RESEARCH ARTICLE

A gradient-based, GPU-accelerated, high-precision contour-segmentation algorithm with application to cell membrane fluctuation spectroscopy

Michael Mell[1]*, **Francisco Monroy**[1,2]*

1 Mechanics of Biological Membranes and Biorheology, Dpto. Química Física I, Universidad Complutense, Madrid, Spain, **2** Translational Biophysics, Instituto de Investigacion Biomédica Hospital Doce de Octubre (i +12), Madrid, Spain

* michael.mell@gmx.net(MM); monroy@ucm.es(FM)

OPEN ACCESS

Citation: Mell M, Monroy F (2018) A gradient-based, GPU-accelerated, high-precision contour-segmentation algorithm with application to cell membrane fluctuation spectroscopy. PLoS ONE 13 (12): e0207376. https://doi.org/10.1371/journal.pone.0207376

Editor: Jorge Bernardino de la Serna, Science and Technology Facilities Council, UNITED KINGDOM

Received: April 16, 2018

Accepted: October 30, 2018

Published: December 6, 2018

Copyright: © 2018 Mell, Monroy. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: This work was supported by the Ministry of Economy and Competitiveness (MINECO), Spain (Grant No. FIS2015-70339-C2-1-R) and Comunidad de Madrid, Spain (Grant No. S2013/MIT-2807). MM thanks the FPU program (MEC) for financial support.

Competing interests: The authors have declared that no competing interests exist.

Abstract

We present a novel intensity-gradient based algorithm specifically designed for nanometer-segmentation of cell membrane contours obtained with high-resolution optical microscopy combined with high-velocity digital imaging. The algorithm relies on the image oversampling performance and computational power of graphical processing units (GPUs). Both, synthetic and experimental data are used to quantify the sub-pixel precision of the algorithm, whose analytic performance results comparatively higher than in previous methods. Results from the synthetic data indicate that the spatial precision of the presented algorithm is only limited by the signal-to-noise ratio (SNR) of the contour image. We emphasize on the application of the new algorithm to membrane fluctuations (flickering) in eukaryotic cells, bacteria and giant vesicle models. The method shows promising applicability in several fields of cellular biology and medical imaging for nanometer-precise boundary-determination and mechanical fingerprinting of cellular membranes in optical microscopy images. Our implementation of this high-precision flicker spectroscopy contour tracking algorithm (HiPFSTA) is provided as open-source at www.github.com/michaelmell/hipfsta.

Introduction

Quantitative imaging is progressively empowering the analytic toolbox of cellular biology with high-performance observational facilities accessing new biophysical markers resolved in space and time [1,2]. Optical microscopy imaging combined with computational technologies are opening translational promise as provide the integrative biologist with unprecedented mines of data to be exploited in high-performance analytic approaches, which are allowing to gain further insight beyond conventional imaging with immunofluorescence [3]. Fluctuation spectroscopy with optical contrast microscopes is a simple, yet powerful biophysical method to probe non-invasively mechanical properties of biological cells by imaging the shape fluctuations of the cell membrane contour [4]. When applied to living cells undergoing metabolically-driven (non-equilibrium) fluctuations, the fluctuation method has allowed for getting new insights into cell mechanics, particularly in studies with red blood cells, hereinafter RBCs [5–12]. The performance of contour-segmentation algorithms for tracking membrane fluctuations from video-microscopy images has always played a crucial role in the advance of this technique by enabling a more precise determination of the cell contour-position [13]. Accordingly, various segmentation methods have been proposed with contrast imaging [7,13–21], which were accompanied by technological advances in digital image processing and computational power. The contour-segmentation algorithm we present here harnesses the computational power of general purpose graphics processing units (GPGPU) having recently become available to significantly improve on current segmentation methods in the sub-pixel performance of nanometer resolution. The super-localization software here proposed may enable observation of novel dynamical phenomena in cell biology and may allow for significantly more precise results that could be exploited efficiently in cell phenotyping approaches useful in biomedical contexts. The new algorithm is potentially implementable to more sophisticated optical settings [1], including diffraction phase microscopy (DMP) [22] and phase-contrast modes in digital holographic microscopy (DHM) [23]. These adaptations could push quantitative imaging further into novel software developments that take advantage of the enhanced contour detection provided by our new method.

Technically speaking, the new segmentation algorithm analyzes optically contrasted (either bright-field or phase-contrast) microscopy images of cells possessing a characteristic halo at the cell boundary. The so-called “halo effect” is nonlinearly tied to the coherence of the illumination and the optical contrast of the sample [24], and provides an exploitable framework for image formation in different contrast modes under varying degrees of coherence [25,26]. This contrast halo consists of an intensity minimum and maximum, which encodes the information on the membrane position as the result of a mismatch in the refractive indices of the medium inside and outside the cell (see Fig 1A and 1B). Early segmentation algorithms used the intensity maximum of the halo to determine the location of the cell-contour [14,16,19,27]. However, it was later pointed out by Döbereiner [18] and Pécréaux [13] that the location of the interface is actually placed at the maximal gradient of the halo intensity [28] and this has become the preferred method for the membrane localization [13,16–18,20,21]. Pécréaux and cols. [13] proposed the currently widespread segmentation algorithm, where the interface location is defined by the intercept of a linear fit to the intensity profile at the contour halo and the average background intensity (see Fig 1B). This linear fit is performed to the pixel-intensity values along three of the four principle directions of the pixel-grid (2x parallel, 2x diagonal) to calculate the final position from their weighted average [13]. In the Pécréaux’ algorithm segmentation is performed sequentially along the contour using a number of different conditions to determine the segmentation direction. More recently, new segmentation methods have been proposed to locate the contour positions by directly determining the maximum of the intensity gradient [20,21]. In particular, Usenik et al. [21] used interpolation to calculate the image intensity along radial directions starting from the center of the contour. The contour position is determined by the intensity gradient maximum along each individual line (see Fig 1(C)). By calculating an updated barycenter of the contour, the process is repeated for each image until the center of the contour converges to a fixed position. This allows the individual refinement of each contour-coordinate, an improvement that is not possible using previous methods. However, the direct usage of the numerical gradient in the Usenik’s method has drawbacks, since it amplifies image noise. Therefore, those authors use Gaussian image filter to reduce noise, which should be expected to reduce resolution [21].

To overcome the limitations of these two algorithms, our proposed method combines ideas from both, putting substantial emphasis on image oversampling as an original characteristic of

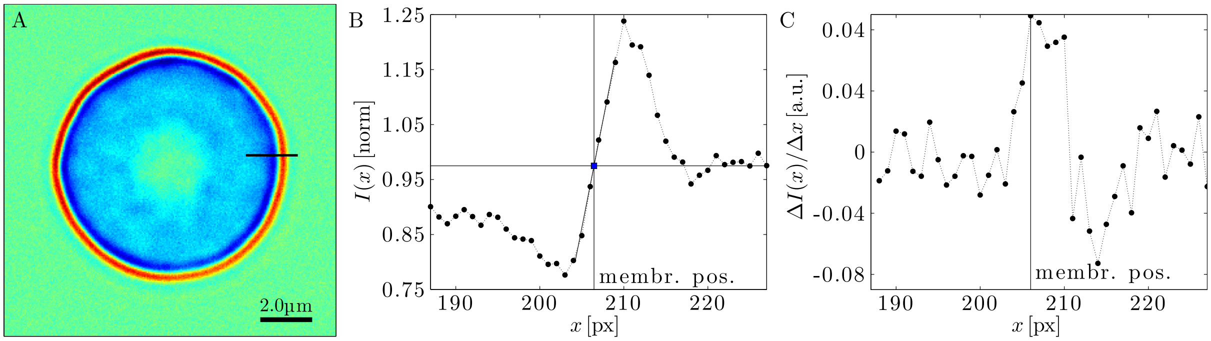

Fig 1. Phase contrast causes an intensity gradient at the location of the RBC membrane. (A) Bright-field image of a RBC. (B) Intensity profile along the black line in (A). The vertical line indicates the contour-position as found with the method described in [13]. The background intensity and fit are also indicated (horizontal and diagonal line). (C) Intensity-gradient of (B). The vertical line indicates the maximum. The noise in the image cause two maxima at the peak, which makes localizing the absolute peak value difficult without filtering.

https://doi.org/10.1371/journal.pone.0207376.g001

the proposed method. In Pécréaux' method each point in the contour is determined from the precedent one, which introduces an evident bias that makes that method especially sensible to the choice of the initial condition. To increase the robustness of our algorithm, we follow the Usenik's method for which each contour coordinate $\boldsymbol{P}_i$ is refined independently at a fixed contour angle ϕ_i over various iterations per image. Motivated by Pécréaux, our method determines each contour coordinate $\boldsymbol{P}_i$ from the weighted sum of a large number of linear fits over a range of local angles φ_j centered on the contour normal $\boldsymbol{n}_i$ of the intensity halo. In our proposed method, the positions for performing the linear fits become independent of the pixel-grid by interpolating pixel-intensities, which overcomes a subtle, but significant issue with Pécréaux' method that is related with the fixed positions of the centers of the pixels. These mutual improvements have allowed us to generate a new hybrid algorithm, more robust and precise than the previous ones, which enhances the accuracy of the segmentation method down to 2nm spatial precision in determining changes in the position of the contour halo. The new algorithm has been implemented with movies of dynamic fluctuations recorded by digital video-microscopy with high-velocity cameras at short exposition times (shorter than millisecond). This allows an ultrafast readout of the contour fluctuations (up to several tents kHz) that, applied to cell membranes, allows imaging the instantaneous snapshots necessary for probing the shape fluctuations of the cell membrane. A typical imaging session left less than one-second, the time necessary to record all the data required to perform the time-average involved in a statistically relevant fluctuation spectrum and to detect the time correlations due to the intrinsic dynamics of the fluctuation modes of membrane deformation. Further mathematical treatment of these membrane fluctuations, especially by Fourier analysis, makes possible a deep understanding of membrane mechanics in terms of usual elasticity models [29]. All these improvements empower the proposed method with a significantly higher performance than previous ones, particularly in terms of enhanced spatiotemporal accuracy that allows resolving the membrane fluctuations with a higher precision not only in amplitude but also in spatial and time resolution, both in real and reciprocal domains. Such a breakthrough in segmentation performance should be crucial to approach new problems of cell mechanics where membrane fluctuations can be exploited as a relevant observable. Imaging membrane fluctuations could thus become an excellent observational method for non-invasive and non-stressing probing the mechanical phenotypic traits of cellular membranes where the intrinsic rigidity of the cytoskeleton necessarily imposes low amplitudes and short correlation times and distances of the membrane fluctuations.

Method

Microscopy hardware

To observe the surrounding border of a cell, or of a lipid vesicle, we take advantage of wide-field transillumination video-microscopy [30]. Transillumination techniques require two lenses, condenser and objective, both used to focus the light on each side of the sample. In an inverted bright-field (BF) microscope, the sample is transilluminated from above by focusing light (with the condenser) on the entire field of view where the object to observe is placed. The resulting BF image is formed after the light has been transmitted through the sample and collected with the objective lens from its focal plane. Then, the image is recorded on a video camera. Because of the transmittance difference between the object interior and the outer medium, an Airy pattern due to edge-diffraction is produced at the border of the object. Consequently, when the BF image is formed, the object appears surrounded by a brighter "halo", which delimits the contour of the border [31]. BF images of our biological samples tend to have low contrast, because vesicles and most cells are not strongly light-absorbing, however additional contrast can be gained in the phase-contrast mode [30]. In a Phase Contrast (PhC) microscope, phase shifts produced in light passing through a transparent specimen with contrasted optical densities are converted to brightness changes in the image. PhC is based on refractive index differences [32], indeed optical edges between regions with different refractive indices cause light to refract in an amount (phase-shift) that depends on how much the refractive index changes. Most simply, PhC microscopes generate an image by comparing how much phase-shift is produced at each location in the sample relative to how much light is not phase-shifted. Physically, this construction occurs via interference between refracted (phase-shifted) and non-refracted light beams. The PhC microscope is fitted with a special condenser and a phase-contrast objective, which contain both phase rings to control phase contrast. PhC images may be made to appear dark against a bright background (positive contrast) or bright against a dark background (negative contrast). The borders of images are surrounded by a characteristic phase-contrast "halo". Both, BF and PhC techniques are highly sensitive and compatible with the short exposure times involved with high frame rate recording.

The proposed contour-segmentation algorithm has been conceived to work with halo-edged contrasted microscopy images, indeed it can be interoperable with images obtainable from different transillumination modes used to generate optical contrast. Because the halo contrast is tied to the coherence of the illumination, it can be controlled by the aperture of the condenser. Here, we use indifferently both, BF and PhC images, although images from other contrast modes, such as dark-field, differential interference contrast (DIC) and cross-polarized microscopy are intrinsically different to typical halo-contrasted images, they could be also analyzed by introducing changes in the parameters that determine the definition of the contrast profile in the proposed algorithm.

In this work, microscopy images are obtained with an inverted microscopy (Nikon Eclipse 80i), which is equipped with high-performance objectives with all possible aberrations, included chromatic, sufficiently minimized. The BF mode is implemented at wide-field illumination with white light from a mercury lamp, and an apochromatic bright-field objective (x100, 1.45 N.A.). The PhC mode is implemented with a condenser equipped with an annular phase plate that focusses the illuminating light from a cold-light LED source (150mW max) in the sample, and an apochromatic phase-contrast objective (x100, 1.45 N.A.). The optical system is equipped with auto-focus, which allows for automatic correction of eventual image defocusing once the equatorial-plane of the contour object has been defined. The microscopy images are continuously recorded with high-dynamical range digital camera (Photron FASTCAM SA-3). This camera has a CMOS sensor allowing for very high sample rates through the parallel and independent read-out of each individual pixel. Each pixel possesses separated electronics for converting the accumulated charge in a pixel into a digital signal. As a result, each pixel possesses different electronic characteristics regarding its read-out noise and dark-current. The whole setup is mechanically isolated by placing on top of an anti-vibration table (Integrity 1VCS, Newport).

Software: Algorithm description

Image preprocessing. Because a CMOS high-velocity camera with an in-parallel pixel-grid readout is used, the contrast images possess embedded a pixel-grid structure, which if not corrected, can affect the segmentation results. Furthermore, aberrations in the optical path can lead to artifacts in the image that can affect the segmentation result. To correct both artifacts we perform a background-correction of each movie frame before segmentation. For this we record and average over a large number of dark-field images D_d as well as a large number of empty ("flat") frames F_f (usually ~1000 frames each). The dark-field images are recorded once for the given exposure-time and serve to correct the cameras dark-current. The flat-fields are recorded previous to and for each measurement (each movie), since their image intensity needs to match that of the measurement as close as possible. The corrected frame I_i^{final} is then obtained from I_i by performing the following operation for each pixel:

$$I_i^{\text{final}} = \frac{I_i - \langle D_d \rangle_d}{\langle F_f - \langle D_d \rangle_d \rangle_f}$$

We perform the averaging over $\langle D_d \rangle_d$ and $\langle F_f \rangle_f$ in order to minimize any noise introduced when performing the correction above.

Contour determination. **Coordinates.** The segmentation algorithm uses both Cartesian $\boldsymbol{P}_i$ and polar (ϕ_i, R_i) coordinates relative to the contour center $\boldsymbol{O}$ to represent the $i = 0...N - 1$ contour coordinates (see Table 1 for parameter values) that define the N contour sites. Additionally, it uses the contour normal $\boldsymbol{n}_i$ which is calculated from the halo gradient at every contour site i. To perform its work, the algorithm uses the $\boldsymbol{P}'_i$, $\boldsymbol{n}'_i$, and $\boldsymbol{O}'$ of the previous iteration (indicated by the prime) and uses them as the starting points to determine the $\boldsymbol{P}_i$, $\boldsymbol{n}_i$ and $\boldsymbol{O}$ in the following iteration. This allows the positions contour sites i to be refined independently of each other in an iterative manner, as we will describe below. Initial segmentation of the first image is performed sequentially, since no previous coordinates $\boldsymbol{P}'_i$ are available. It is initiated by providing an approximate starting point $\boldsymbol{P}'_{\boldsymbol{i}=\boldsymbol{0}}$ on the halo and an approximate contour center $\boldsymbol{O}'$. The point $\boldsymbol{P}'_{\boldsymbol{i}=\boldsymbol{0}}$ is then refined once to obtain the refined coordinate $\boldsymbol{P}_{\boldsymbol{i}=\boldsymbol{0}}$ or $(\phi_{i=0}, R_{i=0})$. The radius R_i is then used as the starting point $(\phi_{\mathrm{i+1}}, R_i)$ to calculate the following coordinate $\boldsymbol{P}'_{\boldsymbol{i}=\boldsymbol{1}}$, which then is again refined to obtain $\boldsymbol{P}_{\boldsymbol{i}=\boldsymbol{1}}$. This is then continued for all N contour angles, which are fixed to the values $\phi_i = (2\pi/N) \times i$. After initial segmentation is completed, the algorithm writes the obtained coordinate set $\boldsymbol{P}_{\boldsymbol{i}}$ and $\boldsymbol{O}$ to $\boldsymbol{P}'_i$ and $\boldsymbol{O}'$ to use them as the input for the following segmentation iteration. From then on, all contour sites i are

Table 1. Parameter values used for imaging and tracking RBCs.

N	Φ	M	f_{interp}	n_{fit}
2048	π/4	64	20 px^{-1}	100
n_{bkgr}	L_Y	Δ_{shift}	Δ_{coord}	Δ_{center}
400	400	10 px	0.01 px	0.01 px

Parameter values used for imaging and tracking RBCs to obtain the results presented in this paper.

https://doi.org/10.1371/journal.pone.0207376.t001 determined independently of each other by using the $\boldsymbol{P}'_i$, $\boldsymbol{n}'_i$ and $\boldsymbol{O}'$ of the previous iteration, even when switching from on video image to the next.

Pixel grid. The pixel size determines the maximal resolution for sampling image intensities from phase contrast halo. Without interpolation of image pixels, the processing of image intensities by an algorithm will therefore be confined to the location of pixels within the pixel grid. In particular, fitting of the halos intensity is then confined to the center position of individual pixels and can only be performed parallel and diagonal to the pixel-grid. However, for any closed contour the maximal gradient of the halo will point into every possible angular direction of the image, including direction not commensurate with those of the pixel grid. Without interpolation, the rectangular geometry of the pixel grid therefore represents a significant limitation to algorithms that are based on intensity gradients or the fitting of pixel intensities. Pécréaux and cols. handled this limitation by calculating *local* contour coordinates along three of the four pixel-grid directions, while excluding the direction closest to the contours tangent (see Fig 7 in ref. [13]). Each *local* contour coordinate is obtained by fitting a line to the halo intensity and calculating its intercept point with the background intensity at the outside contour region, which is calculated as an averaged value along the radial direction. To minimize mapping errors due to optical heterogeneities, only single cells that remain isolated from each other are considered. The final contour coordinate at a given pixel position is then defined by the weighted average of these *local* contour coordinate, where each direction is weighted by the corresponding slope of the intensity gradient in that direction. Consequently, the direction that is closest to the contour normal will be counted most since in that direction the slope will be the steepest. In our algorithm, we incorporate the concept of using weighted averages of different directions but use a more sophisticated sampling technique to find the direction and amplitude of the maximal normal gradient.

Image sampling and gradient fits. By interpolating and resampling the pixel intensities of the image during segmentation, we are neither limited to the discrete pixel positions nor the diagonal and parallel directions of the pixel grid. Instead, using the contour coordinate $\boldsymbol{P}'_i$ from the previous segmentation iteration as the base-point, we determine a local contour positions $\boldsymbol{p_j}$ for a range of angles φ_j ($j = 0...M - 1$) (see Fig 2A) by fitting a line to the halo intensity and calculating its intercept with the background intensity similarly to Pécréaux' method. The angle range Φ of φ_j is centered on the contour normal $\boldsymbol{n'_j}$ of the previous iteration (see Eq (2)). The intensity values that are used for the calculations are obtained by interpolating the pixel intensities along a line in each direction φ_j.

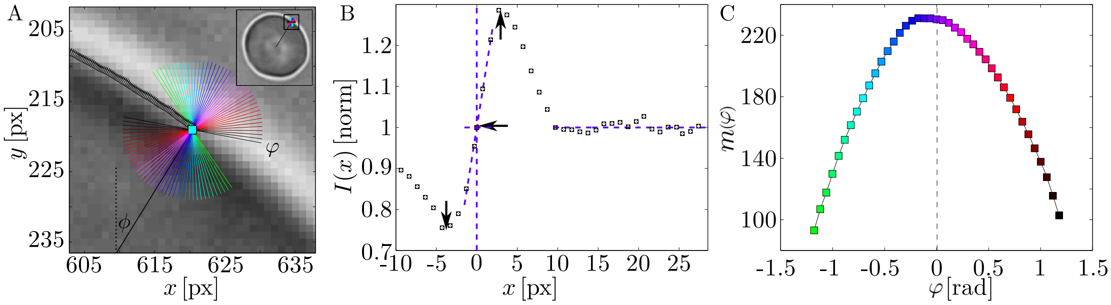

Fig 2. The new algorithm uses oversampling of the intensity gradient to determine the membrane position with sub-pixel precision. (A) Illustration of the local angles φ_j and their position on the halo. (B) Determination of local coordinate $\boldsymbol{p_j}$ in a given direction φ_j. The fit is performed between the intensity minimum and maximum indicated by the arrows. This corresponds to **Method 1** explained in the text. Note that the fit position (horizontal arrow) lies in-between the positions and intensity values of the individual pixels (black squares). This is made possible by the linear interpolation. The background intensity is determined just outside the halo. (C) Fit slopes as a function of the local angle. The maximum is perpendicular to the halo.

https://doi.org/10.1371/journal.pone.0207376.g002

The gradient fit in the direction φ_j is determined by performing a linear fit to the intensity using the equation $y_j = m_j\, x + b_j$. By using an oversampling factor f_{interp} (see Table 1), we are able to finely adjust the location along the intensity profile at which the linear fit is performed. As we will see in the analysis section, this is important to avoid pixel artifacts. The background intensity $\langle I_j \rangle$ is calculated from a line in the direction of φ_j just outside the halo of the phase contrast image by averaging over n_{bkgr} interpolated intensity values (see Fig 2B). We determine the background intensity in this manner and not centered about $\boldsymbol{P}_i'$ (like Pécréaux does), because many cells, including RBCs, cause substantial light absorption and other image inhomogeneities, so that the light intensities inside are modified.

Contour vector-coordinates. The local contour positions $\boldsymbol{p_j}$ in the direction of each angle φ_j are then given by the intercept of the fitted straight with the background intensity

$$x_j = \frac{\langle I_j \rangle - b_j}{m_j}$$

using the expressions

$$\boldsymbol{p}_j = \boldsymbol{P}_i' + \boldsymbol{x}_j \boldsymbol{e}_j,$$

where $\boldsymbol{e}_j$ is the unit vector in the direction of the local angle φ_j. Finally, the updated contour coordinate $\boldsymbol{P}_i$ in the direction of ϕ_i is determined by calculating the weighted mean of the $\boldsymbol{p}_j$ using their corresponding inclines m_j as weights:

$$\boldsymbol{P}_i = \frac{\sum_j \boldsymbol{p}_j m_j}{\sum_j m_j} \tag{1}$$

The updated contour normal $\boldsymbol{n}_i$ is calculated similarly through the weighted mean of the unit vectors $\boldsymbol{e}_j$ weighted by their corresponding gradient slopes m_j (indicated as black arrows in Fig 2A):

$$\boldsymbol{n}_i = \frac{\sum_j \boldsymbol{e}_j m_j}{|\sum_j \boldsymbol{e}_j m_j|} \tag{2}$$

Since we perform this operation during various iterations for each video image, the algorithm iteratively finds the optimal contour position $\boldsymbol{P}_i$ and normal $\boldsymbol{n}_i$, about which the final directions φ_j, at which we sample, are distributed symmetrically.

Coordinate sanity check. After each segmentation iteration, the algorithm checks the contour coordinates $\boldsymbol{P}_i$ for any spurious outliers that could have been tracked incorrectly. This is important as the contour center in each iteration is determined as the weighted sum of $\boldsymbol{P}_i$, which is weighted by the distance ds_i between them (see below). If a contour coordinate $\boldsymbol{P}_i$ is a strong outlier and far from its neighboring coordinates $\boldsymbol{P}_{i-1}$ and $\boldsymbol{P}_{i+1}$, this can cause the center to jump and potentially lead to the failure of the algorithm. To avoid this, each coordinate $\boldsymbol{P}_i$ from the current iteration is compared to its value of the previous iteration $\boldsymbol{P'}_i$. If the distance between the two is larger than the tolerance Δ_{shift}, then the value of $\boldsymbol{P}_i$ and the contour normal $\boldsymbol{n}_i$ is replaced with the linear interpolation of its closest neighboring coordinates that were tracked correctly, so that:

$$\text{if}: \quad |\boldsymbol{P}_i - \boldsymbol{P}_i'| > \Delta_{\mathrm{shift}}$$

$$\text{then set}: \quad \boldsymbol{P}_i = \frac{\boldsymbol{P}_{\boldsymbol{i-\alpha}} - \boldsymbol{P}_{\boldsymbol{i+\beta}}}{\alpha + \beta}$$

and calculate the normal vector as:

$$\boldsymbol{n}_i = \frac{\boldsymbol{n}_{i-\alpha} + \boldsymbol{n}_{i+\beta}}{\boldsymbol{\alpha} + \boldsymbol{\beta}} \Big/ \left| \frac{\boldsymbol{n}_{i-\alpha} + \boldsymbol{n}_{i+\beta}}{\boldsymbol{\alpha} + \boldsymbol{\beta}} \right|$$

Here $i - \alpha$ and $i + \beta$ are the indices of the closest neighboring coordinates that were tracked correctly. Note, that the value of $\Delta_{\text{coord-shift}}$ needs to be chosen as function of the frequency at which the video is recorded, since for lower frequencies the movement of an object contour will likely be larger between consecutive movie frames.

Calculation of center coordinate. The updated contour center $\boldsymbol{O}$ for the following segmentation iteration is calculated after the coordinates were checked and possibly corrected. Following Pécréaux the center is calculated as the weighted mean of the contour coordinates $\boldsymbol{P}_i$

$$\boldsymbol{O} = \sum_{i=0}^{N-1} \boldsymbol{P}_i \frac{ds_i + ds_{i+1}}{2}$$

where the weighting factors ds_i are the distances between adjacent contour points:

$$ds_i = |\boldsymbol{P}_i - \boldsymbol{P}_{i-1}| = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$$

Calculating the contour center in this way, we weight coordinates that are farther apart (e.g. for fixed contour-angles farther from the center) more strongly than closely spaced coordinates. This makes the algorithm converge more rapidly to the optimal center of the contour than the barycenter of the points $\boldsymbol{P}_i$ would.

Coordinate interpolation. Because we determine the contour coordinates $\boldsymbol{P}_i$ through the weighted mean of the local coordinates $\boldsymbol{p}_j$, the contour coordinates $\boldsymbol{P}_i$ could start to "wander" away from their originally intended location at the desired angle ϕ_i. This can happen especially when, either the weighting factors m_j or the local coordinates $\boldsymbol{p}_j$, are unevenly distributed tangentially to the contour. To correct this undesirable effect, the contour coordinates are updated after each segmentation iteration, using linear interpolation, to determine their position at the desired angle ϕ_i. Briefly, we first denote the coordinate value of $\boldsymbol{P}_i$ directly after averaging over its local coordinates $\boldsymbol{p}_j$ with $\tilde{\boldsymbol{P}}_i$ at contour angle $\tilde{\phi}_i$; then, we determine the intercept between the straight $\boldsymbol{y}_i = \alpha \boldsymbol{e}_{\phi_i} + \boldsymbol{O}$ and the contour-segment $\tilde{\boldsymbol{P}}_{i-1}\tilde{\boldsymbol{P}}_i$ through which it passes, where $\boldsymbol{e}_{\phi_i}$ is the unit-vector in the direction of the desired angle ϕ_i and $\boldsymbol{O}$ is the updated contour center. Finally, this result yields the final position $\boldsymbol{P_i}$. To determine the line-segment $\tilde{\boldsymbol{P}}_{i-1}\tilde{\boldsymbol{P}}_i$ through which $\boldsymbol{y}_i$ passes, we determine the two adjacent contour coordinates $\tilde{\boldsymbol{P}}_{i-1}$ and $\tilde{\boldsymbol{P}}_i$ with corresponding angles $\tilde{\phi}_{i-1}$ and $\tilde{\phi}_i$, so that $\tilde{\phi}_{i-1} < \phi_i \leq \tilde{\phi}_i$. The contour coordinates are then updated with the interpolated coordinates and written to $\boldsymbol{P}'_i$ as input for the following iteration. Note that we do not interpolate the value of $\boldsymbol{n}_i$, since we hope the contour normal to converge to its correct value by itself over the course of the iterations. Furthermore, the linear interpolation does not represent a loss in precision, since it only pins the position of each coordinate to its desired contour angle ϕ_i, but we still let the coordinate converge to its optimal position at that angle.

Segmentation termination. To determine when to terminate the segmentation process of an image, the algorithm uses two conditions for the contour coordinates $\boldsymbol{P}_i$ and the contour center $\boldsymbol{O}$. After each iteration it checks whether the positions of the $\boldsymbol{P}_i$ and $\boldsymbol{O}$ have changed more than a user-defined tolerance Δ_{coord} and Δ_{center} respectively. Thus, segmentation of an image is finished, when the following conditions are met:

$$|\boldsymbol{O} - \boldsymbol{O}'| < \Delta_{\mathrm{center}} \ \text{ and } \ |\boldsymbol{P}_i - \boldsymbol{P}_i'| < \Delta_{\mathrm{coord}} \, \forall \, i \epsilon N,$$

where again $\boldsymbol{P}'_i$ and $\boldsymbol{O}'$ are the corresponding values from the previous iteration. Furthermore, the algorithm will terminate the segmentation process of an image after a user-defined, maximum number of segmentation iterations N_{max}, if the convergence condition was not achieved previously. Note that Δ_{coord} does not correspond to the segmentation precision $\delta_{1\sigma}$ of the algorithm, which is discussed further below. Δ_{coord} usually has a significantly lower value than $\delta_{1\sigma}$, since it is the convergence criterion for an individual image only, while $\delta_{1\sigma}$ is the result of coordinate fluctuations between images caused by their finite SNR. In case after maximal iterations the convergence criterion is not raised with a given image, the algorithm will restart the segmentation with the next image using the contour coordinates obtained in the former, although not converged. In practice, this does not pose a big problem, since only very few images (0 up to 10, maximum, among several tens thousands of frames in a movie) do not converge. If the criterion is continually not met, when the tolerance is increased, then the image material is considered o insufficient quality.

Refinement of linear fit position. To avoid artifacts that can arise from performing the linear fit at a location away from where the gradient is the largest, the algorithm determines for each local direction φ_j at each $\boldsymbol{P}_i$ the position at which m_j is maximal. Two methods were conceived for doing this:

Method 1 For each direction φ_j we determine the location of the intensity minimum I_{min} and maximum I_{max} of the halo and situates the center position for the linear fit at the location in-between the two extremes, where the intensity value is similar to $I_{\mathrm{center}} = I_{\mathrm{min}} + (I_{\mathrm{max}} - I_{\mathrm{min}})/2$. On the programming level this is done by finding the maximum and minimum values in the array Y_j holding the intensity values of each direction φ_j and then determining the index ι_{cen} of the intensity value that comes closes to the value of I_{center}. The linear fit is then performed over the index-range $[\iota_{\mathrm{cen}} - f_{\mathrm{interp}} \cdot n_{\mathrm{fit}}/2, \iota_{\mathrm{cen}} + f_{\mathrm{interp}} \cdot n_{\mathrm{fit}}/2]$ of Y_j, where n_{fit} is the length over which the linear fit is done (in [px]) and f_{interp} is the number of how many points will be interpolated for each pixel.

Method 2 In this case the algorithm performs a direct search on Y_j for the array-index ι_{cen} at which the incline is maximal. It does this by performing linear fits centered at array-indexes ι over the range $[L_Y - f_{\mathrm{interp}} \cdot N_\iota/2, L_Y + f_{\mathrm{interp}} \cdot N_\iota/2]$, where L_Y is the length of the array Y_j containing the intensity values (L_Y is the same for all φ_j), N_ι is the user-defined search range and again f_{interp} is the integer of how many points will be interpolated for each pixel. This refined search is performed for each local direction φ_j.

Combined with the various segmentation iterations performed for each movie frame, both methods converge to the position where the intensity gradient is maximal. For images of quasi-circular cell contours, both of the described methods were found to give very similar segmentation performance for the final algorithm and will therefore not be treated separately in the following discussion. However, the second method has a significant advantage in cases, where the halo of the optical image is not pronounced, such as bright-field image recordings of RBC ghosts or of giant unilamellar vesicles (GUVs), which both exhibit poor contrast. In these cases, the first method can fail to correctly determine the position of the I_{min} and I_{max}, rendering the algorithm unusable. The second method is much more robust in these cases. A trade-off, however, is that it is more compute intensive due to the much higher number of linear fits being performed. This results in longer segmentation times. For dense cells, such as RBCs, which provide phase contrast images with high contrast, the first method is therefore usually preferred whose segmentation performance will be analyzed in the following sections.

Implementation on GPU. The described algorithm was designed to be parallelizable in order to be run on consumer general purpose graphics processing units (GPGPUs). All operation for the local angles φ_j, such as the pixel interpolations, linear fits and intercept calculations are run in parallel for all contour coordinates $\boldsymbol{P}_i$, so that all values $\boldsymbol{p}_{j,i}$, $\boldsymbol{e}_{j,i}$ and $m_{j,i}$ are determined independently of each other during each iteration. The obtained results are then reduced to obtain $\boldsymbol{P}_i$ and $\boldsymbol{n}_i$ using Eqs (1) and (2). A flow chart of the algorithm is shown in Fig 3, where the region inside the dashed rectangle represents the highly parallelizable section. This design makes the algorithm ideally suited for acceleration on GPUs. Furthermore, since GPUs are specially designed to handle graphics, they possess dedicated hardware units for performing the bi-linear interpolation of the pixel intensities, which greatly accelerates this operation.

The algorithm was implemented using the Open Compute Language (OpenCL), making the code portable and executable on GPUs from different vendors and also conventional CPUs. Comparing the execution time of our implementation on an AMD Radeon HD 7970 to that of an Intel Xeon 5760, we registered a 100 × speed-up on GPU with respect to the CPU version of the algorithm, when imaging RBCs using the segmentation parameters listed in Table 1. Our implementation of this high-precision flicker spectroscopy contour tracking algorithm (HiPFSTA) has been made publicly available at: www.github.com/michaelmell/hipfsta

Analysis of an individual image with five segmentation iterations takes roughly 0.1s. It depends on the number of iterations per image, which for RBCs is usually 3 to 5 iterations using the listed parameter values. Therefore, execution on the GPU enables the processing of movies recorded with the FASTCAM SA-3 in a reasonable time of ~2.2hours (for $8 \cdot 10^4$ movie frames of a 40s video recorded at 2000FPS). On the CPU mentioned above this would not be feasible.

Analysis of segmentation performance

Tracked object examples

In its current form, the algorithm can be applied to spherical and non-spherical objects, which fulfill two requirements: 1) They exhibit a sufficiently strong contrast halo necessary for the algorithm to function correctly. 2) It is possible to uniquely assign an individual radius value R_i to a corresponding angle ϕ_i, when describing the contour in polar coordinates. The second requirement is the result of the interpolation used to fix the angular position ϕ_i of the contour coordinates and could be relieved by conceiving a different method. The segmentation algorithm has so far been successfully applied to various objects including RBCs, *E. coli*, GUVs and lymphocytes for which we show example images overlaid with their tracked contour in Fig 4. All images where obtained using bright-field microscope setup except for the image of *E. coli*, which was obtained using a phase-contrast objective. As we will see in the next section, the segmentation precision crucially depends on the optical contrast afforded by the object.

Real-space

Our segmentation algorithm has been developed to overcome the shortcomings detected in the previous algorithm proposed in Ref. [13], for which we have found that the discretization of the pixel-grid leads to artifacts in the final contour coordinates in the algorithm. The reason for those artifacts is that the initial position for performing the linear fit is confined to the coordinates of the pixel-grid, *i.e.* the pixel centers, which determine the coordinates of every point in the halo. When the position of the halo moves against the pixel-grid, this will gradually change the position at which the linear fit is performed and therefore affect the incline m of the fitted line as the fit-position draws closer to the intensity minimum or maximum of the

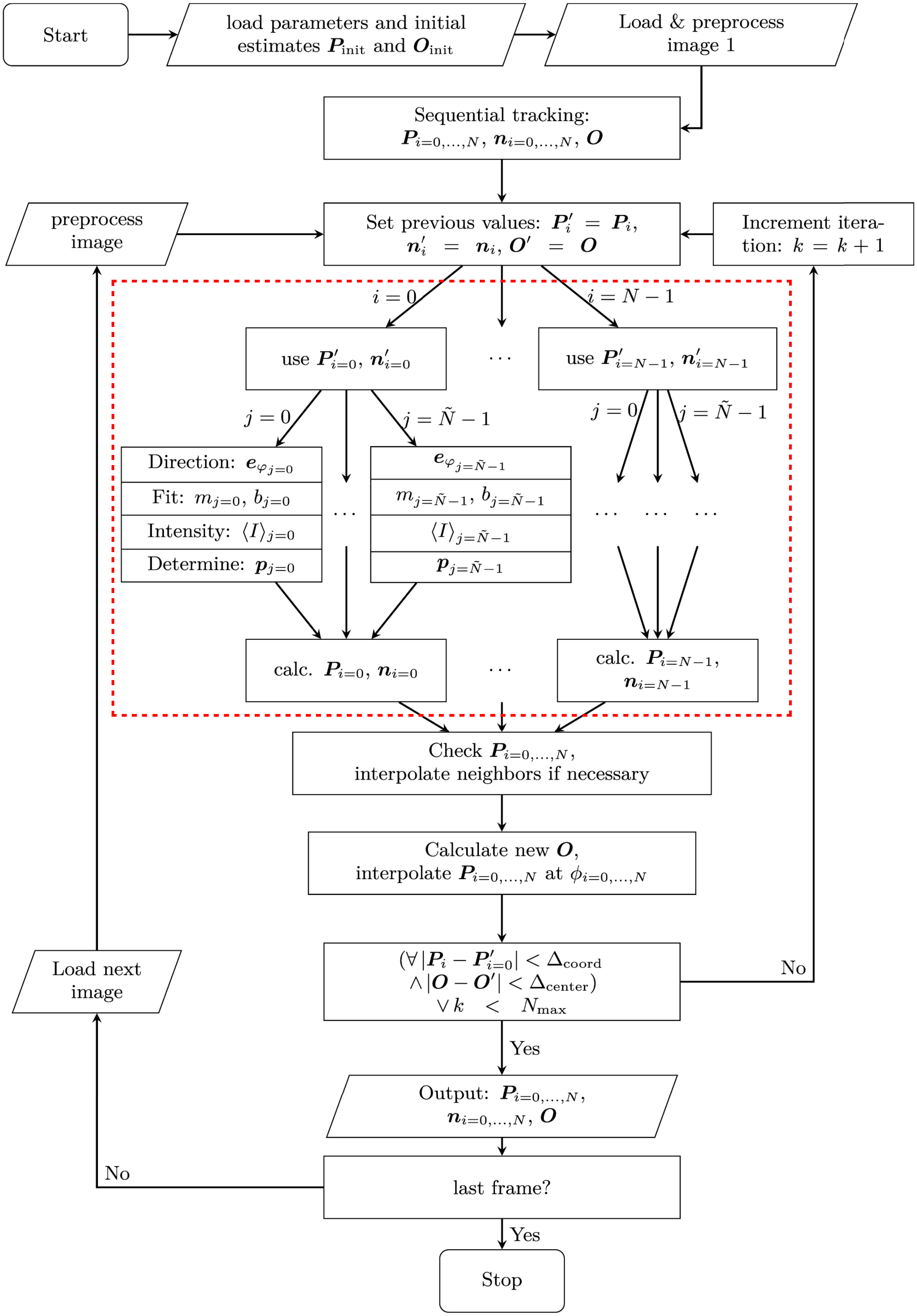
Start
load parameters and initial estimates $\boldsymbol{P}_{\text{init}}$ and $\boldsymbol{O}_{\text{init}}$
Load & preprocess image 1
Sequential tracking: $\boldsymbol{P}_{i=0,\ldots,N}$, $\boldsymbol{n}_{i=0,\ldots,N}$, $\boldsymbol{O}$
preprocess image
Set previous values: $\boldsymbol{P}'_i = \boldsymbol{P}_i$, $\boldsymbol{n}'_i = \boldsymbol{n}_i$, $\boldsymbol{O}' = \boldsymbol{O}$
Increment iteration: $k = k + 1$
$i = 0$
$i = N - 1$
use $\boldsymbol{P}'_{i=0}$, $\boldsymbol{n}'_{i=0}$
$\ldots$
use $\boldsymbol{P}'_{i=N-1}$, $\boldsymbol{n}'_{i=N-1}$
$j = 0$
$j = \tilde{N} - 1$
$j = 0$
$j = \tilde{N} - 1$
Direction: $\boldsymbol{e}_{\varphi_{j=0}}$
Fit: $m_{j=0}$, $b_{j=0}$
Intensity: $\langle I \rangle_{j=0}$
Determine: $\boldsymbol{p}_{j=0}$
$\ldots$
$\boldsymbol{e}_{\varphi_{j=\tilde{N}-1}}$
$m_{j=\tilde{N}-1}$, $b_{j=\tilde{N}-1}$
$\langle I \rangle_{j=\tilde{N}-1}$
$\boldsymbol{p}_{j=\tilde{N}-1}$
$\ldots$ $\ldots$ $\ldots$
calc. $\boldsymbol{P}_{i=0}$, $\boldsymbol{n}_{i=0}$
$\ldots$
calc. $\boldsymbol{P}_{i=N-1}$, $\boldsymbol{n}_{i=N-1}$
Check $\boldsymbol{P}_{i=0,\ldots,N}$, interpolate neighbors if necessary
Calculate new $\boldsymbol{O}$, interpolate $\boldsymbol{P}_{i=0,\ldots,N}$ at $\phi_{i=0,\ldots,N}$
$(\forall\,|\boldsymbol{P}_i - \boldsymbol{P}'_{i=0}| < \Delta_{\text{coord}}$ $\wedge\,|\boldsymbol{O} - \boldsymbol{O}'| < \Delta_{\text{center}})$ $\vee\, k < N_{\max}$
No
Yes
Load next image
Output: $\boldsymbol{P}_{i=0,\ldots,N}$, $\boldsymbol{n}_{i=0,\ldots,N}$, $\boldsymbol{O}$
No
last frame?
Yes
Stop

Fig 3. Flow-chart of the segmentation algorithm illustrating the steps that are taken in processing the movie. The section indicated by the red rectangle indicates the embarrassingly parallel part of the algorithm, which can easily be adapted for parallel processing on a GPU.

https://doi.org/10.1371/journal.pone.0207376.g003

halo. This is illustrated in Fig 5A, where we show three fits at different positions of the intensity profile of the halo. Fitting close to the maximum or minimum alters the incline and consequently the final, tracked contour coordinate. While in reality fitting should not occur this close to the extremes, this effect is still appreciable. Fig 5B shows the effect of this on the tracked contour of a *synthetic* phase-contrast image of a smooth, structureless circle, where the solid red line represents the position of the phase boundary of the circle used for the image synthesis (see Methods). For illustration purposes, this synthetic dataset does not contain any noise and thus has an infinity signal-to-noise ratio (SNR = ∞; see Methods). The dashed black line and the black squares show the pixel position used for fitting in Pécréaux' algorithm and the tracked contour coordinates. The white diamonds represent the tracked contour coordinates from our algorithm. As can be seen, the contour coordinates calculated from that algorithm move slightly in the opposite direction, when "jumps" in the pixel positions occur. This occurs whenever the pixel position, about which the linear fit is performed, "jumps" transversely to the halo and thus produces a slight change in the incline of the fitted straight *m* as illustrated in Fig 5A. Thus, although the image does not contain any noise, the algorithm itself causes artifacts in the final contour.

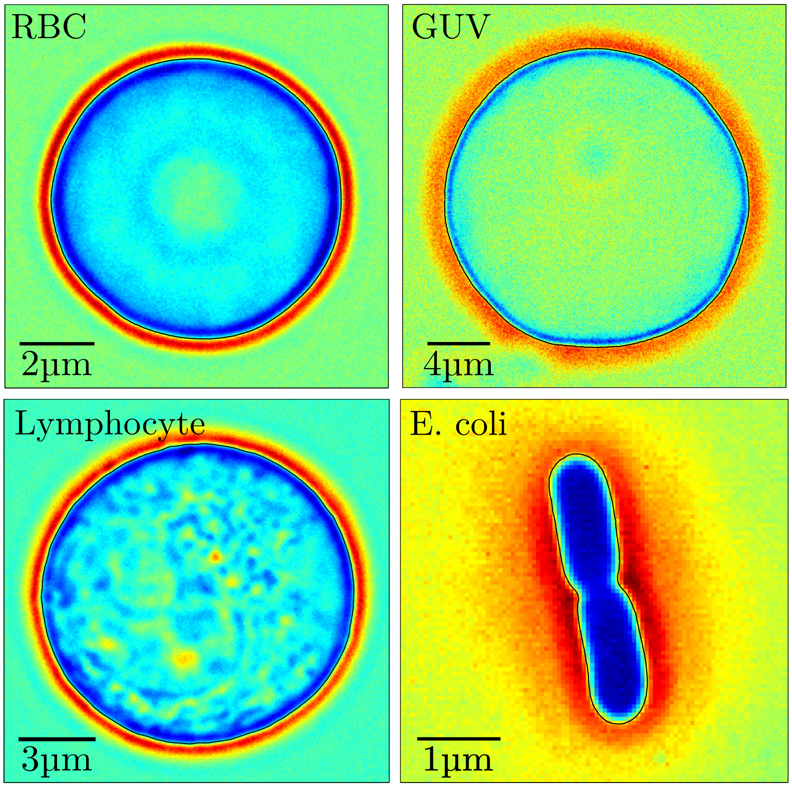

Fig 4. Examples of microscopic objects that have successfully been tracked using our new segmentation algorithm. The obtained contour is indicated by the black line that is overlaid on each image. All images were recorded with bright-field light-microscopy, except for the E. coli, which was recorded using a phase-contrast objective (100x, NA1.45).

https://doi.org/10.1371/journal.pone.0207376.g004

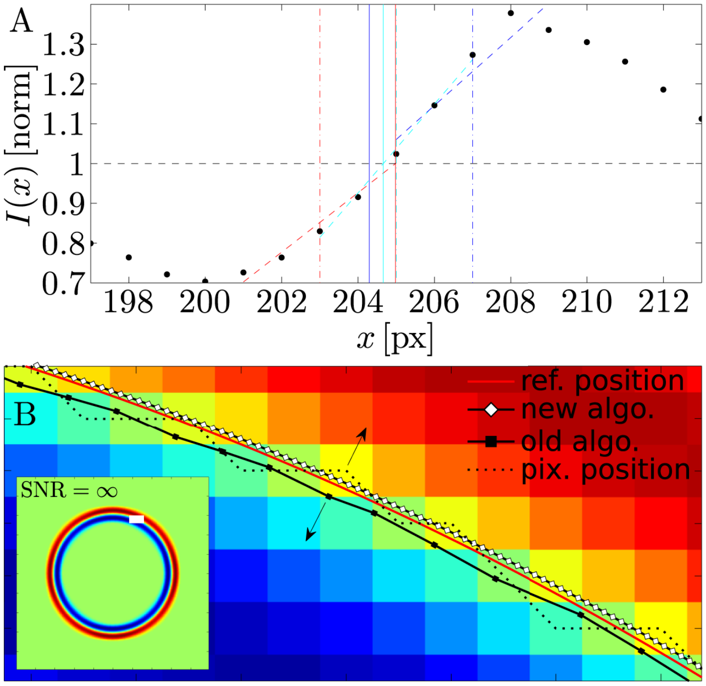

Fig 5. Image pixilation causes jitter in the positions obtained with the classical algorithm. (A) Illustration of how fitting at an offset from the gradient maximum affects the obtained fit-slope and results in a deviation in the final coordinate. (B) Comparison of the reference contour and the contour coordinates obtained from the previous method [13], and our refined algorithm here described. The artifact described in (A) leads to transverse shifts in the final position (black squares), whenever the pixel-position about which the fit is performed (dashed line) shifts transversely to the contour. The red line represents reference position of the circle used in the synthesis of the image with SNR = ∞, which is shown in the inset indicating the shown area with a white rectangle. The tracked coordinates obtained from the new algorithm (white diamonds) follow the reference position precisely.

https://doi.org/10.1371/journal.pone.0207376.g005

Although this effect is subtle, it can lead to misinterpretations when analyzing the contour coordinates in real-space. This is evident when comparing time series of the contour radii $R(\phi_i,t_n)$ at a fixed angle ϕ_i as produced either by the algorithm in Ref. [13] or by our algorithm for an artificial lipid vesicle. The distribution of $R(\phi_i,t_n)$ from the former algorithm exhibits maxima and minima, which are spaced approximately 1px apart as shown in Fig 6A. This

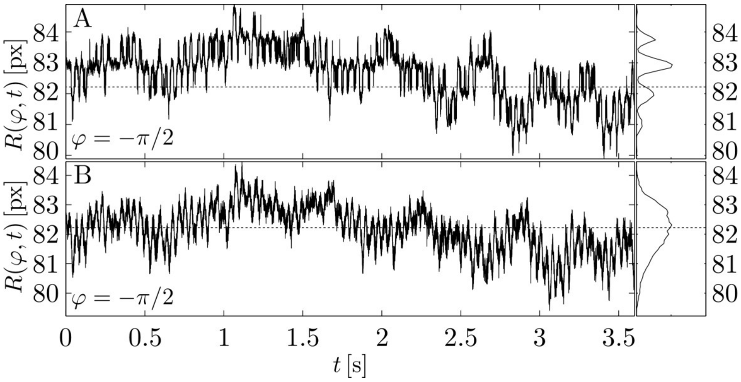

Fig 6. Comparison of the radius fluctuation time-series obtained from a measurement of a POPC GUV using Pécréaux' algorithm (A) and our algorithm (B). The artifacts from imprecise fitting leads to various maxima in the displacement-distributions spaced 1px apart. This artifact is not present when using our new method.

https://doi.org/10.1371/journal.pone.0207376.g006

behavior is the direct consequence of the artifact due to pixelation described above (with pixelation we refer to the discretization of the continuous image into the discrete pixel grid). It is the reason we take great care to optimize the fit-location in our algorithm (see **Refinement of linear fit position**) and is not observed in the radius time series obtained from the same dataset using the new segmentation algorithm shown in Fig 6B.

In order to quantify the precision of the new algorithm experimentally, we recorded and tracked movies of dried RBCs, which are used as static reference objects. The dried RBC are well suited as reference fluctuation-less objects, since they exhibit a similar level of contrast and halo-shape as living RBCs. Additionally we synthesized computer-generated phase contrast images as proposed by Usenik[21]. To determine the segmentation precision, we measure the variability of the contour coordinates obtained with our algorithm. For this, we calculate the distance $\delta_{i,n}$ between corresponding coordinates at the same contour angle ϕ_i from consecutive frames $n - 1$ and n, so that $\delta_{i,n} = |\boldsymbol{P_{i,n}} - \boldsymbol{P_{i,n-1}}|$. Since the contours in the recorded image of dried RBC and the synthetic images are static, this gives us information on how much the contour position varies between frames due to the finite precision of the algorithm. We then calculate the distribution P of the inter-coordinate distances $\delta_{i,n}$ and define its standard-deviation 1σ as the segmentation precision $\delta_{1\sigma}$. Note that since $\delta_{1\sigma}$ is obtained from the differences of coordinates pairs its value is expected to be $\sqrt{2}\times$ larger than the error of the individual coordinates. The reason we determine the segmentation precision in this manner and not using the radius fluctuation $\langle h(t)^2\rangle = \langle (R(t) - \langle R(t)\rangle)^2\rangle$ over the whole time-series, is because our method will not be affected by slight shifts in focus or vibrations of the experimental setup, which can occur on time-scales of various seconds. The positional accuracy can be estimated theoretically by determining the error in the x-position σ_x of the intercept between the linear regression of the intensity gradient with slope m and the average image intensity $\langle I\rangle$. This straight forward calculation yields (see SM):

$$\sigma_x = \frac{\sigma_I}{m}\sqrt{\frac{1}{U} + \frac{1}{T}} \tag{3}$$

with standard deviation σ_I of the image intensity and the slope m of the intensity halo. The proportionality constant $\sqrt{1/U + 1/T}$ depends on the number of fitted values U and averaged intensity values T. Note that σ_x does not correspond directly to $\delta_{1\sigma}$. As mentioned above $\delta_{1\sigma}$ is $\sqrt{2}\times$ larger than the coordinate error, which in turn are the weighted average of M fits in the directions of the local angles φ_j, each of which possess a different error σ_x due to different slopes m (see Fig 2(C)). However, the proportionality $\delta_{1\sigma} \propto \sigma_I/m$ should remain unchanged, since the directions with largest slope m will dominated the weighted sum. To test the theoretical estimate, we determine the segmentation precision for images with different signal-to-noise ratios (SNR) and contrast levels (different m) using synthetic images (see Methods). The segmentation precision $\delta_{1\sigma}$ is inversely proportional to the image SNR, $\delta_{1\sigma} \propto 1/\text{SNR}$ as can be seen in Fig 7A. After renormalizing the abscissa to σ_I/m all data points fall onto single line confirming $\delta_{1\sigma} = \sigma_I/m \cdot K$ with proportionality constant K. We can compare the value of K to that of $\sqrt{1/U + 1/T}$ to obtain an estimate of the precision improvement due to the weighted averages (see Eq (1)) compared to a single fit. Although each fit consists of n_{fit} data-points (see Table 1) the number of *independent* pixel values being fit is actually $U = n_{\text{fit}}/f_{\text{interp}} = 5$. Accordingly, $T = n_{\text{bkgr}}/f_{\text{interp}} = 20$ for the number of *independent* pixel values being averaged for the background intensity. We therefore have $\sqrt{1/U + 1/T} = 0.5$. From Fig 7B, we obtain the proportionality constant $K = 0.54$. Taking into account the factor $\sqrt{2}$ from calculating $\delta_{1\sigma}$, we estimate an improvement of $0.5/(0.54/\sqrt{2}) = 1.31$.

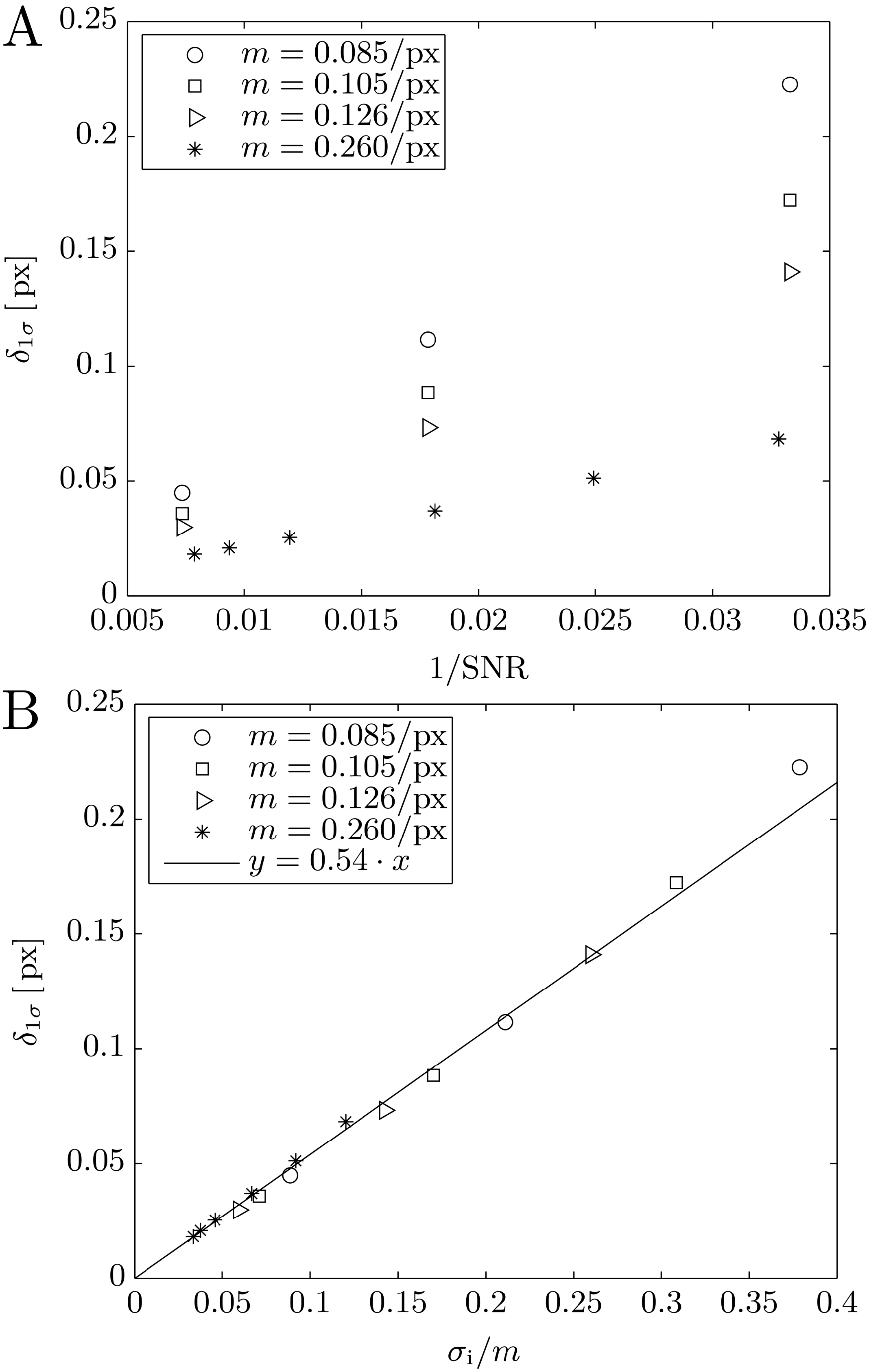
A
$m = 0.085/\mathrm{px}$
$m = 0.105/\mathrm{px}$
$m = 0.126/\mathrm{px}$
$m = 0.260/\mathrm{px}$
$\delta_{1\sigma}$ [px]
0
0.05
0.1
0.15
0.2
0.25
0.005
0.01
0.015
0.02
0.025
0.03
0.035
1/SNR
B
$m = 0.085/\mathrm{px}$
$m = 0.105/\mathrm{px}$
$m = 0.126/\mathrm{px}$
$m = 0.260/\mathrm{px}$
$y = 0.54 \cdot x$
$\delta_{1\sigma}$ [px]
0
0.05
0.1
0.15
0.2
0.25
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
σ_{i}/m

Fig 7. Segmentation precision depends on image noise and the slope of the intensity gradient. (A) Segmentation precision $\delta_{1\sigma}$ versus $1/SNR$ for a dried RBC at different exposure times ('⁎') and for three synthetic datasets ('□,○ and Δ') with different level of contrasts/slopes m. The precision values from each dataset fall onto a straight determined by the contrast/slope m. (B) After renormalizing x-axis to σ_i/m all values fall onto a straight $y = 0.54 \cdot x$ allowing for easy estimation of the segmentation precision from the two known quantities σ_i and m. Note that the contrast/slope values are for to unit background intensity.

https://doi.org/10.1371/journal.pone.0207376.g007

From Eq (3) it follows that we can improve segmentation precision by increasing the SNR of the image through higher illumination intensity or longer exposure times. Furthermore, the amount of contrast produced at the object boundary plays a crucial role in determining how precise the algorithm is able to track its contour, since it determines m. Moreover, since m and σ_I are easily obtained from a single image, this allows for simple estimation of the expected segmentation precision. Table 2 lists the data shown in Fig 7. As indicated, we achieve a positional accuracy $\delta_{1\sigma}$ of 0.01px to 0.03px for SNR-values of 127 to 84 corresponding to typical experimental frame rates of 1500 to 3000fps. For our setup with an effective pixel-size of 50nm this translates into a positional accuracy of below 0.9nm to 1.25nm for dried RBCs specimens. We also estimated the precision for RBCs and GUVs in buffer, which exhibit a lower contrast and therefore lower value for m. We find 0.05px and 0.3px respectively using values for m taken from bright field microscopy observations. For completeness we also include the value of $\delta_{1\sigma}/\sqrt{2}$, which is expected to represent the error an individual coordinate, instead of their difference. For a RBC in buffer, we estimate a value of $\delta_{1\sigma}/\sqrt{2} = 0.04$px corresponding to 1.85nm.

The segmentation precisions for GUVs previously reported by Pécréaux [13] and Usenik [21] are not exactly comparably to our results, since these authors used phase-contrast, instead of bright field microscopy. We can however attempt to estimate the precision that we would expect from our algorithm. From Eq (1) and Fig 1 we estimate an intensity-normalized $m = ((I_{\max} - I_{\min})/\langle I \rangle)/n_{\text{pix}} = ((525 - 435)/480)/8$px, where $I_{\min}$ and $I_{\max}$ are the minimum and maximum intensity of the halo, $\langle I \rangle$ is the average and n_{pix} is the number of pixels in the gradient-range. For σ_I we estimate $\sigma_I = 3/\langle I \rangle = 0.007$, assuming a standard deviation of 3 grey level units. Using these values we obtain an estimated segmentation precision of $\delta_{1\sigma} = 0.54 \cdot \sigma/m = 0.16$px, which is comparable to the 0.1px-precission that Pécréaux claimed for his algorithm [13]. Usenik et al. [21] claimed a precision of 26.9 to 34.1nm for their algorithm at a pixel-size of 35nm. Therefore, their precision is not significantly below pixel-size and we expect our algorithm to perform significantly better. However, the value for the latter case was achieved at SNR = 23 dB, which for our SNR-definition corresponds to a very low SNR = $10^{23/20} = 14.13$ (see Methods). Finally, for RBCs, the accuracy of our new software implementable with contrast imaging is competitive with other sophisticated microscopy hardware such as the diffraction

Table 2. Values for the dried RBC datasets compared to RBCs and GUVs in buffer solution.

	$\text{RBC}_{\text{dried}}$	$\text{RBC}_{\text{dried}}$	$\text{RBC}_{\text{dried}}$	$\text{RBC}_{\text{dried}}$	$\text{RBC}_{\text{dried}}$	$\text{RBC}_{\text{dried}}$	$\text{RBC}_{\text{buffer}}$	GUV
FPS	1500	2000	3000	6000	10000	15000	2000	1500
SNR	127.3	106.8	83.9	55.2	40.1	30.4	101.5	137.9
m	0.2355	0.251	0.259	0.272	0.272	0.273	0.1	0.014
σ_I/m	0.033	0.037	0.046	0.067	0.092	0.120	0.097	0.517
$\delta_1\sigma$[px]	0.018	0.021	0.025	0.037	0.051	0.068	0.052*	0.279*
$\delta_{1\sigma}/2^{½}$ [px]	0.013	0.015	0.018	0.026	0.036	0.048	0.037*	0.197*

Values for the dried RBC datasets shown in Fig 7 compared to typical values for measurements of RBCs and GUVs in buffer solution. The imaging precision $\delta_{1\sigma}$ and $\delta_{1\sigma}/2^{½}$for the latter two cases was extrapolated from their values for σ_I/m using the equation $\delta_{1\sigma} = 0.54\ \sigma_I/m$ (indicated by *).

https://doi.org/10.1371/journal.pone.0207376.t002 phase microscope [33] used by the authors of references [9,12,34], who report an accuracy of 3.3nm for the RBCs thickness-fluctuations due to the DPMs limited optical path stability [22,33]. However, their experimental setup uses a 40× (NA0.65) oil-immersion objective with 400nm resolution. Since we use an oil-immersion objective with 100 × magnification (NA 1.4), we should have better lateral resolution along the contour, which should translate into a diffraction-limited lateral resolution of around 200nm. The proposed method deals with the analysis of the contrast halo, which has been implemented as a super-localization software that allows contour segmentation at sub-pixel resolution with bright field/phase contrast images. However, our super-localization method can be considered broader in scope, as potentially implementable with other optical settings no dealing with contrast halos. In dark-field, for instance, amplitude dependent glows formed at the rim of the object are potentially detectable by our software. Cells with sufficient optical contrast, such as RBCs, may be efficiently segmented using images obtained upon a high scattering luminosity. Differential interference contrast (DIC) has no halo yet, however, contour segmentation could be performed after image integration leading to sufficient membrane contrast.

Fourier space analysis. To analyze the performance of our algorithm in Fourier space and quantify the effect that limited segmentation accuracy has on, we calculate the spectra obtained from the synthesized datasets. Calculation of the spectra is done as described by Pécréaux [13]. Briefly, a tracked contour (R_i,ϕ_i) is transformed into a height-profile (h_i,ϕ_i) above and below a circular equilibrium radius R. The h_i are then decomposed into their Fourier components, n, which for the continuous case can be written as

$$h(\phi) = R\left[1 + \sum_{n=1}^{\infty} a_n \cos(n\phi) + b_n \sin(n\phi)\right],$$

where R is the equilibrium radius of the contour. The spectrum is then given by the RMSD of the modulus $|c_n| = \sqrt{a_n^2 + b_n^2}$. The spectrum is then given by the RMSD of the modulus $|c_n| = \sqrt{a_n^2 + b_n^2}$. Here we use the normalization proposed by Pécréaux [13], so that absolute values are directly comparable to his work. The final spectrum is then given by

$$\langle |u_n|^2 \rangle = \frac{\pi \langle R \rangle^3}{2} \left(\langle |c_n|^2 \rangle - \langle |c_n| \rangle^2 \right),$$

where the wavenumber q relates to the mode number n through $q = n/R$.

Above a given wavenumber compatible with a minimal resolvable wavelength, the segmentation algorithm in Ref. [13] produces a spectrum dominated by noise, which continues to increase towards higher wavenumbers. The authors attribute this to the discrete detection of the contour, which causes an increased floor noise as the wavelength of the mode becomes smaller than 4 × pixel-size. He therefor argues that as a result of a such discretization there exist two main limitation to his algorithm: 1) the amplitude precision orthogonal to the contour, and 2) the sampling precision in parallel to the contour [13].

The spectra obtained for the synthetic spherical object with the new algorithm have a significantly different shape than the one obtained from Pécréaux' algorithm (see Figs 8 and 9). Whereas our algorithm yields a spectral shape that is compatible with the pixel-grid structure, the algorithm in Ref. [13], however, yields spurious amplitudes in the whole range of wavenumbers analyzed. Indeed, our algorithm computes lower amplitudes, which are compatible with the level of noise artificially introduced for these ideal spherical objects. A representative example is shown in Fig 8 and compared to the spectrum obtained from Pécréaux' algorithm for a synthesized dataset (SNR = 106.7). The shape of the fluctuation spectrum is reminiscent of a

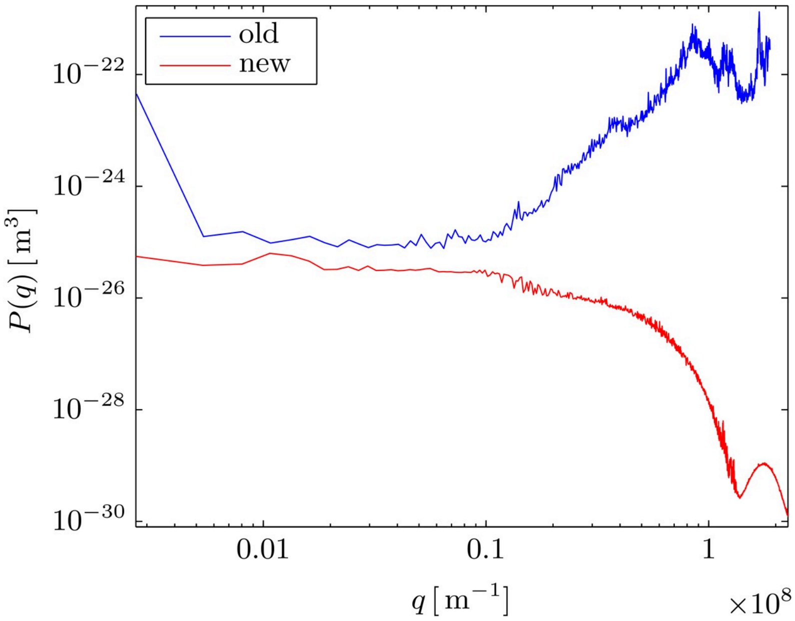

Fig 8. Comparison of spectra obtained from the same dataset (SNR = 106.7) using the new and old algorithm. The spectrum from the old algorithm increase at high q, since the pixilation artifact causes increased noise as the wavelength draws closer to the pixel-size. The spectrum from the new algorithm exhibits a minimum at the location corresponding to wavenumber q_{pix} of a pixel.

https://doi.org/10.1371/journal.pone.0207376.g008

sinc-function, with its first minimum located roughly at the wavenumber q_{pix} corresponding to the size of 1px. This spectrum represents the spectral noise floor resulting from the finite SNR of the synthesized images. It is the result of the noise of the individual pixels, whose influence on the final spectrum is convoluted with the frequency response of the linear interpolation used in the segmentation algorithm. A linear interpolation can be written as a convolution of the sample values with triangular pulse [35], whose Fourier transform and thus frequency response is proportional to a sinc^2 function. It is this dependency of the form factor of static objects, which causes the noise floor *not* to be evenly distributed across the spectrum, but to follow a sinc-like shape corresponding to the structure form factor of noisy pixels. By changing the pixel size in the synthesized images, we confirm the inverse scaling behavior expected for the pixel-structure form factor (see Fig 9A). Note that it is therefore possible to change the shape of the spectral noise floor by changing the effective pixel-size either experimentally or through binning and other forms of digital image pre-processing, such as rescaling with bi-cubic interpolation.

Our findings lead us to a slightly different conclusion of the spectral increase of Pécréaux' algorithm at large wavenumbers. Considering the Nyquist-Shannon sampling theorem the theoretical cutoff for spatial frequencies that can still be resolved is $q_{\mathrm{Ny}} = N/(2 \cdot R)$ [36]. For Pécréaux' algorithm with on average one contour coordinate point per pixel this should approximately be the inverse of two times the wavenumber of the image pixel-size: $q_{\mathrm{Ny}} \sim q_{\mathrm{pix}}/2$. Modes with wavenumbers $q > q_{\mathrm{Ny}}$ should exhibit aliasing and therefor increase in spectral amplitude. This is observed for synthetic datasets using our algorithm, when $q > q_{\mathrm{Ny}} = N/(2 \cdot R) = 2048/(2 \cdot 61\mathrm{px}) = 16.8\mathrm{px}^{-1}$ (see SM). The fact that the cross-over wavenumber q_c above which noise starts dominating in Pécréaux' algorithm is much smaller than $q_c = q_{\mathrm{pix}}/2$, suggests that the observed increase in spectral amplitude is primarily due to the artifacts in the determined orthogonal contour positions (see Figs 5 and 6). These transverse artifacts due to pixelation will increasingly affect the mode amplitudeswhen the modes' wavenumbers become

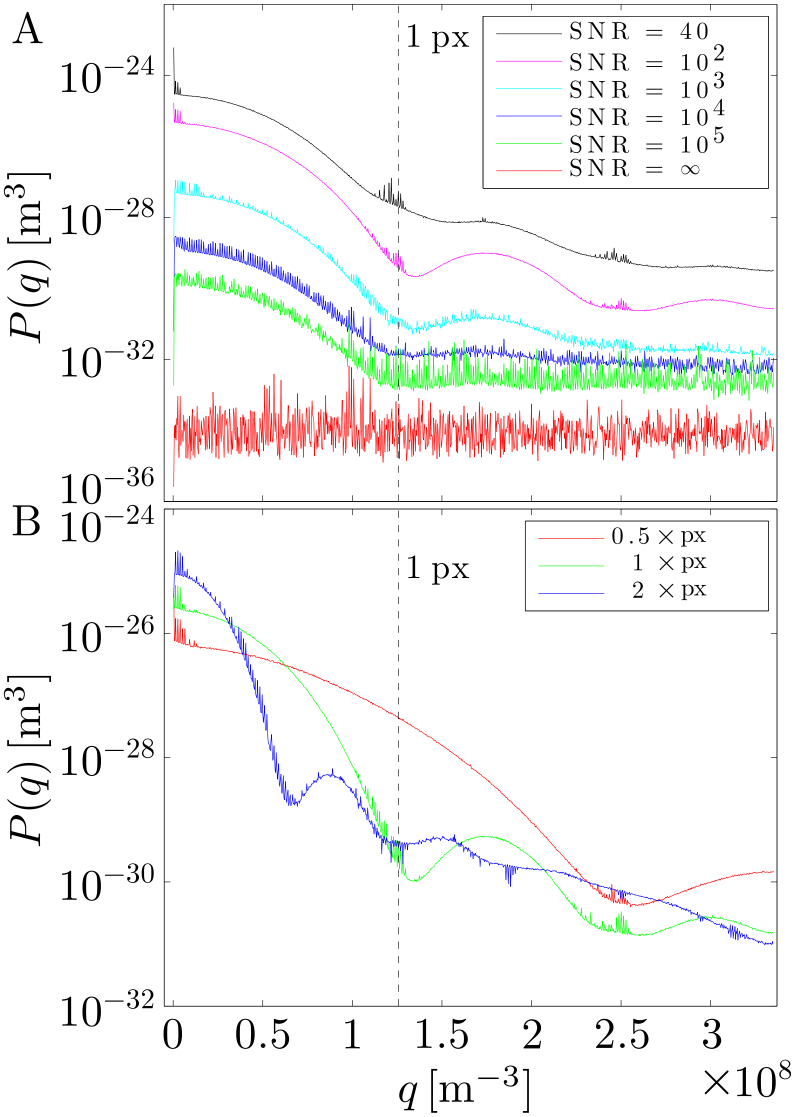

Fig 9. Shape and level of the noise floor depends on the pixel size and SNR. (A) Illustration of the inverse scaling behavior of the spectrum with pixel size. 1 × px corresponds to the experimental pixel size. The shape of the noise floor is the result of the pixel-noise being convoluted with kernel from the linear interpolation. (B) Plot of the spectra obtained for different SNRs. The noise floor decreases for higher (better) SNRs. At high SNRs (SNR $\geq 10^4$) it begins to enter the noise due to numeric imprecision. Since this noise is independent of pixels, spectrum for SNR = ∞ is flat (red line).

https://doi.org/10.1371/journal.pone.0207376.g009

similar to q_{pix}. The new segmentation algorithm does not produce these artifacts in the transverse coordinate, neither propagates errors along the longitudinal position in the contour, differently to the previous method, which determines coordinates in a given point from the formers. That method is thus prone to propagate possible errors to adjacent points in the contour, which favors (in Fourier space) multiplicative deviations at high wavenumbers ($q \geq q_{px}$), just in the region where it becomes inapplicable.

To better understand the effect of image noise on the resulting spectrum from our algorithm, we synthesized various datasets at different SNRs. The synthetized images correspond to ideal circular contours with a pixelation noise superposed. As expected, the noise floor decreases with increasing SNR (see Fig 9A), which can be interpreted as the form-factor of the

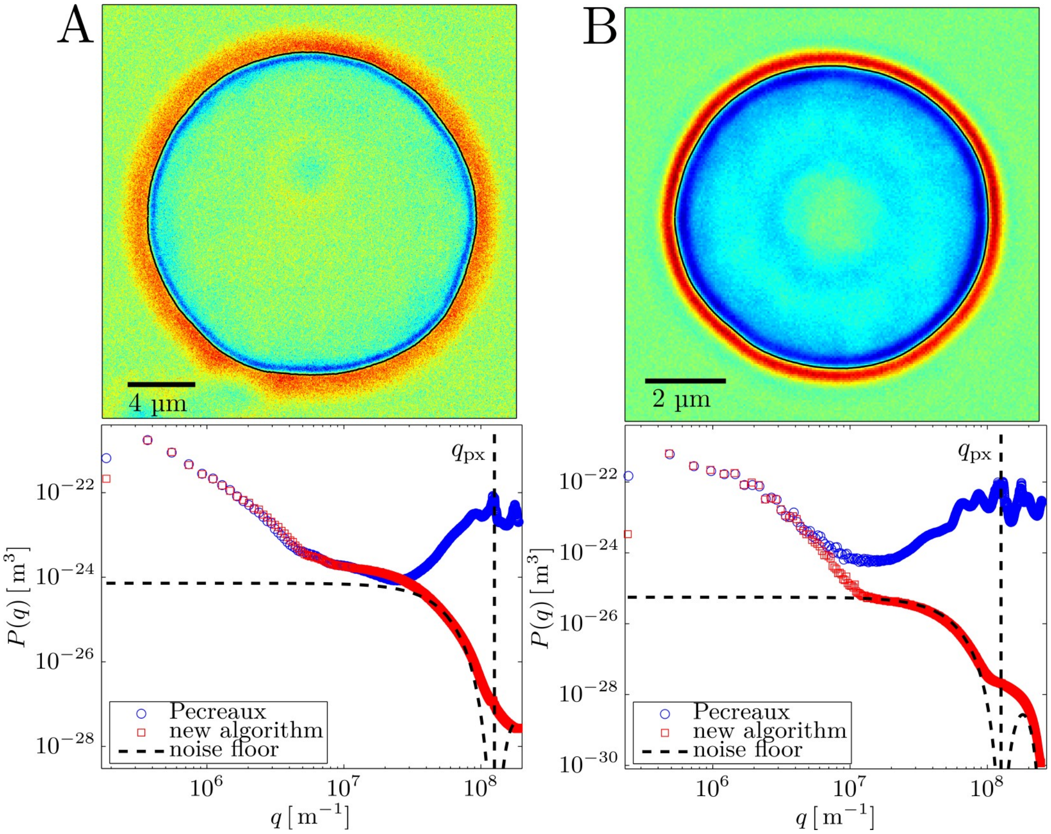

Fig 10. Example flicker spectra of GUV and RBC datasets obtained with the classical and new segmentation algorithms. Top panels: Instantaneous snapshot of a fluctuating GUV made of POPC (A) a healthy RBC undergoing flicker motions (B). The microscopy images are taken with a CMOS camera in the PhC mode at the equatorial plane of the quasi-spherical objects (see Methods for details), and correspond to one frame taken arbitrarily in a movie recorded at 6kfps. Lower panels: Fourier space fluctuation spectra obtained from the statistical analysis of the membrane contour fluctuations. Results from the proposed new contour-segmentation algorithm are compared with results from the classical algorithm described in [13]. The pixelation form-factor, as empirically obtained from optically equivalent synthetic contours (non-fluctuating but including similar noise than the CMOS camera), are plotted as dashed lines with the characteristic sinc-like shape of a squared pixel grid. These lines represent the minimal floor noise corresponding to the instrumental "fluctuations" due to the electronic noise of the CMOS camera implemented with the current optical configuration. The wavenumber corresponding to a distance of one-pixel in Fourier space is plotted as a vertical dotted line labelled $q_{px} = 2\pi/px$.

https://doi.org/10.1371/journal.pone.0207376.g010

noisy pixel-grid; this is, in Fourier space, the convolution between an instrumental white noise (the electronic noise introduced by the camera) with the linearly-interpolated square grid the contour is embedded. At very high SNRs the numerical precision of the segmentation algorithm itself becomes the limiting factor. Since it is independent of the images' pixels, it results in a q-independent noise floor in the spectrum that for SNR $\geq 10^4$ increasingly dominates the high-q range of the spectral noise floor from pixel-noise, until it is the only noise component at SNR = ∞. This suggests that the segmentation precision of our algorithm may experimentally only be limited by the image SNR and in the future could be improved by using new low-noise, digital imaging devices such as scientific CMOS (sCMOS) cameras [37].

Experimental fluctuation spectra. Although this work is almost focused in the description of the algorithmic aspects of our improved segmentation method, we will finish by discussing on its applicability to study the spectrum of the membrane fluctuations in real fluctuating objects. Fig 10 compares the results obtained from our proposed algorithm with those from the Pecréaux' method, in two very different cases: A) Thermal fluctuations in giant unilamellar vesicles (GUVs) made of a very flexible, single lipid (POPC) bilayer membrane, and a relatively low optical contrast between the vesicle lumen and the exterior. B) Membrane fluctuations in healthy red blood cells (RBCs) with a complex lipid bilayer coupled to a rigid spectrin skeleton. Due to the presence of high amounts of hemoglobin, and other cytoplasmic proteins, the optical contrast is in this case much higher than with the former case of empty GUVs. The upper images correspond to instantaneous photographs of the near-circular fluctuating objects at the equatorial plane, and the lower plots represent the amplitude spectra of the membrane fluctuations as decomposed in Fourier space (see section "Fourier space analysis"). In the case of flexible GUVs, the amplitude of the shape fluctuations is large, which makes the spectral amplitudes to be quite high, and resolvable, at low wavenumbers corresponding to distances larger than the pixel grid (q_{pix} = 1.25 10^{-7} m^{-1}). At $q << q_{pix}$ both algorithms give almost identical results, which indicates the mutual robustness of the two segmentation algorithms to determine spatial correlations between membrane elements separated far beyond one pixel. However, since the optical contrast is low in this case, hence the SNR quite low, then the floor noise arising from contour pixelation is rather high, at $q \geq q_{pix}$ becoming of the same order as the fluctuation amplitudes (see Fig 10A bottom). In that high-q regime ($q > q_{pix}$), the new algorithm calculates spectral amplitudes that correspond to the structure factor of the pixelized contour, which is nearly describable as a sinc-like function that can be considered the floor noise intrinsic to the considered optical configuration (see Fig 9). However, the Pecréaux' algorithm returns in this regime spurious values of the spectral amplitudes, which arise from the sub-pixel artifacts described above. As a consequence, Pécréaux' method is quite limited in this sub-pixel regime, where the spectral amplitudes become chiefly determined by a high pixelation noise, in this case dominated by the errors in discriminating the positions of very neighboring points in the contour. Our method, however, is very robust in determining the contour position independently of the radial direction chosen, which makes the segmentation uncertainty independent of the lateral distance between two points along the contour. In our method, consequently, the pixelation noise is just minimal, and stands at the value corresponding to the instrumental noise of the image obtained from a given optical configuration. This situation becomes especially evident with RBC images (see Fig 10B), where the higher optical contrast allows a higher SNR that translates into a much lower pixelation floor noise than in less contrasted GUV membranes. For RBCs, and other real cells, the sinc-like structure factor of the pixel grid is more than 10-fold lower than for GUVs, allowing for robust analysis of the fluctuation amplitudes at much higher wavenumbers than the previous methods. A notable advantage of the proposed algorithm, associated to its intrinsic robustness and low uncertainty in lateral segmentation, is that the floor noise due to pixelation is completely recognizable and predictable, which allows for correct identification and eventual subtraction from further deconvolution analysis. From the compared analysis above, it seems quite clear that the proposed algorithm can bring improved measurements that result in newer, and deeper, analyses than possible with former approaches.

Conclusion

We have developed a high-precision flicker spectroscopy contour tracking algorithm based on an intensity-gradient segmentation schema (HiPFSTA, available open-source at www.github.com/michaelmell/hipfsta), which is designed to run on graphical processing units and makes extensive use of image-oversampling using the processing power of these processors. Segmentation precision was tested using images of dried RBC and synthetic images and quantified in real-space as well as Fourier space, relevant for flickering spectroscopy. The algorithm was shown to achieve segmentation precision down to 2nm for experimental images of dried RBCs at SNRs of about 100, which corresponds to frames rate of 2000fps in our experimental setup using a high-velocity CMOS camera (FASTCAM SA-3, Photron). The segmentation precision was shown to depend on the optical contrast of the observed object and the SNR of the image. Results from the spectra in Fourier space of synthetic images suggest that segmentation precision is mainly limited by the image SNR opening the possibility for future improvements to the experimental setup with newly developed, low-noise scientific CMOS (sCMOS) sensors [37].

The segmentation precision of our algorithm was found to be significantly better than that reported for the previous methods by Usenik [21] and Pécréaux [13], who reported precisions of 25 to 36nm for the position of the cell contour and an estimated 0.1px, respectively. The latter method was directly compared to our algorithm and found to exhibit significant segmentation artifacts, the source of which was concluded to be its partial dependence on the pixel grid. These artifacts were also found to be the reason for the spurious increase in spectral amplitudes in Fourier space towards high wavenumbers, which limited the observable q-range to the first fifty modes in the previous method [13]. This artifact, and associated properties, are not exhibited by the new algorithm, which can be practically extended up to much higher Fourier modes ($n_{max} \approx 1000$), ranging in the microscopic domain ($q_{max} \approx 0.3$ nm^{-1}), where the fluctuation dynamics of the cytoskeleton elements can be probed over nanoscopic distances, $d \approx 2\pi/q_{max} \approx 20$ nm. The robustness of the proposed algorithm is largely superior to previous approaches to cell contour segmentation of microscopy images, in terms not only of transverse accuracy (essential to exactly determine the amplitude of the fluctuations with sub-pixel precision) but also of lateral resolution (crucial to extend spatial Fourier analysis below sub-pixel longitudinal distances). Our algorithm, and its implementation with GPU's, has minimized the number of computational artifacts associated to any sub-pixel interpolation, so under optimal optical performance (high contrast and low camera noise), it is able to work within the instrumental limit, which determines a perfectly detectable and analyzable floor noise. The high precision of our optical microscopy method is comparable to the precision of much more sophisticated methods, such as interferometric microscopy. Our method is complementary, since their interferometric method gives a two-dimensional height map of the cell, while our method yields the membrane displacement at the equatorial plane. Finally, the spatial precision and temporal sampling rate of our method is comparable to those reported for tracking methods with optical traps. However, our method gives us access to the whole cell contour, whereas optical traps are only able to probe single points of the membrane. Finally, the new algorithm was shown to be usable for biological cells with different properties and shapes, including RBC, *E. coli* and lymphocytes as well as biomimetic objects such as giant unilamellar vesicles. It is expected that this method will allow for much more refined results, when applied to RBC and other mechanically compliant cells, which may yield new insights into their cell-mechanics using conventional optical contrast microscopy. The method is sufficiently robust and versatile that is potentially adaptable to other optical modes and microscopy settings.

Materials and preparation methods

Microscope and camera

The experimental setup consisted of a Nikon Ti2000 microscope equipped with a 100 × (NA1.4) bright-field objective followed by additional 1.5 × and 2.25 × optical zooms giving a total 337.5 × zoom. This optical system images onto a Photron FASTCAM SA-3 high-speed digital camera with CMOS sensor capable of recording at 2000fps at full resolution of 1024 × 1024 pixels with a pixel-size of 17μm, yielding an effective pixel-size of (17μm/px)/

337.5 = 50nm/px, which was confirmed experimentally. To reduce memory usage, the required pixel-area was reduced to 256 × 256 pixels, when recording videos of RBCs.

Preparation of dried RBC

RBCs were extract using the finger pricking method from a voluntary, healthy donor. Approximately 20 − 30μL of blood was aspirated using a pipette and immediately diluted inside 1mL solution containing phosphate buffered saline (Sigma-Aldrich, Germany), 5mM glucose (Riedel-de Haen, Seelze, Germany) and 1mg/mL bovine serum albumin (Sigma-Aldrich, Germany). After extraction the blood is washed with a centrifuge (Mikro 120, Hettich, Germany) (5 × 10 min, 2347.8G). After each centrifugation pass the supernatant is removed and the remaining RBCs are resuspended in 1mL of centrifugation buffer. The RBCs were then incubated for 2 h in a fixation buffer with the same composition as the centrifugation buffer with an added 3% of glutaraldehyde (Sigma-Aldrich, Germany). After incubation, the buffer solvent is exchanged with pure water (Milli-Q) after centrifugation (2 × 10 min, 2347.8G) and substitution of the supernatant. Then, 80μL of the resulting RBC-containing solution is spread on a cover-slide and dried in an oven at 40˚C.

Synthetic images were calculated following the method proposed by Usenik [21]. We convolute an image containing a circle consisting of 1's on a background of 0's with two Gaussians of two different widths σ_1 and σ_2. The radius of the circle was chosen to be 61px, in order to approximately corresponds to the radius of RBC as recorded with our experimental setup, since 61px · 50nm/px = 3μm is roughly the radius of RBC. By subtracting the two resulting images and choosing σ_1 and σ_2 accordingly, we obtain an image with a halo at the position of the circle boundary, which is very similar to that of the experimentally obtained images of the RBCs. Finally, white noise with a Gaussian amplitude distribution was added to the image pixels to simulate the image noise.

Signal-to-Noise Ratio of experimentally obtained images is determined by calculating the averaged image intensity $\langle I \rangle$ and its standard deviation σ_I inside a feature-less ROIs *after* the image corrections has been performed. The SNR definition we user here is then given by SNR = $\langle I \rangle / \sigma_I$. Note that this definition is different from the typical logarithmic definition used by Usenik (Usenik et al., 2011), where $\mathrm{SNR}_{\mathrm{Usenik}} = 20 \log[\langle I \rangle / \sigma_I]$.

The authors wish to thank José A. Rodrigo (Department of Optics, UCM) for fruitful discussions on the noise structure factor. We also thank Julia Gala and Victor G. Almendro-Vedia for providing us with microscopy images of lymphocyte and *E. coli* cells. The work was supported in part by Ministerio de Economía y Competitividad (MINECO, Spain) under grant FIS2015-70339-C2-1-R and by Comunidad de Madrid under grant S2013/MIT-2807. MM thanks the FPU program (MEC) for financial support.

Supporting information

S1 File. High-precision flicker spectroscopy contour tracking algorithm: www.github.com/michaelmell/hipfsta.
(PDF)

S2 File. Containing S1 Fig: Distributions of the height fluctuations of glutaraldehyde-fixed RBC in aqueous buffer; S1 Table: RMSD of the distributions shown in S1 Fig; S2 Fig: Comparison of the spectra obtained from the synthetic dataset; S3 Fig: Fluctuation spectra at different SNR values obtained with the algorithm from reference [13].
(PDF)

Acknowledgments

The authors wish to thank José A. Rodrigo (Department of Optics, UCM) for fruitful discussions on the noise structure factor. We also thank Julia Gala and Victor G. Almendro-Vedia for providing us with microscopy images of lymphocyte and *E*. *coli* cells. The work was supported in part by Ministerio de Economía y Competitividad (MINECO, Spain) under grant FIS2015-70339-C2-1-R and by Comunidad de Madrid under grant S2013/MIT-2807. MM thanks the FPU program (MEC) for financial support.

Author Contributions

Formal analysis: Michael Mell, Francisco Monroy.

Funding acquisition: Francisco Monroy.

Investigation: Michael Mell, Francisco Monroy.

Methodology: Michael Mell, Francisco Monroy.

Project administration: Francisco Monroy.

Resources: Francisco Monroy.

Software: Michael Mell.

Supervision: Francisco Monroy.

Validation: Michael Mell, Francisco Monroy.

Writing – original draft: Michael Mell, Francisco Monroy.

Writing – review & editing: Francisco Monroy.

References

1. Quantitative Imaging in Cell Biology, 1st ed. Edited by J. Waters and T. Wittmann, in Methods in Cell Biology series, Volume 123 (Academic Press, 2014).
2. Nombela-Arrieta C., Pivarnik G., Winkel B., Canty K.J., Harley B., Mahoney J.E., Park S-Y., Lu J., Protopopov A. and Silberstein L.E. 2013. Quantitative imaging of haematopoietic stem and progenitor cell localization and hypoxic status in the bone marrow microenvironment. Nature Cell Biol. 15, 533. https://doi.org/10.1038/ncb2730 PMID: 23624405
3. Carpenter A.E. 2007. Software opens the door to quantitative imaging. Nature Methods. 4, 120. https://doi.org/10.1038/nmeth0207-120 PMID: 17264858
4. Smith A. 2016. Biophysics: Alive and twitching. Nature Phys. 12, 378.
5. Rodriguez-Garcia R. I. Lopez-Montero M. Mell G. Egea N.S. Gov and Francisco Monroy. 2015. Direct Cytoskeleton Forces Cause Membrane Softening in Red Blood Cells. Biophys. J 108, 2794–2806. https://doi.org/10.1016/j.bpj.2015.05.005 PMID: 26083919
6. Turlier H., Fedosov D. A., Audoly B., Auth T., Gov N. S., Sykes C., Joanny J.-F., Gompper G. and Betz T. 2016. Equilibrium physics breakdown reveals the active nature of red blood cell flickering. Nature Phys, 12, 513–519 (2016)
7. Yoon Y.Z., H Hong A. Brown D. C. Kim D.J. Kang V. Lew and Cicuta P. 2009. Flickering analysis of erythrocyte mechanical properties: Dependence on oxygenation level, cell shape, and hydration level. Biophys J 97, 1606–1615. https://doi.org/10.1016/j.bpj.2009.06.028 PMID: 19751665
8. Montzel C., Schmidt D., Kleusch C., Kirchenbüchler D., Seifert U., Smith A-S, Sengupta K. and Merkel R. 2015. Measuring fast stochastic displacements of bio-membranes with dynamic optical displacement spectroscopy. Nature Commun 6, 8162
9. Ben-Isaac E., Park Y., Popescu G., Brown F.L.H., Gov N.S., Shokef Y., 2011. Effective Temperature of Red-Blood-Cell Membrane Fluctuations. Phys. Rev. Lett. 106, 238103. https://doi.org/10.1103/PhysRevLett.106.238103 PMID: 21770546
10. Betz T., Lenz M., Joanny J.-F., Sykes C., 2009. ATP-dependent mechanics of red blood cells. Proc. Natl. Acad. Sci. U. S. A. 106, 15320–5. https://doi.org/10.1073/pnas.0904614106 PMID: 19717437

11. Park Y., Best C. a., Kuriabova T., Henle M.L., Feld M.S., Levine A.J., Popescu G., 2011. Measurement of the nonlinear elasticity of red blood cell membranes. Phys. Rev. E 83, 051925. https://doi.org/10.1103/PhysRevE.83.051925 PMID: 21728589

12. Park Y., Best C.A., Auth T., Gov N.S., Safran S.A., Popescu G., Suresh S., Feld M.S., 2010a. Metabolic remodeling of the human red blood cell membrane. Proc. Natl. Acad. Sci. 107, 1289–1294. https://doi.org/10.1073/pnas.0910785107 PMID: 20080583

13. Pécréaux J., Döbereiner H.G., Prost J., Joanny J.F., Bassereau P., 2004. Refined contour analysis of giant unilamellar vesicles. Eur. Phys. J. E 13, 277–290. https://doi.org/10.1140/epje/i2004-10001-9 PMID: 15103522

14. Engelhardt H., Duwe H.P.P., Sack mann E., 1985. Bilayer bending elasticity measured by Fourier analysis of thermally excited surface undulations of flaccid vesicles. J. Phys. Lettres 46, 395–400. https://doi.org/10.1051/jphyslet:01985004608039500

15. Faucon J.F., Mitov M.D., Méléard P., Bivas I., Bothorel P., Meleard P., Mitov M. D., and P J.F.F.B., 1989. Bending elasticity and thermal fluctuations of lipid membranes. Theoretical and experimental requirements. J. Phys. 50, 2389–2414. https://doi.org/10.1051/jphys:0198900500170238900

16. Duwe H.P., Kaes J., Sackmann E., Bending E.S., 1990. Bending elastic moduli of lipid bilayers: modulation by solutes 51, 945–962

17. Strey H., Peterson M.A., Sackmann E., 1995. Measurement of erythrocyte membrane elasticity by flicker eigenmode decomposition. Biophys. J. 69, 478–488. https://doi.org/10.1016/S0006-3495(95)79921-0 PMID: 8527662

18. Döbereiner H.-G., Evans E., Kraus M., Seifert U., Wortis M., 1997. Mapping vesicle shapes into the phase diagram: A comparison of experiment and theory. Phys. Rev. E 55, 4458–4474. https://doi.org/10.1103/PhysRevE.55.4458

19. Häckl W., Seifert U., Sackmann E., 1997. Effects of fully and partially solubilized amphiphiles on bilayer bending stiffness and temperature dependence of the effective tension of giant vesicles. J. Phys. II 7, 1141–1157.

20. Evans J.J., 2008. An image-processing technique for measuring the dynamic movement of cell membranes. Comput. Biol. Med. 38, 545–54. https://doi.org/10.1016/j.compbiomed.2008.01.013 PMID: 18358462

21. Usenik P., Vrtovec T., Pernuš F., Likar B., 2011. Automated imaging and analysis of phospholipid vesicle contours in phase contrast microscopy images. Med. Biol. Eng. Comput. 49, 957–66. https://doi.org/10.1007/s11517-011-0789-0 PMID: 21710319

22. Park Y., Popescu G., Badizadegan K., 2006. Diffraction phase and fluorescence microscopy. Opt. Express 14, 8263–8. https://doi.org/10.1364/OE.14.008263 PMID: 19529201

23. Kim M.K. 2010. Principles and techniques of digital holographic microscopy. SPIE Rev. 1. 018005.

24. Zernike F. 1955. How I discovered phase contrast. Science 121, 345–9. PMID: 13237991

25. Nguyen T.N., Edwards C., Goddard L.L. and Popescu G. 2016. Quantitative phase imaging of weakly scattering objects using partially coherent illumination. Opt. Express 24, 11683–93. https://doi.org/10.1364/OE.24.011683 PMID: 27410094

26. Kandel M.A., Fanous M., Best-Popescu C. and Popescu G. 2018. Real-time halo correction in phase contrast imaging. Biomed. Opt. Express 9, 623–35. https://doi.org/10.1364/BOE.9.000623 PMID: 29552399

27. Méléard P., Gerbeaud C., Pott T., Fernandez-Puente L., Bivas I., Mitov M.D., Dufourcq J., Bothorel P., 1997. Bending elasticities of model membranes: influences of temperature and sterol content. Biophys. J. 72, 2616–29. https://doi.org/10.1016/S0006-3495(97)78905-7 PMID: 9168037

28. Wilson T., Sheppard C.J.R.J.R., 1981. The halo effect of image processing by spatial frequency filtering. Optik (Stuttg). 59, 19–23.

29. Boal D.H. Mechanics of the Cell, 2nd ed. (Cambridge Univ Press, Cambridge, 2012)

30. Inoué S. and Spring K.R. Videomicroscopy: The Fundamentals, 2nd ed (Plenum Press, New York, 1997)

31. Wilson T. and Sheppard C. J. R. 1981. The Halo Effect of Image-Processing by Spatial-Frequency Filtering. Optik 59, 19–23.

32. Zernike F. 1942. Phase contrast, a new method for the microscopic observation of transparent objects. Part I. Physica 9, 686–698.

33. Popescu G., Ikeda T., Dasari R., Feld M., 2006. Diffraction phase microscopy for quantifying cell structure and dynamics. Opt. Lett. 31, 775–777. https://doi.org/10.1364/OL.31.000775 PMID: 16544620

34. Park Y., Best C.A., Badizadegan K., Dasari R.R., Feld M.S., Kuriabova T., Henle M.L., Levine A.J., Popescu G., 2010b. Measurement of red blood cell mechanics during morphological changes. Proc. Natl. Acad. Sci. U. S. A. 107, 6731–6736. https://doi.org/10.1073/pnas.0909533107 PMID: 20351261

35. Smith, J.O. 2010. Linear interpolation frequency response in Phys Audio Sign Process. https://ccrma.stanford.edu/~jos/pasp04/Linear_Interpolation_Frequency_Response.html

36. Guillemin, E.A., 2013. Theory of linear physical systems: theory of physical systems from the viewpoint of classical dynamics, including Fourier methods. Courier Corporation.

37. Fowler B., Liu C., Mims S., Balicki J., Li W., Do H., Appelbaum J., Vu P., 2010. A 5.5Mpixel 100 frames/sec wide dynamic range low noise CMOS image sensor for scientific applications. Proc. SPIE. https://doi.org/10.1117/12.846975