

Towards a Realistic Distribution of Cells in Synthetically Generated 3D Cell Populations

David Svoboda and Vladimír Ulman

Centre for Biomedical Image Analysis
Masaryk University, Brno, Czech Republic
{svoboda,ulman}@fi.muni.cz

Abstract. In fluorescence microscopy, the proper evaluation of image segmentation algorithms is still an open problem. In the field of cell segmentation, such evaluation can be seen as a study of the given algorithm how well it can discover individual cells as a function of the number of them in an image (size of cell population), their mutual positions (density of cell clusters), and the level of noise. Principally, there are two approaches to the evaluation. One approach requires real input images and an expert that verifies the segmentation results. This is, however, expert dependent and, namely when handling 3D data, very tedious. The second approach uses synthetic images with ground truth data to which the segmentation result is compared objectively. In this paper, we propose a new method for generating synthetic 3D images showing naturally distributed cell populations attached to microscope slide. Cell count and clustering probability are user parameters of the method.

Keywords: distance map, 3D imaging, cell populations, cross-correlation, simulation.

1 Introduction

The credibility of synthetically generated datasets is based on their ability of mimicking the real data. When simulating cells, we need to deal with the following three aspects:

- cell shape
- cell internal texture
- mutual positions of individual cells

The first two aspects have already been deeply studied by many groups [4, 6–8, 13], however those who focused on proper generation of cell populations worked only with 2D image data [4, 6, 8]. The extension to 3D is not straightforward as it brings new methodological issues and as it is a problem of higher computational demands, both in time and memory. In [13] the truly 3D cell populations were generated but the distribution was not solved. The cells were uniformly spread across the whole field of view (FOV). The clusters had to be formed manually by

setting the fixed position for each cell individually. The only constrain applied on every cell was to prevent from overlaps.

Let us emphasize the main difference when generating a cell population into 2D image versus 3D image. In the first place, since a 2D image contains a packed visual information, which is naturally 3D, it is not unusual to see cells that appear overlapping in 2D images. However, cells cannot overlap in 3D images nor in reality. They can touch each other but they obviously cannot occupy the same volume physically. But in 2D images it is visually acceptable when two cells share the same space. In addition, cell position within an 3D image is controlled not only by so called clustering effect, which gathers cells into groups. Its z -position is highly influenced by gravity, i.e. the cells tend to attach to microscope slide. This is not the case of 2D images where such problem does not exist. Last but not least, 3D images are of higher capacity so it is not recommended to pointlessly repeat some operations as it worsens the efficiency of the generation process.

One of the pioneer papers focused on the problem of generation of more than a single cell in an image was published by Lockett et al. [5]. This paper was focused on the development of cell segmentation algorithm in 3D. For the purposes of determining the limits of tested algorithms the authors prepared a small set of basic geometrical objects (sphere, ellipsoids, etc.). These objects were gathered into clusters, further blurred, and finally affected by noise to imitate real microscopic data. Next year, Solorzano et. al [10] adopted this principle for testing their own thick tissue segmentation algorithm. Three years later, Grigoryan et al. [3] generated a group of uniformly distributed spheres of various diameters. Some level of overlap was allowed but too overlapped spheres were merged. Dufour et al. [2] can be considered to belong among the first researchers that were creating synthetic time-lapse sequences of moving cells. In their work, only two spheres were positioned randomly under constrain that they can't overlap and that they touch at least once during the sequence.

The researchers mentioned so far generated fully 3D synthetic image data but they employed only basic geometrical shapes and their number was limited to units. In the following, the individual research groups started to generate cells with sophisticated shapes and internal structures, and eventually the whole populations of such cells. The fundamental ideas how to prepare a computer generated large cell population were presented by Lehmassola et al. [4]. In their paper, the authors focused on all the aspects that can influence the quality of generated data, however in 2D only. The placement of individual cells was generated repeatedly until the suitable position and allowed overlap were established. The ideas defined in their paper influenced many authors. The idea of repetitive search for suitable position was also adopted in [6,8]. In [15] the excessive overlap of two cells was solved by fixing one cell and slight movement with the other one. Svoboda et al. [13] adopted ideas presented in [4] and extended them into 3D. However, the aspect of cell populations and their proper generation was omitted in this work.

In this paper, we solve the task of generating the populations of synthetic cells, with user given strength of the clustering effect, and their credibility. We

```

1: FOV  $\leftarrow$  empty 3D image {create empty scene}
2: for counter=1:NumberOfCells do
3:   Mask  $\leftarrow$  create mask of a new cell
4:   Corr  $\leftarrow$  FOV  $\star$  Mask {perform cross-correlation}
5:   Locations  $\leftarrow$  Corr  $<$   $\delta$  {apply binary template matching}
6:   Locations(:, :, 1:end- $\kappa$ ) = false {keep to the microscope slide}
7:   Map  $\leftarrow$  DistanceMap(FOV) {compute distance map}
8:   Candidates  $\leftarrow$  (Map  $<$   $\mu$  ) & Locations {keep only positions in cluster}
9:   Pos  $\leftarrow$  Rand(Candidates) {select one candidate}
10:  FOV  $\leftarrow$  Merge(FOV, Move(Mask,Pos)) {add mask to the image}
11: end for
12: Output  $\leftarrow$  FOV

```

Algorithm 1. An iterative algorithm for cell population generation

propose an algorithm that generates the cell population that resembles the real cell populations.

2 Method

The outline of proposed algorithm is drafted in the Algorithm 1. The particular steps are described in detail in individual subsections.

In the very beginning, the FOV is defined. Here, we understand the FOV as a 3D image that corresponds to the real specimen placed between cover glass and microscope slide. Initially, it is an empty image in which all the cells will be generated. The main loop is responsible for an addition of each newly generated cell into the FOV. During one iteration only one new cell is created.

2.1 Create Mask of a New Cell

First of all, the initial rough shape (Mask) is prepared (see Fig. 1(a)) and then the internal structure is created within the given mask. In this work, we adopted the approach designed in [12]. In particular, we generated HL60 cell nucleus or single microsphere, as proposed there. In general, any cell type, i.e. shape and texture, can be used provided its shape can be uniquely described with a binary mask. In the following text, we will show that there are no constraints put on the initial cell shape. The mask even need not be compact nor convex.

2.2 Perform Cross-Correlation

As soon as the initial shape of a new cell is known, we can use the cross-correlation technique to find out which positions in the FOV are suitable for the location of this cell. Provided all the previously generated cells are stored in the FOV as binary masks (see Fig. 1(b)), the cross-correlation of the FOV and the binary mask of the new cell should give zero value at the positions where the new

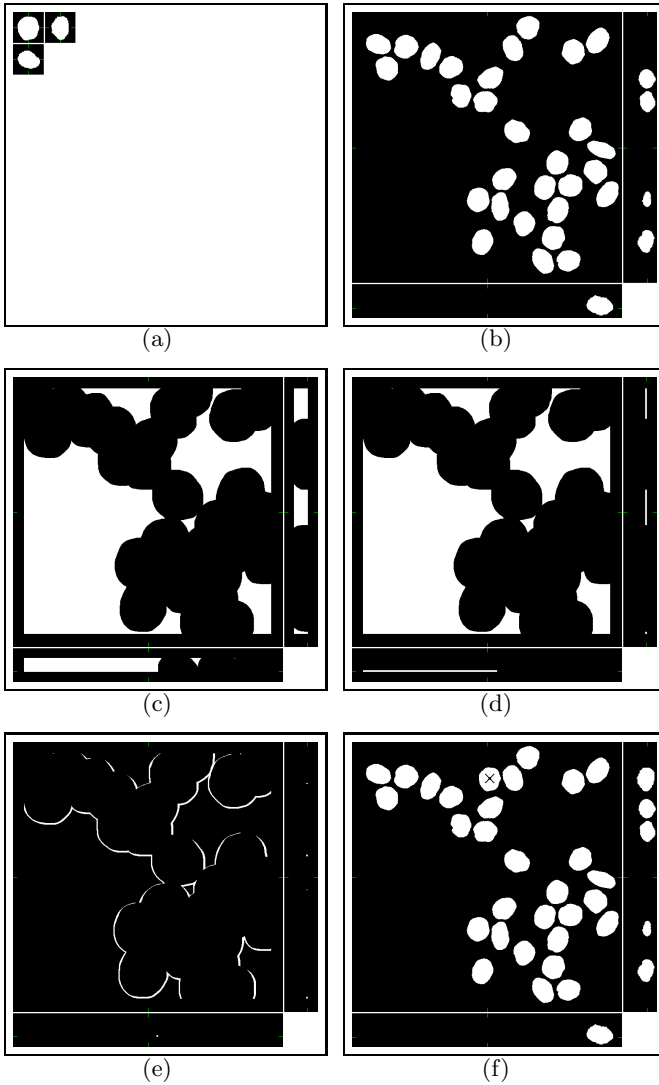


Fig. 1. An example of a placement of a new cell (a) into the population of already generated 29 cells (b) with probability of clustering 75%. Black pixels denote background. First, all the possible locations, where the new cell can be placed, are detected via cross-correlation technique and marked as true with white colour (c). Some locations, that occur far from the microscope slide, are rejected (d). Since cells are forced to form clusters, majority of locations that would result in sparse distribution of cells within the available space are also rejected (e). Only one from the remaining locations is selected and the new cell is placed there. The results is merged with all the previously generated cell masks. The 30th cell added is marked with cross (f). Each 3D figure consists of three individual images: the top-left image contains selected xy -slice, the top-right image corresponds to selected yz -slice, and the bottom one depicts selected xz -slice

cell may be placed without overlap. Let us analyze this idea from the practical point of view. In case the size of FOV is $1300 \times 1030 \times 100$ voxels¹ and the size of Mask is $100 \times 100 \times 80$ voxels we cannot afford to compute the cross-correlation with naive convolution with flipped kernel due to very high computational cost. To reduce the time complexity we suggest to compute the convolution of such large images using fast Fourier transform (FFT). The improvement is, nevertheless, at the expense of increased memory demand. The memory demand can be, however, considerably reduced if the technique designed in [11] is used.

2.3 Apply Binary Template Matching

Due to the fact that the cross-correlation was performed via FFT, we must expect some cumulative errors during the computation. Therefore, the suitable positions of the Mask in the FOV are not represented only with pixels with zero values. We must accept some tolerance δ , which we will understand as a threshold value. All the positions with value less than δ are accepted. The rest is rejected. We have found out experimentally that $\delta = 0.5$ is the most suitable value.

At this moment, all the available positions correspond to Locations (see white voxels in Fig. 1(c)) where the currently inspected cell may appear without threat of overlapping any other previously generated cell.

2.4 Keep to the Microscope Slide

When dealing with 3D images, we must also take into account the fact that all the generated cells are subjected to gravity. In this sense, the positions of cells cannot be randomly spread across the whole FOV. Namely, the z -positions of each generated cell are limited in such a way that the cell should seem to be attached to the surface of microscope slide (see Fig. 1(d)).

As the slide is not perfectly flat and some impurity may appear in the real specimens, we introduced a tolerance κ that represents these imperfections. It prevents cells from lying exactly in the same z -plane, which would seem quite unnatural.

2.5 Compute Distance Map

Many approaches [4, 6, 8] used distance map [1] of the FOV to directly form the area of possible location of newly generated cells. The values in the distance map were further modified by Gaussian function in order to give a higher probability to be selected to close positions prior to the further ones. This way, the available positions in the vicinity of already existing cells in the FOV were marked as the most suitable for the placement of new cell. Random selection of the position respecting these probabilities led naturally to populations of cell forming

¹ This is a size of standard full-frame real 3D image acquired using CCD camera Micromax 1300-YHS attached to microscope Zeiss 200M, Zeiss Plan-Apochromat 100x/1.40.

the clusters. However, this also generated many overlaps. To avoid them, the authors [4, 6, 8] repeatedly tried new positions until the lower level of overlap or even no overlap was detected. As explained in the introduction, this is not suitable (due to unrealistic overlaps of cells in 3D) nor very effective (due to the iterative position search) in the case of 3D images.

In this work, we also compute the distance map but the results are not submitted to Gaussian function to express the importance of each individual item in the `Locations`. The distance map is only used to narrow the number of available positions in the `Locations` to those that give rise to clusters.

2.6 Keep Only Positions in Cluster

The set of all available `Locations` for a new cell is restricted to those that appear near to some already generated cells in `FOV`. The `Locations`, where the distance map is lower than some user defined threshold μ , are kept while the others are rejected (see Fig. 1(e)). The value of variable μ is typically very small ($\mu \approx 0.4$ microns).

If one wants to control also the size of gaps between the individual cells in the generated clusters it is enough to submit the distance map to the morphological erosion [9] with the structuring element of appropriate size. Of course, this erosion must be performed prior to the thresholding of the distance map.

The existence of cell clusters that occur inside the cell populations is driven by the decision whether each individual newly generated cell wants to join some previously generated cluster or not. The decision is a stochastic variable with Bernoulli distribution, where success probability $p = 0.6$ means that the cell wants to join some cluster for 60%. When the cell refuses to join (with probability $1 - p$), the sections 2.5 and 2.6 are skipped.

2.7 Select One Candidate

The candidate position that will be used for a placement of newly added cell is selected from the `Locations` that passed the previous steps of the algorithm and were not rejected. The selection is performed via random process with uniform distribution.

2.8 Add Mask to the Image

Finally, the `Mask` is shifted to the position selected in the previous step and merged with the `FOV` (see Fig. 1(f)).

3 Examples of Generated Data

Using the previously described algorithm a set of populations of HL60-cell nuclei was generated. We intentionally omit inspection of the shape and internal

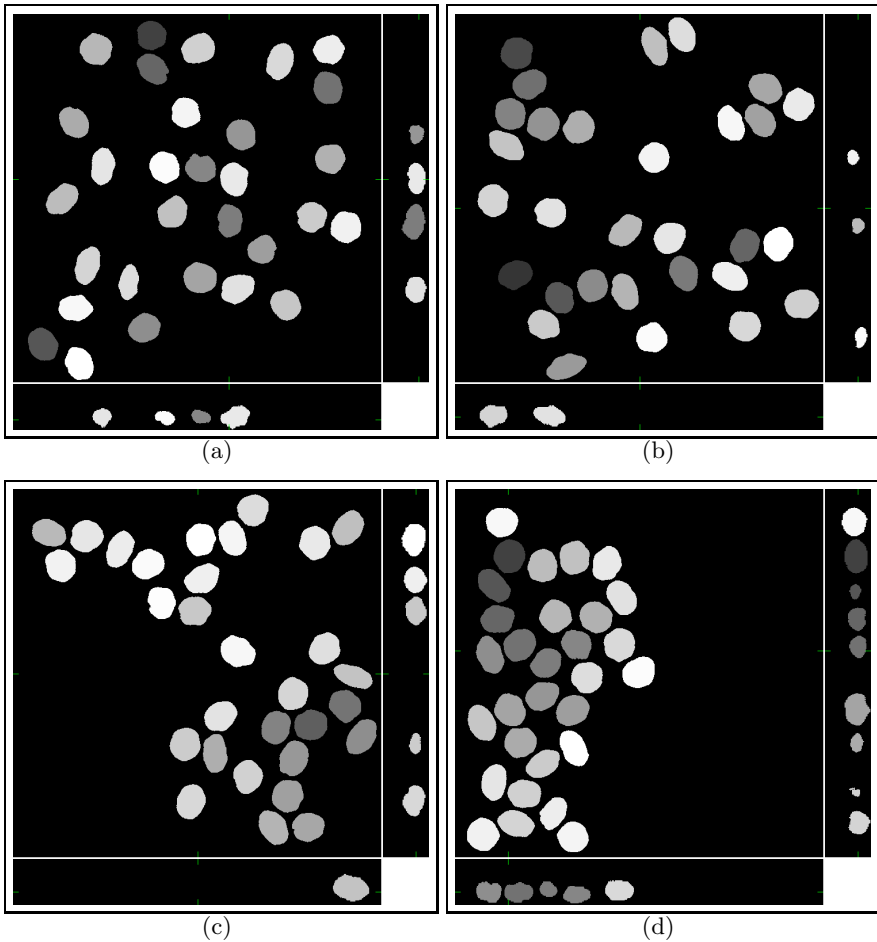


Fig. 2. Synthetic cell population with 30 cells. The masks were generated with clustering probability set to a) 0%, b) 50%, c) 75%, and d) 100%. Each cell mask is displayed with its own label so that individual cells can be clearly recognized. In this sense, the masks are depicted with different grayscale intensity levels. Each 3D figure consists of three individual images: the top-left image contains selected xy -slice, the top-right image corresponds to selected yz -slice, and the bottom one depicts selected xz -slice.

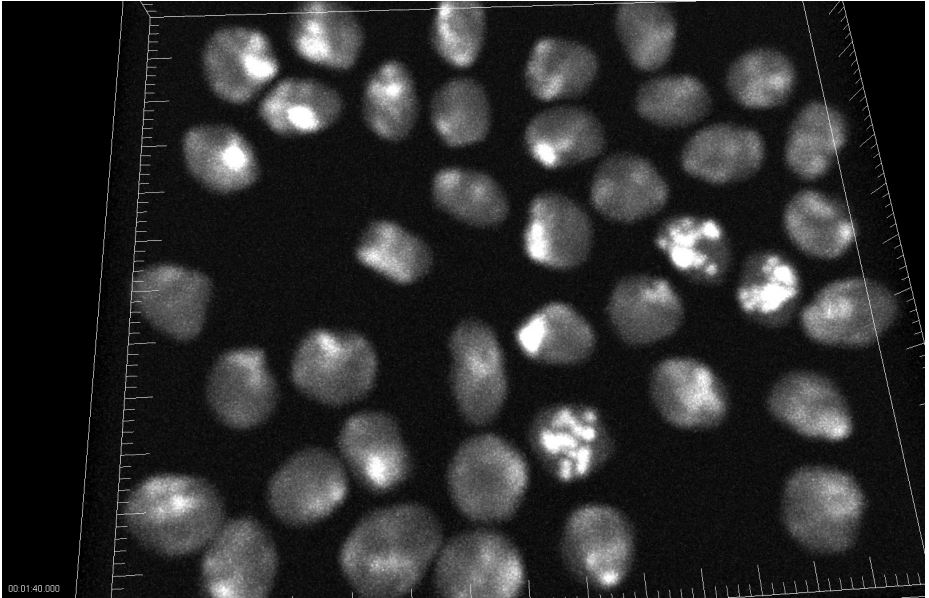


Fig. 3. An example of computer generated data. The initial distribution of cells was prepared by the newly proposed algorithm. Image data were visualized using Imaris[®] (Bitplane AG, Switzerland).

structure of individual cells. The focus of the paper is on correct and realistic distribution of cells within the populations. See Fig. 2 for examples of cell populations generated with different probabilities of clustering.

Even though this paper is not dedicated to real-time computation, we believe that the potential user of the method might be interested in time needed for generation of one simple cell population. Let us assume we have an empty image of $640 \times 640 \times 50$ voxels. In this case, the generation of 20 cells (each with diameter of 56 voxels) takes approximately 17 minutes on Intel Xeon 2.83GHz quad core, 32 GB. For clarification, what happens if we change the volume of the initial image or the amount of generated cells, we should analyze the complexity of the algorithm. It can be shown that the time of generation is linearly dependent on both the volume of image and the amount of generated cells. This fact was also verified in the parametric study we performed on the same computer. Regarding the probability of clustering, this parameter only controls whether the distance transform [1] is called. Hence, its influence on the computational time of the whole algorithm is neglectable.

The results of this new algorithm together with the principles of generating single live cell proposed in [14] have been used in *Cell Tracking Challenge* joint to the conference *ISBI 2013*. An example of one simulated cell population is depicted in Fig. 3.

4 Conclusion

In this paper, we introduced a novel algorithm that solves the task of generation of realistically looking-like cell populations in the field of fluorescence microscopy. The user can simply drive the level of clustering of each population with just a single number. Since the simulation happens inherently 3D, generated cells tend to attach to the bottom of the volume of interest due to the simulation of gravity. The proposed algorithm is not limited to any specific type of cells. For example, slides covered with microspheres, that are typically used for the measurement of real PSF, can be generated with this algorithm as well. In general, the population of any object, that is uniquely determined by a binary mask, can be generated.

As mentioned in the previous section, the algorithm has already been employed in the recent projects. For example, we remind simulated datasets, in which this algorithm has been used for creating of initial distributions of cell populations, used in the *Cell Tracking Challenge* joint to the conference *ISBI 2013*. Furthermore, the proposed algorithm has also been newly embedded into a generator of static populations of cells in fluorescence microscopy called CytoPacq. Its source codes written in C++ including the newly proposed method are freely available under GNU GPL from <http://cbia.fi.muni.cz/simulator/>.

Acknowledgement. This research was supported by the Grant Agency of the Czech Republic (Grant No. P302/12/G157).

References

1. Danielsson, P.E.: Euclidean distance mapping. *Computer Graphics and Image Processing* 14, 227–248 (1980)
2. Dufour, A., Shinin, V., Tajbakhsh, S., Guillen-Aghion, N., Olivo-Marin, J.C., Zimmer, C.: Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces. *IEEE Transactions on Image Processing* 14(9), 1396–1410 (2005)
3. Grigoryan, A.M., Hostetter, G., Kallioniemi, O., Dougherty, E.R.: Simulation toolbox for 3D-FISH spot-counting algorithms. *Real-Time Imaging* 8(3), 203–212 (2002)
4. Lehmußola, A., Ruusuvoori, P., Selinummi, J., Huttunen, H., Yli-Harja, O.: Computational framework for simulating fluorescence microscope images with cell populations. *IEEE Trans. Med. Imaging* 26(7), 1010–1016 (2007)
5. Lockett, S.J., Sudar, D., Thompson, C.T., Pinkel, D., Gray, J.W.: Efficient, interactive, and three-dimensional segmentation of cell nuclei in thick tissue sections. *Cytometry* 31, 275–286 (1998)
6. Malm, P., Brun, A., Bengtsson, E.: Papsynth: simulated bright-field images of cervical smears. In: *Proceedings of the 2010 IEEE Int. Conference on ISBI, ISBI 2010*, pp. 117–120. IEEE Press (2010)
7. Murphy, R.: CellOrganizer: Image-derived models of subcellular organization and protein distribution. *Methods Cell. Biol.* 110 (2012)

8. Rajaram, S., Pavie, B., Hac, N.E.F., Altschuler, S.J., Wu, L.F.: Simucell: a flexible framework for creating synthetic microscopy images. *Nat. Methods* 9(7), 634–635 (2012),
<http://www.biomedsearch.com/nih/SimuCell-flexible-framework-creating-synthetic/22743763.html>
9. Soille, P.: *Morphological Image Analysis*. Springer (1999)
10. Solórzano, C.O.D., Rodriguez, E.G., Jones, A., Pinkel, D., Gray, J.W., Sudar, D., Lockett, S.J.: Segmentation of confocal microscope images of cell nuclei in thick tissue sections. *Journal of Microscopy* 193, 212–226 (1999)
11. Svoboda, D.: Efficient computation of convolution of huge images. In: Maino, G., Foresti, G.L. (eds.) *ICIAP 2011, Part I. LNCS*, vol. 6978, pp. 453–462. Springer, Heidelberg (2011)
12. Svoboda, D., Kašík, M., Maška, M., Hubený, J., Stejskal, S., Zimmermann, M.: On simulating 3d fluorescent microscope images. In: Kropatsch, W.G., Kampel, M., Hanbury, A. (eds.) *CAIP 2007. LNCS*, vol. 4673, pp. 309–316. Springer, Heidelberg (2007)
13. Svoboda, D., Kozubek, M., Stejskal, S.: Generation of digital phantoms of cell nuclei and simulation of image formation in 3d image cytometry. *Cytometry Part A* 75A(6), 494–509 (2009)
14. Svoboda, D., Ulman, V.: Generation of synthetic image datasets for time-lapse fluorescence microscopy. In: Campilho, A., Kamel, M. (eds.) *ICIAR 2012, Part II. LNCS*, vol. 7325, pp. 473–482. Springer, Heidelberg (2012)
15. Xiong, W., Wang, Y., Ong, S.H., Lim, J.H., Jiang, L.: Learning cell geometry models for cell image simulation: An unbiased approach. In: *Proceedings of Int. Conference on ICIP*, pp. 1897–1900 (2010)