AUTOMATIC SEGMENTATION OF NUCLEI IN 3D MICROSCOPY IMAGES OF C.ELEGANS

Fuhui Long, Hanchuan Peng, and Eugene Myers
Janelia Farm Research Campus, Howard Hughes Medical Institute, Ashburn, Virginia, USA

ABSTRACT

Automatic segmentation of nuclei in 3D microscopy images is essential for many biological studies including high throughput analysis of gene expression level, morphology, and phenotypes in single cell level. The complexity and variability of the microscopy images present many difficulties to the traditional image segmentation methods. In this paper, we present a new method based on 3D watershed algorithm to segment such images. By using both the intensity information of the image and the geometry information of the appropriately detected foreground mask, our method is robust to intensity fluctuation within nuclei and at the same time sensitive to the intensity and geometrical cues between nuclei. Besides, the method can automatically correct potential segmentation errors by using several post-processing steps. We tested this algorithm on the 3D confocal images of *C.elegans*, an organism that has been widely used in biological studies. Our results show that the algorithm can segment nuclei in high accuracy despite the non-uniform background, tightly clustered nuclei with different sizes and shapes, fluctuated intensities, and hollow-shaped staining patterns in the images.

KEYWORDS: Image segmentation, microscopy, biological cells, shape, intensity

1. INTRODUCTION

Automatic segmentation of nuclei in 3D microscopy images is an essential yet very challenging problem in many biological studies where the quantitative or high throughput analysis of nuclei gene expression level, morphology, and cell phenotypes is needed. Most of the difficulties come from the variability and complexity of the images caused by the intrinsic properties of the samples, as well as the staining and imaging procedures. In these images, the background intensity generally varies at different locations. Nuclei can be tightly clustered, making it hard for biologists to tell the exact boundaries between nuclei. Typically the intensities within nuclei are highly non-uniform and fluctuated. In many cases, the intensity variation within a nucleus can be even bigger than that between nuclei. Besides, the intensity profiles of nuclei can be very complicated due to the imperfect staining or intrinsic intra-cellular characteristics. For example, the unstained nucleolus can make a nucleus appear hollow-shaped. Furthermore, the sizes and shapes of nuclei can vary a lot in an image. All these problems have presented difficulties to most of the traditional image segmentation methods such as thresholding, region growing, and edge-based approaches (see [1] for a review) that have been successfully applied to other applications.

Watershed algorithm has been demonstrated more suitable for segmenting biological images [2-4]. It takes an image as topographic relief and uses morphological flooding operations to determine regions and region boundaries [2,5]. There are generally two ways to use watershed algorithm, intensity-based and shape-based. The intensity-based approaches use intensity gradient image as the input to the watershed algorithm, whereas shape-based approaches use the shape information of the thresholded image of the original as the input. When dealing with the images with such properties as described earlier, however, the intensity-based method can lead to many over- and wrong- segmentations of nuclei even with the help of pre-selected markers and sophisticated merging processes, due to the existence of fluctuated and hollow shaped staining patterns. The shape-based approaches, although more robust to intensity fluctuation and typically generates much smaller number of over-segmented regions, may also produce an unacceptable number of wrong partitions of nuclei when the thresholded image does not have clear 'neck'-shaped structure between nuclei.

In this paper, we present a new watershed approach that uses both the intensity information of the image and the geometrical information of the foreground mask to segment 3D microscopy images with all the challenging properties described earlier. The diagram of the algorithm is shown in Fig. 1. The algorithm uses the intensity information to determine the appropriate foreground mask with clear 'neck'-shaped structure. The geometrical information of the foreground mask is then used by the shape-based watershed method to produce the right partition of nuclei. An adaptive intensity- and shape-based region grouping method is developed to merge over-segmented regions, the number of which is usually small. Since the segmented nuclei are usually smaller than their true sizes due to the process of producing neck-shaped foreground mask, we then used region dilation to refine nuclei boundaries. The method also includes several post-processing steps, the purposes of which are to correct segmentation errors based on the statistical information of the nuclei already segmented. These steps include recovering missed nuclei, splitting big regions, and post-processing small regions. We applied this algorithm to the 3D confocal images of *C.elegans* which process all the variability and complexity mentioned earlier and found the algorithm can segment this type of images in high accuracy.

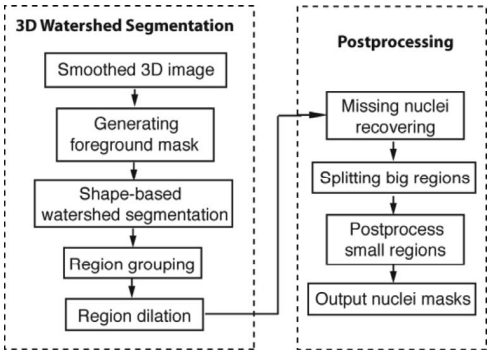

Figure 1. Diagram of the algorithm

2. GENERATING FOREGROUND MASK

To reduce noises and make intensity distributions within nuclei regions less fluctuated, we first used the Gaussian kernel to smooth a 3D image stack (3×3×3 Gaussian kernel for our data). The smoothed image was then used to extract the foreground mask that largely contains most of the nuclei and at the same time processes clear 'neck'-shaped structure.

2.1 Fill hollow-shaped patterns

To generate an appropriate foreground mask, we first filled the hollow-shaped patterns into more or less solid shaped pattern. For this purpose, we thresholded the image at different intensity level l, where $l \in [l_{min}, l_{max}]$, to obtain the binary images $T_l(I)$. For each $T_l(I)$, we used morphological 'fill' operation [1], denoted $o(T_l(I))$, to fill the holes in $T_l(I)$. The union of the pixels that belong to the difference image of $o(T_l(I))$ and $T_l(I)$ at different intensity level l constitute the hollow region E, i.e.,

$$E = \cup_l \{p \mid p \in (o(T_l(I)) - T_l(I))\} \quad (1)$$

We then used the following formula to revise the intensity of pixels within the hollow region E:

$$I'_{p\in E} = b + [\max(I_{p\in E}) - I_{p\in E}] \quad (2)$$
$$b = \text{median}(I_{p\in E'-E}) \quad (3)$$
$$E' = E \oplus e \quad (4)$$

where $\oplus$ denotes morphological dilation operation; E' is the result of dilating E with a spherical element e of 1 pixel in radius; b is the constant baseline intensity, computed as the median value of the intensities of pixels belonging to E' but outside of E. Eqn. (2)~(4) set the intensities of the peripheral pixels of region E lower than those of the central pixels, thus removing the hollow effect.

2.2 Global thresholding

After filling the hollow-shaped regions, we used Ostu's method [6] to adaptively determine the threshold T_0 that separates the foreground pixels from the background pixels. This method takes foreground and background pixels as two classes and chooses the intensity level that minimizes the intra-class intensity variance as the threshold. We then thresholded the smoothed and hollow-filled image at T_0, obtaining the binary mask image, denoted M_0 (see Fig.2b as an example).

2.3 Shrinking the mask by morphological reconstruction

When there is significant autofluorescence signal or when many nuclei are tightly clustered thus making the fluorescence signal from out-of-focus nuclei leak into in-focus nuclei, M_0 may easily contain many background pixels between nuclei. As the result, M_0 may lack of 'neck'-shaped structure between nuclei regions in many places of the image, making the shape-based watershed fail. To solve this problem, we used morphological reconstruction [7] to further 'shrink' the foreground mask M_0. More specifically, we extracted h-domes from the foreground pixels marked by M_0. An h-dome is a connected component of pixels such that every pixel in the dome has an intensity value greater than those of the pixels surrounding the dome and the intensity difference between any two pixels in the dome is less than h. Once the h-domes were extracted, we set the pixels within h-domes to 1 and the remaining to 0, obtaining the shrunk mask of nuclei areas, denoted M (Fig.2c). Such process may miss a small number of weakly stained nuclei, we recovered those nuclei in post-processing (see section 5.1).

3. SHAPE-BASED 3D WATERSHED SEGMENTATION AND REGION GROUPING

3.1 Shape-based 3D watershed segmentation

Once the foreground mask image M with clear 'neck' structure is extracted, we used 3D shape-based watershed approach to segment individual nucleus. The method first applied the distance transform to M. The distance transform of a foreground pixel in the mask is the distance (we use Euclidean distance metric) between the pixel and the nearest background pixel. The transformed image is then reverted and the intensities of background pixels set to negative infinity.

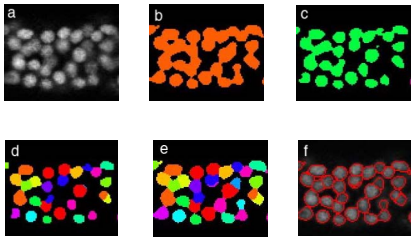

Figure 2. (a) A single slice of a small part of the 3D image of *C. elegans*. (b) Foreground mask M_0. (c) Shrunk foreground mask M. (d) Result of watershed segmentation and region grouping. (e) Result of region dilation. (Note that as regions were dilated in 3D and we only show the result on a single slice, a blue region that cannot be seen in figure d now can be seen in figure e). (f) The same result of (e) represented by overlaying region contours on (a).

After that, we applied 3D watershed algorithm to the transformed image. The watershed transform was implemented by the 3D version of the recursive immersion method proposed by Vincent and Soille [2]. It takes an image as topographic relief being immersed in a lake, with holes pierced in local minima. The catchment basins which are regions associated with these minima are filled up with water starting at these local minima and are successively expanded as water level increases. A catchment basin at water level $l+1$ can be either a new minimum or an extension of a catchment basin at water level l determined by the geodesic influence zone of the catchment basin within $T_{l+1}(I)$ (i.e, the thresholded image of I at level l+1). Dams are built at positions where water comes from different basins. The whole process iterates from the lowest to the highest intensity level. As the result, the image is partitioned into regions (corresponding to catchment basins) separated by dams (corresponding to watersheds).

3.2 Region grouping

Compared to the intensity-based watershed, the above method generates much lower degree of over-segmented regions which enables us to use a hierarchical pair-wise grouping scheme to merge over-segmented regions. Assuming the watershed segmentation generates N regions, we first computed the $N \times N$ convexity matrix R, where the value of element r_{ij} is the convexity of the region resulted from merging regions i and j. We also identified regional maxima by extracting h-domes. Each time we selected regions m and n that correspond to the maximum value of r_{ij} as the candidate regions to be merged, i.e., $(m,n) = arg(max(r_{ij}))$, $i, j \in [1,N]$. If regions m and n can be merged, the size of the resulted region should be less than S_{max}, i.e., $S_m + S_n < S_{max}$, and its pixel intensities should have only one regional maximum. If the above merging conditions are not satisfied, we set r_{mn} to 0. Otherwise, we merge regions m and n by setting region n's label equal to region

m's label, i.e., $L_n = L_m$. We also set size $S_m = S_m + S_n$ and $S_n = 0$. We then updated the convexity matrix R by setting $r_{nj} = r_{jn} = 0$ ($\forall j \in [1,N]$) and re-computing $r_{mj} = r_{jm}$ for those j that are adjacent to the new region m. We repeated the above process until no regions can be further merged. As the result, we iteratively generated N, N-1, N-2,…, and N-K number of regions. We then selected the kth ($k \in [1,K]$) iteration as the final merging result if the least convex region in the kth iteration has the maximum convexity among those in iterations 1 to K. Fig.2d shows an example of the result of watershed segmentation and region grouping.

4. REGION DILATION

Since the shrunk foreground mask makes nucleus regions smaller than their real sizes, we refined region boundaries using the following dilation approach. Given the foreground mask M_0 before shrink and the union of segmented regions Ω, we took any $p \in M_0$-Ω as foreground pixel and $p' \in \Omega$ as background pixel. We computed the distance transform of p and assigned p to a segmented region in Ω that is closest. By doing this, we also obtained the equal-distance contour for all pixels p in M_0-Ω.

We thus started from pixels that were closest to region i and gradually included the pixels that were further away based on the equal-distance contour. Since the h-dome extraction process described above may miss nuclei (which will be recovered in later steps; see 5.1), we used the following constraints to prevent dilating a segmented region into a missing region: 1) assigning foreground pixels to regions should keep regions highly convex; 2) the size of the dilated region should not exceed S_{max}. Fig.2e and 2f show an example of region dilation.

5. POST-PROCESSING

Most of the nuclei can be correctly segmented using the above steps. However, some weakly stained nuclei can be missed and segmented regions might need to be further split or merged. The post-processing steps used the statistical information of the segmented nuclei as the refined knowledge to correct these segmentation errors.

5.1 Recovering missing nuclei

We first excluded segmented nuclei by setting the intensities of all the pixels in Ω to 0. We then estimated the background level of the remaining pixels in mask M_0 using Ostu's method [6] and obtained a binary mask image M' after thresholding the remaining pixels at that level. Since the intensities of pixels in M_0-Ω are in general lower than those in Ω, the mask image M' may contain ring-shaped structures surrounding Ω. We used morphological opening operation [1] followed by median filtering to remove such artifacts. After that, we detected the connect components in M' and obtained a set of isolated regions that are potentially missing nuclei. We took those that are big enough as the recovered missing regions containing one or multiple nuclei, the later can be further segmented using region splitting method in 5.2. Fig.3 gives an example of nuclei recovering.

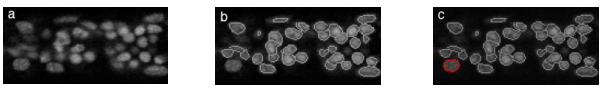

Figure 3. An example of recovering missing nuclei. (a) Original image. (b) Regions segmented using the method in section 2~4. (c) A missing nucleus region (highlighted by red boundary) is recovered.

5.2 Splitting big regions

We computed the convexity and size of each segmented region, as well as the median convexity r_{med} and median size of the regions S_{med}. We selected those satisfying the following conditions as regions that need further splitting: 1) the size S_i is bigger than $k_1 * S_{med}$; and 2) the convexity r_i is less than $k_2 * r_{med}$, where $k_1 > 1$ and $k_2 < 1$ are constants determined empirically ($k_1 = 1.5$, $k_2 = 0.9$ for our experiments). For the selected regions, we then determined the number of nucleus it most likely to contain by dividing the size of the region S_i by the median size S_{med}, i.e., $n = S_i / S_{med}$. We then used Gaussian mixture model [8] to split the region into n sub-regions, each corresponding to one nucleus. Fig.4a and 4b show examples of big region splitting.

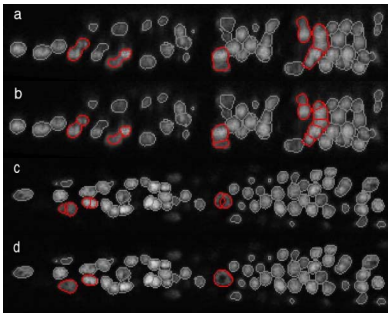

Figure 4. Post-process of big and small regions. Big regions before (a) and after (b) splitting are highlighted by red contour. Small regions before (c) and after (d) merging are highlighted in the same way.

5.3 Post-processing small regions

We selected regions whose sizes S_i are smaller than $k_3 * S_{med}$, where k_3 is a constant smaller than 1, as regions need further merging ($k_3 = 0.5$ for our experiments). To merge a small region i with its neighboring regions, we computed the convexity vector $[r_{ij}]$, where the value of element r_{ij} is the convexity of the region resulted from merging regions i with one of its neighboring region j. We then found the neighboring region j' with $r_{ij'} = \max(r_{ij})$. If it further satisfies $r_{ij'} > r_{med}$ and $S_i + S_{j'} < S_{max}$ we merged region i with j'. Otherwise we set $r_{ij'} = 0$ and found the next neighboring region satisfying the merging condition. We repeated the process until region i could no longer be merged with other regions.

If region i cannot be merged with any neighboring region, it is likely to be a small piece of a nucleus whose other pieces are missing. In this case we removed region i and used the method in section 5.1 to recover the entire nucleus. Fig. 4c and 4d show examples of small regions before and after merging.

6. ACCURACY EVALUATION

To quantitatively evaluate segmentation accuracy, we developed a 3D tool that allows us to annotate the segmentation results in detail. In principal, there are four types of segmentation errors: 1) missing nuclei; 2) falsely detecting regions that do not correspond to nuclei; 3) taking multiple nuclei as a single nucleus; and 4) splitting a single nucleus into multiple pieces. Denote the number of segmented nuclei as P and the number of nuclei of the above four cases as P_m, P_e, P_g, and P_s respectively. Then the number of cell regions that are correctly segmented, denoted as P_c, is computed as $P_c = P - (P_e + P_g + P_s)$. By manual annotation, we can obtain the true number of cells P_t. We then used P_c / P_t to quantify the segmentation accuracy.

7. RESULTS AND DISCUSSION

We used the 3D images of newly hatched first larval stage

hermaphrodites of *C. elegans* to test our algorithm, which processes all the challenging properties described earlier. Nuclei were stained with DAPI. The images were acquired using a Leica confocal microscope with 63x/1.4 oil lens.

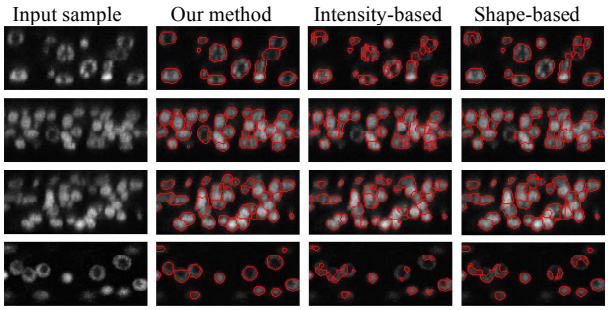

Figure 5. Comparison of our algorithm against traditional intensity-based and shape-based watershed. The first column images are sample patches of the 3D image. The second to the fourth column images are the corresponding segmentation results using our algorithm, the intensity-based, and the shape-based algorithm respectively.

We first compared our method against traditional intensity-based (using gradient image with regional maximum as seed markers) and shape-based watershed on small 3D image patches. The results are shown in Fig.5. The number of regions that are correctly segmented by our method is very much close to the true number, as shown in Table 1. In contrast, the other two methods fail to generate accurate results for our data where hollow-shaped, non-uniform stained, and tightly clustered nuclei are presented. As shown in Fig.5 and Table 1, the intensity-based method generates a much smaller number (less than half) of correctly segmented regions. Shape-based method is slightly worse than our method for tightly clustered image patches but is still significantly worse than our method when hollow-shaped nuclei are presented.

Table 1. Segmentation accuracy evaluation of Fig.5, with each row corresponding to the same row in Fig.5. P_t is the true number of nuclei, P_{c1}, P_{c2}, and P_{c3} are the number of correctly segmented regions generated by our method, the intensity-based, and shape-based methods respectively.

rows	P_t	P_{c1}	P_{c2}	P_{c3}
1	14	14	3	11
2	31	29	12	26
3	32	28	16	26
4	15	13	3	8

We then applied our method to the entire 3D images of *C. elegans*. Fig.6 shows an example of our segmentation results. In this example, most of the nuclei are correctly segmented, including the big hollow-shaped nuclei in the trunk part and the tightly clustered nuclei in the head part.

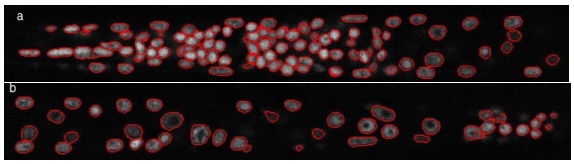

Figure 6. An example of the segmentation results of the 3D images of *C.elegans*. For clarity purpose, the image is displayed in two halves. The head and part of the trunk are displayed in (a). The remaining part of trunk and the tail are displayed in (b).

Table 2 shows the evaluation of segmentation accuracy for 5 image stacks. Due to noises and signal loss in staining and imaging processes, some nuclei may be invisible to human observers. Thus the true number of nuclei (P_t) in these stacks may deviate from the theoretical number of cells, 550. The average segmentation accuracy of the 5 stacks is 93.43%, which is quite satisfactory given the complexity of the data. We noticed that most of the errors come from head part where fluorescence signals from out-of-focus nuclei contribute significantly to the in-focus nuclei, making the boundary unclear. For the trunk part which has less clustered nuclei but many more hollow-shaped nuclei, the algorithm can achieve ~98% accuracy (not shown in Table 2). Form Table 2 we derived that the algorithm generates about 0.26% (computed by P_e/P_t) non-nuclei regions, and misses about 1.2% (computed by P_m/P_t) of nuclei that have very weak fluorescence signal. The major segmentation errors come from regions that need splitting (about 2.22%, computed by P_s/P_t) and merging (about 3.33%, computed by P_g/P_t). Most of these errors come from the parameters k_1, k_2 and k_3 we empirically determined for selecting candidate regions. A possible way to improve is to use training samples to determine k_1, k_2 and k_3.

Table 2. Segmentation accuracy evaluation of five 3D *C.elegans* images. See section 6 for the explanation of parameters P, P_m, P_e, P_s, P_g, P_c, and P_t.

Stacks	P	P_m	P_e	P_s	P_g	P_c	P_t	P_c/P_t
S_1	538	9	2	16	26	494	528	93.56%
S_2	547	8	2	9	24	512	550	93.09%
S_3	547	6	0	7	20	520	550	94.55%
S_4	520	8	3	12	14	491	530	92.64%
S_5	537	2	0	16	6	515	552	93.30%

ACKNOWLEDGEMENTS

We thank Drs. Xiao Liu and Stuart Kim in the Department of Developmental Biology, Stanford University, for providing the 3D images of *C.elegans*.

REFERENCES

1. R.C. Gonzalez, and R. E. Woods, Digital Image Processing, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 2002.
2. L. Vincent, and P. Soille, Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Trans. On Pattern Analysis and Machine Learning*. Vol. 13, No.6, 583-598, 1991.
3. N. Malpica et al, Applying watershed algorithms to the segmentation of clustered nuclei. *Cytometry*, Vol. 28, 287-297, 1997.
4. V Grau, U.J. Mewes, M. Alcaniz, Improved watershed transform for medial image segmentation using prior information. *IEEE Trans. on Medical Imaging, Vol.23*, No.4, 447-458, 2004.
5. S. Beucher, and F. Meyer, The morphological approach to segmentation: the watershed transform. in *Mathematical Morphology in Image Processing*, Dougherty, E.R., ed. New York: Marcel Dekker, **12**, 433-481, 1993.
6. N. A. Otsu, Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, **9**(1), 62-66, 1979.
7. L. Vincent, Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. *IEEE Trans. on Image Processing*, Vol.2, No.2, 176-201, 1993.
8. H.C. Peng, and E.W. Myers, Comparing in situ mRNA expression patterns of Drosophila embryos, *RECOMB 2004*, 157-166, San Diego, USA, Mar, 2004.