UC San Diego
UC San Diego Previously Published Works

Title

Using Articulated Models for Tracking Multiple C. elegans in Physical Contact

Permalink

https://escholarship.org/uc/item/7sw41894

Journal

Journal of Signal Processing Systems, 55(1-3)

ISSN

1939-8018 1939-8115

Authors

Huang, Kuang-Man
Cosman, Pamela
Schafer, William R

Publication Date

2008-06-07

DOI

10.1007/s11265-008-0182-x

Peer reviewed

Using Articulated Models for Tracking Multiple *C. elegans* in Physical Contact

**Kuang-Man Huang · Pamela Cosman ·
William R. Schafer**

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 the body poses for further feature extraction. All joint points in our model can be also considered to be the pseudo skeleton points of the worm body. It solves the problem that a previously presented morphological skeleton-based reversal detection algorithm fails when two worms touch each other. The algorithm has many applications in the study of physical interactions between *C. elegans*.

K.-M. Huang (✉) · P. Cosman
Department of Electrical and Computer Engineering,
University of California at San Diego,
La Jolla, CA 92093-0407, USA
e-mail: houston21@hotmail.com

W. R. Schafer
Cell Biology Division,
MRC Laboratory of Molecular Biology at Cambridge,
Cambridge CCB2 0QH, UK

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 and used to follow and analyze animals. Some of these systems [1–4] perform tracking of individual worms at high magnification, while others have been designed to track multiple worms at lower magnification [5, 6].

Both types of system currently in use have disadvantages for the collection of behavioral data. Single-worm systems can provide a considerable amount of information about each animal that is recorded, but since statistically-significant characterization of any worm type requires the analysis of multiple animals, collecting data one animal at a time is often frustratingly slow. On the other hand, multiple-worm recordings do not typically provide as much information as single worm recordings due to their lower magnification. In addition, in most existing multi-worm systems, any time two individuals touch, segmentation of separate animals is difficult and so their individual identities are lost by the system. When the animals separate, the system is unable to determine the correspondence

between individuals before and after touching. This is the problem we solve in this paper. The inability to separately segment and track individual animals when they touch seriously 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 [6, 7], by their very nature 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 makes it possible to accurately resolve the individual body postures of two worms in physical contact with one another by using a modeling algorithm. Model matching algorithms can be roughly divided into two categories [8, 9]. The first class is "contour based" which represents objects in terms of their boundaries. One popular approach is called active contours or "snakes", which deform elastically to fit the contour of the target structure [10]. Another method uses combinations of trigonometric functions with variable coefficients to represent objects [11, 12].

The second class consists of "appearance-based" approaches. In this case a model is used to simulate the complete appearance (shape, color, texture, etc.) of the target object in the image. Bajcsy and Kovacic [13] describe a volume model that deforms elastically to cover the object region. In [14], a model composed of a collection of parts is used to represent objects in terms of a constellation of local features. All parts in this model are constrained with respect to a central coordinate system. There have been other part-based modeling algorithms. Fischler and Elschlager [15] introduce an articulated model with all parts arranged in a deformable configuration which is represented by spring-like connections between pairs of parts. This articulated model has recently been used for tracking an individual person in videos [16, 17]. In [18], this method is further improved with an efficient algorithm for finding the best match of a person and a car in images. A number of methods to track a 3D human figure using articulated models also have been proposed. [19] uses a 3D articulated model combined with a modified particle filter to recover full articulated human body motion. [20] tracks a 3D human body with 2D contours using information from multiple cameras with different viewpoints. [21] presents an algorithm which projects 3D motion of a figure onto an image plane instead of using multiple cameras.

Multiple object tracking has been an intensive area of research due to its many applications [22–24]. However, traditional methods fail when the objects are in close proximity or present occlusions. A snake, for example, can be used to track individual deformable moving binary

objects. But when two binary objects touch each other, the boundary between them is not represented in the image data, and so a snake algorithm would likely fail without usable image information at the boundary. Tracking multiple objects separately in videos is achievable by using appearance-based models. Some methods track and separate people with occlusions by using cues such as appearance (color and texture of clothing, etc.) from frame to frame [25–28]. Some approaches are able to track multiple rigid objects in simple interactions. In [29], Khan uses particle filtering combined with a pairwise penalty function that only depends on the number of pixels which overlap between the appearance templates associated with the two targets to track multiple ants. In [30], Qu uses the "magnetic potential" and "inertia potential" to solve the "error merge" and "false object labeling" problems after severe occlusion of targets.

Although there have been numerous algorithms for multiple-object tracking, additional issues arise when tracking *C. elegans* worms. The worm tracking problem differs from previous tracking experiments in several respects: (1) The worms are not rigid bodies; they are highly deformable, (2) the actual worm body is transparent, and (3) the worm moves almost entirely in a 2D plane. The fact that the worm body is transparent means that color information is completely unavailable and even grayscale can be unreliable. Therefore we cannot use the color or texture to easily distinguish the animals from the background of transparent bacteria layers. To solve this technical problem, the illumination of the microscope is chosen to make the worm body look dark and the background look bright in grayscale images. This means the grayscale images are very nearly binary, and in fact we convert them to binary as a pre-processing step. While the binary nature of the images makes it relatively easy to compute body posture features for single-worm videos, it is a difficult problem to track and distinguish touching deformable binary blobs when there is more than one object in the scene. The fact that the worm moves almost entirely in a 2D plane also makes this tracking problem somewhat different from many prior studies. Although portions of a worm's body can cross over/under its own body or the body of another worm, the worm's body is sufficiently flat that such crossings do not involve appreciable curvature up and out of the plane of the agar plate. Thus the projection of the worm's body onto the plane of the plate has a roughly constant length.

These three significant differences make the worm-tracking problem quite unique compared to other multiple object tracking problems, such as those involving people and cars. Prior research on tracking multiple *C. elegans* is limited to [31] and [32], of which the latter is a preliminary version of the research presented here. In [31], a deform-

able worm model and several motion patterns are used to define an overall locomotory space of *C. elegans* and to describe its general dynamic movements. Using a combination of "predicted energy barrier" and multiple-hypothesis tracking, multiple touching worms can be identified and separated with a high success rate. The algorithm presented in [31] focuses on the crawling mechanism, which means sudden position-shift of the worm bodies caused by the swimming mechanism of the worm or even the movement of the stage may lead to loss of a track. Our work differs from [31] in that the algorithm presented here can accommodate both crawling and swimming mechanisms or other causes of sudden shifts such as stage movement. Our work also differs in that we introduce various pixel-based, feature-based, and human observation based methods for evaluating the accuracy of the body matching algorithm, and we evaluate some features (angle change rate, reversals) extracted from the matched body positions.

In this paper, we also choose an appearance-based method to match the bodies of *C. elegans*. We combine a part-based articulated model [18, 33] with a dynamic programming algorithm to determine the correct location of the individual worm bodies. Our work accurately resolves the individual body postures of two worms in physical contact with one another and identifies them correctly before and after they touch each other, and can still maintain track of the worms when their bodies have non-crawling displacement. Furthermore, we solve the problem that the skeleton-based reversal detection algorithm in [34] fails when two worms touch each other because of the difficulty of obtaining morphological

skeletons. The basic algorithms described here appeared in abbreviated form in a preliminary paper [32]. This longer version includes an expanded exposition of the algorithm, as well as completely new results (in Tables 2 and 4) and expansions to the earlier results presented in Tables 1 and 3. In particular, we now show that a parameter such as angle change rate (related to body curvature) can be extracted from the model, verifying that our algorithm accurately simulates worm body poses, and we also examine the rate of movement before and after touching, and we extract parameters during the time the animals touch (reversal rate, duration of touching). Our algorithm shows good performance in the analysis of real worm videos, and should have many applications in the study of *C. elegans* behavior.

2 Materials and Methods

2.1 Strains and Culture Methods

Routine culturing of *C. elegans* was performed as described [35]. All worms analyzed in these experiments were young adults; fourth-stage larvae were picked the evening before the experiment and tracked the following morning. Exactly two experimental animals were placed on a plate and they were allowed to acclimate for 5 minutes before their behavior was analyzed. Plates for tracking experiments were prepared fresh the day of the experiment; 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

Table 1 Comparison results between automatically and manually generated models.

File name		005	006	007	008	011	012	015	016	017	018	020
Automatically generated model against manually generated model	Non-touching (%)	82.7	77.8	81.1	74.7	77.8	79.6	74.1	75.3	79.8	72.1	82.4
		77.5	76.8	81.3	80.1	80.7	81.1	69.2	79.2	81.1	74.9	67.7
		(78)	(25)	(45)	(5)	(104)	(38)	(113)	(46)	(139)	(144)	(56)
	Touching (%)	80.2	73.3	76.6	76.0	78.2	76.0	76.6	77.8	79.4	80.2	79.4
		73.8	75.3	78.4	77.2	78.7	79.2	78.4	80.7	78.6	81.9	77.7
		(122)	(176)	(155)	(144)	(55)	(94)	(64)	(51)	(61)	(55)	(143)
Automatically generated model (predicted positive; %)		89.5	92.4	87.9	91.2	92.0	90.5	94.7	85.0	90.9	84.2	92.1
		86.2	89.3	88.5	91.5	87.5	86.5	80.5	90.9	90.2	85.4	86.4
Automatically generated model (true positive; %)		83.3	82.8	89.8	75.3	68.3	76.9	70.6	77.7	82.3	83.3	83.1
		77.9	78.3	88.2	82.4	83.5	83.0	74.5	80.4	81.7	80.1	73.7
Manually generated model (predicted positive; %)		87.1	86.2	80.9	82.4	92.4	88.4	88.2	86.6	84.9	80.7	87.5
		86.2	88.8	85.2	85.9	85.0	84.9	83.8	89.3	87.3	89.1	87.1
Manually generated model (true positive; %)		81.8	75.5	83.9	68.8	67.6	73.9	68.0	78.5	76.8	78.6	80.1
		76.6	76.7	85.1	77.2	80.0	80.4	72.0	75.9	78.3	76.9	71.3

The correct percentages between automatically generated model and manually generated model for the two worms are listed in rows 1–3 and 4–6. Predicted positive values and true positive rates for the two worms are listed in rows 7–8 and 9–10 (automatically generated model) and rows 11–12 and 13–14 (manually generated model). Values in parenthesis are the numbers of image frames.

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.

2.2 Acquisition of Image Data

C. elegans locomotion was tracked with a Zeiss Stemi 2000-C microscope mounted with a Cohu high performance CCD video camera essentially as described [4]. The microscope was outfitted for brightfield illumination from a 12 V 20 W halogen bulb reflected from a flat mirror positioned at an angle of approximately 45°. To record the locomotion of animals, an image frame of the animals was captured every 0.125 s (8 Hz) with a resolution of 640× 480 pixels and then saved as AVI video files. The microscope was fixed to the magnification of 2.5× during observation. In order to focus on the goal of identifying the body postures of two worms in physical contact, each video used in this experiment contains exactly two animals. For each video, the recording is initiated when the worms are separated. The recording continues until they touch and then move apart. The length of every video is different, ranging from 30 to 178 s because the time length for each pair of worms to aggregate and separate is different. All acquired images are grayscale with intensity level from 0 to 255 and contain exactly two animals of low intensity (dark) surrounded by bright background. The images are sometimes cluttered with other objects such as worm eggs, blobs of food, or irregularities in the surface of the agar. In order to save a great deal of computation and make further processing easier, a local thresholding was applied on the grayscale images by using a 5×5 moving window. The center pixel inside the moving window was assigned to 1 if the mean value of the window was less than 70% of the background pixel value or the standard deviation was larger than 30% of the mean value. Otherwise, the center pixel was assigned to 0 as background [4].
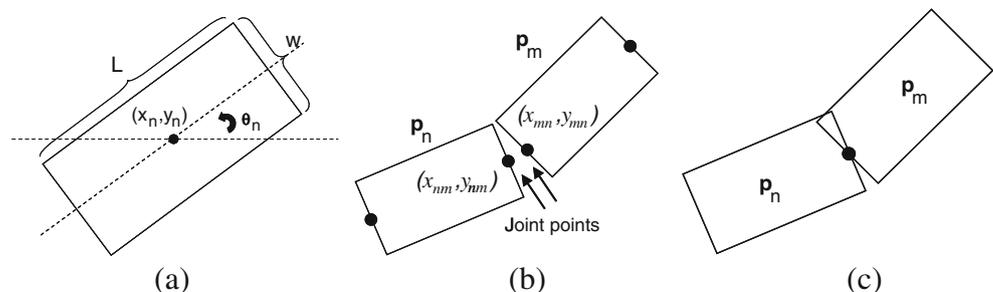
2.3 Worm Model

Because the *C. elegans* worm body is more or less cylindrical, in this paper, we model its projection as being composed of N rectangular parts with length L and width W in the ratio 2:1. To generate a more accurate and robust worm body model, parameters N, L and W are learned from the image data. They can differ for different input videos. For a given video, let l and w be the average length and width of the worm body calculated from all non-touching frames. These values are calculated automatically using the method described in [4]. Typical values of l and w were approximately 95 and 6 pixels respectively. Then we set $W= 0.9w$ and $L=2W$ and $N = \text{round}\left(\frac{l}{L}\right)$. In this experiment, W ranged from 4 to 8 pixels (L ranged from 8 to 16 pixels) and N ranged from six to nine parts. The position of each part in the image 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) but also might be chosen to not be coincident. When (x, y, θ) is determined for each of the N rectangular parts composing the worm body model, we refer to this as a worm body pose.

We seek to find the best match of the worm model to the actual binary worm image data. The concept of best match incorporates both how well the rectangular parts fit the image pixel data, and also how well the rectangular parts fit worm body anatomy. By "worm body anatomy" we mean how well the parts fit with each other into a smooth worm body (for example, adjacent parts should not have large gaps between their joint points) [15, 18, 33], as will be discussed in the next paragraphs.

We begin by considering how well a rectangular part fits the image pixel data and deterministically examining the set of K most plausible pixel positions from the object (K can vary based on different image resolutions used in different experiments). These are the positions which have the lowest match cost to place our rectangular part. The match cost $m(I,p_i)$ of a part p_i with 12 different possible orientation angles (15°, 30°, 45°,…, 180°) at every

Figure 1 **a** One rectangular part and its parameters, L and W are the length and width of the rectangle and (x, y, θ) specifies the coordinates of the center and the orientation of the part, **b** two parts of the worm model and their joint points, **c** two parts of the worm model with the joint points coinciding.

(a)   (b)   (c)

possible integer pixel 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 [18, 33]. The entries of this convolution kernel are defined by the following equation:

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

where $S = \sum\limits_{(x,y)\in\text{Match}} e^{-\left(\frac{2x^2}{W^2} + \frac{2y^2}{L^2}\right)}$, W and L are the width and length of a rectangular part and (x,y) are the coordinates relative to the center of the kernel $\left(\frac{-W}{2} \leq x \leq \frac{W}{2}, \frac{-L}{2} \leq y \leq \frac{L}{2}\right)$. In this convolution kernel, points close to the y axis $(x=0)$ have larger weights. Despite the fact that we set $W=0.9w$ which means that the rectangular model part has a width that is only 90% of the average width of the animal in non-touching frames, in some images, it is possible that the original worm body width is slightly smaller than the width W of the rectangular part. By using this kernel, pixel positions close to the middle of the body will have relatively smaller match cost than positions close to the edge of the body, which means the likelihood for a part to be placed along the middle line is larger than other possible positions. By exhaustively evaluating the match cost for each integer pixel position and each of the 12 orientation angles, we generate a list of the K most plausible positions for individual rectangular body parts.

For each image frame, as will be discussed in Section 2.3.2, we generate a list of M plausible body poses, from which we would like to choose the best ones for each of the two worms. These M poses 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 the following:

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

where $|.|$ denotes absolute value, and $x_{m,n}$, $x_{n,m}$, $y_{m,n}$ and $y_{n,m}$ are the xy coordinates of joint points between adjacent parts p_m and p_n (Fig. 1a). The angle θ_m is the orientation angle in degrees of the part p_m. W_x, W_y and W_θ are weights for the cost associated with a horizontal (x-direction) offset between joint points of adjacent parts, a vertical (y-direction) offset between joint points of adjacent parts, and a difference in the orientation angle between the two parts. The deformation cost attains its minimum value of 0 when two parts have the same orientation angle and the joint points between them coincide. In the next two sections, we show how to generate a list of M plausible body poses, which is decided by how well the model matches the object in the image and how well it lowers the deformation cost. To make finding the best match computationally efficient, we use a dynamic programming algorithm.

2.3.1 Dynamic Programming

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 [36, 37]. We try to minimize the cost function defined by both the match quality of the model and the restrictions between pairs of parts [15, 38]. The model contains N rectangular parts. We suppose the general cost function of each part p_i can be expressed by the following equation:

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

where p_i is defined by (x_i, y_i, θ_i) as previously described, $d(p_{i-1}, p_i)$ defined in Eq. 1 measures how much the adjacent parts p_{i-1} and p_i 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.

2.3.2 Worm Body Poses Sampling

To generate a list of body poses, we begin by taking the $\frac{K}{3}$ (x_0, y_0, θ_0) triples with the lowest match cost from the previously generated list of K positions, to place our first part p_0, which is one of the two end parts (it could be either the head or the tail). The reason why the lowest match cost for placement of one single part will correspond almost surely to one of the two ends is because background pixels on three 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 two sides of the central "match" region will appropriately correspond with the "no match" region of the kernel. Using only the one-third best positions of all K positions only for part p_0 slightly reduces complexity without significantly increasing the chance of generating bad worm body poses. After the part p_0, all K positions will be possible candidates to place parts from p_1 to p_{N-1}.

The cost function $E_0(p_0)$ of each part p_0 is just the match cost at the position because p_0 is the first part of the model. If $E_0(p_0)$ of each p_0 is known and we define $p_{i-1}(p_i)$ to be the best position for the part p_{i-1} as a function of the position of the next part p_i, then the best position of the part p_0 in the input image I is:

$$p_0(p_1) = \arg\min_{p_0}(d(p_0, p_1) + m(I, p_1) + E_0(p_0)) \quad (3)$$

That is, the best position of the part p_0 can be decided given the position of its next part p_1. The minimum cost function E_i^* (p_1) of the given part p_1 (minimized over all possible values of p_0) becomes:

$$E_1^*(p_1) = \min_{p_0}(d(p_0, p_1) + m(I, p_1) + E_0(p_0)) \qquad (4)$$

Based on the same concept, the best location of the part p_{i-1} ($1 < i < N$) as a function of the position of the next part p_i is:

$$p_{i-1}(p_i) = \operatorname*{argmin}_{p_{i-1}}\Big(d(p_{i-1}, p_i) + m(I, p_i) + E_{i-1}^*(p_{i-1})\Big)$$
$$(5)$$

and then E_i^* (p_i) will be:

$$E_i^*(p_i) = \min_{p_{i-1}}\Big(d(p_{i-1}, p_i) + m(I, p_i) + E_{i-1}^*(p_{i-1})\Big) \qquad (6)$$

We continue this forward process until the other end of the model p_{N-1} is reached. Finally for the part p_{N-1}, the best configuration is the one that has the minimum cost E_{N-1}.

$$p_{N-1opt} = \operatorname*{argmin}_{p_{N-1}}\Big(E_{N-1}^*(p_{N-1})\Big)$$

and the optimum positions of all parts can now be traced in the backward step from the part p_{N-1} to p_0 by using Eqs. 3, 4, 5 and 6. In our experiment, we chose the M configurations with minimum cost to be the plausible poses for each frame.

2.3.3 Multi-Worm Match

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

Type A For any subsection of the video in which the two worms are clearly separated, after M possible worm poses are composed in each frame, we apply a dynamic programming algorithm again *over the time domain* to find the best *temporal sequence of poses* that move smoothly for the first worm within that first separated section of the video. This time we try to minimize the cost function L that combines the match cost of the whole worm body pose m_{total}, which is equal to E_{N-1} in Eq. 2, and the Euclidean distance d_{total} between the current and the previous worm body poses:

$$L = W_{\text{d}} \times d_{\text{total}} + W_{\text{m}} \times m_{\text{total}} \qquad (7)$$

The weight W_{d} is associated with the distance between the worm body in one frame and in the previous one. The weight W_{m} is associated with the pixel mismatch between the pose of the worm body model in a frame and the actual foreground pixels in that frame. We set W_{m} to be 1, and then the selection of the W_{d} parameter adjusts the relative cost given to within-frame pixel mismatch compared to inter-frame pose mismatch. Based on limited experimentation by a human observer, W_{d} was chosen to be $5e^{-2}$. The Euclidean distance between two body poses is calculated as the following:

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

where p_n^i is the nth part of the body pose in the ith image frame. Then we remove the first worm from the images in the sequence and repeat the dynamic programming algorithm to find the best sequence for the second worm.

Type B For the close/touching portion of the video, we begin by using the fact that the area where two worms touch each other is thicker than other areas to divide touching worms (Figs. 2a, 2b). For any close/touching frame, in order to remove non-touching worm body parts, we first erode the touching worm body object $\frac{W}{2}$ times (where W is the width of a rectangular part in the body model) with a 3×3 structuring element (Fig. 2c), then we subtract the eroded image from the original binary image to get a new image. The two worms may be only partially separated in the new image (Fig. 2d). But this method will increase the chance of finding good worm body poses by heightening the match cost for those body poses over-

(a)   (b)
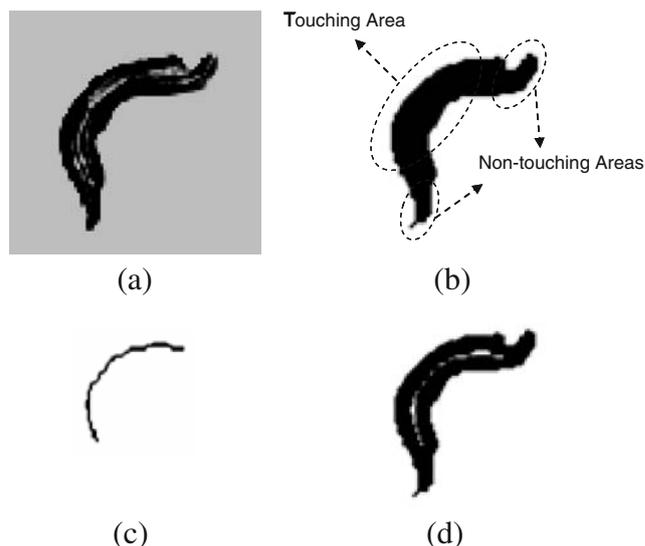
(c)   (d)

Figure 2 **a** Two worms in the original grayscale image, **b** two worms in the binary image, **c** the binary image after eroding, **d** image b minus image c shows the two worms partially separated.
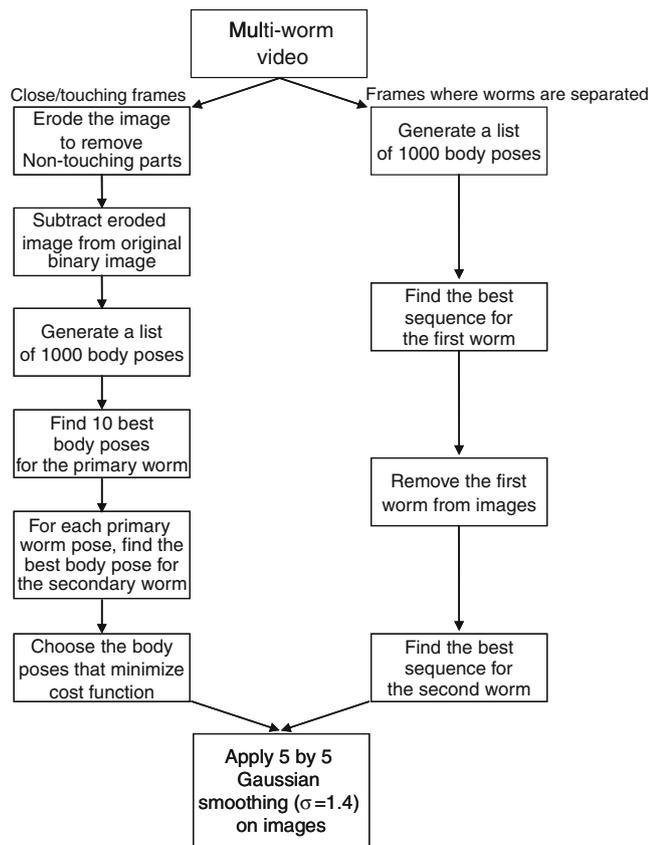
```
┌─────────────────┐
│   Multi-worm    │
│     video       │
└─────────────────┘
```

Close/touching frames — Frames where worms are separated

```
┌─────────────────┐          ┌─────────────────┐
│ Erode the image │          │ Generate a list │
│ to remove       │          │ of 1000 body    │
│ Non-touching    │          │ poses           │
│ parts           │          └─────────────────┘
└─────────────────┘

┌─────────────────┐          ┌─────────────────┐
│ Subtract eroded │          │ Find the best   │
│ image from      │          │ sequence for    │
│ original        │          │ the first worm  │
│ binary image    │          └─────────────────┘
└─────────────────┘

┌─────────────────┐          ┌─────────────────┐
│ Generate a list │          │ Remove the first│
│ of 1000 body    │          │ worm from images│
│ poses           │          └─────────────────┘
└─────────────────┘

┌─────────────────┐          ┌─────────────────┐
│ Find 10 best    │          │ Find the best   │
│ body poses      │          │ sequence for    │
│ for the primary │          │ the second worm │
│ worm            │          └─────────────────┘
└─────────────────┘

┌─────────────────┐
│ For each primary│
│ worm pose, find │
│ the best body   │
│ pose for the    │
│ secondary worm  │
└─────────────────┘

┌─────────────────┐
│ Choose the body │
│ poses that      │
│ minimize cost   │
│ function        │
└─────────────────┘

        ┌─────────────────┐
        │ Apply 5 by 5    │
        │ Gaussian        │
        │ smoothing (σ=1.4)│
        │ on images       │
        └─────────────────┘
```

Figure 3 Block diagram of the multi-worm tracking algorithm.

lapping with the non-filled area. If the two worms are only close to each other without touching or crossing each other, the new image will be just equal to the original binary image because there will be no object in the eroded image.

The dynamic programming approach can make the process of finding the best sequence more computationally efficient, and therefore we use it for the portion of the video where the worms do not touch, which is the majority of the video and also the portion of the video where the body-fitting is easier. However, for touching/close frames, it is a more difficult problem to simulate the two worm body poses correctly. When worms touch, the binary foreground blob can be much larger than a typical single worm body, and therefore choosing good body poses in a frame requires not only (as in Section 2.3.2) the match cost between image blob pixels and model pixels, and the deformation cost of deforming the model, but also must rely on the overlapping cost of the two worm bodies (a cost which doesn't enter into the case where the two worms are far apart), as well as the distance between the current possible pose and the previous poses (a cost which only entered into the dynamic programming in the second step, as we try to find the sequence of poses).

For this reason, we modify our approach. For each touching/close frame, one of the two worms will be chosen

as the primary worm. As discussed in Section 2.3.2, we have a set of M plausible body poses for the frame, obtained using the dynamic programming approach to minimize the match cost and deformation cost. From this set, first the best H body poses are chosen to be candidates for the primary worm based on both the match cost of the whole body pose and the distance between the pose in the current frame and the pose in the previous frame. For each of these H candidates, we find the best body pose for the secondary worm from those M poses to fill the remainder of the object based on the overlapping cost. The overlapping cost is defined to be 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 single set of body poses for the two worms, which is the set that achieves minimum value of the overlapping cost plus the Euclidean distance between the current pose and the previous pose for both worms from those H 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 in both the separated portion and the close/touching portion of the video, we apply a two dimensional Gaussian filter to smooth the final results of every frame. The block diagram of the whole multi-worm match process is shown in Fig. 3 and the pseudo code of the main program is written in the Appendix.

For the reversal detection discussed later, after the best match configurations of both worms are decided, we can manually assign one of the two end parts to be the head/tail part by using the original images as references. The manual assignment does not need to be done on each frame. It is done only once per video.

3 Results

The experiments were performed using Matlab on a 2.33 GHz Pentium-IV desktop computer. The algorithm was run on 29 different videos which contained 10,579 frames in total and each sequence varies from 131 to 498 frames. K, M, and H were set to be 3,000, 1,000, and 10 respectively, where K is the number of sampled pixel positions, M is the number of sampled worm body poses for each image, and H is the number of the candidate body poses for each worm in each frame as previously described. The weights W_x, W_y, and W_θ in Eq. 1 were chosen in the ratio 4:4:3 based on limited subjective evaluation over a set of values. The choice of relative weight W_θ should allow the worm body pose to have appropriate bending but still agree with the worm body anatomy which can be verified by human observers. Binary images are obtained from the
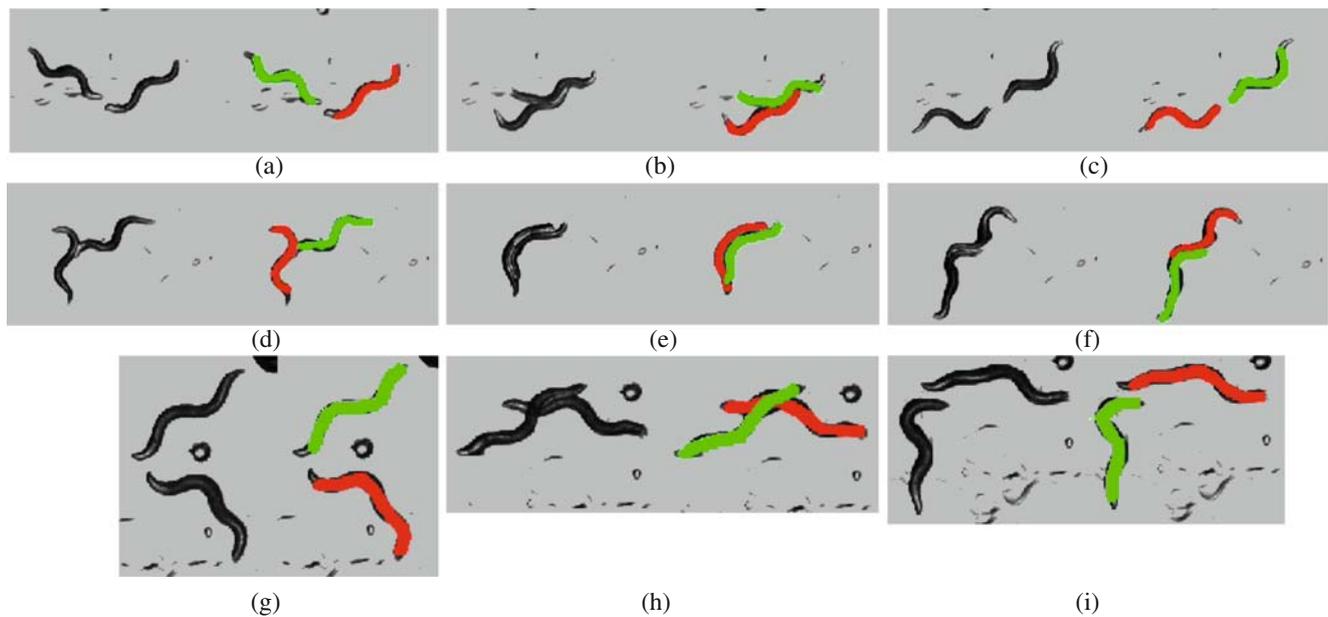
Fig. 4 Nine images from three videos show the best matching configuration. Images **a**–**c** are frames 13, 111, 134 extracted from the first video, images **d**–**f** are frames 86, 143, 206 from the second video, and images **g**–**i** are frames 16, 63, 91 from the third video (with two worms crossing each other). In each image pair, the *left side* shows the original grayscale image and the *right side* shows the matching result.

original images by using the thresholding algorithm from [4]. Some pictorial results are shown in Fig. 4. Images a, b and c are frames 13, 111, 134 extracted from the first video, images d, e and f are frames 86, 143, 206 from the second video, and images g, h, and i are frames 16, 63, 91 from the third video (with two worms crossing each other). In each image, the left side shows the original grayscale image and the right side shows the matching result. These examples illustrate the ability to identify two worms correctly before and after they touch each other. Their body poses are simulated and clearly distinguished during the time the two animals are touching.

First, we evaluate how well the algorithm performs on the three major goals: (1) to find the best body poses for both worms for further extraction of motion related features, (2) to identify two worms correctly before and after they touch each other in every video and (3) to detect

reversals even when morphological skeletons are not available due to the two worms touching.

3.1 Good Estimated Pose and Motion-Related Features

In [31], tracking accuracy is evaluated using the "editing extent," which is defined as the distance between the automatically and manually detected head locations, normalized by the worm length for those worms that are considered incorrectly segmented by the human observer. In order to emphasize that our algorithm can simulate the highly deformable nature of the worm body, we use pixel-based, feature-based and human observer based methods to evaluate the match quality of our algorithm. 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 1,913 images (including 793 non-touching frames and 1,120 touching frames) randomly chosen from 11 videos with a different pair of worms in each video. Given an original image and the number of
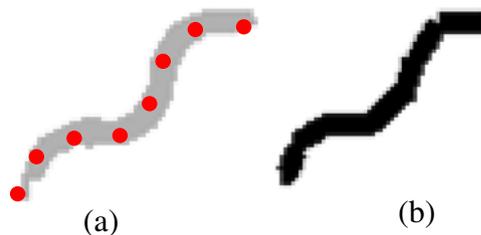

Figure 5 **a** Eight joint points chosen manually for the seven parts in this frame, **b** body pose built from manually selected joint points.

Figure 6 An example of the comparison between the automatically generated model and the manually generated model. The *black area* is covered by both models; the *gray area* is the difference between these two models.

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 (Fig. 5a). Then these points are used to compute x, y and θ of each part and to build worm body poses (Fig. 5b). 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 percentages defined by the following equation:

$$\text{correct percentage} = \frac{N_{AM}}{N_A}$$

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. A higher value for this percentage indicated that the body poses decided by the algorithm agree more with the manually generated body poses (Fig. 6).

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 two different scores:

$$(1)\ \text{predicted positive value} = \frac{N_{MO}}{N_M} \text{ and } (2)\ \text{true positive rate} = \frac{N_{MO}}{N_O}$$

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. Predicted positive value is an indication of the probability that a given pixel which our model says is part of the worm body actually is part of the worm body. True positive rate tells us the percentage of the actual worm body covered by our model. The predicted positive value will decrease and the true positive rate will increase if we use models with larger sizes (for example, if we choose model part width W = average body width w instead of $W=0.9w$).

All results are listed in Table 1. We notice that the correct percentages between automatically generated model and manually generated model all range from 72% to 83% except in two videos (015 and 020). In these two videos, the result is better for touching frames than non-touching frames. That is because the background is not very clear due to the un-evenness in the bacteria layer or crawl track left by the worms in these videos, which may cause the size of the binary worm body in some images to be abnormally larger than its usual size and predicted positive values to be very low. Predicted positive values are also over 85% and true positive rates are higher than 70% for almost all frames. In order to reduce the possibility of two worms overlapping in our results for touching frames, the width W of the model is always chosen to be 90% of the actual average body width as calculated from non-touching frames. For this reason, the model tends to be covered by the whole worm body which will cause the predicted positive value to be generally larger than the true positive rate. The results of the comparison between manually generated models and original images are also listed in the last four rows in this table, which clearly shows that our automatic matching algorithm outperforms human observers.

The angle change rate is also computed from both the manually generated model and the model generated by our algorithm. The results are shown in Table 2. The angle change rate is defined in [1, 4] as the ratio of the average angle difference between every pair of consecutive segments connected by skeleton points. The angle change rate is an important feature characterizing body postures of mutants, and was shown in [1], for example, to be

Table 2 Angle change rate verification results.

File name		005	006	007	008	011	012	015	016	017	018	020
Automatically generated model	Separated	32.6	33.7	37.9	37.0	29.8	29.2	31.7	33.1	32.3	36.1	36.6
		33.0	30.6	31.0	39.0	30.9	35.1	34.9	29.3	32.4	34.3	41.3
	Close	30.0	33.7	36.2	32.4	27.6	30.6	25.0	30.0	30.2	28.8	38.0
		31.4	30.3	31.8	33.6	33.6	32.1	34.3	24.8	29.3	32.3	35.4
Manually generated model	Separated	33.5	34.5	40.7	34.6	33.3	30.7	31.9	33.5	34.0	37.5	36.1
		36.2	32.7	32.5	40.5	29.8	34.3	38.9	24.7	34.6	34.2	43.4
	Close	30.4	33.6	41.0	34.2	28.6	31.0	29.9	30.9	32.5	29.4	42.2
		30.6	29.4	32.3	34.4	30.8	34.3	31.2	24.5	33.3	31.7	38.6
Difference Percentage	Separated (%)	2.6	2.2	6.9	6.8	10.4	4.9	0.7	1.0	5.0	3.8	1.2
		8.6	6.5	4.6	3.7	3.8	2.5	10.4	18.5	6.3	0.2	4.8
	Close (%)	10.4	2.6	0.6	1.2	16.4	0.9	6.9	8.3	4.4	27.6	14.4
		18.2	11.4	0.6	17.8	3.4	0.2	24.7	0.9	3.8	7.9	12.3

The two numbers for each parameter are the angle change rates for the two worms

Table 3 Identification results and reversal verification results.

Number of tested videos	29	
Number of correct identifications	26	
Number of wrong identifications	3	
Number of reversals correctly detected with results from our algorithm and the method in [22]	Touching 57 (98.3%)	Overall 86 (96.6%)
Number of wrong detections	6 (9.5%)	7 (7.5%)
Number of reversals missed	1 (1.7%)	3 (3.4%)

significant in distinguishing between goa-1(n1134) mutants and wild-type. In this paper, we use the joint points of the parts to be our skeleton points to calculate angle change rate. From Table 2, we see that the difference percentages of average angle change rate between our algorithm and manually generated models are lower than 10% for non-touching (separated) frames and 20% for touching frames (close) in most of the videos.

3.2 Correct Identification

Both worms in all videos are also tracked by a human observer to see if our method can correctly identify worms separately after they touch. From Table 3, we see that our program can identify both worms correctly in 26 of 29 videos.

3.3 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 backward movement is defined as a reversal. In [34], 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 reversal detection algorithm requires the skeleton points from each worm body which can not be obtained using the method in [39] when two worms touch each other. By using our modeling algorithm, after all parameters of all parts are obtained with our algorithm, all joint points can be considered to be pseudo skeleton points. We can use the two joint points nearest the two ends to be our reference points to detect reversals. Table 3 shows that there were 86 reversals correctly detected by our automatic algorithm. Of these, 57 occurred during the close/touching portion of the video. In addition, the automatic algorithm incorrectly declared seven events to be reversals (six of them were in the close/touching portion of the video). Only three actual reversals were missed (of which one was in the close/touching portion of the video). So our algorithm has a high rate of correct detections while maintaining a low rate of false alarms and false dismissals.

3.4 Analyzing Physical Contact

As described in Section 2.3.3, our algorithm can automatically locate those portions of the video where two worms are close to or touching each other. We can therefore calculate the length of time they are in physical contact. We also examine the average speed and rate of reversals from the video portions before and after the two worms touch. The results are shown in Table 4. All worms were npr-1 (ky13) mutants. In Table 4, we show the data for 6 individual videos as well as the average values over all 29 videos in the last column. Row 2 shows the length of time the worms touch each other. The average speed and the rate

Table 4 Experimental results from six individual videos as well as average values over all 29 videos.

File name	001	002	003	004	005	006	Average
Time length of two worms touching (s)	17.4	21.6	19.8	10.6	16.4	27.0	22.14
Ave speed before touching (pixel/s)	5.7	6.0	9.2	8.0	7.8	12.1	8.21
	6.8	3.9	8.0	10.6	6.8	5.8	8.11
Rate of reversals before touching (1/s)	0.17	0.14	0	0.19	0.06	0	0.07
	0.11	0.09	0.05	0.19	0	0.04	0.07
Ave speed after touching (pixel/s)	7.6	7.3	13.2	8.6	9.2	7.0	10.33
	9.9	11.3	12.7	7.1	10.4	9.7	10.55
Rate of reversals after touching (1/s)	0	0	0	0	0.12	0.04	0.05
	0	0	0	0	0.06	0	0.04

Row 2 shows the length of time the worms touch each other. The average speed and the rate of reversals before touching events are in rows 3–4 and 5–6. The average speed and the rate of reversals after touching events are in rows 7–8 and 9–10. Other than the time length of touching, the parameters listed have two numbers for each video because they represent results for the two worms in the video.

of reversals before touching events are in rows 3–4 and 5–6. The average speed and the rate of reversals after touching events are in rows 7–8 and 9–10. One can observe that the animals move faster after physical contact than before.

4 Conclusion

This paper presents a method that combines 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. For each video, we begin by using a dynamic programming algorithm to generate many worm body poses in every frame. For those portions of the video where the two worms are clearly separated, a dynamic programming algorithm is used again to find the best sequences over time for both worms. For those portions of the video where the two worms are close or touching each other, we find the best match configuration for the two worms based on the Euclidean distances between pairs of body poses in adjacent image frames.

There are several contributions in this paper. First, the presented method allows us to identify two worms correctly before and after they touch each other in 90% of our videos. Second, we can use these models to accurately resolve the individual body postures of two worms in physical contact with one another. We note that this tracking algorithm for two worms is fully automated and requires no human annotation at any point (including no need for human annotation of the first frame). When this algorithm was combined with a previously described algorithm for detection of reversals, human annotation was required to identify head and tail once per video, which was needed to identify reversals. However the tracking algorithm itself involves no human intervention, so it is suitable for analyzing large numbers of videos where human annotation would be tedious. 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.

This algorithm will provide many applications towards characterizing physical interactions between animals. Previous research on automated analysis of C. elegans videos has shown that large numbers of biologically relevant features can be automatically extracted. These features include, for example, body length and width, average speed, curvature, depth of body bends, and frequency and duration of reversals. These features and others have been shown to be important in classifying and characterizing many different mutant strains [4]. Using the algorithm described in this paper, the body poses of two worms can be identified, so the various features extracted in prior research for single-worm videos can now be extended to videos with two worms. In this paper, we examined two of these features (angle change rate and reversals) in these two-worm videos. In addition, new features characterizing the interactions themselves such as average duration of bodily contact can be extracted across different mutant strains and different environmental conditions. We intend to characterize how interactions between animals affect various features of body movement and posture in future work.

Appendix: Pseudo Code for the Multi-Worm Tracking Algorithm

This appendix provides pseudo code for the multi-worm tracking algorithm. The worm video data and source code in Matlab for our algorithms are available from the web site [40].

    READ l—learned length of worm body
    READ w—learned width of worm body
    COMPUTE W—width of rectangle as $0.9 \times w$
    COMPUTE L—length of rectangle as $2 \times W$
    COMPUTE N—number of rectangles as round(l/L)
    SET Index1 to 1—the indices of subsections where two worms are separated
    SET Index2 to 0—the indices of subsections where two worms are close or touching

Pseudo code for dividing the whole video into two subsections (close/touching and non-touching)

    FOR each image frame in the video
    COMPUTE the number of worms
    IF the number of worms is equal to 2
    COMPUTE match cost at each integer pixel position
    OBTAIN K integer pixel positions with the lowest match cost
    OBTAIN M worm body poses with the lowest match cost plus deformation cost.
    COMPUTE Dc—the distance between two centroids in the current image frame
    READ Dp—the distance between two centroids in the previous image frame
    IF $Dc \geq l$ and $Dp \geq l$
    ADD the image frame to subsection of Index1
    ELSE IF $Dc \leq l$ and $Dp \geq l$
    INCREMENT index2
    ADD the image frame to subsection of Index2
    ELSE IF $Dc \geq l$ and $Dp \leq l$
    INCREMENT index1
    ADD the image frame to subsection of Index1
    END IF

ELSE

OBTAIN new binary image where two worms are partially separated

COMPUTE match cost at each integer pixel position

OBTAIN K integer pixel positions with the lowest match cost

OBTAIN M worm body poses with the lowest match cost plus deformation cost

ADD the image frame to subsection of Index2

END IF

ENDFOR

Pseudo code for finding the best sequences within one non-touching subsection

FOR subsection 1 to Index1

Finding the best sequence for one worm

FOR each image in the subsection

IF the current image is the first image in the subsection

FOR each worm body pose in the current image

SET accumulated cost function = 0

END FOR

ELSE

FOR each worm body pose in the current image

FOR each possible worm body pose in the previous image

READ accumulated cost function of the worm body pose in the previous image

COMPUTE Euclidean distance d between the current and the previous body poses

COMPUTE C = Euclidean distance d + match cost of the current body pose + accumulated cost function of the previous body pose

END FOR

OBTAIN the minimum of C and the corresponding body pose in the previous image

SAVE accumulated cost function of the current worm body pose = min(C)

SAVE the corresponding worm body pose in the previous image

END FOR

END IF

END FOR

OBTAIN the minimum overall accumulated cost function and the corresponding worm body pose in the last image

OBTAIN corresponding body poses in previous image frames

Finding the best sequence for the second worm

OBTAIN the image sequence with the first worm being removed

OBTAIN the best image sequence for the second worm using the above method

END FOR

Pseudo code for finding the best sequences within one close/touching subsection

FOR subsection 1 to Index2

FOR each image in the subsection

SET the primary worm differently from the previous image frame

OBTAIN the 10 best worm body poses for the primary worm

FOR each primary worm body pose

OBTAIN the best worm body pose for the secondary worm

END FOR

OBTAIN the best set of two worm body poses

END FOR

END FOR

Gaussian smoothing

SHOW new video

References

1. Baek, J., Cosman, P., Feng, Z., Silver, J., & Schafer, W. R. (2002). Using machine vision to analyze and classify *Caenorhabditis elegans* behavioral phenotypes quantitatively. *Journal of Neuroscience Methods*, *118*, 9–21.
2. Cronin, C. J., Mendel, J. E., Mukhtar, S., Kim, Y. M., Stirbl, R. C., Bruck, J., et al. (2005). An automated system for measuring parameters of nematode sinusoidal movement. *BMC Genetics*, *6*, 5.
3. Feng, Z., Cronin, C. J., Wittig Jr., J. H., Sternberg, P. W., & Schafer, W. R. (2004). An imaging system for standardized quantitative analysis of *C. elegans* behavior. *BMC Bioinformatics*, *5*, 115.
4. Geng, W., Cosman, P., Berry, C., Feng, Z., & Schafer, W. R. (2004). Automatic tracking, feature extraction and classification of *C. elegans* phenotypes. *IEEE Transactions on Biomedical Engineering*, *51*, 1811–1820.
5. Dhawan, R., Dusenbery, D. B., & Williams, P. L. (1999). Comparison of lethality, reproduction, and behavior as toxicological endpoints in the nematode *Caenorhabditis elegans*. *Journal of Toxicology and Environmental Health Part A*, *58*(7), 451–462.
6. de Bono, M., Tobin, D. M., Davis, M. W., Avery, L., & Bargmann, C. I. (2002). Social feeding in *Caenorhabditis elegans* is induced by neurons that detect aversive stimuli. *Nature*, *419* (6910), 899–903.
7. Liu, K. S., & Sternberg, P. W. (1995). Sensory regulation of male mating behavior in *Caenorhabditis elegans*. *Neuron*, *14*(1), 79–89.
8. McInerney, T., & Terzopoulos, D. (1996). Deformable models in medical image analysis: a survey. *Medical Image Analysis*, *1*(2), 91–108.
9. Cootes, T. F., & Taylor, C. J. (2001). Statistical models of appearance for medical image analysis and computer vision. *Proc. SPIE Medical Imaging*, *4322*, 236–248.

10. Kass, M., Witkin, A., & Terzopoulos, D. (1987). Active contour models. *International Journal of Computer Vision*, *1*(4), 321–331.
11. Scott, G. L. (1987). The alternative snake—And other animals. *3rd Alvey Vison Conference*. Cambridge, England, pp. 341–347.
12. Staib, L. H., & Duncan, J. S. (1992). Boundary finding with parametrically deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*(11), 1061–1075.
13. Bajcsy, R., & Kovacic, A. (1989). Multi-resolution elastic matching. *Computer Graphics and Image Processing*, *46*, 1–21.
14. Burl, M. C., Weber, M., & Perona, P. (1998). A probabilistic approach to object recognition using local photometry and global geometry. *European Conference on Computer Vision*. Freiburg, Germany, June, pp. 628–641.
15. Fischler, M. A., & Elschlager, R. A. (1973). The representation and matching of pictorial structures. *IEEE Transactions on Computers*, *22*(1), 67–92.
16. Hogg, D. (1983). Model based vision: A program to see a walking person. *Image and Vision Computing*, *1*(1), 5–20.
17. Rohr, K. (1993). Incremental recognition of pedestrians from image sequences. *IEEE Conference on Computer Vision and Pattern Recognition*. New York, USA, June, pp. 9–13.
18. Felzenszwalb, P. F., & Huttenlocher, D. P. (2000). Efficient matching of pictorial structures. *IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head, USA*, June, pp. 66–73.
19. Deutscher, J., Blake, A., & Reid, I. (2000). Articulated body motion capture by annealed particle filtering. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, *2*, 126–133.
20. Kakadiaris, L., & Metaxas, D. (2000). Model-based estimation of 3D human motion, pattern analysis and machine intelligence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(12), 1453–1459.
21. Sidenbladh, H., Black, M. J., & Fleet, D. J. (2000). Stochastic tracking of 3D human figures using 2D image motion. In D. Vernon (Ed.), *European Conference on Computer Vision* (pp. 702–718). Dublin, Ireland: Springer LNCS 1843 (June).
22. Bar-Shalom, Y., Fortmann, T., & Scheffe, M. (1980) Joint probabilistic data association for multiple targets in clutter. *Proceedings of the Conference on Information Sciences and System*.
23. Isard, M., & MacCormick, J. (2001). BraMBLe: A Bayesian Multiple-Blob Tracker. *Proceedings of the International Conference on Computer Vision*, pp. 34–41.
24. MacCormick, J., & Blake, A. (2000). A probabilistic exclusion principle for tracking multiple objects. *International Journal of Computer Vision*, *39*(1), 57–71.
25. Ioffe, S., & Forsyth, D. A. (2001). Human tracking with mixtures of trees. *International Conference on Computer Vision, Vancouver, Canada*, vol. 1, pp. 690–695 (July).
26. Mori, G., & Malik, J. (2002). Estimating human body configurations using shape context matching. *European Conference on Computer Vision, Copenhagen, Denmark*, vol. 3, pp. 666–680 (May).
27. Sullivan, J., & Carlsson, S. (2002). Recognizing and tracking human action. *European Conference on Computer Vision, Copenhagen, Denmark*, vol. 1, pp. 629–644 (May).
28. Mittal, A., & Davis, L. (2003). M2 tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *International Journal of Computer Vision*, *51*(3), 189–203.
29. Khan, Z., Balch, T., & Dellaert, F. (2005). MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(11), 1805–1819.
30. Qu, W., Schonfeld, D., & Mohamed, M. (2007). Real-time interactively distributed multi-object tracking using a magnetic-inertia potential model. *IEEE Transactions on Multimedia*, *9*(3), 511–519.
31. Roussel, N., Morton, C. A., Finger, F. P., & Roysam, B. (2007). A computational model for *C. elegans* locomotory behavior: application to multiworm tracking. *IEEE Transactions on Biomedical Engineering*, *54*(10), 1786–1797.
32. Huang, K., Cosman, P., & Schafer, W. R. (2007). Automated tracking of multiple *C. elegans* with articulated models. *IEEE International Symposium on Biomedical Imaging*, pp. 1240–1243.
33. Felzenszwalb, P. F., & Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, *61* (1), 55–79.
34. Huang, K., Cosman, P., & Schafer, W. R. (2006). Machine vision based detection of omega bends and reversals in *C. elegans*. *Journal of Neuroscience Methods*, *158*, 323–336.
35. Brenner, S. (1974). The genetics of *Caenorhabditis elegans*. *Genetics*, *77*, 71–94.
36. Amini, A., Weymouth, T., & Jain, R. (1990). Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(9), 855–867.
37. Angel, E., & Bellman, R. (1972). *Dynamic programming and partial differential equations*. Orlando, FL: Academic.
38. Lowem, D. G. (1991). Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*(5), 441–450.
39. Gonzalez, R., & Woods, R. (2002). *Digital image processing* (2nd ed.). NJ: Prentice Hall.
40. University website: http://www.code.ucsd.edu/~pcosman/C_elegans. html.

Dr. Kuang-Man Huang was born in Taiwan in 1977. He received the B.S. degree with award in Electro-physics from National Chiao-Tung University, Hsin-Chu, Taiwan, in 1999, and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of California at San Diego in 2003 and 2008, respectively. He was a research assistant with the Cosman Lab on Information Coding as well as with the Schafer Lab in the Center for Molecular Genetics, UCSD, from 2003 to 2008.

Pamela Cosman obtained her B.S. with Honor in Electrical Engineering from the California Institute of Technology in 1987, and her M.S. and Ph.D. in Electrical Engineering from Stanford University in 1989 and 1993, respectively. She was an NSF postdoctoral fellow at Stanford University and a Visiting Professor at the University of Minnesota during 1993-1995. In 1995, she joined the faculty of the Department of Electrical and Computer Engineering at the University of California, San Diego, where she is currently a Professor and Director of the Center for Wireless Communications. Her research interests are in the areas of image and video compression and processing. Dr. Cosman is the recipient of the ECE Departmental Graduate Teaching Award (1996), a Career Award from the National Science Foundation (1996-1999), and a Powell Faculty Fellowship (1997-1998). She was a guest editor of the June 2000 special issue of the IEEE Journal on Selected Areas in Communications on 'Error-resilient image and video coding,' and was the Technical Program Chair of the 1998 Information Theory Workshop in San Diego. She was an associate editor of the IEEE Communications Letters (1998-2001), and an associate editor of the IEEE Signal Processing Letters (2001-2005). She was a senior editor (2003-2005) and is now the Editor-in-Chief of the IEEE Journal on Selected Areas in Communications.

She is a member of Tau Beta Pi and Sigma Xi, and a Fellow of the IEEE. Her web page address is http://www.code.ucsd.edu/cosman/.

William Schafer received the A.B. degree in biology (summa cum laude) from Harvard University in 1986, and the Ph.D. degree in biochemistry from the University of California, Berkeley in 1991. From 1992-1995 he was a Life Sciences Research Foundation postdoctoral fellow at the University of California, San Fransicso. From 1995-2006 he was on the faculty of the Division of Biological Sciences at the University of California San Diego. He is currently Programme Leader in Cell Biology at the Medical Research Council Laboratory of Molecular Biology in Cambridge. He has been a Visiting Fellow at Clare Hall, Cambridge (2004-2005), a Visiting Professor in the Department of Physics at the Ecole Normale Superieure in Paris, and is currently a Bye-fellow at Downing College, Cambridge. His research has focused on the molecular and cellular mechanisms underlying neural circuit function, and on the quantitative analysis of behavioral phenotypes. Dr Schafer has received a Presidential Early Career Award for Scientists and Engineers, a Klingenstein Fellowship in Neuroscience, a Beckman Young Investigator Award, and a Sloan Fellowship in Neuroscience.