

Multi-Environment Model Estimation for Motility Analysis of *Caenorhabditis elegans*

Raphael Sznitman¹, Manaswi Gupta¹, Gregory D. Hager¹, Paulo E. Arratia², Josué Sznitman^{3*}

1 Department of Computer Science, Johns Hopkins University, Baltimore, Maryland, United States of America, **2** Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, Pennsylvania, United States of America, **3** Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, New Jersey, United States of America

Abstract

The nematode *Caenorhabditis elegans* is a well-known model organism used to investigate fundamental questions in biology. Motility assays of this small roundworm are designed to study the relationships between genes and behavior. Commonly, motility analysis is used to classify nematode movements and characterize them quantitatively. Over the past years, *C. elegans*' motility has been studied across a wide range of environments, including crawling on substrates, swimming in fluids, and locomoting through microfluidic substrates. However, each environment often requires customized image processing tools relying on heuristic parameter tuning. In the present study, we propose a novel Multi-Environment Model Estimation (MEME) framework for automated image segmentation that is versatile across various environments. The MEME platform is constructed around the concept of Mixture of Gaussian (MOG) models, where statistical models for both the background environment and the nematode appearance are explicitly learned and used to accurately segment a target nematode. Our method is designed to simplify the burden often imposed on users; here, only a *single* image which includes a nematode in its environment must be provided for model learning. In addition, our platform enables the extraction of nematode 'skeletons' for straightforward motility quantification. We test our algorithm on various locomotive environments and compare performances with an intensity-based thresholding method. Overall, MEME outperforms the threshold-based approach for the overwhelming majority of cases examined. Ultimately, MEME provides researchers with an attractive platform for *C. elegans*' segmentation and 'skeletonizing' across a wide range of motility assays.

Citation: Sznitman R, Gupta M, Hager GD, Arratia PE, Sznitman J (2010) Multi-Environment Model Estimation for Motility Analysis of *Caenorhabditis elegans*. PLoS ONE 5(7): e11631. doi:10.1371/journal.pone.0011631

Editor: Vladimir Brezina, Mount Sinai School of Medicine, United States of America

Received: May 18, 2010; **Accepted:** June 23, 2010; **Published:** July 22, 2010

Copyright: © 2010 Sznitman et al. 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.

Funding: The authors have no support or funding to report.

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

* E-mail: sznitman@princeton.edu

Introduction

Since its introduction in the laboratory over thirty years ago [1], the nematode *Caenorhabditis elegans* has become a ubiquitous model organism to study fundamental questions in biology [2]. In particular, *C. elegans* is now widely used as a platform for drug screening and development [3,4], as well as for modeling various aspects of human diseases [5,6]. In the quest to understand the relationships between genes and behavior, this small, approximately 1 mm long roundworm offers a number of advantages for laboratory applications. These include a short life cycle, the availability of many mutants to explore gene functions, knowledge of its complete cell lineage [7,8], simplicity of the nervous system and its wiring [9], and a fully sequenced genome [10].

A widespread strategy to investigate the genetic basis of behavior is to classify nematode movements and characterize them quantitatively. Traditionally, motility quantification has been based on crawling assays [11–14], where *C. elegans* is observed to crawl on a substrate (e.g. agar plate). This is shown for example in Fig. 1(a). In the recent past, however, the number of environments used for nematode motility assays has vastly expanded. Studies of *C. elegans*' motility behavior now include various swimming assays [13,15–19], as shown in Fig. 1(b)–(d) and (g). In parallel, with the widespread availability of microfabrication techniques, nematode

motility assays are increasingly conducted in microfluidic environments [20–23]. An example of such environments is shown in Fig. 1(e) and (f). This latter platform has become particularly attractive for high-throughput drug screening applications [24,25]. Overall, with the growing variations in environments used for nematode behavioral assays, users are in need of reliable image analysis tools capable of extracting quantitative data across a wide spectrum of experimental mediums.

The analysis of motility behavior has traditionally relied on qualitative observations to describe *C. elegans*' locomotion and discriminate between wild-type and mutant nematodes. In many instances, however, qualitative variations between strains are not apparent to the trained eye (as in [26]). Such limitations have spurred the development of automated image analysis systems in an effort to deliver relevant phenotypic differences between nematode strains [27–36]. While the bulk of the research effort has been directed at analyzing locomotive traits of individual nematodes, some multi-worm tracking and feature extraction systems have also been developed [37–39]. Yet, the majority of state-of-the-art image analysis systems are designed for a *specific* environment, most commonly crawling [27,30–34,37] or swimming assays [29,39]. These systems induce a tradeoff between either limiting the range of possible assays a researcher will investigate for motility analysis or customizing segmentation

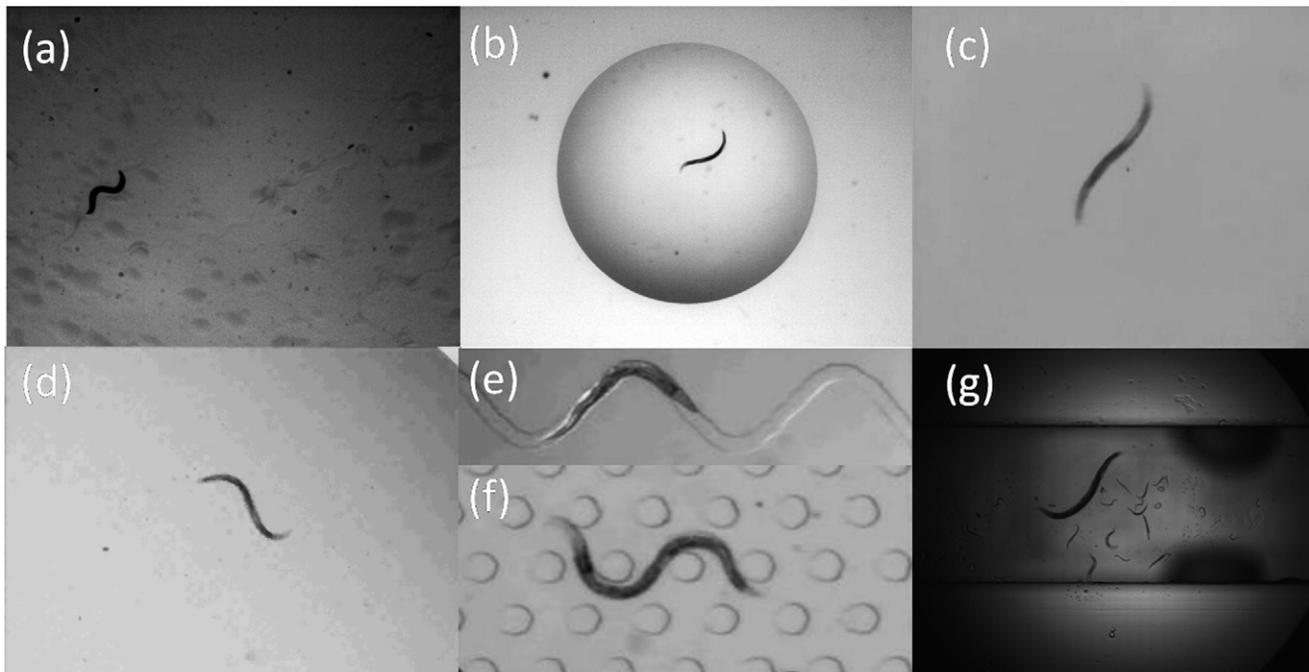


Figure 1. Examples of environments used in *C. elegans* motility assays. (a) Nematode crawling on an agar plate (Video S2). (b) Nematode swimming in a 5 ml drop of M9 buffer solution (Video S4). (c) Nematode swimming in a solution of gelatin dissolved in M9 (source: berrigel0.0perc.mov, Supplementary Material in Berri *et al.* [15]). (d) Nematode swimming inside a fluid-filled chamber (source: SM2.avi, Supplementary Material in Pierce-Shimomura *et al.* [13]). (e)–(f) Nematode locomotion in a microfluidic substrate (source: Supplemental Videos 2 and 4 in Lockery *et al.* [21]). (g) Nematode swimming in a shallow acrylic channel filled with M9 (Video S5). Nematodes shown in (a) through (g) are wild-type (N2) *C. elegans* and are all approximately 1 mm long.
doi:10.1371/journal.pone.0011631.g001

parameter selection across varying environments. However, an optimal system for the user is one which ideally bypasses such compromise.

Current image analysis systems provide users with morphological and locomotion features to quantify behavioral phenotypes of *C. elegans*. Such features include amongst other nematode speed [36,37], wavelength and frequency of body undulations [29,30], body curvature [27,31], and omega bends [33]; several of which make use of nematode centerline data, also known as ‘skeletons’. In practice, features are extracted from binary images, or *segmentations*, separating the nematode from its environment, or *background*. Several analysis systems compute binary images by applying a simple intensity-based threshold at each pixel location [30,31,35,36,40,41]. Most commonly, this involves having the user manually select an appropriate range of intensities which characterizes the nematode. A variation to this approach has been the use of an adaptive threshold where nematode intensities, or *appearance*, are assumed to significantly differ from the average background intensities [27,33,34]. While these methods have shown promising capabilities, the range of environments for which they can be used for is in fact limited. This limitation is illustrated in Fig. 2 where the pixel intensity distributions of the nematode and background are respectively plotted for the environments shown in Fig. 1. Here, distributions are assumed to be Gaussian and parametrized with the mean and standard deviation of pixel intensities. In none of the cases shown can a single threshold separate any pair of distributions without causing significant errors (Fig. 2). While threshold-based techniques can still be used to compute accurate segmentations, this requires significant effort on the user-end to adjust appropriate threshold values along with other noise canceling schemes (*e.g.* median filters, morphological

operators, background subtraction, etc.). Altogether, this tedious process makes thresholding ill-suited for applications in complex background environments.

More recently, alternative approaches to thresholding techniques have been pursued. For example, the work of Stauffer and Grimson [42] has been applied to the problem of nematode segmentation [38,39]. Here, the idea is to systematically learn how background pixels are individually distributed and use this information to segment the nematode. The learning process is done using a set of *training* images to statistically model the appearance of the background. Namely, each pixel is modeled by means of a Mixture of Gaussians (MOG), where the parameters of the model are learned from the training image set. This approach has been recently shown to provide excellent results for nematode segmentation [38,39] as well as for other applications [42–44]. A major drawback, however, of modeling pixels with MOGs is that many parameters must be learned; this requires a large set of nematode-free training images. This condition largely prohibits extracting nematode segmentations from arbitrary sequences (*e.g.* open-access material).

In the present study, we propose a novel framework for image analysis of *C. elegans* that is versatile across a wide range of environments. Moreover, our system is designed to greatly simplify the burden imposed on the user end; only a *single* image from a sequence which includes a nematode in its environment must be provided. From this input, models for *both* the background and the nematode appearance are individually learned using MOGs (see Methods). These models are then applied to segment the nematode in subsequent images. Next, we provide an original algorithm for extracting nematode skeletons for applications to *C. elegans* behavioral assays. Nematode segmentation and skeletonizing

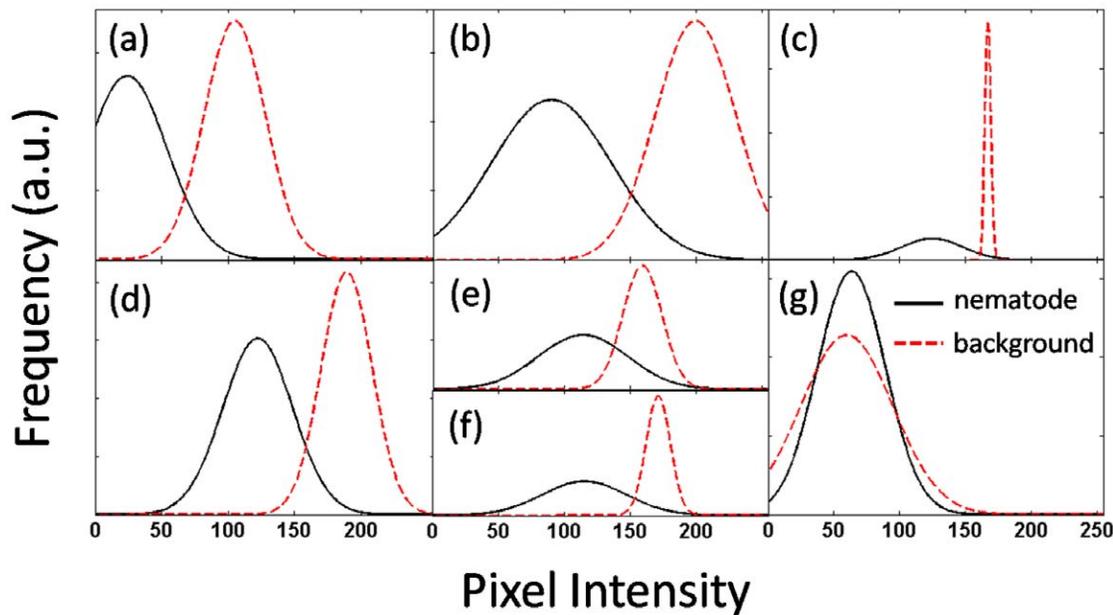


Figure 2. Pixel intensity distributions of nematode and background environment. Plots (a) through (g) correspond to distributions obtained from the images of Fig. 1. Grayscale pixel intensities vary between 0 (black) and 255 (white). Distributions are assumed to be Gaussian and parametrized with the mean and standard deviation of pixel intensities. doi:10.1371/journal.pone.0011631.g002

algorithms have been packaged together in a software for straightforward use by a broad range of researchers. We test our image analysis system on representative locomotive environments (Fig. 1) and compare performances with a state-of-the-art thresholding method (see Results). Finally, we illustrate some examples of motility metrics (*e.g.* body curvature) which are commonly sought from nematode skeleton data (see Discussion). We discuss future directions for algorithmic improvement (*e.g.* multi-worm tracking) as well as new directions for potential applications (*e.g.* cell tracking).

Methods

The *Multi-Environment Model Estimation* (MEME) framework consists of two sequential stages. As a first step, (i) the user provides information, allowing a model for the nematode and the background environment to be learned. In the second stage, (ii) the nematode and background models are used to segment the nematode and then extract its skeleton for a sequence of images. In stage (i), the user is required to input a hand-segmentation of the nematode and its corresponding width for a *single* image (see Video S1). This approach can be viewed as a form of “One Shot Learning” [45,46], where model learning occurs only once, from a single labeled example. A flowchart of the MEME framework is schematically shown in Fig. 3.

We briefly introduce the notation used throughout the article. We define the sequence of images provided by the user as $\mathcal{I} = \{I_1, \dots, I_N\}$ for N discrete time steps, where at each time step t , $I_t \in [0, 255]^{n \times m}$; n and m are the width and height of each image, respectively. We denote the user provided data (U) as a triple $U = \{I_U, S_U, W_U\}$, where I_U is an image from \mathcal{I} , S_U is the nematode body segmentation and W_U is the nematode width (Fig. 3). We specify the intensity models derived from U as F_W for the worm and F_B for the background. For any given image I_t included in \mathcal{I} , we define the computed segmentation of the nematode as S_t and the list of ordered pixel coordinates describing the nematode skeleton as L_t .

Nematode Segmentation

Our first step is to provide an automatic mechanism to compute accurate nematode segmentations for a single target using a static camera (as in [38,39]). To do this, we build on the idea of using Mixtures of Gaussians (MOG) [42–44] to construct accurate and robust appearance models for the background and the nematode. As it is often the case for MOG methods, each pixel in an image is treated as a random variable which can be modeled by summing K weighted Gaussian distributions. This can be formally written as

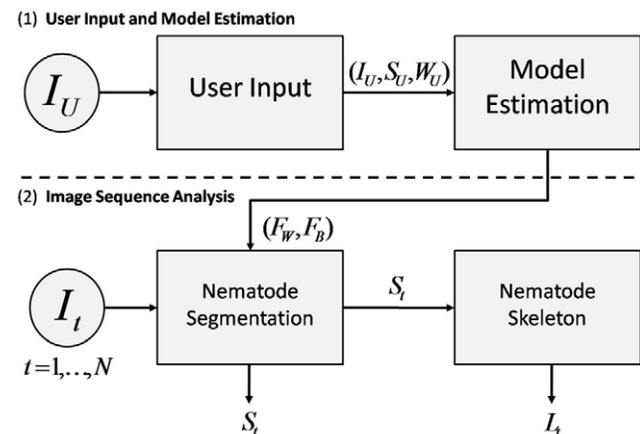


Figure 3. Multi-Environment Model Estimation (MEME) framework overview. The system consists of two components: (1) a user input and learning stage and (2) an image analysis stage. In stage (1), the user provides an image (I_U) and marks the nematode boundary (S_U) and width (W_U). From this input, appearance models for the nematode (F_W) and background (F_B) are learned. In (2), for each image (I_t) in a sequence, nematodes are then segmented (S_t) by using F_W and F_B . Nematode skeletons (L_t) are then extracted from these segmentations. doi:10.1371/journal.pone.0011631.g003

$$P(x) = \sum_{i=1}^K \pi_i \mathcal{G}(x | \mu_i, \Sigma_i), \quad (1)$$

where x is a pixel intensity value, π_i, μ_i and Σ_i are respectively the weight, mean and covariance of the i th Gaussian distribution \mathcal{G} . These parameters are usually estimated by means of an Expectation-Maximization (EM) algorithm [47]. The intuition behind this model is that each individual Gaussian represents the appearance a particular pixel may take. Therefore, combining each Gaussian provides a way to model complex pixel observations. Typically, doing this over all pixels in an image is an effective way to model background scenes [42–44].

A consequence of this approach is that the number of MOGs used is considerable (*i.e.* the total number nm of pixels in an image) and the number of parameters required is very large ($3Knm$). In turn, a substantial number of images is needed to estimate the MOG parameters accurately as each image only provides a single sample for each MOG (see [42] for more details). Moreover, the entire background scene must be visible when attempting to estimate these parameters, since each image is used to model the background and not the nematode. This latter condition becomes problematic when image sequences always contain a nematode in the field of view (*e.g.* Supplementary Videos available in references [13,15,21]).

To avoid such drawbacks, we choose instead to model the nematode appearance (F_W) in addition to the background model (F_B) by means of MOGs. To learn the parameters of F_W , we use the information gathered from the user. Namely, S_U provides

the pixel region of I_U containing the nematode as delineated by the user (Fig. 4a). From this region, we randomly select pixel locations and extract $[d \times d]$ image patches from I_U around each location. These patches are then vectorized and treated as independent samples. We denote this feature extract process as $f(u, v; I, d) = x \in \mathbb{R}^{d^2}$, where we select a patch around pixels (u, v) for any given image I . Notice that when $d=1$, this reduces to sampling the selected pixels only; Fig. 4(b) shows the histogram of intensities for the case $d=1$. In general, applying this transformation allows for modeling intensities with respect to image patches, as this approach carries more information than individual and independent pixels. Computing an appropriate value for d is done by using a linear model (*i.e.* $d = \alpha_1 \frac{W_U}{\max(m, n)} + \alpha_0$, where α_1 and α_0 are constants). The samples extracted are then used to estimate the parameters of F_W by using the EM algorithm. Figure 4(d) illustrates a visual representation of the estimated MOGs of F_W for $K=2$ and $d=1$.

Next, modeling of the background (F_B) is done by partitioning the image (I_U) into \tilde{n} distinct and non-overlapping cells, $\mathcal{C} = \{C_1, \dots, C_{\tilde{n}}\}$, as shown in Fig. 4(c). Each cell (C_c) is then treated as a random variable and modeled with its own independent MOG. Hence, $F_B = \{F_B^{C_1}, \dots, F_B^{C_{\tilde{n}}}\}$, where each pixel in I_i is associated with a unique $F_B^{C_c}$; in our implementation, we choose $\tilde{n} = 10 \times 10$. The parameters of each $F_B^{C_c}$ are computed from extracted samples in the partition C_c . Similarly to building the nematode model, samples are $[d \times d]$ pixel patches from I_U , which have been vectorized. Examples of intensity distributions for

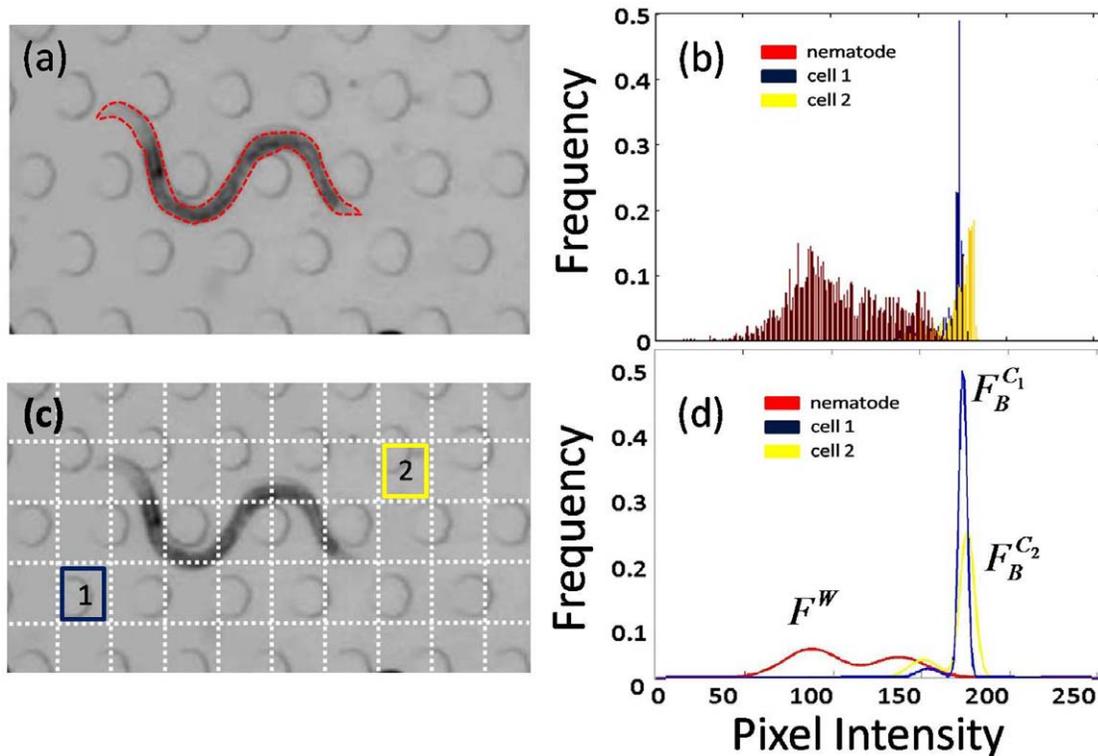


Figure 4. User input and nematode segmentation. Figures (a) through (d) illustrate the stages of the segmentation process in MEME for a sample environment (source: Supplemental Video in Lockery *et al.* [21]). (a) The user selects the boundary of the nematode on a given image. From this manual segmentation, the distribution of nematode features can be extracted. (b) Distribution of pixel intensities from the nematode region. Here, $d=1$ for illustrative purposes. (c) The background scene is partitioned into a grid, where each cell corresponds to a particular pixel block. Two arbitrary cells are labeled for clarity; their corresponding intensity distributions are shown in (b). For both the nematode and the cells, MOG parameters are then learned. (d) Representation of the MOG models for the nematode (F^W) and the two background cells ($F_B^{C_1}$ and $F_B^{C_2}$). doi:10.1371/journal.pone.0011631.g004

two arbitrary cells (Fig. 4c) are displayed in Fig. 4(b) along with their corresponding MOG representations in Fig. 4(d). Two consequences arise from such partitioning. First, only a total of $3K\tilde{n}$ parameters need to be estimated, as opposed to $3Knm$. Secondly, a single image is sufficient to estimate these parameters, as background regions covered by the nematode can still be modeled by neighboring pixels in a cell. This reduces the number of training images required and relaxes the constraint that training images must only contain background pixels. Note that when a cell is reduced to a single pixel ($\tilde{n} \approx nm$), F_B is similar to the model described in [38,39]. Alternatively, when a cell corresponds to the entire image ($\tilde{n} = 1$), F_B is similar to models typically used by thresholding techniques [27,33,34].

Nematode segmentation for image I_t can then be computed at each pixel (u,v) , belonging to cell C_c , as

$$S_t(u,v) = \begin{cases} 1 & \text{if } r(f(u,v; I_t, d), c) > 1, \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

where

$$r(x,c) = \frac{F_W(x)}{F_B^{C_c}(x)} = \frac{\sum_{i=1}^K \pi_i^W \mathcal{G}(x | \mu_i^W, \Sigma_i^W)}{\sum_{i=1}^K \pi_i^{C_c} \mathcal{G}(x | \mu_i^{C_c}, \Sigma_i^{C_c})}. \quad (3)$$

The procedure above allows one to compute the nematode segmentation for a given image I_t , where $K=2$ in our system. Notice that using the ratio of MOGs (see Eq. (3)) is an effective way to avoid any form of thresholding. This is due to the fact that both F_W and F_B are explicitly modeled. Finally, opening and closing morphological operations are used to smooth nematode segmentations.

Nematode Skeleton

Over the years, a large number of methods have been used to extract skeletons from segmented nematodes. Methods have ranged from using specific nematode models [34,38], to heuristically constructing the nematode's medial axis [27,33,35,38,41]. While these various methods have shown success, they are generally influenced by the quality of the segmentation. In an attempt to reduce sensitivity to segmentation noise, we propose an original algorithm which balances geometric features (*i.e.* nematode boundary) and global shape (*i.e.* nematode undulating posture) in a seamless framework. The proposed method has the advantage of being intuitive and simple to implement.

We cast our problem once again in a probabilistic manner such that the nematode skeleton (L_t) is considered to be a sequence of discrete unknown skeleton pixel locations ($L_t = \{l_1, \dots, l_M\}$), where each location is a point on the skeleton and must be determined. It is assumed here that either the head or tail pixel location (l_0) is initially known; determining L_t is then viewed as a sequential Bayesian estimation problem [48–50]. Given the initial position l_0 , we infer the location of the next point (l_1) by observing the likelihood of potential locations (*e.g.* the likelihood of a pixel being l_1) as well as the history of directions between subsequent pairs of points (*e.g.* from l_0 to l_1). The ‘skeleton’ algorithm is iterative such that a new location along L_t is inferred at each iteration step (τ). To infer all points in L_t , this process is simply repeated.

First, the input of our algorithm is the segmentation of the nematode for a given image (S_t). A skeleton pixel location is defined as $l_\tau = (u_\tau, v_\tau)$, where u_τ and v_τ are pixel locations in S_t .

Let v be a discrete random vector describing the direction from l_τ to $l_{\tau+1}$, such that $v \in \mathcal{V} = \{-1, 0, 1\}^2$. This corresponds to l_τ being one pixel away from $l_{\tau+1}$. Let $P_\tau(v)$ be the corresponding probability distribution of v at iteration step τ . As more skeleton pixels are inferred, the distribution $P_\tau(v)$ will evolve. Initially this distribution is uniform, as no prior information between l_τ and $l_{\tau+1}$ is known. The initial position (l_0) is found by using maximal response locations when running a coarse corner detector on S_t . Selecting the following point on the skeleton can then be computed by maximizing the Maximum a Posteriori (MAP) estimator,

$$l_{\tau+1} = l_\tau + \arg \max_v P(l_\tau | v) P_\tau(v), \quad (4)$$

where $P(l_\tau | v)$ is the likelihood that direction v leads to the next skeleton point and is modeled by

$$P(l | v) = \frac{D(l+v)}{\sum_{\tilde{v} \in \mathcal{V}} D(l+\tilde{v})}. \quad (5)$$

Here, $D(l)$ is the distance computed when applying the Chamfer distance transform [51,52] to S_t . This transformation computes the Euclidean distance of each pixel in S_t to its closest nematode boundary pixel. An example of this distance transform is shown in the contour plot of Fig. 5. Here, the boundary of the nematode has a distance of zero, while values of D increase steadily for pixels approaching the medial axis of the nematode. Equation (4) then implies that skeleton locations are picked by (i) weighing how likely pixels are to be at the center of the segmented nematode, combined with (ii) the history of the chosen vector directions. This strategy is particularly useful in cases where the segmentation is noisy, as the history of vector directions guides where the following pixel location should be located. In order to remove the possibility of selecting the same pixel several times, l_τ is removed from possible future locations by setting $D(l_\tau) = 0$.

Once $l_{\tau+1}$ is determined, the distribution $P_\tau(v)$ must be updated for the following iteration. Using Bayes rules, $P_{\tau+1}(v)$ is computed for $\forall v \in \mathcal{V}$ by

$$P_{\tau+1}(v) = \frac{1}{\mathcal{Z}} P(l_\tau | v) P_\tau(v), \quad (6)$$

where \mathcal{Z} is a normalization constant.

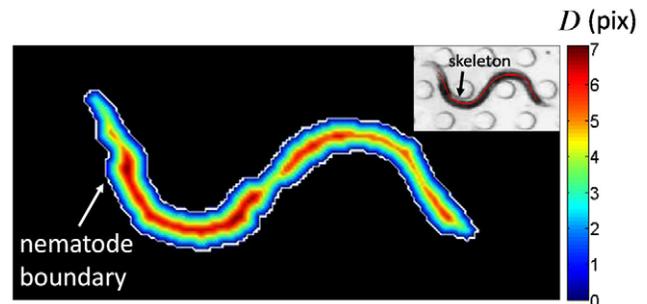


Figure 5. Computing the nematode skeleton. Representation of the Chamfer distance transform field (D) applied to the segmented nematode of Fig. 4. The value associated at each pixel of the image is the Euclidean distance (in pix) to the closest point of the nematode boundary; the distance on the boundary is zero and higher distances lie towards the nematode medial axis. (**Inset**) Resulting skeleton is achieved by balancing geometric features (*i.e.* Chamfer distance) and global shape (*i.e.* nematode curvature). doi:10.1371/journal.pone.0011631.g005

Inferring $L_t = \{l_1, \dots, l_M\}$ for a given image I_t then consists in the following algorithm. First, l_0 is given at iteration step $\tau=0$. Three steps are then repeated: (i) compute the next skeleton location from Eq. (4); (ii) update the distribution of v from Eq. (6); and finally (iii) increment the iteration step. These operations are repeated until a point on the boundary is encountered (i.e. $D(l_M)=0$). An example of the resulting L_t (skeleton) is shown in the inset of Fig. 5.

Results

We aim at providing a versatile nematode segmentation framework with performances comparable to, or better than, state-of-the-art image analysis systems [27,30,31,33–36,40,41]. To compare the MEME framework against such systems, we evaluate our algorithm quantitatively for a series of image sequences obtained for various *C. elegans* locomotive environments. The sequences include one or more data sets for behavioral assays such as (i) crawling on substrates (Video S2 and S3), (ii) swimming in a drop (Video S4), (iii) swimming in shallow acrylic channels (Video S5), (iv) locomotion in a gelatin-based solution (Video S6), and (v) locomotion in a microfluidic substrate (Video S7). A total of 13 image sequences are investigated (see Table S1 for complete listing and data source). In each sequence, the target nematode is present in all images. The MEME framework is implemented using Matlab; computing the nematode segmentation and skeleton for a 640×480 pixel size image requires approximately 1 second on a standard PC (i.e. 2.0 Ghz).

The state-of-the-art method of choice for comparison with MEME is an in-house developed thresholding algorithm [18,19], similar to standard intensity-based threshold approaches [30,31,35,36,40]. To perform a fair comparison between MEME and the thresholding framework, both methods are initially provided with a single image to tune their respective parameters.

As described for MEME (see Methods), the user selects from the initial image (i) the nematode region and (ii) the nematode width (Video S1). For the threshold-based method, all images of a sequence are first used to compute a background image of the environment by pixel averaging. Background subtraction is then applied to each image. Next, several thresholds are used to prune the remaining background pixels. These are manually selected by optimizing segmentation results on the initial image (Table S1). Finally, opening and closing morphological operators are used to smooth and discard final background regions. Note that in the case where the number of images in the sequence is small, background subtraction is omitted and only threshold intensities are used.

In order to quantitatively evaluate any segmentation algorithm, results must be compared to a *ground truth* [44]. For the present purpose, the ground truth is set as the true, or optimal, nematode segmentation provided by an expert. Hence, we manually segment a small set of images ($n=30-40$) from each sequence (e.g. Fig. 6, second row) and compare the performance of each algorithm to this image sub-set. Determining such ground truth allows for a precise definition of correct and incorrect pixel classification.

The performance of a segmentation algorithm can be evaluated by measuring two distinct metrics [44]: (i) the *surface error* and (ii) the *nematode yield*. The former quantity computes the proportion of pixels which are misclassified by the algorithm over the entire image. This metric is mathematically defined as

$$\epsilon_t = \frac{1}{|S_t|} \sum_{p \in S_t} |G_t(p) - S_t(p)|, \quad (7)$$

where p is a pixel location and G_t is the ground truth image for I_t . Hence, ϵ_t attributes equivalent weight to errors on the

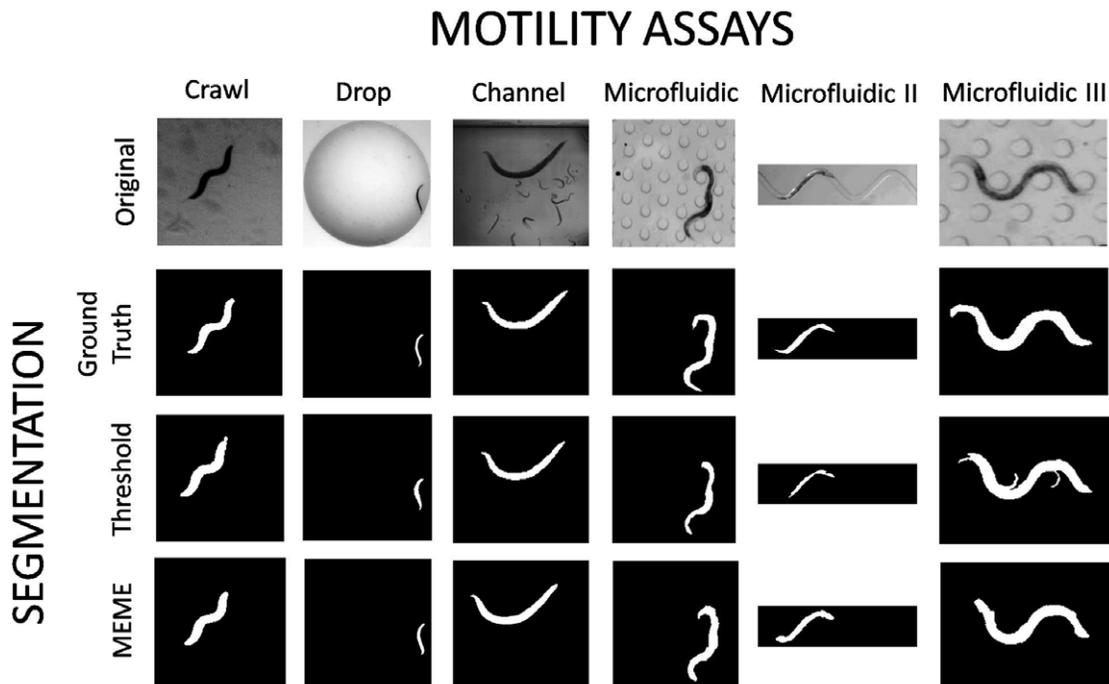


Figure 6. Nematode segmentation for various locomotive environments. (top row) Snapshots of raw images are respectively shown for crawling (Video S2), swimming in a drop (Video S4), swimming in a channel (Video S5), and locomotion in various microfluidic substrates (source: Supplemental Videos in Lockery *et al.* [21]). Comparisons between nematode segmentations are respectively shown for (i) the ground truth, i.e. hand-segmented nematodes (second row), (ii) a threshold-based approach [18,19] (third row), and (iii) the Multi-Environment Model Estimation (MEME) algorithm (bottom row). See Table S1 for data on all 13 cases investigated. doi:10.1371/journal.pone.0011631.g006

background and the nematode regions. The nematode yield, however, only computes the proportion of the nematode region which is correctly segmented. Consequently, misclassified pixels belonging to the background have no impact on the nematode yield. This metric is defined as

$$\eta_t = \frac{1}{|\mathcal{N}|} \sum_{p \in \mathcal{N}} |G_t(p) - S_t(p)|, \quad (8)$$

where \mathcal{N} is the set of pixels which satisfy $G_t(p) = 1$. Together, ϵ_t and η_t provide a quantitative and reliable measure of segmentation performance [44].

Qualitative segmentation results are shown for a selection of motility environments in Fig. 6 as well as in the Videos S2 to S7. In general, the MEME method is capable of segmenting nematodes at least as well as the thresholding method. For the “Crawl”, “Drop”, and “Channel” environments (first three columns, Fig. 6), both methods yield qualitatively similar results. Here, the environments illustrate a relatively homogenous background. However, in the complex “Microfluidic” environments (Fig. 6), results contrast more sharply between the two approaches; MEME provides cleaner segmentations which capture more closely the original shape of the nematodes.

Figure 7 shows the results from the computation of the surface error (Fig. 7a) and the nematode yield (Fig. 7b) for the environments of Fig. 6. Data for ϵ_t and η_t is reported in Table

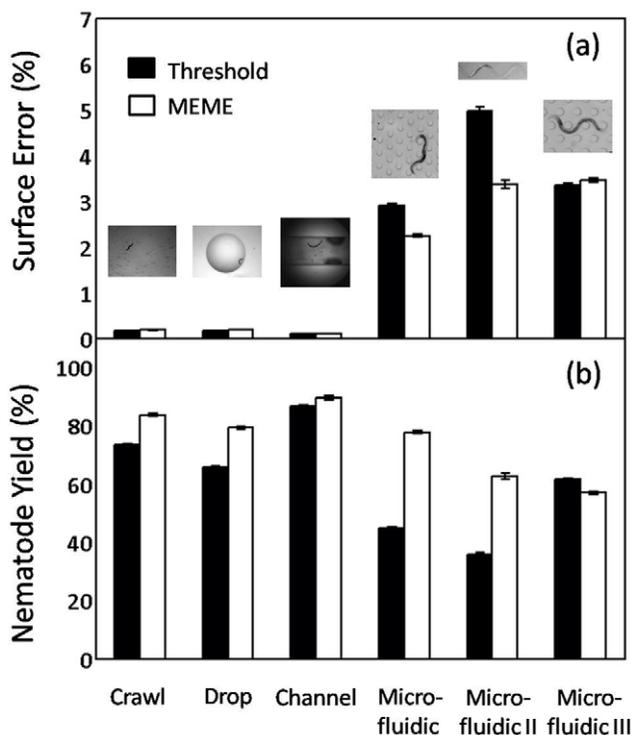


Figure 7. Performance evaluation of nematode segmentation algorithms. Here, the Multi-Environment Model Estimation (MEME) algorithm is compared to a state-of-the-art thresholding approach [18,19] for the environments shown in Fig. 6. (a) Surface error (ϵ_t): proportion of pixels which are misclassified by an algorithm over the entire image (see Eq. (7)). (b) Nematode yield (η_t): proportion of the nematode region which is correctly segmented (see Eq. (8)). Complete data on surface error and nematode yield is available in Table S1 for all 13 cases investigated.

doi:10.1371/journal.pone.0011631.g007

S1 for the complete 13 image sequences. In general, computations of surface error (ϵ_t) illustrate comparable performances between MEME and the threshold-based approach (Fig. 7a). Yet, in two complex “Microfluidic” environments, MEME does significantly better. Note that for all environments investigated here (Fig. 7a and Table S1), ϵ_t remains below 10%. In fact, for homogeneous background environments such as “Crawl”, “Drop”, and “Channel”, $\epsilon_t \ll 1\%$, emphasizing the good results obtained both by MEME and the threshold-based approach.

We observe, in contrast, significant improvements in nematode yield (η_t) when using MEME compared to the thresholding method (Fig. 7b). From the set of 13 assays tested here, 10 cases show examples of MEME significantly outperforming the threshold-based method (Table S1); in some cases, with margins greater than 20 percentage points (e.g. “Microfluidic” and “Microfluidic II”, Fig. 7b). In the remaining environments where the thresholding method performs relatively better (e.g. “Microfluidic III”, Fig. 7b), the differences in η_t remain however small, *i.e.* between 1.79 and 4.71 percentage points. Overall, our MEME algorithm outperforms the threshold-based approach for the overwhelming majority of cases examined.

Discussion

Our experiments show that MEME provides a reliable framework to obtain nematode segmentations of *C. elegans* across various locomotive environments. In addition, MEME offers significant improvements over alternative image analysis systems available; these include (i) better, or similar, performances compared to state-of-the-art thresholding approaches [18,19], (ii) no nematode-free image sequence required for learning background appearances [38,39]; and (iii) a small amount of user input needed, *i.e.* a single hand-segmentation of the nematode and a marking of its width (Video S1). This last improvement is particularly attractive from a user point of view as substantial effort may be needed with thresholding techniques to obtain similar results. Overall, these attributes make the MEME framework both attractive and straightforward to use for a broad range of researchers.

While computing good nematode segmentations with threshold-based methods is possible (Fig. 7), this process can quickly become laborious. Indeed, several iterations are required by the user to find optimal thresholds for a given environment (Table S1). The main complication arises from the non-uniform backgrounds and appearance (*i.e.* pixel intensities) which characterize many environments. For example, a single threshold is incapable of distinguishing between the nematode and the background in the presence of pillars in microfluidic substrates (e.g. Fig. 6, last column). Similarly, single thresholds cannot adapt to specific locations in an image. This becomes crucial for accurate segmentation of nematodes in environments where lighting conditions may not be uniform (e.g. Video S5).

In general, the improvement observed with MEME can be attributed to two main reasons: (1) the nematode appearance model is explicitly learned and used to help decide whether pixels belong to the nematode. In practice, when using threshold-based methods, many of the regions which are considered “not background” after applying a threshold do not resemble the nematode at all (e.g. pillars in “artificial dirt” assays of Lockery *et al.* [21], Fig. 1f and Video S7). Using both the nematode and background appearance models significantly reduces the need of using intense pruning schemes to reject such regions. (2) The background scene is partitioned into a grid of sub-regions (Fig. 4c), where each cell is explicitly modeled. This allows for local

variations in intensities to be grouped by region, providing a localized statistical model for each area of the background scene. This strategy has the advantage of appropriately modeling backgrounds where large variations in lighting occur (e.g. Fig. 1g).

In cases where the nematode appearance differs significantly from the background, such as in crawling and swimming assays (e.g. Fig. 1a and b), we observe nematode yields (η_t) beyond 80% (Fig. 7 and Table S1). In contrast, more complex environments can substantially reduce this performance (e.g. microfluidic substrates). The main difficulty therein lies in that only pixel intensities are modeled; this represents an important limitation when pixel intensities of the nematode and the background are too similar. A typical illustration of this problem occurs at the head and tail of *C. elegans*, where the nematode extremities are often transparent against the background. For example, this problem is observed in microfluidic substrates (e.g. Fig. 1e and f) where the ends of the nematode are lost during the segmentation process. A

direct consequence of this is the truncated length of nematode skeletons (e.g. inset of Fig 5 and Video S7).

Our MEME framework is currently optimized for segmenting a single target nematode within an image sequence. Nevertheless, scenarios where multiple nematodes enter the scene in subsequent images are still supported by our algorithm as long as only one nematode is present in the input image. That is, an arbitrary number of nematodes may be segmented for a given image sequence. Note, however, that cases where the appearance of either the nematode or the background changes significantly over the length of an image sequence will cause improper segmentations. Furthermore, extracting skeletons remains a challenge in some scenarios. For example, cases where the nematode coils on itself, or when its head and tail touch (e.g. omega bend), are currently not supported with the implemented skeleton algorithm. In the former case, the segmentation simply does not provide a correct shape representation of the nematode (i.e. a closed circle as opposed to a

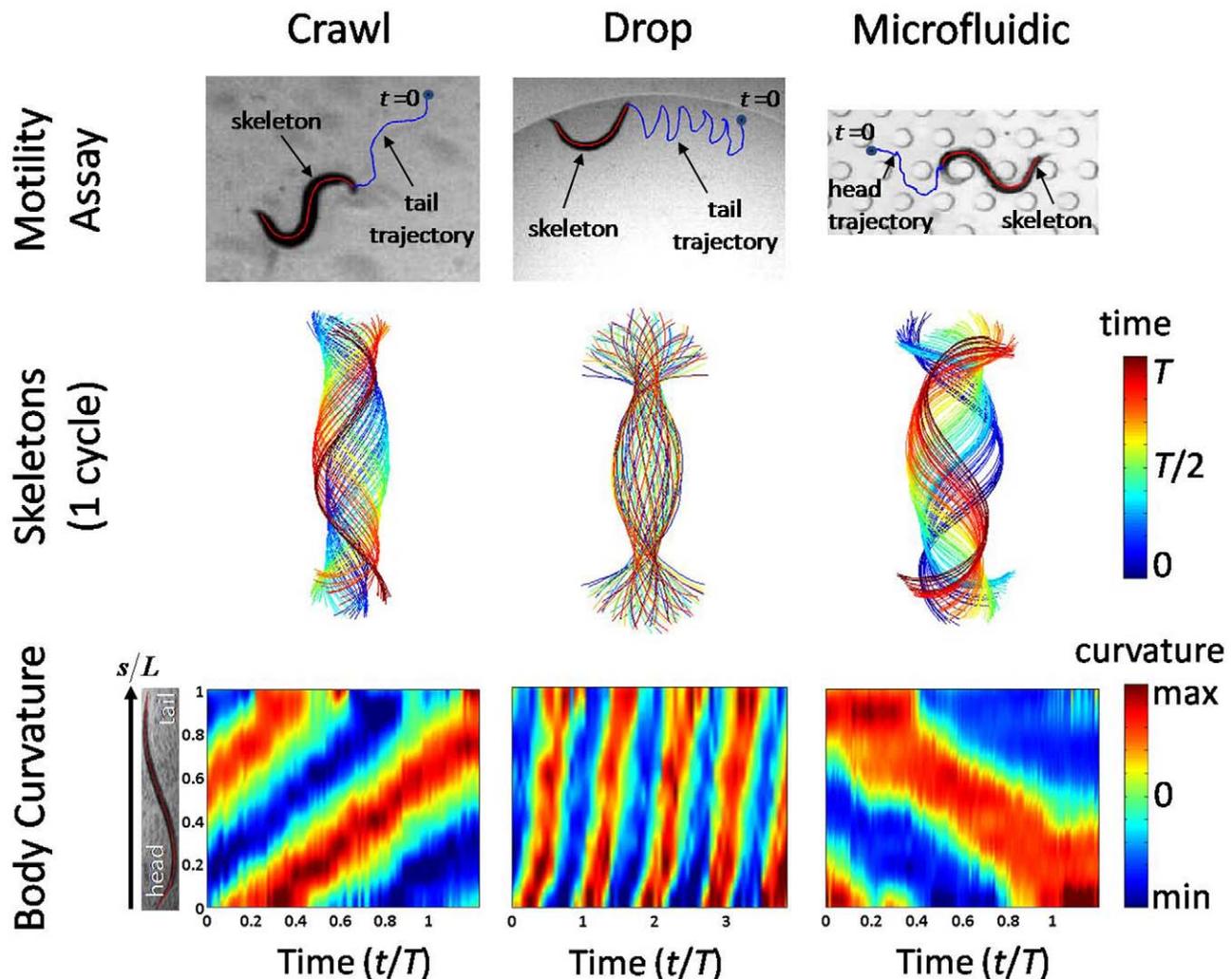


Figure 8. Examples of nematode locomotive features in sample environments. Here, nematode skeleton data is shown for a crawling assay (left column), for swimming in a 5 ml drop (middle column), and for locomotion in a microfluidic substrate obtained from Lockery *et al.* [21] (right column). Additional skeleton data is shown in Videos S2 to S7. **(top row)** Tracking data of path swept by nematode head (or tail) over multiple beating cycles. **(middle row)** Color-coded temporal evolution of *C. elegans* skeletons over one beating period (T). Results reveal a distinct envelope of body postures for each environment. **(bottom row)** Spatio-temporal contour plot of body curvature (κ) along the length of the nematode's skeleton. Red and blue colors represent positive and negative curvature values, respectively. The y -axis corresponds to the dimensionless position (s/L) along the *C. elegans'* body length where $s=0$ is the head and $s=L$ is the tail.
doi:10.1371/journal.pone.0011631.g008

‘snake’). The problem lies in the fact that estimating the medial axis of the nematode with the Chamfer distance transform is ill-suited. In principle, the latter scenario (*i.e.* omega bend) is not problematic. In practice, however, initializing the skeleton algorithm is ill-posed; there is no a priori knowledge as to where the head or tail lie.

Motility Metrics

We briefly discuss the feasibility of using nematode skeletons obtained with MEME (Videos S2 to S7). Skeleton data often provides the building blocks to quantify locomotive traits of *C. elegans* [11,13,16,18,19]. Here, we illustrate some of these motility metrics across sample environments. In Fig. 8 (top row), nematode tracking data is shown over multiple body bending cycles for crawling on a substrate (left column), swimming in a drop (middle column), and locomotion in so-called “artificial dirt” (right column), *i.e.* a microfluidic substrate (Supplemental Videos in Lockery *et al.* [21]). Trajectories swept by the nematode tail (or head) are labeled, illustrating striking differences in the travel paths adopted by *C. elegans* as a function of the surrounding environment. Snapshots of nematode skeletons over one beating cycle are shown in Fig. 8 (middle row); the time evolution of skeletons is color-coded as a function of the corresponding beating period (T). Plots reveal the existence of well-confined envelopes of body postures which vary dramatically with motility assay. Here, envelopes of postures are constructed using a principal component analysis (PCA) to find the skeleton’s principal axis and orientation at each instant in time. Further metrics including the nematode wavelength as well as the amplitude of body undulations can be obtained in a straightforward manner from the construction of such envelopes [18].

Next, we illustrate measures of body curvature (κ) along the nematode’s length (Fig. 8, bottom row); such plots have been shown to characterize swimming and crawling gaits [13,16,18]. Curvature is defined as $\kappa(s,t) = d\phi/ds$, where ϕ is the angle made by the tangent to the x -axis at each point along the nematode skeleton; s is the arc-length coordinate spanning the nematode’s head ($s=0$) to its tail ($s=L$). The spatio-temporal evolution of κ is shown over several beating cycles for each environment. Here, curvature values are color-coded; red and blue represent positive and negative values of κ , respectively. Note that the vertical axis in each contour plot corresponds to the non-dimensional body position (s/L), where L is the nematode length. Each contour plot shows the existence of highly periodic, well-defined diagonally oriented lines. These diagonal lines are characteristic of bending waves of motion which propagate in time along the nematode body length (*i.e.* traveling waves).

In Fig. 8, forward motion displays curvature lines with a positive slope (left and middle column); waves are initiated at the nematode head [13,18]. Conversely, backward motion displays lines with a negative slope, where bending motion is initiated at the tail (right column). In general, a number of motility metrics may be directly extracted from such curvature contour plots. For example, the body bending frequency (f) may be obtained by applying a one-dimensional (1D) Fast Fourier Transform (FFT) to the curvature field κ at multiple body positions s/L [18]. Similarly, the wave speed (c) may be directly extracted from the slope of the curvature κ propagating along the nematode’s body; the wavelength $\lambda = c/f$ is then computed in a straightforward manner. With our MEME platform, nematode skeleton data is made ready available for use for motility analysis of *C. elegans*.

Future Directions

The proposed MEME framework provides researchers with an attractive and reliable platform for nematode segmentation and

‘skeletonizing’ across a large spectrum of *C. elegans* motility assays. The MEME software is freely available upon request (contact person: J. Sznitman; website for download will be provided). Improving our system to further assist researchers conduct quantitative analysis of *C. elegans* is of course desired. In the near future, one immediate goal is to provide segmentations and skeletons simultaneously for multiple nematodes. This ‘upgrade’ would be of great interest for high-throughput drug screening applications [24,25]. From a performance point of view, combining larger sets of image features (*e.g.* edges, texture, etc.) with MOG models may provide better appearance models for difficult environments. This may yield better segmentation results, in particular at the nematode extremities (*i.e.* head and tail). Finally, our MEME platform is not restricted to image analysis of *C. elegans* only. For instance, MOG methods may be used for applications relating cell tracking and motility [53–55]. We illustrate here an example of such possible application with Albino Swiss mouse embryo fibroblast cells (Video S8).

Supporting Information

Table S1 Compiled data on segmentation results for the Multi-Environment Model Estimation (MEME) and threshold-based algorithms. Performances of each algorithm (*i.e.*, surface error and nematode yield) are evaluated for 13 different image sequences representative of various locomotive environments (*e.g.*, crawling on agar plate, swimming in a channel or a drop, locomotion in microfluidic substrates).

Found at: doi:10.1371/journal.pone.0011631.s001 (0.37 MB PDF)

Video S1 MEME software tutorial shown for a sample image sequence (source: berrigel2.0perc.mov, Supplementary Material in Berri *et al.* [2009]).

Found at: doi:10.1371/journal.pone.0011631.s002 (9.57 MB AVI)

Video S2 Example of crawling assay. From left to right: (i) raw image, (ii) binary segmentation from MEME, and (iii) resulting skeleton superimposed on raw image. Here, a young adult, wild-type (N2) *C. elegans* is seen crawling on an agar plate. Nematode is approximately 1 mm long (image resolution: 1/78 mm/pix; image acquisition rate: 28 frames per second).

Found at: doi:10.1371/journal.pone.0011631.s003 (9.57 MB AVI)

Video S3 Example of crawling on a substrate. From left to right: (i) raw image, (ii) binary segmentation from MEME, and (iii) resulting skeleton superimposed on raw image. The original data is obtained from the Supplementary Information (Movie 1) in Pierce-Shimomura *et al.* (2008).

Found at: doi:10.1371/journal.pone.0011631.s004 (0.87 MB AVI)

Video S4 Example of swimming assay in a liquid drop. From left to right: (i) raw image, (ii) binary segmentation from MEME, and (iii) resulting skeleton superimposed on raw image. Here, a young adult, wild-type (N2) *C. elegans* is seen swimming in a 5 ml drop of M9 buffer solution. Nematode is approximately 1 mm long (image resolution: 1/78 mm/pix; image acquisition rate: 28 frames per second).

Found at: doi:10.1371/journal.pone.0011631.s005 (9.54 MB AVI)

Video S5 Example of swimming assay in a shallow channel. From left to right: (i) raw image, (ii) binary segmentation from MEME, and (iii) resulting skeleton superimposed on raw image. Here, a young adult, wild-type (N2) *C. elegans* is seen swimming in a narrow acrylic channel filled with M9 buffer solution. Details are given in Sznitman *et al.* (2010). Nematode is approximately 1 mm

long (image resolution: 1/400 mm/pix; image acquisition rate: 125 frames per second).

Found at: doi:10.1371/journal.pone.0011631.s006 (8.27 MB AVI)

Video S6 Example of nematode locomotion in a gelatin-based solution. From left to right: (i) raw image, (ii) binary segmentation from MEME, and (iii) resulting skeleton superimposed on raw image. The original data is obtained from the Supplementary Information (berrigel2.0perc.mov) in Berri et al. (2009).

Found at: doi:10.1371/journal.pone.0011631.s007 (3.75 MB AVI)

Video S7 Example of nematode locomotion in “artificial dirt”, i.e., a microfluidic substrate. From left to right: (i) raw image, (ii) binary segmentation from MEME, and (iii) resulting skeleton superimposed on raw image. The original data is obtained from the Supplementary Information (Video 2) in Lockery et al. (2008).

Found at: doi:10.1371/journal.pone.0011631.s008 (1.20 MB AVI)

Video S8 Application of the MEME software to segmentation of Albino Swiss Mouse Embryo Fibroblast cells (3T3 Line). The original video (Video 2) is extracted from live-cell imaging videos

available at Nikon Microscopy U (<http://www.microscopyu.com/moviegallery/livecellimaging/3t3/index.html>), as obtained with Differential Interference Contrast (DIC) microscopy.

Found at: doi:10.1371/journal.pone.0011631.s009 (5.75 MB AVI)

Acknowledgments

Strains of *C. elegans* were obtained from the *Caenorhabditis elegans* Genetic Stock Center (University of Minnesota), supported by the National Institutes of Health (Bethesda MD, USA). The authors would like to thank Dr. R. Ghosh (Lewis-Sigler Institute for Integrative Genomics, Princeton U.), X. Shen (Department of Mechanical Engineering and Applied Mechanics, U. of Penn), and Dr. A.E.X. Brown (MRC Lab of Molecular Biology, Cambridge U.) for their support and helpful discussions.

Author Contributions

Conceived and designed the experiments: RS JS. Performed the experiments: RS MG. Analyzed the data: RS MG JS. Contributed reagents/materials/analysis tools: GH PA JS. Wrote the paper: RS JS.

References

- Brenner S (1974) The genetics of *Caenorhabditis elegans*. *Genetics* 77: 71–94.
- Rankin CH (2002) From gene to identified neuron to behaviour in *Caenorhabditis elegans*. *Nature Rev Genetics* 3: 622–630.
- Link EM, Hardiman G, Sluder AE, Johnson CD, Liu LX (2000) Therapeutic target discovery using *Caenorhabditis elegans*. *Pharmacogenomics* 1: 203–217.
- Jorgensen EM, Mango SE (2002) The art and design of genetic screens: *Caenorhabditis elegans*. *Nature Rev Genetics* 3: 356–369.
- Chamberlain JS, Benian GM (2000) Muscular dystrophy: the worm turns to genetic disease. *Curr Biol* 10: 795–797.
- Silverman GA, Luke CJ, Bhatia SR, Long OS, Vetica AC, et al. (2009) Modeling molecular and cellular aspects of human disease using the nematode *Caenorhabditis elegans*. *Pediatr Res* 65: 10–18.
- Sulston JE, Horvitz HR (1977) Post-embryonic cell lineages of the nematode *Caenorhabditis elegans*. *Dev Biol* 56: 110–156.
- Sulston JE, Schierenberg E, White JG, Thomson JN (1983) The embryonic cell lineage of the nematode *Caenorhabditis elegans*. *Dev Biol* 100: 64–119.
- White JG, Southgate E, Thomson JN, Brenner S (1986) The structure of the nervous system of the nematode *C. elegans*. *Phil Trans R Soc Lond B Biol Sci* 314: 1–340.
- CES Consortium (1998) Genome sequence of the nematode *C. elegans*: a platform for investigating biology. *Science* 282: 2012–2018.
- Karbowski J, Cronin CJ, Seah A, Mendel JE, Cleary D, et al. (2006) Conservation rules, their breakdown, and optimality in *Caenorhabditis* sinusoidal locomotion. *J Theor Biol* 242: 652–669.
- Pierce-Shimomura JT, Morse TM, Lockery SR (1999) The fundamental role of pirouettes in *Caenorhabditis elegans* chemotaxis. *J Neurosci* 19: 9557–9569.
- Pierce-Shimomura JT, Chen BL, Mun JJ, Ho R, Sarkis R, et al. (2008) Genetic analysis of crawling and swimming locomotory patterns in *C. elegans*. *Proc Natl Acad Sci USA* 105: 20982–20987.
- Tavernarakis N, Shreffler W, Wang S, Driscoll M (1997) unc-8, a deg/enac family member, encodes a subunit of a candidate mechanically gated channel that modulates *C. elegans* locomotion. *Neuron* 18: 107–119.
- Berri S, Boyle JH, Tassieri M, Hope IA, Cohen N (2009) Forward locomotion of the nematode *C. elegans* is achieved through modulation of a single gait. *HFSP J* 3: 186–193.
- Korta J, Clark DA, Gabel CV, Mahadevan L, Samuel ADT (2007) Mechanosensation and mechanical load modulate the locomotory gait of swimming *C. elegans*. *J Exp Biol* 210: 2383–2389.
- Ghosh R, Emmons SW (2008) Episodic swimming behavior in the nematode *C. elegans*. *J Exp Biol* 211: 3703–3711.
- Sznitman J, Purohit PK, Krajacic P, Lamitiina T, Arratia PE (2010) Material properties of *Caenorhabditis elegans* swimming at low Reynolds number. *Biophys J* 98: 617–626.
- Sznitman J, Shen X, Purohit PK, Arratia PE (2010) The effects of fluid viscosity on the kinematics and material properties of *C. elegans* swimming at low Reynolds number. *Exp Mech* DOI 10.1007/s11340-010-9339-1.
- Chronis N, Zimmer M, Bargmann CI (2007) Microfluidics for in vivo imaging of neuronal and behavioral activity in *Caenorhabditis elegans*. *Nature Methods* 4: 727–731.
- Lockery SR, Lawton KJ, Doll JC, Faumont S, Couthard SM, et al. (2008) Artificial dirt: microfluidic substrates for nematode neurobiology. *J Neurophysiol* 99: 3136–3143.
- Park S, Hwang H, Nam SW, Martinez F, Austin RH, et al. (2008) Enhanced *Caenorhabditis elegans* locomotion in a structured microfluidic environment. *PLOS One* 3: e2550.
- Qin J, Wheeler AR (2007) Maze exploration and learning in *C. elegans*. *Lab Chip* 7: 186–192.
- Rhode CB, Zeng F, Gonzales-Rubio R, Angel M, Yanik MF (2007) Microfluidic system for on-chip high-throughput whole-animal sorting and screening at subcellular resolution. *Proc Natl Acad Sci USA* 104: 13891–13895.
- Shi W, Qin J, Ye N, Lin B (2008) Droplet-based microfluidic system for individual *Caenorhabditis elegans* assay. *Lab Chip* 8: 1432–1435.
- Bessou C, Giuglia JB, Franks CJ, Holden-Dye L, Segalat L (1998) Mutations in the *Caenorhabditis elegans* dystrophin-like gene *dys-1* lead to hyperactivity and suggest a link with cholinergic transmission. *Neurogenetics* 2: 61–72.
- Baek JH, Cosman P, Feng Z, Silver J, Schafer WR (2002) Using machine vision to analyze and classify *Caenorhabditis elegans* behavioral phenotypes quantitatively. *J Neurosci Meth* 118: 9–21.
- Buckingham SD, Sattelle DB (2008) Strategies for automated analysis of *C. elegans* locomotion. *Invert Neurosci* 8: 121–131.
- Buckingham SD, Sattelle DB (2009) Fast, automated measurement of nematode swimming (thrashing) without morphometry. *BMC Neuroscience* 10: 84.
- Cronin CJ, Mendel JE, Mukhtar S, Kim YM, Stürb RC, et al. (2005) An automated system for measuring parameters of nematode sinusoidal movement. *BMC Genetics* 6: 5.
- Feng Z, Cronin CJ, Wittig JH, Sternberg PW, Schafer WR (2004) An imaging system for standardized quantitative analysis of *C. elegans* behavior. *BMC Bioinformatics* 5: 115.
- Geng W, Cosman P, Berry CC, Feng Z, Schafer WR (2004) Automatic tracking, feature extraction and classification of *C. elegans* phenotypes. *IEEE Trans Biomed Eng* 51: 1811–1820.
- Huang KM, Cosman P, Schafer WR (2006) Machine vision based detection of omega bends and reversals in *C. elegans*. *J Neurosci Meth* 158: 323–336.
- Huang KM, Cosman P, Schafer WR (2008) Automated detection and analysis of foraging behavior in *Caenorhabditis elegans*. *J Neurosci Meth* 171: 153–164.
- Hoshi K, Shingai R (2006) Computer-driven automatic identification of locomotion states in *Caenorhabditis elegans*. *J Neurosci Meth* 157: 355–363.
- Tsibidis GD, Tavernarakis N (2007) Nemo: a computational tool for analyzing nematode locomotion. *BMC Neuroscience* 8: 86.
- Ramot D, Johnson BE, Berry TL, Carnell L, Goodman MB (2008) The parallel worm tracker: a platform for measuring average speed and drug-induced paralysis in nematodes. *PLOS One* 3: e2208.
- Roussel N, Morton CA, Finger FP, Roysam B (2007) A computational model for *C. elegans* locomotory behavior: application to multiworm tracking. *IEEE Trans Biomed Eng* 54: 1786–1797.
- Tschepenkis G, Bianchi L, Metaxas DN, Driscoll M (2008) A novel computational approach for simultaneous tracking and feature extraction of *C. elegans* populations in fluid environments. *IEEE Trans Biomed Eng* 55: 1539–1549.
- Geng W, Cosman P, Baek JH, Berry CC, Schafer WR (2003) Quantitative classification and natural clustering of *Caenorhabditis elegans* behavioral phenotypes. *Genetics* 165: 1117–1126.
- Huang KM, Cosman P, Schafer WR (2009) Using articulated models for tracking multiple *C. elegans* in physical contact. *J Sign Process Syst* 55: 113–126.
- Stauffer C, Grimson W (1999) Adaptive background mixture models for real-time tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition*. pp 246–252.
- Piccardi M (2004) Background subtraction techniques: a review. In: *IEEE International Conference on Systems, Man and Cybernetics*. pp 3099–3104.

44. Sznitman R, Lin H, Gupta M, Hager G (2009) Active background modeling: Actors on a stage. In: IEEE 12th International Conference on Computer Vision, Workshop on Visual Surveillance. pp 1222–1228.
45. Fleuret F, Blanchard G (2005) Pattern recognition from one example by chopping. In: Proceedings of the Neural Information Processing Systems conference (NIPS). pp 371–378.
46. Fei-Fei L, Fergus R, Perona P (2006) One-shot learning of object categories. IEEE Trans Pattern Analysis and Machine Intelligence 28: 594–611.
47. Laird NM, Rubin DB, Dempster AP (1997) Maximum likelihood from incomplete data via the em algorithm. J Roy Stat Soc B 39: 1–38.
48. Kalman RE (1960) A new approach to linear filtering and prediction problems. ASME J Basic Eng 82: 35–45.
49. Isard M, Blake A (1998) CONDENSATION—conditional density propagation for visual tracking. Int J Comp Vis 29: 5–28.
50. Sznitman R, Jedynak B (2010) Active testing for face detection and localization. IEEE Transactions on Pattern Analysis and Machine Intelligence 99: in press.
51. Barrow HG, Tenenbaum JM, Bolles RC, Wolf HC (1977) Parametric correspondence and chamfer matching: Two new techniques for image matching. In: Proceedings of the 5th International Joint Conference on Artificial Intelligence. pp 659–663.
52. Gavrilu DM (2007) A bayesian, exemplar-based approach to hierarchical shape matching. IEEE Trans on Pattern Analysis and Machine Intelligence 29: 1408–1421.
53. Dublin-Thaler BJ, Hofman JM, Cai Y, Xenias H, Spielman I, et al. (2008) Quantification of cell edge velocities and traction forces reveals distinct motility modules during cell spreading. PLOS One 3: e3735.
54. Li K, Chen M, Kanade T, Miller ED, Weiss LE, et al. (2008) Cell population tracking and lineage construction with spatiotemporal context. Med Image Anal 12: 546–566.
55. Xiong G, Feng C, Ji L (2006) Dynamical Gaussian mixture model for tracking elliptical living objects. Patt Rec Lett 27: 838–842.