

Segmentation and Track-analysis in Time-lapse Imaging of Bacteria

Sajith Kecheril Sadanandan, Petter Ranefall, Klas E. G. Magnusson, *Student member, IEEE*, Alexis Baucharin, Özden Baltekin, Joakim Jaldén, *Senior member, IEEE*, Johan Elf, Carolina Wählby

Abstract—Genetically identical cell express different amounts of their genes and respond differently to change in their environment. Large-scale microscopy-based experiments are needed to characterize the dynamics of such cell-to-cell variability as well as the phenotypic diversity in response to perturbations in growth conditions. The rich data from image-based experiments requires robust and efficient analysis. In this work, we have developed tools to analyze bacterial cells growing in monolayers in a microfluidic device. Individual cells are identified using a novel curvature based approach and tracked over time for several generations. The resulting tracks are thereafter assessed and filtered based on track quality for subsequent analysis of bacterial growth rates. The proposed method performs better than the state-of-the-art methods for segmenting phase contrast and fluorescent images, and we show a 10-fold increase in analysis speed.

Index Terms—*E. coli*, segmentation, time-lapse, microscopy, tracking.

I. INTRODUCTION

LIVE cell experiments pave the way to understand the complex biological functions of living organisms. Many live cell experiments require monitoring of cells under different conditions over several generations. Isogenic cells display cell-to-cell variability even when grown under similar conditions [1]. To study the origin and consequences of such variation it is necessary to monitor many individual cells for extended periods of time to reach statistically testable conclusions [2]. Time-lapse experiments usually generate large quantities of data, which become extremely difficult for human observers to evaluate in an unbiased way [3]. Thus, automated systems are necessary to analyze such datasets in order to reach robust and reproducible results.

Time-lapse imaging of growing bacterial cells are important both to answer fundamental biological questions related to the bacterial cell cycle as well as to study response to changes in growth condition due to changes in nutrients or antibiotics [4]. Based on the growth conditions and imaging modalities, various automated image segmentation and tracking packages were developed. For example *MicrobeTracker* [5] was designed to segment phase contrast images and detect fluorescent

S. K. Sadanandan, P. Ranefall and C. Wählby are with the Department of Information Technology, Uppsala University, Sweden and SciLifeLab, Sweden. C. Wählby is also with Broad Institute of Harvard and MIT, Cambridge, MA, USA. e-mail: sajith.ks@it.uu.se

A. Baucharin, Ö. Baltekin and J. Elf are with the Department of Cell and Molecular Biology, Computational and Systems Biology, Uppsala University, Sweden

K. E. G. Magnusson and J. Jaldén are with ACCESS Linnaeus Centre, KTH Royal Institute of Technology, Stockholm, Sweden

spots in a parallel fluorescent channel in bacterial cells grown isolated or in micro-colonies on agarose pads. *Schnitzcells* [6] was specifically designed to analyze fluorescent time-lapse images of *E. coli* grown on agarose, and *MAMLE* [7] was also designed to analyze *E. coli* from phase contrast and fluorescent images.

Most image segmentation methods rely on raw pixel intensities to get an initial segmentation result, and further refined segmentation depends on this initial segmentation. *Microbe-Tracker* finds an initial segmentation using Otsu's thresholding method [8] followed by edge detection and watershed segmentation. In the final step, active contours are applied to refine object boundaries. *MicrobeTracker* needs manual correction of the first frame to get satisfactory segmentation results for time-lapse images. It is difficult to analyze large image sequences in *MicrobeTracker* due to its inherent memory problems, and it is often necessary to modify the code for practical applications. *MAMLE* uses range filtering to find an initial segmentation result, followed by multi-scale edge detection and a maximum likelihood classification to correct over- and under-segmentation. In *Schnitzcells*, initial segmentation is achieved by edge detection followed by post processing to correct segmentation errors.

Phase contrast images of *E. coli* exhibit high-intensity regions inside cellular regions comparable to, or even brighter than, regions between cells. Relying on raw intensity therefore leads to over- and under-segmentation at the same time. The problem becomes amplified when trying to track cells over time. Even a 1% error in detection of cells in every frame renders the cell lineage useless for further analysis if many cells are tracked over a long time.

Previously published cell tracking algorithms rely on model evolution, where a model of the cell is evolved over time using techniques such as active contour models [9] or level sets [10], and tracking by detection [11]. Tracking by detection involves two stages; segmentation and tracking. Sometimes both these steps are combined together to get final tracking result [12].

In this work, we use tracking by detection, i.e., separate the segmentation and tracking problems and solve them separately, followed by a quality control and refinement step where some of the segmentation and tracking errors are corrected. Cell segmentation is done using our novel Curvature Based Approach, hereafter called as CBA, and tracking is done using a state-of-the-art tracking algorithm [13]. In the following sections we present our segmentation methodology and compare it with that of *MicrobeTracker* and *MAMLE* on phase contrast as well as fluorescence images, from our own and previously

published experiments. After segmentation, we track the cells through the time-lapse sequence and then perform post tracking segmentation correction to get a final segmentation result. Finally, we show how the combined segmentation and tracking approach can be applied to quantify differences in cell growth rate under different experimental conditions.

II. METHODOLOGY

A. Image Acquisition

The bacterial cell colonies in our own experiments were grown on a specially designed microfluidic device in Polydimethylsiloxane(PDMS)[14] with a growth chamber, a trap, of size $40 \times 40 \times 0.9\mu m$, which is open to growth media exchange at two ends. The cells grow in a single layer and as the size of the microcolony gets larger than the trap, excess cells leave the trap through the outlet, thus maintaining the colony size of approximately 200 cells nearly constant throughout the experiment. Images were captured using an inverted microscope fitted with separate cameras for phase contrast and fluorescent channels. The microscope was equipped with a TIR based hardware autofocus that keeps the cells in focus over days. 51 traps can be monitored in parallel and growth conditions can be changed in 2 seconds [15] using computer controlled pumps. Phase contrast images were acquired every 30 seconds at $125ms$ exposure time using CFW-1312M (Scion Corporation) and fluorescent images were acquired every 60 seconds using an EMCCD camera (Andor Technologies) and DPSS laser excitation at $514nm$ (Coherent). The MG1655 bacteria were grown in M9 media supplemented with glucose or glycerol for growth rate comparison and amino acids. The fluorescent cells express turboRFP constitutively from a chromosomally integrated promoter. Bacterial cell colonies from the *MicrobeTracker* and *Schnitzcells* datasets were grown as described in [5] and [6].

B. Image Preprocessing

The input phase contrast images are of size 1360×1024 pixels and contain the cell colony as well as some regions of the microfluidic device. We aligned the image sequences based on image cross correlation to account for the stage repositioning inaccuracy that occurred when cycling though different traps during the image acquisition process. The aligned images were manually cropped to the cellular region. Since the stack was aligned, manual selection on the first frame was enough to crop the entire image stack.

C. Curvature Based Contrast Enhancement

In the input phase contrast images, the *E. coli* cells appear as dark rod-shaped objects on a brighter background as shown in figure 2. The *E. coli* colony is tightly packed so that the intensity values between the cells are often similar to those inside the cells, and it is common to see high intensity regions inside cells, which implies that any purely intensity based approach for segmentation will result in erroneous output. However, we have observed that there is a general intensity variation occurring between the cells. To detect these regions

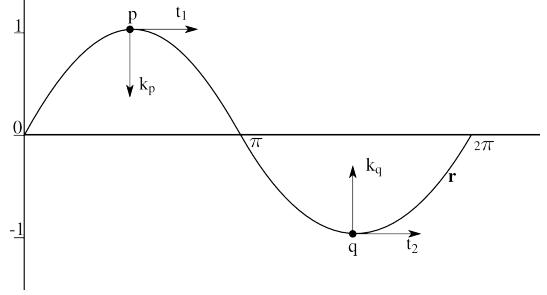


Fig. 1. k_p is the curvature at point p with negative value and k_q is the curvature at point q with positive value for the curve \mathbf{r}

of intensity variation, we used the separation of principal curvature of the intensity surface [16]. The regions were thereafter separated based on the minimal curvature as described below.

The curvature was found using techniques from differential geometry [16]. Consider a 1D case, where r is a curve as shown in figure 1 and p and q are two points on the curve. We know that the gradient of the curve with respect to the arc length gives the tangent, t , at that point, i.e., $t = r'$. The rate of change of tangent direction as we move along the curve is the curvature of the curve, i.e., $t' = kn$, where n is the unit normal vector to the curve that is perpendicular to the tangent and k is the curvature. So we have $r'' = kn$. This shows that the curvature at a point is the second derivative of the curve at that point. The sign of the curvature is determined by whether the slope is increasing or decreasing. Here we can see that it is negative in the maximum point and positive in the minimum point as shown by k_p and k_q (arrow pointing upward as positive and arrow pointing downward as negative). We extended the same idea to the 2D case. Consider a gray scale image as a surface in 3D with (x,y) being the spatial coordinates and $I(x,y)$ being the gray level intensity at that particular spatial location. Following a similar convention as for the 1D case, the image surface is assumed to be continuous with partial derivatives existing at least to order 2 [16]. Here, we first made the image smooth by convolving it with a Gaussian kernel. We set the standard deviation of the Gaussian to 1.4 pixels, which is approximately $1/10^{th}$ the width of the *E. coli* cells, found experimentally.

A particular point on the image surface has an infinite number of curves passing through it. Out of all these curves there are two curves that are particularly interesting. They are the curve with maximum curvature and the curve with minimum curvature, which are orthogonal to each other. These curvatures are equal to the eigenvalues of the Hessian matrix [17]. The Hessian is the second derivative matrix of the image, calculated for every pixels, which is created as

$$H = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$
 where I_{xx} and I_{yy} are the 2nd derivatives of the image taken in the x - and y -directions, and I_{xy} is the derivative of the image taken first in the x -direction and then in the y -direction, using discrete approximations [18]. The eigenvalues can be calculated as follows

$$k_{1,2} = \frac{\text{trace}(H) \pm \sqrt{\text{trace}(H)^2 - 4 \times \det(H)}}{2} \quad (1)$$

Here, k_1 and k_2 are the principal curvatures with $k_1 < k_2$. In phase contrast images of *E. coli*, consider two rod shaped cells lying parallel to each other, the cells are dark and the region between the cells is bright. When we calculate principal curvatures in the region between the cells, one is perpendicular to the major axis of the cell and its curvature is negative, and the other one is parallel to the major axis of the cell with zero or small value near zero curvature (positive or negative depending on local intensity values). Taking the lowest value of the two gives the curve with the greatest curvature magnitude in the negative direction at that point. In this way we can enhance the contrast of the image in bright background regions between cells while avoiding enhancing variations inside the darker cell regions, as shown in figure 2.

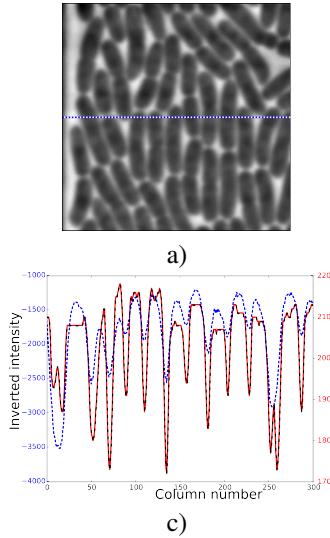


Fig. 2. a) Original input image, b) curvature based contrast enhanced image and c) plot showing pixel values from the same row from a (red line), and b (blue line). Note that the plotted pixel values in c are inverted for display.

D. Object Segmentation

We used the presented curvature-based enhancement step to enhance the contrast in the images. Next, we segmented out the cells using a repeated thresholding approach. The contrast-enhanced image is an image with floating point values. In order to make the threshold computation easier, we normalized the image and quantized it to 256 intensity levels. A single threshold value was not sufficient to separate all individual cells, and watershed segmentation resulted in ambiguities in the positioning of the edges of the cells. We therefore used multiple thresholds and prior knowledge about the cell area and the cell shape, in the form of major and minor axes lengths, to filter out the cells from background regions.

For each threshold level, the image was labeled and each object fitted with an ellipse. The ellipse parameters are found using moments [19] as follows. We create a matrix M such that,

$$M = \begin{bmatrix} m_{02} & m_{11} \\ m_{11} & m_{20} \end{bmatrix}$$

$$\text{major axis} = 4 \times \sqrt{\frac{\lambda_1}{m_{00}}} \quad (2)$$

$$\text{minor axis} = 4 \times \sqrt{\frac{\lambda_2}{m_{00}}} \quad (3)$$

λ_1 and λ_2 are eigenvalues of moment matrix M , m_{pq} is the p, q^{th} central moment in x and y axis respectively. m_{00} is 0^{th} central moment (area of the object).

The ellipse parameters of individual objects were analyzed as follows. The objects were filtered based on the major and minor axis length to remove very large and very small regions. The major and minor axes lengths are given as parameter to the algorithm. For each object, a weight was calculated as follows and assigned to the object.

$$\text{weight} = 0.5 \times \text{residual area ratio} + 0.5 \times \text{convexity} \quad (4)$$

where the residual area ratio (RAR) is found as

$$\text{RAR} = \frac{\min(\text{area}, \text{ellipse area})}{\max(\text{area}, \text{ellipse area})} \quad (5)$$

and convexity

$$\text{convexity} = \frac{\text{area}}{\text{area of convex hull of object}} \quad (6)$$

Ellipse area is found as

$$\text{Ellipse area} = \frac{\pi \times \text{major axis} \times \text{minor axis}}{4} \quad (7)$$

From a computational point of view it is time-consuming to apply a threshold at all the 256 intensity levels and analyze the size and shape of every binary object present in the image. We have already seen that the cellular regions exhibit higher intensity values and edges between cells exhibit lower intensity values on the contrast-enhanced image. Thus we made the assumption that the histogram is bimodal, with the largest peak representing object and background pixels, while the smaller peak represents edges. We therefore started our repeated thresholding at the intensity value at the local minimum of the histogram (assuming this is well below the intensity of the cells), and set the ending threshold as the intensity of the largest peak. These values may change depending on the dataset, but proved robust on the four different datasets evaluated here.

For each threshold, weight values were assigned to every object in the range $[0, 1]$, resulting in a stack of segmentation results. We reduced the stack by a maximum projection, i.e., find the maximum value for every x, y co-ordinate through the stack found, thus keeping the per-object segmentation result with the highest weight. This projection image was thresholded and holes were filled to obtain the final object segments. We found the threshold experimentally and set it to 0.75. All objects below this threshold were considered to be image background. The remaining objects were subjected to further intensity based filtering to remove artifacts and smoothing to remove sharp corners as described below.

TABLE I
PARAMETERS SET FOR MORPHOLOGICAL SNAKES

Parameter	Value
Alpha	1000
Sigma	4.2
Smoothing	3
Threshold	25
Balloon	-1
Number of iterations	2

E. Object Filtering

Our object filtering is based on the assumption that the intensity distribution of the cells follows a Gaussian distribution, and that outliers are debris. Approximations of the mean and the standard deviation of the distribution were computed as follows. All the intensity values for the particular image are normalized. From the list of normalized intensities, the histogram of intensity is found and interpolated using spline interpolation [20]. On the interpolated distribution, we find the position of the peak, use it as mean, and the half maximum of the peak on both sides of the peak using linear search. We find the full width at half maximum (FWHM) by the difference of the right and left half maximum. Then the standard deviation is thereafter estimated by [21].

$$\text{Estimated standard deviation} = \frac{\text{FWHM}}{2\sqrt{2\log(2)}} \quad (8)$$

We introduced a threshold, 3 standard deviations above the mean. All objects that had values less than this threshold were considered for further processing. We performed the intensity based filtering on all images in the sequence separately to compensate for variations in image intensity.

F. Object Smoothing

Object smoothing is required only for such applications where the segment accuracy is more important than speed of execution. The output objects from the filtered segmentation sometimes have sharp corners. To smooth the borders we applied morphological geodesic active contours [22], which is a fast level set based contour evolution method. In this method, level set evolution is done using dilation and erosion of binary images. We set the parameters as shown in Table I and used the intensity normalized contrast-enhanced image, i.e., the low eigenvalue image, as reference image for the level set evolution. All the parameters have same meaning as mentioned in [22]. Each segment was smoothed individually to avoid potential merging of segments.

G. Segmentation Evaluation Approach

We evaluated the segmentation performance using recall, precision and the harmonic mean of the two, referred to as the F-score on a per object basis. These were found as follows

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$F - score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

where TP is true positive, FP is false negative and FN is false negative.

To have a fair comparison of the evaluated methods, all objects touching the image boundary were removed. The exact positioning of the segmentation outlines may vary between different segmentation approaches, and it is not straightforward to say if a 'tight' or a more 'loose' fit of an outline is more correct. This may introduce bias to the quantification of precision and recall. We therefore adjusted the manually curated ground truth using simple dilations or erosions to make the area distribution of the cells in the ground truth as similar as possible to that of the output segmentation masks. In the label-based dilation, a square structuring element of size 3×3 was considered and the center pixel was assigned the label with maximum occurrence, excluding the background pixels.

After this adjustment of the ground truth image an object based segmentation evaluation was done. To evaluate the segmentation result per object, we created a list entry for every object. We thereafter considered each segment from the ground truth and found the segment with maximum overlap in the segmentation result from the corresponding algorithm and marked the corresponding entry in the list as visited. This was done for all the segments in the ground truth image. The segments that were not associated with any detected segment were considered as false negative cells, and similarly, detected segments that failed to get associated to any of the segments from the ground truth image were considered as false positives. This ensured that we account for missing as well as false detections. For every segment in the ground truth we find pixels that overlap with the associated segment from the respective algorithm and consider those pixels as true positive. Pixels in the ground truth segment that are not overlapping are considered as false negatives and similarly those not overlapping with the segmentation result is considered as false positive. We repeat this process on all the segmentation results. Then the histogram of the performance metrics were created at 100 bin levels as shown in figure 7 and the corresponding area distribution is shown in figure 6.

H. Cell Tracking

The segmented cells were tracked over time and cell lineages were created by a batch tracking algorithm that uses iterative application of the Viterbi algorithm [13]. The algorithm adds one cell at a time to the existing cell tracks in such a way that the probability of the tracks obtained is maximized. The algorithm uses the entire image sequence to create a track, thereby improving the tracking accuracy. In addition to tracking cells, the algorithm can handle events like cell division, cell death, cells entering and leaving the field of view, multiple cells present in a single segment, and cells that are missing in some frames. The basic parameters that were set for tracking are shown in Table II. The algorithm normally uses probabilities of cell migration events that are estimated by assuming that the centroid of a cell in one frame follows

TABLE II
PARAMETERS SET FOR TRACKING ALGORITHM

Parameter	Value
probability that a segment contains 0 cells	0.000001
probability that a segment contains 1 cell	0.93
probability that a segment contains 2 cells	0
probability of mitosis	0.20
probability of cell death	0
standard deviation of the cell velocity in pixels per frame	20

a Gaussian distribution centered on the centroid of the cell in the previous image. This does however not work well for rod shaped bacteria, as the centroids of adjacent bacteria can be very close to each other. To overcome this problem, we replaced the migration probability with the Jaccard similarity index of the binary masks of the segmented regions in the two images. The Jaccard similarity index is computed as the number of pixels in the intersection of the regions divided by the number of pixels in the union of the regions, and therefore takes values between 0 and 1.

I. Cell Track Analysis

The tracking algorithm tracks cells over time from the first frame to the last frame. For our present experiments, we want to analyze the cell growth over time from the birth of a cell to its division. Due to noise and the complexity of the dataset, many tracks are generated, and not all of them are correct. Rather than keeping all tracks, we try to sort out correct tracks based on a priori information. Out of all the tracks that were generated, we filtered out only those tracks that start with a cell division and end with the next cell division. We used an area measure to confirm the cell division at the beginning and at the end of each track. At the beginning of the track we check if the area of the parent cell is within a tolerance level to that of the sum of the areas of the current cell and its sibling cell. Similarly, at the end of the track, we check if the area of the current cell at the end of the track is within a tolerance to the sum of the areas of the child cells. The error for parent and child cells are calculated as follows

$$\text{child error} = \left| 2 - \left\{ \frac{\text{final area}}{\text{children area}} + \frac{\text{children area}}{\text{final area}} \right\} \right|$$

$$\text{parent error} = \left| 2 - \left\{ \frac{\text{siblings area}}{\text{final area}} + \frac{\text{final area}}{\text{final area}} \right\} \right|$$

If both of these values were within a tolerance of 1% (found experimentally), we considered that track as a candidate track. Due to errors in the detection of cell divisions, and sudden movement of cells, some tracks, even though satisfying the above criteria, may not be representing the growth of a particular cell and the resulting tracks jumps from one cell to another. We therefore included an additional evaluation of the filtered tracks, as described below.

J. Evaluation of Track Quality

To analyze the growth of a cell, it is sufficient to analyze the increase in major axis length of the object, since the width of all the cells are nearly constant. Cells grow more or less exponentially over time, meaning that the major axis length of a cell is approximately a straight line in logarithmic scale. To further evaluate the correctness of a track, we evaluated the noise level of this line using RANSAC [23] line fitting. Next we found the mean of the absolute difference of the estimated line (in logarithmic scale) and the observed major axis values. All the tracks were sorted in increasing order of this mean error value, and the tracks that had errors less than a specific threshold were considered for further post-tracking segmentation correction as described below. This threshold was chosen based on the application and the acceptable error tolerance. In this experiment we set the value to 3 pixels.

K. Post-tracking Segmentation Correction

After the final quality control, some of the high-quality tracks may still contain single frame outliers where the segmentation algorithm has failed. We corrected for such segmentation errors using the assumption that the cells grow very little from one frame to the next frame. We considered a segment as erroneous if the absolute difference between the major axis length of the segment and the estimated major axis length by RANSAC fitting was above 10 pixels (found experimentally). To replace the erroneous segmentation result, we found the segment from the previous and next image. Then an average segment out of the two was calculated and placed in the image, as illustrated in figure 3. To find the average segment, we needed to find three things; 1) average position, 2) average orientation and 3) average shape. Average position was found as the mean position of the centroids of the two segments while average orientation was found by the average orientation of the major axes of the two segments. Then, to find the average shape, we first rotated both the segments such that their major axes were parallel to the x-axis and the centroids located at the middle of the image. Then signed distance transforms were performed on both these images. Finally, the average image was obtained as the average of the two signed distance transforms. This final averaged segment was placed in the erroneous frame replacing the erroneous segment.

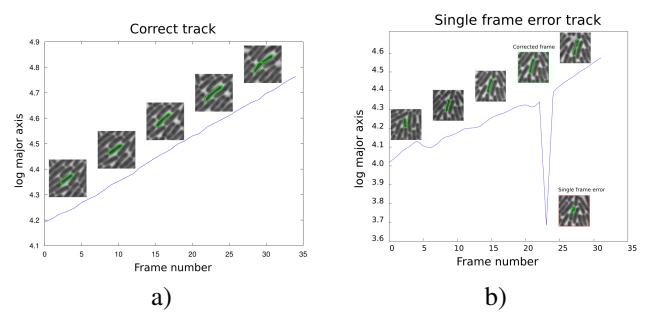


Fig. 3. a) track showing cell growth without any error b) track with single frame error and corresponding corrected frame.

III. RESULTS AND DISCUSSION

A. Segmentation Results

We compared the segmentation results of our Curvature Based Approach, referred to as CBA, with those of *MicrobeTracker* [5] and *MAMLE* [7]. As it has already been shown that *MAMLE* outperforms *Schnitzcells* [6] under many circumstances, we did not compare against this approach. We evaluated the segmentation performance on three *E. coli* phase contrast images: one from the *MicrobeTracker* example data, one from *Schnitzcells* (which was also used by the authors of *MAMLE*), and one from our own dataset. The *MicrobeTracker* and *Schnitzcells* examples are obtained for *E. coli* cells growing as largely isolated cells or micro-colonies on agarose pads whereas our dataset shows densely growing cells in PDMS microfluidics. It is important to note that neither *MicrobeTracker* nor *Schnitzcells* was developed to handle data of the type we have in our microfluidics experiments and that relative performance can vary depending on specific imaging conditions. We also evaluated the performance on one fluorescent image from our own dataset. Here, we inverted the fluorescence microscopy images and then processed them in the same way as the phase contrast images. We manually created ground truth for all the images. All raw images and their corresponding ground truths are shown in the two top rows of figure 4. Our proposed method outputs the segmentation result as a binary image while the *MicrobeTracker* output is in the form of boundary coordinates. Therefore, special care was needed when performing the evaluation. The contours of two nearby cells detected by *MicrobeTracker* overlapped in some cases, and to avoid considering the overlapping regions as new segments we evaluated one contour at a time, converting each contour to a binary image by dilation followed by hole filling and erosion.

We also wanted to evaluate the improvement in performance of CBA with and without intensity filtering and contour smoothing, resulting in a total of five algorithms; 1) CBA, 2) CBA including intensity based filtering, 3) CBA including intensity filtering and smoothing, 4) *MicrobeTracker*, and 5) *MAMLE*. figure 4 shows output from each of the approaches, and figure 5 shows zoomed-in versions of the same images.

Evaluation results are shown in figure 7. The figure 6 shows that the area distributions are maintained prior to evaluation of segmentation results as discussed in the methods section. To summarize the comparison of the segmentation algorithms, we tested a range of thresholds on the F-score to quantify what proportion of the cells was correctly identified at different requirements on achieved F-score. We finally set a threshold at 0.8 for the F-score, representing a decent bacterial outline, and found the percentage of bacterial cells that have an F-score greater than 0.8 for each tested algorithm. The results show that using CBA, 87% of the cells fall above this threshold, as compared to 48% of the cells when using *MicrobeTracker*, and 83% when using *MAMLE* on the phase contrast image from our dataset(phase data 1). For our fluorescence microscopy image (fluorescent data), the value is 87% for CBA while it is 46% for *MicrobeTracker* and 76% for *MAMLE*. For the *MicrobeTracker* phase contrast image (phase data 1), the

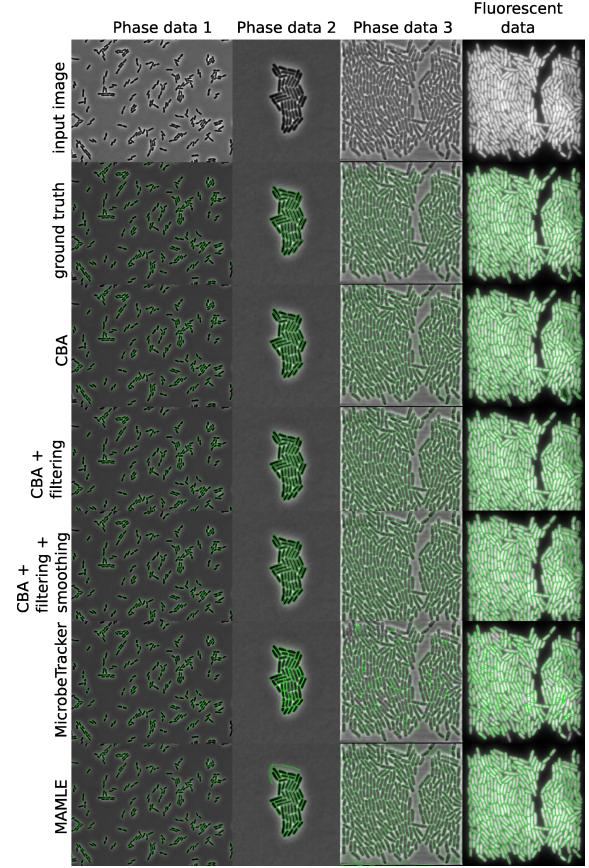


Fig. 4. phase data 1 is from the *MicrobeTracker* dataset, phase data 2 from *Schnitzcells* dataset, phase data 3 and fluorescent data from our dataset. Phase data 1 and 2 are from experiments on agarose pads whereas phase data 3 and fluorescent data are from a growth medium in a PDMS device.

values are 91% for CBA while 82% for *MicrobeTracker* and 90% for *MAMLE*. For the *Schnitzcells* phase contrast image (phase data 2), the values are 78% for CBA, 46% for *MicrobeTracker* and 85% for *MAMLE*.

The performance of CBA was only marginally improved when adding intensity based filtering and contour smoothing. However, the contour smoothing may be necessary for applications where exact edge positioning is crucial, e.g., when relating the position of sub-cellular signals to the cell edge.

B. Run Time Evaluation

Our algorithm, CBA, was implemented in Python as a multithreaded program. We compared the execution speed of the methods on the four datasets. The experimental results show that the execution speed depends on the number of cells present in the image. It was found that our method, CBA, is an order of magnitude faster than the state-of-the-art methods. The results show that CBA is 10 times faster than *MicrobeTracker* and 8 times faster than *MAMLE* on average. The time taken by each method on the four datasets is shown in figure 8. We performed the speed evaluation on a laptop with a 2.7GHz i7 processor and 16GB RAM running on Windows 7.

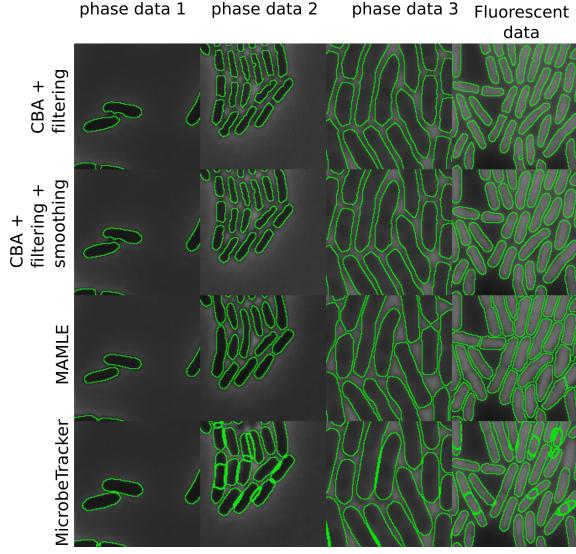


Fig. 5. Zoomed in versions of selected regions from figure 4

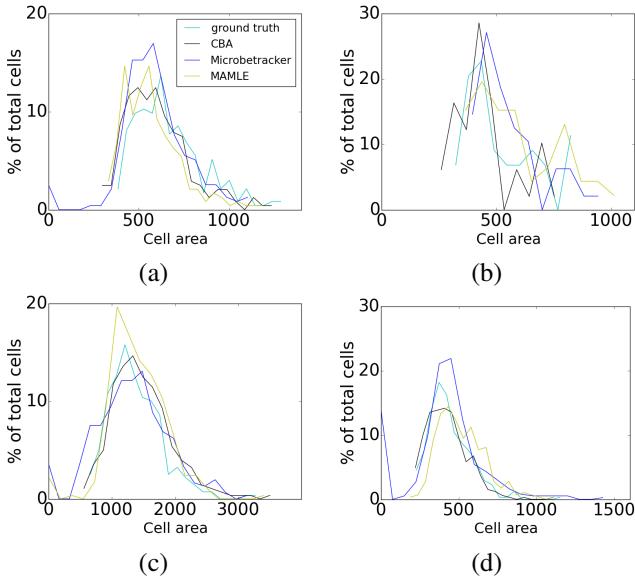


Fig. 6. Area distribution of segmented cells for (a) phase data 1, (b) phase data 2 (c) phase data 3 and (d) fluorescent data

C. Tracking Results

To further evaluate the usability of the segmentation results produced by our proposed method, we tested its performance on a large time-lapse experiment. We applied our segmentation algorithm to time-lapse images from a dataset comprising 500 images with approximately 250 cells per image. 500 frames correspond to 250 min in real time. Cell tracking was done using the Viterbi algorithm as described in the Methods section, and we obtain a total of 10259 tracks. The tracks were then subjected to filtering to extract candidate tracks. The track analysis stage found 1529 tracks satisfying the criteria of having cell division at the beginning and also at the end. Since we consider cell division at the start and end of the track, essentially all tracks including the first or last 30 min are discarded. Many tracks towards the beginning and the end

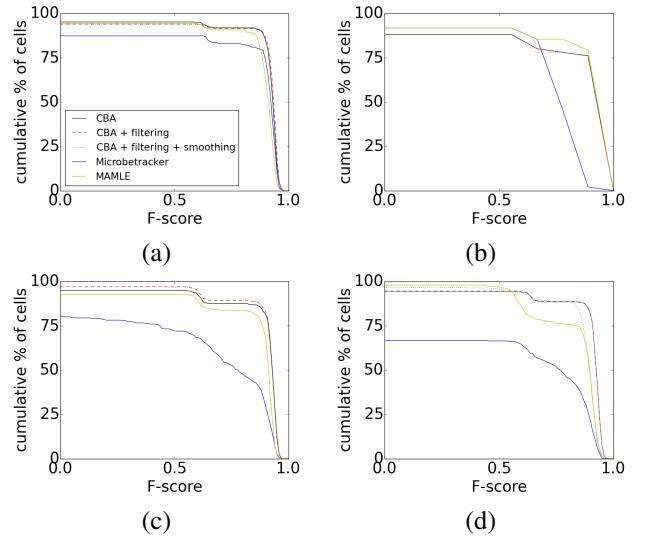


Fig. 7. Cumulative % of cells above the particular F-score for (a) phase data 1, (b) phase data 2 (c) phase data 3 and (d) fluorescent data

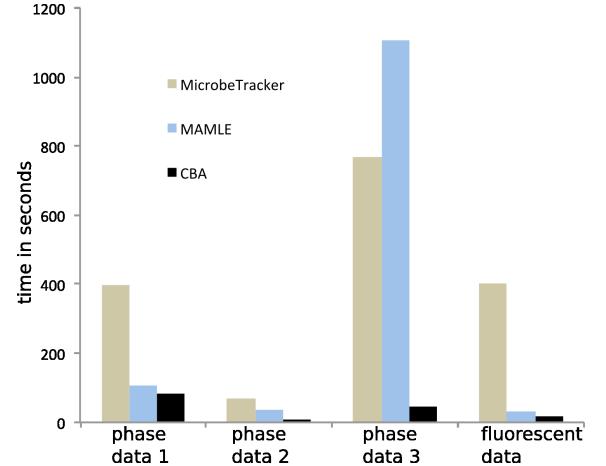


Fig. 8. Execution times of *MicrobeTracker*, *MAMLE* and *CBA* on four datasets.

of the sequence were excluded from further analysis. All the 1529 tracks were sorted in increasing order based on RANSAC fit error, as shown by the solid dashed line in figure 9 (note the logarithmic scale on the fit error). We further searched for and corrected segmentation errors in the tracks. The solid line shows the improved result, where a total of 1238 tracks fell below the 3 pixel error threshold prior to track correction, and 1274 fell below the threshold afterwards. In frames 100 to 400 40% of the identified cells are assigned to a track.

D. Comparison of Growth Rate at Different Treatments

The second experiment included in this work was to find the growth rate of *E. coli* cells in different media. The first dataset was obtained from *E. coli* cells using glycerol as medium and the second dataset was obtained using glucose as medium. We wanted to know how fast the cells grow and divide in these two media. We found that the slope of the RANSAC fitted line was 0.0120 ± 0.0021 pixels/frame in logarithmic scale (mean \pm standard deviation) for cells in glucose (red) and $0.0092 \pm$

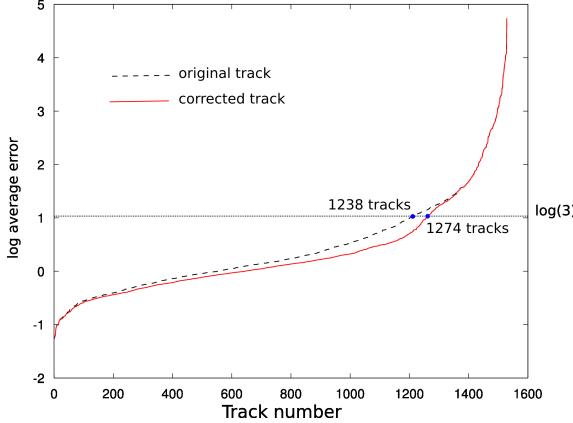


Fig. 9. RANSAC fit error of the major axis length in cell tracks, for the original segmentation and the error corrected segmentation, shown in logarithmic scale.

0.0015 pixels/frame in logarithmic scale for cells in glycerol (blue). There are some cells that grow faster in glycerol and cells that grow slower in glucose as can be seen in figure 10.

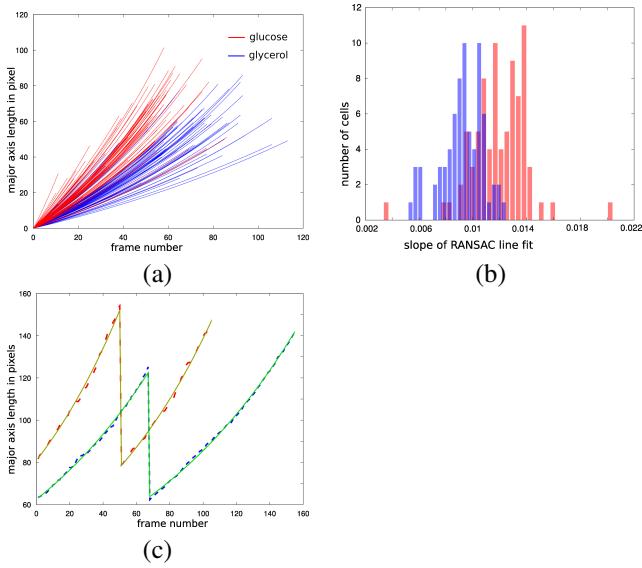


Fig. 10. (a) Change in major axis length over time for two datasets with cells in glucose (red) and glycerol (blue) (b) Histogram of slope found by RANSAC line fit for two datasets. (c) A sample lineage of cells grown in glucose(raw data in red and RANSAC in yellow) and glycerol(raw data in blue and RANSAC in green)

IV. CONCLUSIONS AND FUTURE WORK

In this work, we developed a fast and robust *E. coli* cell segmentation method that can be used as input for tracking cells in time-lapse microscopy images. We also showed that our method works better than the state-of-the-art methods both in terms of accuracy and speed. The parameters required to tune for different datasets are few and depend only on the size of cells present in the input image. To show the usability of our method on actual biological datasets, we did segmentation and tracking on two different datasets and found the growth rates of the cells and their variability. This clearly shows that

the proposed method has the potential to be used for a wide variety of experiments involving *E. coli* cells. In the future, we would like to speed up the processing further approaching the speed of the data acquisition.

ACKNOWLEDGMENT

The authors would like to thank the financial support from the Swedish Research council, grant 2012-4968 to Carolina Wählby.

REFERENCES

- [1] M. B. Elowitz, A. J. Levine, E. D. Siggia, and P. S. Swain, "Stochastic gene expression in a single cell," *Science*, vol. 297, pp. 1183–1186, 2002.
- [2] Y. Taniguchi, P. J. Choi, G. Li, H. Chen, M. Babu, J. Hearn, A. Emili, and X. S. Xie, "Quantifying *e. coli* proteome and transcriptome with single-molecule sensitivity in single cells," *Science*, vol. 329, pp. 533–538, 2010.
- [3] Q. Wu, F. Merchant, and K. Castleman, *Microscope Image Processing*. Massachusetts: Academic Press, 2008.
- [4] M. Wallden and J. Elf, "Studying transcriptional interaction in single cells at sufficient resolution," *Current Opinion in Biotechnology*, vol. 22, pp. 81–86, 2011.
- [5] O. Sliusarenko, J. Heinritz, T. Emonet, and C. Jacobs-Wagner, "High-throughput, subpixel precision analysis of bacterial morphogenesis and intracellular spatio-temporal dynamics," *Mol. Microbiol*, vol. 80, pp. 612–627, 2011.
- [6] J. W. Young, J. C. W. Locke, N. R. A. Altinok, T. Bacarian, P. S. Swain, E. Mjolsness, and M. B. Elowitz, "Measuring single-cell gene expression dynamics in bacteria using fluorescence time-lapse microscopy," *Nat. Protocols*, vol. 1, pp. 80–88, 2012.
- [7] S. Chowdhury, M. Kandhavelu, O. Yli-Harja, and A. S. Ribeiro, "Cell segmentation by multi-resolution analysis and maximum likelihood estimation (mamle)," *BMC bioinformatics*, vol. 14, p. S8, 2013.
- [8] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Sys. Man. Cyber.*, vol. 9, pp. 62–66, 1979.
- [9] C. Zimmer, E. Labruyere, V. Meas-Yedid, N. Guillen, and J.-C. Olivo-Marin, "Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: a tool for cell-based drug testing," *IEEE Trans. Medical Imaging*, vol. 21, p. 12121221, 2002.
- [10] O. Dzyubachyk, W. A. van Cappellen, J. Essers, W. J. Niessen, and E. Meijering, "Advanced level-set-based cell tracking in time-lapse fluorescence microscopy," *IEEE Trans. Medical Imaging*, vol. 29, pp. 852–867, 2010.
- [11] R. Bise, Z. Yin, and T. Kanade, "Reliable cell tracking by global data association," in *Proceedings of IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE, 2011, pp. 1004–1010.
- [12] F. Jug, T. Pietzsch, D. Kainmller, J. Funke, M. Kaiser, E. van Nimwegen, C. Rother, and G. Myers, "Optimal joint segmentation and tracking of escherichia coli in the mother machine," in *Proceedings of Bayesian and grAphical Models for Biomedical Imaging*, ser. Lecture Notes in Computer Science, T. A. D. P. A. R. a. M. Jorge Cardoso, Ivor Simpson, Ed., vol. 8677. Berlin: Springer, 2014, pp. 25–36.
- [13] K. Magnusson, J. Jalden, P. Gilbert, and H. Blau, "Global linking of cell tracks using the viterbi algorithm," *IEEE Trans. Med. Imag.*, 2014.
- [14] G. Ullman, M. Wallden, E. G. Marklund, A. Mahmudovic, I. Razinkov, and J. Elf, "High-throughput gene expression analysis at the level of single proteins using a microfluidic turbidostat and automated cell tracking," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 368, pp. 1–8, 2012.
- [15] P. Hammar, P. Leroy, A. Mahmudovic, E. Marklund, O. G. Berg, and J. Elf, "The lac repressor displays facilitated diffusion in living cells," *Science*, vol. 336, pp. 1595–1598, 2012.
- [16] T. J. Willmore, *An introduction to differential geometry*. Oxford: Oxford University Press, 1959.
- [17] J. A. Thorpe, *Elementary topics in differential geometry*. New York: Springer-Verlag, 1979.
- [18] C. Woodford and C. Philips, *Numerical methods with worked examples: Matlab edition*. New York: Springer-Verlag, 2012.
- [19] W. Burger and M. J. Burge, *Principles of digital image processing: Core algorithms*. New York: Springer-Verlag, 2009.

- [20] E. Jones, E. Oliphant, and e. a. P. Peterson. (2001) Scipy: Open source scientific tools for python. [Online]. Available: <http://www.scipy.org/>
- [21] W. Eric. (2014) Full width at half maximum. [Online]. Available: <http://mathworld.wolfram.com/FullWidthatHalfMaximum.html>
- [22] P. Marquez-Neila, L. Baumela, and L. Alvarez, “A morphological approach to curvature-based evolution of curves and surfaces,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, pp. 2–17, 2014.
- [23] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Comm. of the ACM*, vol. 24, pp. 381–395, 1981.