

Data and text mining

Straightening *Caenorhabditis elegans* imagesHanchuan Peng^{1,*}, Fuhui Long¹, Xiao Liu², Stuart K. Kim² and Eugene W. Myers¹¹Janelia Farm Research Campus, Howard Hughes Medical Institute, Ashburn, VA 20147 and²Department of Developmental Biology, Stanford University Medical Center, Stanford, CA 94305, USA

Received on June 13, 2007; revised on October 11, 2007; accepted on November 6, 2007

Advance Access publication November 19, 2007

Associate Editor: Jonathan Wren

ABSTRACT

Motivation: *Caenorhabditis elegans*, a roundworm found in soil, is a widely studied model organism with about 1000 cells in the adult. Producing high-resolution fluorescence images of *C.elegans* to reveal biological insights is becoming routine, motivating the development of advanced computational tools for analyzing the resulting image stacks. For example, worm bodies usually curve significantly in images. Thus one must ‘straighten’ the worms if they are to be compared under a canonical coordinate system.

Results: We develop a worm straightening algorithm (WSA) that restacks cutting planes orthogonal to a ‘backbone’ that models the anterior–posterior axis of the worm. We formulate the backbone as a parametric cubic spline defined by a series of control points. We develop two methods for automatically determining the locations of the control points. Our experimental methods show that our approaches effectively straighten both 2D and 3D worm images.

Contact: pengh@janelia.hhmi.org

Supplementary information: The example data sets and programs are available upon request.

1 INTRODUCTION

Fluorescence image analysis is one of the most important tools for studying various biological questions, especially in the nematode *Caenorhabditis elegans*, which has a small, transparent body. More strikingly, *C.elegans* has essentially invariable cell lineages and cell fates, providing a unique chance to investigate biological processes at the level of single cells. High-resolution multiplex fluorescence 2D/3D images of the entire worm are being generated more and more routinely in the course of these studies. Because worms usually have different curved shapes and other morphological features, an automatic worm straightening technique is desired for both biological and engineering reasons. Biologically, it is easier to compare features of various worms if these worms are straight with respect to their anterior/posterior axes. Practically, the bounding box of a straightened worm (Fig. 1c) is on average 80% smaller than the bounding box of the original curved worm in the raw image (Fig. 1a). Thus, worm straightening reduces the image file size significantly, saving a great deal of storage space and I/O time during file access.

Several characteristics of worm images complicate our problem as follows:

- First, because specific features of a worm are to be measured in these images, the straightened worm should have the least possible deformation of cell morphology, intensity and relative cell locations, compared to the original worm in the raw image.
- Second, the fluorescent ‘stain’ typically lights up only the nuclei of the worm instead of its entire body. Thus, the skin, or cuticle, of the worm is usually unstained. This makes it difficult to estimate the bending parameters directly.

The application of worm straightening *per se* is new. Given the stereotypic location of cells, registering two worms after straightening is a simple matter of a rigid transformation followed by scaling. The worm straightening algorithm (WSA) presented here can be applied to various *C.elegans* strains such as PHA4, UNC54, etc. and is one of several major tools in our project to build a digital worm cell atlas (Long *et al.*, 2007).

2 METHODS

2.1 Overall scheme of WSA

The crucial observation about worm images (Fig. 1a) is that a worm is globally nonlinear (i.e. curved), but locally smooth with a very gentle curl. That is, the worm can be viewed as a manifold where the local space about any point is to first approximate a rectilinear Euclidean space. The goal can then be considered to chart the manifold from the pixel intensities of the image. We reduce this to a 1D problem by seeking the principal curve or ‘backbone’ (Fig. 1b) that represents the anterior/posterior (A/P) line that passes from head to tail through the center of a perfectly straightened worm. That is, this curve is the 1D manifold for the A/P axis of the worm. We use it to chart the entire worm manifold by generating a series of 1-pixel separated planes orthogonal to the backbone and then restacking these along a straight A/P line (Fig. 1c). Because each local transform is a rigid rotation, the resolution of the straightened image is merely a function of the sub-pixel interpolation error when a straight line is rotated in space. Thus the overall warping inaccuracy is minimized naturally, meeting our design goal. This approach is much simpler than deformation models, such as non-rigid 3D image registration based on the identification and matching of landmark-points and the generation of a deformation field such as a thin-plate-spline based on the matching. In over 100 samples tested thus far our cutting-plane-based method always involves locally

*To whom correspondence should be addressed.

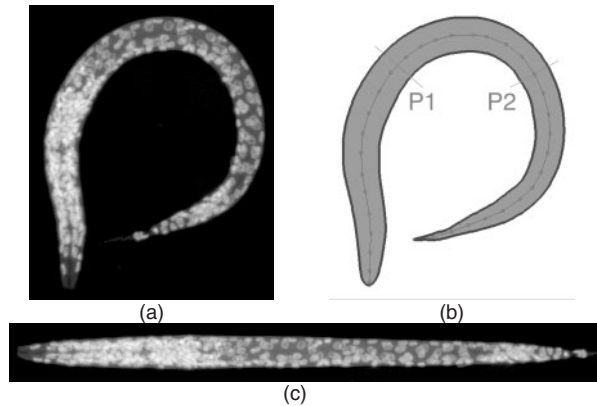


Fig. 1. Schematic illustration of WSA. (a) A curved worm image. (b) The detected backbone (red), the respective control points (red dots) and orthogonal cutting lines/planes (purple lines P1 and P2). Also shown is the worm boundary (blue), which is not used in the BDB^- algorithm (see Section 3.1) but is used in the faster BDB^+ algorithm (see Section 3.2). (c) The straightened worm image.

small displacements and produces biologically sensible results, obviating the need for such complex and computationally intensive methods.

Within this framework, the key problem is how to find the backbone of a worm. We present two algorithms, BDB^- and BDB^+ , to detect the worm backbone under different scenarios. The algorithm BDB^- detects the backbone for a worm when the outer boundary of the worm is hard to detect precisely, whereas the algorithm BDB^+ works only when a worm image has a clear boundary.

There are many related image analysis techniques such as tracing or analyzing a curved or tree-like structure in images of blood vessels, neurons and even *Arabidopsis* (Al-Kofahi *et al.*, 2002; Mace *et al.*, 2006; Sun, 1989). However, these methods do not directly solve our problem due to its different properties such as the lack of contour information (for the BDB^- case). On the other hand, similar to our nuclear staining and image analysis platform for the worm, we note that there was another recent image analysis study of fly embryos (Luengo Hendriks *et al.*, 2006).

2.2 Worm strain and DAPI staining

The worm strain PD4251 was provided by Andrew Fire. This strain produces green fluorescent protein (GFP) in all body wall muscle nuclei in the L1 stage larvae (Fire *et al.*, 1998). For DAPI staining, worms were first frozen by liquid nitrogen, thawed in 20°C acetone and then fixed in 80 mM KCl, 20 mM NaCl, 10 mM EGTA, 5 mM spermidine-HCl, 15 mM Na-mM piperazine-*N,N'*-bis[2-ethanesulfonic acid], pH 7.4 and 2.5% formaldehyde for 30 min and stained with 5 ng/μl DAPI in $0.5 \times M9$.

2.3 Microscopy

We used a Leica SP2 AOBs confocal microscope with a 405 nm UV laser to record fluorescence from DAPI-stained L1 stage worms mounted in 50% glycerol. To reduce random noise, we used a scan speed of eight and averaged over two scans. To maximize resolution, we used a $63\times$ oil objective lens with N.A. = 1.40 and scanned each 2D plane at 2048×2048 pixels.

3 BACKBONE DETECTION

It is natural to define a smooth curve as a cubic spline through a train of control points (i.e. the red dots in Fig. 1b).

The backbone detection problem is then equivalent to determining the location of a sequence of control points. This parametric representation allows us to march along the backbone at any desired spacing (e.g. 1-pixel per step) to produce the sequence of orthogonal cutting planes for restacking.

3.1 BDB^- : backbone detection without the worm boundary

For worm images where only the nuclei of cells are lit up using fluorescent labels, it is often hard to detect a clear worm boundary, except for the cases to be discussed in Section 3.2. The general backbone detection problem is thus how to find a smooth mid-axis through the set of scattered worm nuclei.

This problem is related to detecting the ‘principal curve’ of a set of scattered points (Hastie, 1994). However, the existing principal curve detection methods (Hastie, 1994; Kégl *et al.*, 2000) cannot be employed directly, because worm cells have varying morphologies, intensities and sizes, and so cannot be simplified to a set of points such as the nuclei centers. In effect, we must generalize the problem to consider an intensity weighted set of pixels.

In our algorithm for *backbone detection without the worm boundary* (BDB^-), our intuition is to define the backbone control points as those that form a sufficiently smooth curve and minimize the sum of the distances from the backbone to the nearest stained cells. To this end, we define W as all the pixels whose intensity is more than one standard deviation above the mean of the distribution of pixel values in the image. The algorithm computes a backbone with respect to W . While in some sense W is an estimate of the portion of the image that is worm versus background, the algorithm is not sensitive to the precise definition of W . Moreover, because the cell positions in *C.elegans* are highly stereotyped, even if the curve produced is not precisely the A/P axis, the algorithm produces consistently the same relative curve in space across different worms because it is a function of the light intensity distribution.

3.1.1 Initialization of control points In BDB^- , we initialize the set of control points, denoted as Ω , by randomly selecting a small set of pixel locations (e.g. 50) from W . We then compute the minimum spanning tree (MST) of the complete graph of the selected points where each edge is weighted with the distance between the points. Doing so takes $O(|\Omega|^2 \lg |\Omega|)$ time (Corman *et al.*, 2001). We next compute the longest path, or diameter, of the MST to give us an ordered sequence of control points that will constitute the start point for the energy minimization procedure in the next subsection. The diameter can be computed using two breadth-first-search passes (Corman *et al.*, 2001) over the MST. Note carefully that the initial points not on the diameter are removed from Ω , and they usually fall in short MST branches.

3.1.2 Energy minimizing deformable backbone model We consider an energy function minimization approach to find the best locations for the final backbone control points. Like many other deformable models such as snakes and active contours (Kass *et al.*, 1987; Xu and Prince, 1998), we define an energy model for a curve and seek the curve

that minimizes energy. The energy model consists of an external or image energy term, and two internal energy terms that constrain the backbone curve to be smooth and succinct.

Consider the external energy first. A nucleus is indicated by the photons recorded at the pixels covering this nucleus most of which are bright and so in W . Thus, the worm is reasonably indicated by the photon distribution across W . So we define the distance between a control point $c \in \Omega$ and W as the sum of distances squared between c and all photons that are closer to c than to any other control point. Because the number of photons at a pixel location p is proportional to the pixel intensity, denoted as $I(p)$, we define the distance between the control point c and a pixel p as the sum of squared Euclidean distances between c and each unit of pixel intensity at pixel p :

$$D(c,p) = I(p) \cdot \|c - p\|^2 \quad (1)$$

Consider the Voronoi decomposition of W with respect to the points in Ω . Let $\Theta(c_k)$ be the Voronoi region of c_k , that is the set of pixels in W that are closer to c_k than to any other control point. We define the external energy E_{image} as the sum of the average distance-squared between c_k and the photons in its Voronoi region. Formally,

$$E_{\text{image}} = \sum_{k=1}^{|\Omega|} \frac{1}{N_k} \sum_{p \in \Theta(c_k)} D(c_k, p) \quad (2)$$

$$N_k = \sum_{p \in \Theta(c_k)} I(p) \quad (3)$$

$$\Theta(c_k) = \{p \in W : \forall j \|c_k - p\| \leq \|c_j - p\|\} \quad (4)$$

For the internal energy, we have two constraints on a worm backbone:

- It should have the shortest possible length while still spanning the entire worm.
- It should be as smooth as possible.

For constraint (a), we define the energy term E_{length} in Equation (5) below, which is the sum of the squared distances between the all the consecutive pairs of control points.

$$E_{\text{length}} = \sum_{k=1}^{|\Omega|-1} \|c_k - c_{k+1}\|^2 \quad (5)$$

For constraint (b), we define the energy term $E_{\text{smoothness}}$,

$$E_{\text{smoothness}} = \sum_{k=2}^{|\Omega|-1} \|c_k - \frac{1}{2}(c_{k-1} + c_{k+1})\|^2 \quad (6)$$

When $c_k = (c_{k-1} + c_{k+1})/2$ for every k then $E_{\text{smoothness}}$ is 0, its smallest possible value. This occurs when c_k is midway on the straight line-segment between c_{k-1} and c_{k+1} , for each k , implying the control points are evenly spaced on a straight line. Intuitively, a curve cannot be any smoother. Note that this energy term favors evenly spaced control points.

The overall energy function E that the BDB⁻ algorithm minimizes has the following form,

$$E = \alpha E_{\text{image}} + \beta E_{\text{length}} + \gamma E_{\text{smoothness}} \quad (7)$$

where α, β, γ are three positive coefficients selected to control the relative contribution of each energy term to the final curve.

3.1.3 Adjusting control points In a classic snake/active contour formulation for a deformable model, the Euler–Lagrange equation from the calculus of variations is usually used to minimize the energy function (Kass et al., 1987; Xu and Prince, 1998). In that case, the external energy term can be often pre-calculated once before the deformable model evolves. However, our situation is different in that every time we update the control points, the respective external energy term also changes because the Voronoi decomposition changes. Thus we consider the direct and simpler energy minimization condition that when E is minimal, the respective derivatives at the locations of control points must be 0:

$$\frac{\partial E}{\partial c_k} = \alpha \frac{\partial E_{\text{image}}}{\partial c_k} + \beta \frac{\partial E_{\text{length}}}{\partial c_k} + \gamma \frac{\partial E_{\text{smoothness}}}{\partial c_k} = 0 \quad (8)$$

Based on Equations (2), (5) and (6), for any $k \in \{3, \dots, |\Omega|-2\}$, Equation (8) is equivalent to

$$0 = \alpha \left(c_k - \frac{1}{N_k} \sum_{p \in \Theta(c_k)} (I(p) \cdot p) \right) + \beta(2c_k - c_{k-1} - c_{k+1}) + \gamma(2c_k - \frac{3}{2}c_{k-1} - \frac{3}{2}c_{k+1} + \frac{1}{2}c_{k+2} + \frac{1}{2}c_{k-2}) \quad (9)$$

Equation (9) suggests one way to minimize the objective function in Equation (7), is to iteratively adjust the locations of the control points by solving Equation (9) for c_k :

$$c_k^{\text{new}} \leftarrow \frac{\left(\alpha \left(\frac{1}{N_k} \sum_{p \in \Theta(c_k)} (I(p) \cdot p) \right) + \beta(c_{k-1} + c_{k+1}) \right) + \gamma \left(\frac{3}{2}c_{k-1} + \frac{3}{2}c_{k+1} - \frac{1}{2}c_{k+2} - \frac{1}{2}c_{k-2} \right)}{\alpha + 2\beta + 2\gamma} \quad (10)$$

To make the denominator of Equation (10) simple, we chose α, β, γ as 1, 0.5, 0.5, respectively. Other choices around these values make little difference to the final results.

For the boundary cases $k \in \{1, 2, |\Omega|-1, |\Omega|\}$, we similarly derive iteration formulas, which are each slightly different. We omit them as their derivation is straightforward and their statement lengthy.

Computing each new set of control points with Equation (10) takes $O(|W|)$ time, the dominant term being to compute the sum in the α -term. Given this, we do not employ a sophisticated Voronoi region algorithm to compute $\Theta(c_k)$, but simply do so using a breadth-first pixel-based search from all the control points simultaneously.

3.1.4 BDB⁻ Algorithm The BDB⁻ algorithm is summarized as follows:

- Initialize the control points as the diameter of the MST of the complete distance graph whose vertices are bright pixels randomly selected on the worm body.
- Find the Voronoi region of each control point in $O(|W|)$ time.
- Update the control points using Equation (10) in $O(|W|)$ time.
- Check whether or not the control points have shifted significantly. If yes, go to step 2. Otherwise go to step 5.

- (5) Use the spline function to generate a smooth parametric backbone curve of the control points.

It is possible to add some further minor changes to the above algorithm, such as increasing or decreasing the number of control points in order to get a spacing in the final result. However, for the cases we tested these variations do not change the results dramatically. Moreover, if even more speed were required one could incrementally compute the E_{image} with respect to the changes in $\Theta(c_k)$ observing that these changes get smaller with each iteration. However, we did not find this necessary and we report in the experiments in Section 4 only results for the basic BDB^- scheme given above.

The major advantage of BDB^- is that it works even for grayscale images without a clear worm boundary. It also requires only a very ‘loose’ initialization of the control points. BDB^- can be applied to both 2D and 3D images.

3.2 BDB^+ : Backbone detection with worm boundary

In some cases it is possible to find a meaningful worm boundary in an image. For example, due to auto-fluorescence the worm body region may have reliably higher intensity than the image background. The worm body may be even more visible when all z-sections in a 3D worm image stack are overlaid together. As biologically *C.elegans* can only bend along the dorsal-ventral direction except at the tip of the head, a worm often lies laterally on a slide. Therefore, summing up all z-sections is a reasonable simplification of the 3D straightening problem to a 2D problem. Worm boundaries can also be extracted in bright-field, Nomarski and other imaging modalities. Thus we develop an algorithm, BDB^+ , for *backbone detection with a clear worm boundary* for 2D images.

We assume the 2D worm boundary has already been extracted, and the problem is how to find the best backbone given this boundary. The desired backbone control points are those having equal distances to the left and right worm boundaries with a regular spacing between them.

Note that our definition is similar to the ‘morphological skeleton’ of an elongated shape. A morphological skeleton can be detected using the medial axis transform (Blum, 1964). Image morphological operations such as thinning can also generate similar results (Jain, 1989). However, the skeleton found in this way is often broken or branched and needs post-processing such as skeleton linking to rectify these artifacts.

We start with the idea of Subsection 3.1, using the diameter of an MST of a randomly selected set of points to initialize all the control points of a backbone curve, and then develop the iterative procedure below that refines the control points to be equidistant from the left and right boundaries. The one difference in the initialization step is that W is now defined as the region within the supplied boundary B .

We seek to refine the control points so that they are equidistant from the left and right boundaries and well spaced along the length of the worm. But first, in order to do so, we must divide the supplied boundary B into a left and right half, B_L and B_R , respectively. Initially, this is done crudely by using the points on the boundary B that are closest to the first and last control points as the division points between the left and

right halves. Then the first and last control points are removed as you will see shortly that they would not properly adjust in the next step given this crude division.

Let $B_L(c)$ be the closest point to c on the left boundary and let $B_R(c)$ be the closest point to c on the boundary. For each control point c , we adjust its position to be midway between $B_L(c)$ and $B_R(c)$:

$$c^{\text{new}} \leftarrow \frac{1}{2}(B_L(c) + B_R(c)) \quad (11)$$

We iterate Equation (11) for each control point until the position of the control point stabilizes. Typically this occurs very rapidly, in 3–6 rounds.

Now that the control points are positioned properly, we recompute B_L and B_R , this time using the point midway between $B_L(c)$ and $B_R(c)$ along the perimeter of B , with c being the first control point and the last control point, respectively. These two midway division points are then added as two control points, corresponding to the anterior and posterior tips of a worm.

Due to the random initialization, the initial spacing of the control points is generally not uniform. We first remove one of any pair of points that are too close together, arbitrarily picking the point to discard. With the remaining sequence we add a control point midway between each pair and then iteratively refine its position with Equation (11).

The BDB^+ algorithm is summarized as follows:

- (1) Initialize the control points as the diameter of the MST of the complete distance graph whose vertices are randomly selected from within the supplied boundary B .
- (2) Divide B into B_L and B_R using the closest points to the first and last control point as the division points. Remove the two control points.
- (3) Update each control points using Equation (11).
- (4) Divide B into B_L and B_R using the points midway along B between $B_L(c)$ and $B_R(c)$ as the division points, where c is the first and last control point, respectively.
- (5) Delete points that are too close together. Insert a control point midway between each consecutive pair of control points and refine with Equation (11)

Given an accurate boundary B , the algorithm, although admittedly heuristic, is fast and produces an accurate A/P axis. Indeed, this was the first algorithm we designed for the problem. However, because the tail is faint, the boundary estimation there is wrong $\sim 10\%$ of the time and one ends up truncating the tail. Moreover, the head does move in the dorsal–ventral axis so a 3D approach is preferred. These two problems, ultimately lead to our developing the slower but more robust BDB^- algorithm.

4 EXPERIMENTS AND DISCUSSIONS

We evaluate WSA using both 2D and 3D images. In the following, we first compare the performance of several methods, and then show the significance in straightening 3D images using WSA.

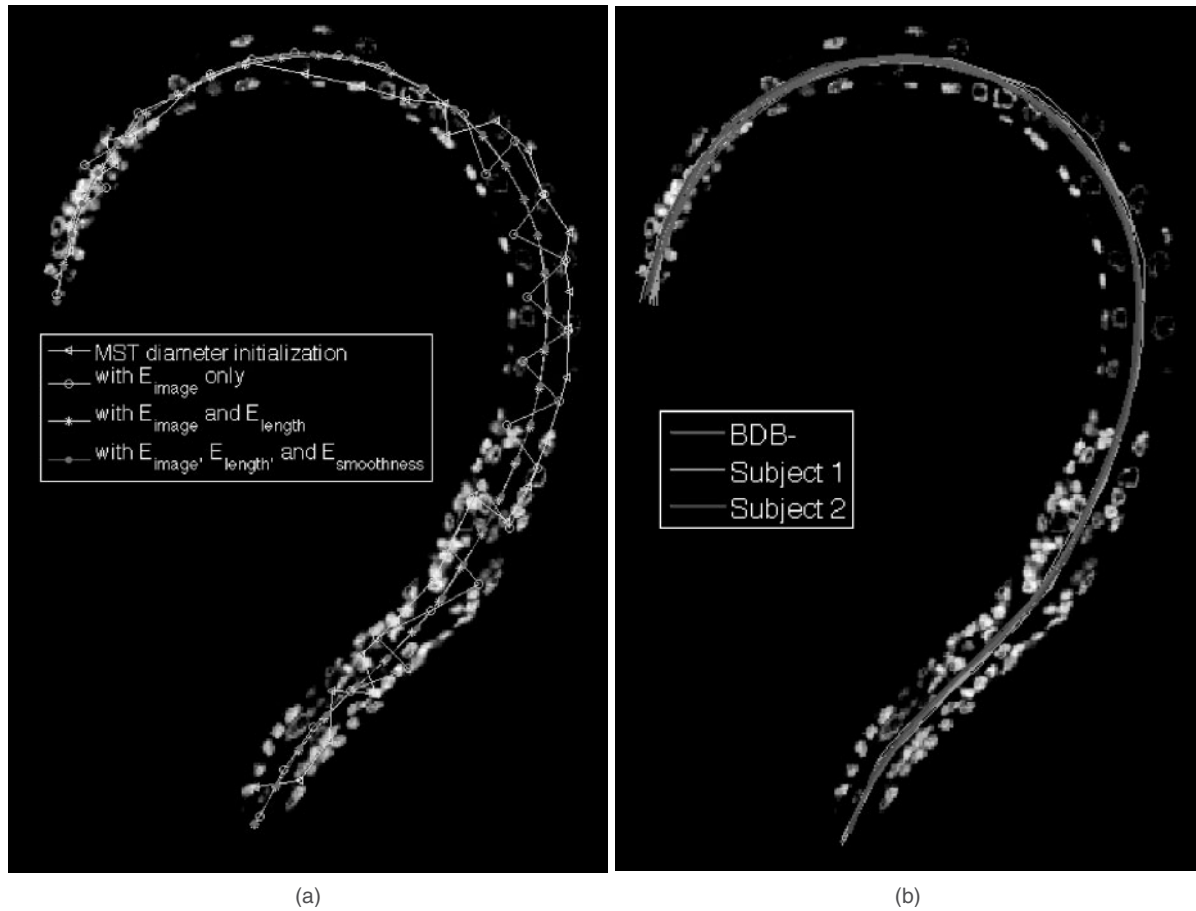


Fig. 2. Backbone detection using BDB^- . Different energy terms (plotted in different colors) are used for comparison. (a) Comparison with different energy terms. (b) Results of 10 BDB^- backbones (obtained based on 10 random initializations) overlaid together with 20 manually drawn backbones (produced by two subjects).

4.1 Evaluation of BDB^-

For BDB^- , we are not aware of other existing methods that operate under the same conditions, so we investigated the significance of the three energy terms in Equation (7) and compared the performance against backbones drawn manually by human experts.

Figure 2 shows the comparison of the initial backbone, and the results with various combinations of energy terms included in the total energy. For better visualization of these results, instead of drawing the resulting cubic splines, we just draw straight lines between consecutive control points.

Figure 2a illustrates the contribution of the energy terms for BDB^- . The initial backbone that is the diameter of the MST of randomly selected bright pixels is shown in yellow. It has the overall trend of the backbone, but deviates significantly from the mid-axis of the worm. The green line, which is the result obtained if only the E_{image} term is minimized, is reasonably meaningful for the top-left portion of the worm, but is not correct for the remainder. The major reason is that there is no constraint on the backbone length. With the length term added in, the cyan curve, the backbone becomes quite reasonable. It is refined further by the addition of the smoothness term giving

the red line. In some cases just the length or smoothness term alone gives a good curve, but across multiple trials we observed empirically that both were necessary to get a consistently acceptable result. One example is shown in Fig. 3 that demonstrates that using all energy terms leads to a better result than using E_{image} and E_{length} , but leaving out $E_{\text{smoothness}}$.

While visually the BDB^- result obtained using all three energy terms is reasonable, it is interesting to ask how well it coincides with what a human observer would consider to be the medial axis. So we compared the BDB^- result against backbones manually drawn by human subjects. With 10 different random initializations, we produce 10 BDB^- backbones and asked two human subjects to each draw 10 backbones manually. We overlaid all 30 backbone curves in Figure 2b. It is apparent that the BDB^- backbones agree well with the manual drawings.

Because the BDB^- backbones vary between runs on the same image due to the random initialization, we sought to measure how much variation there was from run to run on the same data. Due to the fact that different runs also generate backbones with different numbers of control points, it is hard to compare them in curve-space. Instead we used the backbones

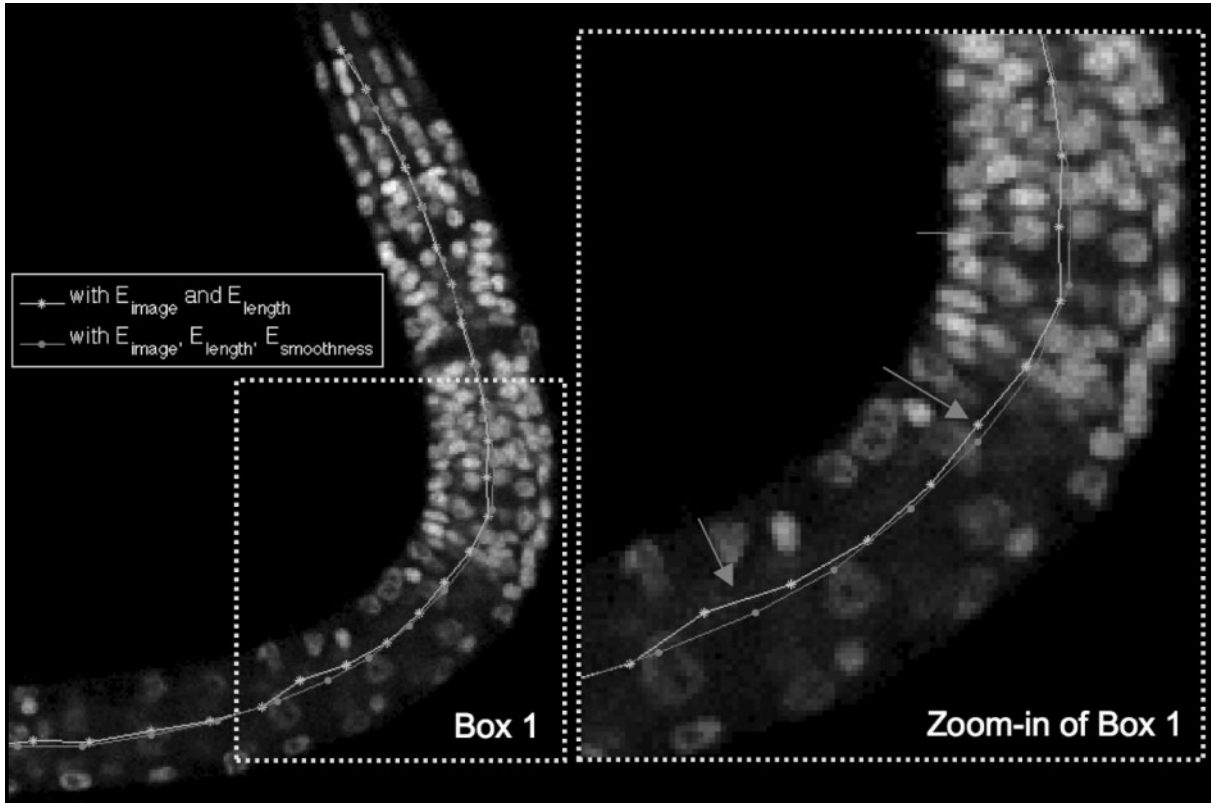


Fig. 3. Comparison of backbones detected with and without the smoothness energy term. The less satisfactory regions of the backbone curve are labeled using purple arrows in the zoom-in box.

to straighten the worm, and then computed the maximal cross-correlation (Haralick and Shapiro, 1992) between each pair of straightened worm images. As summarized in Table 1, the average cross-correlation of pair-wise BDB^- backbones is 0.944 ± 0.037 , which is higher than those for the two human subjects, i.e. 0.828 ± 0.052 and 0.911 ± 0.020 . One would expect that a computer algorithm would be more consistent and have less variation, and this is confirmed when one computes the same statistics for the manually drawn backbones. In addition, cross-correlation between the BDB^- and the human subject results (i.e. 0.838 ± 0.047 and 0.834 ± 0.032) are higher than the correlation between the two subjects (i.e. 0.822 ± 0.045), indicating our computer program can generate results that better agree with, or at least are comparable to, the manual work of different subjects.

We also tested the robustness of BDB^- for perturbed image data, similar to the case of sampling variation during image acquisition using a microscope. We randomly set 10% and 20% of the image pixels to have 0 intensity, and for each case we produced 10 independently perturbed images, which we then straightened. We computed the pair-wise maximal cross-correlation between each case and the 10 trial results produced using the original data. As shown in Table 2, the average pair-wise correlation among these different sets of straightened images is consistently high, i.e. 0.945 ± 0.031 and 0.945 ± 0.033 , indicating the BDB^- method is very robust.

Table 1. Average normalized cross-correlation of straightened images of the same input worm image

Score	BDB^-	Subject 1	Subject 2
BDB^-	0.944 ± 0.037	0.838 ± 0.047	0.834 ± 0.032
Subject 1	—	0.828 ± 0.052	0.822 ± 0.045
Subject 2	—	—	0.911 ± 0.020

As the matrix is symmetrical, the unnecessary entries are replaced using ‘—’.

Table 2. Statistics of cross-correlation of straightened images for perturbed input image via randomly setting the intensity of 10% or 20% image pixels to 0

Score	10%	20%
Original image	0.945 ± 0.031	0.945 ± 0.033

For each case, the statistics was drawn based on 10 independent trials, and thus 100 pair-wise maximal cross-correlation values between this case and the 10 independent trials for the original image.

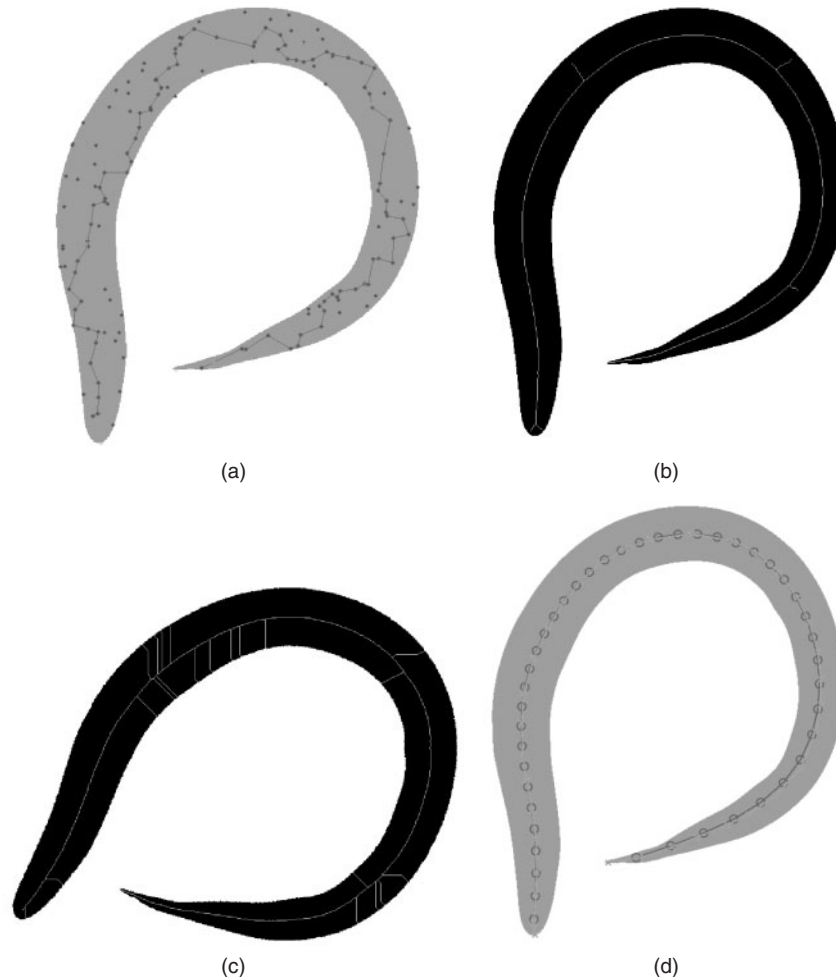


Fig. 4. Backbone detection results using different methods. (a) Our BDB^+ result, where the backbone (green curve and dots) evolves from the MST diameter (red line) produced for a random sub-graph, whose vertexes (150 blue and red dots) are randomly sampled from the entire set of pixel vertexes on the worm body ($>80\,000$ pixels in this image). (b) Morphological image skeleton. (c) Morphological image thinning for the image rotated 30° . (d) The backbone detected using BDB^- (red) overlaid with the BDB^+ backbone (green).

4.2 Evaluation of BDB^+

Figure 4a shows the result of our WSA BDB^+ method applied to 2D images obtained by projecting the sum of pixel intensities along the z-axis. We initialize the control points randomly (blue and red dots) and identify the diameter of the MST (red line) spanning these points. The points on the diameter are then adjusted with Equation (11) till they converge to the stable backbone shown in green. BDB^+ is stable with respect to the random selection of initial control points and the orientation of the worm body. Its computational complexity is proportional to the number of control points used and the length (in pixels) of the worm body boundary.

Figure 4 also shows the comparison results of morphological image skeleton (Fig. 4b) and thinning (Fig. 4c) algorithms. Morphological operations typically lead to skeletons that are branched as seen in the figures or less frequently have small breaks that are not noticeable at the scale drawn here. In addition, these morphological procedures are often sensitive to

the orientation of a worm body, as exemplified by Figure 4c. In regards to computational complexity, these methods are linear in the number of image pixels. Post-processing such as branch removal and major-segment linking, are usually needed to transform their initial results into meaningful backbones. Note that one possibility is to use the MST of BDB^+ to refine the results of image morphological operations.

Figure 4d shows the backbone produced by BDB^- for the same image, without using the worm boundary information. This example demonstrates both BDB^- and BDB^+ generate reasonable backbones that are also close to each other.

4.3 Application to 3D worm image straightening

We now discuss the application of WSA to its intended target, straightening high-resolution 3D fluorescence images of worms.

First, there is a minimal loss of image information in straightening, as shown in Figure 5. The original data has three color channels, each for a different set of cells. The blue channel

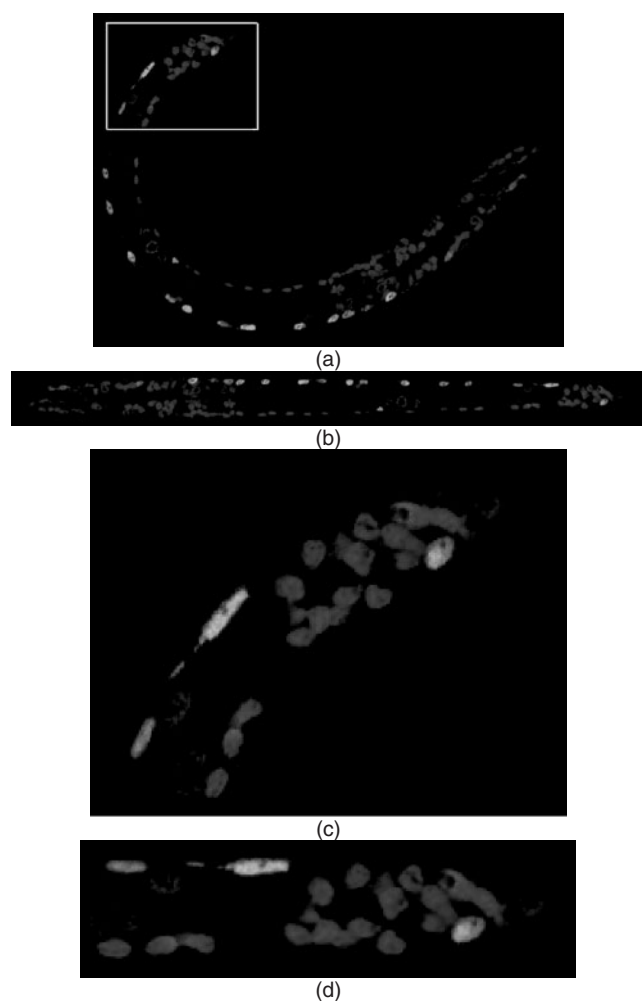


Fig. 5. Straightening results on a 3D image. Only one z-section is shown. (a) The original worm (25% of the real size), (b) the straighten worm (20% size), (c) tail region of the original worm (80% size) and (d) tail region of the straightened worm (80% size).

is the DAPI staining of nuclear DNA in every cell. The green channel is nuclear-localized GFP in the 81 body wall muscle cells, and the red channel is mCherry in a subset of target cells of biological interest. It can be seen that the original worm body in Figure 5a is straightened into a rod-shape in Figure 5b. The blow-up views of the tail region in Figure 5c and d allow inspection of the details. Because we use 1-pixel spacing between all cutting-planes to restack the worm, the resolution in the straightened image is the same as the original image, although the straightened worm may look slightly smoother due to sub-pixel interpolation. The overall loss of information is minimal, as can be seen in the example of the tail region of Figure 5c and d. We have not observed any visible distortion of nuclear features (e.g. intensity) in any of the stacks generated to date.

We quantitatively measured four features of randomly selected image objects (e.g. a single nucleus or a group of touching nuclei—the latter was also considered for generality)

Table 3. Examples of features of corresponding image objects (a single nucleus or a group of nuclei) that were independently extracted from images before and after straightening

Image object	Type of information	Size	Mean intensity	Surface area	Circularity
A	Before	8010	39.643	2138.801	1.161
	After	8080	39.065	2179.074	1.184
	Absolute difference	70	0.579	40.272	0.023
	Ratio	0.009	0.015	0.019	0.019
B	Before	13799	42.868	4018.091	1.736
	After	13229	42.688	3908.548	1.737
	Absolute difference	570	0.180	109.543	0.001
	Ratio	0.042	0.004	0.028	0.001
C	Before	16846	41.149	3952.034	1.387
	After	16770	40.321	3910.396	1.371
	Absolute difference	76	0.8287	41.638	0.0157
	Ratio	0.0045	0.0203	0.0106	0.0114
Average ratio of 24 objects		0.0285	0.0089	0.0233	0.0091

For the second column, ‘type of information’, we showed the features before and after the straightening, as well as the absolute value of the difference, and the ratio defined as $\text{difference}/(0.5 * (\text{before} + \text{after}))$. Also showed are the overall statistics of 24 image objects randomly selected from 3D images of three *C.elegans* individuals and measured in the same way.

Table 4. 3D image-stack file sizes before and after straightening

Image number	Before (MB)	After (MB)	Size reduction (%)	Average size reduction (%)
1	358	74	79.33	79.76
2	352	68	80.68	
3	393	80	79.64	
4	412	85	79.37	

before and after straightening: (1) size (number of voxels), (2) mean pixel intensity, (3) surface area and (4) circularity (quotient of the length of long axis against that of the short axis of an image object). Table 3 gives examples of these quantities. It can be seen that the differences between image object features before and after straightening, manifested by the respective ‘ratio of difference’ values, are consistently small, i.e. around or $<2-3\%$ for all four types of features.

We have applied WSA to our on-going digital worm cell atlas project (Long *et al.*, 2007), that routinely produces large 3D worm image stacks. Table 4 gives several examples of image file sizes before and after straightening. For these examples WSA reduces the image file size by nearly 80% on average, which is significant for our project as it involves thousands of such images. Obviously, one could achieve the same space savings by sparsely encoding the relevant pixels in an unstraightened worm, but this is considerably more complicated than keeping the rectilinear image stack after straightening.

With a straightened worm body, it is easy to measure morphological features of a worm. For example, we measured the length of 31 straightened first larval stage worms that are synchronized to be within 2 h after hatching. On average their length is $212.75 \pm 19.11 \mu\text{m}$.

ACKNOWLEDGEMENTS

The authors thank Andrew Fire for providing reagents and advice. 3D image stacks were generated in the Cell Sciences Imaging Facility of Stanford University. The authors acknowledge the financial support of the Larry L. Hillblom Foundation for X.L. The work of X.L. and S.K. was funded by the Ellison Medical Foundation and the NIH.

Conflict of Interest: none declared.

REFERENCES

- Al-Kofahi, K.A. et al. (2002) Rapid automated three-dimensional tracing of neurons from confocal image stacks. *IEEE Trans. Inf. Technol. Biomed.*, **6**, 171–187.
- Blum, H. (1964) A transformation for extracting new descriptions of shape. In *Proceedings of the Symposium on Models for the Perception of Speech and Visual Form*. MIT Press, Cambridge.
- Corman, T.H. et al. (2001) *Introduction to Algorithms*. MIT Press, Cambridge, MA, USA.
- Fire, A. et al. (1998) Potent and specific genetic interference by double-stranded RNA in *Caenorhabditis elegans*. *Nature*, **391**, 806–811.
- Haralick, R.M. and Shapiro, L.G. (1992) *Computer and Robot Vision*. Vol. II, Addison-Wesley, pp. 316–317.
- Hastie, T. (1994) Principal curves and surfaces. *Ph.D. Thesis*. Stanford University.
- Jain, A.K. (1989) *Fundamentals of Digital Image Processing*. Prentice Hall.
- Kass, M. et al. (1987) Snakes: active contour models. *Int. J. Comput. Vis.*, **1**, 321–331.
- Kégl, B. et al. (2000) Learning and design of principal curves. *IEEE Trans. PAMI*, **22**, 281–297.
- Long, F. et al. (2007) A 3D digital cell atlas for the first larval stage of *C. elegans* hermaphrodite. *HHMI JFRC Technical Report*.
- Luengo Hendriks, C.C. et al. (2006) 3D morphology and gene expression in the *Drosophila* blastoderm at cellular resolution I: data acquisition pipeline. *Genome Biol.*, **7**, R123.
- Mace, D.L. et al. (2006) Quantification of transcription factor expression from Arabidopsis images. *Bioinformatics*, **22**, e323–e331.
- Sun, Y. (1989) Automated identification of vessel contours in coronary arteriograms by an adaptive tracking algorithm. *IEEE Trans. Med. Imag.*, **8**, 78–88.
- Xu, C. and Prince, J. (1998) Sankes, shapes, and gradient vector flow. *IEEE Trans. Image Process.*, **7**, 359–369.