A Novel Method for Splitting Clumps of Convex Objects Incorporating Image Intensity and Using Rectangular Window-Based Concavity Point-Pair Search

Muhammad Farhan, Olli Yli-Harja, Antti Niemistö

Department of Signal Processing, Tampere University of Technology,
P.O. Box 553, FI-33101 Tampere, Finland,
muhammad.farhan@tut.fi, olli.yli-harja@tut.fi, antti.niemisto@tut.fi

Corresponding author: Muhammad Farhan

Email Address: muhammad.farhan@tut.fi

Phone: +358 40 4652111

Fax: +358 3 3115 4989

Postal address: Department of Signal Processing, Tampere University of Technology, P.O. Box 553, FI-33101 Tampere Finland.

Abstract

A novel nonparametric concavity point analysis-based method for splitting clumps of convex objects in binary images is presented. The method is based on finding concavity point-pairs by using a variable-size rectangular window. The concavity point-pairs can be either connected with a straight split line or with a line that follows a path of minimum or maximum intensity on an accompanying grayscale image. Using straight lines can result in non-smooth contours. Therefore, post-processing steps that remove acute angles between split lines are proposed. Results obtained with images that have clumps of biological cells show that the method gives accurate results.

Keywords: *Image segmentation, clump splitting, intensity-based splitting, concavity point, split line*.

1. Introduction

In digital imaging domain, it is often observed that the objects in an image clump together. This might occur due to high density of objects in an image area or objects being extremely close to each other that due to optical projections the objects in the image appear to be touching each other and forming clumps [1]. In some application domains, the objects in a scene being imaged might actually overlap and form clumps, for example, the image of objects moving on a conveyor belt [2]. Our target application area is microscopic imaging of biological cell cultures, where clumping of cells occurs naturally. This is because some cell types, such as yeast cells, and many different bacteria have the tendency to grow in clumps.

In these imaging applications and also in many applications in the field of computer vision, see for example [3] and [4], accurate automated image analysis requires that the clumps are split into their constituent objects. For example, it is necessary to extract single cells from an image in order to study the dynamics of single-cell gene expression [5]. As the objects forming clumps usually have similar intensity values and often inconspicuous edges, the general image segmentation methods fail to separate the individual objects from the clump. Therefore, in high-throughput automated image analysis involving such images, a post-processing step of clump splitting is typically performed after the initial segmentation.

Many of the clump splitting methods found in the literature assume the objects in the image to be convex, see for example [3, 6-11]. Using this assumption they try to find specific points, called concavity points, on contour segments where the object ceases to be convex. Clump splitting is then achieved by joining pairs of such points. When the clumps are complex, it is common for these methods to suffer from under-splitting. Another problem with these methods is that they depend on several user-defined parameters to get the pair for a concavity point. This causes their performance to degrade since it is difficult or even impossible to optimize the parameters to get high overall split accuracy when the image set is large with varying object sizes, shapes and the extent of their overlap. Another important issue that is not addressed well in earlier methods is the fact that there tends to be some holes within the clump when the number of objects in the clump increases. Under-splitting occurs if they are not taken into account while finding the split lines. Moreover, none of the earlier methods take due advantage of the intensity values of the image in order to split the clumps more accurately.

Here in this paper, we present a comprehensive method for splitting clumps based on the concavity point analysis, taking the aforementioned issues into account. We propose a new method for the detection of concavity points. The method uses the definition of convexity to find the maximum curvature points from concave region of the contour. We also propose a novel method which uses variable-size rectangular window to search for the best concavity point-pair. With this technique the dependency on the user-defined parameters is reduced along with an increase in the segmentation accuracy. Also, we have incorporated the prominent corner points on the contour of holes inside clumps to get the complete set of split lines.

Moreover, we have developed an algorithm to follow the minimum/maximum intensity path between two points, to be used for the images where the intensity values can be used as a clue to split the clump. In addition, we present a post-processing technique to be employed in the case when the intensity values cannot be used for finding the split line. It removes the residual objects or the ones that do not conform to the objects presumed in the image based on *a priori* knowledge about the object shapes.

The rest of the paper is organized as follows: In Section 2, we present a review of state-of-the-art clump splitting methods. Section 3 presents the proposed method whereas Section 4 describes the proposed post-processing technique. In Section 5, we present and discuss the quantitative and qualitative results obtained by

applying the proposed method and two other methods reviewed in Section 2 on the synthetic images of cell populations as well as on cell microscopy images. Section 6 concludes the paper.

2. Review of concavity point analysis-based methods

The concavity point analysis-based methods mostly utilize a general step-wise procedure, such as, detecting concavity points, finding candidate split lines and choosing the best split lines [3, 6-13]. An alternative approach is to use concavity points to segment the object contour and to fit ellipses to the contour segments to split the clumps [14-16]. Here, we highlight the deficiencies of those concavity point analysis-based methods which by far produce the best results to the best of our knowledge. Moreover, most of these deficiencies have been rectified in the method proposed in this paper.

The method by Kumar et al. [8] fails to find all concavity points when there are multiple concavity points in a concavity region. Moreover, its expression for saliency, a feature used to shortlist candidate split lines, gives a highly nonlinear relationship between the depth of concavity and the length of split lines. This results in long invalid split lines when allowing a reasonable length split line for a concavity point with less concavity depth. Also, the directional vector used to give the orientation of a concavity region is defined in such a way that in many cases it does not match the perceived orientation of the concavity. This method is modified by us in [6] to achieve improvement in the problematic areas. However, it suffers from parameter dependency and also results in over-splitting along with producing objects irrelevant to the actual objects present in the image.

The method by Wang in [3] uses polygonal approximation to smooth the object contour. This may deform the shape of the objects, and can even result in a loss of concavity points which have small concavity depths. The method picks some significant concavity points as candidate points and splits the clump from them. However, as the objects are split, some of those significant concavity points disappear, causing loss of potential split lines between them and some non-significant points. To make a split line the method poses the requirement that the concavity region of the second concavity point should lie within the cone formed by the extension of the vertices of the first concavity point towards it. However, using the cone can be misleading in cases where the angle between the vertices is small or there are two concavity points in the cone. In the latter case, the method prefers the concavity point with higher degree. However, the length of the split line is a

much more significant parameter to decide between the split lines. The method performs morphological opening of holes through a minimum distance path found between the corner of the hole and a point on the object contour. However, not all the corner points of a hole should have a split line through them, also the minimum distance path may not yield the optimal split line.

The approach used by Liang in [9] for detection of concavity points causes invalid concavity points since thresholding the angle near concavity alone is not a good criterion without considering the depth of the concavity. It accepts the shortest possible path of lowest possible gray values provided that the ratio of the length of the large and the small object contours is less than a predefined threshold. However, this may lead to false split lines because a path of lowest possible gray values between a concavity point and a contour point is not optimal unless it is found using some directional search.

The method in [13] finds concavity points on the basis of distance between potential concavity point-pairs, from inside and along the contour, and not on the basis of concaveness. However, sometimes a point that is somewhat further is the best pair for a concavity point rather than the nearest point. The method then finds a split path in the intensity patch formed by a rectangular window between concavity points. However, such a split path based on image intensity is usually a curve which tends to go outside that window, and therefore a directional search is needed to find the path. The method in [12] uses watershed segmentation to get initial clump splitting and then eliminates false split lines resulting due to over-splitting. Then it applies concavity point analysis-based clump splitting which is similar to the method in [8] and has many of the same issues.

The methods based on concavity points and ellipse fitting [14-16] start with performing polygonal approximation of the contour and then detect concavity points by using the angle between the vertices or the changing angle of tangents to the contour. The concavity points are used to segment the contour and ellipse fitting is performed on those contour segments. However, these methods are typically computationally complex and parameter-dependent [8, 17]. Moreover, the ellipse fitting is not able to split complex clumps into individual objects because of the absence of contour segments inside the clumps and due to unknown number of objects in the clump. Moreover, when the image set contains objects of varying shapes and sizes, these methods may not be able to perform accurately.

3. Method

This section describes the proposed novel clump splitting method, the steps of which are delineated by the flowchart in Figure 1. The method operates on binary images obtained after initial segmentation; however, intensity values of the image can also be used as additional information for finding the split lines. The method attempts to separate all the individual objects in a clump at once. However, it is iterative in the sense that it repeats clump splitting on the objects that could not get split in the initial phase. The advantage of performing splitting at once, besides being faster, is that a concavity point can have more than one split line through it, whereas when each split line is considered one by one, once a split line is drawn then another potential split line may be lost. This happens because the objects get separated and the two points involved in that split line do not exist anymore so as to be considered as pairs for some other concavity points.

FIGURE 1

3.1. Image Pre-processing

Sometimes there exist holes within the clumps of objects which are formed due to the clustering of several objects together, Figure 2(a) shows an example of such a case. In order to accurately split such clumps, prominent points on their contour (for example, blue squares in Figure 2(b)) should be paired with other such points or the concavity points (for example, red squares in Figure 2(a)) to form the split lines. All such holes and their corresponding prominent points should be found in the very beginning so that during the pair-search for a particular concavity point those points are also considered. The prominent points are found by analyzing the points on the contour segments of holes. Within a particular contour segment, the contour point having the largest distance from its corresponding imaginary local chord is the desired point provided that the midpoint of that chord lies on the background.

FIGURE 2

3.2. Concavity point detection

The next step is to detect all those points on the contour of the clumped object which are the points of intersection of two touching objects. Since it is assumed in the later step that every obtained concavity point is valid, concavity point detection needs to be performed carefully such that no single point that is taken was a

result of boundary irregularities. A predefined minimum concavity depth value is employed to serve this purpose. There are several methods [7, 9-11] in the literature that are used to detect the concavity points in clumped objects. However, they often fail to determine all the concavity points present in a clump. Here we develop a new technique which is very simple and effective, and is based on the definition of convexity.

The idea is to take two contour points and imagine a line between them, see for example, blue or yellow lines in Figure 2(a) where green circle indicates the initial point of the line. Taking too distant or close points may cause failure in detection of valid concavity points. However, it is observed through experiments with different images that a line between the end points of a 20 pixels long contour segment is applicable to clumps of objects of any shape with the contour length greater than 20 pixels. If that entire line resides inside the object (such as blue lines in Figure 2(a)), the convexity of the object is assured along that segment of the contour. In contrast, if the line passes through the background (such as yellow lines in Figure 2(a)) then there lies a concavity point along that segment of the contour. In this latter case, the next step is to find the distance of each of the points on that contour segment with their respective imaginary local chords(such as red line in Figure 2(a)) such that the midpoint of the respective chord lies on the background, otherwise that particular point is ignored. The imaginary local chord is obtained by joining sixth adjacent contour point on either side of the current point. Finally the point which gives the maximum of the distance is selected as the concavity point provided the distance value is larger than or equal to 2 pixels ($\sqrt{2}$ being the distance between diagonal pixels).

In the case of satisfying the convexity criterion, the third adjacent point to the previous first point is taken as the initial point, whereas if the convexity criterion is not satisfied, the third adjacent point to the previous second point is taken as the initial point of the next contour segment. Again a 20 pixel long contour segment is taken and the process is repeated until the starting point is reached. The concept is illustrated in Figure 2(a).

3.3. Concavity point-pair search

There are $\binom{N}{2}$ possible split lines for a clump with *N* concavity points. Even if the intersecting lines and the lines passing through the background are omitted, the remaining split lines may not necessarily be all valid. Therefore, the preferred approach is to find the best split line or the concavity point-pair for each of the

concavity point in the clump, rather than analyzing every possible split line. In a previous method in [6], we used different features to choose the best split line for every concavity point from the list of possible split lines. However, we observed that there are cases in which it is very difficult to decide between two possibilities of the best split lines because of using the same parameters for the whole image set.

Thus in order to focus on finding only the best split lines individually for each concavity point and also to eliminate user-defined threshold values, we take into consideration the fact that the split lines should be found within a specific region along the directional vectors (purple arrows in Figure 3) associated with the concavity point. The directional vector should ideally bisect the region in the vicinity of the concavity point. This is realized by defining it as a vector with its tail on the midpoint of the imaginary local chord (red line in Figure 2(a)), corresponding to that concavity point, and originating towards the concavity point. This concept leads us to start searching for the pair of a concavity point in the area along and on either side of its directional vector giving rise to a variable-size rectangular window as illustrated in Figure 3.

FIGURE 3

The idea is that first the directional vector associated with a concavity point is found. Then, two points are picked on the object contour, one on either side of the concavity point and both equally distant to it, and a line is formed between them. This line is then extended on either side using those two points along with the directional vector. The length of this line is referred to as width of the window *w*. The reason behind extending the line is that when the region around a concavity point is narrow and deep then the window formed by using the contour points would always be thin and at an undesired location, see for example, Figure 3(a). Thus formation of the window should be independent of how the contour progresses beyond the concavity point. Figure 3(b) illustrates the point where gradually increasing window width leads to the successful detection of the concavity point-pair (red window).

The other two corner points of the rectangular window are found at a distance *h* on the other side of the contour and in the direction of the directional vector using basic trigonometric relations. That distance *h* is a parameter which defines the length of the window and depends on the maximum length of the split line that is allowed in a certain image set. Once the coordinates of the four points are obtained, all the pixels within the rectangular area bounded by them constitute a rectangular window. This window is then used as a mask to

search for the pair of the concavity point under consideration. The idea here is to use variable size window. Therefore, initially a small window width and a comparatively large window length are chosen based on the prior knowledge about the object size in the image set. The window width is gradually increased until a concavity point is found inside the window or the window width approaches its maximum value.

The purpose of using small window width at the beginning is to avoid the case of having two concavity points inside the search window. Intuitively, split lines must be as short as possible; therefore, even if there exist two concavity points in the window, the concavity point having the smallest distance from the subject concavity point is accepted as its pair. If the window width reaches its maximum value and no concavity point is found inside the window, then the window length is iteratively increased until a concavity point is found inside the window or the maximum window length is reached.

This whole process is repeated for every concavity point present in the clumped object and a list of the concavity point-pairs is formed. Next, any such concavity point is found that was left without being assigned a pair. Often the pairing concavity point for such a concavity point was discarded in the initial concavity point detection phase due to lack in concaveness or due to boundary irregularities. In such cases and in the case when there is only one concavity point in a clump, a line is drawn from that concavity point to a point on a segment of object contour in the direction of the directional vector associated with the concavity point. The point is chosen from a contour segment such that, among a certain number of points in that segment, it has maximum distance from its corresponding local chord provided the midpoint of the chord lies on the background. This point-pair is also added to the list of concavity point-pairs.

During this whole process, the prominent points on the contour of the holes are also considered while searching the pairing point for a particular concavity point. However, there are cases in which the clumps are so complex that there are many such holes lying inside them. In such cases, often a split line is also realized by joining a pair of prominent points belonging to two different holes lying nearby. Therefore, like individual concavity points, the pairing points for all the prominent points of all the holes of an image are also found so that all valid split lines are identified.

3.4. Split line formation

Once the concavity points are obtained, there can be two different approaches for obtaining the split lines: making a list of concavity point-pairs, as described in the previous section, and joining them through straight lines, or finding a path of minimum/maximum intensity from a concavity point to a point in another concavity region or to an already drawn split line. The former approach is appropriate when the image intensity values cannot be used as conclusive evidence for determining the split path. Even though this approach may separate the clumps into their correct number of constituent objects, it may not give the correct individual object areas. Moreover, for some complex clumps straight lines can produce erroneous results, since for a given set of images long split lines may be allowed, but making such long straight lines may not match the underlying objects despite being algorithmically correct, see for example Figure 4(c).

FIGURE 4

Furthermore, this approach may lead to under-segmentation. This can be avoided by using an iterative procedure in which it is checked if the result of the first round of clump splitting yielded larger objects. A scaled value of the constraint for the smallest allowed object is used to decide if an object requires further processing. Clump splitting is then performed iteratively on such objects while maintaining that the over-splitting is not achieved.

When the original image has discernible intensity variations along the region where the objects seem to merge together, see for example Figure 4(a), then we should rather find the minimum/maximum intensity path to effectively split the clumps than opting for a straight line between concavity point-pairs. Here, we deduced an algorithm which finds the minimum/maximum intensity splitting path using the directional vector associated with the concavity point which guides the algorithm in the right direction and prohibit it from straying.

FIGURE 5

Here, we use a 3x3 mask centered at the current point, starting from the concavity point, to locate the next low/high intensity valued pixel in the intensity image. Depending on the direction of the directional vector associated with a concavity point, one of the four 3x3 search masks, illustrated in Figure 5(a), is used. For example, if the angle of the directional vector with respect to the horizontal line is in the range $0 < \theta \leq \pi/2$

then the top-right mask is used. Similarly, for the case, $\pi/2 < \theta \leq \pi$, the top-left mask is used and so on. Notice the equality and inequality condition while choosing the mask as otherwise the search might go in the wrong direction. Since the directional vector remains fixed, once a search mask has been chosen for a concavity point, it is used unchanged. Now, if this new point with lowest/highest intensity in the 3x3 neighborhood does not correspond to a background pixel in the binary image, then it is assigned the background pixel value and made the current point. The center of the mask is put on it and the procedure is repeated until a point is reached which corresponds to a background pixel in the binary image.

The end point of the split line found in the procedure is compared with the points of all the concavity regions for the clumped object, and only if it is part of one of them or part of the image border then the line can be retained, otherwise it is discarded. This ensures that the line is made between the subject concavity point and a point on the concavity region in the direction of the directional vector or with a point at image border. In any case, the size of each objects resulting from this new line must be larger than the value for the smallest allowed object in the image, otherwise the line is discarded. In this way, the obtained split lines resemble a lot to the lines that an expert would draw. Consequently, it helps in splitting complex clumps more accurately with better realization of actual individual object areas. Figure 4(d) illustrates the case where the usage of intensity information results in accurate clump splitting, thus giving the actual object areas as compared to the ones obtained by splitting using straight split lines in Figure 4(c). Figure 5(b) shows an example of using the mask (top right mask in Figure 5(a)) to find the path of minimum intensity (black curve in Figure 5(b)) from a concavity point to another one, which is a curve rather than the straight line (red line in Figure 5(b)). Colors along the minimum intensity path refer to steps of the path finding procedure.

4. Post-processing technique

When the intensity information is not used, the clump splitting method described in the previous section defines mere straight lines between the concavity point-pairs without considering the relationship that can exist between the split lines. For example, sometimes there are two split lines through a particular concavity point making an acute angle between them, as shown in Figure 6(a). Moreover, sometimes the other two concavity points involved in those two split lines also share a split line between them which results in a

triangular object in an image, as illustrated in Figure 6(b). If there is prior information about the object shapes and also the objects are known to have smooth boundaries, then both those cases lead to an output image with objects not corresponding to the topology of the underlying image objects. Moreover, in the latter case there is an extra object which is not in accordance with the shape of the objects actually present in the image. Therefore, we need to post-process the resulting image from the initial clump splitting to make the final split lines mimic the manually obtained split lines.

FIGURE 6

Here, we propose a post-processing technique to solve these cases. The process begins with finding the two cases by checking the degree of all the concavity points present in the object. The term degree is used here to specify the number of split lines passing through a concavity point. By going through the list of concavity point-pairs, such concavity points are found whose degree is two. Then the other two concavity points are taken which share the split lines with the first concavity point and it is checked if there exist a split line between them or they do not share any split line with any other concavity point. Once either of these conditions get fulfilled then a triangle is formed between the three points, if it was not already there, and the centroid of that triangle is found. After finding the centroid, the concavity point-pairs formed by those three concavity points are removed from the initial list and are replaced with three point-pairs each involving the centroid and one of the three concavity points. Figure 7 shows the output of the post-processing step for our example cases of Figure 6.

FIGURE 7

Sometimes it might also happen that a concavity point has three split lines passing through it, see for example Figure 8(a). In such a case, those three lines can be perceived as two pair of lines emerging from that concavity point. Then the pair of lines which give smaller of the two angles are analyzed. Associated with those two split lines are the two other concavity points. If the degree of only one of those two concavity points is two then the split line involving that concavity point is discarded from the list. If the degree of both the concavity points is two then the line involved in the wider of the two angles is discarded. If neither of the two concavity points have degree two then the normal post-processing is performed one after the other for the two pair of lines. Figure 8(b) shows the result of the post-processing applied on the image of Figure 8(a).

FIGURE 8

5. Results and Discussion

5.1. Image acquisition and Benchmark image set generation

Validation of image analysis methods is traditionally performed by comparison of results obtained by them with the ground truth created by manual analysis. Manual creation of ground truth, however, is time consuming, laborious and observer-dependent, especially in the case of high-throughput microscopic image analysis where we have very large sets of images. Therefore, the manual validation becomes impractical. Instead a benchmark set of synthetic images having varying properties mimicking the microscopic images, like the ones generated by SIMCEP tool [18] can be used.

Here we use both real microscopy images as well as synthetic images to evaluate our method. We have two different test cases based on whether the intensity information is usable for splitting or not. For the case of utilizing image intensity for splitting, we use the microscopic images and the corresponding manually obtained ground truth results. For the other case we use benchmark synthetic image set for which the ground truth information is available. This can be considered in a way that the first set performs both qualitative and quantitative evaluation whereas the second set gives the quantitative measures of our method.

For the first test case, Case I, we first acquired two sets of bright field images of the budding yeast *Saccharomyces cerevisiae* cells of varying sizes and shapes with a Leica TCS SP2 microscope. For each image, z-stacks comprising 20 images were captured using a 100X oil immersion objective (NA 1.40) but only one of them is included in the test image set. This image was selected by first finding the best in-focus image from these z-slices by using the Tenengrad method [19]. Generally, a slightly out-of-focus image is chosen because of its assistance in giving better segmentation accuracy. Therefore, the image just below (about 300 nm) the best in-focus slice was selected into the test image set. Segmentation of the images was carried out using the method from [20].

Another set for Case I is obtained from Saccharomyces Cerevisiae Morphological Database (SCMD) [21]. The set contains fluorescent images of budding yeast *S. cerevisiae* which provides an ideal scenario for testing the method against varying object sizes. The image set contains more than 300 images but since the analysis is to be performed manually, due to unavailability of ground truth, we chose to use just the first 40

images for obtaining quantification measures. Segmentation of the images is carried out by a local thresholding method that is based on the classic threshold selection method by Otsu [22].

TABLE 1

For the second test case, Case II, that is to test and validate the proposed method, we used the benchmark set of synthetic images of cell populations with realistic properties generated with the SIMCEP simulation tool [18]. They were generated using the package of files, downloadable from: http://www.cs.tut.fi/sgn/csb/simcep/, and by varying the parameters. The entire set contains simulated images of cell populations with the corresponding ground truth images in which the cells are represented as binary markers which are used for validation. The idea here is to create such image set which contains images with cell clumps of varying sizes and complexity. Therefore, the generated images consist of overlapping nuclei with three different values of clustering probabilities. For each clustering probability we used eight different values for the amount of overlap and simulated 50 images (altogether 1200 images constituting 24 image sets), each of which contains 200 cells with approximately 10 cell clusters per image. Table 1 shows the necessary parameters and their values to be used for generating the image set.

5.2. Performance evaluation parameters

The quantitative performance evaluation is performed using precision and recall analysis. We obtained true positives (TP), false positives (FP), and false negatives (FN) and the precision (PR) and recall (RC) are then obtained by

$$PR = TP/(TP + FP) \; ; \quad RC = TP/(TP + FN). \qquad (1)$$

A high value of PR implies that a high percentage of the objects detected by the method are actually the objects of the ground truth image. It decreases once the method detects objects not actually there in the ground truth. On the other hand, a high value of RC specifies that a high percentage of the objects of ground truth image are detected by the method. Furthermore, we use F-measure (FM) [23] which can be obtained by

$$FM = 2/(1/PR + 1/RC), \qquad (2)$$

and is considered to be a more robust measure of segmentation accuracy.

5.3. Results and discussion

The proposed method was applied on all the image sets of both the test cases. For Case I, that is, the case where image intensity is useful in splitting clumps, we performed manual analysis on all the image sets to obtain the TP, FP, and FN values and calculated the PR, RC and FM measures. The first two sets contained bright field microscopy images of yeast cells and the third set contained fluorescent microscopy images of yeast cells obtained from SCMD [21]. It is worth mentioning here that there is a difference between the images of the first set and the other two sets in that they provide the cases when the path of minimum and maximum intensity, respectively, is searched for between the two points to split the clumps. We additionally employed the non-intensity-based method (indicated by NI in Table 2) on these image sets to compare its performance with the one obtained by employing image intensity (indicated by I in Table 2). Table 2 lists the results for this test case.

TABLE 2

The results of applying the proposed method on a bright field microscopy image of yeast cells (an image from Set 1) are shown in Figure 9 whereas Figure 10 depicts the results of applying the proposed method on fluorescent microscopy image (an image from Set 3). The quantitative values of Table 2 manifest that the results obtained from the proposed method are accurate irrespective of whether or not the image intensity is used. Moreover, from Table 2, Figure 9 and Figure 10, it is clear that the proposed method gives better quantitative as well as qualitative results when the image intensity is used. However, it is also clear that the results from this approach depend on accuracy of the initial image segmentation. It is evident from both the Figures that the non-intensity-based method struggles in the situations when the clumps are either very complex or touch the image borders, causing some of the concavity points to be missed. However, its performance is still very promising in that it gives such quantitative measures which were not previously achievable.

For the second test case, we evaluated the proposed method using the benchmark image set for which we have the binary images containing masks representing the cells in the ground truth. In addition, we also applied the method from Kumar et al. [8] as well as our previous method [6] on these images to compare the proposed method against them. The analysis was performed by measuring the performance parameters after

applying these methods on the simulated images and comparing the results with the ground truth. The obtained performance parameters for 24 image sets are presented in detail in Table 3 where subscripts 1, 2, and 3 stand for proposed method, methods from Farhan et al. [6], and Kumar et al. [8] respectively. Each entry corresponds to the overall value for the set of 50 images (the number of cells is 10 000).

It is clear from Table 3 that our method outperforms the other methods proving its significance in resolving complex clumps. Table 3 shows that the proposed method performs accurately when the probability and the amount of overlap are small. In this case, the main deviation in the results is caused by the amount of overlap. When it is around 0.3 there is not much difference between results for different probabilities of overlap. However, when the amount of overlap increases from 0.35 the increase in probability of overlap causes degradation in the performance of the method. Nevertheless, the F-measure for the worst case (probability of overlap = 0.6, amount of overlap = 0.5) is 0.91, against 0.81 for our modified method and 0.58 for method by Kumar et al. The F-measure of 0.91can still be considered as high accuracy especially considering that accurate splitting of the clumps for such images with heavy overlapping is not always possible even for a human observer. Moreover, this performance may improve even further if image intensity can be used as the evidence of the split path as we have observed in the earlier case that the splitting of complex clumps is efficiently done when image intensity is used. Figure 11 shows a representative image from Case II and the result of applying the proposed method on it. It manifests the accuracy of our method in splitting complex clumps. However, it can also be seen that there is some artifact (top left quadrant of the Figure 11 (d)) which can be easily dealt with by using the image intensity information.

TABLE 3

It must be emphasized that, unlike the other methods, the proposed method does not need any user-defined parameters, nor does it depend on predefined threshold values of features. For example, the method in [8] requires threshold values for concavity depth, Saliency, CC and CL alignment etc. Similarly, the method in [3] requires threshold values for angles and lengths to find the degree of concavities for their classification. Moreover, the method in [9] uses threshold values for angle of concavity, length of split line, ratio of the longest to the shortest contour of objects resulting from splitting etc. However, in our method, since the window size is varied until a pairing point is obtained, or in the case of using image intensity, only the

directional vector of a concavity point is used to find its pair, so there is absolutely no need for any parameter values for point-pair selection. We only applied a minimum object size constraint to prevent the method from splitting smaller objects. Therefore, before applying the split lines to a clump it is ensured that they do not result in a smaller object otherwise the lines are discarded. The constraint also helps in deciding whether another phase of clump splitting is needed for larger non-convex objects. This constraint is obtained for every image set by looking at the size of the smallest allowed object in that set.

One of the problematic cases is also the clumps touching the image borders. In many image processing applications objects that touch the image borders are removed as a preprocessing step. However, we did not remove them, because, when images have large clumps (see for example Figure 11 mid- and bottom-left), that would result in many valid individual objects (cells) being removed unnecessarily. Another reason that we included the objects touching the image border is to show the promising results achieved by our intensity-based splitting. It is performing remarkably well in those areas where the concavity point is lost either due to complex clumps or due to being outside the image area because of having resulted from the objects touching the image border (see for example, in Figure 9 objects at bottom-right as well as in Figure 10 objects at top-right and bottom left). Therefore, our method processes such objects, and in most cases, even when the image intensity is not used, is able to resolve individual objects from such clumps that are away from the image borders. However, in the case of not using image intensity, due to the incomplete information near the borders of the image, object splitting is sometimes not accurate in those parts. In certain applications it might be reasonable to remove any objects that touch the image border after the clump splitting step. However, for the sake of showing the raw results from the clump splitting method we have not done so here.

FIGURE 9

FIGURE 10

FIGURE 11

6. Conclusion

We presented a novel non-parametric concavity point analysis-based clump splitting method which takes into account holes in the clumps and if possible the image intensity too to find the split lines. In the case of not utilizing image intensity, a rectangular window mask is used for finding the pairing points of a split line. This makes the method independent of user-defined parameters even if the image set is large and contains objects of variable sizes. A post-processing step using *a priori* knowledge about shape of the objects ensured that the final image contains objects conforming to the ones present in the actual image. Moreover, when the image intensity can be used as a clue for finding the split lines, a split line is obtained for every concavity point using its directional vector which guides the search for the splitting path of minimum/maximum intensity. The advantage of this approach is that it can accurately split complex clumps besides producing the output similar to the one obtained by a human observer.

Quantitative and qualitative measures illustrate the outstanding performance of our method for diverse sets of images having clumps of varying objects sizes and the probability and amount of overlap. Although the method is non-parametric in nature, a minimum object size constraint is used for a particular image set to restrain the method from splitting smaller objects. Even though the target application of the method was microscopy images containing clumps of cells of convex shape, it can be applied to a wide range of applications with images containing clumps of any convex objects.

7. Acknowledgements

This work was supported by the Academy of Finland (application number 213462, Finnish Programme for Centre of Excellence in Research 2006-2011; application number 121830, Post-Doctoral Researcher's Project 2008-2010). We are extremely thankful to anonymous reviewers for their critical evaluation and useful comments and suggestions which helped us improve the manuscript.

References

[1] O. Schmitt, M. Hasse, Radial symmetries based decomposition of cell clusters in binary and gray level images, Pattern Recognition 41 (6) (2008) 1905-1923.

[2] D. Balthasar, T. Erdmann, J. Pellenz, V. Rehrmann, J. Zeppen, L. Priese, Real-time detection of arbitrary objects in alternating industrial environments, in: Proceedings of the12th Scandinavian Conference on Image Analysis, 2001, pp. 321–328.

[3] W.X. Wang, Binary image segmentation of aggregates based on polygonal approximation and classification of concavities, Pattern Recognition 31 (10) (1998) 1503-1524.

[4] G. Zhang, D.S. Jayas, N.D.G. White, Separation of touching grain kernels in an image by ellipse fitting algorithm, Biosystems Engineering 92 (2) (2005) 135-142.

[5] R. J. Taylor, D. Falconnet, A. Niemistö, S.A. Ramsey, S. Prinz, I. Shmulevich, T. Galitski, C.L. Hansen, Dynamic analysis of MAPK signaling using a high-throughput microfluidic single-cell imaging platform, Proceedings of the National Academy of Sciences of USA 106 (10) (2009) 3758-3763.

[6] M. Farhan, O. Yli-Harja, A. Niemistö, An improved clump splitting method for convex objects, in: Proceedings of the 7th International Workshop on Computational Systems Biology, 2010, pp. 35-38.

[7] G. Fernandez, M. Kunt, J-P. Zryd, A new plant cell image segmentation algorithm, in: Proceedings of the 8th International 7Conference on Image Analysis and Processing, 1995, pp. 229-234.

[8] S. Kumar, S.H. Ong, S. Ranganath, T.C. Ong, F.T. Chew, A rule-based approach for robust clump splitting, Pattern Recognition 39 (6) (2006) 1088-1098.

[9] J. Liang, Intelligent splitting in the chromosome domain, Pattern Recognition 22 (5) (1989) 519-532.

[10] W. Wang, H. Song, Cell cluster image segmentation on form analysis, in: Proceedings of the 3rd International Conference on Natural Computation, 2007, pp. 833-836.

[11] Q. Wen, H. Chang, B. Parvin, A Delaunay triangulation approach for segmenting clumps of nuclei, in: Proceedings of the 6th IEEE International Symposium on Biomedical Imaging, 2009, pp. 9-12.

[12] Q. Zhong, P. Zhou, Q. Yao, K. Mao, A novel segmentation algorithm for clustered slender-particles, Computers and Electronics in Agriculture 69 (2) (2009) 118-127.

[13] H. Wang, H. Zhang, N. Ray, Clump splitting via bottleneck detection and shape classification, Pattern Recognition 45 (7) (2012) 2780–2787.

[14] X. Bai, C. Sun, F. Zhou, Splitting touching cells based on concave points and ellipse fitting, Pattern Recognition 42 (11) (2009) 2434–2446.

[15] G. Cong, B. Parvin, Model-based segmentation of nuclei, Pattern Recognition 33 (8) (2000) 1383-1393.

[16] S. Kothari, Q. Chaudry, M. D. Wang, Automated cell counting and cluster segmentation using concavity detection and ellipse fitting technique, in: Proceedings of the 6th IEEE International Symposium on Biomedical Imaging, 2009, pp. 795-798.

[17] M. Farhan, Automated Clump Splitting for Biological Cell Segmentation in Microscopy using Image Analysis, M.S. Thesis, Tampere University of Technology, Finland, November 2009.

[18] A. Lehmussola, P. Ruusuvuori, J. Selinummi, H. Huttunen, O. Yli-Harja, Computational framework for simulating fluorescence microscope images with cell populations, IEEE Transactions on Medical Imaging 26 (7) (2007) 1010-1016.

[19] Y. Sun, S. Duthaler, B.J. Nelson, Autofocusing in computer microscopy: selecting the optimal focus algorithm, Microscopy Research and Technology 65 (3) (2004) 139-149.

[20] A. Niemistö, T. Aho, H. Thesleff, M. Tiainen, K. Marjanen, M-L. Linne, O. Yli-Harja, Estimation of population effects in synchronized budding yeast experiments, in: Proceedings of the International Society for Optical Engineering, SPIE 2003. Image Processing: Algorithms and Systems II, 5014 (2003) 448-459.

[21] T. L. Saito, M. Ohtani, H. Sawai, F. Sano, A. Saka, D. Watanabe, M. Yukawa, Y. Ohya, S. Morishita, SCMD: Saccharomyces cerevisiae Morphological Database, Nucleic Acids Research 32 (2004) D319-D322.

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

[23] T. Fawcett, “An introduction to ROC analysis,” Pattern Recognition Letters 27 (8) (2006) 861-874.

Figures

Figure 1 – Flowchart delineating the steps performed by our clump splitting method.

The different steps involved in getting the final clump split image. The input to the method consists of original intensity and its binarized image along with the binary images containing holes and labeled objects.

Figure 2 – Image depicting the scenario of clump with holes. Directional vectors and concavity points are also highlighted.

(a) A synthetic image with clump of objects having holes inside the clump. The process of finding concavity points on the left where all blue lines reside inside the object whereas yellow lines pass through the background indicating that the object ceases to be convex there. The points in such contour segments (for example, red squares) which give maximum distance from their local chords (red lines outside object contour in the top) are identified as concavity points. (b) Contour of the clumped object. Black and red arrows point towards the directional vectors associated with the object concavity point and prominent point of holes, respectively.

Figure 3 – Variable-size rectangular window-based concavity point-pair search.

Variable-size rectangular window-based concavity point-pair search. Orientation of directional vectors (purple arrows) associated with the top two concavity points. Rectangular window with varying width and length (shown with dashed blue, green and red lines) aligned in the same direction as the directional vector in order to search for the concavity point-pair. (a) Formation of window with points taken from object contour. (b) Formation of window irrespective of concavity region around the concavity point.

Figure 4 – Illustration of the usage of image intensity for finding split lines.

Illustration of the usage of image intensity for finding split lines. (a) Original bright field intensity image. (b) Segmented image. (c) Result of straight split lines. (d) Split lines obtained by employing image intensity.

Figure 5 – Masks and procedure to find the minimum/maximum intensity path for splitting.

Masks and procedure to find the min/max intensity path to get the split lines. (a) The angle formed by the directional vector associated with the particular concavity point is used to select the appropriate mask. Arrows inside indicate the range of angles corresponding to a particular mask. Black and red pixels are Don't care. (b) Procedure to find minimum intensity path for an example case where the directional vector is such that the top right mask was used to search for the path of minimum intensity (black curve) instead of taking straight line (red). Colors along the minimum intensity path refer to steps of the path finding procedure.

Figure 6 – Post-processing cases after initial clump splitting.

Post-processing cases. (a) Object with two split lines making acute angle between them. (b) A triangle is formed between the three concavity points.

Figure 7 – Result after post-processing for the case of two split lines through a concavity point.

Result after post-processing. The objects of Figure 6(a) and (b) after the application of image post-processing.

Figure 8 – Illustration of the case of three split lines through a concavity point and its post-processing.

(a) A clumped object with the case of three split lines through a concavity point. (b) Resulting image after image post-processing.

Figure 9 – Results of proposed clump splitting method for a bright field microscopy image containing clumps of yeast cells.

(a) A bright field image of yeast cells. (b) Segmented image. (c) and (d) Resulting image after application of the proposed method with and without using image intensity.

Figure 10 – Results of proposed clump splitting method for a fluorescence microscopy image containing clumps of yeast cells.

(a) A fluorescent image of yeast cells clumped together (obtained from SCMD [21]). (b) Segmented image. (c) and (d) Resulting images after application of the proposed method with and without using image intensity.

Figure 11 – Results of clump splitting method for a synthetic microscopy image containing clumps of cells. (a) A synthetic microscopy image generated from SIMCEP simulation tool with cell clustering probability = 0.5 and amount of overlap = 0.45. (b) Segmented image provided by the tool itself. (c) Binary image containing masks representing cells in the ground truth. (d) Resulting image after application of the proposed method on the image in (b).

Vitae:

Muhammad Farhan received the degree of Master of Science (MSC in IT) with distinction in 2010 from Tampere University of Technology (TUT), Tampere, Finland. He has worked as a Research Assistant at the Department of Signal Processing, TUT in 2009-2010. Currently he is working as a Researcher and a PhD student at the same department. His research interests include signal and image processing, biomedical image analysis, and pattern recognition.

Olli Yli-Harja received the degree of Doctor of Science (Technology) in computer science and applied mathematics in 1989 from Lappeenranta University of Technology, Finland. During 1988-1998 he was a research scientist at the Technical Research Centre of Finland, Helsinki University of Technology, and University of Helsinki. During 1998-2001 he was senior researcher at the Institute of Signal Processing in Tampere University of Technology and in 2005 a visiting scientist on University of Texas M.D. Anderson Cancer Center in Houston, Texas, USA. Currently he is a Professor in the Department of Signal Processing in TUT. His research interests include computational systems biology, image analysis, complexity and non-linear filters.

Antti Niemistö received the degree of Doctor of Science (Technology) in signal processing in 2006 from Tampere University of Technology, Tampere, Finland. He has been with the Department of Signal Processing at TUT since 1999. He visited The University of Texas M. D. Anderson Cancer Center in Houston, Texas, USA during 2003-2004. In 2007-2008 he was a Postdoctoral Fellow at the Institute for Systems Biology (ISB) in Seattle, Washington, USA. Currently he is a Research Fellow at the Department of Signal Processing in TUT. His research interests include biomedical image analysis and nonlinear signal and image processing. His current work focuses on developing image analysis methods for microscopy applications in cell and molecular biology, in particular for quantitative analysis of live cells in microfluidic devices.

Author Biography

Muhammad Farhan received the degree of Master of Science (MSC in IT) with distinction in 2010 from Tampere University of Technology (TUT), Tampere, Finland. He has worked as a Research Assistant at the Department of Signal Processing, TUT in 2009-2010. Currently he is working as a Researcher and a PhD student at the same department. His research interests include signal and image processing, biomedical image analysis, and pattern recognition.

Olli Yli-Harja received the degree of Doctor of Science (Technology) in computer science and applied mathematics in 1989 from Lappeenranta University of Technology, Finland. During 1988-1998 he was a research scientist at the Technical Research Centre of Finland, Helsinki University of Technology, and University of Helsinki. During 1998-2001 he was senior researcher at the Institute of Signal Processing in Tampere University of Technology and in 2005 a visiting scientist on University of Texas M.D. Anderson Cancer Center in Houston, Texas, USA. Currently he is a Professor in the Department of Signal Processing in TUT. His research interests include computational systems biology, image analysis, complexity and non-linear filters.

Antti Niemistö received the degree of Doctor of Science (Technology) in signal processing in 2006 from Tampere University of Technology, Tampere, Finland. He has been with the Department of Signal Processing at TUT since 1999. He visited The University of Texas M. D. Anderson Cancer Center in Houston, Texas, USA during 2003-2004. In 2007-2008 he was a Postdoctoral Fellow at the Institute for Systems Biology (ISB) in Seattle, Washington, USA. Currently he is a Research Fellow at the Department of Signal Processing in TUT. His research interests include biomedical image analysis and nonlinear signal and image processing. His current work focuses on developing image analysis methods for microscopy applications in cell and molecular biology, in particular for quantitative analysis of live cells in microfluidic devices.

Table 1 – Parameters to create benchmark synthetic image set from SIMCEP tool.

Parameter settings for creation of benchmark image set containing clustered nuclei with increasing clustering probability and amount of overlap.

Parameter	Value
Probability of clustering	0.4, 0.5, 0.6
Amount of overlap	0.15, 0.2,…, 0.5
Number of Image Sets	24
Images per set	50
Cells per image	200
Total number of cells for 1 set	10 000

Table 2 – Performance parameters obtained from the proposed clump splitting method for microscopy images.

Performance parameters for three image sets of Case I containing 740, 858 and 1242 total number of cells constituting clumps. (See text for abbreviations)

Set	TP	FP	FN	PR	RC	FM
Set 1_I	727	9	13	0.988	0.982	0.985
Set 1_NI	726	15	14	0.979	0.981	0.980
Set 2_I	841	6	17	0.993	0.980	0.987
Set 2_NI	826	8	32	0.990	0.963	0.976
SCMD_I	1219	18	23	0.985	0.981	0.983
SCMD_NI	1198	21	44	0.982	0.964	0.973

Table 3 – Performance parameters obtained from the clump splitting methods for synthetic images.

Performance parameters obtained after application of the proposed method (Subscript 1), Farhan et al. (Subscript 2) and Kumar et al. (Subscript 3) on 24 image sets generated from SIMCEP simulation tool each containing 10 000 total number of cells with or without clumps. Text in column "Set" is interpreted as PN_1ON_2 where P = probability, $0.N_1$ = probability of overlap, O = Overlap, $0.N_2$= Amount of overlap.

Set	PR_1	RC_1	FM_1	FM_2	FM_3	Set	PR_1	RC_1	FM_1	FM_2	FM_3
P4O15	0.999	0.995	0.997	0.985	0.858	P5O35	0.992	0.948	0.970	0.897	0.664
P4O20	0.998	0.992	0.995	0.972	0.802	P5O40	0.990	0.923	0.955	0.871	0.633
P4O25	0.996	0.986	0.991	0.951	0.761	P5O45	0.988	0.897	0.940	0.855	0.633
P4O30	0.995	0.974	0.984	0.935	0.726	P5O50	0.988	0.880	0.931	0.835	0.606
P4O35	0.994	0.960	0.977	0.909	0.698	P6O15	0.998	0.992	0.995	0.977	0.814
P4O40	0.992	0.944	0.968	0.890	0.673	P6O20	0.998	0.988	0.993	0.960	0.754
P4O45	0.990	0.922	0.954	0.870	0.664	P6O25	0.997	0.980	0.988	0.936	0.710
P4O50	0.987	0.896	0.939	0.851	0.652	P6O30	0.994	0.957	0.975	0.912	0.689
P5O15	0.998	0.993	0.995	0.982	0.842	P6O35	0.992	0.937	0.963	0.873	0.619
P5O20	0.998	0.990	0.994	0.967	0.788	P6O40	0.990	0.906	0.946	0.854	0.603
P5O25	0.997	0.984	0.990	0.948	0.729	P6O45	0.989	0.874	0.928	0.831	0.590
P5O30	0.996	0.969	0.982	0.922	0.708	P6O50	0.988	0.842	0.909	0.812	0.583

Figure 1

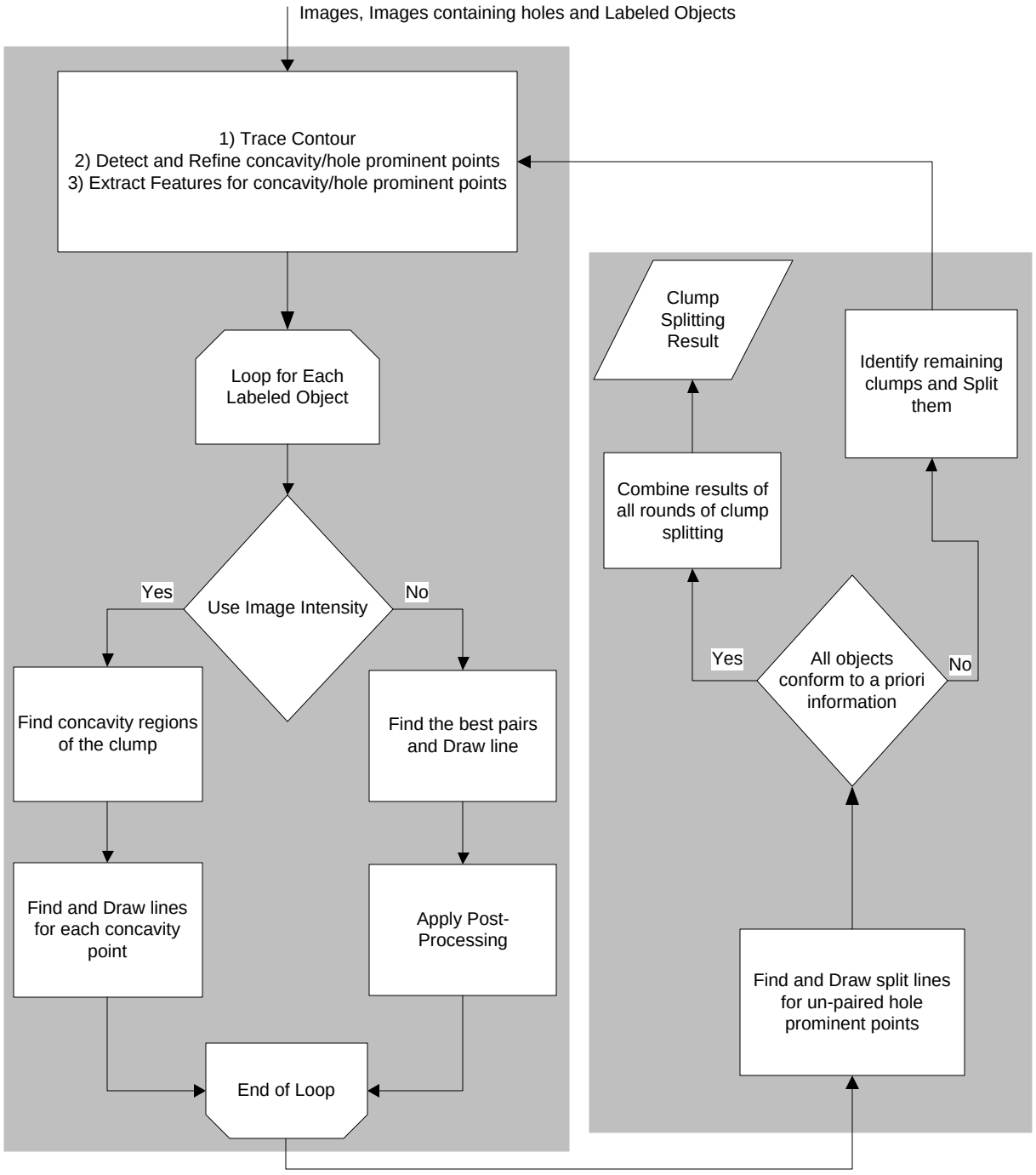
Images, Images containing holes and Labeled Objects
1) Trace Contour
2) Detect and Refine concavity/hole prominent points
3) Extract Features for concavity/hole prominent points
Loop for Each Labeled Object
Use Image Intensity
Yes
No
Find concavity regions of the clump
Find and Draw lines for each concavity point
Find the best pairs and Draw line
Apply Post-Processing
End of Loop
Clump Splitting Result
Identify remaining clumps and Split them
Combine results of all rounds of clump splitting
All objects conform to a priori information
Yes
No
Find and Draw split lines for un-paired hole prominent points

Figure 2a

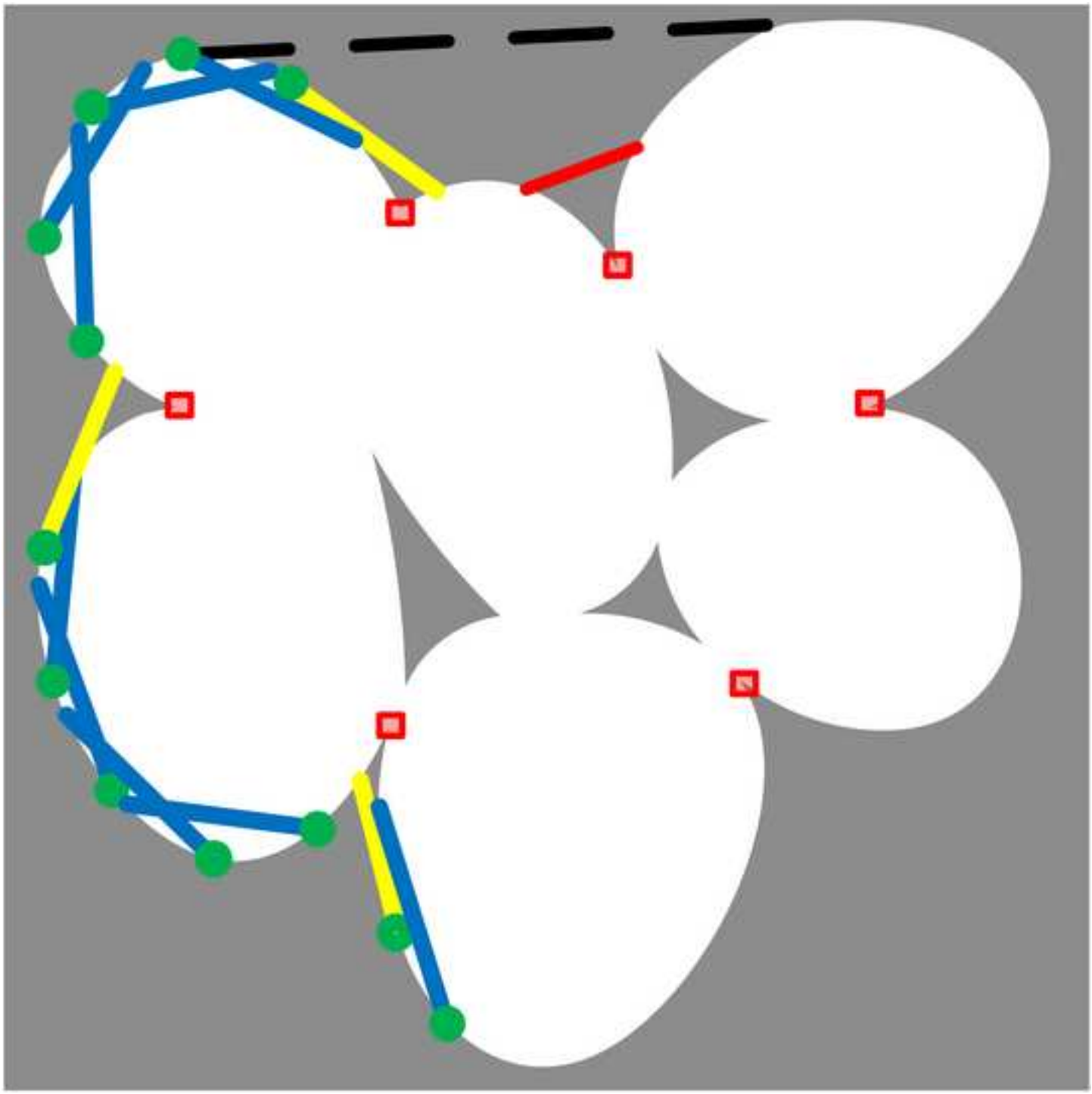

Figure 2b

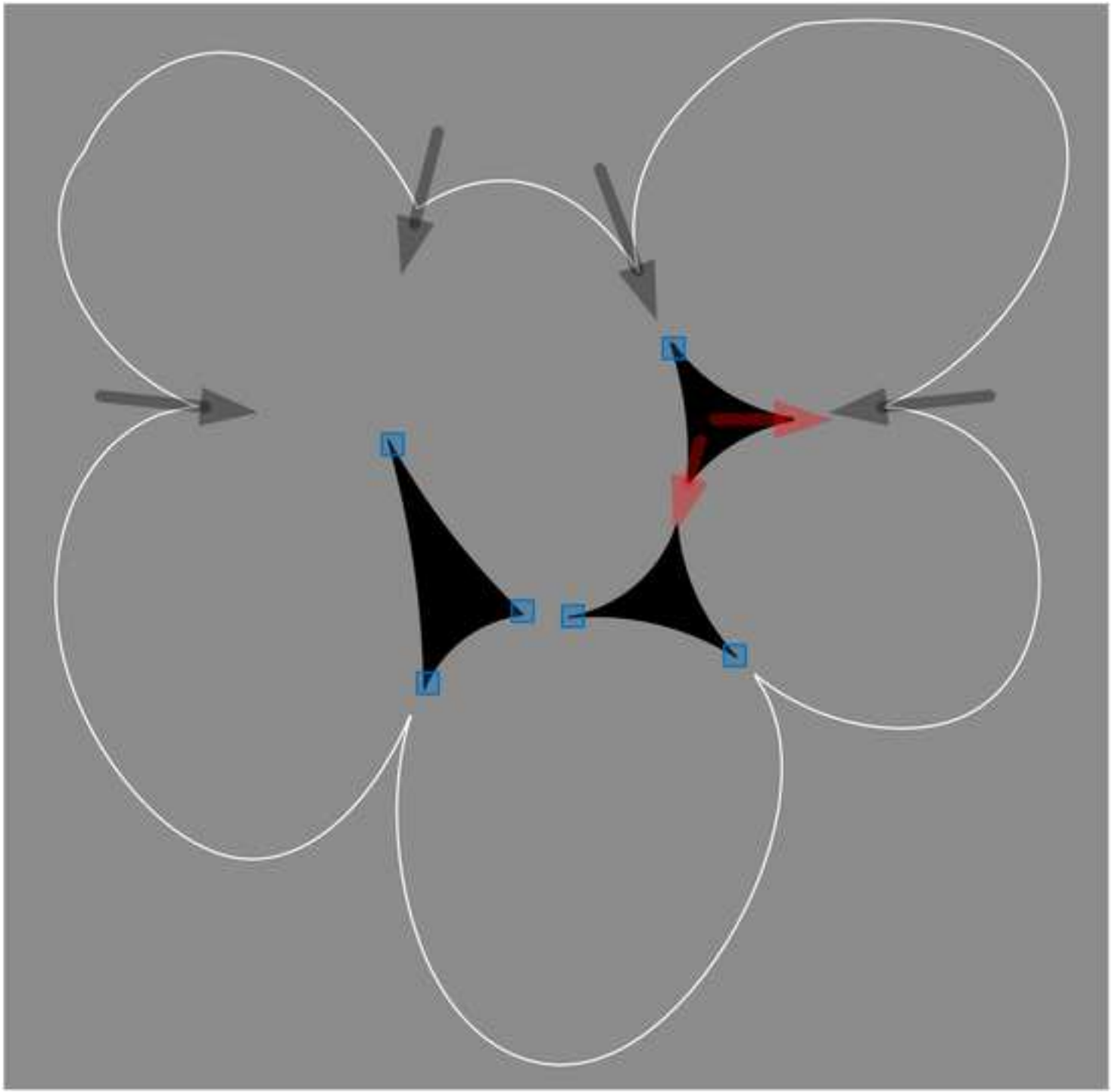

Figure 3a

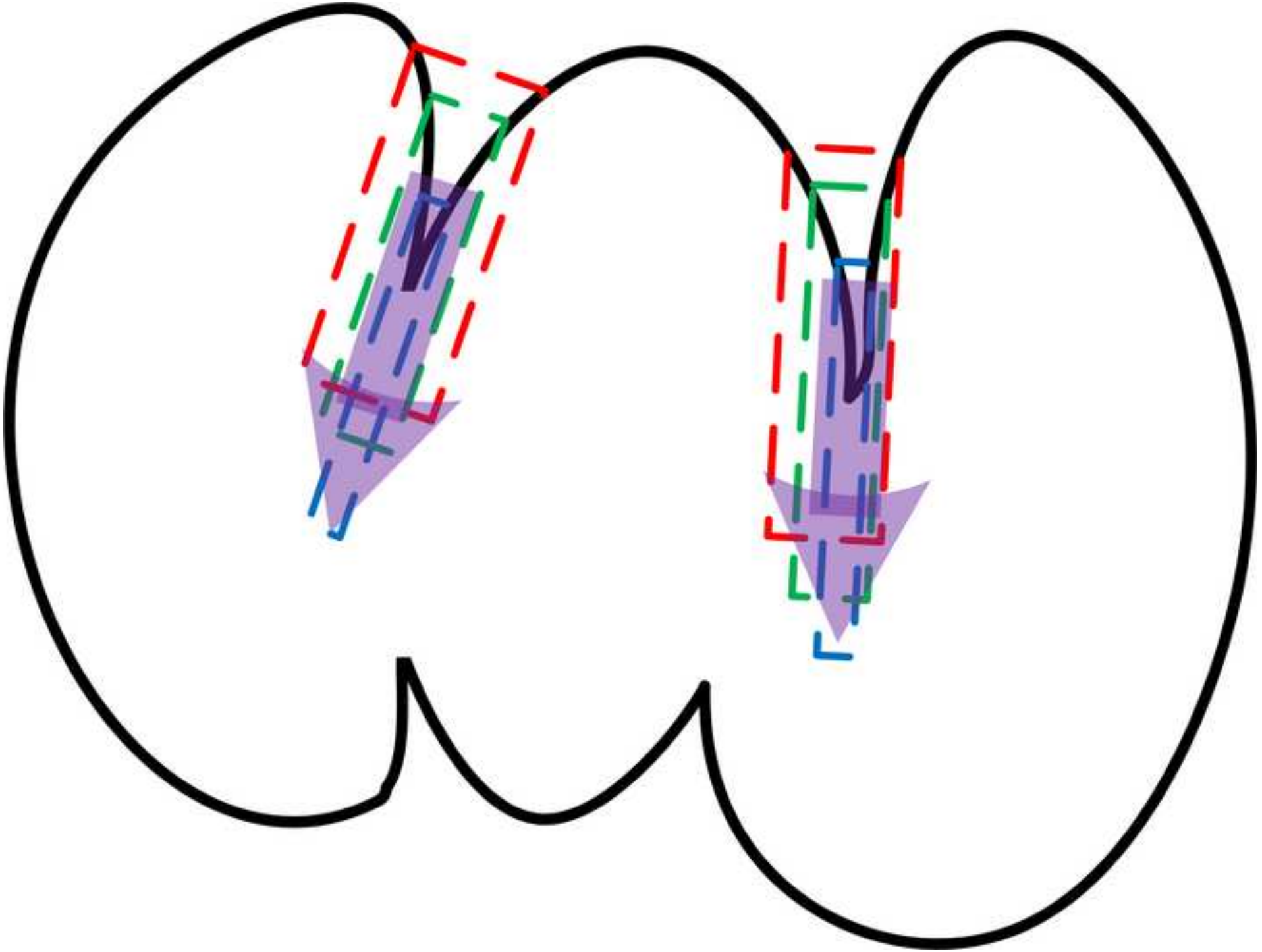

Figure 3b

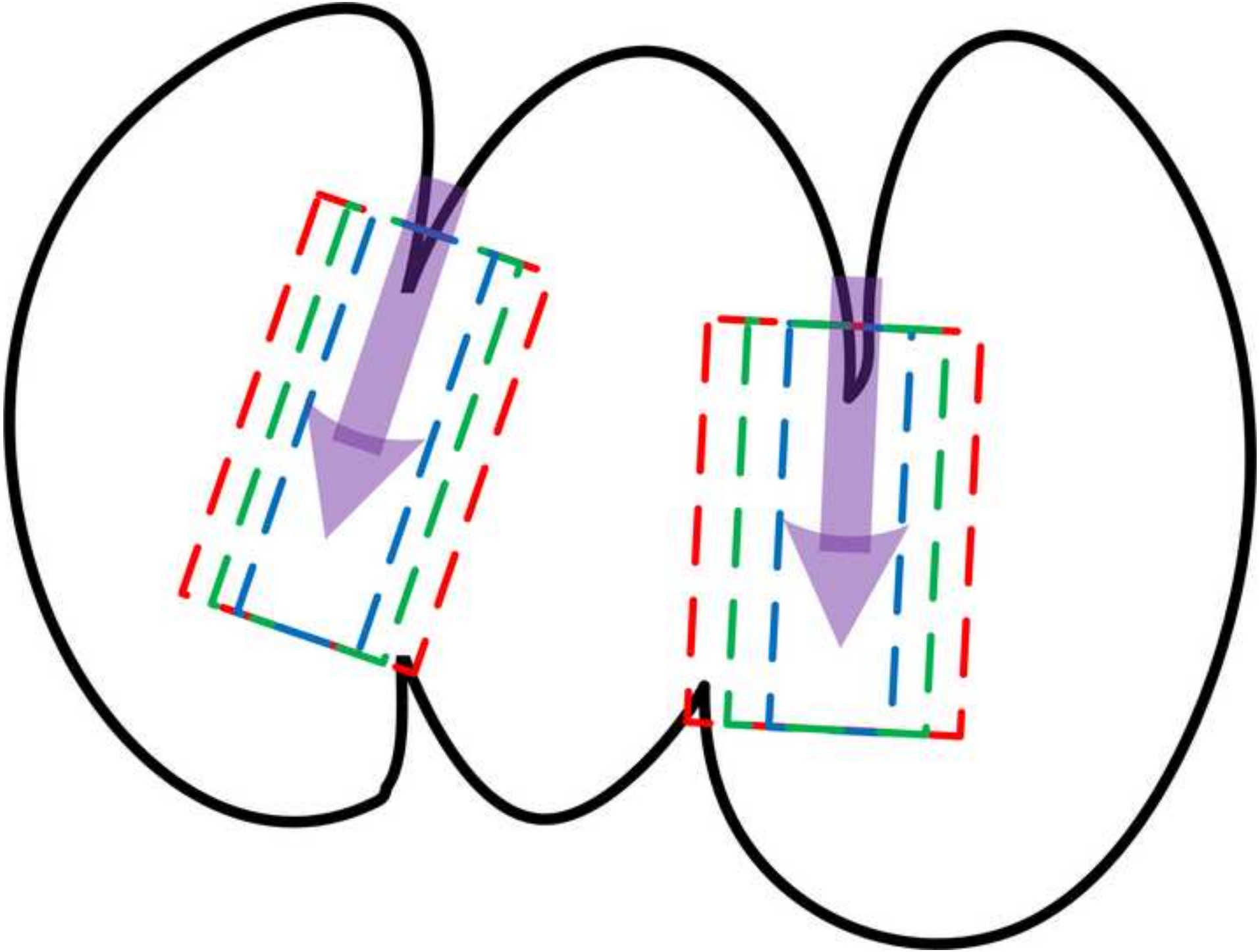

Figure 4a

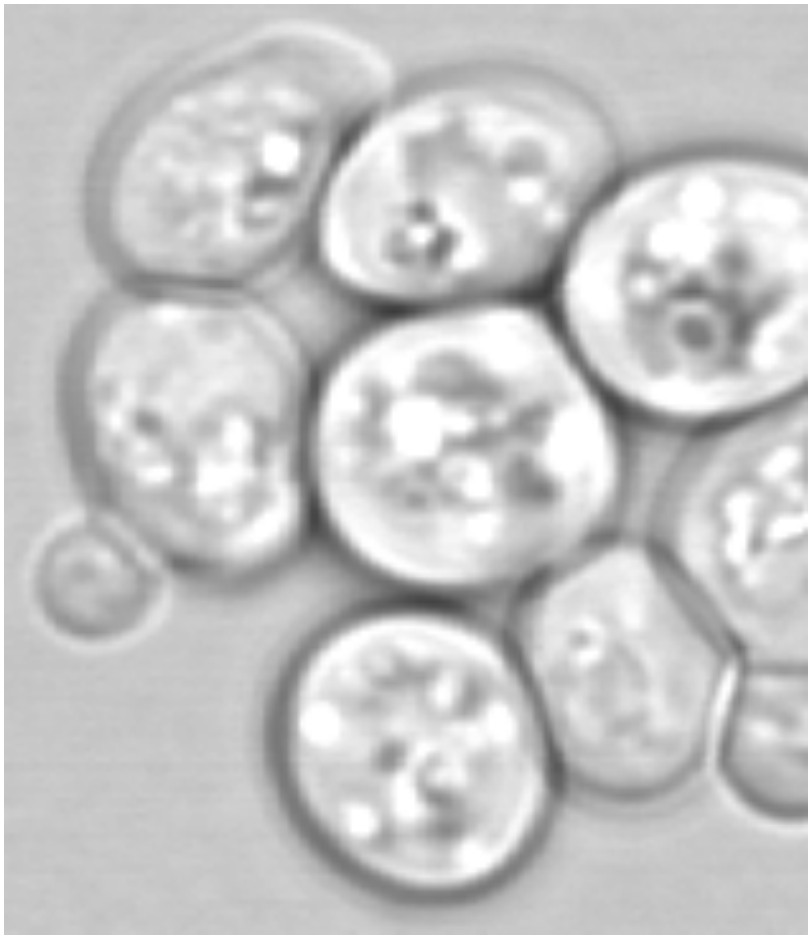

Figure 4b

Figure 4c

Figure 4d

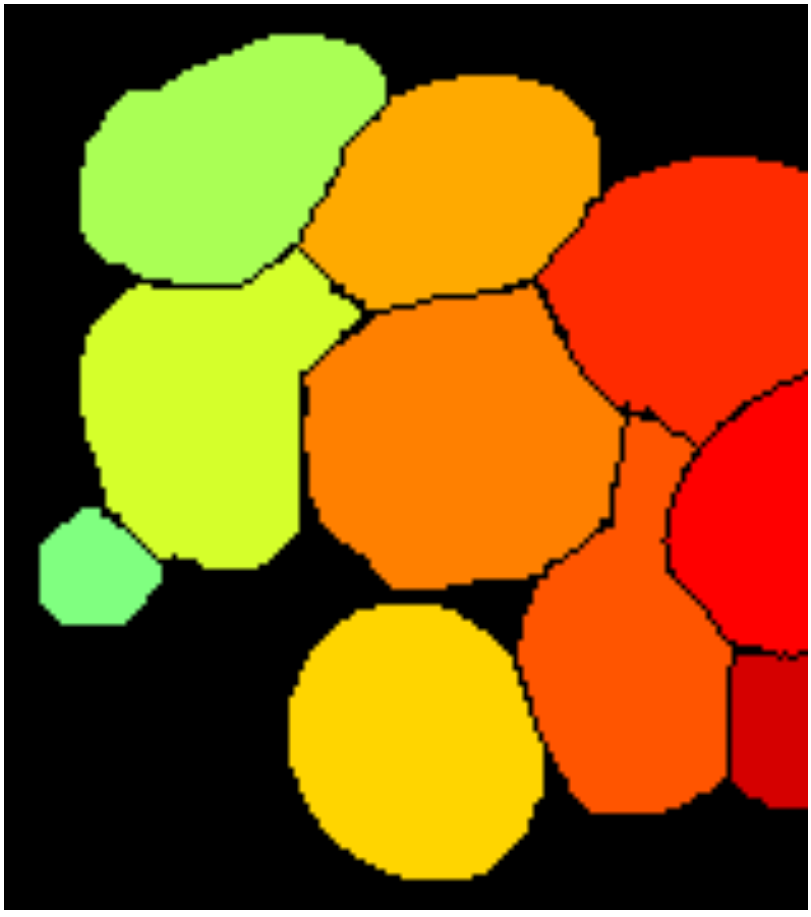

Figure 5a

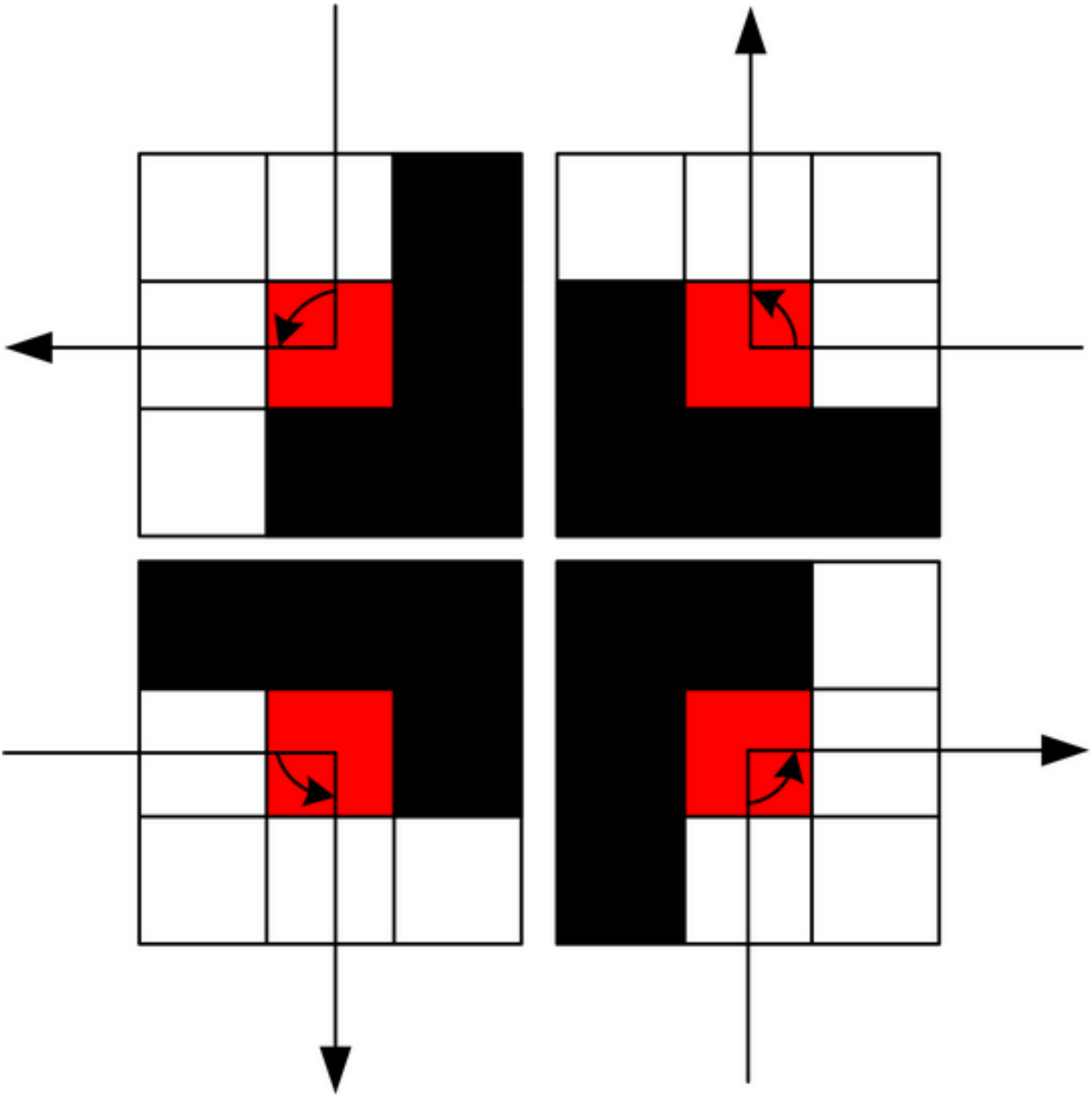

Figure 5b

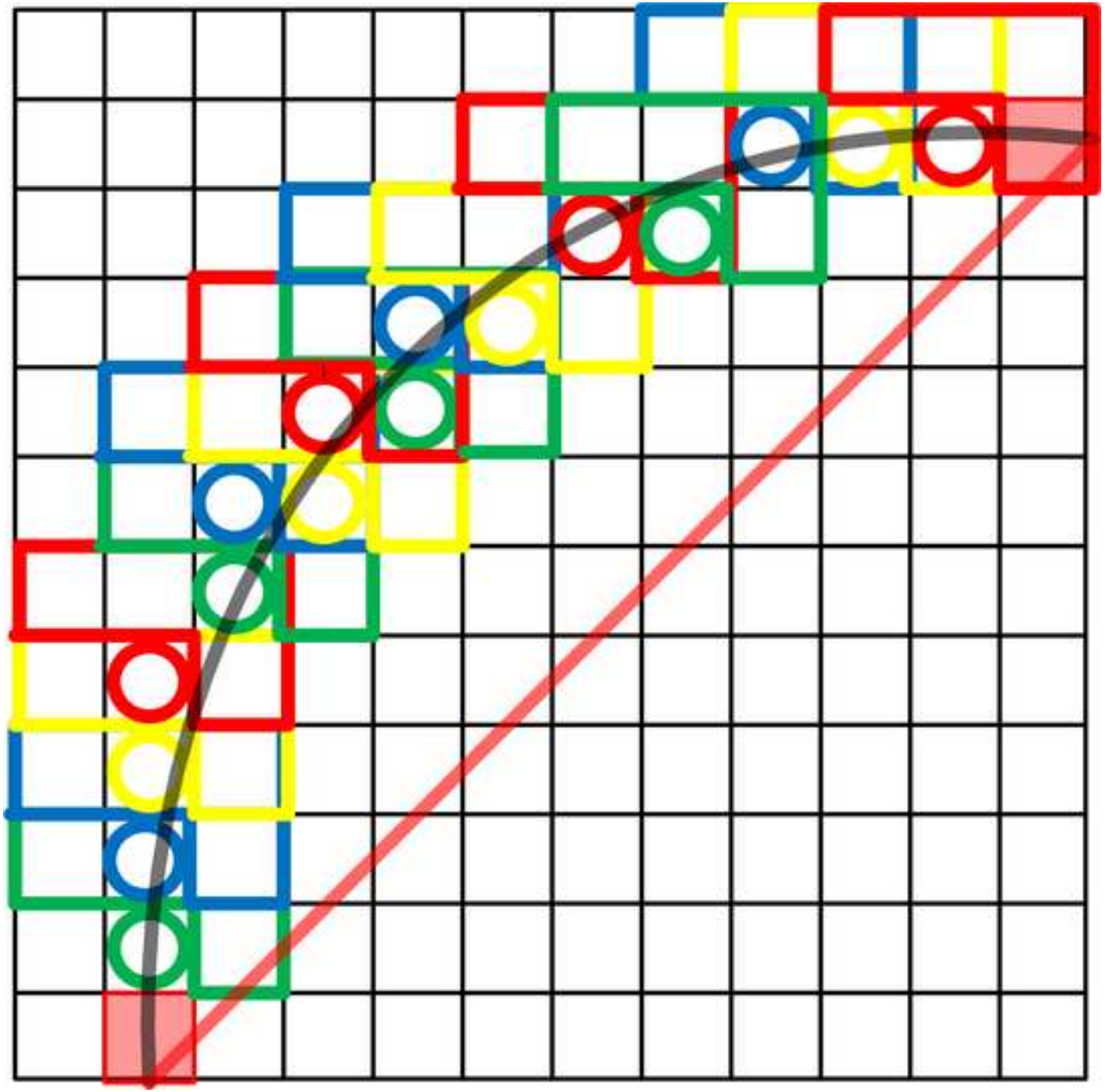

Figure 6a

Figure 6b

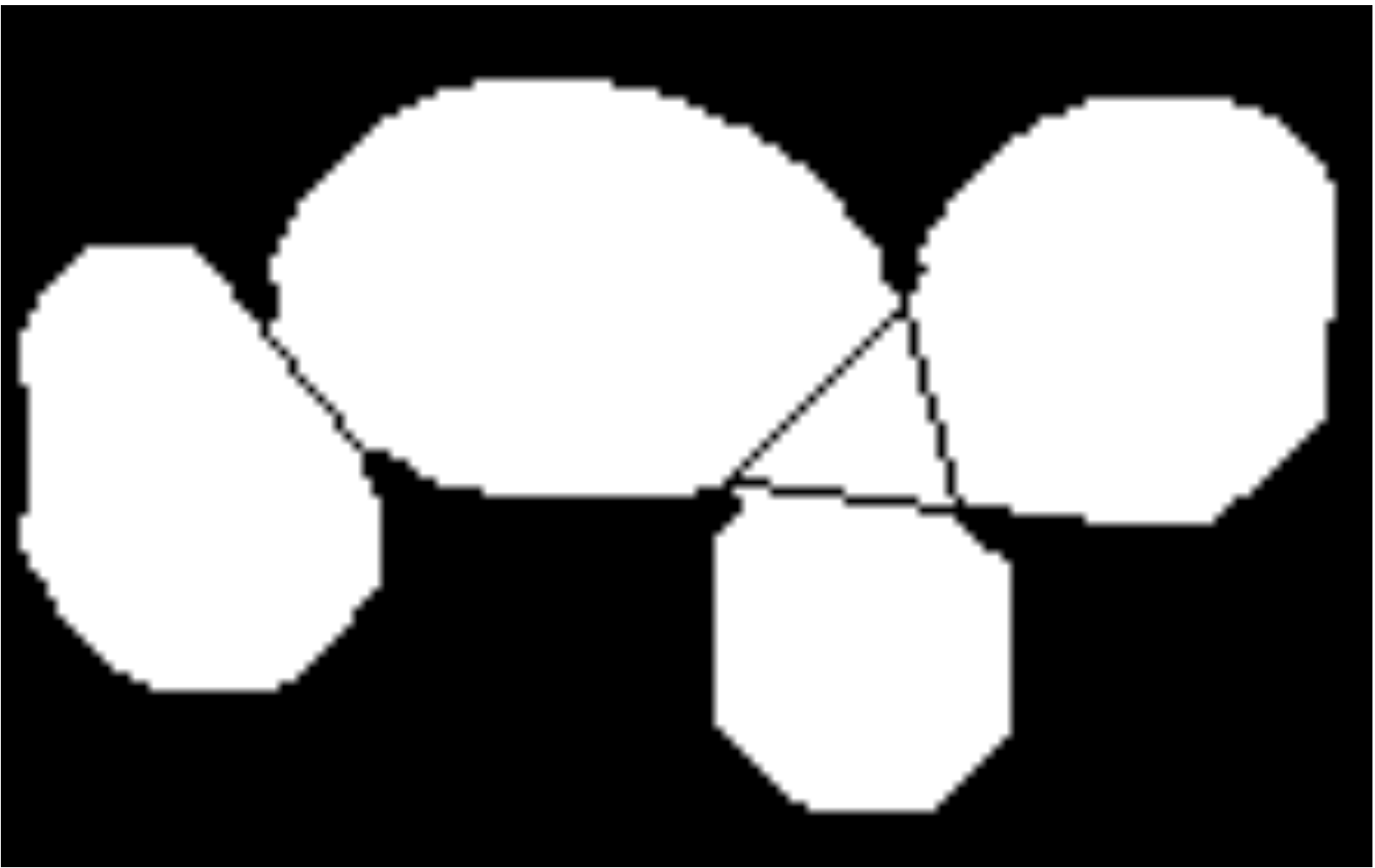

Figure 7a

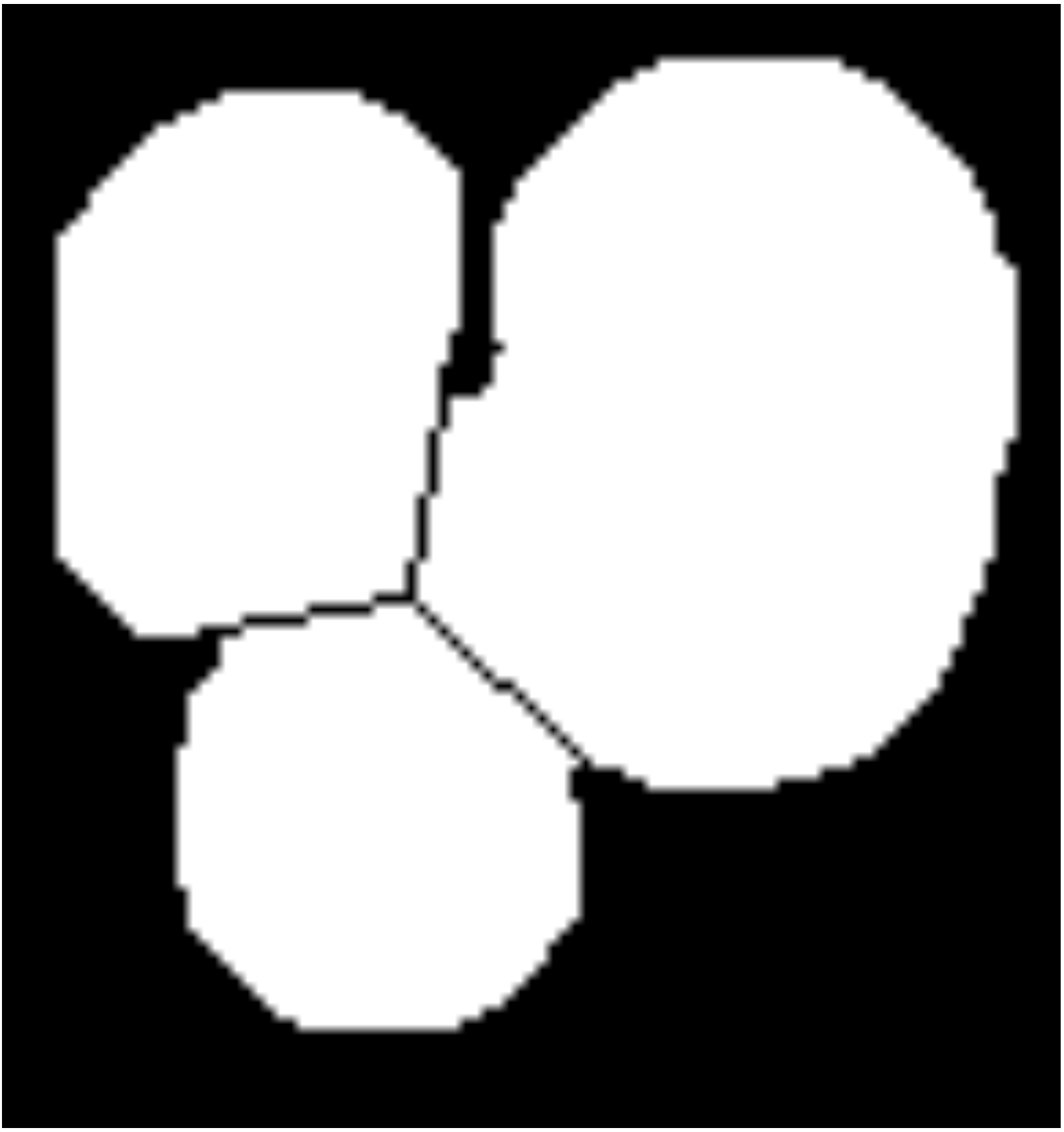

Figure 7b

Figure 8a

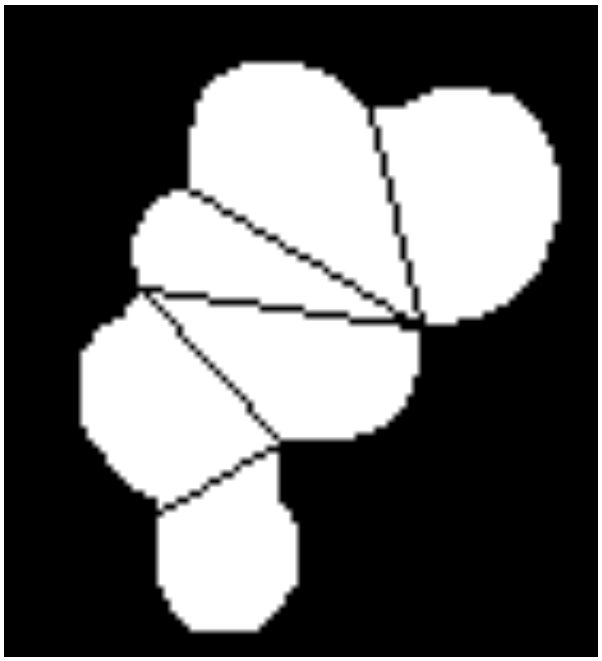

Figure 8b

Figure 9a

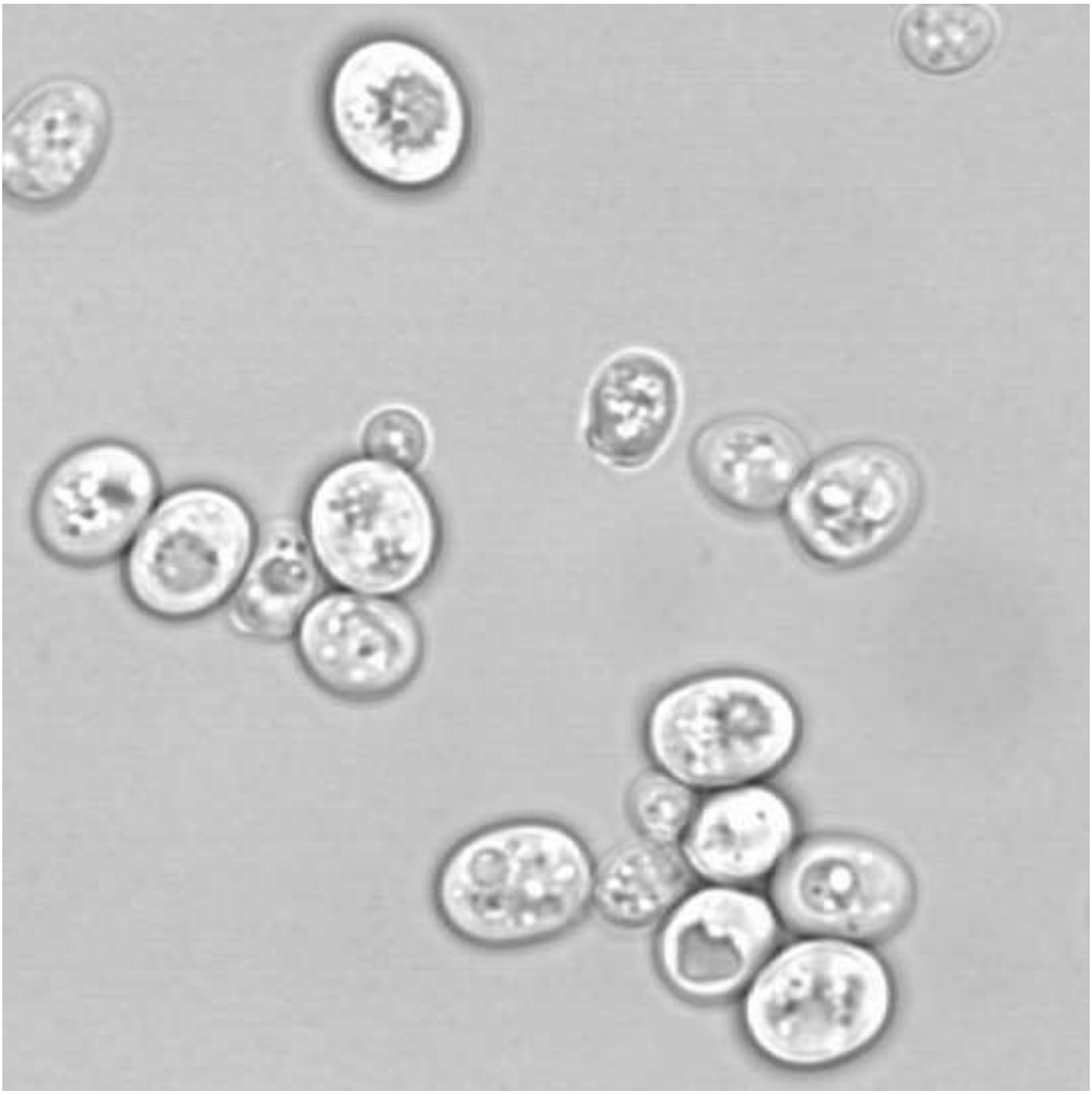

Figure 9b

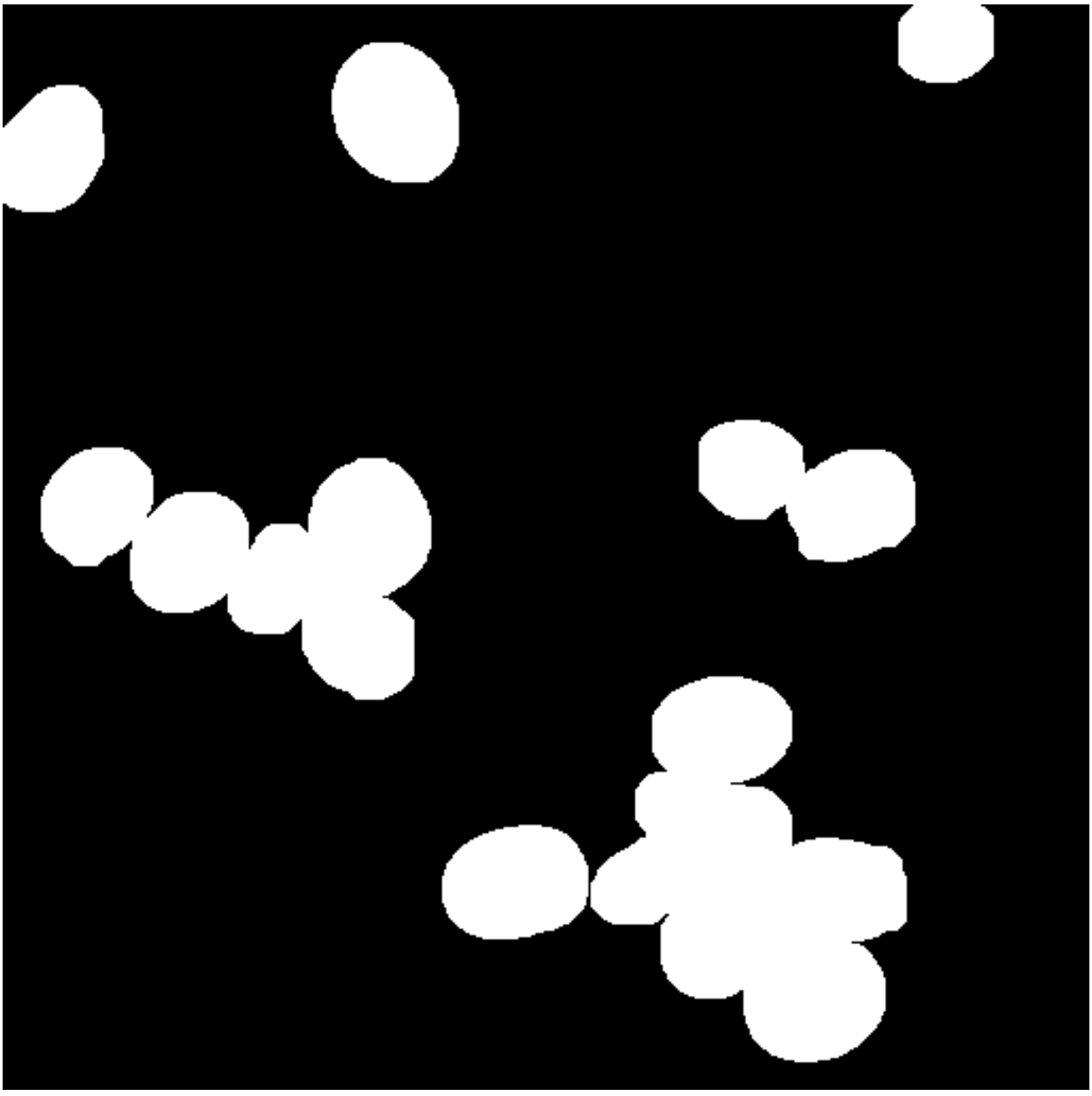

Figure 9c

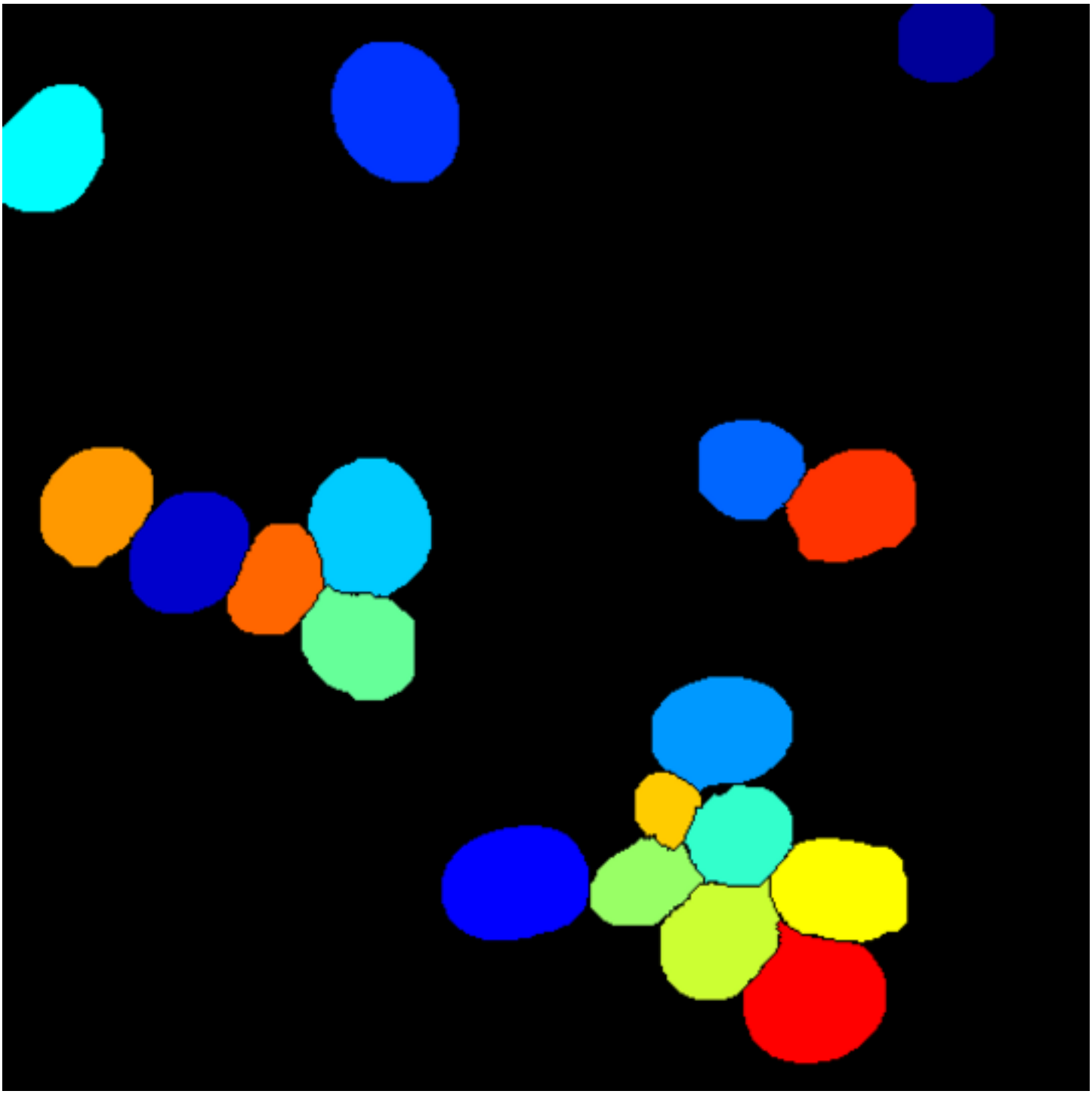

Figure 9d

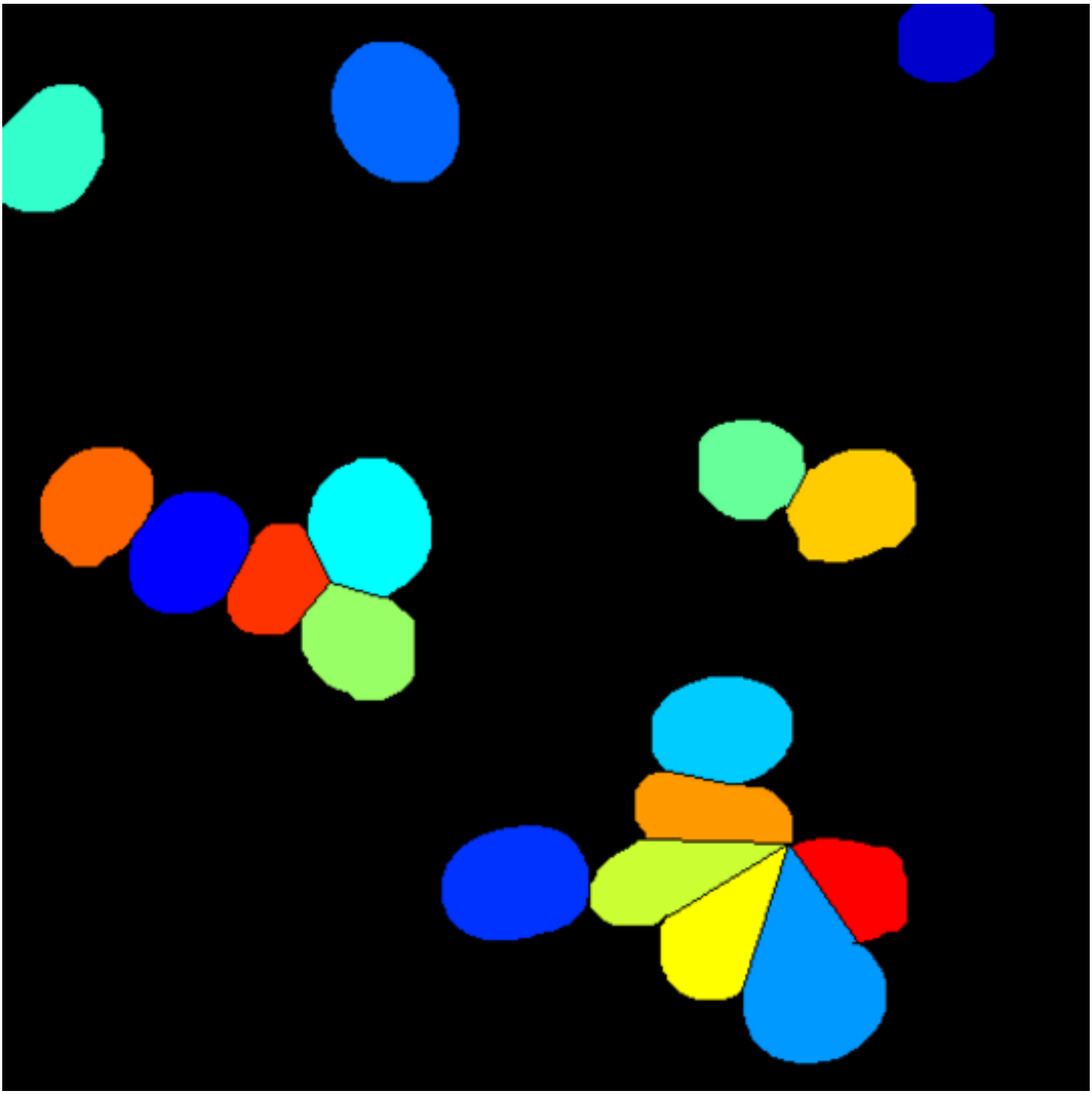

Figure 10a

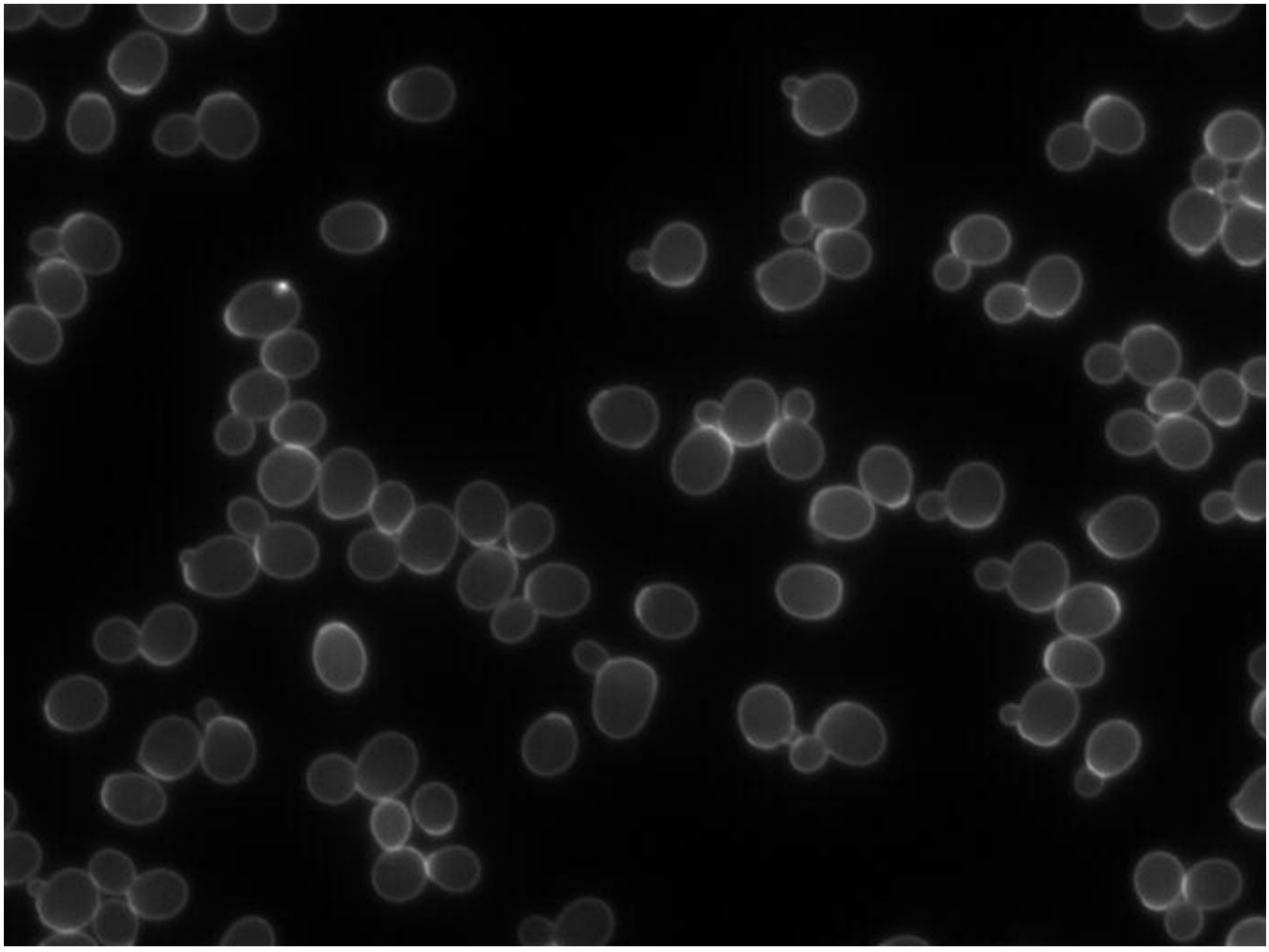

Figure 10b

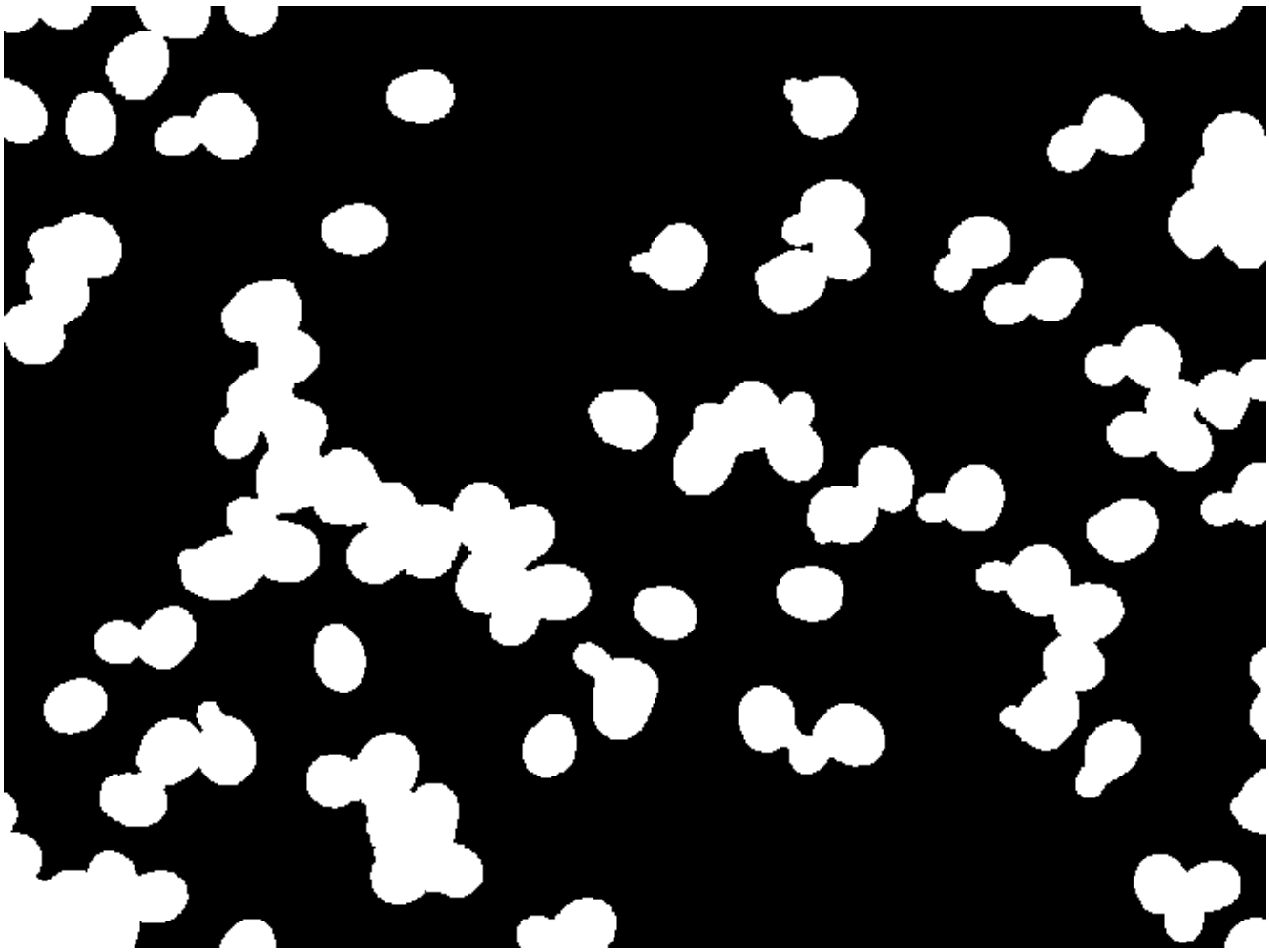

Figure 10c

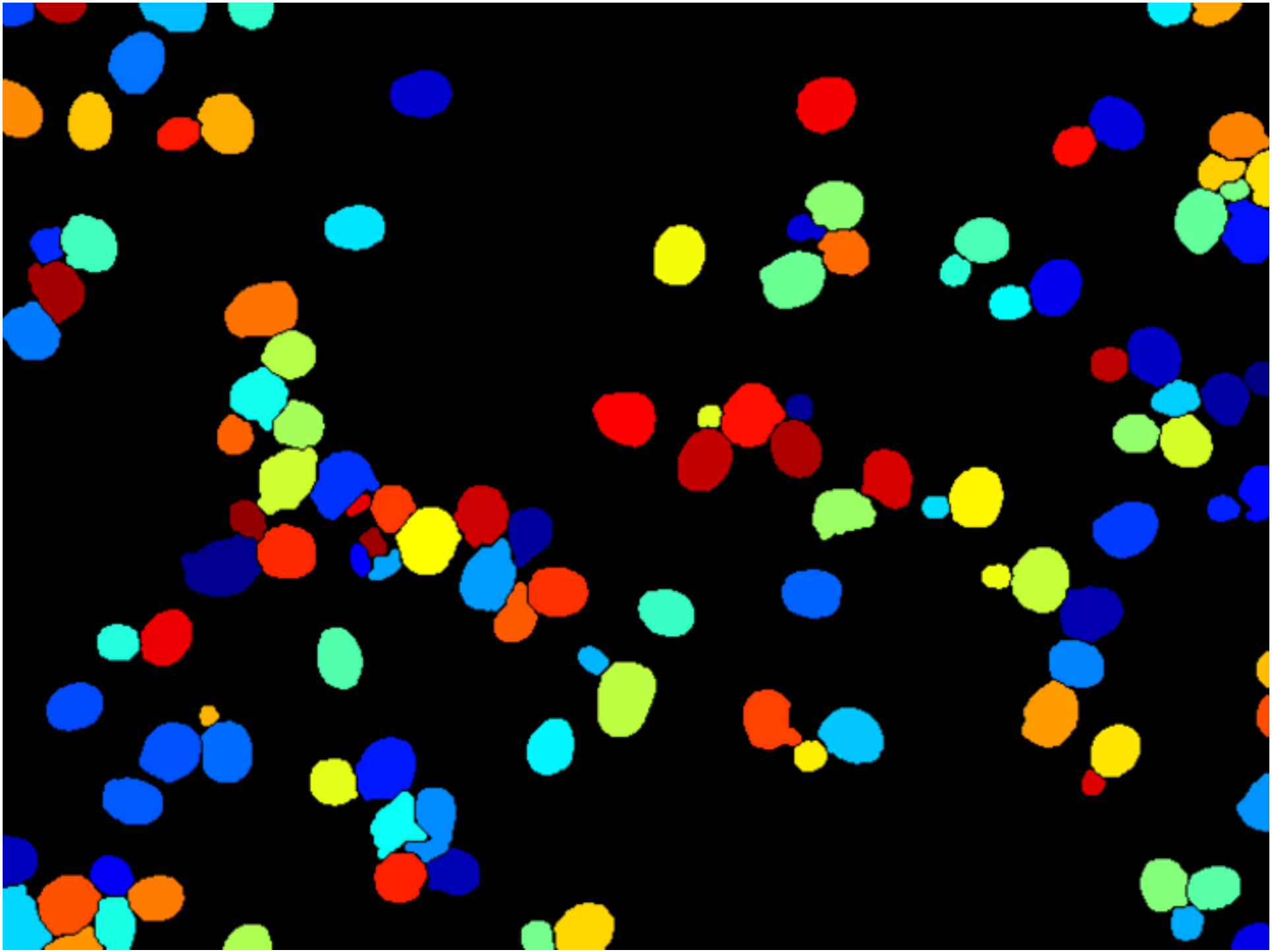

Figure 10d

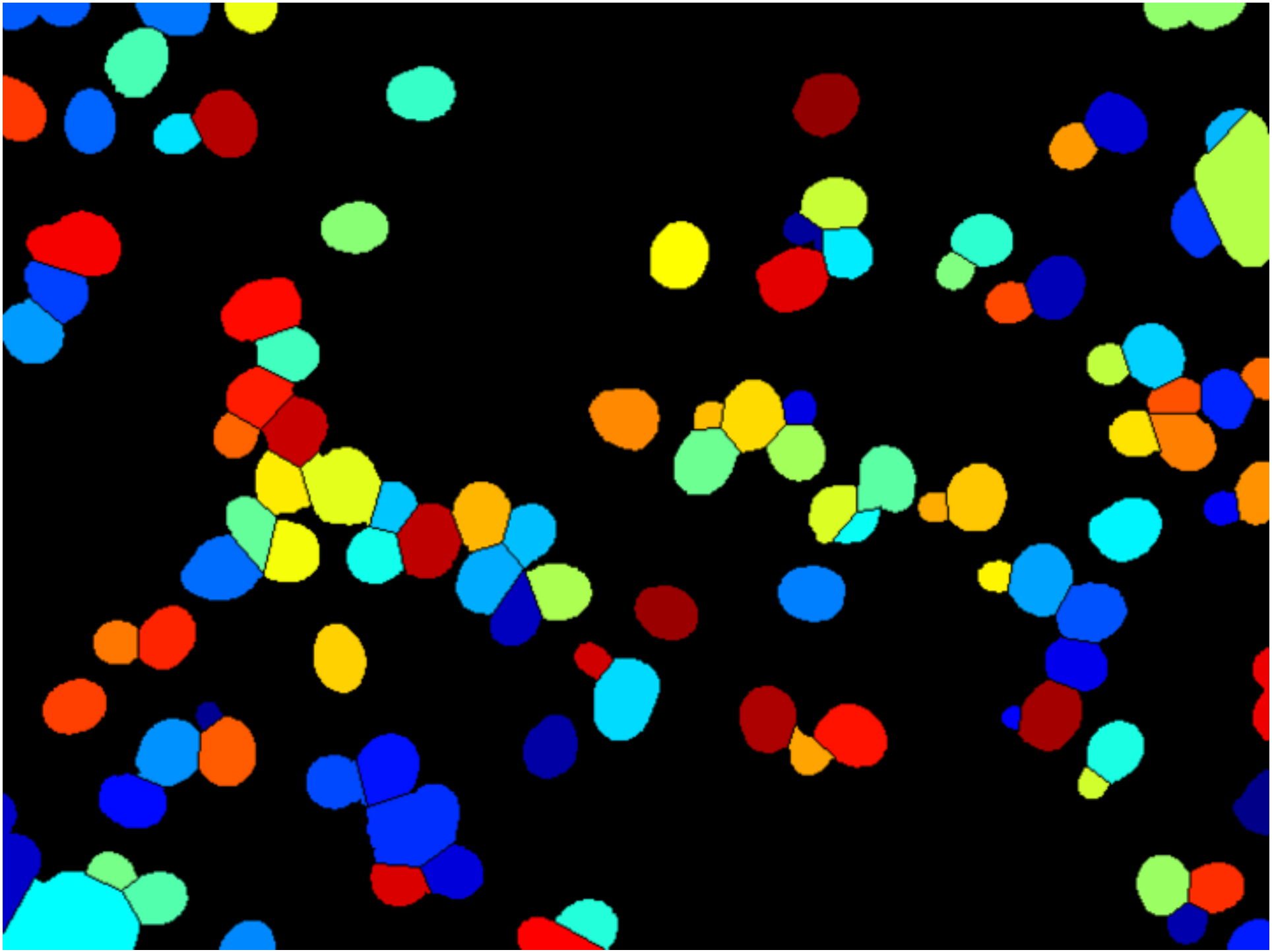

Figure 11a

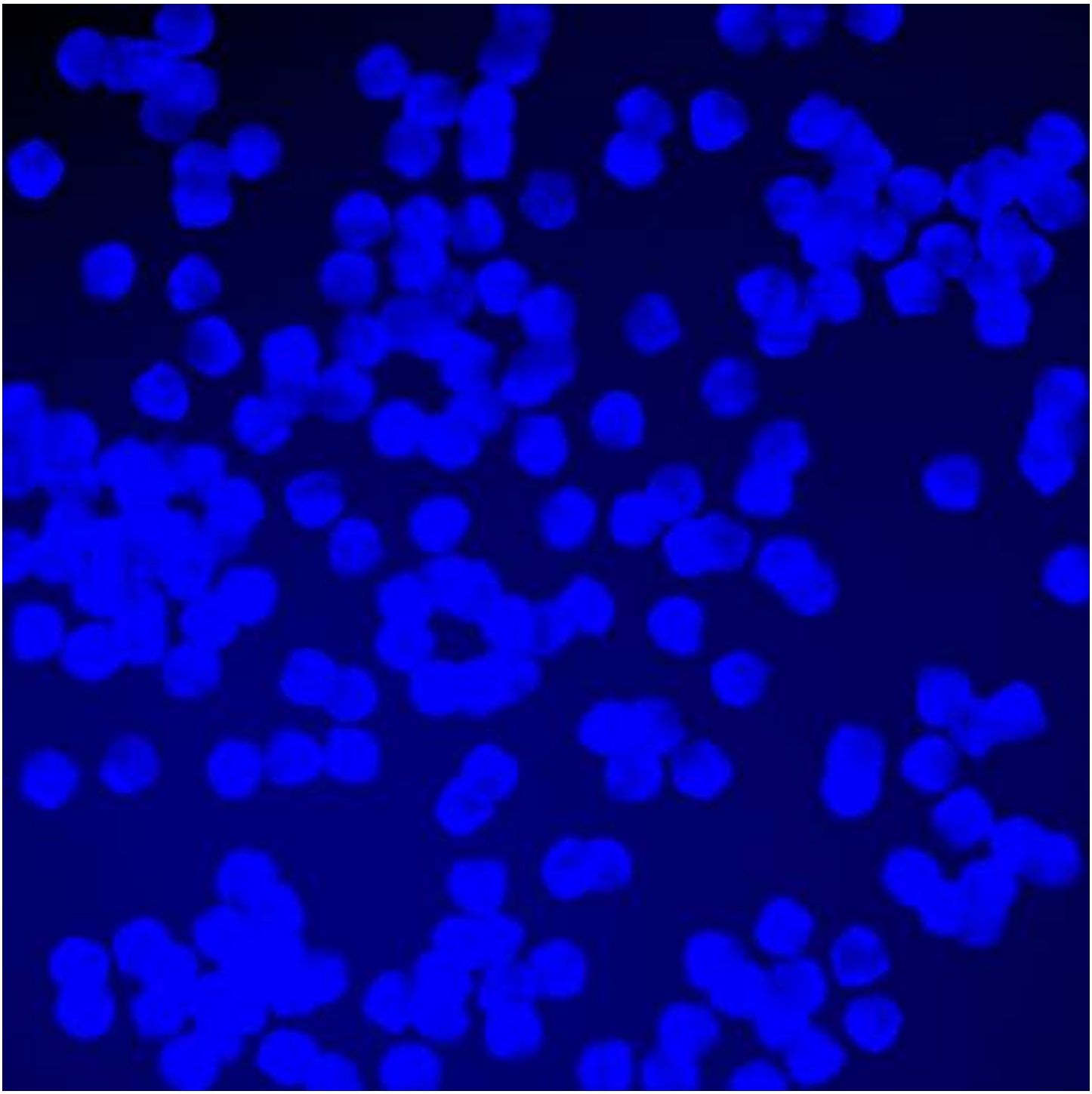

Figure 11b

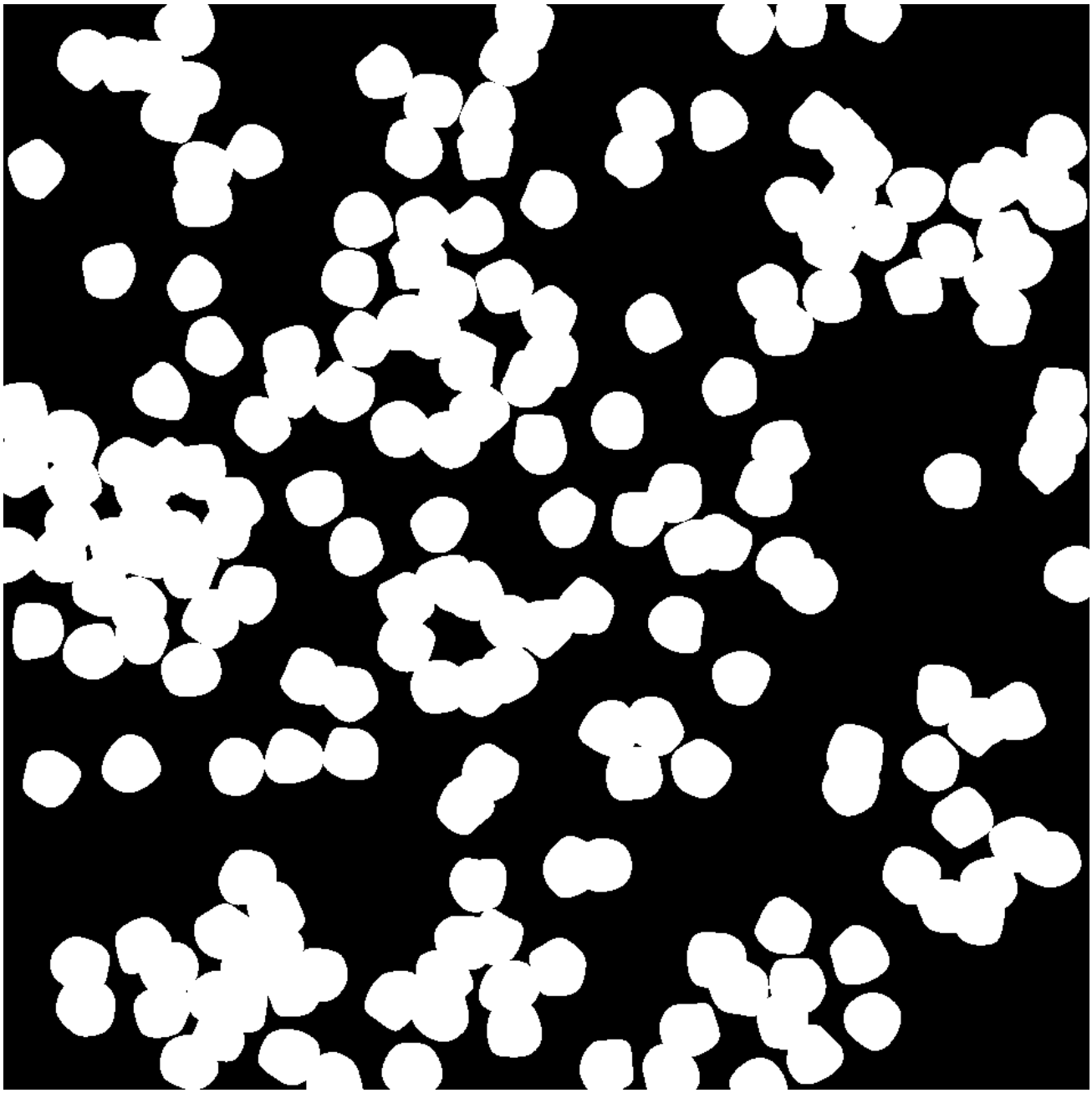

Figure 11c

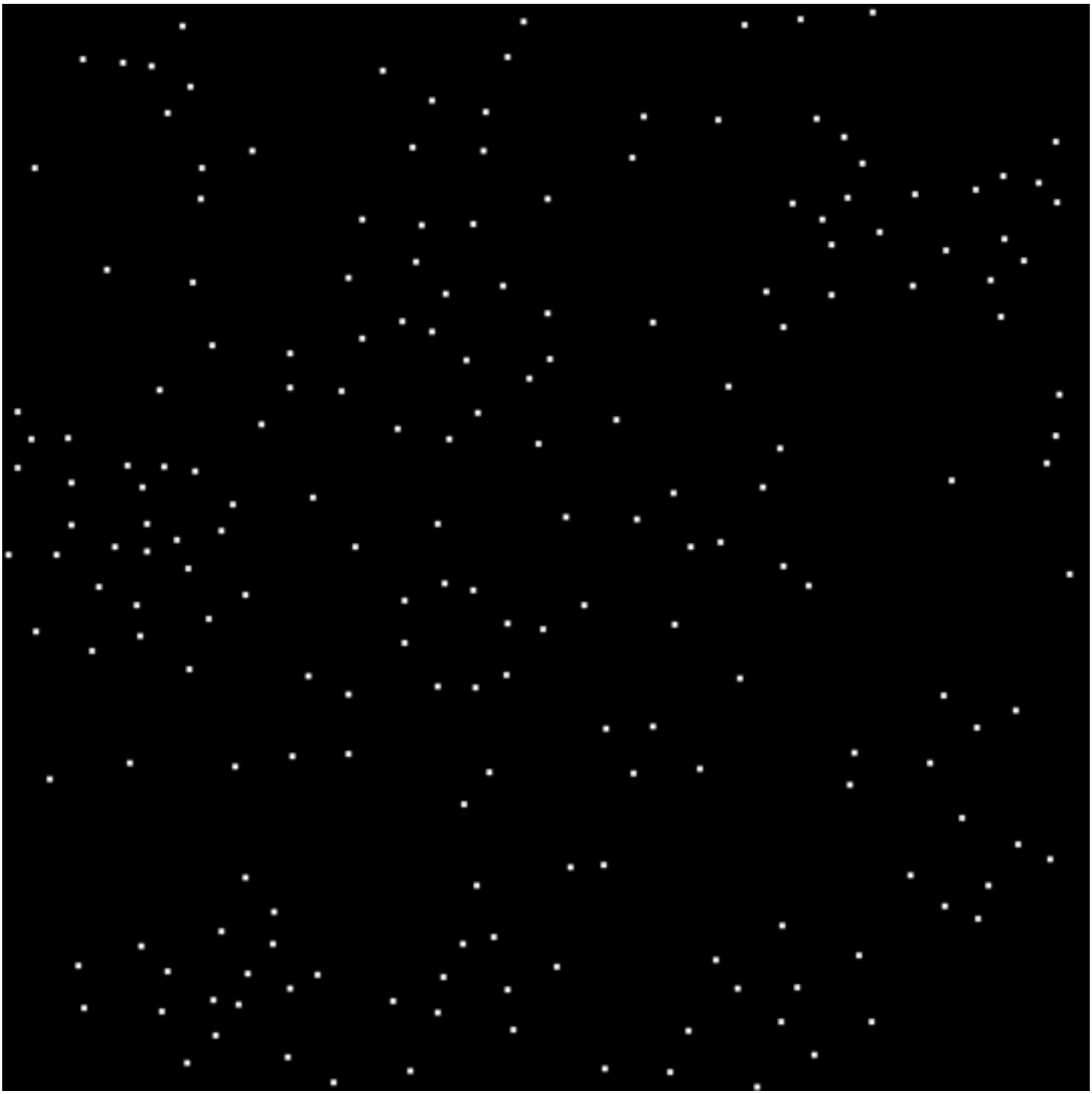

Figure 11d