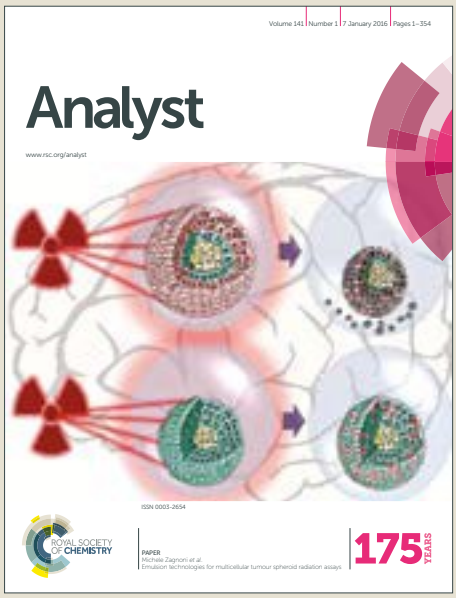
Analyst

Accepted Manuscript

This article can be cited before page numbers have been issued, to do this please use: M. Ossandon, J. Balsam, H. A. Bruck, K. Kalpakis and A. Rasooly, *Analyst*, 2017, DOI: 10.1039/C6AN02517J.



ROYAL SOCIETY OF CHEMISTRY

rsc.li/analyst

View Article Online
DOI: 10.1039/C6AN02517J

Computational streak mode cytometry biosensor for rare cell analysis

Miguel Ossandon[1, 2], Joshua Balsam[3], Hugh Alan Bruck[3], Konstantinos Kalpakis[2] and Avraham Rasooly[1*]

[1]National Cancer Institute, Rockville, MD 20850

[2]University of Maryland Baltimore County

[3]University of Maryland, College Park, MD 20742

[*]Address correspondence to: Miguel Ossandon, NIH/NCI, 9609 Medical Center Drive Rockville, MD 20850, Phone: (301) 240-276-5714 e. mail: ossandom@mail.nih.gov

View Article Online
DOI: 10.1039/C6AN02517J

ABSTRACT

Streak mode imaging flow cytometry for rare cell detection involves imaging moving fluorescently labeled cells in video mode with a CCD camera. The path of the moving cells results in a "streak", whose length is proportional to the exposure time. The dynamic imaging conditions introduce detection challenges (e.g., images with high signal-to-noise ratio (SNR) backgrounds), especially for enumerating cells using low resolution webcams or smartphone cameras suitable for point of care testing (POCT).

To overcome the imaging challenges, a new approach called a "computational biosensor" was developed. It involves combining biosensing hardware with computational algorithms to "computationally transduce" measureable signals from events captured by the hardware. The computational biosensor quantifies potential cells based on the streak intensity, length and relative location of the streaks in consecutive frames. Cell identification consists of three parts: (1) finding streaks, (2) identifying candidate cells, and (3) filtering out spurious cells to identify true cells.

Samples of 1 cell/mL were analyzed in batch sizes of 30 mL at flow rates of 10 mL/min and imaged at 4 frames per second (fps). The detected cells were annotated, and the SNR was calculated. For images with SNR greater than 4.4 dB, the total detected cells (TD) compared with ground truth (GT) is 98%, while 66% were detected for low SNR. For true positive cells detected compared with ground truth (TP/GT), 91% were detected for high SNR. This demonstrated the new analytical capabilities of the computational biosensor to enumerate rare cells in large volumes not possible with current technologies.

Keywords: flow cytometry, image analysis algorithm, microfluidics, fluorescence imaging, circulating tumor cells, image enhancement

View Article Online
DOI: 10.1039/C6AN02517J

1. INTRODUCTION

Detection of rare cells, including circulating tumor cells (CTCs), have many promising clinical applications including: cancer prognosis, early detection of metastasis-capable malignancy, guidance in selection of treatment and treatment efficacy (to help prevent patients from being exposed to ineffective therapeutics), and treatment monitoring [1, 2]. Flow cytometry, which is commonly used for cell analysis, is not well suited for detection of rare cells, because large sample volumes are needed due to the low cell concentration. Most current flow cytometers utilize microfluidic sheathing integrated with optics through a technique known as "hydrodynamic focusing". The latest technologies in this area allow the analysis of several thousand particles per second [3, 4]. However, as the cell concentration decreases, these techniques require higher flow rates than the device can accommodate to process a large enough volume to detect the cells in a reasonable period of time. Several approaches have been proposed to address the challenges of large volumes analysis in flow cytometry [5]. A low cost and simple approach can be based on Lab on a Chip with CMOS or CDD-based multichannel detectors [6-8] suitable for point of care testing (POCT).Recently, optofluidic fluorescent imaging cytometry on a cell phone with a spatial resolution of 2 μm was described [9]. While very mobile and versatile, the flow rate of this system is 1 μL/min, which limits analysis to small volumes.

Mobile phones have been effectively used as mobile POCT devices in mHealth applications[10, 11]. However, the cameras in these phones have less versatile optical systems than webcams or cameras (i.e., they do not have interchangeable lens systems). They also have limited computational capabilities compared to laptop or tablet platforms. To adapt flow cytometry to analysis of larger sample volumes that are associated with rare cell detection, a streak mode imaging, wide-field flow cytometer for high volume/high throughput based on low cost high noise camera was developed using a webcam [12, 13] (figure 1).

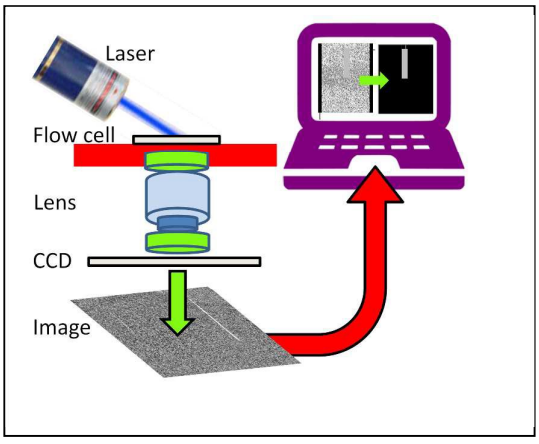


Figure 1; Schematic of webcam-based wide-field flow cytometer *– The flow cytometer consists of a sensing element, excitation source and flow cell. The sensing element consists of the internal elements of a webcam, a 12 mm f/1.2 CCTV lens, two green emission filters, and a computer to collect and analyze data. The excitation source is a low cost 450 nm 1 W laser module.*

View Article Online
DOI: 10.1039/C6AN02517J

Streak mode imaging is based on video imaging flow cytometry using a wide flow cell to achieve high throughput flow and laser excitation of a fluorescently stained cell signal for analysis by a computer. The high flow rate moving cells are imaged with a webcam in low frame rate exposure, so the path of the moving cells results in a "streak", whose length is proportional to the exposure time. The wide high throughput flow cell was used instead of the conventional narrow hydrodynamic focusing cells used in traditional flow cytometry. The wider flow field enables analysis of higher volumes (e.g., 20 ml instead of 20 ul) at lower flow rates, which is not possible with most conventional flow cytometers. The wide-field flow cytometer is based on a technique used in Particle Image Velocimetry (PIV) known as "streak photography". In streak photography exposure times and flow velocities are set such that the particles are imaged as short "streaks"[12]. Since streaks are imaged with large number of pixels, they are easily distinguished from the noise, which appears as "speckles". This increases the detection capabilities of the device, making it more suitable for analysis using current low sensitivity, high noise webcams or mobile phone cameras. In addition, since the images are taken at low speed, the file size is reduced by a factor of 40[12]. However, while the wide-field streak cytometer enabled the development of a low cost streak mode imaging wide-field flow cytometer for point-of-care testing which is capable of rapid counting at very low cell concentrations (e.g., 0.89 cell/mL)[12, 13], it required tedious manual counting of the cells which is very time consuming and subjective. So, there is a need to automate the cell counting for streak mode imaging wide-field flow cytometry.

In the last few years, several approaches for automated cell detection and tracking were developed, and commercial and public domain software tools and methods for cell motion tracking and counting have been intensively reviewed[14, 15]. General purpose tools for cell analysis, such as TimeLapseAnalyzer [16] and CellProfiler [17] and other more specialized cell tracking tools, such as CellTrack [18], CellTracker [19], and combined particle filter-active contour[20], provide a frame work for image processing/object-tracking based on traditional edge detection and segmentation methods for cell identification. Unfortunately, traditional methods such as Canny-style edge detection fail to accurately identify moving streaks over a moving background, producing a high false positive rate. Other open-source tools, such as MTrack2 (an ImageJ plugin), do not rely on automatic edge detection, but rely on manual image thresholding and filter object by size to detect and track cells. Since manual thresholding is pre-selected by the user based in average cell intensity, this tool fails to detect faint cells in backgrounds with high noise levels, such as those often encountered in streak mode imaging wide-field flow cytometry.

View Article Online
DOI: 10.1039/C6AN02517J

In this paper, we present a "computational biosensor", which is a new biosensing concept combining biosensor hardware with computational algorithms to produce new diagnostic measurements. Unlike the current use of computational analysis to process data from biosensors or computers to operate them, a computational biosensor extracts measurable signals from a biosensor that is incapable of producing those signals directly. This is accomplished through "computational transduction" of an event captured by the biosensing platform. This is similar to approaches in other measurement fields, such as optical metrology. For example, a technique known as "Digital Image Correlation", pioneered by one of the authors, utilizes correlation algorithms with deformation mapping to extract full-field deformation measurements from optical images of objects before and after they are loaded [21]. Biosensors have been classically designed to directly produce measurements based on "transduction events", such as a change in light intensity due to a chemical reaction (i.e., chemiluminescence). In our computational biosensor, the event itself is embedded within the image, and does not directly produce a signal that can be measured. Thus, we have to use computational algorithms to "reveal" and “identify” the event by using multiple images, which is a new measurement approach not relying on edge detection or manual thresholding of individual samples. Our method is based on image quantizing, morphological open/close, filtering of a 4.4 dB SNR (for background noise reduction), and a decision rule for cell identification and tracking In this paper, we describe our computational biosensor approach to overcome the challenge of accurate detection and counting of cells in streak mode imaging using a low-resolution webcam for wide-field flow cytometry suitable for POCT.

2. Materials and methods

2.1 Webcam-based flow cytometer

The device and the data used in this paper were reported previously[12, 13]. The device is a webcam-based wide-field flow cytometer (Figure 1). The webcam-based flow cytometer uses a wide flow cell for high throughput flow, instead of the conventional narrow hydrodynamic focusing cells used in traditional flow cytometry. A green emission filter with center wavelength 535 nm and bandwidth 50 nm (Chroma Technology Corp., Rockingham, VT) was used for detecting fluorescent emission. For fluorescent excitation, a 1 W 450 nm low cost laser was used (Hangzhou BrandNew Technology Co., Zhejiang, China). A Point Grey Chameleon CCD Camera with a resolution of 1296x964 and a maximum frame rate of 25 fps in region of interest (ROI) mode, and equipped with a c-mount CCTV lens (Pentax 12 mm f/1.2) was used as the photodetector. The CCD camera was connected to a 32-bit Windows-

View Article Online
DOI: 10.1039/C6AN02517J

based laptop computer via a USB2 port. The camera control software was used to set camera parameters (exposure time, frame rate, and gain) and to save video in uncompressed format. The flow cell consisted of three functional layers: (1) a plain glass or quartz microscope slide lower layer, (2) a middle layer laser machined from 3M 9770 double-sided adhesive transfer tape to define the geometry of the fluid channel, and (3) a glass or quartz microscope slide on the top layer that has two holes drilled for the inlet and outlet ports aligned with the ends of the fluid channel layer. The middle layer tape was fabricated using an Epilog Legend CO_2 65 W laser cutter (Epilog, Golden, CO) as described in our previous work [20].

2.2 Cell SYTO-9 labeling

As described in previous work[12, 13], fluorescently stained THP-1 human monocytes were used to simulate rare cells. Cells were centrifuged and re-suspended in PBS. To 1 mL of suspended cells, 10 µL of SYTO-9 dye (5 mM stock concentration) was added, and then allowed to rest at room temperature in the dark for 20 min. After labeling, cells were diluted to a level of approximately 1 cell/µL (measured by microscopy). From this relatively high concentration, lower concentrations consisting of 27 samples at 1 cell/mL were generated by single-step dilution. Samples of cells at 1 cell/mL concentration were injected into the wide flow cell in batch sizes of 30 mL with flow rate set to 10 mL/min.

2.3 Computational tools

Statistical analysis was performed with Excel. The algorithm for the computational biosensor streak detection and cell counting is implemented in MATLAB R2014b. The source code for this procedure will be posted in MATLAB central.

3. RESULTS AND DISCUSSION

3.1 Webcam-based flow cytometer

As described in material and methods, the imaging flow cytometer [20] consists of four modules (1) a webcam utilized as imaging sensor, (2) a blue 450 nm 1W laser excitation source, (3) a high throughput flow-cell, and (4) a focusing stage for image focusing and alignment. The sensor includes the internal electronics of the webcam, a 12 mm f/1.2 CCTV lens and two green emission filters. The webcam was connected to a computer which was used to power the webcam and to collect and analyze data. The fluid handling system include the flow-cell (Figure 1) and a programmable syringe pump. As discussed in our previous work [20] in order to maximize residence time of cells in the interrogation the flow channel was widen to 20 mm.

View Article Online
DOI: 10.1039/C6AN02517J

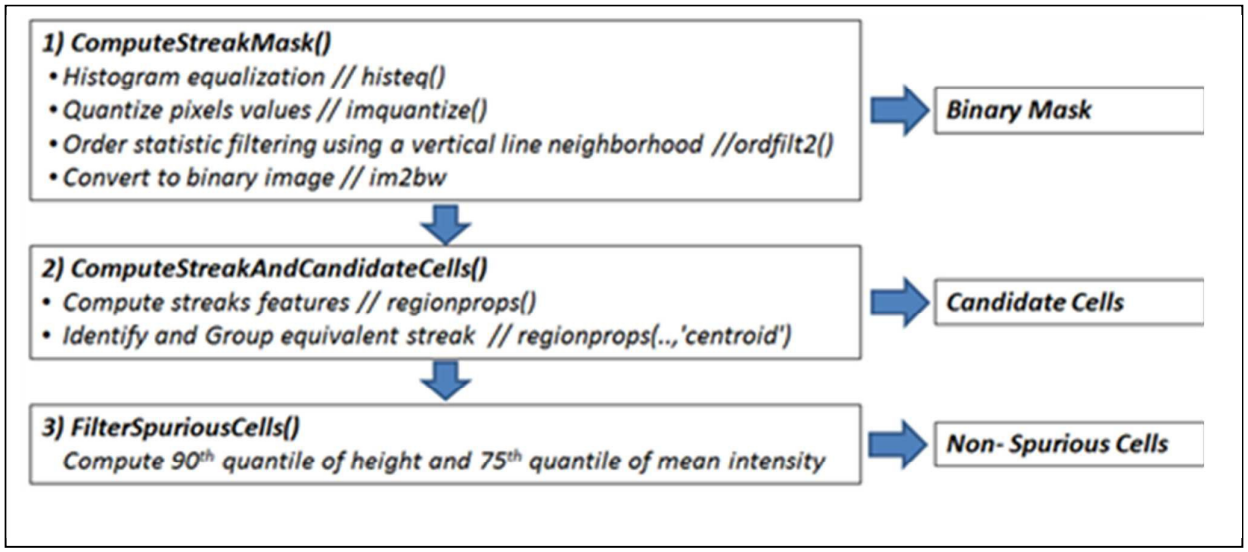
Channel depth was not increased in order to keep the flow field within the depth of field of the lens.

In order to provide approximately uniform excitation across the width of the channel, the laser source was injected into the side of the flow cell such that it formed a linear band of excitation across the center of the field of view, as seen in Figure 1. The distance between the imaging lens and the flow cell was set such that images of cells were projected onto, at most, a 3x3 array of pixels. When cells are imaged on more than one pixel, photon flux per pixel is a constant and independent of the distance between flow cell and lens. The largest possible distance was chosen in order to maximize field of view (FOV). In the setup used here, the lens was placed at a distance of approximately 20 mm from the webcam CMOS, enabling the lens to focus at very close range on the entire potential area for cell detection in the flow cell.

3.2 Computational Biosensor Streak Detection and Cell Counting Algorithm

The computational biosensor quantifies potential cells based on the streak intensity, length and relative location of the streaks in consecutive frames. The process includes: identifying true cells from streaks and matching detected cells to ground truth cells.

Our algorithm consists of three major steps: (i) create a binary mask with all potential streaks locations, (ii) identify candidate cells based on the binary mask developed in the previous step, (iii) filter out spurious candidate cells to identify true cells. These steps are illustrated in flow chart 1. After identifying the true cells, they can be matched to the ground truth cells.



Flow Chart 1: Cell Counting Algorithm *Three step process for automating streak detection and cell counting consisting of: (1) finding streaks, (2) identifying candidate cells, and (3) filtering out spurious cells to identify true cells from the streaks*

View Article Online
DOI: 10.1039/C6AN02517J

Regarding the ground truth numbers, it should be noted that if the cells are not sufficient bright they could be missed. In such a case, the cell will not be counted in the ground truth. For cases used in this paper, the ground truth cells were identified independently by three operators (and were found to be in agreement). In addition, previous work[12] showed that manual counting has high accuracy for expected concentration of 1 cell/ml and 0.1 cell/ml (manually determined to be 0.91 cell/mm and 0.083 cell/ml, respectively).

Identifying true cells by computationally detecting streaks: Cell identification consists of three parts that enable the embedded signal to be revealed and analyzed: (A) computationally detecting streaks, (B) identifying candidate cells from the streaks, and (C) filtering out spurious cells to identify true cells.

A. Computationally detecting streaks: Streaks are defined as vertical components that are expected to belong to a cell. The result of this step is a binary mask that including all possible streaks in the image frame. To create the mask, the following image enhancement procedure was used: (1) two level thresholding, (2) noise reduction using a 2D order-statistic filter, and (3) conversion to binary mask.

Thresholding: First, 40 pixels in left and right margins were removed to eliminate high noise and artifacts showed in the margins of the frame. Then, the intensity of the image was adjusted by histogram equalization and 2 threshold values of the image intensity were selected using Otsu's method [21] quantizing the image into three levels. In order to fill small gaps along the boundary foreground objects, a 3x2 pixels rectangle was slid across each frame to morphologically close the image. After this step the foreground objects (potential streaks) are more defined, but there is still high noise in the background. The result of this procedure is show in figure 2(a).

Noise Reduction: In order to reduce the background noise we performed a 2D order-statistic filtering. This consisting of sliding a 23x1 pixels rectangle across the frame, replacing the pixel value at the rectangle origins with the 3rd smallest of the pixel contained in the rectangle. The result of this operation will eliminate pixels with low values around the foreground object, but will preserve pixels with small values inside the foreground object. The resulting image is showed in figure 2(b).

Conversion to Binary Mask: The resulting three level threshold image from the previous step is converted to binary images using and all the foreground objects (potential streaks)

View Article Online
DOI: 10.1039/C6AN02517J

smaller than 50 pixels are eliminated (Figure 2(c)). Then, in order to discard extrusions from the boundary of foreground objects and eliminate non-vertical short objects, we perform morphological opening with an 81x1 rectangle. The final result of this procedure is a binary mask that contain vertical regions of the frame where

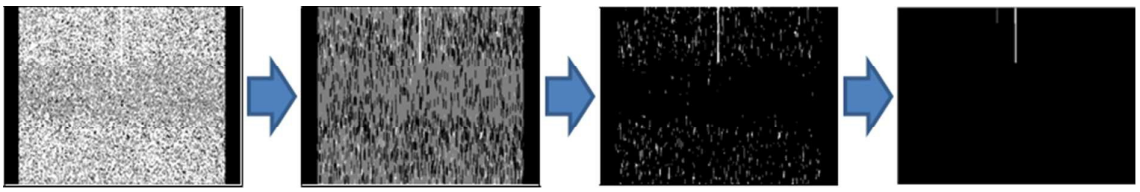


Figure 2: computational analysis for detecting streaks*: (a) and (b) Noise reduction through 2D order-statistic filtering. (c) The two thresholds image is converted to a binary image and all the foreground objects smaller than 50 pixels are eliminated. (d) Morphological opening procedure. All foreground vertical objects shorter than 81 pixels are discarded along with all extrusions from the boundary of the objects. The result is a binary mask with all potential locations of potentials streaks.*

B. Identifying candidate cells: The binary mask from the previous step is overlaid with the original image (Figure 3(a)), streaks are then identified in the original frame and enclosed in a bounding box. The following features are computed for each streak: (a) area, (b) bounding box (centroid, height, and width), (c) major and minor axes length of enclosing ellipsoid, (d) eccentricity, (e) orientation, (f) perimeter, and (g) descriptive statistics (min, max, mean, median, quantiles, variance, sum) of the values of the streak's pixels. Most of these features are provided by the "regionprops" MATLAB command. Streaks (across all frames) are grouped and labeled as equivalent if they are expected to belong to the same cell, based on the displacement of their centroids within a tolerance level. The streaks that are in the same equivalence class are now identified as a candidate cell. For each candidate cell, we compute aggregates (min, max, mean, median, range, variance) of the features of its streaks. Note that in MATLAB's image coordinate system, the x–axis (y–axis) runs along the image's width (height), increasing from left–to–right (top–to-bottom) with the (0, 0) point at the upper-and–left–most pixel. By the end of this step the streaks are annotated and candidate cells are identified for each frame (Figure 3(b)).

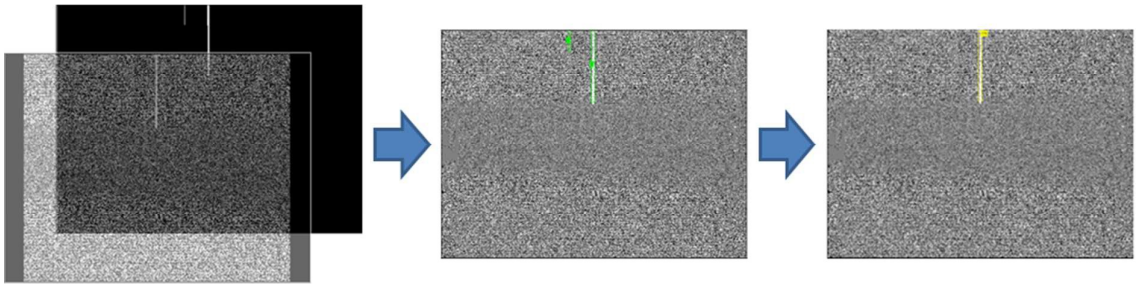
View Article Online
DOI: 10.1039/C6AN02517J

C. Filtering spurious candidate cells and identifying true cells: Candidate cells are evaluated according to intensity and length to eliminate spurious cells. For this purpose, we compute the 90th quantile of streak height and the 75th quantile of the steak mean intensity.

To determine the specific cutoff values that will be used for all the cases, we proceeded by optimizing the performance on a randomly selected case among those with streaks easily identifiable visually. We chose the values that produce the highest sensitivity while having small false positive rate. For example, in step C, since *a priori* we expect few streaks (as the nominal concentrations in 1 cell/ml), the values for the height/intensity quantile cutoff parameters are expected to be rather high. Using our process for parameter value selection, we determined that streaks with height either under 90th quantile or intensity under the 75th quantile have low probability to be a true positive cell (these cutoff values gave us the best sensitivity with small false positive rate). These cutoff values were determined by optimizing the performance of the algorithm to produce high sensitivity while having small false positive rate on a randomly selected case among those with streaks easily identifiable visually.

Changes in the parameters may affect the performance of the algorithm. For example, decreasing the intensity or height quantile cutoff value, may increase the number cell detected (true positive cells) but may also increase the number of false positive cells. On the other hand, increasing the values of these parameters may have the opposite effect on the true positive and false positive cells.

We then eliminated all the streaks in the frame with height or mean intensity below this quantile. The final result of this procedure is shown in figure 3(c) where the true cell is annotated and the spurious cell showed in the previous step (Figure 3(b)) is eliminated.



(a) *(b)* *(c)*

View Article Online
DOI: 10.1039/C6AN02517J

Figure 3: Streaks identification *(a) Overlaying the binary mask with the original image. (b)Two objects are identified and classified as candidate cells (green lines). (c) The short spurious cell is eliminated and the non-spurious (true cells) are annotated (yellow line)*

3.2 Procedure for matching detected cells to ground truth cells

In order to facilitate the automatic evaluation of various cell detection/identification algorithms, we developed a MATLAB application for matching the detected cells to the ground truth cells (GT). Ground truth cells were identified by manually reviewing each video file, and providing for each such cell the location and video frame(s) in which it occurs. For matching detected cells to the ground truth cells, the following procedure was developed. First, a complete weighted bipartite graph is built having as vertices all of the detected cells '*v'* and all the ground–truth cells '*u'* in all the video frames, and edges between *(v, u)* with weight equal to the minimum (among all frames where cells appear) of the distance between the bounding boxes of the cells corresponding to *v* and *u* (in the same frame). Next, we remove all edges with weight above a user provided threshold, and replace the edge weights *w(v, u)* with *wmax − w(v, u),* where *wmax* is the maximum edge weight. Finally, we compute a maximum weight matching in the resulting graph. This matching will associate at most one occurrence of a detected cell with at most one occurrence of a ground truth cell. Note that some detected cells may not be matched with any ground truth cell; in which case, we consider such detected cells as "false positives" (FP). Similarly, some ground truth cells may not be matched with any detected cell; in which case, we consider such cells as "false negatives" (FN). The matched detected cells are considered "true positives" (TP). Then we calculated False Positive Rate (FPR= FP/(FP+TP)), False Negative Rate (FNR= FN(FN+TP)) and True Positive Rate (TPR) = TP/(FP+TP)). In addition, we calculated the detection rate as the total detected over the Ground Truth (TD/GT) and the Sensitivity which is the ratio between TP and Ground Truth (TP/GT) for each sample (table 1).

3.3 Measuring the Efficiency of the algorithm for streak mode imaging cytometry

Our algorithm was evaluated in video files from 27 samples of 30 ml each containing 1cell/ml. Table 1 shows the results of our automated cell enumeration algorithm as compared with the ground truth. The samples are ordered by the average Signal-to-Noise Ratio (SNR) of the ground truth cells. The SNR of a cell is the maximum of the SNR's of all of its occurrences. The SNR of a cell occurrence is computed from the intensity values of the pixels within its bounding box and a noise box. The noise box is the area consists of the 5 outer pixels of 12 pixels dilation of the bounding box (7 pixels away from the signal). The signal consists of all the pixels enclosed within cell's bounding box; this ensures that pixels associated with the cell

View Article Online
DOI: 10.1039/C6AN02517J

are not used in estimating the background noise. The background noise is the average of the pixels in the noise rectangle. SNR was calculated as:

$$SNR = 10\, log_{10}\left(\frac{\sigma^2_{Sig-Bkg}}{\sigma^2_{Noise}}\right) ,$$

where Bkg is estimated as, $Bkg = \mu_{Noise}$. Using the formula above, we calculated the average SNR for each sample shown in the last column of table 1 (SNR).

Table 1: Analysis of algorithm sensitivity: calculation of 30 ml samples sorted by the average SNR of their ground truth cells. ID=Sample ID; FP=Number of false Positive cells; FN= False negative; TP=True Positive; FPR False Positive Rate; FNR= False Negative Rate; TPR=True Positive Rate; TD=Total Cells Detected; GT=Ground Truth Cells, TD/GT =Level of detection and TP/GT =Sensitivity, GT=Ground Truth Cells, TD/GT =Level of detection and TP/GT =Sensitivity

ID	FP	FN	TP	FPR	FNR	TPR	TD	GT	Detection Rate	Sensitivity	SNR
27	2	8	12	0.14	0.40	0.86	14	20	0.70	0.60	2.86
28	1	8	22	0.04	0.27	0.96	23	30	0.77	0.73	3.03
35	3	17	13	0.19	0.57	0.81	16	30	0.53	0.43	3.41
34	6	22	7	0.46	0.76	0.54	13	29	0.45	0.24	3.42
32	0	10	13	0.00	0.43	1.00	13	23	0.57	0.57	3.50
31	3	10	19	0.14	0.34	0.86	22	29	0.76	0.66	3.55
29	1	14	16	0.06	0.47	0.94	17	30	0.57	0.53	3.71
23	4	6	28	0.13	0.18	0.88	32	34	0.94	0.82	4.05
26	2	11	26	0.07	0.30	0.93	28	37	0.76	0.70	4.08
30	1	14	11	0.08	0.56	0.92	12	25	0.48	0.44	4.15
33	0	8	17	0.00	0.32	1.00	17	25	0.68	0.68	4.27
24	5	2	27	0.16	0.07	0.84	32	29	1.10	0.93	4.41
17	4	4	20	0.17	0.17	0.83	24	24	1.00	0.83	4.52
25	1	4	29	0.03	0.12	0.97	30	33	0.91	0.88	4.56
22	1	3	24	0.04	0.11	0.96	25	27	0.93	0.89	4.58
15	1	1	22	0.04	0.04	0.96	23	23	1.00	0.96	4.64
20	1	5	26	0.04	0.16	0.96	27	31	0.87	0.84	5.02
21	3	3	18	0.14	0.14	0.86	21	21	1.00	0.86	5.08
19	2	4	25	0.07	0.14	0.93	27	29	0.93	0.86	5.09
18	1	3	28	0.03	0.10	0.97	29	31	0.94	0.90	5.37
13	1	3	27	0.04	0.10	0.96	28	30	0.93	0.90	5.91
12	4	0	21	0.16	0.00	0.84	25	21	1.19	1.00	6.02
16	0	1	20	0.00	0.05	1.00	20	21	0.95	0.95	6.20
14	0	1	30	0.00	0.03	1.00	30	31	0.97	0.97	6.25
11	2	1	25	0.07	0.04	0.93	27	26	1.04	0.96	6.46
9	1	3	24	0.04	0.11	0.96	25	27	0.93	0.89	7.42
10	2	1	24	0.08	0.04	0.92	26	25	1.04	0.96	7.44

In table 1, the upper part (above the line) includes low SNR samples (< 4.4 dB), while the lower part of the table includes the samples with high SNR (>4.4 dB). From the reported results, it appears that a SNR of 4.4 dB would be desirable for best performance of the algorithm, and therefore a potential cut off value. Statistically, the average number of cells in the ground truth (GT) is 27.4 cells per 30 ml, where there is no significant difference between samples with high SNR (average 26.8 cells per 30 ml) and low SNR (28.4). However, using a

View Article Online
DOI: 10.1039/C6AN02517J

t-test (p=0.39), a significant difference between low (average 16.7) and high SNR (average 24.4) in the true positive (TP) groups (p=0.003) was found, which corroborated the 4.4 dB SNR as a cut off value.

3.4 Sesitivity of the algorithm for streak rare cell detection

We calculated the correlation between sesitivity(TP/GT) in high SNR and low SNR clustering of high and low SNR samples and performed a clustering of high and low SNR samples. The data suggested that *i*n samples with high SNR, there is a high concordance between the TP and the GT with a narrow range of cells detected (between 18 to 30 cells per 30 ml as seen in figure 4(a). A t-test shows no significant difference between the TP and GT for high SNR ($p>0.07$). The coefficient of variation (CV) for these samples is also similar (0.1420 for TP and 0.1486 for GT), suggesting that for samples with high SNR, the algorithm counting is accurate. However, for samples with low SNR, there is a lower concordance between TP and GT with a wider range of cells detected (between 7 to 20 cells per 30 ml as seen in figure 4(b), which indicates the extent of variability in these groups. Moreover, there was significant difference between TP and GT ($p=0.00007$) in the low SNR group, where the coefficient of variation for these groups was 0.39 for TP and 0.17 for GT (similar to the CV of 0.1486 for the high SNR GT). So, samples with low SNR show greater variability and higher CV for TP. For the total detected cells (TD) compared to the ground truth (TG) (table 2), the ratio TD/GT for high SNR is 0.98 (suggesting that 98% of the cells can be counted) and for low SNR the ratio is 0.66. However, when comparing true cells detected (TP) with the GT (sensitivity), the TP/GT ratio for high SNR is 0.91 and for low SNR is 0.59 (table 2). The reason for the difference is that the TD value may include false positive cells. The plot of sensitivity vs true positive rate (figure 4(c)) shows the clustering of high and low SNR samples. High SNR samples (orange dots, cluster I) have high sensitivity and high true positive rates, while samples with low SNR (blue dots, cluster II) have lower true positive rates. Possible reasons for a low SNR include low quality imaging and thermal noise of the CCD compounded by longer exposure times. These imaging issues can be resolved using lower thermal noise detectors, decreasing exposure time, or increasing the number of measurements using larger numbers of samples by lowering the volume of each sample.

View Article Online
DOI: 10.1039/C6AN02517J

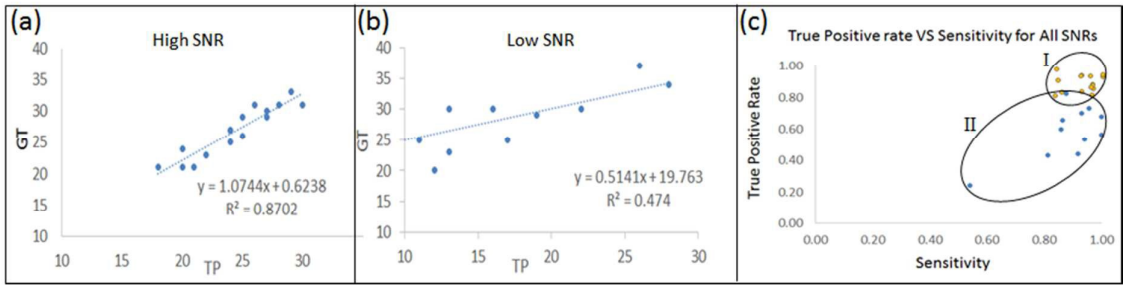


Figure 4: The correlation between ground truth cells (GT) and Total Cells Detected (TD); (a) The correlation between TP and GT in high SNR (b) The correlation between TP and GT in low SNR (c) clustering of high and low SNR samples.

Table 2: Summary of the performance of the algorithm: *The table shows false positive rate (FPR), false negative rate(FNR), true positive rate (TPR), sensitivity (TP/GT) and detection rate (TD/GT) at different levels of SNR. The algorithm performs well on samples with SNR greater of equal to 4.41 dB*

SNR	FPR	FNR	TPR	Sensitivity	Detection Rate
>= 4.41 dB	0.07	0.09	0.93	0.91	0.98
< 4.41 dB	0.12	0.42	0.88	0.59	0.66
All SNRs	0.09	0.22	0.91	0.78	0.85

The algorithm developed in this paper is specific for this instrument. However, the general principles of the algorithm can be applied to other streak imaging applications as well. For other applications, we may need to apply cut off values different from the one determined in this paper, or we may to include parameters not considered in this paper. However, the fundamental principles of the algorithm to evaluate streak images can still be translated to new devices. Therefore, such cut off values could be determined using the approach presented in this paper.

The proof-of-concept described here using a single wavelength fluorescence excitation and single wavelength emission measurements is limited to the detection of a single marker. However, detection of CTCs and other cells normally require multiple fluorescent markers (e.g. EpCAM+ and CD45) using multiple wavelength excitation and multiple wavelength emission measurements. Alternative optical approaches, some of which are simple, could be used to adapt our approach to detection of multiple fluorescent markers. For example, a beam splitter with two cameras, each with a different emission filters, would be capable of capturing signals from two different wavelengths.

View Article Online
DOI: 10.1039/C6AN02517J

Another approach, that is slightly more complicated, is to use alternating multiple excitation sources (e.g., alternating light source or alternating excitation filters) each specific to a fluorophore wavelength excitation. If the sources are synchronized with the video imaging, then it would be possible to detect multiple fluorescent markers, and to assign a cell image to each fluorophore. In this case, streaks will not be continuous, but will have gaps depending on the frequency of the light source. In this case, our detection algorithm can be easily extended by: (1) running the algorithm (streak and cell detection steps) on image sequences of each fluorescent marker independently, and (2) running the spurious cell elimination step by combining cells identified by any of the markers.

CONCLUSION

We have developed a computational biosensor for automated cell counting for streak mode imaging in a wide-field flow cytometer that we previously developed for POCT. The algorithm consists of three parts: (1) finding streaks, (2) identifying candidate cells, and (3) filtering out spurious cells to identify true cells. To find streaks, the following image enhancement procedures were used: (1) a two level threshold to expose foreground objects, (2) a 2D order-statistic filter to reduce noise, and (3) conversion to a binary mask containing just streaks. The binary mask is overlaid on the original image to identify candidate cells, and then spurious cells are filtered based on the streak intensity and length.

Our algorithm performed well (average 91% sensitivity) in the detection of cells in concentrations of 1 cell/ml in samples with SNR greater or equal to 4.41 dB (table2). For samples with SNR less than 4.41 dB the algorithm failed to detect ~ 41% of the true cells (sensitivity 0.59) (table 2). Our algorithm for cell detection relies heavily on the ability to differentiate cells from noise in each frame, therefore the algorithm performed poorly for faint cells. In order to detect these faint cells other techniques based on recognition of pixel spatial patterns and expected relative location in the frame may be used. Possible reasons for low SNR include low quality imaging and thermal noise of the CCD compounded by the long exposure time. The use of higher quality imager (e.g. cool CCD) will improve SNR and detection level, however even with low quality imaging the technology has the unique capability of directly analyzing low number of cells (e.g. 1 cell/ml) in large volume (e.g. 30 ml).

In addition, the algorithm in this work is implemented in Matlab, and heavily depends on the software's dedicated libraries. While this implies license fees that are incompatible with resource-poor settings, Matlab was used for prototyping and as a proof-of-principle for the algorithm. It is envisioned that customized code can be developed using open source C++

View Article Online
DOI: 10.1039/C6AN02517J

libraries, such as OpenCV, in order to use the algorithm in resource-poor settings. In addition, the technology can eventually be adapted to mobile phones, which are more common than webcams in resource-poor settings. A resource-poor solution could ultimately aim for processing the images directly on the phone, but this would require modifying the algorithm and code to be suitable with such a resource, which may be too challenging. A more reasonable alternative is to use remote processing to overcome the phone's limited computational power to perform complex analysis. For remote processing, the mobile phone can be used to transmit the data in the form of a video file to a more powerful computational platform, which is capable of using the current algorithm for processing the images.

The computational biosensor we have developed for automated detection and quantification of rare cell concentrations is of interest for POCT, but not limited to POCT applications. There are many scientific, clinical and industrial applications for low number of cell counting. For clinical microbial applications, detection and analysis of low cell concentrations will enable earlier detection and treatment of microbial infections, such as blood stream infection or sepsis, and will prevent spreading of infectious diseases, such as *Mycobacterium tuberculosis, Streptococcus, Pseudomonas* or *chlamydia* the leading cause of sexually transmitted infection (STD).

For food safety early detection of microbial contamination of food borne pathogens, such as *shigella, campylobacter, salmonella* or *E. coli*, enables corrective measures to be taken before outbreak of foodborne diseases. Similarly, early detection of microbial pathogens enables prevention of the spread of microbial plant pathogens. SYTO 9 green fluorescent nucleic acid stain can be used for both DNA and RNA. It has been used for bacteria staining [22, 23], including stain of live and dead Gram-positive [24] and Gram-negative bacteria[25], and it is a component of LIVE/DEAD Viability Kits, including BacLight Bacterial Viability Kits. SYTO-9 was detected by fluorescence microcopy and flow cytometry. In these SYTO-9 assays, a single bacteria can be detected. While we did not test the method with bacteria, based on the wide use of SYTO 9 for bacterial staining, the technology can most likely be used for bacterial detection, including food safety and sterility analysis. Wide-field flow cytometry combined with cell streak mode imaging results in a simpler, cheaper, and more portable flow cytometer which may facilitates the expansion of cell-based clinical diagnostics, especially in resource-poor settings.

View Article Online
DOI: 10.1039/C6AN02517J

ACKNOWLEDGEMENTS

This work was supported by the National Cancer Institute (NCI)

References

1. Z. Zhang, *Oncotarget*, 2014, ePub.
2. C. Alix-Panabieres and K. pantel, *Clinical Chemistry*, 2013, **59**, 110-118.
3. J. P. Golden, J. S. Kim, J. S. Erickson and F. S. Ligler, *Lab on a chip*, 2009, **9**, 1942-1950.
4. P. B. Howell, J. P. Golden, L. R. Hilliard, J. S. Erickson, D. R. Mott and F. S. Ligler, *Lab on a chip*, 2008, **8**, 1097-1103.
5. E. K. Zuba-Surma and M. Z. Ratajczak, *Methods Cell Biol*, 2011, 207-230.
6. M. M. Ngundi, S. A. Qadri, E. V. Wallace, M. H. Moore, M. E. Lassman, L. C. Shriver-Lake, F. S. Ligler and C. R. Taitt, *Environmental science & technology*, 2006, **40**, 2352-2356.
7. S. Sun, M. Ossandon, Y. Kostov and A. Rasooly, *Lab-on-a-chip*, 2009, **9**, 3275-3281.
8. S. Sun, J. Francis, K. E. Sapsford, Y. Kostov and A. Rasooly, *Chemical*, 2010, **146**, 297-306.
9. H. Zhu and A. Ozcan, *Journal of visualized experiments*, 2013, **JOVE(74)**, e50451.
10. I. Navruz, A. F. Coskun, J. Wong, S. Mohammad, D. Tseng, R. Nagi, S. Phillips and A. Ozcan, *Lab on a chip*, 2013, **13**, 4015-4023.
11. Q. Wei, H. Qi, W. Luo, D. Tseng, S. J. Ki, Z. Wan, Z. Gorocs, L. A. Bentolila, T. T. Wu, R. Sun and A. Ozcan, *ACS nano*, 2013, **7**, 9147-9155.
12. J. Balsam, H. A. Bruck and A. Rasooly, *Biosens Bioelectron*, 2015, 154-160.
13. J. Balsam, H. A. Bruck and A. Rassoly, *Analyst*, 2014, 4322-4329.
14. E. Meijering, O. Dzyubachyk and I. Smal, in *Methods in Enzymology*, Elsevier, 2012, vol. 504, ch. 9, pp. 183-200.
15. A. J. Hand, T. Sun, D. C. Barber and D. R. Hose, *Journal of Microscopy*, 2009, **234**, 62-79.
16. J. Huth, M. Buchholz, J. M. Kraus and H. A. Kestler, *Computer Methods and Programs in Biomedicine*, 2011, **104**, 227-234.
17. A. E. Carpenter, T. R. Jones, M. R. Lamprecht, C. Clarke, I. H. Kang, O. Friman, D. A. Guertin, J. H. Chang, R. A. Lindquist, J. Moffat, P. Golland and D. M. Sabatini, *Genome Biol*, 2006, **7**, R100.
18. A. Sacan, H. Ferhatosmanoglu and H. Coskun, *Bioinformatics*, 2008, **24**, 1647-1649.
19. H. Shen, G. Nelson and D. B. Kell, *Chemometrics and Intelligent Laboratory Systems*, 2006, **82**, 276-282.
20. J. Balsam, H. A. Bruck and A. Rasooly, *Analyst*, 2014, DOI: 10.1039/c4an00669k.
21. M. Sezgin and B. Sankur, *Jurnal of Electronic Imaging*, 2004, **13**, 146-168.
22. M. Mols, M. Ceragioli and T. Abee, *Int J Food Microbiol*, 2011, **151**, 119-122.
23. P. Hildebrandt, K. Surmann, M. G. Salazar, N. Normann, U. Volker and F. Schmidt, *Cytometry A*, 2016, **89**, 932-940.
24. V. Liato, S. Labrie and M. Aider, *J Ind Microbiol Biotechnol*, 2016, DOI: 10.1007/s10295-016-1859-y.
25. J. S. Kim, E. J. Lee and Y. J. Kim, *Foodborne Pathog Dis*, 2014, **11**, 645-651.