

Methods

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

Interactive Exploration for Continuously Expanding Neuron Databases

Zhongyu Li ^a, Dimitris N. Metaxas ^b, Aidong Lu ^a, Shaotong Zhang ^{a,*}^aDepartment of Computer Science, University of North Carolina at Charlotte, USA^bDepartment of Computer Science, Rutgers University, USA

ARTICLE INFO

Article history:

Received 15 July 2016

Received in revised form 15 February 2017

Accepted 16 February 2017

Available online 17 February 2017

Keywords:

Neuron Morphology

Large-Scale Retrieval

Binary Coding

Online Updating

User Interaction

ABSTRACT

This paper proposes a novel framework to help biologists explore and analyze neurons based on retrieval of data from neuron morphological databases. In recent years, the continuously expanding neuron databases provide a rich source of information to associate neuronal morphologies with their functional properties. We design a coarse-to-fine framework for efficient and effective data retrieval from large-scale neuron databases. In the coarse-level, for efficiency in large-scale, we employ a binary coding method to compress morphological features into binary codes of tens of bits. Short binary codes allow for real-time similarity searching in Hamming space. Because the neuron databases are continuously expanding, it is inefficient to re-train the binary coding model from scratch when adding new neurons. To solve this problem, we extend binary coding with online updating schemes, which only considers the newly added neurons and update the model on-the-fly, without accessing the whole neuron databases. In the fine-grained level, we introduce domain experts/users in the framework, which can give relevance feedback for the binary coding based retrieval results. This interactive strategy can improve the retrieval performance through re-ranking the above coarse results, where we design a new similarity measure and take the feedback into account. Our framework is validated on more than 17,000 neuron cells, showing promising retrieval accuracy and efficiency. Moreover, we demonstrate its use case in assisting biologists to identify and explore unknown neurons.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

Analyzing single neuron properties, such as cell types, brain regions, functions, and development stages, is usually a fundamental task for understanding the nervous system and brain working mechanism. Given the huge numbers of neuron cells in the human brain, it is infeasible to understand every neuron's properties through traditional biological experimentation and quantitative computation. Generally, neuron morphology plays a major role in determining neuron's network connectivity, functional and physiological properties. It is therefore reasonable and essential to explore neuronal properties according to their morphologies. Recent developments in the frontiers of neuroscience (e.g., BigNeuron [1]) have greatly facilitated research in neuron morphology, and an expanding number of neurons are being reconstructed and added to the public repositories [3,2]. These fast-growing large databases provide a new avenue to help biologists explore and analyze neuronal properties [4,40,24]. Specifically, given an unknown neuron, we can retrieve neurons with similar morpholo-

gies in the databases. These retrieved similar neurons can be used to identify the unknown neuron and discover latent knowledge of their morphologies and properties.

Recently, researchers have been actively investigating this neuron morphological retrieval issue. For example, Costa et al. [9] first presented *NBLAST* to measure pairwise neuronal similarity. *NBLAST* considers both the position and local geometry, decomposing neurons into short segments and score matched segments to decide the similarity level among neurons. Subsequently, Wan et al. [49] designed *BlastNeuron* for automated comparison, retrieval, and clustering of 3D neuron morphologies. In the retrieval stage, *BlastNeuron* searches similar neurons via the normalization of the ranked scores in terms of the similarity of feature vectors. Despite their high accuracy, these two methods could be inefficient when handling large-scale neuron databases. Mesbah et al. [32] 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. The *Hashing Forest* has achieved efficient and accurate results in neuron retrieval [7,6]. Nonetheless, it usually needs a large number of binary bits (e.g., larger than 128), and its efficiency can be further improved with shorter codes. Accordingly, how to search similar neurons in large-scale

* Corresponding author.

E-mail address: rutgers.shaotong@gmail.com (S. Zhang).

databases with high efficiency and accuracy is the main focus in neuron retrieval.

As described in [32], binary coding and hashing have been widely applied in content-based image retrieval (CBIR), which target efficient similarity search in large-scale databases [50,56,55,58]. Binary coding/hashing methods usually train a coding function from the batched image databases, and then employ the coding function to transform image features into short binary codes. Many representative methods have been proposed in recent years, including, but not limited to, Spectral Hashing (SH) [51], Iterative Quantization (ITQ) [15], and Asymmetric Inner-product Binary Coding (AIBC) [48]. Despite the good performance in CBIR problem, they may not be directly applicable to the neuron retrieval problem, since the neuron databases are continuous expanding. Due to the recently well-developed neuron tracing techniques, an increasing number of neuron cells are reconstructed and added to the databases in a streamed manner. For example, the NeuroMorpho database [3] usually releases 1000–2000 reconstructed neuron cells with each update. If we re-train the coding function every time from scratch, using both the original and the newly added neuron batch, it is very time-consuming and adversely affects the efficiency of exploration.

Besides the problems of efficiency, there are two limitations influencing the retrieval accuracy of the neuron data. First, the neuron databases lack supervised information, i.e., no sufficient annotations to label the class of every neuron [4,34]. Generally, supervised retrieval is more accurate than unsupervised retrieval, since it can bridge the gap between low-level image descriptors and high-level semantic meaning. In the neuron retrieval problem, if we only consider the low-level neuronal morphologies, the retrieval results may not be consistent with their functions and properties. Second, binary coding can only provide coarse retrieval results for neuron morphological data [27,59,57]. As each neuron cell has a tree-topological structure, the difference among neuron morphologies can be subtle. Moreover, coding functions do not have a one-to-one correspondence when mapping morphological features into binary codes. This may present a difficult question, in which some unrelated neurons are represented by the same binary codes. The above two problems should be addressed to achieve good retrieval performance in the neuron morphological data.

To alleviate these problems, we designed a novel framework to achieve accurate and efficient data retrieval from large-scale neuron morphological databases. Specifically, we employ a matrix sketching technique [25] for binary coding, which can learn coding functions from the sketched neuron data, significantly reducing the matrix size of neuron databases, and continues to give good approximations with an orthogonal matrix. To tackle the continuously expanding neuron databases, we extend the binary coding with an online updating scheme, where the coding function can be updated on-the-fly without accessing whole neuron databases. Subsequently, based on the coarse retrieval results from binary coding, we introduce domain experts/users in our framework, which can give relevance feedback to improve the retrieval accuracy. In our feedback model, domain experts/users are only required to label the relevant samples with respect to query neurons from top-z results. Then, the similarity levels of the unlabeled neurons are re-ranked accordingly through our newly designed similarity measure. To the best of our knowledge, this is the first work that focuses on the interactive exploration of the continuously expanding neuron databases.

The remaining paper is organized as follows: Section 2 briefly reviews work related to 3D neuron morphology and content-based image retrieval. Section 3 provides the details of binary coding and interactive neuron retrieval. Followed by experimental results and discussion of potential use cases in Section 4. Finally, Section 5 concludes the paper and presents future work.

2. Preliminaries

This paper pursues neuron exploration via the retrieval of morphological data, which has a strong multidisciplinary component that involves a nexus of ideas from neuroscience, machine learning and information retrieval. In this section, we provide an introduction to 3D neuron morphology and content-based image retrieval.

2.1. 3D neuron morphology

Benefiting from recent advances in microscopy imaging and neuron reconstruction [42,61,38], researchers have gradually elucidated the 3D neuron morphology. Currently, two kinds of microscopy imaging techniques can be used to obtain neuron images, i.e., light microscopy (LM) and electron microscopy (EM). LM images have long been used as a fundamental tool for neuroscientists. EM images can usually achieve higher resolution and magnification compared with LM images, but also have a higher cost and require harsher specimen processing. Researchers have developed a variety of methods for the reconstruction of LM and EM images [10,37,54]. Generally, from the original 2D microscopy images to the 3D morphological data, a neuron tracing system consists of several processing steps, including image preprocessing (e.g., noise reduction, deconvolution, mosaicing), segmentation (e.g., soma, dendritic trees, spines, axons segmentation), reconstruction and connection ([5,31,52,16,62,11,18]). Fig. 1(a) illustrates a neuron slice [33] which includes microscopy images acquired from multiple view points, and Fig. 1(b) presents the corresponding 3D reconstructed neuron morphological data through the neuron tracing software Vaa3D [38]. Accordingly, 3D morphological data can provide precise descriptions of neuronal shape and structure.

Given the 3D neuron morphological data, one critical problem is how to mathematically represent these neurons for the further retrieval and analysis, i.e., extracting feature vectors for each neuron. Unlike 2D biological images which can extract features with many well-studied algorithms, extracting good features from 3D neuron data is still a challenging problem. Costa et al. [8] first proposed the concept of neuromorphological space, which introduced many quantitative measurements (e.g., neuronal height, number of branches) based on the neuron's tree-topological structure. Subsequently, many researchers have employed these quantitative measurements as morphological features to represent each neuron [9,49,32]. For example, Wan et al. [49] employed several global measurements to reveal the overall morphology of neurons.

In this paper, we also utilize these quantitative measurements as neuron morphological features. Specifically, as illustrated in Fig. 1(c), we compute three levels of measurements to reflect neuron morphologies comprehensively, i.e., bifurcation, branch and global [23]. For example, the features in branch level indicate the measurements regarding neuron branches that are directly connected to soma, such as the branch length, the Euclidean distance from compartments to somas. The above three levels of measurements are invariant to translation, rotation and the tracing resolution. In the following sections, we employ the assembled measurements as feature vectors to represent each neuron cell for retrieval and analysis.

2.2. Content-based image retrieval

Content-based image retrieval (CBIR) is a long-term research topic which aims at searching similar images by their content. As a comprehensive application in computer vision and machine learning, CBIR has developed many branches for different concerns and targets. In this subsection, we introduce its two branches, i.e., hashing and human interaction, which are the main focus in our neuron retrieval framework.

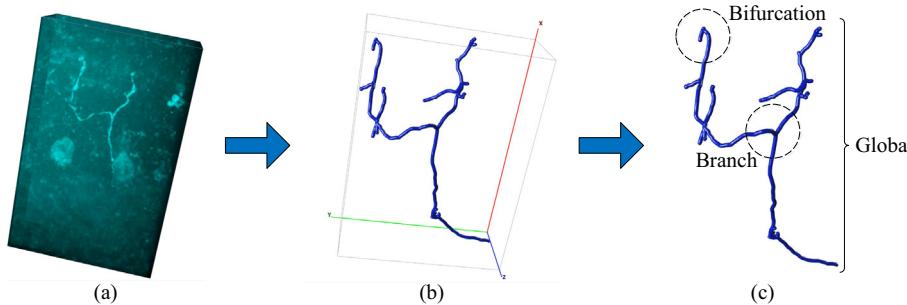


Fig. 1. From original microscopy slice to 3D neuron morphology: (a) original microscopy slice; (b) 3D reconstructed neuron morphology; (c) three levels of measurements for feature extraction.

Numerous methods of binary coding/hashing have been proposed in recent years. In CBIR, by compressing long feature vectors into short binary codes, similarity search will be much more efficient in binary Hamming space compared to high dimensional feature space. The key challenge is how to obtain coding functions which can not only transform feature vectors via binary codes, but also keep similarity and diversity among the original data. One major taxonomy of hashing methods is whether they need specified training data to obtain the coding functions, i.e., data-independent and data-dependent. Locality-Sensitive Hashing (LSH) and its variants [14,20,41] are one of the most representative data-independent methods. Despite the fact that these methods can generalize coding functions to compact any given datasets, they usually need long bits of code to ensure good performance. For the data-dependent category, a large number of methods have been proposed in recent years. While these methods can only learn coding functions for given datasets, they are usually more accurate and efficient. Representative methods include Iterative Quantization (ITQ) [15], AGH [29], Isotropic Hashing (IsoHash) [19], Minimal Loss Hashing (MLH) [35,36], FastHash [26], etc. Despite the good performance that these methods have achieved, all of them are batch based methods, which learn coding functions from one batch of training data. For the continuously expanding neuron databases, as neurons are released in streamed fashion, we cannot directly adopt the above methods for the neuron retrieval problem.

In order to improve retrieval performance and reduce the semantic gap, some CBIR systems introduce domain experts/users in the loop, which can interactively provide relevance feedback for the previous retrieval results. Generally, for an image query and its coarse retrieval results, there are three kinds models to give relevance feedback: (1) positive feedback, where the users only need to select relevant images; (2) positive-negative feedback, where the users need to select both relevant and irrelevant images; (3) positive-neutral-negative feedback, where the users need to specify the degree of relevance for all the retrieved images. A comprehensive review of the early work on relevance feedback for CBIR is presented in [60]. Most of the early approaches use the marked images as individual queries and combine the retrieval results to refine the similarity weights of relevant images [45,44]. In recent years, many algorithms in the machine learning field have been used for the interactive CBIR problem, e.g., random forests [39], graph-cut [46], random walk [21], manifold learning [17]. All these methods can improve the retrieval performance with several rounds of interactive feedback. However, it is still a challenging issue when applying user interaction in large-scale neuron databases.

3. Methodology

This section presents the methodological details of our neuron retrieval framework, including binary coding with online updating and interactive neuron retrieval.

3.1. Overview

Fig. 2 shows an overview of our neuron retrieval framework. The first part is training binary coding model (i.e., coding functions) for the neuron databases. We apply matrix sketching method on the feature vectors which are extracted from the original neuron databases, generating the initial values (including data sketch and virtual sample) for subsequent online updating. When a new neuron batch becomes available, we combine it with the aforementioned virtual sample. This combination can overcome the mean-varying problem in continuously expanding databases. The sketching result can subsequently be used for binary coding, updating the current coding function. The new data sketch and virtual sample are also stored for the next update. In the interactive neuron retrieval part, feature vectors of query neuron and all neurons in current databases can be compressed into short binary codes based on the updated coding functions. Then the similarity search between query neuron and neurons in databases are transformed into the Hamming distance ranking between their binary codes. To further improve the retrieval performance, we present top ranked neurons to users. Users will give relevance feedback to clarify which is similar with the query neuron. Our framework can process the above feedback and re-rank the retrieval results. Finally, the refined retrieved neurons can be used to help biologists to explore and analyze the query neuron.

3.2. Binary coding with online updating

Matrix Sketching based Binary Coding: The goal of binary coding is to compress feature vectors into short binary codes, keeping diversities and similarities among original data. Denote a training neuron database $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_n\} \subset \mathbb{R}^{n \times d}$, which includes n neurons, and each neuron has d dimension of features. We aim to learn a coding function $\mathbf{W} \in \mathbb{R}^{d \times r}$ that every normalized neuron feature in \mathbf{X} can be transformed into r bits of binary codes, i.e., $h(\mathbf{x}_i) = \text{sgn}((\mathbf{x}_i - \bar{\mathbf{X}})\mathbf{W})$, where $\bar{\mathbf{X}}$ is the mean value of \mathbf{X} . Note that feature normalization with zero mean is a crucial step in binary coding, especially for neuron data, because each dimension of features has their physical meaning. To learn effective binary codes, usually two requirements should be satisfied: (1) binary bits are uncorrelated and their variances are maximal; (2) numbers of 0 and 1 are roughly equal in learned binary codes of \mathbf{X} . Same as [50], the requirements are satisfied by maximizing the following objective function:

$$J(\mathbf{W}) = \frac{1}{n} \text{trace}(\mathbf{W}^T (\mathbf{X} - \bar{\mathbf{X}})^T (\mathbf{X} - \bar{\mathbf{X}})\mathbf{W}), \quad \text{s.t. } \mathbf{W}^T \mathbf{W} = \mathbf{I}_{r \times r} \quad (1)$$

Instead of directly optimizing the above objective function, we apply the matrix sketching technique [25,12,22] on the training database to learn coding function from data sketch. Sketching is

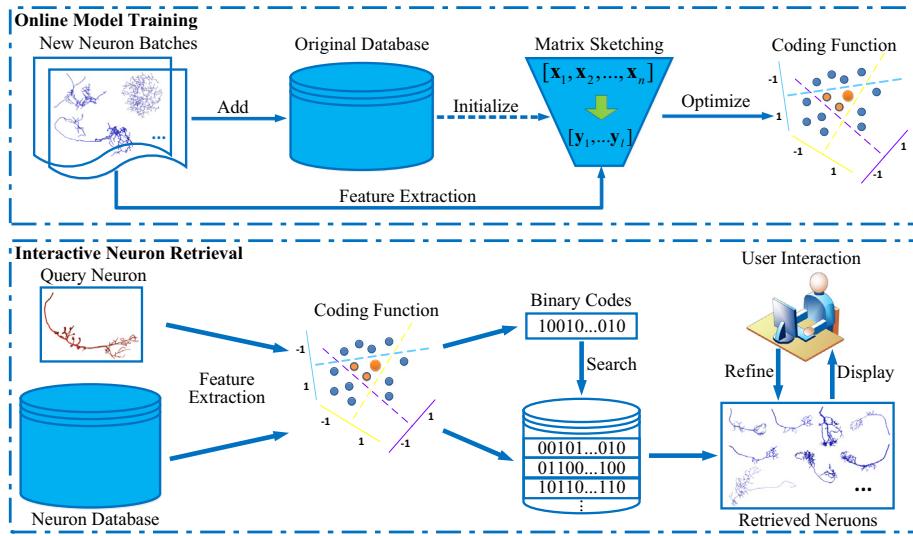


Fig. 2. Overview of our proposed framework, including online model training and interactive neuron retrieval.

a data compression technique which can significantly reduce the data size, without losing much data properties. Specifically, for the neuron database \mathbf{X} , we denote its matrix sketch as $\mathbf{Y} \in \mathbb{R}^{l \times d}$, which has the property $\mathbf{Y}^T \mathbf{Y} \approx (\mathbf{X} - \bar{\mathbf{X}})^T (\mathbf{X} - \bar{\mathbf{X}})$. We employ the *Frequent-directions* (FD) algorithm [13] to compute \mathbf{Y} , as this algorithm can effectively keep the property of matrix sketch. More importantly, FD is a streaming algorithm which can sequentially process the training data. In other words, when a new data batch comes, the FD algorithm will update the current sketch which only consider the new data one by one, without accessing the previously processed data. We show the advantage of such a streaming strategy for the online updated neuron data in the next subsection.

Given the data matrix \mathbf{X} , the FD algorithm can obtain its sketch \mathbf{Y} with a much smaller data size ($l \ll n$). Then, the objective function of binary coding can be re-written as:

$$J(\mathbf{W}) \approx \frac{1}{n} \text{trace}(\mathbf{W}^T \mathbf{Y}^T \mathbf{Y} \mathbf{W}), \quad \text{s.t. } \mathbf{W}^T \mathbf{W} = \mathbf{I}_{r \times r} \quad (2)$$

This objective function is exactly the same as that of Principle Component Analysis (PCA). The optimal coding function \mathbf{W} can be obtained by taking the top r eigenvectors of the data covariance matrix $\mathbf{Y}^T \mathbf{Y}$ [15]. In addition, to alleviate the unbalance of different dimensions in neuron data, we adopt orthogonal rotation \mathbf{R} for the above coding function, where $\mathbf{W} = \mathbf{W} \mathbf{R}$. However, since the sketch \mathbf{Y} is much smaller than the whole training data \mathbf{X} , we cannot learn the optimized \mathbf{R} as ITQ [15], which relies on all training data. Instead, we generate a random orthogonal rotation matrix, which achieves promising accuracy and efficiency in our experiments.

Online Coding Function Updating: When new neuron batches are added to the database, we need to update the coding function accordingly to maintain the retrieval performance. Re-training the coding function from scratch is very time-consuming, and sometimes infeasible when the existing neuron database is too large to load into memory. Considering that the FD algorithm can compute the data sketch in a streaming manner, an intuitive solution is to set the previous database sketch as the initial value, and then employ the FD algorithm to compute the sketch for newly added data. The coding function can be also updated with the newly computed data sketch. Unfortunately, this approach is impractical because of the aforementioned feature normalization requirement in binary coding. As the neuron database is continuously changing, the mean value for normalization is also changed.

How to overcome this mean-varying problem is a critical step to online update the coding function.

Assume \mathbf{B}_k is the newly added batch at round k which include m_k neurons, and the current database is denoted as $\mathbf{X}_k = \{\mathbf{B}_0, \mathbf{B}_1, \dots, \mathbf{B}_k\}$, where \mathbf{B}_0 is the original neuron database. Then, the mean value of \mathbf{X}_k can be computed as:

$$\bar{\mathbf{X}}_k = \frac{\bar{\mathbf{X}}_{k-1} \cdot n_{k-1} + \bar{\mathbf{B}}_k \cdot m_k}{n_k} \quad (3)$$

where $\bar{\mathbf{B}}_k$ is the mean value of \mathbf{B}_k and $n_k = \sum_{i=0}^k m_i$. Obviously, the mean value of the neuron database is changed in each update. To solve this problem, we introduce a virtual sample \mathfrak{I}_k , which considers the difference of mean value between the previous database and the current batch [43,22]:

$$\mathfrak{I}_k = \sqrt{\frac{n_{k-1} m_k}{n_k}} (\bar{\mathbf{B}}_k - \bar{\mathbf{X}}_{k-1}) \quad (4)$$

Combining this virtual sample with the currently added neuron batch, we can obtain a data set $\hat{\mathbf{B}}_k$:

$$\hat{\mathbf{B}}_k = [\mathbf{B}_k - \bar{\mathbf{B}}_k, \mathfrak{I}_k] \quad (5)$$

At round k , we have a new set of data $\hat{\mathbf{X}}_k = \{\mathbf{B}_0 - \bar{\mathbf{B}}_0, \hat{\mathbf{B}}_1, \dots, \hat{\mathbf{B}}_k\}$. According to [43,22], in each update, $\hat{\mathbf{X}}_k$ takes the shift of mean into account and corrects such a shift by the virtual sample in $\hat{\mathbf{B}}_k$. More importantly, combining with Eq. (3), we find that $\hat{\mathbf{X}}_k^T \hat{\mathbf{X}}_k = (\mathbf{X}_k - \bar{\mathbf{X}}_k)^T (\mathbf{X}_k - \bar{\mathbf{X}}_k)$. This property indicates that the data sketch of $\hat{\mathbf{X}}_k$ and $\mathbf{X}_k - \bar{\mathbf{X}}_k$ is the same. Since $\hat{\mathbf{X}}_k$ has no mean-varying problem, we can employ the aforementioned FD algorithm to sketch the continuously updated neuron data, i.e., sketch $\hat{\mathbf{B}}_k$ to obtain \mathbf{Y}_k , initialized by the previous data sketch \mathbf{Y}_{k-1} . Then the coding function can be also updated on-the-fly via the matrix sketching based binary coding. In each update, without accessing the entire neuron databases, we only need to keep the mean value $\bar{\mathbf{X}}_k$ (through Eq. 3), data size n_k (through $n_k = n_{k-1} + m_k$) and the data sketch \mathbf{Y}_k . Therefore, this binary coding with online updating scheme can efficiently tackle the continuously expanding neuron databases for further retrieval and analysis.

3.3. Interactive neuron retrieval

According to the above binary coding method, as illustrated in Fig. 2, we can compress the query neuron and all neurons in the current database into short binary codes, through the learnt coding functions. Then the similar neurons can be retrieved based on their Hamming distance ranking with the query neuron. As discussed in Section 1, binary coding can only provide coarse retrieval results for the neuron morphological data. Therefore, given the coarsely retrieved neurons (e.g., neurons with top-Z minimum Hamming distance), we propose to introduce domain experts/users in the framework, which can interactively provide relevance feedback to refine the retrieval results.

Fig. 3 presents an illustration of our user interaction interface. For a query neuron, it first searches similar neurons based on the aforementioned binary coding method. Then we display the top-z ($z = 16$ in Fig. 3) retrieved neurons to users, and users will compare and observe these neurons to decide whether they are relevant to the query neuron. This feedback scheme is easily implemented since it requires users to give only one-click inputs. Unlike many interactive models which require users to specify the class of the retrieval results [39,53,30], our strategy is particularly suitable for neuron databases which have insufficient annotations to classify every neuron.

After receiving the interactive feedback from users, our framework is able to process this feedback to improve the retrieval performance. Benefiting from the binary coding step which can efficiently provide the coarse retrieval results, we first define the outer scope size Z , where most similar neurons are included in the top-Z coarse results. Subsequently, we define the inner scope size z , which represents the number of neurons that should be presented to the users at each feedback round. In practice, Z is larger than z but much smaller than the size of the whole neuron database. During the interactive neuron retrieval phase, we focus on re-ranking these Z coarse neurons to obtain fine-grained results.

In the t th round of relevance feedback, $\mathbf{V}^{(t)}$ is the set of labeled similar samples from users which include $m^{(t)}$ neurons. As neurons in $\mathbf{V}^{(t)}$ are all similar with the query neuron, we can assemble them together to interpret and re-rank the similarities for the rest of the unlabeled neurons. Denoting $E(\mathbf{x}_i, \mathbf{x}_j)$ as the similarity measure between two neurons \mathbf{x}_i and \mathbf{x}_j , for one unlabeled neuron in t th round of relevance feedback $\mathbf{x}_i^{(t)}$, we re-define its similarity with the query neuron as follows:

$$E_i^{(t)} = \lambda E(\mathbf{x}_i^{(t)}, \mathbf{x}_q) + (1 - \lambda) \frac{1}{m^{(t)}} \sum_{j=1}^{m^{(t)}} E(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)}) \quad (6)$$

where \mathbf{x}_q is the query neuron and $\mathbf{x}_j^{(t)}$ is the neuron in $\mathbf{V}^{(t)}$. The above objective function indicates that if an unlabeled neuron has similarity with the query neuron, it should be similar with the labeled neurons as well to some extent under the trade-off parameter λ .

According to Eq. (6), how to compute the similarity measure is a critical issue for re-ranking the unlabeled neurons. In most scenarios of image retrieval, the similarity measure between two images is defined as the Euclidean distance of their feature vectors. However, this similarity measure cannot be directly applied for the neuron morphological data, since each dimension of neuron features are distinct quantitative measurements which have different levels of representation. As discussed in Section 2.1, we compute three levels of measurements as features based on the neuron's tree-topological structure, i.e., global, branch and bifurcation. Accordingly, we propose to group features into these three levels and assign them with different representative weights (i.e., ω_{gl} , ω_{br} , ω_{bi}) to compute a more accurate similarity measure. For

the two neurons \mathbf{x}_i and \mathbf{x}_j , their new similarity measure can be formulated as:

$$E(\mathbf{x}_i, \mathbf{x}_j) = \omega_{gl} D_{gl}(\mathbf{x}_i, \mathbf{x}_j) + \omega_{br} D_{br}(\mathbf{x}_i, \mathbf{x}_j) + \omega_{bi} D_{bi}(\mathbf{x}_i, \mathbf{x}_j) \quad (7)$$

where D_{gl} , D_{br} , D_{bi} denotes the normalized Euclidean distance of global, branch and bifurcation features respectively. This similarity measure is specifically designed for neuron morphological data. In practice, the representative weights are determined by the neuronal tree-topological structure, and we will discuss it in the experiment.

Finally, with the newly defined similarity measure, we re-rank all the unlabeled neurons in ascending order based on their results in Eq. (6), and present the updated top-z results to users. Users can iteratively give relevance feedback for these z neurons until they are satisfied with the retrieval results.

3.4. Implementation details

Given the query neuron \mathbf{x}_q , and the continuously expanding neuron database $\{\mathbf{B}_0, \mathbf{B}_1, \dots, \mathbf{B}_k\}$, our neuron retrieval method can efficiently obtain the similar neurons based on online binary coding and interactive feedback. We outlined the framework in Algorithm 1.

Algorithm 1. Neuron Retrieval based on Online Binary Coding and Interactive Feedback.

Input: Continuously added neuron database $\{\mathbf{B}_0, \mathbf{B}_1, \dots, \mathbf{B}_k\}$;
 Query neuron \mathbf{x}_q .
Output: top-S retrieved neurons.
 1: Sketch $\mathbf{B}_0 - \bar{\mathbf{B}}_0$ into \mathbf{Y}_0 ;
 2: Initialize data size $n_0 = m_0$, mean value $\bar{\mathbf{X}}_0 = \bar{\mathbf{B}}_0$;
 3: **for** $i = 1 \rightarrow k$ **do**
 4: Sketch $\hat{\mathbf{B}}_i = [\mathbf{B}_i - \bar{\mathbf{B}}_i, \mathfrak{I}_i]$ into \mathbf{Y}_i , initialize by \mathbf{Y}_{i-1} ;
 5: Update \mathbf{W}_i through Eq. (2);
 6: Update data size $n_i = n_{i-1} + m_i$;
 7: Update mean value through Eq. (3);
 8: **end for**
 9: **if** retrieve \mathbf{x}_q is required **then**
 10: Binary encoding \mathbf{x}_q and $\{\mathbf{B}_0, \mathbf{B}_1, \dots, \mathbf{B}_i\}$ through \mathbf{W}_i ;
 11: Rank the Hamming distance in ascending order;
 12: $t = 0$;
 13: **while** users are not satisfied with the retrieval results **do**
 14: $t = t + 1$; (t -th round of user feedback)
 15: Compute $E^{(t)}$ of unlabeled neuron through Eq. (6);
 16: Re-rank the similarity based on $E^{(t)}$;
 17: **end while**
 18: Present the top-S retrieved neurons.
 19: **end if**

In the above neuron retrieval algorithm, we combine binary coding and interactive retrieval to efficiently obtain the fine-grained retrieval results. The binary coding part can efficiently tackle the large-scale and continuously expanding neuron databases, which update the coding function on-the-fly every time a new neuron batch is added. For a query neuron, this part provides the coarse retrieval results and significantly reduces the search scope from tens of thousands to a few hundred (e.g., providing neurons with top-Z minimum Hamming distance). In the interactive retrieval part, users can give relevance feedback for the top-z ranked neurons. Then the proposed algorithm will re-rank the similarity of unlabeled neurons based on the results of Eq. (6). Finally, the Z coarse neurons will be re-ranked repeatedly until users are

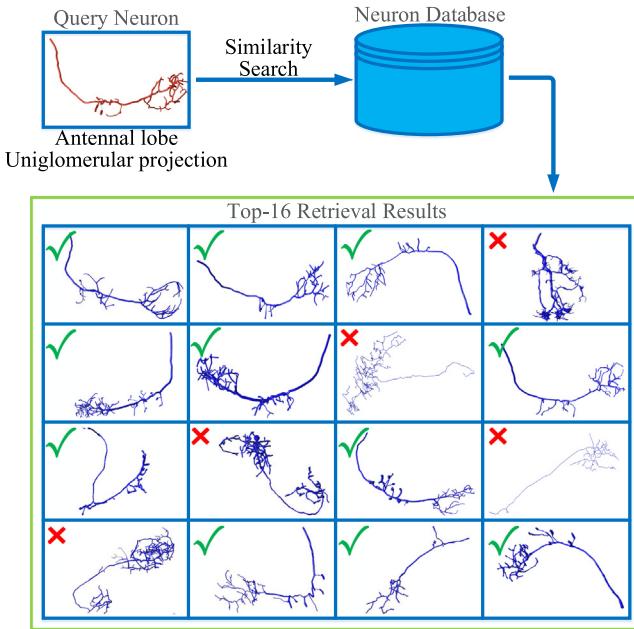


Fig. 3. An illustration of our user interaction interface, users will give feedback by one-click inputs.

satisfied with the retrieval performance, and we will provide the top- S retrieved neurons to biologists to assist the exploration and analysis of the query neuron.

4. Experiment

This section presents the evaluation of our framework for neuron retrieval. We first validate the performance for both the binary coding with online updating and interactive neuron retrieval. Then, we demonstrate an example of its use in neuron exploration and analysis.

4.1. Experimental setting

Our experiments are carried out on the NeuroMorpho [3], which has the largest collection of publicly accessible 3D reconstructed neuron data. Specifically, we use the entire 17,107 *Drosophila Melanogaster* neurons to evaluate the retrieval performance. Following the convention, we employ L-measure toolbox to extract 38 quantitative measurements as morphological features for each neuron [47], including 15 global, 10 branch and 13 bifurcation features respectively. All the experiments are conducted on a 3.6 GHz CPU with 4 cores and 32G RAM, in a MATLAB implementation.

To evaluate the retrieval performance, we select projection neurons as queries for which the brain region is the olfactory antennal lobe, and the cell classes are principal cell and uniglomerular projection (233 such projection neurons in total). We denote such projection neurons as uPNs. Despite that there are ten thousands of neurons in NeuroMorpho database, most of them are not well classified due to the lack of sufficient annotations, i.e., they are not identified in the finest level. We therefore select uPNs, since these are one of the best identified classes in the drosophila brain. This setting is also consistent with [9,49]. Then we evaluate the performance by computing the retrieval precision, which is defined as:

$$\text{Precision} = \frac{|\{uPNs\} \cap \{\text{Retrieved Neurons}\}|}{|\{\text{Retrieved Neurons}\}|} \quad (8)$$

where $|\cdot|$ denote to count the number of samples inside. In the experiments, we compute the average precisions obtained over all queries.

In the binary coding with online updating, due to the requirement of the FD algorithm [13], the sketched data size l should be no larger than the feature dimension d ($d = 38$ as discussed above). Since the feature dimension of neuron data is not high, we set $l = 38$ to preserve the information from the original database as much as possible. In the interactive neuron retrieval, we adopt an outer scope size of $Z = 300$ and an inner scope size of $z = 30$. The trade-off parameter λ is set as 0.3. For the defined similarity measure in Eq. (7), we assign three levels of features with different representation weights. Generally, global features can only represent neurons at a coarse level, and bifurcation features are so subtle that even neurons in same class are different. In practice, we empirically set global, branch and bifurcation features with the weights ratio of 1:2:1, which reflect their respective representative levels.

4.2. Evaluation of binary coding with online updating

In this experiment, we aim to demonstrate that the binary coding part can attain promising performance with the continuously expanding neuron database. We randomly split the 17,107 *Drosophila Melanogaster* neurons into two parts. The first 1,107 neurons are used as the original database, and the remaining 16,000 neurons are equally divided into 100 batches (160 neurons in each), which are sequentially added to simulate the expanding size of the neuron database. Our online binary coding method is compared with the batch-based method. The batch-based method corresponds to the first part in Section 3.2, i.e., matrix sketching based binary coding. For each update, it needs to learn the coding function from scratch, using all neurons in the database. To overcome randomness, we repeat the experiments a hundred times to report the average.

Table 1 presents the accumulated training time of our method and batch-based method at the 20th, 60th and 100th update rounds respectively. Compared with the batch-based method, our binary coding with online updating shows great superiority in computational efficiency, and the superiority becomes more obvious with more rounds of updates, e.g., 51.11s versus 1.88s for one hundred updates. When new neuron batches are added to the database, our method only need to consider these newly added neurons and update the coding function on-the-fly, while the batch-based method needs to take all the neurons into account to re-train the coding function. The merit of this online binary coding method is particularly beneficial in the future, since an increasing number of neurons are reconstructed and added to the databases through the recently well-developed neuron tracing techniques.

Besides the superiority in computational efficiency, our binary coding with online updating also demonstrates its comparable performance in retrieval precision. **Fig. 4(a)** shows the average retrieval precision of two competitive methods, taking their top-10 retrieved neurons into accounts. The learned coding function compact the feature vectors into 32 bits of binary codes in this experiment. According to **Fig. 4(a)**, our online method is able to achieve similar retrieval precision as the batch-based method. Therefore, the binary coding with online updating can significantly improve the computational efficiency without sacrificing the retrieval precision. This is mostly improved by (1) the employed FD algorithm,

Table 1
Training time comparison (in second).

Rounds	20	60	100
Batch-based	1.12	13.24	51.11
Ours	0.26	0.84	1.88

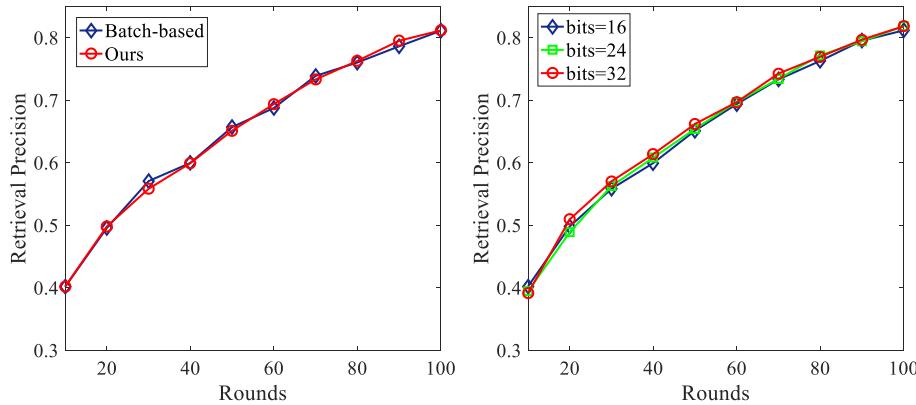


Fig. 4. Evaluation of the retrieval precision in 100 rounds of update: (a) Comparison of our method with the batch-based method; (b) Comparison of our method using different bits of binary codes.

which can compute the data sketch in a streaming manner; and (2) the newly introduced batch of data with virtual samples, which can overcome the mean-varying problem in a continuously expanding neuron database. Regarding the parameter, Fig. 4(b) shows the retrieval precision of our method when using 16, 24 and 32 bits of binary codes in each update. The online method can always achieve good performance using different bits of binary codes. These results verify that the online updated coding function can generate effective and representative binary codes for neuron morphological features.

4.3. Evaluation of interactive neuron retrieval

In this part, we aim to validate that the interactive strategy can actually achieve good performance for the neuron retrieval problem. We will re-fine the coarse retrieved neurons from previous binary coding results, where the neurons are retrieved from the entire 17,107 *Drosophila Melanogaster* neuron database after 100 rounds of database updating. Particularly, we only consider to refining neurons within top- Z minimum Hamming distance (neurons in outer scope size). For the user interaction, users will give feedback for the top- z unlabeled neurons (neurons in inner scope size).

To evaluate the performance of interactive neuron retrieval, at each feedback round, the top- z unlabeled neurons will be automatically labeled using the ground truth in order to simulate the user's feedback. Since our interactive method only requires users to give one-click inputs(relevant or non-relevant), the ground truth can be easily achieved by checking whether the unlabeled neurons are uPNs or not.

We compare our neuron retrieval method with three state-of-the-art methods, i.e., ITQ [15], AGH [29] and MIPS [24], which were all proposed to tackle the retrieval problem for large-scale databases. ITQ [15] is a very effective binary coding method for most natural image retrieval problems. AGH [29] has already achieved excellent retrieval performance in mammogram data [28], and MIPS is specially designed for the neuron morphological retrieval problem. The above three methods are batch based method, which can not process the continuously expanding neuron databases. Thus, for fair comparison, the retrieval model of these methods are trained through the currently entire 17,107 *Drosophila Melanogaster* neuron database. As with the previous experiment setting, we employ the 233 uPNs as queries to validate the retrieval performance.

Table 2 reports the average retrieval precision of four competitive methods under different number of retrieved neurons. For our

Table 2
Retrieval precision of four methods under different number of retrieved neurons.

	top20	top30	top40	top50
ITQ [15]	0.7673	0.7249	0.6948	0.6614
AGH [29]	0.7589	0.7216	0.6951	0.6735
MIPS [24]	0.7923	0.7508	0.7088	0.6828
Ours	0.9015	0.8550	0.7888	0.7092

interactive method, the retrieval precision is recorded after 3 rounds of feedback. According to Table 2, our method can achieve the highest precision under different number of retrieved neurons. These results verify the proposed method is effective for the neuron retrieval problem. It mostly benefits from the interactive strategy which introduces users in the loop to give feedback for the coarse retrieval results. Specifically, based on the user's feedback, our method can re-rank unlabeled neurons by the newly designed similarity measure.

We randomly select a query neuron and present its top-20 retrieval results in Fig. 5 under different rounds of feedback. We employ Vaa3D [38] software to display these neurons. The neurons with green frames are relevant to the query, and neurons with red frames are not relevant to the query. Generally, the retrieval performance is improved greatly from coarse results to the results after user feedback, which verifies the effectiveness of the proposed interactive strategy. We also find that with the increased numbers of feedback rounds, the retrieval performance improves accordingly. This is because of the increasingly labeled neurons, providing more information for re-ranking. In addition, according to Fig. 5, many non-relevant neurons also present similar morphologies with the query, which is usually hard to distinguish through traditional retrieval methods. Thus, our interactive strategy is a good choice for the fine-grained neuron retrieval problem.

In our interactive neuron retrieval, two parameters may influence the final performance, i.e., the outer scope size Z and the inner scope size z . In the interactive part, we only consider the refinement of top- Z ranked neurons from coarse retrieval results. Fig. 6(a) shows the average retrieval precision with different outer scope sizes after the 1st to 10th feedback rounds, taking top-30 retrieved neurons into account. In Fig. 6(a), we find that with the outer scope size ranging from 200 to 500, the retrieval precision has not changed too much. This is because the majority of relevant neurons are already included in a small sized outer scope (e.g., $Z = 300$). A larger outer scope may include more relevant neurons, but non-relevant and noisy neurons are also included which will

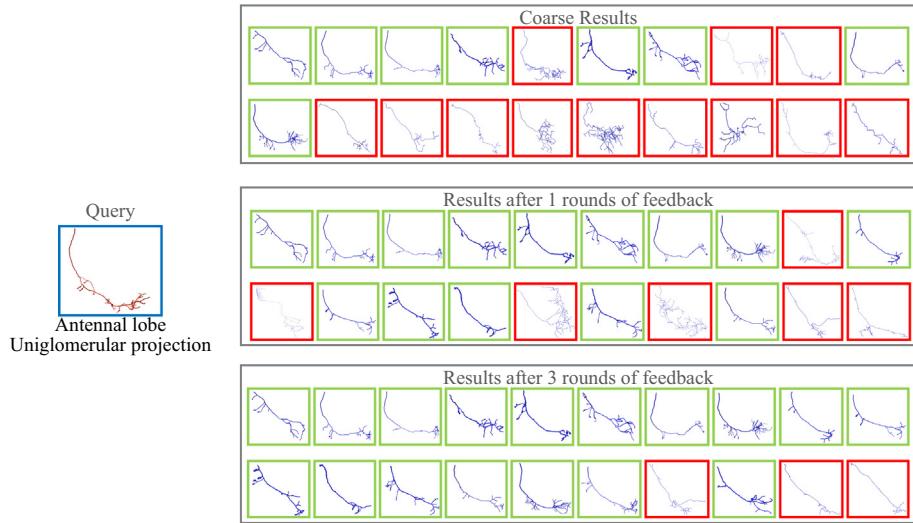


Fig. 5. Query example of the proposed method under different rounds of feedback: green framed neurons were relevant with the query, while red framed were non-relevant neurons.

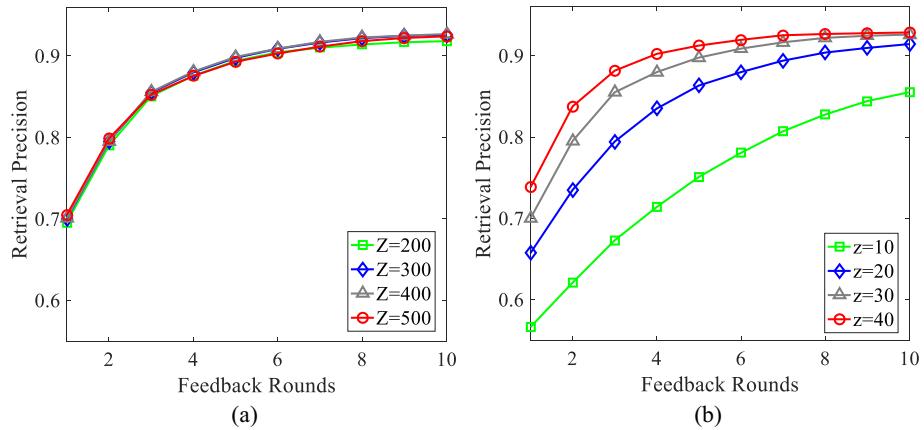


Fig. 6. Retrieval performance with different parameter settings: (a) retrieval precision with different outer scope size after 1st to 10th feedback rounds; (b) retrieval precision with different inner scope size after 1st to 10th feedback rounds.

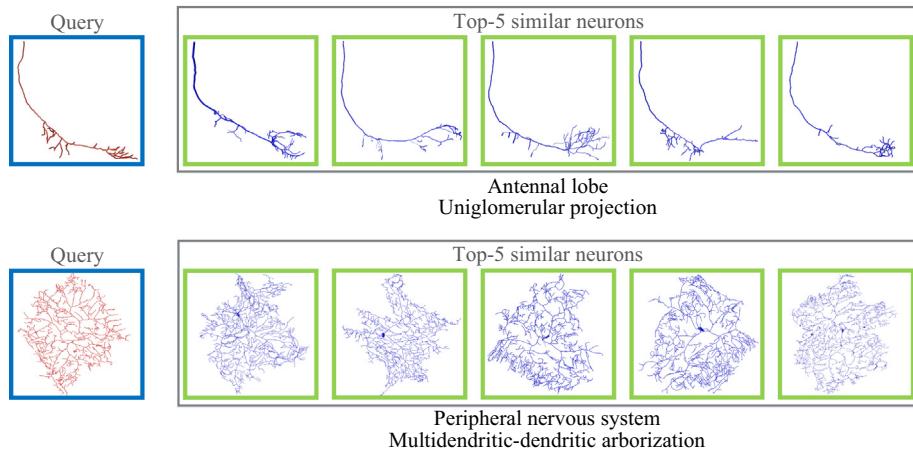


Fig. 7. Illustration of two unknown neurons and their top-5 retrieved neurons through the proposed method.

influence the retrieval performance. Moreover, the inner scope size z is the number of neurons we provide to users in each feedback round, which can also influence the final performance. **Fig. 6(b)**

presents the retrieval precision with different inner scope sizes after the 1st to 10th feedback rounds. According to **Fig. 6(b)**, the larger z can achieve better performance compared with smaller z .

values. This is easy to understand since a larger z will contain more feedback information which can help us to re-rank the coarse retrieved neurons.

4.4. Use cases and discussions

One important use case of our framework is the exploration and analysis for unknown neurons. Currently, despite the fact that an increasing number of neurons are reconstructed and added to the public databases, most of them are not well identified and lack basic annotations, such as cell classes and brain regions. Nanda et al. [34] proposed to annotate brain regions and cell classes for the NeuroMorpho [3] database. They employed the text-based query tool to search neurons with given lengths (e.g., 10, 20 microns) in each region to determine their brain regions, then they identified cell classes based on the brain regions invaded by the neurite terminals of every neuron. This method may be inefficient and unreliable, which the annotations are mainly obtained empirically. Identity of unknown neurons is an urgent demand in current neuron repositories.

Considering that neuron morphologies are associated with their properties, and our neuron retrieval framework can search similar neurons at a fine-grained level. It is reasonable to employ our framework to conduct neuron exploration via examining retrieved neurons, which have similar morphologies. To demonstrate this, we randomly select two query neurons from NeuroMorpho [3], whose neuron types are assumed to be unknown. After running our neuron retrieval framework (retrieve the entire 17,107 *Drosophila Melanogaster* neurons, with 1 rounds of feedback), Fig. 7 illustrates their top-5 similar neurons respectively. For the two query neurons, we find that their corresponding top-5 similar neurons all have the same neuron types, i.e., in NeuroMorpho [3], the 5 neurons in first row are annotated as antennal lobe and uniglomerular projection (uPNs), the 5 neurons in the second row are annotated as peripheral nervous system and multidendritic-dendritic arborization. Therefore, we can infer that the two query neurons also have the same type with their top-5 retrieved neurons. The information provided in NeuroMorpho [3] also verifies our inference about the two query neurons. In practical situations, we can employ more retrieved neurons (e.g., top-30 similar neurons) to statistically identify and analyze query neurons.

The proposed method can efficiently achieve the above neuron retrieval task, as it is designed for the exploration of large-scale neuron databases. The method will be particularly suitable in the future since big data is one major direction in neuroscience [1]. Besides the efficiency, for some specific neuron databases which are not very large (e.g., considering neurons in some specific brain regions with only hundreds of neurons), exhaustive search and comparison can be applied to achieve more accurate results. In addition, extracting more representative features for the 3D neuro-morphological data will be also helpful to improve the neuron retrieval performance.

5. Conclusions

In this paper, we present a novel framework for neuron exploration and analysis, which interactively retrieves similar neurons in the continuously expanding neuron databases. Specifically, our framework achieves neuron morphological retrieval from coarse to fine-grained levels. In the coarse level, we introduce binary coding with online updating to tackle the large-scale and continuously expanding neuron databases. In each database update, coding functions are learned on-the-fly by only considering the newly added neuron data, and the coarse retrieval results are subse-

quently obtained in real-time. In the fine-grained level, we bring users in the loop, which interactively gives relevance feedback for the coarse results. By processing the feedback and re-ranking the coarse neurons, our framework finally obtains a set of fine-grained retrieval results. Experiments verify the efficacy of our neuron retrieval framework and also illustrates its application in neuron exploration. Based on the present work, we will develop a comprehensive tool for efficient and accurate neuron retrieval, which can help biologists to explore and analyze unknown neurons.

Acknowledgement

We would like to acknowledge support for this project from the National Science Foundation (NSF Grant #1629913.).

References

- [1] A. Bigneuron project, <http://www.alleninstitute.org/bigneuron/>, (Accessed June 28, 2016).
- [2] B. Bigneuron released data, <https://github.com/BigNeuron/Data/releases>, (Accessed June 28, 2016).
- [3] Neuromorpho repository, <http://neuromorpho.org/>, (Accessed June 28, 2016).
- [4] R. Armañanzas, G.A. Ascoli, Towards the automatic classification of neurons, *Trends Neurosci.* 38 (2015) 307–318.
- [5] H. Chen, H. Xiao, T. Liu, H. Peng, Smarttracing: self-learning-based neuron reconstruction, *Brain Inf.* 2 (2015) 135–144.
- [6] S. Conjeti, A. Katouzian, A. Kazi, S. Mesbah, D. Beymer, T.F. Syeda-Mahmood, N. Navab, Metric hashing forests, *Med. Image Anal.* 34 (2016) 13–29.
- [7] S. Conjeti, S. Mesbah, M. Negahdar, P.L. Rautenberg, S. Zhang, N. Navab, A. Katouzian, Neuron-miner: an advanced tool for morphological search and retrieval in neuroscientific image databases, *Neuroinformatics* 14 (2016) 369–385.
- [8] L.D.F. Costa, K. Zawadzki, M. Miazaki, M.P. Viana, S.N. Taraskin, Unveiling the neuromorphological space, *Front. Comput. Neurosci.* 4 (2010) 150–163.
- [9] M. Costa, A.D. Ostrovsky, J.D. Manton, S. Prohaska, G.S. Jefferis, Nblast: rapid, sensitive comparison of neuronal structure and construction of neuron family databases, *Neuron* 91 (2014) 293–311.
- [10] A. Fakhry, H. Peng, S. Ji, Deep models for brain em image segmentation: novel insights and improved performance, *Bioinformatics* 32 (2016) 2352–2358.
- [11] D. Feng, C. Lau, L. Ng, Y. Li, L. Kuan, S.M. Sunkin, C. Dang, M. Hawrylycz, Exploration and visualization of connectivity in the adult mouse brain, *Methods* 73 (2015) 90–97.
- [12] M. Ghashami, A. Desai, J.M. Phillips, Improved practical matrix sketching with guarantees, *Eur. Symp. Algorithms* (2014) 467–479.
- [13] M. Ghashami, E. Liberty, J.M. Phillips, D.P. Woodruff, Frequent directions: Simple and deterministic matrix sketching, *ArXiv Preprint:1501.01711*, 1–28, 2015.
- [14] A. Gionis, P. Indyk, R. Motwani, Similarity search in high dimensions via hashing, *VLDB* (1999) 518–529.
- [15] 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.
- [16] S. Gulyanov, N. Sharifai, S. Bleylehman, E. Kelly, M. Kim, A. Chiba, G. Tsechpenakis, Three-dimensional neurite tracing under globally varying contrast, *ISBI* (2015) 875–879.
- [17] J. He, M. Li, H.J. Zhang, H. Tong, C. Zhang, Manifold-ranking based image retrieval, *ACM MM* (2004) 9–16.
- [18] S. Ji, Computational genetic neuroanatomy of the developing mouse brain: dimensionality reduction, visualization, and clustering, *Bioinformatics* 14 (2013) 222–236.
- [19] W. Kong, W.J. Li, Isotropic hashing, *NIPS* (2012) 1646–1654.
- [20] B. Kulis, P. Jain, K. Grauman, Fast similarity search for learned metrics, *IEEE Trans. Pattern Anal. Mach. Intell.* 31 (2009) 2143–2157.
- [21] B. Kulis, P. Jain, K. Grauman, Content-based image retrieval with relevance feedback using random walks, *Pattern Recogn.* 44 (2011) 2109–2122.
- [22] C. Leng, J. Wu, J. Cheng, X. Bai, H. Lu, Online sketching hashing, *CVPR* (2015) 2503–2511.
- [23] Z. Li, R. Fang, F. Shen, A. Katouzian, S. Zhang, Indexing and mining large-scale neuron databases using maximum inner product search, *Pattern Recogn.* 63 (2017) 680–688.
- [24] Z. Li, F. Shen, R. Fang, S. Conjeti, A. Katouzian, S. Zhang, Maximum inner product search for morphological retrieval of large-scale neuron data, *ISBI* (2016) 602–606.
- [25] E. Liberty, Simple and deterministic matrix sketching, *SIGKDD* (2013) 581–588.
- [26] G. Lin, C. Shen, Q. Shi, A. Hengel, D. Suter, Fast supervised hashing with decision trees for high-dimensional data, *CVPR* (2013) 1971–1978.
- [27] K. Lin, H.F. Yang, J.H. Hsiao, C.S. Chen, Deep learning of binary hash codes for fast image retrieval, *CVPR* (2015) 27–35.

[28] J. Liu, S. Zhang, W. Liu, X. Zhang, D.N. Metaxas, Scalable mammogram retrieval using anchor graph hashing, *ISBI* (2014) 898–901.

[29] W. Liu, J. Wang, S. Kumar, S.F. Chang, Hashing with graphs, *ICML* (2011) 1–8.

[30] D. Markonis, R. Schaer, H. Müller, Evaluating multimodal relevance feedback techniques for medical image retrieval, *Inf. Retrieval J.* 19 (2016) 100–112.

[31] E. Meijering, Neuron tracing in perspective, *Cytometry Part A* 77 (2010) 693–704.

[32] S. Mesbah, S. Conjeti, A. Kumaraswamy, P. Rautenberg, N. Navab, A. Katouzian, Hashing forests for morphological search and retrieval in neuroscientific image databases, *MICCAI* (2015) 135–143.

[33] S. Mukherjee, B. Condron, S.T. Acton, Tubularity flow field technique for automatic neuron segmentation, *IEEE Trans. Image Process.* 24 (2015) 374–389.

[34] S. Nanda, M. Allaham, M. Bergamino, S. Polavaram, R. Armañanzas, G. Ascoli, R. Parekh, Doubling up on the fly: neuromorpho.org meets big data, *Neuroinformatics* 1 (2015) 127–129.

[35] M. Norouzi, D.M. Blei, Minimal loss hashing for compact binary codes, *ICML* (2011) 353–360.

[36] M. Norouzi, D.M. Blei, R. Salakhutdinov, Hamming distance metric learning, *NIPS* (2012) 1061–1069.

[37] H. Peng, M. Hawrylycz, J. Roskams, S. Hill, N. Spruston, E. Meijering, G.A. Ascoli, Bigneuron: large-scale 3d neuron reconstruction from optical microscopy images, *Neuron* 87 (2015) 252–256.

[38] 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.

[39] L. Peter, D. Mateus, P. Chatelain, N. Schworm, S. Stangl, G. Multhoff, N. Navab, Leveraging random forests for interactive exploration of large histological images, *MICCAI* (2014) 1–8.

[40] S. Polavaram, T.A. Gillette, R. Parekh, G.A. Ascoli, Statistical analysis and data mining of digital reconstructions of dendritic morphologies, *Front. Neuroanat.* 8 (2014) 1–16.

[41] M. Raginsky, S. Lazebnik, Locality-sensitive binary codes from shift-invariant kernels, *NIPS* (2009) 1509–1517.

[42] E. Rebolledo, K. Karkali, F. Mangione, E. Martín-Blanco, Live imaging in drosophila: the optical and genetic toolkits, *Methods* 68 (2014) 48–59.

[43] D.A. Ross, J. Lim, R.S. Lin, M.H. Yang, Incremental learning for robust visual tracking, *Int. J. Comput. Vision* 77 (2008) 125–141.

[44] Y. Rui, T.S. Huang, S.F. Chang, Image retrieval: current techniques, promising directions, and open issues, *J. Vis. Commun. Image Represent.* 10 (1999) 39–62.

[45] Y. Rui, T.S. Huang, M. Ortega, S. Mehrotra, Relevance feedback: a power tool for interactive content-based image retrieval, *IEEE Trans. Circuits Syst. Video Technol.* 8 (1998) 644–655.

[46] H. Sahbi, J.Y. Audibert, R. Keriven, Graph-cut transducers for relevance feedback in content based image retrieval, *ICCV* (2007) 1–8.

[47] 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.

[48] F. Shen, W. Liu, S. Zhang, Y. Yang, H. Tao Shen, Learning binary codes for maximum inner product search, *ICCV* (2015) 4148–4156.

[49] 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* (2015) 1–13.

[50] J. Wang, S. Kumar, S.F. Chang, Semi-supervised hashing for large-scale search, *IEEE Trans. Pattern Anal. Mach. Intell.* 34 (2012) 2393–2406.

[51] Y. Weiss, A. Torralba, R. Fergus, Spectral hashing, *NIPS* (2009) 1753–1760.

[52] G. Wu, H. Jia, Q. Wang, D. Shen, Sharpmean: groupwise registration guided by sharp mean image and tree-based registration, *NeuroImage* 56 (2011) 1968–1981.

[53] L. Zhang, L. Wang, W. Lin, Semisupervised biased maximum margin analysis for interactive image retrieval, *IEEE Trans. Image Process.* 21 (2012) 2294–2308.

[54] W. Zhang, R. Li, H. Deng, L. Wang, W. Lin, S. Ji, D. Shen, Deep convolutional neural networks for multi-modality isointense infant brain image segmentation, *NeuroImage* 108 (2015) 214–224.

[55] 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 Inf.* 20 (2016) 1377–1383.

[56] 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.

[57] X. Zhang, H. Su, L. Yang, S. Zhang, Fine-grained histopathological image analysis via robust segmentation and large-scale retrieval, *CVPR* (2015) 5361–5368.

[58] 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.

[59] X. Zhang, L. Yang, W. Liu, H. Su, S. Zhang, Mining histopathological images via composite hashing and online learning, *MICCAI* (2014) 479–486.

[60] X.S. Zhou, T.S. Huang, Relevance feedback in image retrieval: a comprehensive review, *Multimedia Syst.* 8 (2003) 536–544.

[61] Z. Zhou, X. Liu, B. Long, H. Peng, Tremap: automatic 3d neuron reconstruction based on tracing, reverse mapping and assembling of 2d projections, *Neuroinformatics* (2015) 1–10.

[62] Z. Zhou, S. Sorensen, H. Zeng, M. Hawrylycz, H. Peng, Adaptive image enhancement for tracing 3d morphologies of neurons and brain vasculatures, *Neuroinformatics* (2014) 1–14.