S}}\mathbf{Z} + 2\lambda\mathbf{A}\mathbf{W}) \\ \mathbf{W} = \mathbf{A}^{\dagger}\mathbf{B}\end{cases} \tag{13}$$

where $\mathbf{A}^{\dagger}$ is the pseudo-inverse of $\mathbf{A}$. Optimal $\mathbf{W}$ of this sub-problem will be acquired until coverage or reach maximum t iterations.

After solving (10), we denote $\mathbf{D} \in \{-1, 1\}^{m\times r}$ as the auxiliary variable for $\mathrm{sgn}(\mathbf{XR})$, then the optimal $\mathbf{R}$ for sub-problem (11) can also be acquired in the same way:

$$\begin{cases}\mathbf{D} = \mathrm{sgn}(\hat{\mathbf{S}}^{\mathrm{T}}\mathbf{H} + 2\lambda\mathbf{X}\mathbf{R}) \\ \mathbf{R} = \mathbf{X}^{\dagger}\mathbf{D}\end{cases} \tag{14}$$

As these $\mathbf{W}$ and $\mathbf{R}$ are the local optimal results of two sub-problems, we denote such alternative iterations between coding matrices and auxiliary variables as the inner loop. To obtain the optimal coding matrices for the objective function (6), several outer alterative iterations between (10) and (11) are still needed until coverage or reach maximum iterations.

3.3. Implementation Details

Given the training neurons $\mathbf{A}$ and $\mathbf{X}$, our framework learns effective coding functions for neuron morphological retrieval using feature hierarchical binary coding with MIPS, as outlined in Algorithm 1.

Algorithm 1. Feature hierarchical binary coding with MIPS.

Input: training data $\mathbf{A}$ and $\mathbf{X}$.
Output: binary coding matrices $\mathbf{W}$ and $\mathbf{R}$.
1: Extract $\mathbf{M}$ types of morphological features for each neuron in the training data;
2: Group features into L hierarchies;
3: Compute the similarity matrix through Eq. (9);
4: Initialize binary coding matrix $\mathbf{W}$ and $\mathbf{R}$ by PCA projections;
5: **repeat**
6: Solving sub-problem (10): compute $\mathbf{W}$ by the inner loop of (13), where $\mathbf{Z} = \mathrm{sgn}(\mathbf{XR})$;
7: Solving sub-problem (11): compute $\mathbf{R}$ by the inner loop of (14), where $\mathbf{H} = \mathrm{sgn}(\mathbf{AW})$;
8: **until** converge or reach maximum T iterations

With the learned coding matrix $\mathbf{W}$, the morphological features of every neuron in the database $\mathbf{a}_i \in \mathbb{R}^{1\times d}$ can be mapped to binary codes via the coding function $h(\mathbf{a}_i) = \mathrm{sgn}(\mathbf{a}_i\mathbf{W})$. Similarly, with coding matrix $\mathbf{R}$, the binary code of query neuron is calculated by $z(\mathbf{q}) = \mathrm{sgn}(\mathbf{qR})$. Then, the similarity search problem between query neuron and the neuron database is transformed as the inner product ranking of their binary codes. For the query neuron, the similar neurons are defined as the neurons with top-K largest inner product, and these similar neurons can further be used to interpret biomedical meanings of the query neuron.

4. Experiment

In this section, we first introduce the experimental setting and present the evaluation metrics of our system for neuron morphological retrieval. Then, we demonstrate its use case in neuron identification and analysis. We also provide in-depth discussions of the proposed neuron retrieval system.

4.1. Experimental setting

Our experiments were carried out on the NeuroMorpho.org database [28], which has the largest collection of publicly accessible 3D reconstructed neuron data. Specifically, we consider the entire 17,107 Drosophia Melanogaster neurons to evaluate the retrieval performance. Each neuron's morphological information was recorded in a SWC format file including point coordinate value, soma's position, etc [2]. We employ L-measure toolbox [34] to extract 38 quantitative measurements as morphological features for each neuron. Then, each morphological feature is normalized with their mean value and standard deviation.

As mentioned in Section 2.1, we calculate the measurements in three levels, i.e., global, branch and bifurcation. In practice, for computational convenience, we also group features in such three hierarchies. The similarity weights of each hierarchy are empirically obtained by the tree-like structures and neuronal properties. Due to the linear inseparability of neuronal structure, unrelated neurons are likely to express similarities in the global viewpoint. On the other hand, neurons with some common properties (e.g., cell types, brain regions) tend to express similarities in branch structure but different in bifurcation if they are not exactly the same. Therefore, we set global hierarchy with low weight to reduce the influence of non-linear structure. Then, we assign highest weight for branch hierarchy and next-highest weight for bifurcation hierarchy to make sure that we can retrieve neurons with common properties and also differentiate them in subtle level. In the experiment, we set global, branch and bifurcation hierarchies with the weights ratio of 1:14:5, which can achieve promising performance for neuron retrieval.

In our system, training data sets $\mathbf{A}$ and $\mathbf{X}$ are random sampled, covering 80% of the whole Drosophia Melanogaster neurons database. For the MIPS based binary coding, maximum iterations of the inner loop and outer loop are 100 and 10 respectively. The trade-off parameter λ is set as 34. The length of binary codes for each neuron is scalable in our method, which is determined by the size of coding matrices $\mathbf{W}$ and $\mathbf{R}$. Generally, $\mathbf{W}$ and $\mathbf{R}$ are initialized by PCA projections. They can also be initialized by random generation if we want to obtain binary codes which are longer than the feature vectors. All experiments were conducted on a desktop with a 3.6 GHz processor of eight cores and 32 G RAM.

4.2. Evaluation of the Neuron Retrieval

To evaluate the efficacy of our method for neuron morphological retrieval problem, we compare the retrieval performance in multiple views with three state-of-the-art unsupervised binary coding and hashing methods:

1. SH [42]: Spectral hashing is a well-known algorithm which harness nonlinear manifold structure to produce neighborhood-preserving compact binary codes;
2. ITQ [9]: Iterative quantization is based on PCA projection for dimensionality reduction and minimizes quantization error via orthogonal transformation. It is a very effective binary coding method for most natural image retrieval problem;
3. AGH [20]: Anchor graph hashing discovers the neighborhood structure inherent in the data to learn appropriate compact codes, which has already shown its excellent performance in mammogram retrieval [18].

As neuron morphology is correlated with their cell types and brain regions, for the Drosophia Melanogaster neuron database which has various cell types (around 100) and brain regions (around 50), we select 233 projection neurons (PN) in olfactory bulb and 19 lateral horn neurons (LH) in protocerebrum as queries, which is consist with [16,40]. In the testing phrase, the correctly retrieved neurons are defined as if they have the same cell types and brain regions with the query neuron.

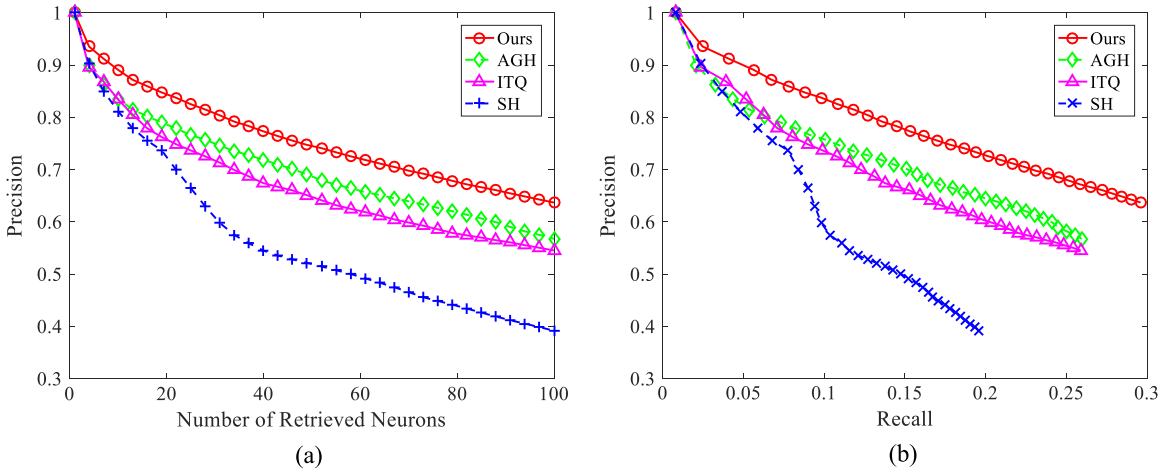


Fig. 3. Retrieval performance of four compared methods, 32 bits of binary codes are used: (a) precision curve; (b) precision-recall curve.

For all the PN and LH queries (252 in total), Fig. 3(a) shows their average retrieval precision of four competitive methods under different number of retrieved neurons. Here we denote retrieval precision as the percentage of correct neurons in all the retrieved neurons. Every method generates 32 bits of binary codes to represent each neuron. According to Fig. 3(a) we can see that our method significantly outperforms all other advanced binary coding methods in terms of retrieval precision. It mostly benefits from the feature hierarchy processing and MIPS based binary coding. As we group features into different hierarchies, in which each hierarchy reflect their corresponding levels of neuron representative, more suitable similarity matrices are obtained for the continued binary coding. Additionally, the employed MIPS baseline is more likely to generate effective binary codes for the linearly inseparable neuron data, since the inner product embedded objective function are more likely to map coding matrices into non-linear space, and the asymmetric optimization strategy provides a convergent solution. Fig. 3(b) provides the precision-recall curves for the above four methods. We can easily observe that the performance of the compared methods is consistent with the above analysis. Our method still performs the best among the compared methods..

We also report the retrieval precision of the compared methods using 16, 24 and 32 bits of binary codes respectively. As SH [42] and ITQ [9] can only generate binary codes shorter than feature vectors (38 dimensions in this paper), we only compare the precision of the four methods with bits which are less than 38. From Table. Table. 1, we find that our method can always achieve the highest precision under different bits of binary codes. These results verify the proposed method can generate more effective and representative binary codes for data residing embedded in the non-linear structure. In addition, from 16 bits to 32 bits, the retrieval precisions of our method are approximately equivalent, which demonstrate that asymmetric design for MIPS based binary coding can help us obtain convergent results in several alternative iterations.

Beside the retrieval precision, the proposed method also demonstrates the computational efficiency in the testing phrase. Compared with traditional similarity search methods such as *k*-Nearest Neighbors, our binary coding method is 30 times faster (252 queries' retrieval in 0.17 s). This merit will be particularly beneficial in the future when more dimensional features are extracted and larger scale databases are used.

Fig. 4 present four random selected query neurons and their corresponding top-5 retrieved neurons through our method. We employ *Vaa3D* [32] to display these neurons. Generally, the retrieved neurons present similar morphologies with their query neurons, which verify the effectiveness of feature extraction procedure and the proposed binary coding method..

4.3. Neuron identification and analysis

With the development of neuron tracing, an increasing number of newly reconstructed neurons are released in recent years. However, most of them lack basic annotations such as cell types, brain regions, transmitters, which block neuroscientists to study their morphologies and structures with associated functions. Therefore, identifying basic characteristic for unknown neurons is an urgent demand for further exploration.

Based on the fine-grained retrieval results, it is reasonable to apply our method for neuron identification. We select a query neuron and assume that its characteristics are unknown. After running the morphological retrieval procedure by our method, Fig. 5 shows the distribution with respect to top-20 retrieved neurons' cell types and brain regions. According to the statistical information presented in Fig. 5, the query neuron most likely locates in olfactory bulb, and it belongs to the class of projection neuron. From these characteristics, we can reasonably infer that the query neuron is relevant to drosophila's olfactory system, and it projects information to other areas. Meanwhile, the information provided in NeuroMorpho.org [28] also verifies our inference about the query neuron..

Table 1
Comparison of retrieval precision with 16, 24, 32 bits of binary codes under different number of retrieved neurons.

Method	top10			top20			top50		
	16-bit	24-bit	32-bit	16-bit	24-bit	32-bit	16-bit	24-bit	32-bit
SH [42]	0.8046	0.8116	0.8115	0.7211	0.7221	0.7264	0.5043	0.5102	0.5192
ITQ [9]	0.8278	0.8298	0.8381	0.7595	0.7599	0.7615	0.6338	0.6394	0.6483
AGH [20]	0.8254	0.8329	0.8353	0.7603	0.7861	0.7980	0.6325	0.6423	0.6874
Ours	**0.8889**	**0.8810**	**0.8909**	**0.8428**	**0.8378**	**0.8438**	**0.7436**	**0.7440**	**0.7451**

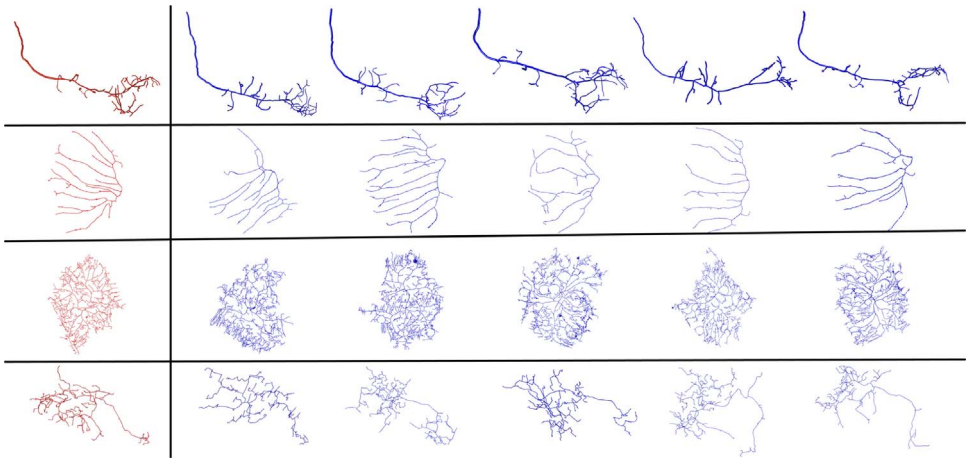


Fig. 4. For each neuron on the left (red), top-5 retrieved neurons on the right (blue) through our method, which illustrate the morphological similarity between query neurons and retrieved neurons.

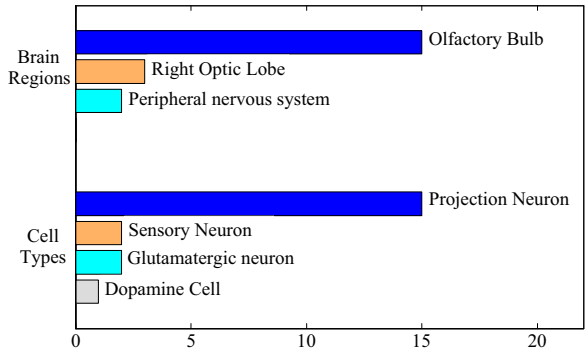


Fig. 5. The distribution of cell types and brain regions for top-20 retrieved neurons.

In addition, we find that Nanda et al. [27] also tried to add annotations of brain regions and cell types for the NeuroMorpho.org [28] database. However, they obtained brain region information from online record and brain image stacks. Then, they used the text-based query tool to search neurons with given distance (10, 20 μm) in each region to determine whether the neuron is mapped to a single region or more than one region. In addition, the authors identify cell types based on the brain regions invaded by the neurite terminals of every neuron. Obviously, compared with this method, our neuron morphological retrieval system is more suitable for the annotation of neuron database. Since the fine retrieval results provide useful reference from other similar neurons, and statistical analysis can be subsequently used to determine the most reliable annotations.

4.4. Discussions

We discuss the benefits and limitations of the proposed neuron morphological retrieval system here.

In the training phase, to obtain effective coding matrices **W** and **R**, we employed alternative iteration strategy in both inner and outer loop. Although such strategy cannot guarantee to achieve the globally optimal solution for the discrete optimization problem, at each step, the local optimum **W** and **R** is obtained with several iterations. Fig. 6(a) illustrates the accumulative value of **W** and **R** at each iteration of the outer loop. We also show the objective values of Eq. (6) with increasing number of iterations in Fig. 6(b). As we can see, **W**, **R** and the objective value can be fast converged within 10 iterations.

Another significant benefit of our proposed retrieval system is the introduction of feature hierarchy. On the one side, the extracted features have different levels of representation for the neuron morphology, considering their diversity is essential; on the other side, calculate each feature's weight is very time-consuming, and it's usually hard to acquire. Therefore, we group features into different hierarchies based on their location in the tree-like structure, then assigning each hierarchical weight empirically. This process not only considers each feature's diversity, but also reduces the computational complexity. Besides, benefited from the MIPS based binary coding design, we can assemble such hierarchical information in the similarity matrix, which

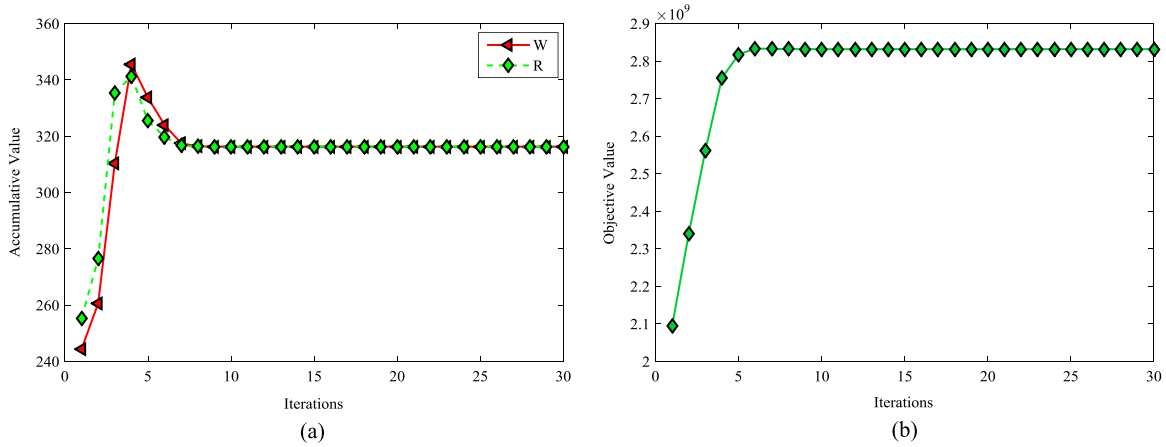


Fig. 6. Convergence properties of our method: (a) accumulative value of **W** and **R** with iterations; (b) objective value with iterations.

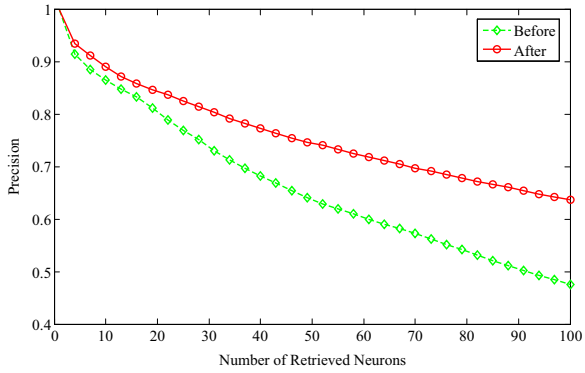


Fig. 7. Retrieval precision before and after feature hierarchy.

result in significantly improvement of the binary coding performance. Fig. 7 presents the retrieval precision curve before and after feature hierarchy, which validates the introduction of feature hierarchy can surely improve the retrieval performance..

There are also some limitations in our neuron morphological retrieval system. The first limitation is the feature extraction method. Restricted by the current morphological analysis techniques, the features are only extracted according to the tree-like structure. In order to meet the feature hierarchy strategy, we only extract 38 dimensional features from global, branch and bifurcation level respectively. Indeed, more kinds of features can be used to represent neurons, which facilitate the learning of more effective binary codes. The second limitation is the weights setting during feature hierarchy. In this paper, the weights are determined empirically, which may not be optimal for the neuron retrieval problem. A better solution is to compute weights through learning and optimization method based on label information. In addition, the trade-off parameter λ is also set empirically. In the experiment, we varied λ from 1 to 1000 and find that $\lambda = 34$ can achieve the best retrieval performance.

5. Conclusions

In this paper, we presented a large-scale morphological retrieval framework for neuron identifcation and analysis. Specifically, we first introduced the feature hierarchy strategy to consider feature diversity with low computational complexity, and then employed a novel binary coding method based on MIPS, which not only achieved fast neuron retrieval, but also diferentiated the linearly inseparable morphological space with high precision. Experimental results ver-ified the efficacy of our neuron morphological retrieval method and also illus-trated its application in neuron identifcation. Based on the present work, we will study how to extract more representative features from 3D neuron morphological data. Furthermore, we will incorporate with few experts' information to automatic learning hierarchy weights, and also design a semi-supervised binary coding method to boost the retrieval precision. We will also apply our framework to explore the relationship between neuron structure and function.

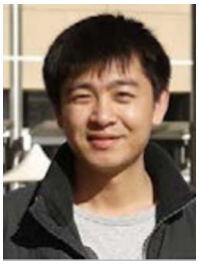
References

[1] G.A. Ascoli, D.E. Donohue, M. Halavi, NeuroMorpho.org a central resource for neuronal morphologies, J. Neurosci. 27 (2007) 9247–9251.
[2] R. Cannon, D. Turner, G. Pyapali, H. Wheal, An on-line archive of reconstructed hippocampal neurons, J. Neurosci. Methods 84 (1998) 49–54.
[3] H. Chen, T. Liu, Y. Zhao, etc., Optimization of large-scale mouse brain connectome via joint evaluation of dti and neuron tracing data. NeuroImage 115, (2015a) 202–213.
[4] H. Chen, H. Xiao, T. Liu, H. Peng, Smart tracing self-learning-based neuron reconstruction, Brain Inform. 2 (2015) 135–144.
[5] L.D.F. Costa, K. Zawadzki, M. Miazaki, M.P. Viana, S.N. Taraskin, Unveiling the neuromorphological space, Front. Comput. Neurosci. 4 (2010) 150.
[6] A. Fakhry, H. Peng, S. Ji, Deep models for brain em image segmentation novel insights and improved performance, Bioinformatics 32 (2016) 2352–2358.
[7] A.X. Falcão, L. da Fontoura Costa, B. Da Cunha, Multiscale skeletons by image foresting transform and its application to neuromorphometry, Pattern Recognit. 35 (2002) 1571–1582.
[8] A. Gionis, P. Indyk, R. Motwani, Similarity search in high dimensions via hashing, in: VLDB, (1999) pp. 518–529.
[9] Y. Gong, S. Lazebnik, A. Gordo, F. Perronnin, Iterative quantization a procrustean approach to learning binary codes for large-scale image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 2916–2929.
[10] S. Gulyanon, N. Sharifai, S. Bleykhman, E. Kelly, M. Kim, A. Chiba, G. Tsechpenakis, Three-dimensional neurite tracing under globally varying contrast, in: ISBI, (2015) pp. 875–879.
[11] R. He, Y. Cai, T. Tan, L. Davis, Learning predictable binary codes for face indexing, Pattern Recognit. 48 (2015) 3160–3168.
[12] S. Ji, Computational genetic neuroanatomy of the developing mouse brain dimensionality reduction, visualization, and clustering, BMC Bioinforma. 14 (2013) 222–236.
[13] M. Jiang, S. Zhang, J. Huang, L. Yang, D.N. Metaxas, Joint kernel-based supervised hashing for scalable histopathological image analysis, in: MICCAI, 2015, pp. 366–373.
[14] W. Kong, W.J. Li, Isotropic hashing, in: NIPS, 2012, pp. 1646–1654.
[15] B. Kulis, P. Jain, K. Grauman, Fast similarity search for learned metrics, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2009) 2143–2157.
[16] Z. Li, F. Shen, R. Fang, S. Conjeti, A. Katouzian, S. Zhang, Maximum inner product search for morphological retrieval of large-scale neuron data, in: ISBI, 2016, pp. 602–606.
[17] G. Lin, C. Shen, Q. Shi, A. van den Hengel, D. Suter, Fast supervised hashing with decision trees for high-dimensional data, in: CVPR, 2014, pp. 1971–1978.
[18] J. Liu, S. Zhang, W. Liu, X. Zhang, D.N. Metaxas, Scalable mammogram retrieval using anchor graph hashing, in: ISBI, 2014a, pp. 898–901.
[19] W. Liu, J. Wang, R. Ji, Y.G. Jiang, S.F. Chang, Supervised hashing with kernels, in: CVPR, 2012, pp. 2074–2081.
[20] W. Liu, J. Wang, S. Kumar, S.F. Chang, Hashing with graphs, in: ICML, 2011, pp. 1–8.
[21] X. Liu, J. He, B. Lang, Multiple feature kernel hashing for large-scale visual search, Pattern Recognit. 47 (2014) 748–757.
[22] M.H. Longair, D.A. Baker, J.D. Armstrong, Simple neurite tracer open source software for reconstruction, visualization and analysis of neuronal processes, Bioinformatics 27 (2011) 2453–2454.
[23] E. Meijering, Neuron tracing in perspective, Cytom. Part A 77 (2010) 693–704.
[24] S. Mesbah, S. Conjeti, A. Kumaraswamy, P. Rautenberg, N. Navab, A. Katouzian, Hashing forests for morphological search and retrieval in neuroscientific image databases, in: MICCAI, 2015, pp. 135–143.
[25] S. Mukherjee, S. Basu, B. Condron, S.T. Acton, Tree2tree2: Neuron tracing in 3d, in: ISBI, (2013) pp. 448–451.
[26] S. Mukherjee, B. Condron, S.T. Acton, Tubularity flow field-a technique for automatic neuron segmentation, IEEE Trans. Image Process. 24 (2015) 374–389.
[27] S. Nanda, M.M. Allaham, M. Bergamino, S. Polavaram, R. Armañanzas, G.A. Ascoli, R. Parekh, Doubling up on the fly neuromorpho. org meets big data, Neuroinformatics 13 (2015) 127–129.
[28] NeuroMorpho, Neuron morphology repository. ⟨http://neuromorpho.org/neuroMorpho/index.jsp/⟩.(accessed 06.10.15).
[29] M. Norouzi, D.M. Blei, Minimal loss hashing for compact binary codes, in: ICML, 2011, pp. 353–360.
[30] M. Norouzi, D.J. Fleet, R.R. Salakhutdinov, Hamming distance metric learning, in: NIPS, 2012, pp. 1061–1069.
[31] R. Parekh, R. Armañanzas, G.A. Ascoli, The importance of metadata to assess information content in digital reconstructions of neuronal morphology, Cell Tissue Res. 360 (2015) 121–127.
[32] H. Peng, Z. Ruan, F. Long, J.H. Simpson, E.W. Myers, V3d enables real-time 3d visualization and quantitative analysis of large-scale biological image data sets, Nat. Biotechnol. 28 (2010) 348–353.
[33] M. Raginsky, S. Lazebnik, Locality-sensitive binary codes from shift-invariant kernels, in: NIPS, 2009, pp. 1509–1517.
[34] R. Scorcioni, S. Polavaram, G.A. Ascoli, L-measure a web-accessible tool for the analysis, comparison and search of digital reconstructions of neuronal morphologies, Nat. Protoc. 3 (2008) 866–876.
[35] F. Shen, W. Liu, S. Zhang, Y. Yang, H.T. Shen, Learning binary codes for maximum inner product search, in: ICCV, (2015a) pp. 4148–4156.
[36] F. Shen, C. Shen, W. Liu, H. Tao Shen, Supervised discrete hashing, in: CVPR, (2015b) pp. 37–45.
[37] F. Shen, C. Shen, Q. Shi, A. van den Hengel, Z. Tang, H.T. Shen, Hashing on nonlinear manifolds, IEEE Trans. Image Process. 24 (2015) 1839–1851.
[38] A. Shrivastava, P. Li, Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips), in: NIPS, 2014, pp. 2321–2329.
[39] M.G. Uzunbas, C. Chen, D. Metaxas, An efficient conditional random field approach for automatic and interactive neuron segmentation, Med. Image Anal. 27 (2016) 31–44.
[40] Y. Wan, F. Long, L. Qu, H. Xiao, M. Hawrylycz, E.W. Myers, H. Peng, Blastneuron for automated comparison, retrieval and clustering of 3d neuron morphologies, Neuroinformatics 13 (2015) 487–499.
[41] J. Wang, W. Liu, S. Kumar, S.F. Chang, Learning to hash for indexing big dataa survey. Proceedings of the IEEE 104, 2016, pp. 34–57.
[42] Y. Weiss, A. Torralba, R. Fergus, Spectral hashing, in: NIPS, 2009, pp. 1753–1760.
[43] Q. Wen, A. Stepanyants, G.N. Elston, A.Y. Grosberg, D.B. Chklovskii, Maximization of the connectivity repertoire as a statistical principle governing the shapes of dendritic arbors. Proceedings of the National Academy of Sciences 106, 2009, 1, pp.

2536–12541.
[44] G. Wu, Q. Wang, D. Zhang, F. Nie, H. Huang, D. Shen, A generative probability model of joint label fusion for multi-atlas based brain segmentation, Med. Image Anal. 18 (2014) 881–890.
[45] J. Xie, T. Zhao, T. Lee, E. Myers, H. Peng, Automatic neuron tracing in volumetric microscopy images with anisotropic path searching, in: MICCAI, 2010, pp. 472–479.
[46] S. Zhang, M. Yang, T. Cour, K. Yu, D.N. Metaxas, Query specific rank fusion for image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 37 (2015) 803–815.
[47] X. Zhang, H. Dou, T. Ju, J. Xu, S. Zhang, Fusing heterogeneous features from stacked sparse autoencoder for histopathological image analysis, IEEE J. Biomed. Health Inform. 20 (2015) 1377–1383.
[48] X. Zhang, W. Liu, M. Dundar, S. Badve, S. Zhang, Towards large-scale histopathological image analysis hashing-based image retrieval, IEEE Trans. Med. Imaging 34 (2015) 496–506.
[49] X. Zhang, F. Xing, H. Su, L. Yang, S. Zhang, High-throughput histopathological image analysis via robust cell segmentation and hashing, Med. Image Anal. 26 (2015) 306–315.
[50] H. Zhao, Z. Wang, P. Liu, The ordinal relation preserving binary codes, Pattern Recognit. 48 (2015) 3169–3179.
[51] Z. Zhou, X. Liu, B. Long, H. Peng, TReMAP automatic 3d neuron reconstruction based on tracing, reverse mapping and assembling of 2d projections, Neuroinformatics 14 (2016) 41–50.
[52] Z. Zhou, S. Sorensen, H. Zeng, M. Hawrylycz, H. Peng, Adaptive image enhancement for tracing 3d morphologies of neurons and brain vasculatures, Neuroinformatics 13 (2015) 153–166.

Zhongyu Li is a Ph.D. student in the Department of Computer Science at the University of North Carolina at Charlotte. He received his B.E. and M.E. degree from Xi'an Jiaotong University in 2012 and 2015 respectively. His research focuses on medical imaging and machine learning.

Dr. Ruogu Fang is an Assistant Professor in the School of Computing and Information Sciences at Florida International University. She received her PhD in Electrical and Computer Engineering from Cornell University in 2014 and B.E in Information Engineering from Zhejiang University in 2009. Dr. Fang's research aims to explore intelligent approaches to bridge the data and medical informatics via machine learning and data mining. She has published in top-tier journals and conferences. Dr. Fang is a guest editor of Computerized Medical Imaging and Graphics, organizer of the International Workshop on Sparsity Techniques in Medical Imaging, and a member of the IEEE and ASNR.



Fumin Shen received his B.S. and Ph.D. degree from Shandong University and Nanjing University of Science and Technology, China, in 2007 and 2014, respectively. Currently he is a lecturer in school of Computer Science and Engineering, University of Electronic of Science and Technology of China, China. His major research interests include computer vision and machine learning, including face recognition, image analysis, hashing methods, and robust statistics with its applications in computer vision.

Dr. Shaoting Zhang is an Assistant Professor in the Department of Computer Science at the University of North Carolina at Charlotte. Before joining UNC Charlotte, he was a faculty member in the Department of Computer Science at Rutgers-New Brunswick (Research Assistant Professor, 2012–2013). He received PhD in Computer Science from Rutgers in 01/2012, M.S. from Shanghai Jiao Tong University in 2007, and B.E. from Zhejiang University in 2005. Dr. Zhang's research is on the interface of medical imaging informatics, large-scale retrieval and machine learning. He has published productively in top conferences and journals and registered multiple patents or invention disclosures. Dr. Zhang is an associate editor of Neurocomputing, guest editor of CMIG, and a senior member of the IEEE.