Rapid Time Resolved Analysis of Filament Motility

G. Ijpma; Z. Balassy; A-M. Lauzon

ABSTRACT In vitro motility assays are a valuable tool in understanding and characterizing motor protein mechanics, but existing algorithms are not optimized for time resolved analyses. We propose an algorithm that combines trace detection with time resolved analysis. By tracking filament ends, we double the number of data points and minimize data loss from crossing filaments. A movement trace formed by each filament end is stored such that the frame number in which a filament either first (filament tip) or last (filament tail) occupies a certain pixel, is stored. A frame number vs distance curve is generated from this trace, which is segmented into regions by slope to detect stop-and-go movement. We generated mock motility videos with known velocities, motile fraction profiles and varying filament lengths. Our algorithm accurately detected velocity and motile fraction changes for velocities less than 0.05 pixels per frame, without manual trace dropping and regardless of filament crossings. Two established algorithms, with manual trace dropping of unrealistic or not analyzable traces, showed much greater error in average and individual filament velocity as well as motile fraction. To demonstrate the utility of the algorithm we tested two actual motility experiments: 1) Adenosine triphosphate (ATP) added to skeletal myosin in rigor state; 2) myosin light chain phosphatase (MLCP) added to phasic smooth muscle myosin. Our algorithm revealed features not previously detectable: 1) a near instantaneous increase in motile fraction paralleled by a slow increase in velocity as the ATP concentration increases and 2) simultaneous reductions in velocity and motile fraction as MLCP diffuses into the motility chamber at very low velocities. Our tests show that the developed algorithm surpasses existing algorithms in time resolved measurements of motile fraction and velocity at a wide range of filament velocities and lengths, with minimal user input and CPU time.

Introduction

In vitro motility or gliding assays are essential tools for quantifying mechanical parameters of the interaction of motor molecules with propelled filaments. A number of (semi)-automated algorithms have been developed to track filaments as they glide over the motility surface (3, 5-8, 10-12, 15, 16). Most of these algorithms rely on identification of the filament in each frame by converting a grey-scale image of fluorescently labelled filaments to a black and-white image with either a manually or automatically set grey value as threshold between black and white. Areas of connected white pixels outline the approximate location of each filament. Some algorithms calculate the centroid of each white area and a velocity is calculated from the movement of the centroid location frame-to-frame, either with very low frame rates or considerable smoothing of the data with a moving average filter (5-8, 11, 16). This method does not distinguish between directed (gliding) or undirected (Brownian noise, circling around a fixed point, *etc*.) motion. Furthermore, sudden changes in lighting or focus can result in a change in size of the white area without an actual change in the filament position or size, resulting in detected false motion. Filaments crossing confuses the algorithms, so any crossing traces are either removed by hand (5, 10, 16) or with an automated algorithm (6). More advanced algorithms can track filaments despite crossing but require some assumptions on filament rigidity (near straight line movement) (2, 8). Lastly, any curved motion results in underestimation as the centroid of a curved filament follows a shorter track than any point of the filament, as it essentially cuts corners. Some of these issues (cutting corners, lighting and focus changes) can be countered by not taking the centroid, but instead taking a centre point on the filament (16).

More advanced software uses the threshold technique to obtain the approximate location of the filament, followed by identification of a more exact location of the filament by fitting segments to a model (12). In theory this gives a much more accurate position estimation for the filament, it allows for analysis of crossing filaments and can provide detailed information about filament shape (curvature, length) and degree of directed vs. non-directed motion. However, this method is CPU intensive and it is unclear what the effect of noise is on the position accuracy. Fully automated filament detection through segmentation has been applied to single images of filament and filament like structures ((17) among others), but the CPU intensiveness of this approach has prevented it from being useful for motility studies.

To reduce the effect of non-directed motion on calculated filament sliding velocities, one can look at the traces generated over multiple frames by each filament (3-5, 15). The trace length minus the filament length indicates the total moved distance by the filament and is indicative of filament velocity. This averaged velocity does not distinguish between stop-and-go movement and slow movement, nor does it allow for time resolved analyses. We previously proposed a combined algorithm using the above described centroid method to determine motile fraction for each filament, i.e. the fraction of time the velocity of movement was above a certain threshold value and combined this with the trace method to determine the velocity of the filament when moving (4). However, when studying slow moving motility from actin propelled by slow motor proteins, directional motion is often less then non-directional motion on a frame-to-frame basis,

particularly for small filaments. Furthermore, this method does not provide an accurate time-resolved analysis.

Here we propose a novel approach to filament tracking which maps the motion of filament ends by storing a trace of the last (tail) and first (tip) frame in which a filament was present in each location. This allows for rapid analysis of actual time resolved filament velocities, minimal rejection from filament crossings, natural filtering of non-directional motion, all of this regardless of the curvature of the filament and filament path.

Methods

Motility preparation

All motility videos were taken from a parallel study (1). Actin was purified from chicken pectoralis acetone powder (9). Skeletal muscle myosin protein was purified from chicken pectoralis muscle (13) and phasic smooth muscle myosin was purified from chicken gizzard (13). Chicken gizzard and pectoralis were obtained from a local slaughterhouse. Myosin light chain phosphatase (MLCP) and myosin light chain kinase (MLCK) were purified from turkey gizzard (14) (gift from A. Sobieszek). Phasic smooth muscle myosin was phosphorylated prior to the motility study for 20 min at room temperature using 5.02mg/ml myosin, 4mM $CaCl_2$, 12.5 mM $MgCl_2$, 14.7 mM imidazole, 0.59 mM EGTA, 18mM DTT, 3.75 μM calmodulin, 5mM ATP and 0.07 μM MLCK. KCl and glycerol were added to final concentrations of 0.3M and 50% respectively. Actin was labelled with tetramethylrhodamine isothiocyanate-phalloidin (Sigma #P1951). Myosin, actin, and motility buffers are described in supplemental methods.

We used an inverted microscope (Eclipse Ti, Nikon) to observe the labelled actin with a 100x magnification oil immersion objective (Nikon CFI Plan Fluor 100X Oil), and recorded with an EM-CCD camera (Hitachi KP-E500) at 30 frames s^{-1} and a Matrox frame grabber (Matrox Mor 2VD-84). An X-Cite 120Q excitation light source was used. To maintain a temperature of 30º C in the flow through chamber an objective heater (Bioptechs) and heating plate slide holder (Chamlide TC) were used. Flow through chambers consisted of a nitrocellulose coated coverslip with two strips of double-sided scotch tape on which a plastic sheet was positioned (Artus Corp, 0.125mm plastic shim) with a 2mm diameter small hole in the centre. A polycarbonate membrane (0.8μm pore size, Millipore Sigma #ATTP02500) was UV glued to the plastic, covering the hole. Injection of ATP or MLCP (3.5 μM) dissolved in motility buffer onto the membrane resulted in convection free diffusion into the flow-through chamber.

Motility assays were performed as described in (4).

Algorithm

Apart from parameter settings, the entire algorithm described below does not require any user input. For a wide range of motility videos recorded under different conditions, on two different set-ups and with 40 fold differences in filament velocity and motile fraction we were able to use a single set of parameter settings. The software is written in Matlab (The MathWorks, Natick, MA) and published under a GNU General Public Licence v3, at https://github.com/Gijpma/Motility-of-Filament-ends

Contrast enhancement and segmentation

The average intensity of each pixel of a set number of consecutive video frames is calculated to reduce thermal noise from the sensor. The number of frames that need to be averaged depends on the noise level. If peak noise grey values are similar to average gray values of the filaments, then frame averaging is required to improve the signal to noise ratio. For all data shown here we found averaging of 2 frames, sufficient. A 2D Gaussian smoothing filter with a standard deviation of 1 is applied to the resulting image to further smooth out thermal noise in the filament images. The image with a much stronger 2D gaussian smoothing (standard deviation of 20) is a good approximation of the amount of lighting each area of the image receives. To correct for uneven lighting, we subtract this heavily blurred image from the filament image, after which the resulting pixel intensities are scaled to fully use the dynamic range available in an 8-bit greyscale image. The image is then converted to black-and-white by setting to a value of 1 all pixels above a threshold (i.e. white). This threshold value is determined by analyzing the number of isolated white areas identified in the first frame for a range of threshold values (see Fig. 1a). For very low threshold values the entire image is connected, resulting in a single identified connected component, while gradually increasing the threshold value rapidly increases the number of identified components as noise pixels are isolated until a maximum is achieved (P1 in Fig. 1A). A further increase in threshold value decreases the number of detected noise pixels and thus connected components, until only the filaments remain. After this point the number of components either increases as the filaments break up into individual islands followed by a decrease to zero, or only a gradual decrease to zero is found if filament fluorescence intensity is inconsistent. We aimed to set the threshold value automatically just at the point at which only filaments are detected. We cannot automatically detect at which point no noise is detected, but only very small changes in the threshold value result in large decreases in the number of detected components when noise is still present. We therefore set the threshold (P3 in Fig. 1A) a fraction beyond the point at which the number of components detected is low, but most likely more than the number of filaments present (here 300, P2 in Fig. 1A). This method worked consistently in detecting filaments without spurious noise issues for over 300 videos recorded on two different motility set-ups and variable lighting conditions and motility quality.

Endpoint identification

After taking a threshold image, all connected components were skeletonized by repeated morphological thinning, i.e. removal of the perimeter pixels until a single pixel width line remains (Fig. 1B). The endpoints and branchpoints of this skeleton image are identified. When an endpoint is in close proximity to a branchpoint, the endpoint is removed. This occurs when two filaments have crossed, but the tip of one of the filaments is close to the body of the other filament. These points need to be removed to avoid analysis errors. Two filament ends in close proximity that do not belong to the same filament are also removed, while two filament ends in close proximity belonging to a single filament are analyzed as a single point by taking the average location of both endpoints. Subsequently endpoints are cross-correlated between frames based on proximity only.

Frame-to-Frame analysis

Once an endpoint is identified from the skeleton image, a square section of the threshold image around the filament end large enough to capture the entire filament end is stored, with all white pixels given the current frame number, and all black pixels the value zero. To help in identification of the direction of movement, any white pixels at the edge of the square image, indicating the presence of the filament body on that edge, are labelled the current frame number plus a fraction between 0 and 1. After all frames are read, two movement traces for each filament end are generated: 1) taking the highest frame value stored for each pixel visited by the filament end; 2) taking the lowest frame value stored for each pixel. The aim is to track the progressive movement of the filament tip edge. A filament end that is moving ahead of the filament body (a filament tip) will show movement of the filament tip in trace 2 (Fig.1C), as the filament tip edge is the first to move over a pixel. However, a filament end that is dragged behind the filament body (the filament tail) has a tail edge that is visiting each pixel last as shown in trace 1 (Fig.1C). We can identify the correct movement direction by finding the portion of pixels in the trace with fractional pixel value, indicating that rather than the tip or tail edge, the filament body presence was stored at these pixels. The trace with the lowest portion of fractional pixel values is taken as the correct direction of movement. If part of a trace indicates a different movement direction, this indicates "tip-overtake": two filament ends move exactly over each other in opposite direction, confusing the algorithm into identifying this as a single tip movement. These traces are rejected and not analysed.

Trace analysis

The trace is reduced to a skeletal representation (single pixel width curve) by repeated morphological thinning (Fig. 1D, red line). The resulting skeleton of the trace is pruned to remove any branches caused by off-track motion, image noise or a variation in lighting and focus (for details see supplementary methods). The resulting 2d curve is smoothed (moving average filter on x and y coordinates) to subpixel coordinates. This is necessary as the total path length is exaggerated by the restriction to whole pixel movement. Intensity values of the sub-pixel 2d curve coordinates are subsequently calculated by interpolation.

To derive a curve of distance vs. frame number of the filament end, we calculated the cumulative distance over the trace starting at the end with lowest intensity. Vertical (same intensity) segments caused by the filament start (tip trace) and end (tail trace) location are removed, resulting in a monotone distance vs. frame number curve (Fig.1E).

Curve segmentation

Because of the coarseness of the distance vs. frame number curve, particularly for low velocity motility (<10 pixels s^{-1}), smoothing of the data is needed before velocities are calculated. However, motility assays often show stop-and-go motion, believed to be caused by dead motor heads. Smoothing blends stopped and moving segments of the curve, creating artificially low moving speeds and shortened stopped sections. Here we first segment the curve into stopped and moving regions and apply smoothing only on the moving segments of the curve. The segmentation is done using the Ramer-Douglas-Peucker algorithm, which reduces the curve to segments in which none of the points on the original curve deviate further from the segment lines

than a given error value (Fig.1F). For each segment, the velocity is calculated as the slope of the segment line, and any velocity below a given threshold value is set to zero. Adjacent moving sections are merged, as well as adjacent stopped sections, after which the moving segments are smoothed with a moving average filter (Fig.1G). An average velocity of all moving filaments is calculated as well as the instantaneous motile fraction, i.e. the proportion of filaments that is moving at any point in time.

Mock motility video generation

We developed a Matlab algorithm for the creation of mock motility videos with known velocity and motile fraction profiles over time, allowing for the evaluation of the accuracy of the proposed method and comparison with existing methods in the literature. To accurately represent filament motility videos, the video generation software allows for control of velocity, motile fraction (including stop-and-go movement), variable distribution of filament size and curvature as well as various modes of background noise. For a detailed description of the methods used, see supplementary methods.

Mock videos

We tested several mock videos, gradually increasing the level of complexity (see Fig 2):

1. Curved filaments; no noise;
2. Curved filaments; noise;
3. Long, overlapping Curved filaments; noise

For each of these 3 scenarios, 3 velocity regimes were created:

a) Variable velocity (changing from 5 to 1 μm s^{-1})
b) Variable velocity with stop-and-go movement (velocity as a, motile fraction 0.5).
c) Low variable velocity with variable motile fraction (velocity changing from 1 to 0.2 μm s^{-1}, motile fraction as in b

The generated videos were sampled at 30 frames s^{-1}, with 25 generated filaments in each frame and changes in velocity and motile fraction were applied between 300 and 500 frames (see figure 2). We calculated root mean squared (RMS) error from the measured difference in velocity of each filament relative to the input velocity of the mock motility video at each frame.

Results

Mock Video Data

We tested the performance of the proposed Motility of Filament Ends (MFE) algorithm on 9 mock motility videos of 1000 frames each. On a 4-core processor using parallel processing of the videos this took 205s. Figures 2A-D show that the RMS noise of the individual filament traces is slightly increased for low (Fig. 2B) and very low velocities (Fig. 2D). Note that the RMS error measures individual filament error, while the velocity values in figure 2 are an average of all detected filaments in the video. Consequently, a larger RMS error does not necessarily correspond with a greater error in the average velocity reading shown here. The introduction of stop and go movement does not increase the noise (Fig. 2C). At very low velocities the motile fraction is underestimated towards the end of the video. Introduction of noise only slightly increases the noise of the analysis as shown in figures 2F-H. Increasing filament length to the

point of extensive filament overlap did not decrease the accuracy of the analysis substantially (Fig 2I-L).

The mock video tests show that the algorithm is capable of accurately detecting a wide range of velocities down to movements less than 1 pixel per 15 frames (0.2 μm s^{-1}, 0.124 μm per pixel). With a cut-off velocity for moving vs. stopped filaments set at 0.15 μm s^{-1}, some filament ends that are detected only towards the end of the video can be mistakenly identified as stopped, resulting in the drop off in motile fraction in Figs. 2D, H and L. If stop and go motion occurs with relatively short moving or stopping times the analysis may identify the motile state of filament ends incorrectly as well. Most variability in individual filament end velocity occurs towards the start and end of the filament end's trace as the tip and tail of a skeletonized trace are most sensitive to noise.

Comparison with existing algorithms

We subsequently compared the performance of the MFE algorithm with two existing algorithms on either side of the complexity spectrum: a filament centroid tracking algorithm previously developed at our lab (4) and a filament fitting algorithm (FIESTA, (12)). We previously developed an algorithm based on the motion of filament centroids, which can do both manual trace dropping for overlapping filaments, or an automated procedure based on machine learning (4). Here we used its manual trace dropping feature for the cleanest results. The FIESTA algorithm aims to find filament locations at subpixel accuracy by dividing detected filaments into small sections containing a filament tip, a filament body or a crossing filament and then fitting models of these features (12). For a specific set of conditions, the authors reported location tracking accuracies on the order of 10 nm. The FIESTA software has the benefit of being able to handle crossing filaments as well as the subpixel accuracy, at the cost of a much greater computation time. However, in tracking the velocity of actual moving filaments the authors showed velocity variability of ~250 nm/s. As initial results showed very large position variability in a subset of filaments in the FIESTA software, likely because of incorrectly identified filaments, we also applied here a manual trace dropping to clean the data. Run time for 3 videos of 1000 frames each was 1650s. Any trace dropping in the proposed MFE algorithm was fully automated and based on criteria specified in the methods. To ensure equal comparisons, the same cut-off velocity to distinguish between stopped and moving filaments (0.15 μm s^{-1}) in the MFE algorithm was used for the FIESTA and centroids tracking software as well.

Neither the FIESTA nor the Centroids tracking algorithm were able to detect movement in the long crossing filament videos used for figure 2I-L. Figure 3 shows that for mock videos of curved filaments with noise for the same three velocity and motile fraction scenarios as in figure 2 E-H, the RMS error for the MFE algorithm is considerably less than both the FIESTA and centroids tracking algorithms for all three videos. Also, the error in both the centroids and FIESTA algorithms results in a reduction in the measured average velocity. Note that we introduced a 20 frame moving average smoothing to the output of the Centroids and Fiesta algorithms to allow a fair comparison with the MFE algorithm which has a similar smoothing built in. The data also show that the introduction of stop and go movement greatly decreases the accuracy of the velocity signal of both the FIESTA and the centroid tracking algorithms, as

stopped filaments can have an occasional apparent velocity above the set cut-off value because of the video noise. A similar issue arises for low velocities (Fig. 3C, F and I) where particularly the FIESTA algorithm results in unstable velocity estimations. At very low velocities the MFE algorithm estimates motile fraction much more accurately then both the centroids and FIESTA algorithms. The FIESTA algorithm does show an improved accuracy of the motile fraction estimation at the end of the video during slow motility as the distinction of stopped vs moving filaments is made instantaneously rather than over a timespan requiring a minimal movement of one pixel.

Application to actin motility data

We tested the MFE algorithm on 2 sets of actual data with changing velocity and motile fraction profile. These videos are part of a future publication for which the current algorithm was developed. In the first video we injected MLCP into a motility assay of phasic smooth muscle myosin propelling actin filaments. As this myosin is relatively slow (0.5 μm s^{-1} starting velocity) the cut-off velocity was set lower to 0.03 μm s^{-1} to ensure accurate detection of any velocity decreases expected to occur in response to dephosphorylation of the regulatory light chain of the myosin heads. The video contained sudden focus changes and changes in illumination occurring around the time of injection of MLCP, which did not affect the filament resolution by the algorithm. The MFE software shows a gradual decrease in motile fraction in parallel with a gradual decrease in velocity, with good filament resolution (Fig. 4A). The very low velocities combined with stop and go motion are unlikely to be detected by existing algorithms based on frame-to-frame or trace analysis. In a second video we injected ATP into an actin motility assay with skeletal myosin. As the actin is originally in a rigor state, the filaments have a motile fraction of zero and a velocity of zero. As the ATP enters the motility chamber through a diffusion membrane (see methods), the actin filaments gradually start moving, increasing both the motile fraction and the velocity. Our data clearly show that motile fraction increases rapidly to a plateau level while velocity increases much more gradually, taking over 30 s to plateau (Fig. 4B top and middle). This indicates that cross-bridge cycling is rate-limited by the amount of available ATP, but that the diffusion of ATP is uniform across the field of view. The video shows rapid break-up of filaments, which is likely caused by excess of dead myosin heads. To assure no ATP remained in the flow cell at the start of the recording, we omitted the usual flush with unlabelled globular actin in the presence of ATP. This flush is meant to reduce filament stoppage and breakage as the actin binds to both live and dead myosin heads, but can’t detach from the dead myosin heads, resulting in blocking of these dead heads. The algorithm detected the change in average length (Fig. 4B bottom).

Limitations

While the proposed algorithm performs well in most situations, under certain conditions filament movement could be misidentified, or filament ends are not properly detected.

- During stop-and-go movement the ability of the algorithm to detect stopped periods is dependent on the cut-off velocity, the image resolution and video noise. As a result, average velocities may be underestimated during rapid stop-and-go turnover.

- The automated threshold detection algorithm uses only the first frame, with a correction for overall illumination changes throughout the video. If the first frame is not representative of the remainder of the video (primarily if the focus is different) the algorithm will have difficulties.
- While non-directive motion is mostly filtered out by the trace detection approach, slow, non-directive motion because of unattached filament ends may be detected as directed motion, particularly for short traces.

Conclusions

We showed that the MFE algorithm is capable of accurate tracking of variable velocity motility assays with a high time resolution and superior accuracy to two of the leading algorithms for tracking filament motility. It does this robustly with minimal complexity, user input and CPU time. The focus on filament ends instead of entire filaments also allows for the unique ability to detect filament motion in assays with long, overlapping filaments. The history retaining method of filament end trace detection combines the benefits of trace-based analyses (detection of directed motion, low velocities detection) with the benefits of frame-to-frame filament location analyses (time resolution).

Author contributions

GI: Conceived study, developed and tested algorithms, wrote manuscript. ZB: performed the experiments, tested algorithm, reviewed manuscript. AL: supervised project, edited manuscript.

References

1. **Balassy Z, Kachmar L, Ijpma G, and Lauzon A-M**. Can We Reproduce the Latch-State in Vitro at the Molecular Level? *Biophysical Journal* 114: 320a, 2018.

2. **Butt T, Mufti T, Humayun A, Rosenthal PB, Khan S, Khan S, and Molloy JE**. Myosin Motors Drive Long Range Alignment of Actin Filaments. *Journal of Biological Chemistry* 285: 4964-4974, 2010.

3. **Hamelink W, Zegers J, Treijtel B, and Blange T**. Path reconstruction as a tool for actin filament speed determination in the in vitro motility assay. *Anal Biochem* 273: 12-19, 1999.

4. **Hilbert L, Cumarasamy S, Zitouni Nedjma B, Mackey Michael C, and Lauzon A-M**. The Kinetics of Mechanically Coupled Myosins Exhibit Group Size-Dependent Regimes. *Biophysical Journal* 105: 1466-1474, 2013.

5. **Homsher E, Wang F, and Sellers JR**. Factors affecting movement of F-actin filaments propelled by skeletal muscle heavy meromyosin. *Am J Physiol* 262: C714-723, 1992.

6. **Månsson A, and Tågerud S**. Multivariate statistics in analysis of data from the in vitro motility assay. *Anal Biochem* 314: 281-293, 2003.

7. **Marston SB, Fraser ID, Bing W, and Roper G**. A simple method for automatic tracking of actin filaments in the motility assay. *J Muscle Res Cell Motil* 17: 497-506, 1996.

8. **Mashanov GI, and Molloy JE**. Automatic Detection of Single Fluorophores in Live Cells. *Biophysical Journal* 92: 2199-2211, 2007.

9. **Pardee JD, and Aspudich J**. [18] Purification of muscle actin. In: *Methods Enzymol*Academic Press, 1982, p. 164-181.

10. **Roman HN, Zitouni NB, Kachmar L, Ijpma G, Hilbert L, Matusovsky O, Benedetti A, Sobieszek A, and Lauzon A-M**. Unphosphorylated calponin enhances the binding force of unphosphorylated myosin to actin. *Biochimica et Biophysica Acta (BBA) - General Subjects* 1830: 4634-4641, 2013.

11. **Root DD, and Wang K**. Calmodulin-Sensitive Interaction of Human Nebulin Fragments with Actin and Myosin. *Biochemistry* 33: 12581-12591, 1994.

12. **Ruhnow F, Zwicker D, and Diez S**. Tracking Single Particles and Elongated Filaments with Nanometer Precision. *Biophysical Journal* 100: 2820-2828, 2011.

13. **Sobieszek A**. Smooth Muscle Myosin: Molecule Conformation, Filament Assembly and Associated Regulatory Enzymes. In: *Airways Smooth Muscle: Biochemical Control of Contraction and Relaxation Respiratory Pharmacology and Pharmacotherapy*, edited by Raeburn D. GMA. Birkhäuser, Basel: 1994.

14. **Sobieszek A, Borkowski J, and Babiychuk VS**. Purification and characterization of a smooth muscle myosin light chain kinase-phosphatase complex. *Journal of Biological Chemistry* 272: 7034-7041, 1997.

15. **von Wegner F, Ober T, Weber C, Schurmann S, Winter R, Friedrich O, Fink RH, and Vogel M**. Velocity distributions of single F-actin trajectories from a fluorescence image series using trajectory reconstruction and optical flow mapping. *J Biomed Opt* 13: 054018, 2008.

16. **Work SS, and Warshaw DM**. Computer-assisted tracking of actin filament motility. *Anal Biochem* 202: 275-285, 1992.

17. **Xiao X, Geyer VF, Bowne-Anderson H, Howard J, and Sbalzarini IF**. Automatic optimal filament segmentation with sub-pixel accuracy using generalized linear models and B-spline level-sets. *Medical Image Analysis* 32: 157-172, 2016.

Figure Legends

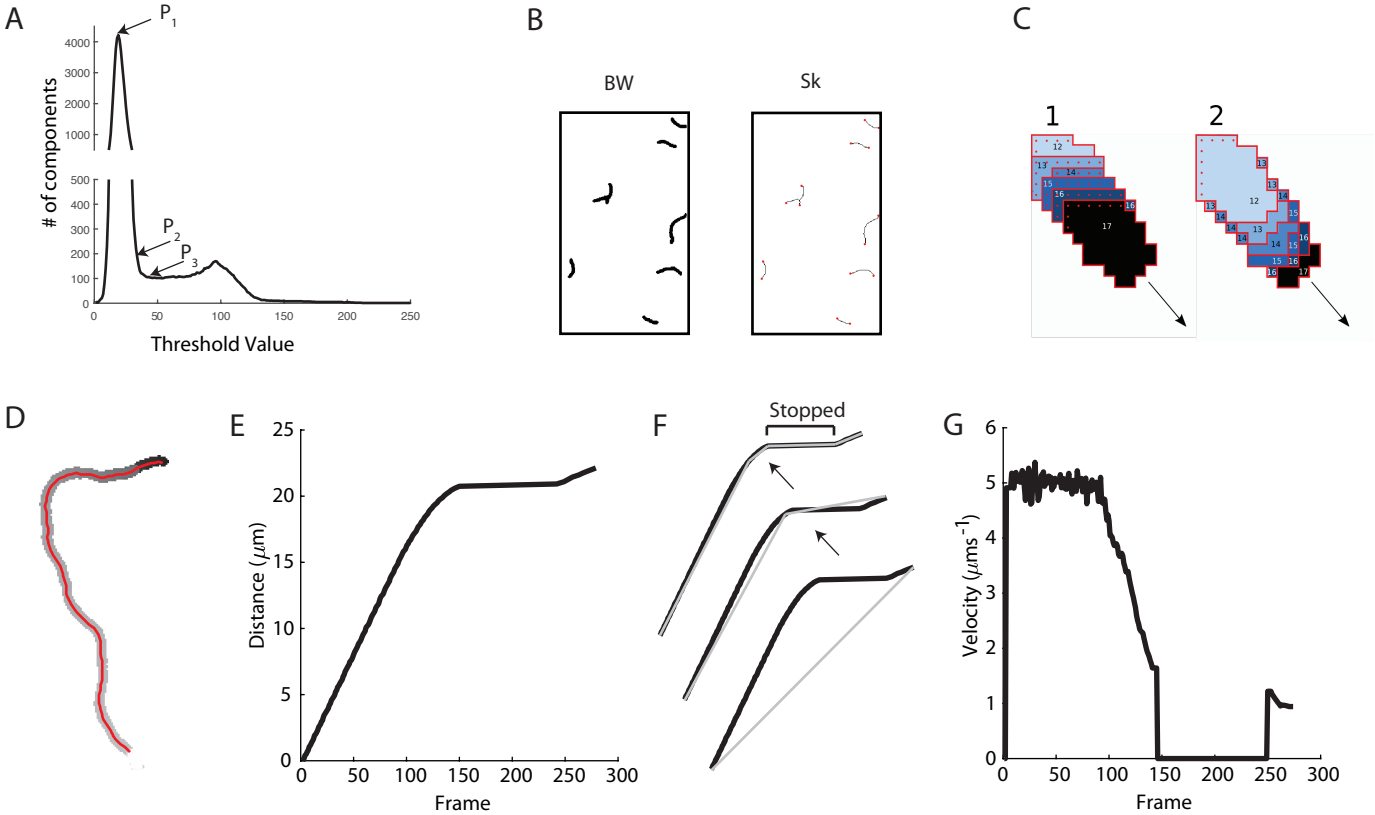
Figure 1: Motility of Filament Ends (MFE) algorithm details. A) sample curve showing the number of components detected for all threshold values. P_1 indicates maximum number of elements detected,P_2 indicates point at which <300 elements are detected and P_3 indicates the calculated threshold. B) left side shows inverted black and white (BW) image after thresholding with value derived from A. Right side shows the same image morphologically thinned filaments (Sk) with red markers at filament ends. C) Tail (1) and tip(2) trace images of the same filament end. In this sample the filament end was first observed in frame 12, and the trace is shown until frame 17. Arrows show actual direction of movement of the filament end, indicating the filament end is a filament tip. Red dots label the edge of the square image around the filament end. The large number of red dots in trace 1 compared to trace 2 indicates that trace 2 has the right orientation of movement. D) Sample of a filament end trace, with morphologically thinned effective centreline in blue. E) Frame vs. Distance calculated from filament end trace. A steep slope indicates fast movement, while a horizontal line segment indicates a stopped filament. F) Ramer-Douglas-Peucker algorithm for segmentation of the Frame-Distance trace. This algorithm segments the trace by starting as a linear connection between the start and end points and repeatedly breaking this line into sections connecting the most distant points on the curve from the line in the previous iteration. G) Resulting velocity vs. frame trace.

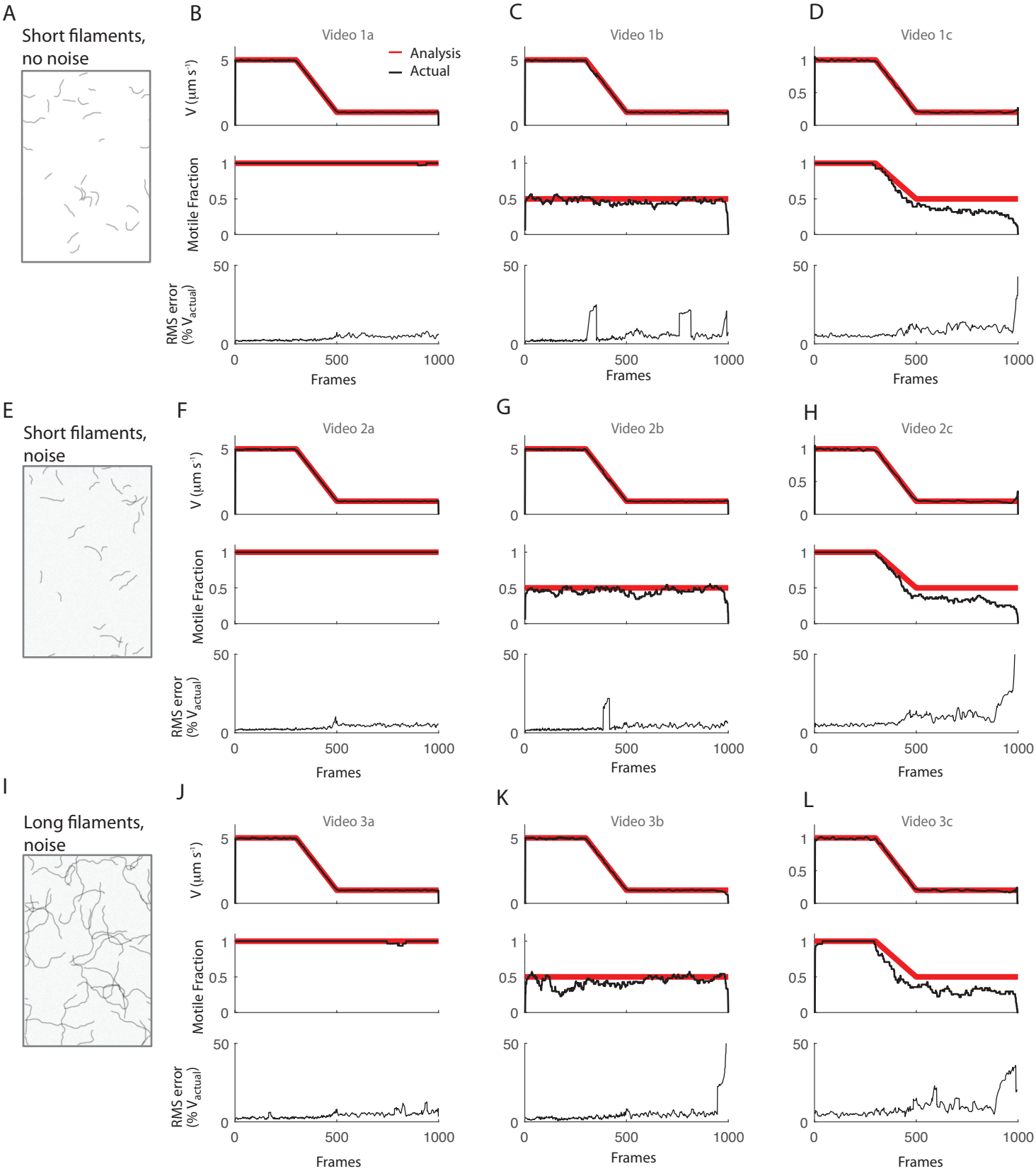
Figure 2: Testing of the MFE algorithm with several mock filament motility videos. A) inverted single frame of mock video used, containing curved filaments with no image noise (video 1a-c, see text). B), C) and D) show the estimated (black) and actual (red) velocity profile (top), measured and actual motile fraction (middle) and fractional RMS error of individual traces in the velocity (bottom) for three different velocity and motile fraction profiles as indicated by the video name (see text). E) inverted single frame of mock video with curved filaments as in A) but with image noise (gaussian speckle, video 2a-c). F), G), and H) velocity, motile fraction and RMS error data for same velocity and motile fraction profiles as in B),C) and D). I) inverted single frame of mock video with long curved filaments with extensive filament overlap and noise (video 3a-c). J), K) and L) show velocity, motile fraction and RMS error as in B),C) and D). Image noise and extensive filament overlap do not affect RMS error of the individual traces nor the average velocity. RMS error is increased for very low velocities, and motile fraction is underestimated at these velocities.

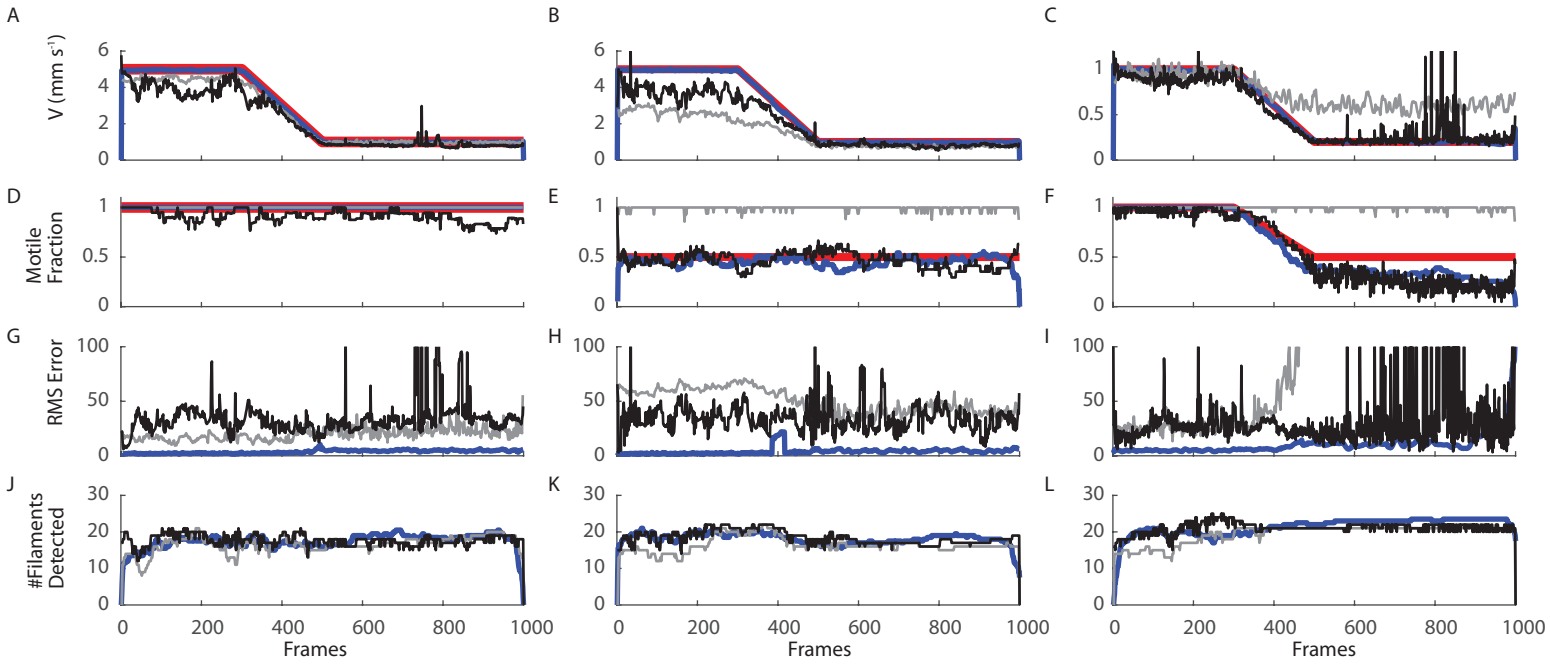
Figure 3: Comparison with Fiesta and Centroid tracking algorithms for mock videos of figure 2 F-H. Fiesta and Centroids tracking algorithm used manual trace dropping of erroneous traces (not correctly identified crossing traces or picked up image noise). A), B) and C) Velocity traces of the three algorithms and the actual velocity of the mock videos used. D), E) and F) Motile fraction for the three algorithms and actual motile fraction. Centroids tracking overestimates motile fraction using the 0.15 $\mu m\ s^{-1}$ cutoff velocity used for all three algorithms. Fiesta and MFE both measure motile fraction accurately except for very low velocities. G), H), and I) fractional RMS error of the individual filament traces. Error is consistently much lower for the MFE algorithm. Note that the centroids algorithm gives a minimum velocity of ~0.5 $\mu m\ s^{-1}$ as a result of image noise, and thus the RMS error at an actual average velocity of 0.2 $\mu m\ s^{-1}$ in figure

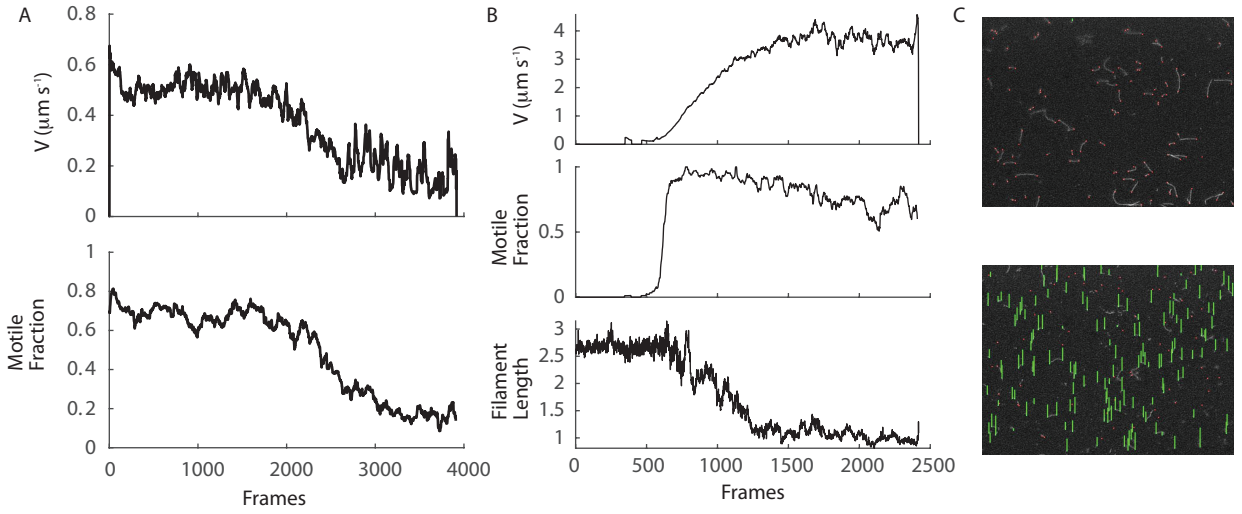
3I is well over 100%. J), K), and L) number of filaments detected in the three videos. Each video contained 25 filaments, including filaments at frame edges and crossing filaments. For the MFE algorithm the number of filament ends was divided by 2 to get to the number of filaments.

Figure 4: MFE performance on actual motility videos. A) Velocity (top) and motile fraction (bottom) profile of actin on phasic myosin in which myosin light chain phosphatase diffuses into the chamber starting at frame 0. B) Velocity (top), motile fraction (middle) and length (bottom) profile of actin motility on skeletal muscle myosin in which ATP starts to diffuse into motility chamber at frame 0. C) 2 frames from output video of MFE algorithm of video in B) showing red markers on stopped filament ends and green bars indicating velocity of moving filament ends. Top is frame 150, bottom is frame 2000.


A
P1
P2
P3
# of components
Threshold Value
B
BW
Sk
C
1
2
D
E
Distance (μm)
Frame
F
Stopped
G
Velocity (μms-1)
Frame


A
Short filaments, no noise
B
Video 1a
Analysis
Actual
C
Video 1b
D
Video 1c
E
Short filaments, noise
F
Video 2a
G
Video 2b
H
Video 2c
I
Long filaments, noise
J
Video 3a
K
Video 3b
L
Video 3c
V (μm s⁻¹)
Motile Fraction
RMS error (% V_actual)
Frames
0
500
1000
5
1
0.5
50


A
B
C
V (mm s-1)
D
E
F
Motile Fraction
G
H
I
RMS Error
J
K
L
#Filaments Detected
Frames
Frames
Frames


A
V (μm s-1)
Motile Fraction
Frames
B
V (μm s-1)
Motile Fraction
Filament Length
Frames
C