Localization and Tracking in 4D Fluorescence Microscopy Imagery

Shahira Abousamra[1], Shai Adar[2], Natalie Elia[2], Roy Shilkrot[1]
[1]Department of Computer Science, Stony Brook University, USA
[2]Department of Life Sciences and the NIBN, Ben-Gurion University of the Negev, Israel
sabousamra@cs.stonybrook.edu, {shaiad, elianat}@post.bgu.ac.il, roys@cs.stonybrook.edu

Abstract

3D fluorescence microscopy continues to pose challenging tasks with more experiments leading to identifying new physiological patterns in cells' life cycle and activity. It then falls on the hands of biologists to annotate this imagery which is laborious and time-consuming, especially with noisy images and hard to see and track patterns. Modeling of automation tasks that can handle depth-varying light conditions and noise, and other challenges inherent in 3D fluorescence microscopy often becomes complex and requires high processing power and memory. This paper presents an efficient methodology for the localization, classification, and tracking in fluorescence microscopy imagery by taking advantage of time sequential images in 4D data. We show the application of our proposed method on the challenging task of localizing and tracking microtubule fibers' bridge formation during the cell division of zebrafish embryos where we achieve 98% accuracy and 0.94 F1-score.

1. Introduction

In cell biology, fluorescence microscopy allows the use of fluorescent indicators to highlight specific targets such as proteins, lipids, ions, etc. [33]. It allows higher visibility of their occurrence, activity and development, thus enabling numerous advancements in understanding cell physiology. Nevertheless, it suffers in terms of image quality, mainly due to the diffraction of light in the microscope optics and often the limited light allowed in order to maintain the cells alive in vivo experiments. To extract insights from the imagery, there is usually a need to segment and classify the various structures and objects of interest, in addition to tracking them through time in live experiments. This is a laborious and time-consuming task for human analysts, especially with large amounts of noisy imagery in which it is often hard to see and track patterns.

A 3D microscopic image ($x \times y \times z$) is formed of z two dimensional depth-slices or layers, each of size $x \times y$, at fixed step size. Together a number of 3D images taken at fixed sequential time periods form 4 dimensional data. Modeling automation tasks with 3D or 4D data can result in high execution time and computation cost, in addition to disk space and memory requirements especially with increasing number of depth-slices. While maximum or mean intensity projection methods are often used to convert 3D fluorescence microscopy images into 2D images [23], we show that they may not work well where an object of interest can vanish in the conversion. An object vanishing can be attributed to a combination of factors, such as: depth-varying lighting conditions, low contrast-to-noise ratio, and small-sized objects. These conditions are likely to occur with 3D fluorescence microscopy imagery depending on the target of the fluorescent indicators. Additionally, the number of slices is often varying. This can occur when increasing the number of slices is important to see more of the sample but is limited by phototoxicity effects and speed of the system. Consequently, biologists may choose to use a varying number of slices during different periods of data acquisition.

In this work we present an efficient and practical method for the localization and tracking of objects of interest in fluorescent microscopy imagery that suffer from these challenging conditions. Specifically, we present a method for region of interest (ROI) extraction, 3D to 2D image compression, classification, and tracking for handling elusive patterns without the need for high processing power and memory by taking advantage of the fourth dimension, time. We demonstrate the efficiency and accuracy of this method in handling depth-varying lighting conditions and noise, low contrast-to-noise ratio, small objects of interest, and varying number of slices, by applying it to the problem of localizing and tracking mircrotubule fibers bridge formation during cell division in zebrafish embryos (see Figures 1 and 2).

The paper is organized as follows: Section 2 presents related work, Section 3 describes the dataset, Section 4 details the proposed method, Section 5 contains the experimental results and finally, conclusion is presented in Section 6.

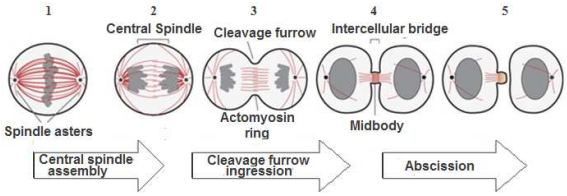

Figure 1. Microtubule fibers organization during cell division. Red fibers in the illustration represent microtubules [12]

2. Related Work

Fluorescence microscopy imagery often suffers from noise. Some of the filtering mechanisms widely used include rolling ball [9], Gaussian blurring, and more advanced spatial filters such as [15], [6], [3]. While some are more computationally expensive than others, spatial filters in general tend to smooth out the edges. Other methods that specifically target microscopy and medical images include [39, 22, 17, 44, 25, 43, 21, 31, 24]: In [39] the noise is modeled as additive noise using spatial filters that is estimated at every point. In [22] the authors try to speed up the process in [39] by making use of multiple channels in fluorescence imagery when they are well separated. In [17] noise reduction is posed as a maximum a posteriori estimation problem and solved using a stochastic random field. It works on contrast to noise ratio higher than 10db. Recently principle component analysis has inspired and been used in denoising methods such as [44, 25]. [43] transforms mixed poisson and gaussian noise into additive gaussian noise to be easily denoised. In [21, 31] structure and edges are better preserved but at the expense of more noise in the resulting image. [24] targets poisson noise. It works well with low signal to noise ratio but not so well with low contrast to noise ratio. In our work, the dataset we use (Section 3) is characterized by depth-varying noise, low contrast-to-noise ratio, and small objects of interest. Our proposed solution reduces noise from the slices and combines them in a 2D representation that amplifies the regions of interest to be used in classification, without sacrificing performance or having a complex, hard to tune model.

Segmentation allows us to localize the objects of interest such as cells or nuclei. Some of the popular segmentation methods in computer vision and biomedical imagery include watershed and its variants [38, 4, 40], morphological and thresholding methods such as top hat transform [5] with otsu thresholding [28] are used in [41], and deep neural networks as in [13, 1, 32, 37]. Some methods target 3D imagery specifically such as [10, 8]. Recently [30] was shown to give better segmentation results in biomedical images than [2, 32] by creating dense connections among features from different scales that are obtained using dilated convolutions. However, our goal here is not segmentation in itself but rather localization. We create an inexact or semi-segmentation of the objects of interest by constructing a binary mask for each slice in a 3D image and combining them using maximum projection to create a mask for the entire stack of slices making up the 3D image. The objects in the mask are localized by their centroids. They represent the fluorescence objects in the image. More details are in Section 4.1.

There are various methods for classification. In [11] a weighted support vector machine is used. [16] uses a random forest that is applied iteratively over different segmentations to select the best one. [29] uses convolutional blocks that are learnt using K-SVD. While methods such as support vector machines (SVM) or random forests mainly rely on manually selected features as in [11, 29], convolutional neural networks (CNN) learn the features that are useful for the classification task. The seminal AlexNet [20] won the ImageNet Large Scale Visual Recognition Challenge competition, and was followed by other outperforming CNNs such as [42, 34, 18]. CNNs were also adopted in microscopy images, for example [27]. As the layers of the network go deeper, more information can be extracted and higher accuracy is achievable, especially with larger input size. However the greater the number of weight parameters that need training, the longer the network takes to train, the harder it is to configure, and the more processing memory and power required. Here, we use a shallow CNN that is fed relatively small 2D patches around regions of interest. The classification results across the lifetime of the object of interest are used to vote for the final classification of that object. Details are in Sections 4.4 and 4.5.

Tracking objects such as cells throughout timeframes of microscopic images in a video capture is often performed using Bayesian filtering and its variations [36]. Kalman and Particle filters are particularly popular [14, 7, 35]. While bayesian filters are complex, recently recurrent neural network was also used for multiple target tracking [26]. In this work, with the observation that an object is the closest to its location in the previous timeframe, we assume a restricted random walk model (Section 4.3), and since the shape of an object can change between sequential timeframes, the model is independent of appearance. This results in a simple tracking model as explained in Section 4.3.

3. Dataset

We are using fluorescence microscopy imagery of zebrafish embryo obtained from The Elia Lab for cellular imaging in Ben-Gurion University[1]. The optical transparency property of zebrafish embryo makes them ideal for studying the physiological activities that occur within. The captured imagery represents a recording of cells divi-

[1] http://lifeserv.bgu.ac.il/wb/elianat/

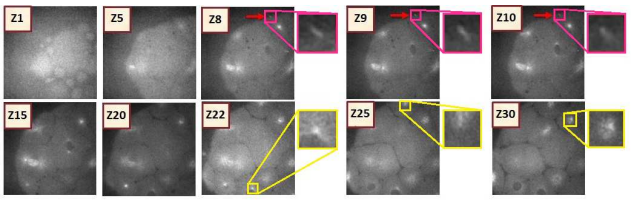

Figure 2. Sample slices from a single timeframe. The Z number indicates the slice number ranging from 1 to 30. The arrow points to a bridge formation. The pink and yellow rectangles are enlargements of bridge and nucleus locations, respectively.

sion within the embryo. Here, the fluorescence markers target the tubulin where the organization of microtubule fibers changes during cell division as shown in Figure 1. The goal is to track the bridge formation of the microtubule fibers over time as a way to analyze the timing of the abscission process (stages 4 and 5 in Figure 1).

A dataset capture consists of consecutive timeframes, each consisting of between 28 and 31 depth slices of 512 $\times$ 512 pixels. The images are 0.324 microns per pixel, with Z step-size of 0.9 microns and 2 minutes interval between consecutive frames. Figure 2 shows some sample slices from a single timeframe. Tracking the bridge formations like the one in Figure 2 indicated by the arrows in slices 8, 9, and 10 is a laborious and time consuming task for biologists. Our objective is to automate the localization and tracking of these bridges. For training and evaluation purposes we are given the location of the bridges in each timeframe as bounding rectangles.

4. Method

Below is an outline of our suggested approach for localizing and tracking the bridge formations followed by a detailed description of the steps involved.

We define a region of interest (ROI) as an area that potentially contains a bridge. We first identify the ROIs in each slice of a timeframe as a binary mask of the slice. The 3D mask formed from the stack of slice masks is then collapsed using maximum projection into a 2D mask that highlights the ROIs in that timeframe as a whole. Similarly, the 3D image slices are compressed into a 2D image using weighted averaging that is based on the ROI 3D mask obtained previously. Using this 2D image representation, the objects retrieved from the ROIs are tracked through time as well as passed to a convolutional neural network for classification. The tracking information along with the classification are used to localize a bridge throughout the timeframes of a capture. The following sections provide detailed description of each step.

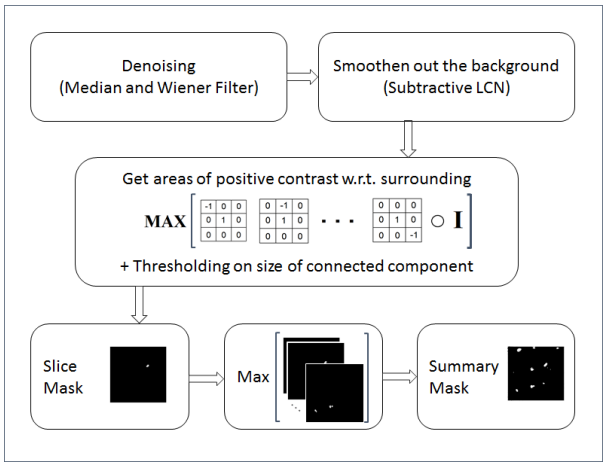

Figure 3. Steps for obtaining ROI mask.

4.1. Acquiring Regions of Interest (ROI)

Figure 3 shows an outline of the process to get the ROIs in each timeframe. Here we present a detailed description of that process.

An inherent property of fluorescence microscopy imagery is the non-uniform noise across the slices that is a mixture of Gaussian and Poisson or shot noise [17]. To get our ROIs, we want to highlight the fluorescence areas and smoothen out the background. This is achieved by applying a median filter followed by adaptive Gaussian filter, also known as a Wiener filter. We then perform subtractive local contrast normalization on the filtered image [19]. The result is an image that highlights locations with higher variance than the filter used in the subtracted image. Locations that have positive contrast with respect to their surrounding pixels are then obtained by convolving the image with 8 (3 $\times$ 3) filters; each have a value of 1 in the center pixel, a value of -1 in one of the edge pixels, and zeros everywhere else (see Figure 3). A maximum over all 8 convolutions is thresholded on the contrast value obtained and the size of the connected component at each pixel. The thresholding on the size of the connected components is specially important to avoid getting scattered outliers around the image. The resulting image is thus the slice mask. Figure 4 shows the stages for getting the slice mask for 3 samples.

Having obtained a mask for every slice in a timeframe, a maximum over all the slice masks gives the ROIs mask for the timeframe as a whole. We call that a *summary mask*. Figure 5 shows samples of the summary mask.

4.2. 3D to 2D Representation

The slice masks are used to combine the slices into a representative 2D image of the timeframe. They are combined in a weighted average manner such that areas containing ROIs in a slice get higher weight from that slice

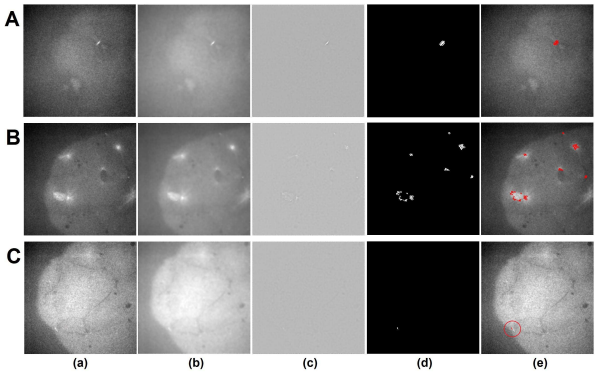

Figure 4. Samples from the steps for obtaining a slice ROI mask. Rows represent different slice samples A, B, and C. Columns: (a) original slice image (b) filtered image (c) subtractive LCN (d) slice ROI mask (e) slice ROI mask overlaid over original image.

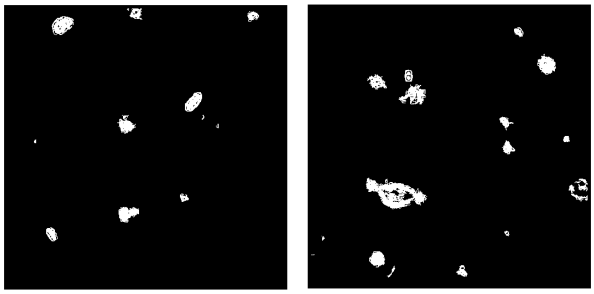

Figure 5. Sample summary masks of 2 different timeframes in the dataset.

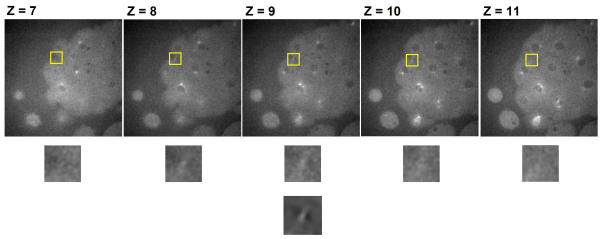

Figure 6. Sample slices from a timeframe showing a bridge exhibiting low contrast to noise ratio. Z is the slice depth, the boxes indicate the bridge location. Second row is the bridge location magnified. Third row is the outcome of the filtering and weighted averaging proposed.

compared to other slices. We experiment with 2 strategies: first, equal contribution strategy, where for each $pixel_{mask}(X, Y, Z) = 1$ in the slice mask at depth Z a 60 $\times$ 60 square region around it in that slice is considered to contribute equally to the final image by setting $W(X-30 : X+30, Y : 30 : Y+30, Z) = 1$. Then the final value of any $pixel(X, Y) = \frac{1}{\sum_{Z=1}^{depth} W(X,Y,Z)} \sum_{Z=1}^{depth} pixel(X, Y, Z)$. Second, a Gaussian contribution strategy, where instead of equal contribution around $pixel_{mask}(X, Y, Z) = 1$ in the slice mask at depth Z, we use a 120 $\times$ 120 Gaussian weight matrix with $\sigma = 10$ and scaled such that the center weight $= 1$. Since a pixel contribution can vary when considering the application of the Gaussian weight to all the slice-mask's 1 pixels, the highest weight is used. Using a similar equation as before, the final value of any $pixel(X, Y) = \frac{1}{\sum_{Z=1}^{depth} W(X,Y,Z)} \sum_{Z=1}^{depth} pixel(X, Y, Z) \times W(X, Y, Z)$. For all other pixels that have no specific slice contribution the average over all slices is used. The pixel intensity values in the weighted averages are not from the original slice images but rather the filtered slices as described in Section 4.1, where the original image slices are median and wiener filtered and then subtractive and divisive local contrast normalization is applied [19].

Figure 6 shows consecutive slices from a single timeframe. The bridge is most apparent in slices 8 and 9 of the 30 depth slices with the highest contrast to noise ratio = 2.56 db. The figure shows the result of our proposed scheme of filtering and slice weighted averaging where the contrast to noise ratio rises to 8.9 db.

Figure 7 shows the 2D representation using 4 different methods: mean pooling, max pooling, weighted average with equal contribution and Gaussian contribution. The mean pooling does a very poor job. The bridges are mostly invisible. Bridges 3 and 4 are barely visible with max pooling. They are apparent with the equal contribution weights but then bridges 6 and 7 are sort of cutoff due to nearby ROIs in different slices. The Gaussian contribution weights does a fair job at maximizing the visibility of the bridges and avoiding cutoffs through a smooth blending between close regions. Although different regions seem to have different lighting conditions due to taking weights from different sets of slices, it is not an issue since we are only interested in having a clear view around the ROI. Note that the description of the conversion from a 3D to a 2D representation of the timeframe so far assumed that no two cells at different depths overlap with each other. This is indeed the case in our dataset with a total depth stack of around 28-30 microns. However, 3D timeframes with overlapped cells may occur in other data sets, for example ones with stacks of larger depths. This case can be handled by converting a 3D timeframe to two or more 2D images, such that each 2D image represents a subset of the timeframe slices where no cells overlap with each other.

4.3. Tracking ROI

The cells, nuclei, and bridges are not static across time but rather move around. Their motion is expressed through displacement between the timeframes. That displacement need not be in a specific direction. With the 2 minute interval between frames, the only observation is that each object remains the closest to its previous location. So it is

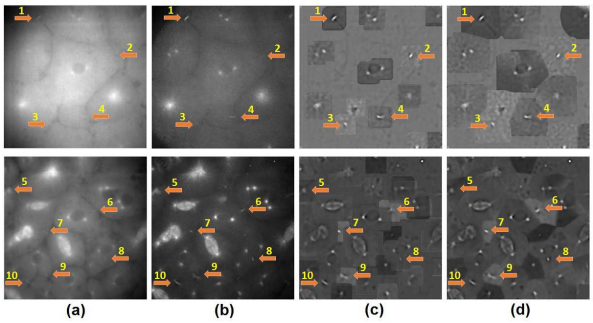

Figure 7. Samples of 2D representation of 3D timeframe images. The arrows point to bridge formation locations. Each row is a different timeframe. The columns represent the 2D pooling as: (a) mean intensity pooling (b) max intensity pooling (c) weighted average with equal contribution strategy (d) weighted average with Gaussian contribution strategy.

more like a restricted random walk where the restriction is in the sense of how far an object can be displaced between consecutive timeframes. We use this observation to track the objects (i.e. ROIs) across time. The objects to track are identified by the centroids of the blobs formed from the summary mask image. Working sequentially across time, for each object J in frame F_t the closest object J' in frame F_{t+1} is found. This can be done simply by multiplying the summary mask of frame F_{t+1} with a Gaussian matrix centered at location J. The size of the Gaussian corresponds to the restriction allowed in the displacement between timeframes. J' then corresponds to the location of maximum resulting value. The pairs (J, J') are processed in ascending order by distance. For each pair (J, J'), object J in F_t is mapped to J' in F_{t+1} if no other object in F_t was already mapped to J'. Otherwise, object J is assumed to cease to exist in frame F_{t+1}.

4.4. Classifying ROI

A convolutional neural network is used for classification. Since we are only interested in bridge patterns, the classes are bridge and nucleus (or other). The network consists of 2 convolutional layers that use same padding, a fully connected layer, and a softmax layer. Each convolutional layer is followed by a rectified linear unit and maximum pooling. The architecture is shown in Figure 8. The input is 60×60 patches taken from the 2D representation around pixels with value 1 in the binary summary mask. Using the training dataset described in Section 3 we have far fewer bridges than non-bridge parts. We deal with this class imbalance by using a weighted loss function where the class weights are inversely proportional with the amount of training data available from that class. Also it is important to note that while it is usually advised to use batch normalization in convolutional neural network it does not work well

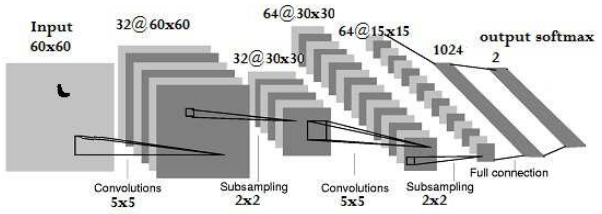

Figure 8. Architecture of the classification convolutional neural network.

in this case; the class imbalance makes the learnt normalization parameters misrepresentative of the data. The training data is composed of about 40 bridge and 80 non-bridge ROIs. For each ROI, 10 random samples in the ROI are used and augmented with flipping and rotation. The network configuration has a learning rate of 0.0005, a batch size of 160 and uses an Adam Optimizer. It seems to converge well after around 75 epochs.

4.5. Tracking and Classification Combined

The result from the classification is combined with the tracking to identify the bridge formations over a time sequence. If an object is classified as a bridge through some but not all of its instances over time we can still find all of its occurrences by combining the results from the classification and tracking.

Let $N_{bridge}(J, F_i)$: the number of patches belonging to object $J's$ ROI in frame F_i that are classified as a bridge, $N_{total}(J, F_i)$: the total number of patches from object $J's$ ROI in frame F_i that are fed to the convolutional neural network for classification, and $P_{bridge}(J, F_i)$: the probability of an object J to be a bridge in frame F_i. Then $P_{bridge}(J, F_i) = \frac{N_{bridge}(J,F_i)}{N_{total}(J,F_i)}$. For an object J in capture C to be classified as a bridge, its instances throughout the capture (as identified by the tracking) need to satisfy a few conditions related to the minimum number of instances with some probability of being a bridge, and its maximum and mean probability throughout the capture. These conditions are summarized as follows:

$$|\forall_{F_i \in capture} P_{bridge}(J, F_i) > 0| \geq 2$$

$$\max_{\forall F_i \in C \text{ s.t. } J \in F_i} P_{bridge}(J, F_i) > p_1$$

$$\operatorname{mean}_{\forall F_i \in C \text{ s.t. } J \in F_i} P_{bridge}(J, F_i) > p_2$$

where p1, p2 are determined experimentally as 0.75, and maximum$(0.25, \frac{0.8}{|instances|})$ respectively. Together these conditions ensure that the object has a high probability of being a bridge in at least one timeframe and is reasonably classified as a bridge throughout its lifetime. Figure 9 shows sample results from the test data.

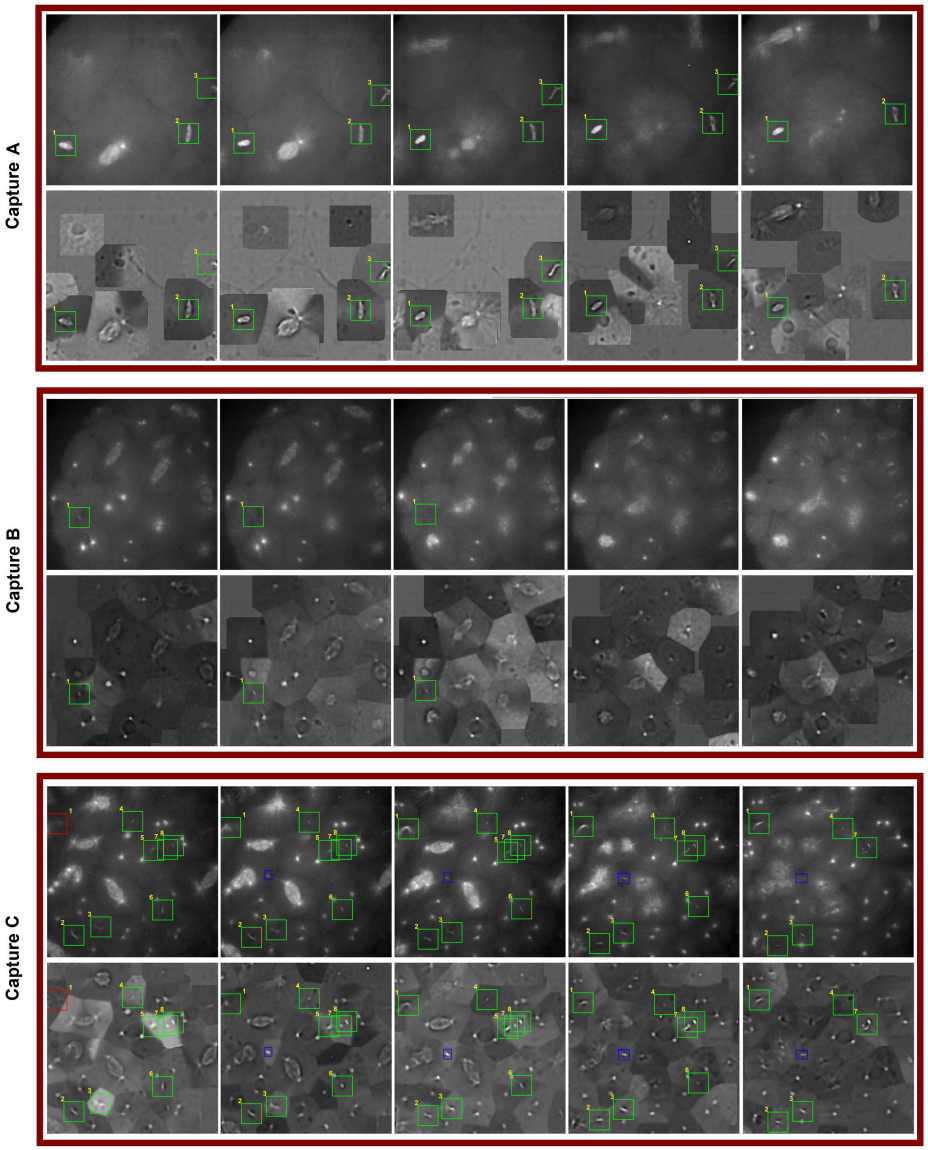

Figure 9. Sample results from the classification and tracking in 3 different captures. 5 consecutive timeframes are shown from each capture. The top and bottom rows in each are the classification and tracking overlaid over maximum pooling and Gaussian weighted pooling respectively. Boxes coloring: green = true positive, red = false positive, and blue = false negative.

5. Experimental Results

The dataset we use is described in Section 3. We have 2 sets of captures. Each set is taken over a single zebrafish embryo sample. We can either take all the combined data we have, shuffle and divide into training and test, or we can use one set for training and the other for testing. We choose the second option so that the test set is completely new and we can evaluate how well the system generalizes. The training set consists of 5 captures of 78 timeframes and 11 bridges in total, and the test set consists of 5 captures of 44 timeframes and 38 bridges in total. The number of bridges is such that a bridge is counted once throughout its lifetime in a single capture.

We run the pipeline as detailed in the previous section on the test set. The results are quantified on 2 levels; the instance level and the bridge level. In the instance level, a true positive is a bridge identified correctly in a single timeframe. In the bridge level, a true positive is a bridge that is identified correctly over the entire course of a capture. Figure 9 shows samples from the results obtained using our proposed method over different captures. The captures are from different stages of the cells population growth. It shows how accurate it performs in different

population sizes. Capture A represents an early stage of the population. In this stage the bridges are large and far apart. They are all correctly classified and tracked. In capture B the population size is larger and so the number of cells featured in the sample imagery increases. From all the ROIs in the capture, the one bridge that occurs in it is correctly identified and tracked. In Capture C the population size has further increased. There are more bridges and are closer to one another and to neighboring nuclei. The false positive in the first frame is actually the beginning of a bridge which becomes apparent in the following timeframes. In the last frame we see that very close bridges are not well disambiguated. They are correctly classified but their very close proximity results in the same identifier given to them. The accuracy, type I error, and type II error for the per-instance statistics are 98%, 0.3%, and 1.5% respectively. Furthermore, to understand the value of combining the tracking with classification we present the per-instance statistics when using *only the classifier* without the tracker: the accuracy, type I error, and type II error are 90.1%, 2.97%, and 6% respectively. Table 1 shows the precision, recall, and F1 scores. The lower accuracy and scores when using the classifier only signify the importance of the combined approach in identifying the bridge instances, making the process more robust to the various elusive patterns that a bridge can take over its lifetime. It should be noted that the difference in the numbers for the per instance and per bridge statistics is due to the fact that the number of instance occurrences of a bridge is not uniform. Some bridges appear in over 10 sequential timeframes and others appear in only 3. Both cases are counted as 1 in the per bridge statistics and hence the difference in the statistical results. Figure 10 shows samples of the patches used in training and Figure 11 shows samples of the test results. The false positives in test data are mostly the beginning of a bridge prior to its clear formation; thanks to the tracking it is identified early on. The false negatives are pretty close to the shape of a nucleus making them hard to identify. The set of true positives shows how well the system generalizes where these exact formations are not present in the training data.

Method	Per-instance / Per-bridge	Precision	Recall	F1 score
Classifier + Tracker	Per-instance	**0.98**	**0.91**	**0.94**
Classifier + Tracker	Per-bridge	0.97	0.85	0.91
Classifier Only	Per-instance	0.76	0.61	0.68

Table 1. Statistical Results.

6. Conclusion

In this paper we have proposed a method for the localization and tracking in 4D fluorescent microscopy imagery with depth-varying noise and lighting conditions, and where the objects of interest are often very small and suffer from very low contrast-to-noise ratio. We have shown the efficiency of our proposal with the localization, classification, and tracking of microtubule fibers bridge formations in zebrafish embryos achieving 98% accuracy. The suggested approach optimizes for both efficiency and complexity without the need for high processing power or memory. We expect our method can be successfully applied to other fluorescence microscopy imagery datasets with the tweaking of few parameters, mainly the thresholding parameters in the ROI extraction step.

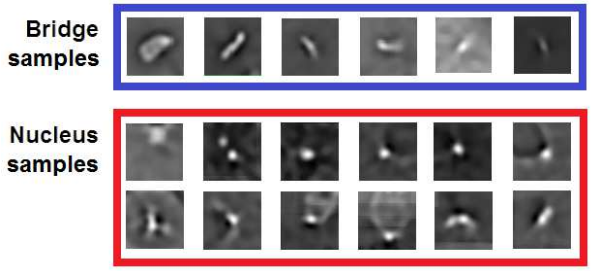

Figure 10. Sample training patches for ROI locations representing bridge formations and parts of nuclei.

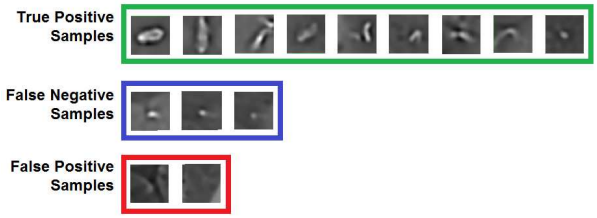

Figure 11. Sample test patches

References

[1] A. S. Aydin, A. Dubey, D. Dovrat, A. Aharoni, and R. Shilkrot. CNN based yeast cell segmentation in multi-modal fluorescent microscopy data. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 753–759, July 2017.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561, 2015.

[3] R. Bernardes, C. Maduro, P. Serranho, A. Araujo, S. Barbeiro, and J. Cunha-Vaz. Improved adaptive complex diffusion despeckling filter. *Opt Express*, 18(23):24048–24059, Nov 2010.

[4] A. Bleau and L. Leon. Watershed-based segmentation and region merging. *Computer Vision and Image Understanding*, 77(3):317 – 370, 2000.

[5] D. S. Bright and E. B. Steel. Two-dimensional top hat filter for extracting spots and spheres from digital images. *Journal of Microscopy*, 146(2):191–200, 1987.

[6] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005.

[7] N. Chenouard, I. Bloch, and J. C. Olivo-Marin. Multiple hypothesis tracking in microscopy images. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1346–1349, June 2009.

[8] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In *MICCAI (2)*, volume 9901 of *Lecture Notes in Computer Science*, pages 424–432, 2016.

[9] T. J. Collins. ImageJ for microscopy. *BioTechniques*, 43(1 Suppl):25–30, Jul 2007.

[10] B. Dong, L. Shao, M. D. Costa, O. Bandmann, and A. F. Frangi. Deep learning for automatic cell detection in wide-field microscopy zebrafish images. In *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, pages 772–776, April 2015.

[11] T. H. Du, W. C. Puah, and M. Wasser. Cell cycle phase classification in 3D in vivo microscopy of Drosophila embryogenesis. *BMC Bioinformatics*, 12 Suppl 13:S18, 2011.

[12] J. P. Fededa and D. W. Gerlich. Molecular control of animal cell cytokinesis. *Nat. Cell Biol.*, 14(5):440–447, May 2012.

[13] C. Fu, D. J. Ho, S. Han, P. Salama, K. W. Dunn, and E. J. Delp. Nuclei segmentation of fluorescence microscopy images using convolutional neural networks. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pages 704–708, April 2017.

[14] A. Genovesio, T. Liedl, V. Emiliani, W. J. Parak, M. Coppey-Moisan, and J. C. Olivo-Marin. Multiple particle tracking in 3-D+t microscopy: method and application to the tracking of endocytosed quantum dots. *IEEE Transactions on Image Processing*, 15(5):1062–1070, May 2006.

[15] M. Ghazal, A. Amer, and A. Ghrayeb. Structure-oriented multidirectional wiener filter for denoising of image and video signals. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(12):1797–1802, Dec 2008.

[16] J. Gul-Mohammed, I. Arganda-Carreras, P. Andrey, V. Galy, and T. Boudier. A generic classification-based method for segmentation of nuclei in 3D images of early embryos. *BMC Bioinformatics*, 15:9, Jan 2014.

[17] S. A. Haider, A. Cameron, P. Siva, D. Lui, M. J. Shafiee, A. Boroomand, N. Haider, and A. Wong. Fluorescence microscopy image noise reduction using a stochastically-connected random field model. *Sci Rep*, 6:20640, Feb 2016.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016.

[19] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *Proc. International Conference on Computer Vision (ICCV'09)*. IEEE, 2009.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[21] D.-J. Kroon, C. H. Slump, and T. J. J. Maal. Optimized anisotropic rotational invariant diffusion scheme on cone-beam ct. In T. Jiang, N. Navab, J. P. W. Pluim, and M. A. Viergever, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010*, pages 221–228, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[22] C. Li, X. Li, H. Cao, H. Jiang, X. Deng, D. Z. Chen, L. Yang, and Z. Shao. Fast background removal method for 3D multi-channel deep tissue fluorescence imaging. In M. Descoteaux, L. Maier-Hein, A. Franz, P. Jannin, D. L. Collins, and S. Duchesne, editors, *Medical Image Computing and Computer-Assisted Intervention MICCAI 2017*, pages 92–99, Cham, 2017. Springer International Publishing.

[23] F. Long, J. Zhou, and H. Peng. Visualization and analysis of 3D microscopic images. *PLoS Computational Biology*, 8(6), 2012.

[24] F. Luisier, C. Vonesch, T. Blu, and M. Unser. Fast interscale wavelet denoising of poisson-corrupted images. *Signal Processing*, 90(2):415 – 427, 2010.

[25] T.-H. Ma, T.-Z. Huang, X.-L. Zhao, and Y. Lou. Image deblurring with an inaccurate blur kernel using a group-based low-rank image prior. *Information Sciences*, 408:213 – 233, 2017.

[26] A. Milan, S. Hamid Rezatofighi, A. Dick, K. Schindler, and I. Reid. Online multi-target tracking using recurrent neural networks. In *AAAI*, pages 4225–4232, 2017.

[27] H. Niioka, S. Asatani, A. Yoshimura, H. Ohigashi, S. Tagawa, and J. Miyake. Classification of C2C12 cells at differentiation by convolutional neural network of deep learning using phase contrast images. *Human Cell*, 31(1):87–93, Jan 2018.

[28] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, Jan 1979.

[29] M. Pachitariu, A. Packer, N. Pettit, H. Dagleish, M. Hausser, and M. Sahani. Extracting regions of interest from biological images with convolutional sparse block coding. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 1745–1753, USA, 2013. Curran Associates Inc.

[30] D. M. Pelt and J. A. Sethian. A mixed-scale dense convolutional neural network for image analysis. *Proceedings of the National Academy of Sciences*, 2017.

[31] J. Rajan, K. Kannan, and M. R. Kaimal. An improved hybrid model for molecular image denoising. *J. Math. Imaging Vis.*, 31(1):73–79, May 2008.

[32] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[33] M. J. Sanderson, I. Smith, I. Parker, and M. D. Bootman. Fluorescence microscopy. *Cold Spring Harb Protoc*, 2014(10):pdb.top071795, Oct 2014.

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[35] I. Smal, W. Niessen, and E. Meijering. Advanced particle filtering for multiple object tracking in dynamic fluorescence microscopy images. In *2007 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1048–1051, April 2007.

[36] S. Srkk. *Bayesian Filtering and Smoothing*. Cambridge University Press, New York, NY, USA, 2013.

[37] D. A. Van Valen, T. Kudo, K. M. Lane, D. N. Macklin, N. T. Quach, M. M. DeFelice, I. Maayan, Y. Tanouchi, E. A. Ashley, and M. W. Covert. Deep Learning Automates the Quantitative Analysis of Individual Cells in Live-Cell Imaging Experiments. *PLoS Comput. Biol.*, 12(11):e1005177, Nov 2016.

[38] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, Jun 1991.

[39] L. Yang, Y. Zhang, I. H. Guldner, S. Zhang, and D. Z. Chen. Fast background removal in 3D fluorescence microscopy images using one-class learning. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 292–299, Cham, 2015. Springer International Publishing.

[40] X. Yang, H. Li, and X. Zhou. Nuclei segmentation using marker-controlled watershed, tracking using mean-shift, and kalman filter in time-lapse microscopy. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 53(11):2405–2414, Nov 2006.

[41] X. Yuan, L. Gu, L. Sun, and T. Ikenaga. Local-threshold 2D-tophat cell segmentation for the two-photon confocal microscope image. In *MVA*, pages 455–458, 2013.

[42] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing.

[43] B. Zhang, M. J. Fadili, J. L. Starck, and J. C. Olivo-Marin. Multiscale variance-stabilizing transform for mixed-poisson-gaussian processes and its applications in bioimaging. In *2007 IEEE International Conference on Image Processing*, volume 6, pages VI – 233–VI – 236, Sept 2007.

[44] Y. Zhang, J. Liu, M. Li, and Z. Guo. Joint image denoising using adaptive principal component analysis and self-similarity. *Information Sciences*, 259:128 – 141, 2014.