

AUTOMATED TRACKING OF MULTIPLE *C. ELEGANS* WITH ARTICULATED MODELS

Kuang-Man Huang, Pamela Cosman, and William R. Schafer[†]

Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA, 92093-0407, USA.

Email: kmhuang@ucsd.edu, pcosman@ucsd.edu

[†]Division of Biological Sciences, University of California at San Diego, La Jolla, CA, 92093-0349, USA.

Email: [†]wschafer@ucsd.edu

Abstract

We present a method for tracking and distinguishing multiple *C. elegans* in a video sequence, including when they are in physical contact with one another. The worms are modeled with an articulated model composed of rectangular blocks, arranged in a deformable configuration represented by a spring-like connection between adjacent parts. Dynamic programming is applied to reduce the computational complexity of the matching process. Our method makes it possible to identify two worms correctly before and after they touch each other, and to find body poses for feature extraction.

Index Terms: articulated model, *C. elegans*, computer vision, dynamic programming, image processing.

1. Introduction:

The nematode *Caenorhabditis elegans* is widely used for studies of nervous system function and behavior. The *C. elegans* nervous system has a small number of neurons, all of which have been individually identified and characterized. Moreover, the ease of genetic manipulation in these animals makes it straightforward to isolate mutant strains with abnormalities in behavior and rapidly identify the mutant gene by molecular cloning. However, in order to rigorously study the relationship between genes and behavior in *C. elegans*, precise quantitative assays for behaviors such as locomotion, feeding and egg-laying are required. Because some of these behaviors occur over long time scales that are incompatible with real-time scoring by a human observer, automated systems consisting of a tracking microscope and image processing software have been developed. Some of these systems [1][2][3] track individual worms at high magnification, while others track multiple worms at lower magnification [4][5].

Both types have disadvantages. Single-worm systems provide a considerable amount of information about each animal recorded, but since statistically-significant characterization of any worm type requires the analysis of multiple animals, collecting data one animal at a time is slow. Multiple-worm recordings do not typically provide as much information due to their lower magnification. In addition, in existing multi-worm systems, any time two individuals touch, their individual identities are lost by the system, and when the animals separate, the system is unable to determine the correspondence between individuals before and after touching. This limits the ability of a multi-worm system to characterize the behavior of an individual in a population over time. Moreover, some behaviors of significant interest to researchers, such as mating and social feeding [5], involve physical interaction between animals. For any automated system to be useful in characterizing these behaviors, it is essential that the position and

body posture of a worm can be followed during and after physical contact with another animal.

Here we describe a new method for tracking multiple *C. elegans* that resolves the individual body postures of two worms in physical contact with one another by using a modeling algorithm. Fischler and Elschlager introduce an articulated model with all parts arranged in a deformable configuration which is represented by connections between pairs of parts [6]. This articulated model was recently used to track an individual person in videos [7].

Tracking multiple people separately in videos is also achievable using articulated models. However, videos used to study behaviors of *C. elegans* are usually grayscale or binary thus we cannot use properties such as color to separate two connected worm bodies. A detailed geometric model combined with non-linear estimation techniques was used in [8] to track multiple *C. elegans* to study mating behavior. In this paper, we use a part-based articulated model to match the bodies of *C. elegans* and combine this with a dynamic programming algorithm to determine the correct location of two individual worm bodies. We identify two worms correctly before and after they touch each other, and find the body poses for both worms for extraction of motion features.

2. Materials and methods:

Culture methods and image acquisition: Routine culturing of *C. elegans* was performed as described [9]. All worms analyzed in these experiments were young adults. Experimental animals were allowed to acclimate for 5 minutes before their behavior was analyzed. A single drop of a saturated LB (Luria broth) culture of *E. coli* strain OP50 was spotted onto a fresh NGM (nematode growth medium) agar plate and allowed to dry for 30 minutes before use. The worms used in these experiments were *npr-1(ky13)* mutants. Unlike the laboratory standard N2 strain which is a solitary feeder, tending to disperse on encountering bacterial food, *npr-1(ky13)* mutants are social feeders, strongly aggregating together, thus providing an opportunity to study touching behavior.

C. elegans locomotion was tracked with a Zeiss Stemi 2000-C microscope mounted with a High Performance CCD video camera essentially as described [3]. To record the locomotion of animals, an image frame of the animals was captured every 0.125 second (8Hz). The recording continues until they touch and then move apart. The length of every video is different, ranging from 30 sec to 178 sec because the time length for each pair of worms to aggregate and separate is different. A local thresholding algorithm was applied to generate binary images [3].

Worm Model: We model a worm as being composed of N rectangular parts with length L and width W in the ratio 2:1. Let l and w be the average length and width of the worm body

calculated from all non-touching frames. Typical values of l and w were approximately 95 and 6 pixels. Then we set $W = 0.9w$ and $L = 2w$ and $N = \text{round}(l/L)$. In this experiment, W ranged from 4 to 8 pixels (L ranged from 8 to 16 pixels) and N ranged from 6 to 9 parts. The position of each part can be defined by the triple (x, y, θ) , which specifies the coordinate of the center and the orientation of the part (Fig. 1a). Adjacent parts are connected by two joint points (Fig. 1b), which may coincide (Fig. 1c) or might not, as discussed later. When the N rectangular parts composing the worm body model all have a (x, y, θ) determined, we refer to this as a worm body pose.

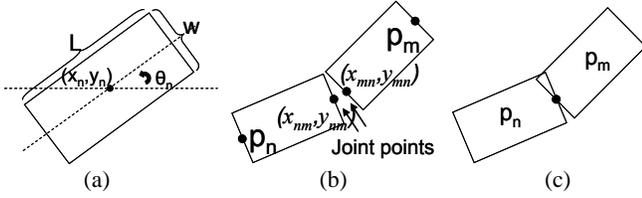


Figure 1: (a) One rectangular part and its parameters, (b) two parts of the model and their joint points, (c) two parts with joint points coinciding.

We seek to find the best match of the worm model to the binary worm image data. The concept of best match incorporates how well the rectangular parts fit the image pixel data, and fit with each other into a smooth worm body (for example, adjacent parts should not have large gaps between their joint points) [6][8], as discussed below. We first consider how well a rectangular part fits the image pixel data. The match cost $m(I, p_i)$ of a part p_i with 12 different possible orientation angles ($15^\circ, 30^\circ, 45^\circ, \dots, 180^\circ$) at every possible integer position (x, y) can be computed by convolving the binary worm image I with a convolution kernel composed of a “match” rectangle with different orientation angles (with the same size as our rectangular part) embedded in a larger “no match” rectangle [8]. The entries of this convolution kernel are:

$$k(x, y) = \begin{cases} -1 & , \text{ if } (x, y) \in \text{No Match} \\ \frac{WL}{S} e^{-\left(\frac{2x^2}{W^2} + \frac{2y^2}{L^2}\right)} & , \text{ if } (x, y) \in \text{Match} \end{cases}$$

where $S = \sum_{(x,y) \in \text{Match}} e^{-\left(\frac{2x^2}{W^2} + \frac{2y^2}{L^2}\right)}$, and (x, y) is the relative coordinate

to the center of the kernel $\left(-\frac{W}{2} \leq x \leq \frac{W}{2}, -\frac{L}{2} \leq y \leq \frac{L}{2}\right)$.

For each image frame, we generate a list of 1000 plausible worm body poses. These are the ones which have the lowest values of the match cost plus the deformation cost. The deformation cost measures how each worm body pose agrees with worm body anatomy. The pairwise deformation cost is defined as:

$$d(p_m, p_n) = W_x \times |x_{m,n} - x_{n,m}| + W_y \times |y_{m,n} - y_{n,m}| + W_\theta \times |\theta_m - \theta_n| \quad (1)$$

where $x_{m,n}, x_{n,m}, y_{m,n}$ and $y_{n,m}$ are the xy coordinates of joint points between adjacent parts p_m and p_n (Figure 1a). The angle θ_m is the orientation angle of the part p_m . W_x, W_y and W_θ are weights for the cost associated with a horizontal offset between joint points of adjacent parts, a vertical offset between joint points of adjacent

parts, and a difference in the orientation angle between the two parts. We use W_x, W_y and W_θ in the ratio 4:4:3. The deformation cost attains its minimum value of 0 when two parts have the same orientation angle and the joint points between them coincide.

Dynamic Programming: To make it less difficult computationally to find the best match, we use a dynamic programming algorithm. Dynamic programming is a class of methods for solving sequential decision problems with a compositional cost structure in which the decisions at one stage become the conditions governing the succeeding stages [10]. We use a dynamic programming algorithm to solve sequential decision problems at two different levels. First, within each frame, we seek to find the best position of a model part given the position of the previous parts. Next, looking across frames, we seek to find the best body pose in the current frame given the body poses in the previous frames. We try to minimize the energy defined by both the match quality of the model and the restrictions between pairs of parts [6][11]. The model contains N rectangular parts. We suppose the general cost function of each part p_i can be expressed by:

$$E_i = d(p_{i-1}, p_i) + m(I, p_i) + E_{i-1} \quad \text{for } i = 1 \text{ to } N-1 \quad (2)$$

where $d(p_i, p_{i+1})$ defined in equation (1) measures how much the adjacent parts p_i and p_{i+1} contribute to the deformation cost, and $m(I, p_i)$ measures how well the part p_i matches the image I with its current position. To generate our list of 1000 plausible body poses, we begin by randomly sampling a (x_0, y_0, θ_0) triple 1000 times from the list of 3000 most plausible pixel positions. These are the positions which have the lowest match cost to place our root part p_0 , which is one of the two end parts (it could be either the head or the tail). The lowest match cost for placement of one single part will correspond almost surely to one of the two ends because background pixels on 3 sides of the central “match” region will be correctly included in the “no match” region of the kernel, whereas for a typical part in the middle of the worm, background pixels only on 2 sides of the central “match” region will appropriately correspond with the “no match” region of the kernel. The energy E_0 of part p_0 is just the match cost at the position because p_0 is the first part of the model. E_0 is then used as the initial condition of the cost function E_i of the next part p_i . We place part p_i at location (x_i, y_i) with orientation angle θ_i where the triple (x_i, y_i, θ_i) minimizes the cost E_i in equation (2). We continue until the other end of the model p_{N-1} is reached. Then the best configuration of the final part is the one minimizing the cost E_{N-1} , which contains the cost of the whole worm body pose.

Multi-worm Match Algorithm: In our videos, the two worms start out separated. Each video can be divided automatically into subsections which are of two types: one type where the two worms are clearly separated (the distance between the worm centroids is longer than the worm length) and the other type where the worms are close to or touch each other.

For any subsection of the video in which the two worms are clearly separated, after 1000 possible worm poses are composed in each frame, we apply a dynamic programming algorithm for the second time to find the best temporal sequence of poses for the first worm within that first separated section of the video. Then we remove the first worm from the images and repeat the dynamic programming to find the best sequence for the second worm.

For the close/touching portion of the video, the area where two worms touch is thicker than other areas (Fig. 2a). We first erode to remove non-touching worm body parts (Fig. 2b), then we subtract the eroded image from the original binary image to obtain a new image with the worms partially separated (Fig. 2c).

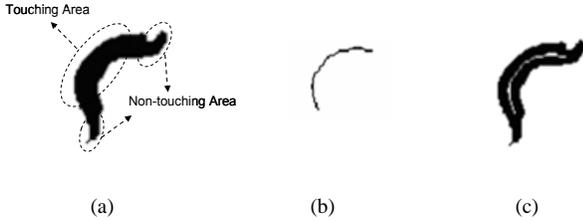


Figure 2: (a) Two worms in the original binary image, (b) the image after eroding, (c) the new image with two worms partially separated.

Dynamic programming reduces complexity, and therefore we use it where the worms do not touch, which is the majority of the video, as well as the portion where the body-fitting is easier. Where worms are close or touching, we abandon dynamic programming in favor of achieving the best body pose determination that we can, at the expense of greater complexity. For each frame, we seek to find the best body pose for that individual frame, using match cost, deformation cost and the distance from the pose in the previous frame. One of the two worms will be chosen as the primary worm. First a group of 1000 plausible body poses is formed by randomly sampling from the 3000 most plausible pixel positions in the image. Then the 10 best worm body poses will be chosen from the group for the primary worm according to a cost function L that combines the match cost of the whole worm body pose m_{total} , which is equal to E_{N-1} in equation (2), and the Euclidean distance d_{total} between the current and the previous worm body poses:

$$L = W_d \times d_{total} + W_m \times m_{total}$$

In this paper, we use W_d and W_m in the ratio $5e^2 : 1$ and the Euclidean distance between two body poses is calculated as:

$$d_{total} = \sum_{n=0}^{N-1} d(p_n^{i-1}, p_n^i)$$

where p_n^i is the n th part of the body pose in the i th image frame.

For each of the 10 best chosen primary worms, we find the one best secondary worm pose from those 1000 samples to fill the remainder of the object based on the overlapping cost. The overlapping cost is the sum of two terms: 1) the number of object pixels covered by both the primary worm body and the secondary worm body, 2) the number of object pixels not covered by any worm body. Then we choose the best overall result from those 10 worm sets. To avoid the whole result being dominated by only one worm, the assignment of the primary worm and the secondary worm alternates from one frame to the next.

As a final step for all frames of the video, we apply a 2D Gaussian filter to smooth the final results. For the reversal detection discussed later, after the best match configurations of both worms are decided, we manually assign one of the two end parts to be the head/tail part by using the original images. The manual assignment is done only once per video.

3. Results and conclusion:

The algorithm was run on 29 videos which contained 10579 frames. Using a 2.33 GHz Pentium-IV desktop computer it takes

less than 1 minute to process each frame. Some pictorial results are shown in Fig. 3. These illustrate the ability to identify two worms correctly before and after they touch. We have three goals: *A*) to find the best body poses for both worms for extraction of motion related features, *B*) to identify worms correctly before and after they touch and *C*) to detect reversals even when morphological skeletons are not available due to touching.

A) Good estimated pose and Motion-related features

We begin the evaluation of the algorithm by comparing how well it does against a manual fit of the body model frame by frame. The algorithm was tested on 750 images (including 153 non-touching frames and 597 touching frames) randomly chosen from

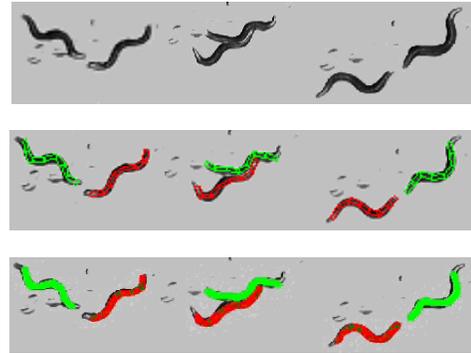


Figure 3: Three images show the best matching configuration. Images are frames 13, 111, 134 extracted from one video. The first row shows the original grayscale images, the second row shows overlays of the determined models and the original images, and the third row shows the matching result. The two worms are represented in different colors.

4 videos with a different pair of worms in each video. Given an original image and the number of parts N in it, we first chose $N+1$ joint points (including two end points) manually in every frame by clicking with the mouse on the image. Then these points are used to compute x , y and θ of each part and to build worm body poses. This is followed by Gaussian smoothing.

We compare the results from our algorithm to these manually built body poses and evaluate the accuracy by computing the correct percentage $= N_{AM}/N_A$, where N_{AM} is defined as the number of object pixels covered by both the automatically generated model and the manually generated model, and N_A is defined as the number of pixels covered by the automatically generated model.

For frames without touching, because the two worms are separated and each worm can be easily extracted to calculate its area, we also compare our matching results against both worm bodies in the original images. We compute predicted positive value $PPV = N_{MO}/N_M$ and true positive rate $TPR = N_{MO}/N_O$ where N_{MO} is the number of pixels covered by both the model (either manually or automatically generated) and the worm body in the original image, N_M is the number of pixels covered by the model, and N_O is the number of pixels covered by the worm body in the original binary image. Sample results are listed in Table 1. The correct percentages between automatically generated model and manually generated model range from 73% to 83%. PPV is also over 86% and TPR is higher than 75% for almost all frames. To reduce the possibility of two worms overlapping for touching frames, the width W of the model is chosen to be 90% of the actual body width. For this reason, the model tends to be covered by the whole worm body and the PPV is generally larger than the TPR. The PPV will

decrease and the TPR will increase if we use models with larger sizes. The results of the comparison between manually generated models and original images are also listed in the last two rows in this table, which clearly shows that our automatic matching algorithm outperforms human observers.

Table 1

Comparison results between automatically and manually generated models. The correct percentages between automatically generated model and manually generated model for the two worms are listed in rows 1 and 2. Predicted positive values and true positive rates for the two worms are listed in rows 3 and 4 (automatically generated model) and rows 5 and 6 (manually generated model).

File name		005	006	007	008
Automatically generated model against manually generated model	Non-touching	82.7% 77.5% (78)	77.8% 76.8% (25)	81.1% 81.3% (45)	74.7% 80.1% (5)
	touching	80.2% 73.8% (122)	73.3% 75.3% (176)	76.6% 78.4% (155)	76.0% 77.2% (144)
Automatically generated model (predicted positive)		89.5% 86.2%	92.4% 89.3%	87.9% 88.5%	91.2% 91.5%
Automatically generated model (true positive)		83.3% 77.9%	82.8% 78.3%	89.8% 88.2%	75.3% 82.4%
Manually generated model (predicted positive)		87.1% 86.2%	86.2% 88.8%	80.9% 85.2%	82.4% 85.9%
Manually generated model (true positive)		81.8% 76.6%	75.5% 76.7%	83.9% 85.1%	68.8% 77.2%

B) Correct identification

Both worms were tracked by a human observer to see if our method can correctly identify worms separately after they touch. Our algorithm identified both worms correctly in 26 of 29 videos.

C) Reversals

C. elegans usually moves in a sinusoidal wave. When a worm is touched or presented with a toxic chemical stimulus, it will switch the direction of the wave, causing the animal to instantaneously crawl backward instead of forward. This is a reversal. In [12], we used two skeleton points near the two ends as our reference points to decide if the worm was moving forward or backward. However, this requires skeleton points from each worm body which can not be obtained with a straightforward skeletonization algorithm when two worms touch and their bodies connect in one binary connected component. By using our modeling algorithm, after parameters of all parts are obtained with our algorithm, joint points can be considered to be pseudo skeleton points. We use the two joint points nearest the ends to be our reference points to detect reversals. There were 89 reversals detected manually by an expert observer in our 29 videos. Of these, our algorithm correctly detected 86 (96.6%). There were only 7 false alarms, so over 92% of detected events are actually reversals.

Conclusion: We combine articulated models and dynamic programming for simulating the body poses of *C. elegans* in multi-worm videos. The models are composed of a number of

rectangular parts arranged in a deformable configuration. We used a human observer to identify worms before and after they touch each other in 29 videos. The automatic algorithm matched this identification in 26 videos, for a correct rate of 90%. We accurately resolve the individual body postures of two worms in physical contact with one another. This modeling algorithm does not use color or texture as cues to separate worms, which makes it possible to be used on grayscale or binary videos. We also showed that reversal behaviors of multiple worms can be accurately detected by using our model even when their bodies are in physical contact with one another.

References:

- [1] J. Baek, P. Cosman, Z. Feng, J. Silver, and W. R. Schafer, "Using machine vision to analyze and classify *Caenorhabditis elegans* behavioral phenotypes quantitatively," *Journal of Neuroscience Methods*, vol. 118, pp. 9-21, 2002.
- [2] C. J. Cronin, J. E. Mendel, S. Mukhtar, Y. M. Kim, R. C. Stirbl, J. Bruck, and P. W. Sternberg, "An automated system for measuring parameters of nematode sinusoidal movement," *BMC Genetics*, vol. 6, pp. 5, 2005.
- [3] W. Geng, P. Cosman, C. Berry, Z. Feng, and W. R. Schafer, "Automatic tracking, feature extraction and classification of *C. elegans* phenotypes," *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 1811-1820, 2004.
- [4] R. Dhawan, D. B. Dusenbery, and P. L. Williams, "Comparison of lethality, reproduction, and behavior as toxicological endpoints in the nematode *Caenorhabditis elegans*," *Journal of Toxicology and Environmental Health Part A*, vol. 58, no. 7, pp. 451-462, 1999.
- [5] M. de Bono, D. M. Tobin, M. W. Davis, L. Avery and C. I. Bargmann, "Social feeding in *Caenorhabditis elegans* is induced by neurons that detect aversive stimuli," *Nature*, vol. 419, no. 6910, pp. 899-903, 2002.
- [6] D. Hogg, "Model based vision: a program to see a walking person," *Image and Vision Computing*, vol. 1, no. 1, pp. 5-20, 1983.
- [7] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient matching of pictorial structures," *IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head, USA*, pp. 66-73, June, 2000.
- [8] E. Fontaine, J. Burdick and A. Barr, "Automated tracking of *C. elegans*," *IEEE International Conference of Engineering on Medicine and Biology Society, New York, USA*, pp. 3716-3719, August, 2006.
- [9] S. Brenner, "The genetics of *Caenorhabditis elegans*," *Genetics*, vol. 77, pp. 71-94, 1974.
- [10] A. Amini, T. Weymouth, and R. Jain, "Using dynamic programming for solving variational problems in Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 9, pp. 855-867, 1990.
- [11] D. G. Lowe, "Fitting parameterized three-dimensional models to images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 5, pp. 441-450, 1991.
- [12] K. Huang, P. Cosman and W. R. Schafer, "Machine vision based detection of omega bends and reversals in *C. elegans*," *Journal of Neuroscience Methods*, vol. 158, pp. 323-336, 2006.