High-speed cell recognition algorithm for ultrafast flow cytometer imaging system

Wanyue Zhao
Chao Wang
Hongwei Chen
Minghua Chen
Sigang Yang

SPIE.

High-speed cell recognition algorithm for ultrafast flow cytometer imaging system

Wanyue Zhao,[a] Chao Wang,[b] Hongwei Chen,[a,*] Minghua Chen,[a] and Sigang Yang[a]
[a]Tsinghua University, National Laboratory for Information Science and Technology, Department of Electronic Engineering, Beijing, China
[b]University of Kent, School of Engineering and Digital Arts, Canterbury, United Kingdom

Abstract. An optical time-stretch flow imaging system enables high-throughput examination of cells/particles with unprecedented high speed and resolution. A significant amount of raw image data is produced. A high-speed cell recognition algorithm is, therefore, highly demanded to analyze large amounts of data efficiently. A high-speed cell recognition algorithm consisting of two-stage cascaded detection and Gaussian mixture model (GMM) classification is proposed. The first stage of detection extracts cell regions. The second stage integrates distance transform and the watershed algorithm to separate clustered cells. Finally, the cells detected are classified by GMM. We compared the performance of our algorithm with support vector machine. Results show that our algorithm increases the running speed by over 150% without sacrificing the recognition accuracy. This algorithm provides a promising solution for high-throughput and automated cell imaging and classification in the ultrafast flow cytometer imaging platform. © *2018 Society of Photo-Optical Instrumentation Engineers (SPIE)* [DOI: 10.1117/1.JBO.23.4.046001]

1 Introduction

Imaging flow cytometry, which images large-scale cells, can reveal important cell parameters, such as amount, shape, and size, by directly capturing images of individual flowing cells in real time. Measuring the composition of blood is a vital step in studying and diagnosing blood diseases or hematological cancer. Apart from the detection accuracy, the detection speed is a critical specification of flow cytometer imaging systems as well. A high-speed flow cytometer imaging system, typically running at about 100,000 cells/s, needs to screen a large enough cell population to find rare abnormal cells that are indicative of early stage diseases. However, the pursuit of high detection speed without sacrificing temporal resolution is considerably restricted by the charge-coupled device or complementary metal–oxide–semiconductor imaging techniques. Many research groups[1–3] have developed ultrafast cell imaging systems based on the time stretch concept.[4] This concept overcomes the trade-off between imaging resolution and detection speed by transforming the spatial information into temporal information. But it also brings the problem about how to deal with large amount of cell images rapidly and accurately as imaging speed raised significantly. Chen et al.[5] proposed a method of deep learning, which classifies cells accurately accompanied by prior training but suffers from high time cost. Guo et al.[6] focused on improving the precision of classification by combining label-free images with fluorescence-assisted results. However, improving cell recognition speed has been largely overlooked in previous work. In this paper, we present a detection and classification algorithm without prior training for ultrafast flow cytometer imaging with improved speed while maintaining good recognition accuracy.

2 Construction of Ultrafast Flow Cytometer Imaging System

In our flow cytometer imaging system (Fig. 1), a broadband-pulsed laser is employed as the light source with a pulse repetition rate of 50 MHz. After propagating through a section of dispersion compensating fiber, the optical pulses are dispersed in the time domain, which leads to the mapping between time domain and wavelength domain. Then, the optical pulse is cast into free space from a collimator. A combination of 1/2 and 1/4 wave plates is used to adjust the polarization state. The laser beam is then spatially dispersed by a diffraction grating. When the dispersed laser beam is focused onto a microfluidic channel with a depth of 100 μm and a width of 100 μm on a polydimethyl-siloxane microfluidic chip, a laterally distributed focused line spot is formed on the focal plane. Along this line spot, different wavelengths are located at different positions, indicating that the wavelength-to-space mapping is established. The cells to be examined flow through the channel and are illuminated by this one-dimensional scanning beam so that the intensity information of pulses forms a two-dimensional image. Finally, the light travels back through the diffraction grating to a photodiode so that spatial information is encoded onto time-domain waveform and received in sequence.

3 Cell Recognition Algorithm

According to Abbe diffraction limit, the resolution limit of our flow cytometer imaging system is $r = 0.61\lambda/\mathrm{NA} \approx 1$ μm, where λ is the source wavelength and NA is the numerical

*Address all correspondence to: Hongwei Chen, E-mail: chenhw@tsinghua.edu.cn
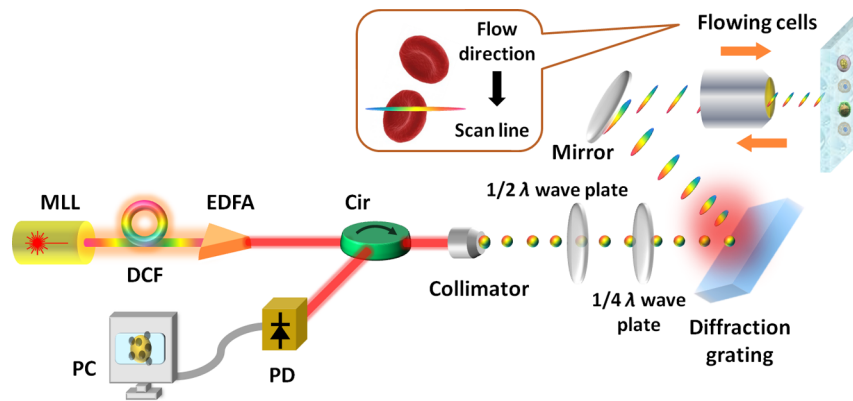
Fig. 1 Ultrafast flow cytometer imaging and analysis system setup. MLL, mode-locked laser; DCF, dispersion compensating fiber; EDFA, erbium-doped fiber amplifier; Cir, circulator; PD, photo-detector; and PC, personal computer.

aperture of the system. As the imaging frame rate reaches laser repetition rate, which is 50 MHz, 5G resolvable points are generated on 100-μm scan line/s. To match the scan rate, we set the oscilloscope sampling rate to 10 GHz. Therefore, 8 GB of data would be produced per second under 8-bit quantization. With such huge amounts of data continuously produced by the imaging system, the speed of cell image recognition is the key to enable continuous operation of the system. An efficient and fast cell detection and classification algorithm is, therefore, highly demanded. Since a large amount of pretraining is needed for typical pattern recognition algorithms, we hope to improve the operation speed and convenience using a method without training set. As shown in Fig. 2, the image background of our system is quite plain with little noise, which makes the application of recognition algorithm without training set possible. So, we proposed a recognition method consisting of a two-stage cascaded cell detection algorithm (Fig. 3) and Gaussian mixture model (GMM) classification algorithm.[7] This method avoids typical training and testing of pattern recognition and analyzes the cells in our images fast and accurately.

3.1 First Stage of Detection

In the first stage of cell detection algorithm, we roughly scan test images such as Fig. 2 obtained by the cell imaging system stated above to find out cell areas standing out from background and isolated from others. To show a significant difference in morphology among different types of cells in our image, we mixed smaller, human red blood cells with larger cultured HeLa cells. The method of obtaining these two kinds of cell suspension is described in detail in Sec. 4. The grayscale test image is first binarized as shown in Fig. 2, because the test image has a uniform gray value within background area and a large contrast between background and cells. Therefore, a simple threshold segmentation method will suffice to binarize the image to separate cells from the background initially.

We then use closing operation on the binary image obtained. Closing operation that consists of an erosion operation and a dilation operation may help smooth edges, reduce noise, and eliminate the isolated background areas surrounded by cell areas. Next, we mark all the connected cell regions individually using 8-connected boundary tracking algorithm.[8] Among all the connected cell domains, there are some small domains of inclusion and large domains of multiple cells besides single-cell domains. For tiny inclusion removal, we set a threshold of the number of pixels under each area in the image according to the approximate size of cells flowing in our system. If the cell domain has less pixels than the threshold value, this domain is considered as an inclusion, which shall not be counted.

3.2 Second Stage of Detection

From the first stage, we have got all regions containing single and multiple cells. In the second stage, our goal is to separate multiple clustered cells. To find out large regions of multiple cells, we set thresholds for the length, width, and aspect ratio of the regions according to prior cell knowledge. If any one of the three values of the cell regions exceeds their threshold value, this domain is considered as a multiple-cell domain for further segmentation. All the connected cell domains left are considered to be the image of a single cell. Then, the length, width, and average gray value of them are recorded for later counting and classification. As for the multiple-cell domains, we use an image segmentation method based on distance transform[9] and watershed algorithm[10] to set the clustered cells in multiple-cell domains apart.

Distance transform is an operation for binary images. It turns a binary image into a gray value distance image where gray value of each pixel means the distance between the pixel and its nearest background pixel. Theoretically, we need to scan all the background pixels to find out the minimum distance between the pixel and background. The computational cost of
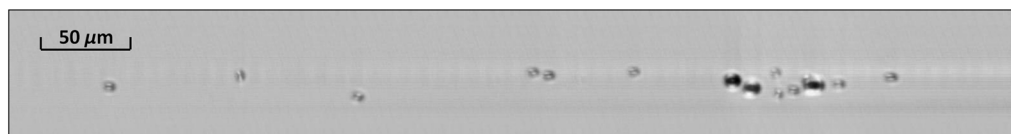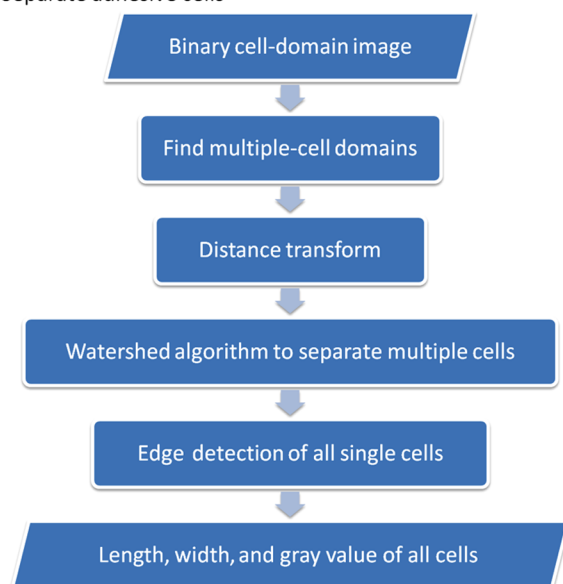

Fig. 2 Image obtained by the ultrafast flow cytometer imaging system showing cell image of red blood cells and HeLa cells.

Stage 1: Find cell domains
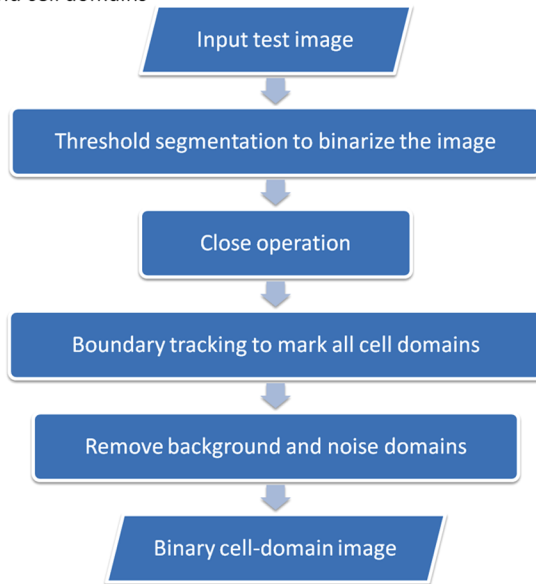
Stage 2: Separate adhesive cells

Fig. 3 Algorithm flow chart of two-stage cascaded detection method.

this global action will be very large unless the size of this test image size is very small. To reduce the computational complexity of distance transform algorithm and improve calculation speed, distance calculation starts from adjacent pixels to find out a local minimum distance in the adjacent area of the pixel. Calculation of local distances is implemented for all the pixels in the domain forward once and then backward once. The approximated global distance is a superposition of the local distances multiplied by corresponding global coordinate. Based on the above-mentioned principles, we select chamfer distance transform algorithm,[11] which operates quickly and simply. This algorithm also obtains calculated distance close enough to the real Euclidean distance. As shown in Fig. 4, distance image transformed from binary image marks pixels of different locations with different intensity. Pixels on the edge of adjacent cells are marked with smaller gray values, which make it easier to divide those adhered cells. In the next step, we employ watershed algorithm to tag the edges among adjacent cells.
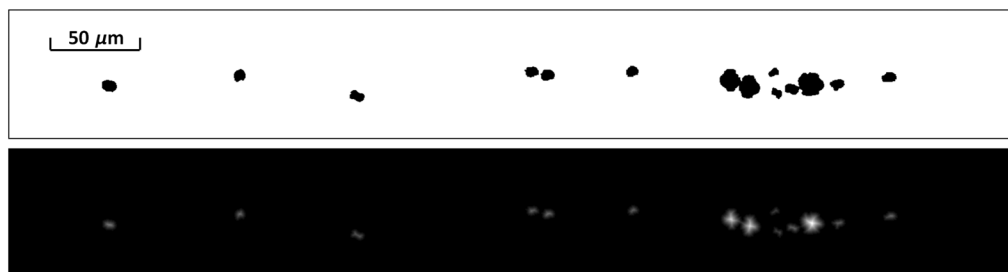
Figure 5 shows the principle of watershed algorithm. The peaks in the figure represent the centers of adjacent cells. When we set the threshold line on a high gray level, the threshold line divides the cell distance image into correct number of cells. As the threshold line continues to go down, the boundary of cells extends with the decreasing threshold line. When their boundaries meet, these cells are not fully merged and begin to adhere slowly. These pixels, where the two cells overlap initially, are the boundary points that need to be marked to separate the cells. This whole process terminates before the threshold line reaches the background domain. Based on the determined boundary points, we are able to separate the multiple-cell domain into several single-cell domains.

Figure 6 shows the single cells we got from Fig. 2 by the above-mentioned detection method. We can see that adhered cells are successfully separated and all single cells are solely marked. After that, an edge image of all single-cell domains is extracted using edge-detection approach based on Sobel operator. Then, we measure and record the length, width, and average gray value of each single-cell domain according to the edge image.

3.3 Classification Algorithm

As the length, width, and average gray value of all single cells are recorded, the cells are classified sequentially by GMM classification. When the distribution of sample points is approximately ellipsoid, GMM algorithm uses GMM distribution to simulate the probability density function generating the sample points and then clusters them by calculating the parameters of GMM distribution. GMM distribution is defined as

Fig. 4 Binarized cell image from Fig. 2 is shown in the first figure and processed into distance image by chamfer distance transformation in the figure below.
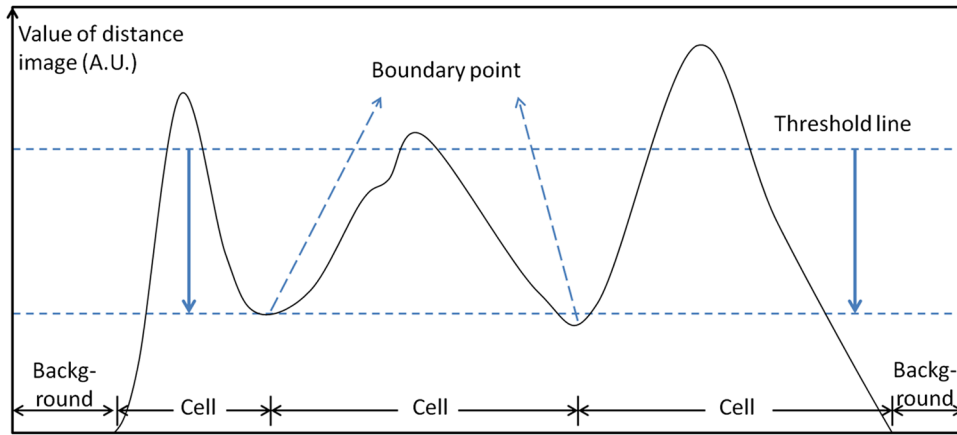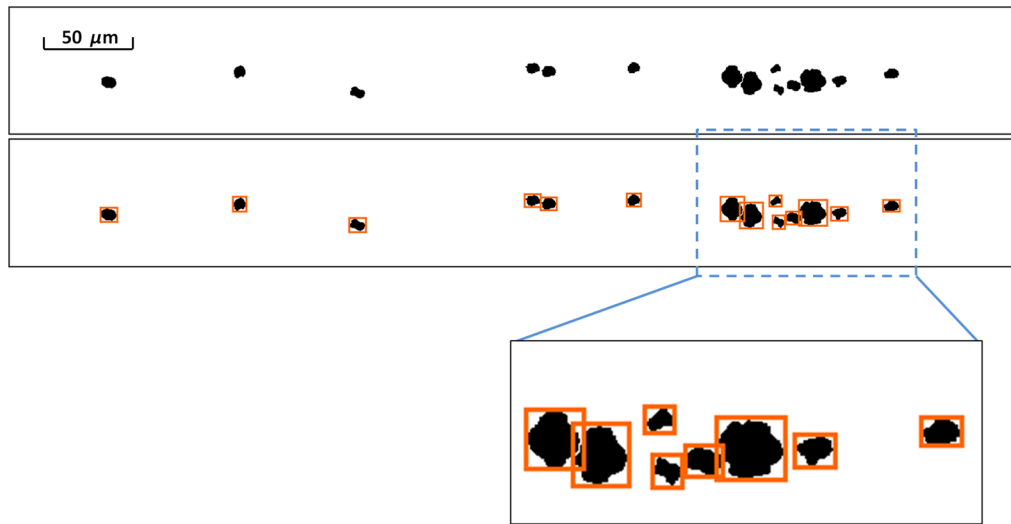
Fig. 5 Principle of watershed algorithm.

Fig. 6 Single cells we obtained from Fig. 2 by the two-stage cascaded detection method are shown in the first figure and then marked by boxes in the figure below.

$$p_M(x) = \sum_{i=1}^{k} \alpha_i \cdot p(x|\mathbf{\mu}_i, \Sigma_i), \qquad (1)$$

where $p(x|\mathbf{\mu}, \Sigma)$ is the probability density function of Gaussian distribution, $\mathbf{\mu}$ is the mean vector, Σ is the covariance matrix, k is the number of Gaussian distributions, and α_i is the mixture coefficient, $\sum_{i=1}^{k} \alpha_i = 1$. To find the best α_i, $\mathbf{\mu}_i$, and Σ_i of GMM distribution, we use maximum likelihood estimation method by finding the maximum of following function:

$$LL(D) = \sum_{j=1}^{m} \ln \left[\sum_{i=1}^{k} \alpha_i \cdot p(x_j|\mathbf{\mu}_i, \Sigma_i) \right], \qquad (2)$$

where D is the sample set and m is the number of samples. When $LL(D)$ reaches its maximum

$$\mathbf{\mu}_i = \frac{\sum_{j=1}^{m} \gamma_{ji} x_j}{\sum_{j=1}^{m} \gamma_{ji}} \qquad (3)$$

and

$$\Sigma_i = \frac{\sum_{j=1}^{m} \gamma_{ji} (x_j - \mathbf{\mu}_i)(x_j - \mathbf{\mu}_i)^T}{\sum_{j=1}^{m} \gamma_{ji}} \qquad (4)$$

can be derived from $\frac{\partial LL(D)}{\partial \mathbf{\mu}_i} = 0$ and $\frac{\partial LL(D)}{\partial \Sigma_i} = 0$. Here, γ_{ji} is the posterior probability of x_j, $\gamma_{ji} = p_M(i|x_j)$. Due to $\sum_{i=1}^{k} \alpha_i = 1$, it can be derived that

$$\alpha_i = \frac{1}{m} \sum_{j=1}^{m} \gamma_{ji}. \qquad (5)$$

Equations (3)–(5) can be calculated by iteration. Procedure of iteration is listed below:

1. Initialize parameter model of Gaussian mixture distribution, including mean $\mathbf{\mu}_i$, covariance matrix Σ_i, and mixed coefficient α_i.

2. The posterior probability γ_{ji} of each cell sample was calculated from the parameter model.

3. Calculate the new mean $\boldsymbol{\mu}_i$, covariance matrix Σ_i, and mixed coefficient α_i.

4. Calculate the maximum likelihood function $LL(D)$. If it stops increasing, classify all cells with GMM distribution and record the result; otherwise, repeat steps 2 to 4.

The reasons why we choose GMM to classify cells include: (a) the size and gray value of cells follow the Gaussian distribution, which makes the algorithm converges fast and classifies accurately, (b) unsupervised clustering works more efficiently than supervised ones, and (c) feature dimension can be expanded easily for other applications.

3.4 Algorithm Time Complexity Analysis

As computational speed is the major advantage of the presented algorithm, we will compare time complexity of the presented algorithm with two other typical cell recognition algorithms. In the two steps of cell detection, the computation time is mainly consumed in the 8-connected boundary tracking algorithm and distance transformation. Both of them scan all sample images once and allocate memory all at once without complex operations, such as query or convolution. Therefore, the processing time of them is linear with the scale of sample. In the worst case, up to 32 operations need to be calculated if boundary tracking and distance transform are performed for all 8 neighborhood pixels of each pixel. Due to the sparsity of the cell images, the average number of actual calculations would not exceed half of the estimated value. Time complexity of GMM classification is $O(\sum_{i=1}^{k} ni) \in O(k^2 n)$,[7] where k is the number of Gaussian models and n is the number of cell samples. So the total time complexity of our cell recognition algorithm can be expressed as $O(N + k^2 n)$, where N is the total number of pixels.

In the following experiment, we would use a typical pattern recognition algorithm called support vector machine (SVM) pattern learning algorithm[12] in control group. It is a commonly used supervised learning recognition algorithm with training group. The training time complexity of SVM is $O(m^2 t^2)$ and the test complexity is $O(m^2 N)$, where t is the number of training samples and m is the dimension of feature vector. As can be seen, our algorithm using unsupervised learning by making cell detection in advance may avoid the significant time overhead required for training of supervised learning and point-by-point window sliding of test image. Another algorithm used for flow cytometry is convolutional neural network (CNN).[5] Its time complexity of convolutional layer l for each sample is $O(M^2 K^2 C_l C_{l-1})$, where M is dimension of feature map, K is the dimension of convolution kernel, and C_l is the number of layer l outputs. So the total time complexity is $O[N(\sum_l M^2 K^2 C_l C_{l-1})]$. Clearly, CNN has a high time complexity due to repetitive convolution operation. It usually requires GPU to help shorten the operation time. In summary, the presented algorithm has lower time complexity than other common cell recognition algorithms theoretically.

4 Experimental Results

We applied the presented recognition method to process images of three kinds of cell suspension: HeLa cells, Jurkat cells, and red blood cells. HeLa cells, a cell line of cervical cancer cells, are adherent cells. We cultured them in Dulbecco's modified eagle medium (DMEM), which contains 10% fetal bovine serum and 1% penicillin and streptomycin. The medium was incubated in a 5% CO_2 incubator at 37°C. When the cells covered about 80% of the bottom of the petri dish, we washed the cells with phosphate-buffered saline and treated with trypsin for 3 min. Trypsin was then aspirated and fresh medium was added. The cells were gently pipetted from the bottom of the petri dish and whipped into single-cell suspension for use. Jurkat cells, an acute T lymphoblastic leukemia cell line, are suspension cells. We cultured them in DMEM medium as well. The medium was incubated in a 5% CO_2 incubator at 37°C. When the cells grew to a dense density, we gently pipetted the cells and whipped them into single-cell suspension for use. To obtain red blood cell suspension, we drew human whole blood and then added edetate anticoagulant and saline with sufficient mixing. The blood was then centrifuged and the supernatant discarded. Concentrated red blood cells were obtained by repeating centrifugation three times and retaining the lower fluid. A few drops of concentrated erythrocytes were added to normal saline and diluted into a red blood cell suspension.

Each type of cell image was obtained by cell suspension flowing on the microfluidic chip at 1 ml/min in 64 ms through our flow cytometer imaging system individually (Fig. 7). The data are reconstructed and processed by MATLAB® 2014b running on a PC with CPU frequency of 2.20 GHz and 8-G memory.
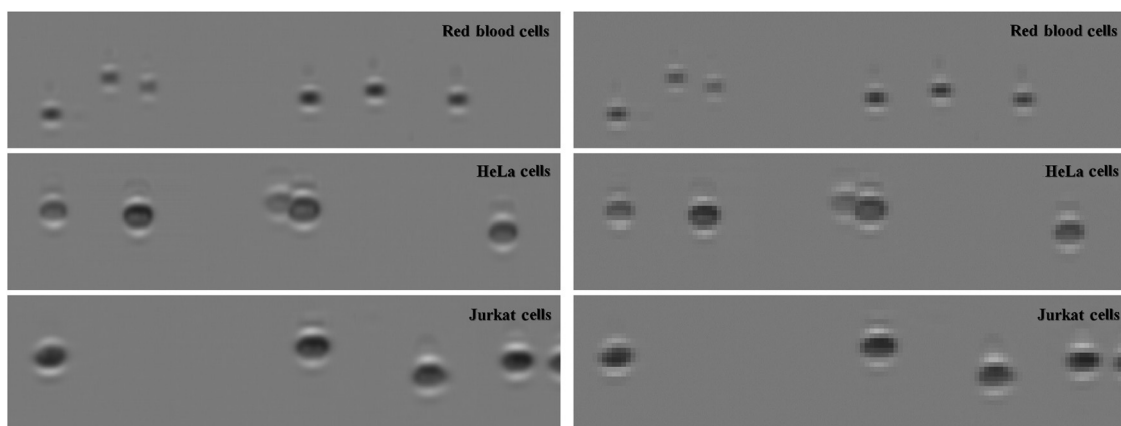
Fig. 7 Image obtained by the ultrafast flow cytometer imaging system showing three types of cells: the left figure is the original image and the resolution of the right figure is 1/10 of the left.

Table 1 The cell detection and classification result of different algorithms.

		Red blood cells	HeLa cells	Jurkat cells	Grand total	Running time (s)
Two-stage detection with GMM classification	Detected cell number	676	415	342	1433	73.2
	False positive (%)	1.60	5.49	5.40	3.63	
	False negative (%)	1.89	6.44	8.24	4.73	
SVM pattern learning algorithm	Detected cell number	676	416	338	1430	190.3
	False positive (%)	0.73	5.25	4.83	3.01	
	False negative (%)	2.61	5.97	7.39	5.07	

**Table 2** The cell detection and classification result of low-quality images.

		Red blood cells	HeLa cells	Jurkat cells	Grand total	Running time (s)
Two-stage detection with GMM classification	Detected cell number	689	405	328	1422	46.2
	False positive (%)	4.06	12.41	11.65	8.29	
	False negative (%)	4.06	15.75	18.47	10.89	
SVM pattern learning algorithm	Detected cell number	687	401	319	1407	87.8
	False positive (%)	5.52	11.93	12.22	8.97	
	False negative (%)	5.81	16.23	21.59	12.60	

The total running time of the two-stage detection algorithm and classification was around 70 to 78 s. The comparison group classified these cells with SVM pattern learning algorithm. The result of classification is shown in Table 1.

Compared to the SVM method, results in Table 1 showed that the presented algorithm increased the running speed by over 150% without sacrificing recognition accuracy. The presented algorithm missed less cells than SVM algorithm. But, its false positive rate was higher. The reason was that the two-stage cascaded detection method would detect some blurry cells or cells in irregular shape, which SVM would miss but may cut large cells into several small ones, which increased the false positive rate. Analyzing the false detection rate, we found that most of undetected cells had small intensity because they flew slightly away from the focal plane. Very few cells were wrongly detected because they overlapped each other too much to be divided. Comparing the recognition rate of red blood cells with the other two types of cells, red blood cells were easier to classify because they were much smaller. SVM algorithm performed a little bit better while distinguishing HeLa cells and Jurkat cells because it would catch more detailed features than our algorithm.

In flow cytometry imaging, the signal-to-noise ratio is usually stable, because of the constant imaging optical path and little impurities in the suspension. The most important impact on image quality is the resolution of imaging affected by sampling rate and resolution limit of the optical system. Therefore, we repeated the recognition experiment described above while reducing image quality as shown in the right figure of Fig. 7 to verify the recognition capability of the presented algorithm on low-quality images. The result is shown in Table 2.

Results in Table 2 showed that the greater the amount of data, the better the presented algorithm could help to speed up the calculation. Moreover, the performance deterioration of the presented algorithm was less severe than SVM algorithm. One reason was that the recognition capability of our algorithm
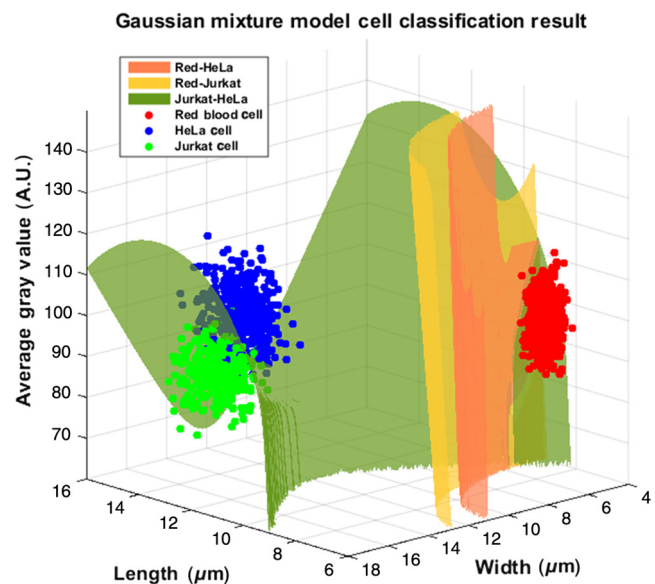
Fig. 8 Classification result of all cells. The surfaces in the figure are the decision surfaces between every two kinds of cells. The decision surface consists of points having the same Mahalanobis distances from every two Gaussian distributions.
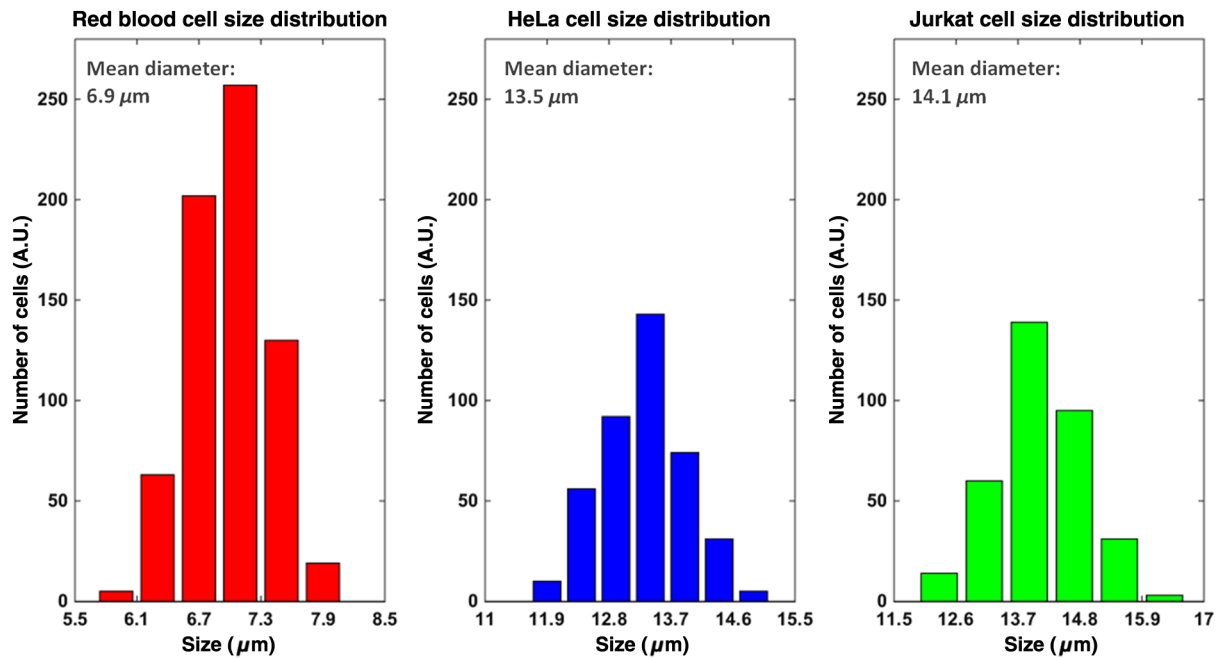
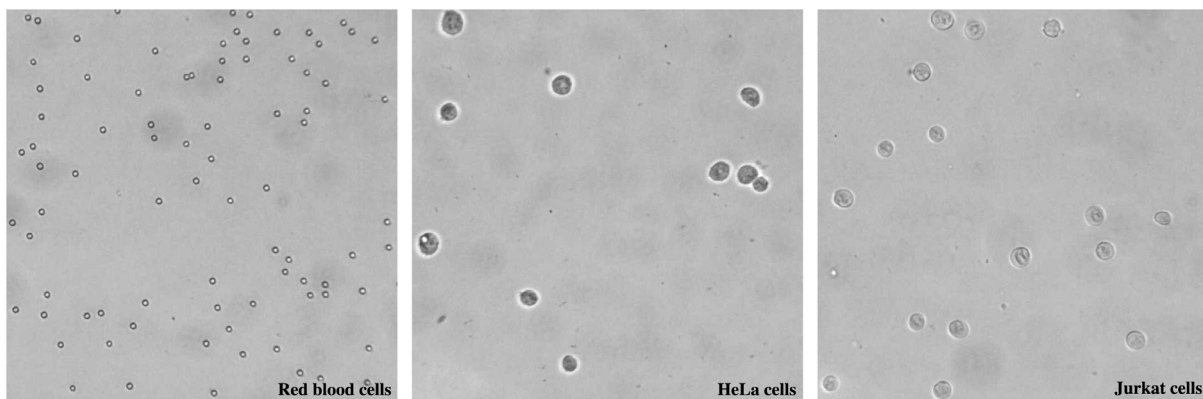**Fig. 9** Cell size statistics.

Fig. 10 Cell image obtained by microscope.

caught up with SVM while the image lost detailed information. The other reason was that SVM would miss more cells in low-quality images while two-stage cascaded detection method would detect more of them by adjusting thresholds. It was proved that our algorithm had more resistibility to the deterioration of picture quality than SVM algorithm.

In Fig. 8, we can see that the decision surfaces calculated by the GMM divided the data points of the three kinds of cells into three groups. Some HeLa cells and Jurkat cells near the decision surface were not correctly identified due to their similar morphology. The cell size statistics is shown in Fig. 9. The size of each kind of cells complied with Gaussian distribution. Figure 10 shows the microscope images of the cell suspension tested by the flow cytometer imaging system. The three groups of cells had mean diameter of 6.9, 13.5, and 14.1 μm, which accorded with the diameter of human red blood cells, HeLa cells, and Jurkat cells, respectively. We expected the ratio among red blood cells, HeLa cells, and Jurkat cells to be $2:1:1$ while preparing cell suspension for experiment. According to our experiment result, the ratio was $1.98:1.18:1$. The difference may result from cell counting error and the inhomogeneity of the cell solution flowing in our imaging system.

5 Conclusion

In summary, a two-stage cascaded cell detection algorithm combining distance transform and watershed algorithm followed by GMM classification is designed and implemented. The main advantage of the proposed algorithm is its high efficiency and reduced computing time due to the fact that background of cell images is homogeneous and can be easily removed, avoiding time-assuming prior training, sliding-window, and massive convolution operations of pattern recognition as usually used in existing flow cytometer systems. The presented algorithm greatly improves the classification speed and maintains cell screening accuracy even if image quality deteriorates significantly. It can also be adapted to image different kinds of cells or particles, which also helps to put this system into practice.

The proposed system is targeted for blood diagnostics or other body fluid diagnostics, such as the detection of cancer

cells in trace amounts that flake off of organs into blood in the early stage of cancer. Compared with traditional cancer detection methods, cancer cells can be found earlier, more conveniently, and more accurately. However, according to the analysis of our experiment results and recent researches, imaging flow cytometry with image recognition can achieve high recognition rate but cannot guarantee 100% accuracy. Other conventional inspection methods are needed to assist detecting sparse events accurately, such as early cancer cells in the blood. Target cells can be gathered by combining our high-speed cell recognition algorithm with cell sorting. In this way, the concentration of target cells can be greatly improved, which would facilitate the following cell-stain and smear microscopy or cell labeling with fluorescent markers for accurate diagnosis. As cell sorting is required, it is necessary to detect and classify cells in real time, which proves the significance of high-speed cell recognition algorithm. Furthermore, optimization of the algorithm, including abandonment of unnecessary background information and algorithm's implementation on field-programmable gate array platforms, would be explored to address the challenge of classifying and sorting cells in real time.

Disclosures

This study claims no conflicts of interest.

References

1. C. L. Chen, A. Mahjoubfar, and B. Jalali, "Optical data compression in time stretch imaging," *PLoS One* **10**(4), e0125106 (2015).
2. K. Tsia, "Performance of serial time-encoded amplified microscope," *Opt. Express* **18**(10), 10016–10028 (2010).
3. C. Lei et al., "Time-stretch high-speed microscopic imaging system based on temporally and spectrally shaped amplified spontaneous emission," *Opt. Lett.* **40**(6), 946–949 (2015).
4. K. Goda, K. K. Tsia, and B. Jalali, "Serial time-encoded amplified imaging for real-time observation of fast dynamic phenomena," *Nature* **458**(7242), 1145–1149 (2009).
5. C. L. Chen et al., "Deep learning in label-free cell classification," *Sci. Rep.* **6**(1), 21471 (2016).
6. B. Guo et al., "High-throughput accurate single-cell screening of Euglena gracilis with fluorescence-assisted optofluidic time-stretch microscopy," *PLoS One* **11**(11), e0166214 (2016).
7. H. Permuter, J. Francos, and I. Jermyn, "A study of Gaussian mixture models of color and texture features for image classification and segmentation," *Pattern Recognit.* **39**(4), 695–706 (2006).
8. X. Liu, J. Xiang, and B. Yang, "A labeling algorithm based on 8-connected boundary tracking," *Comput. Eng. Appl.* **23**(2), 125–126 (2001).
9. G. Borgefors, "Distance transformations in digital images," *Comput. Vision Graphics Image Process.* **34**(3), 344–371 (1986).
10. P. Soille, *Morphological Image Analysis: Principles and Applications*, Springer, Germany (2003).
11. M. A. Butt and P. Maragos, "Optimum design of chamfer distance transforms," *IEEE Trans. Image Process.* **7**(10), 1477–1484 (1998).
12. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discovery* **2**(2), 121–167 (1998).

Biographies for the authors are not available.