# *C. elegans* motility analysis in ImageJ - A practical approach

Summary

This document describes some of the practical aspects of computer assisted analysis of *C. elegans* behavior in practice. *C. elegans* exhibit different movement schemes according to which environment it finds itself in. The first section describes how to optimize raw movies of worms crawling on an agar plates and analyze the speed of movement in terms of body-lengths-per seconds and make a histogram of the instantaneous speeds. The second section describes an alternative method to optimize movies of worms swimming in liquid to quantify movement in terms of body-bends-per-seconds.

General considerations

In order for a computer to be able to able to locate and track motion of worms a reasonable quality movie must be obtained. Make sure that the movie has a good contrast without being too over exposed and that the animals appear as dark shapes (if possible with no highlights) on a brighter background. The below picture from a move which has appropriate contrast.
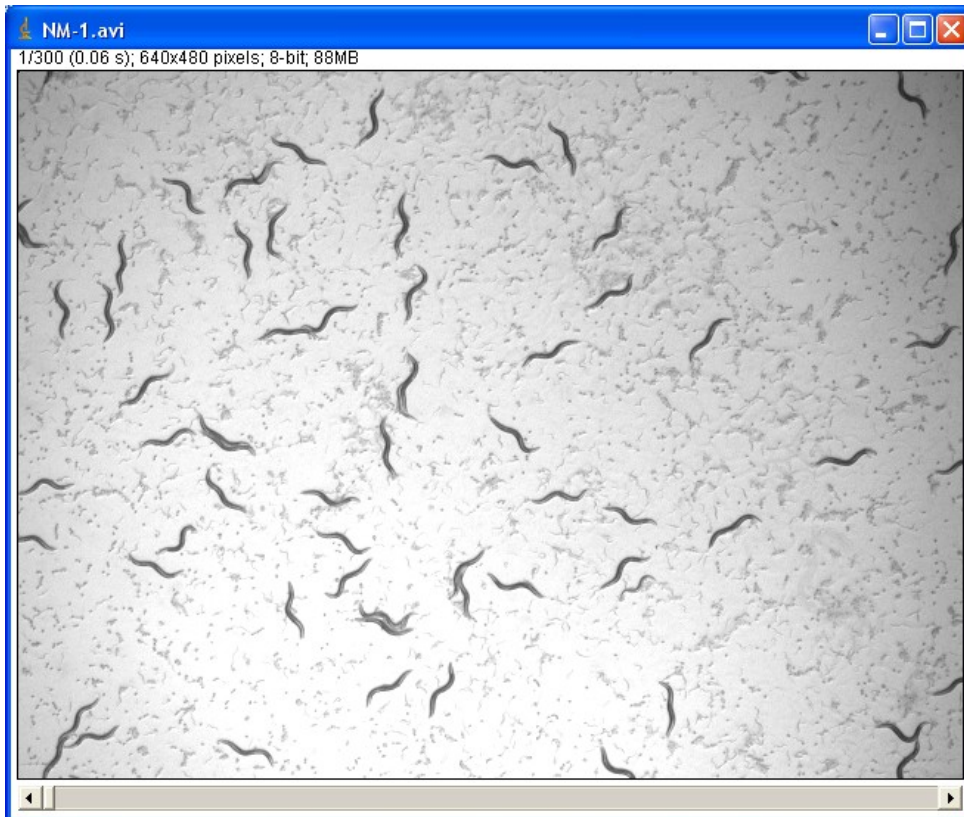
1. Crawling analysis

Loading movies into ImageJ.

Open movie directly in ImageJ[1], by either dragging the files onto the ImageJ bar or by selecting the "File"->"Open.." menu. By using the Quicktime[2] and BIO-formats plugin[3], several other, manufacturer-specific movie formats can be loaded into ImageJ. ImageJ may ask if you want to convert to 8-bit grayscale. Since movies are actually recorded in grayscale check it to save memory.

 or
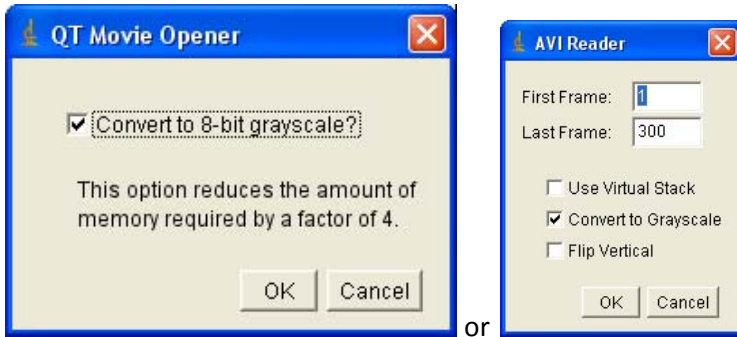
The movie should load up in ImageJ:
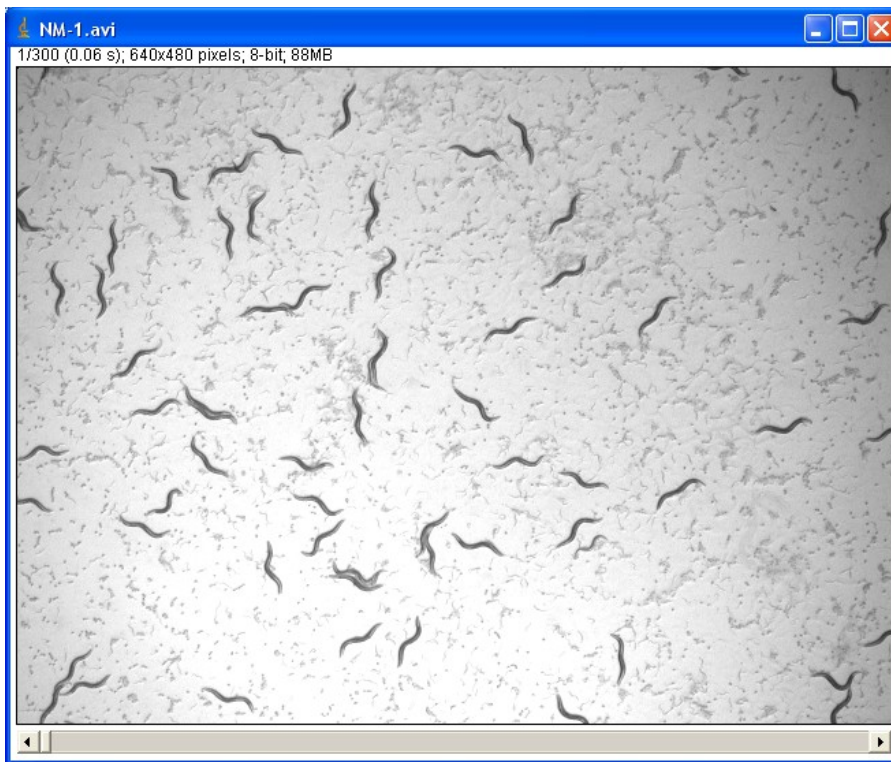
Background subtraction by rolling ball method

Although the worms are clearly visible to the naked eye, a machine would have problems detecting the positions of the animals, because of the uneven background. To remove this uneven background select "Process" -> "Subtract Background..". A dialog like the one below will appear. You can adjust the "Rolling ball radius" and directly see what that does to your image by checking the "Create Background" and "Preview" options. (Remember to uncheck "Create Background" to actually process movie.)
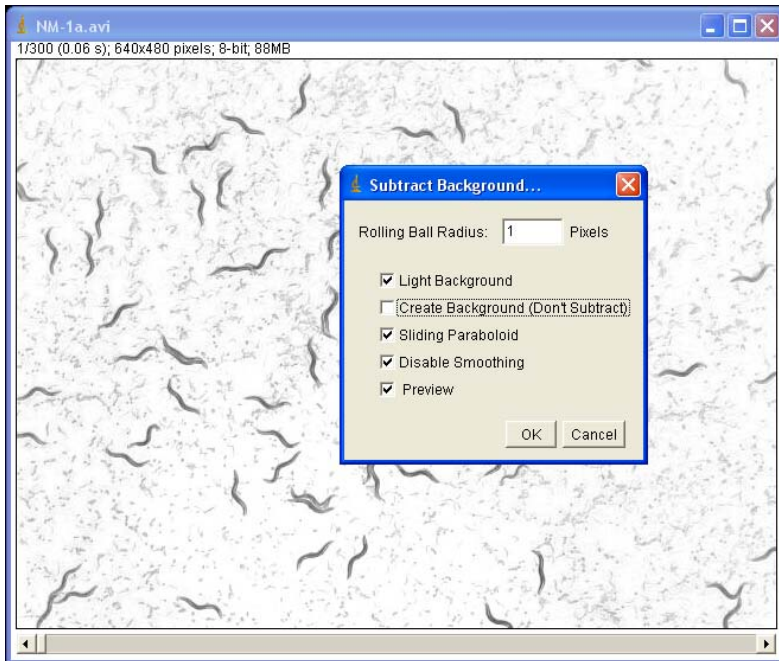

After background-subtraction the movie will look something like this:
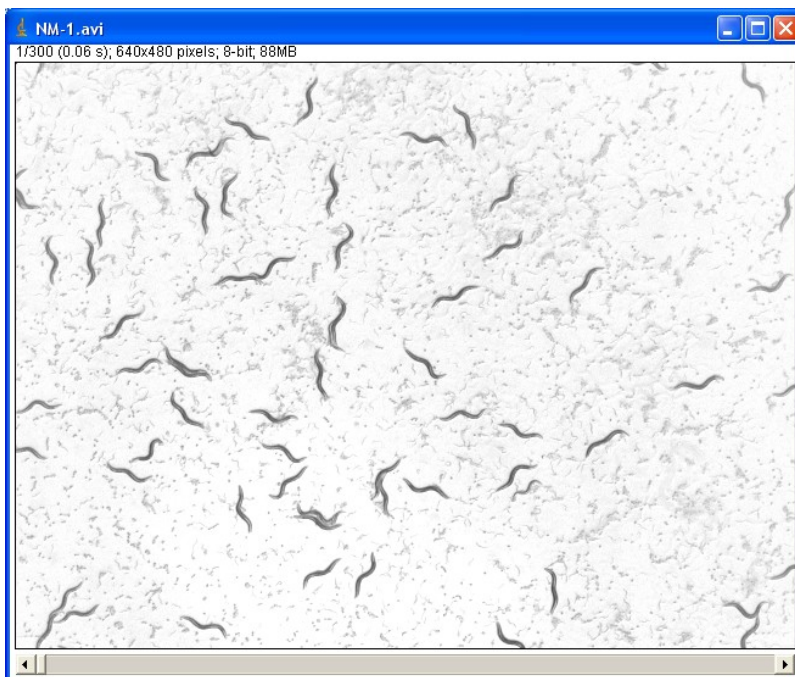
Converting movie to binary

You're now ready to indicate what pixels are worms and what pixels are background by adjusting the threshold. Select "Edit" -> "Adjust->" -> "Threshold". Select the appropriate automatic thresholding algorithm (Usually "MaxEntropy" or "Otsu" give good results) so that worms appear red. Alternatively, adjust the sliders to find a proper level where worms are red and few spots appear on background. Once an appropriate level is found click "Apply".
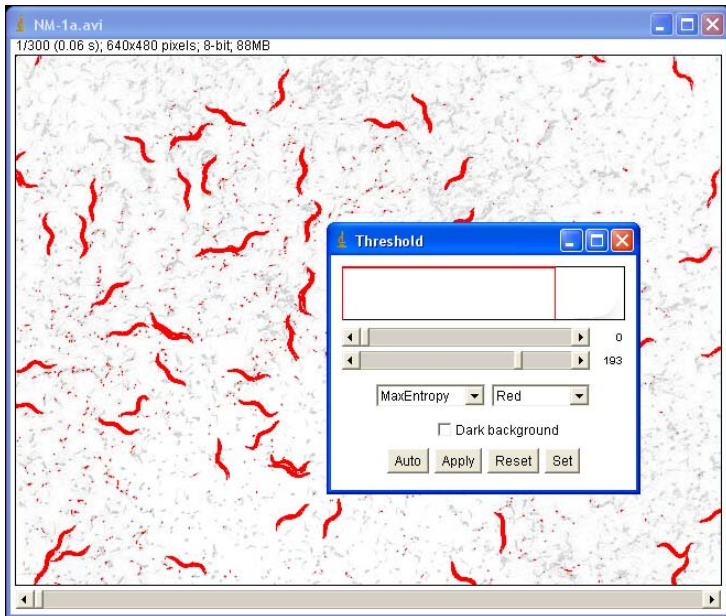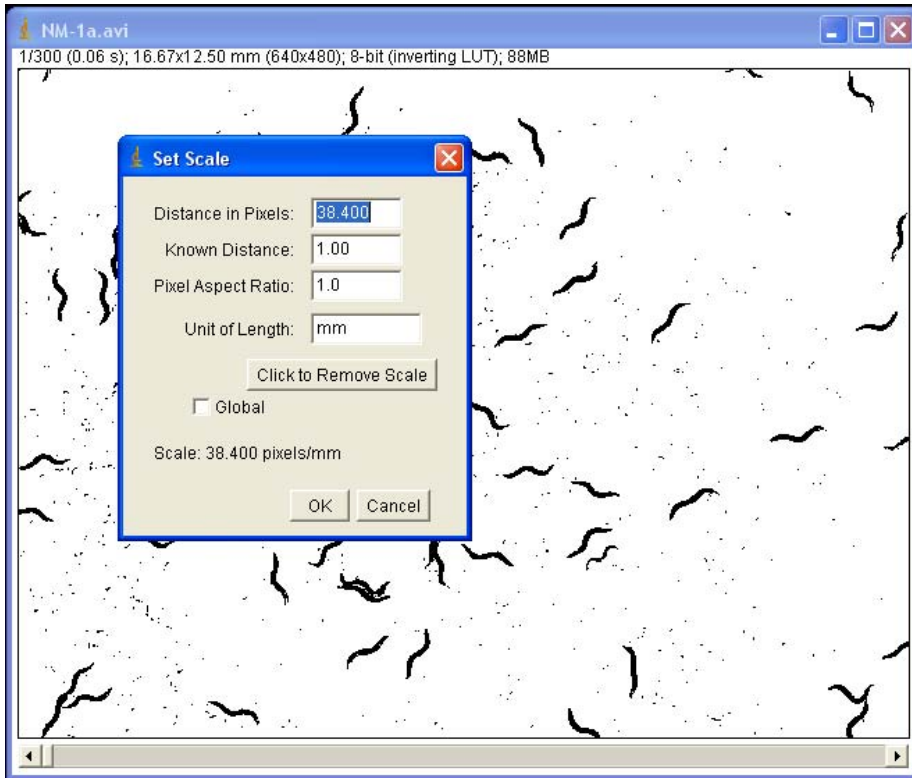

The Thresholded movie will should look something like this – it does not matter if smaller amounts of pixilated background is still present.

Next we inform ImageJ of the real-world scale of the animals in the movie. In this case 38.4 pixels corresponds to one mm in the real world.

Alternatively, one could use 38.4 pixels per 1000µm as scaling or the analysis can be performed in pixel values by using "Click to Remove Scale" option.

Tracking of crawling worms

Now run the plugin "wrMTrck" by selecting "Plugins" -> "wrMTrck". A dialog like the one below will appear:
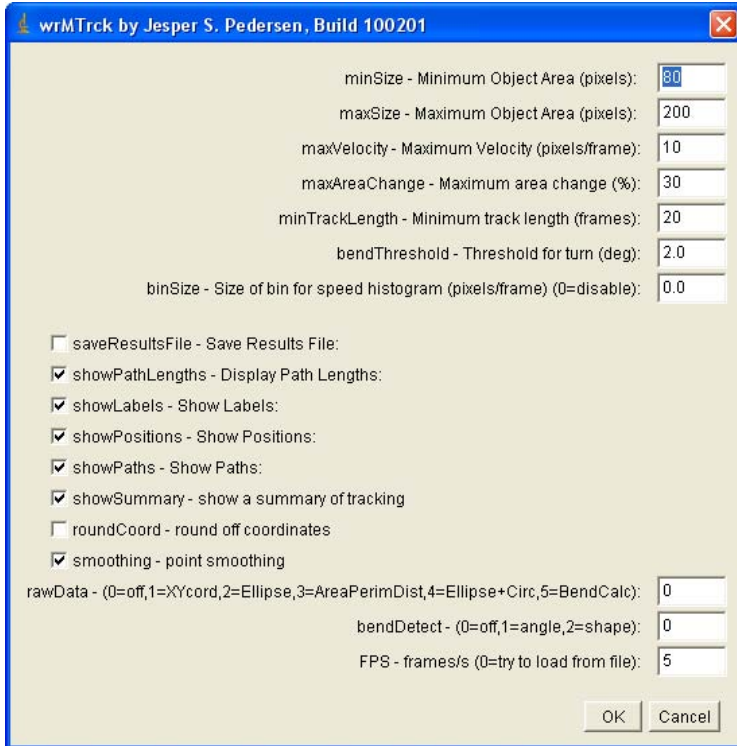
Indicate the minimum (minSize) and maximum (maxSize) worm area in pixels. The maxVelocity is the number of pixels a worm is allowed to travel between two frames. "maxAreaChange" is the % change in area between consecutive frames allowed - for example if two animals collide on the plate each track is broken due to the doubled area. Tracks shorter than "minTrackLength" will be discarded. "FPS" is use to calculate the absolute speed of animals per second.

"bendThreshold" is used for detection of body-bends (see next section). binSize for speed histogram. If desired, wrMTrck will supply raw data from the analysis can in terms of either 1) X and Y coordinates with flags for potential overlaps between objects, 2) Major, minor and angle of ellipse fitting on object, 3) Area, perimeter and distance moved since last frame, 4) Ellipse and Circularity fitting to detect deep bends or 5) bend calculation data.

If "showPaths" was selected, the worm-tracks will appear as lines of different shades of gray on a canvas after the analysis:
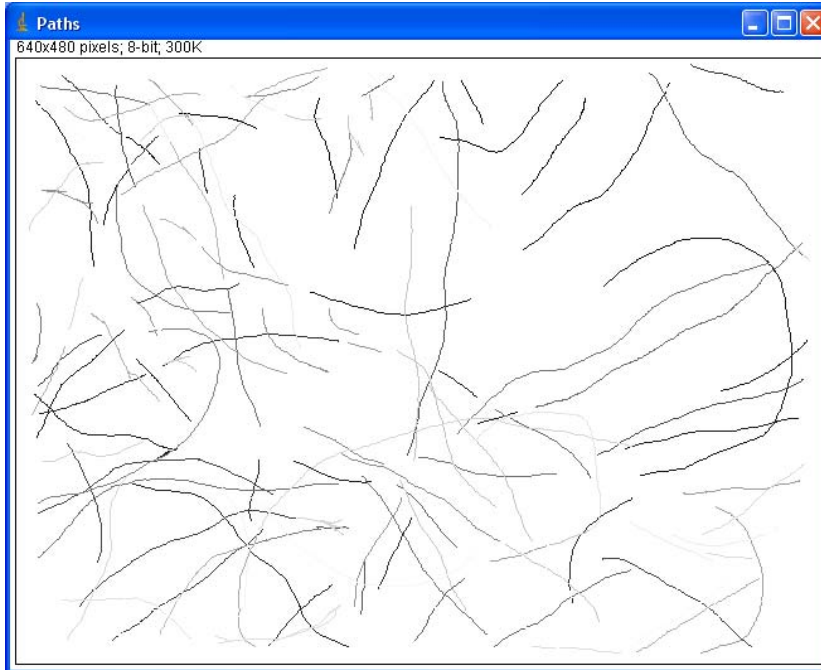
If showLabels and showPositions were checked, you should also see a movie where each worm is indicated by a number, angle, area and x/y-coordinate.

A Results window will pop-up, which contains the tracking analysis of all worms detected:

The columns of the result table contain the following:

Column label	Meaning
Length	Sum of length of all movement vectors between frames given track
Distance	Distance covered by animal from start to finish.
#Frames	Number of frames the animal was tracked.
1stFrame	The first frame where the animal is tracked
Time(s)	The time animal was tracked. (Zero if FPS value not detected correctly)
MaxSpeed	Maximum distance (pixels) covered by one animal between two frames
Area	Average area of the animal tracked (in this case in mm^2)
sdArea	Standard deviation of the area.
Perim	The length the average perimeter (outline) of the animal tracked. This number is approximately two times the length of the animal. (i.e. 1.1-1.25mm in this case). Tracks with abnormal lengths could indicate that two animals moved together as one.
sdPerim	Standard deviation of the perimeter. You may want to discard animals with large sdPerims, since this could indicate collision between animals was not detected by wrMTrck.
AvgSpeed	The average speed calculated from Length/Time(s).
BLPS	Body Lengths Per Second, calculated by dividing the Length of the track by Perim/2 and the time(s).
avgX	The average X-coordinate of the track
avgY	The average Y-coordinate of the track

Note that by checking the "saveResultsFile" the analysis will be automatically saved as a txt file.

Summary of analysis

In addition to the result for individual tracks, wrMTrck generates a summary window in which subsequent files analyzed will be added.

Summary												
File Edit Font												
File	nObj	nFrames	nTracks	totLength	ObjFrames	ObjSeconds	avgSpeed	avgArea	avgPerim	stdSpeed	stdArea	stdPerim
NM-1a	49.0	300	155	15143.09	9377.0	1875.4	8.356919	143.89023	91.99933	2.7683415	12.627668	6.7967916

Column label	Meaning
File	The file name of the movie-file processed.
nObj	the maximum number of objects within the size constraints observed in a single frame.
nFrames	The number of frames in the movie
nTracks	The number of tracks in the movie that had a length longer than minTrackLength
totLength	The sum of the Length of all tracks.
ObjFrames	Number of objects multiplied by the number of frames that they are tracked.
ObjSeconds	The seconds that objects were tracked (ObjFrames /FPS).
avgSpeed	The average speed of all objects tracked in movie
avgArea	The average area of all objects tracked in movie
avgPerim	The average perimeter of all objects tracked in movie
stdSpeed	The standard deviation of the speed of objects
stdArea	The standard deviation of the area of objects
stdPerim	The standard deviation of the perimeter of objects.

-

Histogram analysis

If the field "binSize" contained a value greater than zero the wrMTrck plugin will attempt generate data for a histogram of the instantaneous speeds for each track immediately below the results section. A histogram is given for each animal, and by summing all the histograms the instantaneous speeds for all animals in each bin one can generate an overall histogram for all animals in the movie:
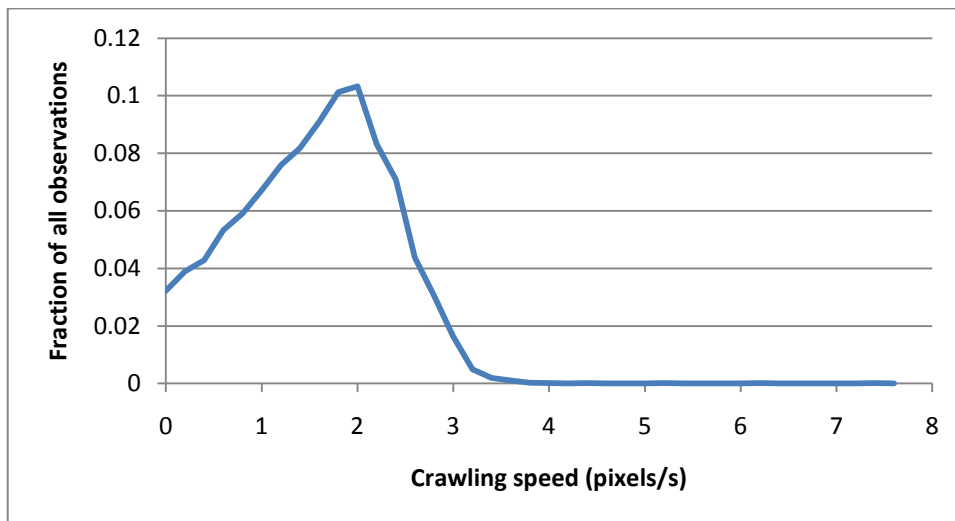
2. Thrashing analysis – Quantification of body-bends

For thrashing analysis it is important to obtain movies with high frame rates; A healthy worm completes a up to approximately two body-bends every second (2Hz or 0.5s per bend), therefore the frame-rate needs to be considerably higher. The new Orca2 worm-room camera is capable of recording at up to 28 frames per second (fps), but movies of ~16 fps are also appropriate. High fps movies are nice but also take up a lot of space on the harddrive – Unfortunately ImageJ has some unfortunate memory limitations so generally it's wise not record movies larger than 500MB, or approximately 1 minute. The image below shows a frame of a movie of day 3 of adulthood N2 animals washed off an NGM plate in M9.

### Removing flickering from 60Hz light-sources

Sometimes the original movie contains flickering from the 60Hz light-source used to illuminate the worms. This flickering can be removed from the movie using the "Stack_Deflicker" plugin, which will normalize the average intensity of all frames to either the minimum, maximum or any given frame number:
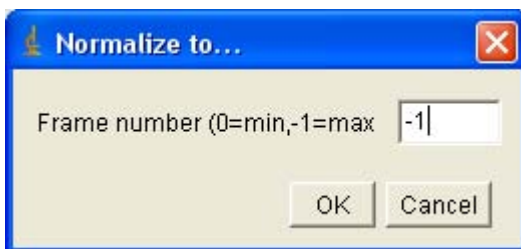
Generating maximum intensity image for background subtraction.

High frame-rates also incurs recording a lot of frames, which means that normal background subtraction may be slow and insufficient. However, ImageJ can rapidly generate Z-projections of even large movies. Choose "Images" -> "Stacks" -> "Z-Project…". Select either Max or Min intensity:
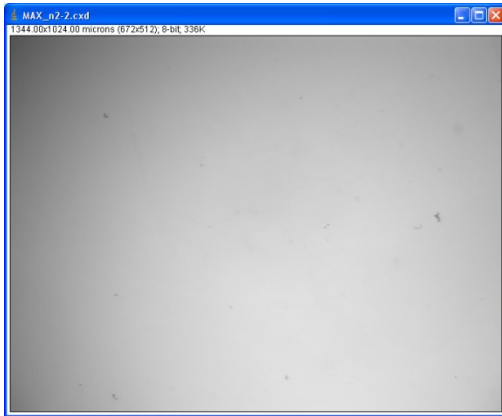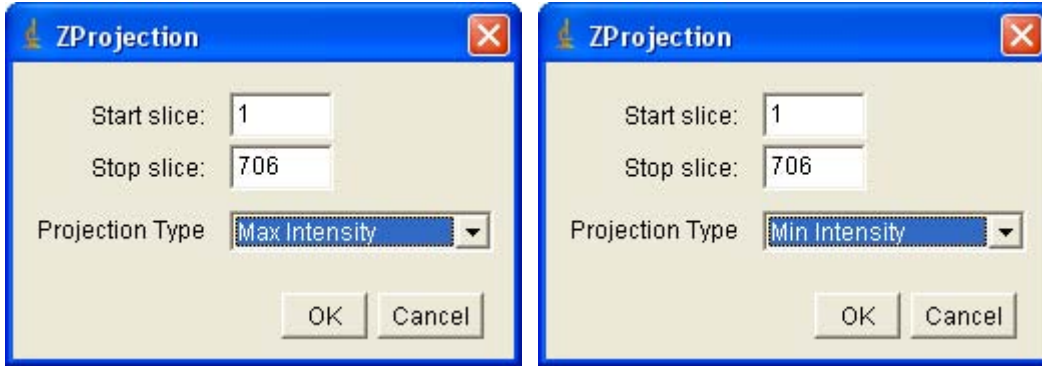
Max intensity   Minimum intensity

Note that the maximum intensity projection contains all constant elements in the movie, while the minimum intensity projection also shows the tracks of the animals. Subtracting this image from the maximum intensity projection from the original movie will remove everything constant in the movie - including immobile animals. In case there are immobile animals the Z-max projection can be used to create a creating background using the "subtract background function".
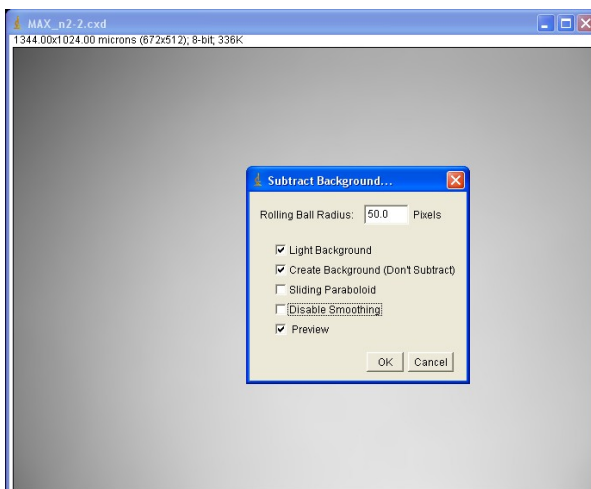
Subtracting maximum projection from movie

Bring up the image calculation window in ImageJ by selecting "Process" -> "Image Calculator…":
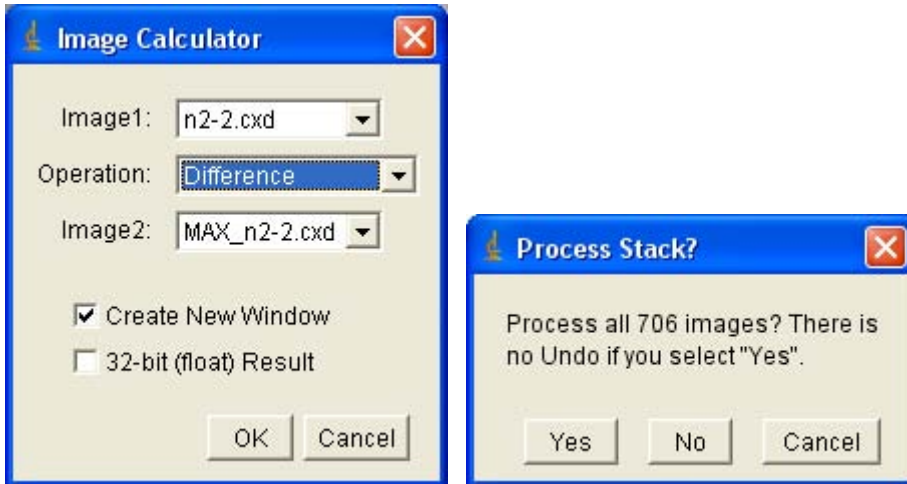
Since high fps movies are generally very large it is wise to uncheck the "Create New Window" since a new window will require as much memory as the original movie. Select the Original movie as Image1 and the maximum projection as Image2. Select "Difference" as the operation and click OK and select yes to process all frames. A movie containing only the worms should appear on your screen. This movie can again be converted into the binary format required for tracking. The binary movie still takes up a large amount of memory (in this case 233 MB). However, by saving this move in ZIP format the movie only takes up about 2 MB of disk space.
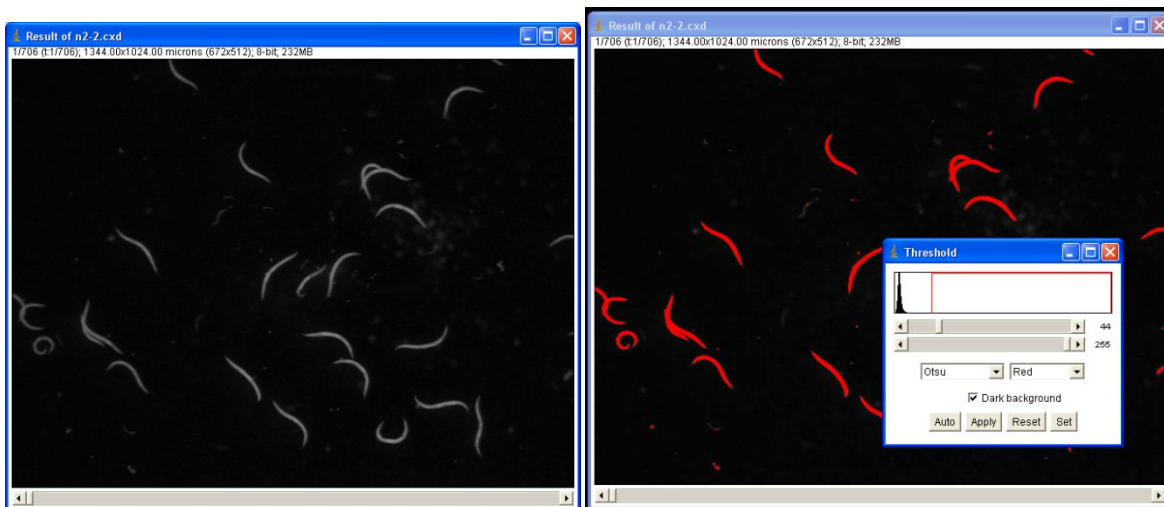
Quantification of body bends in movie

ImageJ features a sophisticated particle analysis package with an ellipse fitting algorithm that reports the angle (the direction in which the worm is pointing, between 0 and 180°, because heads and tails are the same) and the major and minor axis of the fitted ellipse. The images below show worms in various shapes and an example of the fitted ellipses.
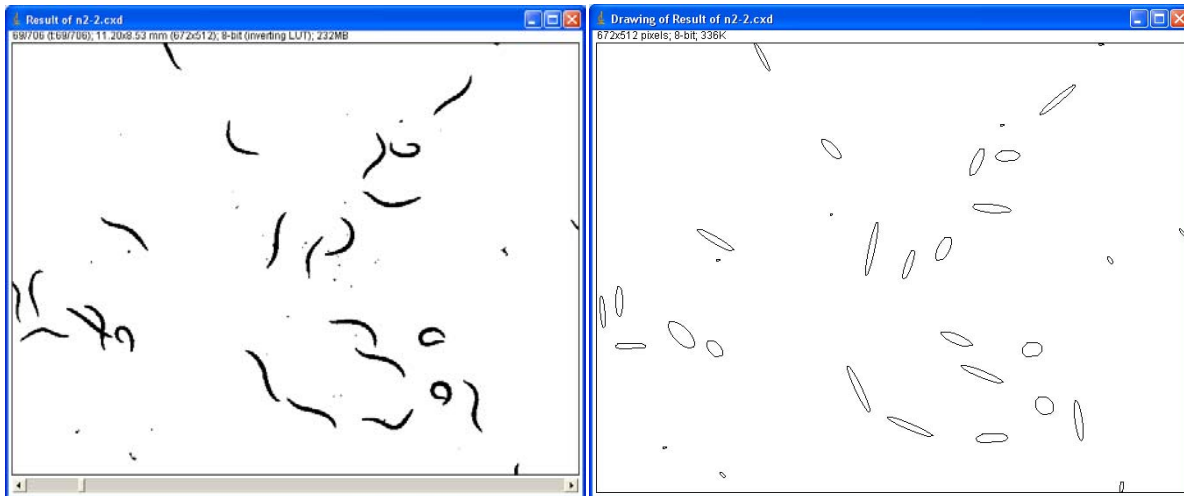
The ImageJ wrMTrck plugin uses the ellipse fitting to count the body-bends using either the angle or the shape (major/minor ratio). The method can be chosen by setting the "bendDetect" parameter:
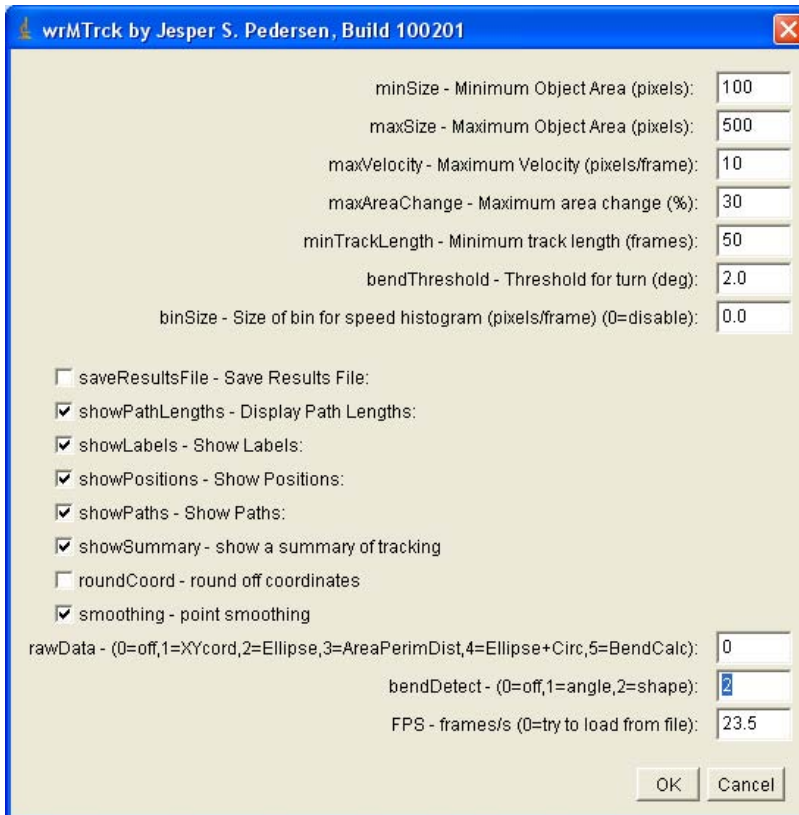
The inner workings of the two algorithms are explained below

1) Angular velocity.
    The angular velocity (red) is calculated as the differences between the angles of an object (blue) in consecutive frames. By summing the angular velocities, and setting the sum to zero every time the direction changes, the green curve is obtained (which cor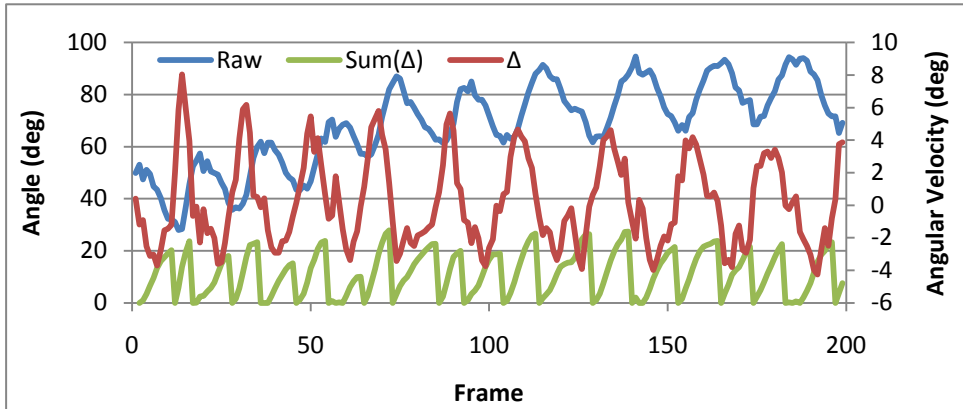responds to the area under the angular velocity curve). Every time this sum is above a given threshold the algorithm counts it as half a body-bend. In the curve below the animal observed completes 9½ body bends, which correspondsto 19 peaks above a threshold of 3.

2) Shape changing velocity.
    The shape of the worm is calculated as the ratio between the major and minor axis of the fitted ellipse (red). The red curve shows 19 peaks indicating that the animal is stretched 19 times. By differentiating this shape parameter and summing the peaks as above the same animal we obtain 38 rapid shape changes indicating that the worm completed 38/4 = 9½ body bends.
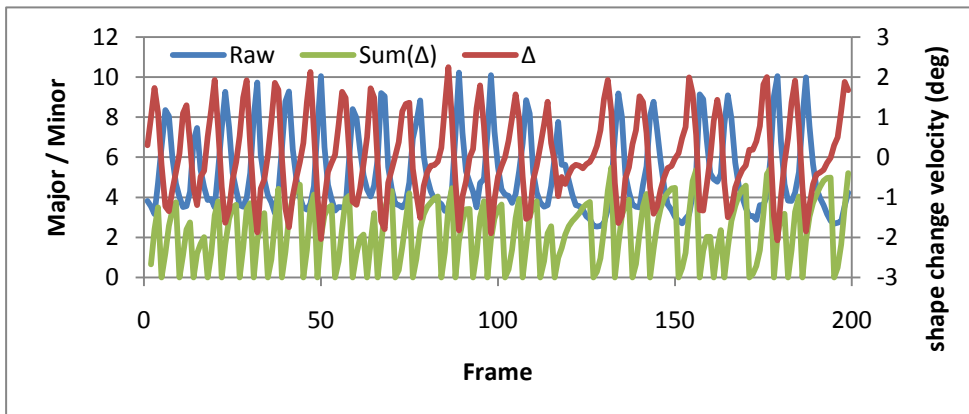
As with the crawling analysis, the wrMTrck plugin tracks individual objects and can display the path and an overview movie f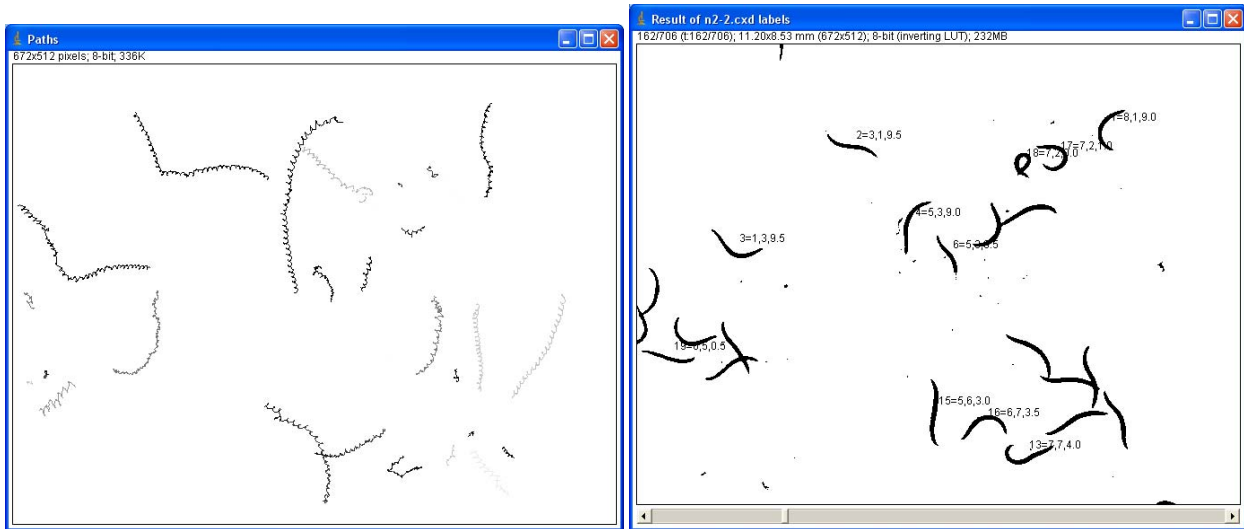or quality control purposes. Next to each tracked object the number of body-bends counted is given, so that the user can easily validate the automated analysis.

A result table showing the result of the analysis for the individual animals, now containing two extra colums "Bends" and "BBPS" (body bends per second).

	Track	Length	Distance	#Frames	1stFrame	Time(s)	MaxSpeed	Area	sdArea	Perim	sdPerim	avgSpeed	BLPS	avgX	avgY	Bends	BBPS
1	1	2.912	1.745	414	0	17.617	0.493	0.079	0.003	2.402	0.056	0.165	0.138	8.801	1.708	18.500	1.050
2	2	4.960	2.749	706	0	30.043	0.471	0.062	0.005	2.210	0.167	0.165	0.149	3.305	1.835	36.500	1.215
3	3	5.749	2.705	697	0	29.660	0.719	0.070	0.010	2.401	0.119	0.194	0.161	1.245	3.597	34.500	1.163
4	4	6.645	3.304	706	0	30.043	0.690	0.085	0.004	2.507	0.160	0.221	0.176	5.294	2.412	33	1.098
5	5	1.015	0.633	125	0	5.319	0.627	0.088	0.002	2.530	0.057	0.191	0.151	6.568	3.906	6.500	1.222
6	6	1.523	0.648	305	0	12.979	0.502	0.057	0.002	1.958	0.091	0.117	0.120	5.780	4.003	13	1.002
7	7	0.445	0.217	83	0	3.532	0.505	0.086	0.001	2.423	0.321	0.126	0.104	8.240	5.797	2	0.566
8	8	0.276	0.099	108	0	4.596	0.188	0.057	0.003	2.146	0.056	0.060	0.056	0.616	5.723	1.500	0.326
9	9	1.105	0.522	96	0	4.085	0.663	0.093	0.004	2.603	0.129	0.270	0.208	4.906	6.486	5	1.224
10	10	0.224	0.083	86	0	3.660	0.272	0.095	0.002	2.435	0.386	0.061	0.050	8.504	6.870	0.500	0.137
11	11	0.474	0.281	86	0	3.660	0.310	0.084	9.967E-4	2.536	0.040	0.129	0.102	9.196	7.219	4.500	1.230

In addition to the data for the individual tracks, a summary of the averge BBPS (avgBBPS) for all animals in the movie as well as the standard deviation of (stdBBPS).

File	nObj	nFrames	nTracks	totLength	ObjFrames	ObjSeconds	avgSpeed	avgArea	avgPerim	stdSpeed	stdArea	stdPerim	Bends	avgBBPS	stdBBPS
Result of n2-2	20.0	706	35	64.28094	8333.0	354.59574468085106	0.1749951	0.07896212	2.393649	0.07690454	0.013009001	0.22864078	372.0	0.9722081	0.4234808

NOTES:

Because the animals wobble and the positions are determined as the center of gravity for the objects, the lengths of the tracks are considerably longer than the distances covered. Verify that the threshold setting used is appropriate, by manually counting the number of body-bends for a few of the tracks. Note that changes in the fps of the movie will affect the appropriate threshold setting.