

Random-Forest-Based Automated Cell Detection in Knife-Edge Scanning Microscope Rat Nissl Data

Shashwat Lal Das

Department of Computer Science
and Engineering
Texas A&M University
sldas@tamu.edu

John Keyser

Department of Computer Science
and Engineering
Texas A&M University
Email: keyser@cse.tamu.edu

Yoonsuck Choe

Department of Computer Science
and Engineering
Texas A&M University
Email: choe@tamu.edu

Abstract—Rapid advances in high-resolution, high-throughput 3D microscopy techniques in the past decade have opened up new avenues for brain research. One such technique developed in our lab is called the Knife-Edge Scanning Microscopy (KESM). The basic principle of KESM is to line-scan image while simultaneously sectioning thin tissue blocks using a diamond microtome. We have successfully sectioned and imaged whole mouse brains and portions of a rat brain processed with different stains to investigate the microstructures within. In this paper, we will present a fully automated soma (cell body) detection method based on random forests, working on Nissl-stained rat brain specimen. The method enables fast and accurate cell counting and density measurement in different brain regions.

I. INTRODUCTION

In this paper, we present a processing pipeline for fast and reliable detection of cell centers in high-resolution, 3D rat brain Nissl data. The detected cell centers can be used as the starting point of many further analysis tasks such as counting cells (cf. stereology [1]), studying their spatial distributions, and cell segmentation.

The kind of high-resolution 3D data that we used in this paper are enabled by advances in high-resolution high-throughput 3D microscopy techniques. These techniques are able to image whole small animal brains at sub-micrometer resolution [2], [3], [4], [5], [6], [7], [8], [9], [10] (see [11] for a review). Depending on the labeling method used, neurons, cell bodies, vasculature networks, etc. can be observed through these techniques.

One of the first such instruments that enabled high-throughput, high-resolution imaging of whole brains is the Knife-Edge Scanning Microscope (KESM) developed in our lab [6], [7], [12], [2], [4], [5] (Fig. 1). The KESM is an instrument for simultaneously sectioning and imaging fixed and stained tissue embedded in plastic. The machine uses a computer controlled three-axis stage to move the specimen underneath a fixed knife, and a high-speed camera captures images of the specimen as it is sectioned. The process enables the capture of fully registered 3D stacks of optical images at sub-micrometer resolution. Compared to processes providing similar resolution, the KESM scanning is very fast, and enables the capture of an entire small animal brain in a few days. Using the KESM, we have been able to image the whole mouse brain and part of the rat cortex stained in Nissl which labels the cell bodies (Fig. 2) [4], [12]. The resolution of each voxel was $0.6 \mu\text{m} \times 0.7 \mu\text{m} \times 1.0 \mu\text{m}$.

In this paper, we present a processing pipeline based on random forests for the analysis of the KESM rat Nissl data. First, a small number of samples are manually labeled to serve as ground truth. Next, a series of random forest classifiers are used to narrow down the cell-center candidate voxels. Finally, the candidate voxels are put through a refinement stage (using mean shift clustering) so that one voxel per cell is identified as the cell center. The main novelty is the custom feature windows used for rapid, successive refinement of the results. Experiments on 6 data sets with each containing ~ 200 cells resulted in F1 score of over 0.90, demonstrating the effectiveness of our approach.

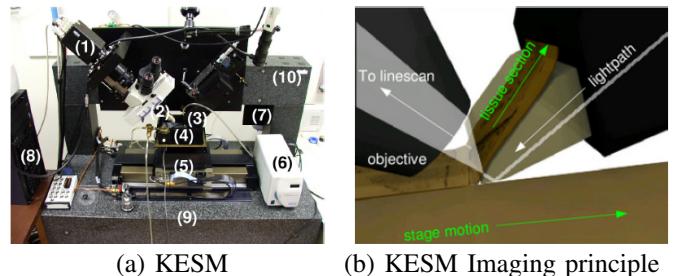


Fig. 1: **Knife-Edge Scanning Microscope.** (a) A photo of the KESM with its major components: (1) high-speed line-scan camera, (2) microscope objective, (3) diamond knife assembly and light collimator, (4) specimen tank (for water immersion imaging), (5) three-axis precision air-bearing stage, (6) white-light microscope illuminator, (7) water pump (in the back) for the removal of sectioned tissue, (8) PC server for stage control and image acquisition, (9) granite base, and (10) granite bridge. (b) KESM imaging principle. [6].

II. BACKGROUND AND RELATED WORKS

A wide range of automated and semi-automated techniques for detecting cells in images can be found in the literature. Cell detection typically forms a part of the cell segmentation process which attempts to determine the number of cells, their spatial arrangement, shapes and extents. Due to the considerable variation in images obtained from different sources, no single method has become dominant either for detection or segmentation.

The primary goal of cell segmentation is to extract the contours of cells in an image. A cell detection process can involve segmentation as a first step, following which the

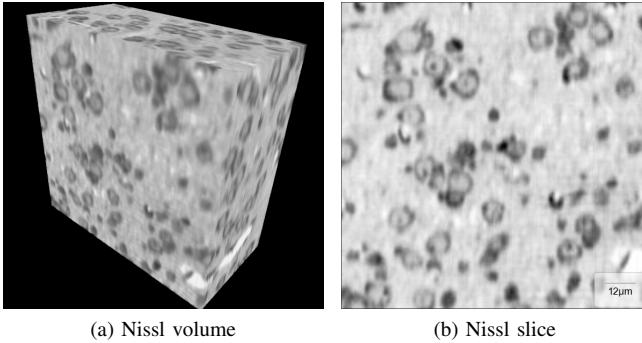


Fig. 2: KESM rat brain Nissl data. (a) A 200 voxels \times 200 voxels \times 100 voxels volume with voxel size $0.6 \mu\text{m} \times 0.7 \mu\text{m} \times 1.0 \mu\text{m}$. (b) A single 200×200 image slice from the volume.

centers of the segmented cells can be returned as the detection results. Numerous segmentation methods have been developed for medical images, and are reviewed in [13], [14], [15], [16]. Approaches include simple image analysis techniques such as thresholding and morphological operations; region segmentation techniques like the watershed algorithm, clustering, and region growing; methods based on deformable models such as active contours and level-sets; ideas from pattern recognition such as artificial neural networks; and graph search techniques, most prominently those using graph cuts.

Much of the early work in cell detection employed image analysis techniques such as finding the peaks in a Euclidean distance map of a binarized image [17]. More recently, Al-Kofahi et al. [18] detected nuclei using the multiscale Laplacian of Gaussian (LoG) where the distance map was used for automatic scale selection. The Laplacian of Gaussian is an operation that produces large responses at blob-like structures in an image, and is used extensively for blob detection. Its scale parameter determines the blob sizes it can detect. Al-Kofahi et al. found shape and size cues from the distance map to aid in selecting the scale parameter and detecting nuclei in heterogeneous clusters.

Binarization is a key step in the above methods that use the distance map and in many others [18], [19], [20], [21], [22]. The binary image must accurately separate the cells from the background and preserve their outlines. Unfortunately, this is a hard task for low-contrast images and can require sophisticated preprocessing steps to be designed for the particular data set. In the data considered in this paper, cell bodies and the background have similar intensity values which makes binarization difficult.

Algorithms which detect interest points based on shape and symmetry have also been applied to cell detection. The Hough transform can detect parametric curves such as circles and ellipses and has been used for nuclei detection [23], but is computationally expensive. Radial symmetry [24] of nuclei was used to detect them in [25] and [26]. These methods need several parameters to be specified for optimal results.

Pattern recognition is gaining a more prominent role in cell detection. D’Souza used template matching followed by

refinement as part of a routine for cell detection in the KESM Nissl-stained rat brain data [27]. A 3-D template of a typical cell with size $25 \times 25 \times 7$ voxels was applied on all voxels in a subvolume. The maxima of the matching scores indicated likely locations of cells. Template matching suffers when target objects vary significantly in appearance or size. Matching multiple templates or large 3-D templates can make results more accurate but significantly slow down the process.

Artificial neural networks (ANNs) were used in [28] for detecting cells in the KESM Nissl data. Principal component analysis (PCA) was performed on intensities in orthogonal cross sections of each voxel to form input vectors with 20 features. Similar features are used in the proposed method, but without the PCA dimensionality reduction. The detection is performed on a $2X$ downsampled input volume. This method could achieve a recall of 82.4% and precision of 92.8%, surpassing the performance of LoG blob detection and the spherical Hough transform (see Table 1).

A similar set of features was used in [29], where the reconstruction error of an input voxel from PCA basis vectors determines whether it is classified as a cell center or off-center voxel. Han et al. [30] used support vector machines (SVMs) to detect nuclei in 2-D cell culture images. They used intensity and gradient information in 20×20 pixel image patches as features for an SVM classifier, achieving over 90% accuracy. Other machine learning approaches such as convolutional neural networks have also demonstrated excellent results in cell detection [31], making this a very promising avenue for exploration.

III. METHODS

We used a cascade of random forests to perform preliminary cell-center candidate detection. Mean shift clustering was applied afterwards for the refinement of the results.

A. Data

The data under consideration are sampled from the somatosensory cortex of a rat brain. Voxel size is the standard KESM resolution: $0.6 \mu\text{m} \times 0.7 \mu\text{m} \times 1.0 \mu\text{m}$. Cell bodies of neurons (3 shows an example) appear as irregular elliptical structures with a dark boundary and a lighter interior region. A nucleolus is typically visible as a dark speckle near the center of a cell body. In this data set, the cell bodies have a diameter of around 20–27 voxels in the x and y dimensions and 10–15 voxels in the z dimension. The variation between the axes is due to the unequal physical dimensions of voxels along them.

Three volumes of size $200 \text{ voxels} \times 200 \text{ voxels} \times 100 \text{ voxels}$ were used as training data for the cascade. The volumes have complementary characteristics in terms of cell size and density. Fiji [32], an image analysis platform, was used to mark the centers of all cells in the three volumes. 396, 182, and 414 cell centers were marked in total, respectively.

To add robustness to orientation, more training data were derived by flipping the original volumes about the coordinate axes. Per volume, three derived volumes are obtained, corresponding to the three axes. The cell center coordinates were also transformed to match the new orientations. Twelve volumes were therefore used for training (three original volumes + 3×3 derived volumes).

B. Cell Center Candidate Detection: Cross-Section Extraction

For cell-center candidate detection, orthogonal cross-sections of a small subvolume around each voxel was used to classify it as a cell center or not. These cross-sections consist of three 2-D image windows or *patches* centered at that voxel (Fig. 3).

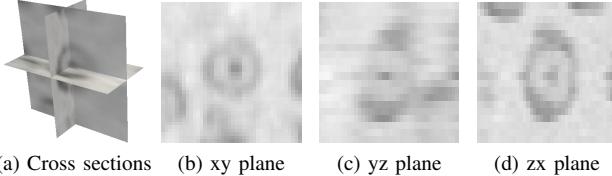


Fig. 3: Orthogonal cross sections of a cell. (a) Cross sections in 3D context. (b), (c), (d) Views of the cross sections.

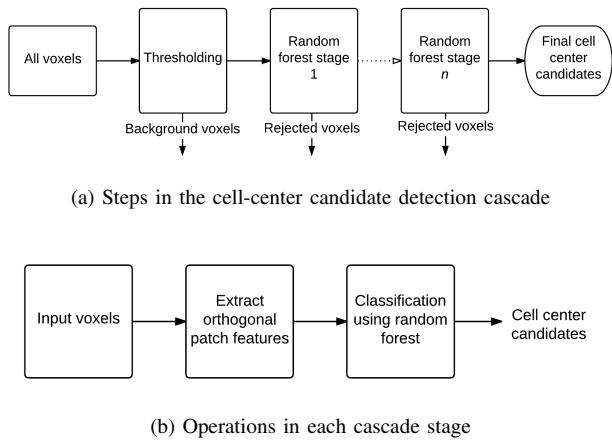


Fig. 4: Cascade used for detecting cell-center candidates. Each stage only operates on voxels accepted by its predecessor. The resulting cell center candidates are processed with mean-shift filters to obtain the final cell center locations.

Orthogonal cross sections are represented in this text as (l_x, l_y, l_z) where l_x , l_y , and l_z are the lengths along the x , y , and z axes. Without removing duplicates, the number of voxels in these cross sections is $l_x \times l_y + l_y \times l_z + l_z \times l_x$. The entire cuboid, on the other hand, would contain $l_x \times l_y \times l_z$ voxels.

Based on typical cell diameters mentioned earlier, a cell in the rat brain data would fit in a 3-D cuboidal window with total length of 27 voxels in the x and y dimensions and 15 voxels in the z dimension. This would require extracting $27 \times 27 \times 15 = 10935$ voxels for each voxel in the image. Using only the cross sections reduces this number to $27 \times 27 + 27 \times 15 + 15 \times 27 = 1539$ voxels without removing redundancies.

Intensities of voxels in these cross sections are used as features for classifiers in the cascade.

C. Cell Center Candidate Detection: Random Forest

To eliminate unnecessary computation, the proposed candidate detection approach uses small windows (Fig. 5a) to eliminate obvious portions of the image such as cell boundaries

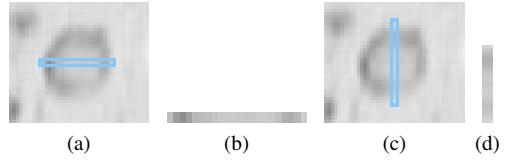


Fig. 5: Thin windows as features. A horizontal patch around the center is shown in (b) and a vertical patch in (d). They can be used to eliminate much of the background whose patches are largely uniform.

and vessels, and gradually increases the window size to process the remaining portions until a satisfactory set of cell center candidates are obtained. This is implemented as a cascade of classifiers.

Cascades of classifiers for object detection consist of a sequence of classifiers where each classifier updates the predictions made by its predecessors (see Fig. 4). Weak learners usually form the initial stages of the cascade. They quickly reject large fractions of the data using a small number of features while retaining all the possible matches. Later stages improve on their results by using more sophisticated and expensive features. Rather than computing expensive features such as large windows for all pixels in an image, they are computed only for those pixels that have passed through earlier stages.

Our approach uses a cascade of random forest classifiers. Random forests are predictors consisting of multiple decision trees whose predictions are combined to obtain the overall output [33]. Each tree in this ensemble is trained independently on a random sample of the data. At each node of a tree, a small random subset (of fixed size) of the features is used to split the node. The predictions of each tree are usually combined through averaging or by choosing the prediction with the most votes.

Each stage in the cascade receives as input a set of cell center candidates from the previous stage. The classifier at that stage eliminates less likely candidates from the set; the updated set of cell center candidates is then passed to the next stage and so on until a final set of candidates is obtained from the last stage. Windows used in this cascade start off small in the initial stages and grow until a maximum size is reached at the final stage. The cascade is structured as follows (see Fig. 7 for representative results).

1) Coarse Thresholding: The volume is eroded using a 3-D ball structuring element with radius $(2, 2, 1)$. Grayscale erosion darkens parts of the cell bodies and also regions surrounding the cells. The eroded image is downsampled by $2X$ and then smoothed using a median filter with radius $(3, 3, 2)$. This produces smooth cell bodies as dark as cell boundaries. This volume is then upsampled back to the original size and thresholded using Otsu's method [34].

2) Random Forest Stages: The fundamental choice for each cascade stage is the size of the window or patch it uses to classify an input voxel. The window must be at least as large as the largest expected cell, otherwise some cells may not be identified. Window dimensions of $27 \times 27 \times 15$ are likely to be

sufficient for identifying cell centers in this data. Cell bodies, however, can generally be detected by long, thin windows. A horizontal $27 \times 1 \times 1$ window around a cell center (5a) would run from the left cell boundary (dark), through the cell body (light), and up to the right boundary (dark). This dark-light-dark pattern (5b) would indicate the presence of a cell, although it cannot be used to localize the cell center vertically. Due to a cell's ring shape, a similar dark-light-dark pattern would occur along the vertical with a $1 \times 27 \times 1$ window (5d). Using such windows sequentially, each time filtering detections from the previous window, allows a cell center to eventually be localized accurately and more efficiently than using a large $27 \times 27 \times 15$ window.

Accordingly, cross sections of the following windows are used in the five cascade stages: i) $1 \times 27 \times 1$, ii) $27 \times 1 \times 1$, iii) $27 \times 27 \times 1$, iv) $27 \times 27 \times 5$, and v) $27 \times 27 \times 13$.

D. Training the Cascade

Cascades for object detection are trained with the emphasis that initial stages produce no false negatives (rejecting a foreground pixel) and the latter stages produce no false positives (accepting a background pixel). This requires stages to be tuned individually for best performance [35]. In the proposed method, a simpler approach is taken where the stages are trained uniformly.

Each stage in the cascade is biased to accept a voxel as a cell center rather than reject it. The expectation is that this will result in a larger number of voxels near cell centers being accepted. The refinement step is capable of handling this increase.

Biassing is done by sampling the training data so that cell centers are overrepresented (i.e., more false positives). The cell center class sampled consists of all the marked cell centers. Non-centers are sampled in three equal-sized chunks of voxels based on distance from the nearest center, d : i) $3 \leq d < 5$, ii) $5 \leq d < 10$, and iii) $10 \leq d < \infty$. The size of each chunk is determined by enforcing the total number of cell centers and the total number of non-cell centers sampled to be equal. Sampling is done independently for each training volume.

All stages are trained using the same data. A stage extracts patches for each training voxel according to its defined window size. If the window extends beyond the volume bounds, the voxel is not used for training by that stage. This can cause some variation in the training data between the stages. Stages are trained independently using the feature vectors that are formed by vectorizing the extracted patches.

Parameters of the random forest classifier are identical for all stages. 30 classification trees are grown with no depth limit for each forest. The maximum number of features used to split a tree node is $\sqrt{\text{number of features}}$, and 4 samples are required to split internal nodes. The random forest implementation used is `RandomForestClassifier` from scikit-learn [36].

The optional thresholding stage does not participate in training.

E. Testing the Cascade

The test volume presented to the cascade is first thresholded, and foreground voxels are sent to the next stage. Each

stage uses its trained random forest classifier to accept or reject a voxel that was accepted by its predecessor. Voxels whose windows would be outside the volume bounds are ignored. Voxels accepted as cell centers by the final stage are considered as the cell center candidates output by the cascade.

F. Cell Center Candidate Refinement Through Mean Shift Clustering

To reduce the cell center candidates generated from the cascade to one candidate per cell, a refinement step is required. The cascade in itself cannot reliably predict one candidate voxel per cell as:

- 1) The uniqueness condition is hard to directly embed into the classification problem, and
- 2) Identifying a unique voxel as a cell center needs at the same time its immediate neighbors to be rejected. The choice of features in this approach make this unlikely: windows around a voxel and those around its neighbors are too similar for a classifier to discriminate between them.

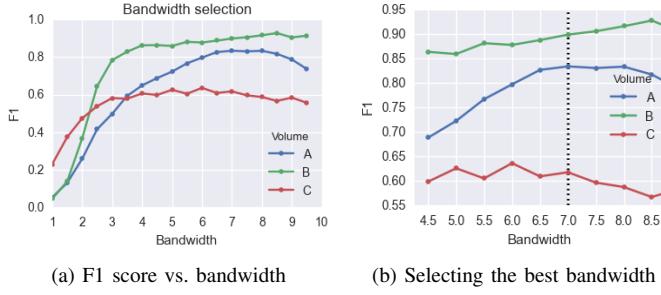
Candidates generated from the cascade form small clusters near cell centers. Cell centers can be localized by finding the centers of these k clusters. The refinement task is thus reduced to identifying these clusters and replacing them with their k centers.

As the number of clusters k is unknown, it must first be determined before using an algorithm such as k -means, or a clustering algorithm that does not require this knowledge must be used. The proposed refinement approach uses one such algorithm called *mean shift clustering* that has been successful in computer vision problems.

Mean shift [37] is a non-parametric density-based clustering procedure. It finds modes on the density surface of the data through steepest ascent. Density estimation is typically done by radially symmetric kernels such as flat kernels or Gaussian kernels [38]. These kernels are defined by a bandwidth parameter h . The choice of kernel determines how the density is computed at each data point. Points where the gradient of the estimated density is zero are modes of the density surface. Mean shift is a technique to find these modes and is inherently non-parametric, but requires the kernel to be specified. Essentially, it works by applying the kernel over each data point and moving it along the local gradient. This is done iteratively until convergence to points with zero gradient.

The cascade is trained on two of the three training volumes (A, B, and C) and is used to predict cell center candidates in the other. Mean shift is performed on these candidates by varying the mean shift kernel bandwidth from 1 to 9.5 in increments of 0.5. This is repeated with the roles of the training volumes interchanged. The value that leads to the best detection accuracy scores is used as the bandwidth for novel inputs.

Fig. 6 shows the detection accuracy with different choices of bandwidth for the three volumes. A bandwidth of 7 yields a good balance of performance for all the volumes. This value is chosen as the bandwidth for the detection method.



(a) F1 score vs. bandwidth

(b) Selecting the best bandwidth

Fig. 6: Mean shift bandwidth selection based on best F1 scores for training volumes. (a) F1 scores for all bandwidths considered. (b) Narrowed to bandwidths in [4.5, 9] where the highest F1 scores are achieved. A good value of bandwidth is 7 voxels.

G. Evaluation

The proposed cell detection method is evaluated on the basis of accuracy and speed. Speed is an essential metric as the eventual goal is to process large data sets, potentially terabytes in size, in reasonable time.

Detection accuracy is measured by assessing the agreement between the cell centers predicted by the algorithm and the cell centers labeled manually, i.e., the ground truth. In this work, this is done by computing the precision, recall, and F1 score, which are often-used accuracy measures for binary classification problems:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (1)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (2)$$

$$F_1 = \frac{2TP}{2TP + FP + FN}, \quad (3)$$

where TP = true positive, FP = false positive, and FN = false negative.

A match between the predicted cell center and ground truth is declared if the two are within 5 μm in Euclidean distance.

IV. RESULTS

Six volumes (volumes 1–6) of size about 200 voxels \times 200 voxels \times 100 voxels were used as the test data for the rat brain data set. In all, 1,540 cell centers were manually labeled in these volumes. The volumes differed substantially in terms of cell size, density, and presence of other structures like blood vessels.

A. Accuracy

The detection process can be tracked by visualizing the cell center candidates at each stage of the cascade. Figure 7 shows the stage-by-stage candidate filtering for a slice of volume 3. Thresholding (7b) rejected obvious background regions. The first random forest stage (7c) used a thin vertical window to filter out candidates from the foreground of the thresholded image. It identified regions near cell centers but, as expected,

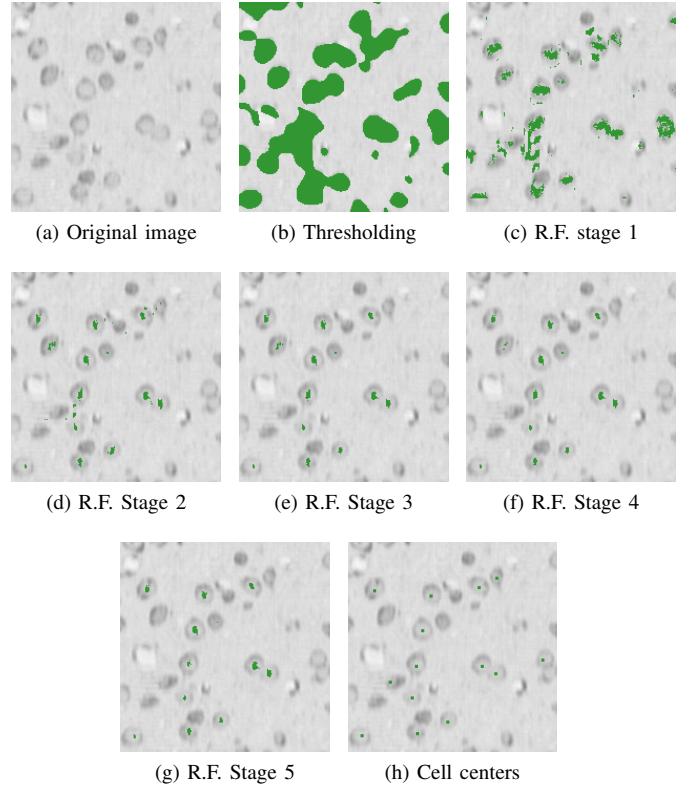


Fig. 7: Cell center candidates at each stage in the cascade for a slice of volume 3. Each stage improves the candidates (in green) output by the previous stage.

was unable to horizontally localize cell centers. By the final stage (7g), cell centers were localized accurately enough for mean shift to be applied. Figure 7h shows the cell centers found by applying mean shift to the output of the cascade. It includes centers detected within four slices above and below the given slice.

Fig. 8 shows blocks within the six volumes with the predicted cell centers. These also clearly show the differences in cell sizes and density across the volumes: volume 1 had large cells with low density; volume 2 had large cells with higher density; volume 3, volume 4, and volume 5 had moderately sized cells with high density; and volume 6 had moderately sized cells with lower density. Cell bodies in these figures (in red) were extracted using a simple smoothing and thresholding method and were therefore only crudely segmented, shown here for visualization purposes only. Large cells like those in volumes 1 and 2 appeared to contain holes and were particularly poorly segmented.

Table I shows a summary of the results, with a comparison to an approach that used artificial neural networks on a similar KESM Nissl data set (which included comparison with Laplacian-of-Gaussian-based and Hough-transform-based approaches as well) [28]. In all cases with cell count > 100 (5 out of 6 volumes we tested), our method gave F1 score > 0.90 , which is a good improvement over previous approaches (F1 = 0.873 [28] and other methods tested therein).

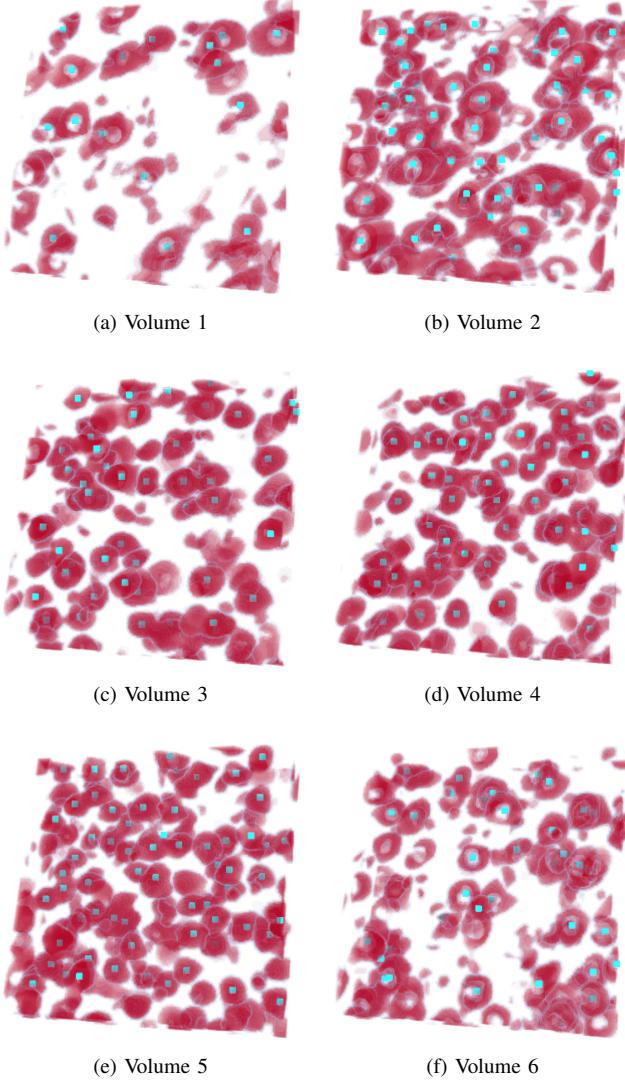


Fig. 8: Predicted cell centers in blocks of volumes 1 through 6.

B. Speed

Speed tests were performed on a laptop running 64-bit Arch Linux. The laptop featured a 2.5 GHz Intel Core i5-3210M processor with 2 physical (4 logical) cores and 3 MB L3 cache, and 6 GB DDR3 SDRAM. Random forests and feature extraction were both set to single-threaded operation.

The time taken for training the cascade for the above runs was between 10 s to 11 s, with a mean of 10.24 s.

Predicting cell centers on one volume of the test set ($200 \times 200 \times 100$, which is about 2 MB in size) using the cascade took between 11 s and 14 s, with a mean of 11.85 s.

Using multiple random forest or feature extraction threads can provide a further boost to speed. If the volumes are downsampled first (as done in [29], [28]), the speed can improve by an order of magnitude with a small possible loss of accuracy, making it comparable to the speed of the ANN

TABLE I: Detection Accuracy

Vol.	# Cells	Detec.	TP	FP	FN	Prec.	Recall	F1
1	78	91	73	18	5	0.802	0.936	0.864
2	170	179	159	20	11	0.888	0.935	0.911
3	161	153	146	7	15	0.954	0.907	0.930
4	220	224	200	24	20	0.893	0.909	0.901
5	228	208	203	5	25	0.976	0.890	0.931
6	138	134	127	7	11	0.948	0.920	0.934

Comparison with Mayerich et al. [28], based on 14 volumes								
ANN	2,158	N/A	N/A	N/A	N/A	0.928	0.824	0.873
LoG	2,158	N/A	N/A	N/A	N/A	0.659	0.934	0.773
Hough	2,158	N/A	N/A	N/A	N/A	0.726	0.754	0.739

ANN: Artificial Neural Network; LoG: Laplacian of Gaussian; Hough: Hough-transform

method in [28] which had a throughput of 2.19 MB/s using the CPU implementation (note that this is test time, not training time).

V. DISCUSSION

The main contribution of this paper is as follows. We developed a processing pipeline for fast and reliable detection of cell centers in high-resolution rat brain Nissl data. The detected cell centers can be used as the starting point of many further analysis tasks such as counting cells (cf. stereology [1]), studying their spatial distributions, and cell segmentation. Our method requires only a small amount of manual labeling and can process large volumetric data sets without further manual processing.

Simple extensions to the proposed detection methods can potentially lead to more accurate results and lower processing time. Key improvements that could be made are listed below.

(1) Cells larger than the window size are a major source of detection errors. One solution to achieve scale invariance is to apply the detection methods in multiple resolutions: the training stage remains the same, and in the testing stage the predictors operate on the original volume as well as a downsampled version. This ensures that cells fit inside the window at least in the downsampled volume.

(2) Although orthogonal patch features were successful for detecting cells, they are expensive to extract. This is less of a problem with the cascade method as larger patches are extracted only for a small fraction of the input. Haar-like features have been known to perform well in object detection [35] while being efficient to compute using integral images. Such features can provide speedups for the mouse brain data. Different kinds of features or higher order features should also be tested for improving accuracy while allowing for the window size to be decreased.

(3) The bandwidth parameter used for mean shift in the rat data detection method was fixed based on the training data. For test data with cell sizes and density significantly different than the training data, an automatic bandwidth selection method [39] would be preferable. The bandwidth selection step showed that different bandwidths are optimal for each of the volumes in the training set: 8.5 for volume A, 7.0 for volume B, and 6.0 for volume C. The choice of 7.0 as the bandwidth was a compromise, making the detection method suboptimal.

(4) Analysis of the results strongly suggests that more training data consisting of differently-sized cells, both small and large, can be of huge benefit. This can be done as an alternative to multiresolution detection once it is verified that cells of all sizes are included in the training set. A large number of training examples are likely not required; it is more important to make cells in the training data diverse than repetitive.

(5) Our proposed method requires multiple stages in the pipeline, so there can be a concern of slow down. However, as we have reported in the results section, the speed is comparable with an ANN-based implementation [28]. Furthermore, in our method training time is similar to test time, while methods based on ANN may have much longer training time. The speed of our method can be further improved by fine tuning of the code. For example, after each stage in the pipeline, we can keep an index of the detected foreground voxels so that subsequent computing is only done on or around these voxels. Furthermore, GPU-based or map-reduce-based parallelization can also greatly improve speed [28], [29].

(6) Finally, our ultimate goal is to use our approach to locate every cell center in our whole-brain KESM mouse data

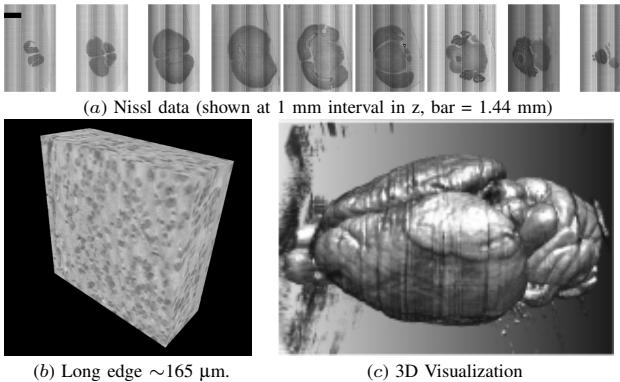


Fig. 9: KESM Mouse Nissl Data Set. Whole-brain Nissl data set from the KESM is shown. Adapted from [4], [12]. (a) Coronal sections spanning the entire brain, from anterior to posterior, at an interval of 1 mm, are shown. Voxel resolution = $0.6 \mu\text{m} \times 0.7 \mu\text{m} \times 1.0 \mu\text{m}$. (b) A $256 \times 256 \times 101$ voxel cube is shown ($\sim 165 \mu\text{m}$ on the long edge). This figure shows the 3D-nature of the KESM data set at its full resolution. (c) 3D visualization of the KESM Nissl data volume registered to the Waxholm space [40]. Fine details such as the folds in the cerebellum can be seen.

set (Fig. 9). Our preliminary work shows that the same basic approach can be used, but the details have to be adjusted to account for differences in data characteristics between rat and mice [41]. For example, compared to the rat data, the cells tend to be smaller in the mouse data. Our test run on the mouse data gave precision = 0.889, recall = 0.813, and F1 = 0.850 (1,662 cells to be detected).

VI. CONCLUSION

This research was aimed towards developing an accurate and fast automated cell detection method for KESM Nissl

images. A processing pipeline was developed, based on random forests. Training sets consisting of a small number of volumes with labeled cell centers were required. These data sets were used by each method to output the locations of the detected cell centers given novel volumes. The methods were evaluated by validating the detected cell centers on manually labeled test sets. Results were found to be accurate for data that the methods were designed and trained for. The method executed fast enough to be run on commodity hardware. The cascade architecture performed particularly well with high computational efficiency. Our method is expected to enable processing KESM data at large scales and ultimately obtain cell counts and locations in entire rat and mouse brains.

ACKNOWLEDGMENTS

KESM was designed by Bruce H. McCormick (Texas A&M University, US patent #US 6,744,572) and constructed by Bernard Mesa (Micro Star Tech). We would like to thank Jaerock Kwon (Kettering University) and David Mayerich (University of Houston) for data collection, Louise C. Abbott for specimen preparation, and Chul Sung (IBM) for Fig. 9c. This research has been partly funded by National Science Foundation (NSF) grants #1208174 and #0905041 (CRCNS program). Knife-Edge Scanning Microscope data acquisition was supported by National Institutes of Health/National Institute of Neurological Disorders and Stroke (NIH/NINDS) grant #1R01-NS54252 (MSM program) and NSF grant #0079874 (MRI program). Further development of KESM technology, in collaboration with 3Scan (Todd Huffman, Cody Daniels, Matthew Goodman; <http://3scan.com>), is currently funded by NSF grant #1256086 (IDBR program). The rat and mouse KESM Nissl data will be made available on our online atlas <http://kesm.org>, along with already available Golgi and India ink (vasculature) data. This paper is largely based on first author Shashwat Lal Das's thesis [41]. Shashwat Lal Das is now at Google.

REFERENCES

- [1] J. C. Fiala and K. M. Harris, "Extending unbiased stereology of brain ultrastructure to three-dimensional volumes," *J. Am. Med. Inform. Assoc.*, vol. 8, pp. 1–16, 2001.
- [2] Y. Choe, L. C. Abbott, D. Han, P.-S. Huang, J. Keyser, J. Kwon, D. Mayerich, Z. Melek, and B. H. McCormick, "Knife-edge scanning microscopy: High-throughput imaging and analysis of massive volumes of biological microstructures," in *High-Throughput Image Reconstruction and Analysis: Intelligent Microscopy Applications*, A. R. Rao and G. Cecchi, Eds. Boston, MA: Artech House, 2009, pp. 11–37.
- [3] Y. Choe, "Physical sectioning microscopy," in *Encyclopedia of Computational Neuroscience*, 1st ed., D. Jaeger and R. Jung, Eds. New York: Springer, 2014, dOI 10.1007/978-1-4614-7320-6_92-1 (4 pages).
- [4] Y. Choe, L. C. Abbott, D. E. Miller, D. Han, H.-F. Yang, J. R. Chung, C. Sung, D. Mayerich, J. Kwon, K. Micheva, and S. J. Smith, "Multiscale imaging, analysis, and integration of mouse brain networks," in *Neuroscience Meeting Planner, San Diego, CA: Society for Neuroscience*, 2010, program No. 516.3. Online.
- [5] Y. Choe, D. Mayerich, J. Kwon, D. E. Miller, J. R. Chung, C. Sung, J. Keyser, and L. C. Abbott, "Knife-edge scanning microscopy for connectomics research," in *Proceedings of the International Joint Conference on Neural Networks*. Piscataway, NJ: IEEE Press, 2011, pp. 2258–2265.
- [6] D. Mayerich, L. C. Abbott, and B. H. McCormick, "Knife-edge scanning microscopy for imaging and reconstruction of three-dimensional anatomical structures of the mouse brain," *Journal of Microscopy*, vol. 231, pp. 134–143, 2008.

- [7] J. R. Chung, C. Sung, D. Mayerich, J. Kwon, D. E. Miller, T. Huffman, L. C. Abbott, J. Keyser, and Y. Choe, "Multiscale exploration of mouse brain microstructures using the knife-edge scanning microscope brain atlas," *Frontiers in Neuroinformatics*, vol. 5, p. 29, 2011.
- [8] P. S. Tsai, B. Friedman, A. I. Ifarraguerri, B. D. Thompson, V. Lev-Ram, C. B. Schaffer, Q. Xiong, R. Y. Tsien, J. A. Squier, and D. Kleinfeld, "All-optical histology using ultrashort laser pulses," *Neuron*, vol. 39, pp. 27–41, 2003.
- [9] T. Ragan, L. R. Kadiri, K. U. Venkataraju, K. Bahlmann, J. Sutin, J. Taranda, I. Arganda-Carreras, Y. Kim, H. S. Seung, and P. Osten, "Serial two-photon tomography for automated *ex vivo* mouse brain imaging," *Nature Methods*, vol. 9, pp. 255–258, 2012.
- [10] P. J. Keller, A. D. Schmidt, J. Wittbrodt, , and E. H. K. Stelzer, "Reconstruction of zebrafish early embryonic development by scanned light sheet microscopy," *Science*, vol. 322, pp. 1065–1069, 2008.
- [11] B. A. Wilt, L. D. Burns, E. T. W. Ho, K. K. Ghosh, E. A. Mukamel, and M. J. Schnitzer, "Advances in light microscopy for neuroscience," *Annual Reviews of Neuroscience*, vol. 32, pp. 435–506, 2009.
- [12] Y. Choe, D. Mayerich, J. Kwon, D. E. Miller, C. Sung, J. R. Chung, T. Huffman, J. Keyser, and L. C. Abbott, "Specimen preparation, imaging, and analysis protocols for knife-edge scanning microscopy," *Journal of Visualized Experiments*, vol. 58, p. e3248, 2011, doi: 10.3791/3248.
- [13] J. Rogowska, "Chapter 5 - overview and fundamentals of medical image segmentation," in *Handbook of Medical Image Processing and Analysis (Second Edition)*, I. N. Bankman, Ed. Burlington: Academic Press, 2009, pp. 73–90.
- [14] N. Sharma and L. M. Aggarwal, "Automated medical image segmentation techniques," *Journal of Medical Physics / Association of Medical Physicists of India*, vol. 35, no. 1, pp. 3–14, 2010.
- [15] D. J. Withey and Z. Koles, "Medical image segmentation: Methods and software," in *Joint Meeting of the 6th International Symposium on Noninvasive Functional Source Imaging of the Brain and Heart and the International Conference on Functional Biomedical Imaging, 2007. NFSI-ICFBI 2007*, Oct. 2007, pp. 140–143.
- [16] E. Meijering, "Cell segmentation: 50 years down the road [life sciences]," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 140–145, Sep. 2012.
- [17] H. Irshad, A. Veillard, L. Roux, and D. Racoceanu, "Methods for nuclei detection, segmentation, and classification in digital histopathology: A review, current status and future potential," *Biomedical Engineering, IEEE Reviews in*, vol. 7, pp. 97–114, 2014.
- [18] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam, "Improved automatic detection and segmentation of cell nuclei in histopathology images," *IEEE transactions on bio-medical engineering*, vol. 57, no. 4, pp. 841–852, Apr. 2010.
- [19] N. Malpica, C. O. de Solzano, J. J. Vaquero, A. Santos, I. Vallcorba, J. M. Garca-Sagredo, and F. del Pozo, "Applying watershed algorithms to the segmentation of clustered nuclei," *Cytometry*, vol. 28, no. 4, pp. 289–297, Aug. 1997.
- [20] H. Ancin, B. Roysam, T. E. Dufresne, M. M. Chestnut, G. M. Ridder, D. H. Szarowski, and J. N. Turner, "Advances in automated 3-d image analyses of cell populations imaged by confocal microscopy," *Cytometry*, vol. 25, no. 3, pp. 221–234, Nov. 1996.
- [21] Q. Yang and B. Parvin, "Harmonic cut and regularized centroid transform for localization of subcellular structures," *IEEE transactions on bio-medical engineering*, vol. 50, no. 4, pp. 469–475, Apr. 2003.
- [22] J. Stegmaier, J. C. Otte, A. Kobitski, A. Bartschat, A. Garcia, G. U. Nienhaus, U. Strhle, and R. Mikut, "Fast segmentation of stained nuclei in terabyte-scale, time resolved 3d microscopy image stacks," *PLOS ONE*, vol. 9, no. 2, p. e90036, Feb. 2014.
- [23] E. Cosatto, M. Miller, H.-P. Graf, and J. Meyer, "Grading nuclear pleomorphism on histological micrographs," in *19th International Conference on Pattern Recognition, 2008. ICPR 2008*, Dec. 2008, pp. 1–4.
- [24] G. Loy and A. Zelinsky, "Fast radial symmetry for detecting points of interest," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 959–973, Aug. 2003.
- [25] Q. Yang and B. Parvin, "Perceptual organization of radial symmetries," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*, vol. 1, Jun. 2004, pp. I–320–I–325 Vol.1.
- [26] M. Veta, A. Huisman, M. Viergever, P. J. Van Diest, and J. P. W. Pluim, "Marker-controlled watershed segmentation of nuclei in hematoxylin-stained breast cancer biopsy images," in *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, Mar. 2011, pp. 618–621.
- [27] A. C. D'Souza, "Automated counting of cell bodies using nissl stained cross-sectional images," Master's thesis, Texas A&M University, Texas A&M University, 2007.
- [28] D. Mayerich, J. Kwon, A. Panchal, J. Keyser, and Y. Choe, "Fast cell detection in high-throughput imagery using GPU-accelerated machine learning," in *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, Mar. 2011, pp. 719–723.
- [29] C. Sung, J. Woo, M. Goodman, T. Huffman, and Y. Choe, "Scalable, incremental learning with MapReduce parallelization for cell detection in high-resolution 3d microscopy data," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, Aug. 2013, pp. 1–7.
- [30] J. W. Han, T. P. Breckon, D. A. Randell, and G. Landini, "The application of support vector machine classification to detect cell nuclei for automated microscopy," *Machine Vision and Applications*, vol. 23, no. 1, pp. 15–24, Jan. 2012.
- [31] C. Malon and E. Cosatto, "Classification of mitotic figures with convolutional neural networks and seeded blob features," *Journal of Pathology Informatics*, vol. 4, no. 1, p. 9, 2013.
- [32] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, "Fiji: an open-source platform for biological-image analysis," *Nature Methods*, vol. 9, no. 7, pp. 676–682, Jul. 2012.
- [33] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
- [34] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [35] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001*, vol. 1, 2001, pp. I-511–I-518 vol.1.
- [36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and . Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, p. 28252830, Oct. 2011.
- [37] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, Aug. 1995.
- [38] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.
- [39] D. Comaniciu, V. Ramesh, and P. Meer, "The variable bandwidth mean shift and data-driven scale selection," in *Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001. Proceedings*, vol. 1, 2001, pp. 438–445 vol.1.
- [40] G. A. Johnson, A. Badea, J. Brandenburg, G. Cofer, B. Fubara, S. Liu, and J. Nissanov, "Waxholm space: An image-based reference for coordinating mouse brain research," *Neuroimage*, vol. 53, pp. 365–372, 2010.
- [41] S. Lal Das, "Cell detection in Knife-Edge Scanning Microscopy images of nissl-stained mouse and rat brain samples using random forests," Master's thesis, Department of Computer Science and Engineering, Texas A&M University, 2014.