

Fast, versatile and quantitative annotation of complex images

Kathleen Bates^{1,2}, Shen Jiang³, Shivesh Chaudhary^{4,2}, Emily Jackson-Holmes^{4,2}, Melinda L Jue², Erin McCaskey³, Daniel I Goldman⁴ & Hang Lu^{*1,2}

ABSTRACT

We report a generic smartphone app for quantitative annotation of complex images. The app is simple enough to be used by children, and annotation tasks are distributed across app users, contributing to efficient annotation. We demonstrate its flexibility and speed by annotating >30,000 images, including features of rice root growth and structure, stem cell aggregate morphology, and complex worm (*Caenorhabditis elegans*) postures, for which we show that the speed of annotation is >130-fold faster than state-of-the-art techniques with similar accuracy.

METHOD SUMMARY

We developed a smartphone app for Android phones in Java, utilizing Firebase as the database backend for storing images and user drawings. Images from the database are either randomly or sequentially presented to users, who can freeform draw on each image and upload the annotated image and annotated contours to the database for quantitative analysis. The app conforms to material design and focuses on a clean user interface for better usability and a smooth drawing experience.

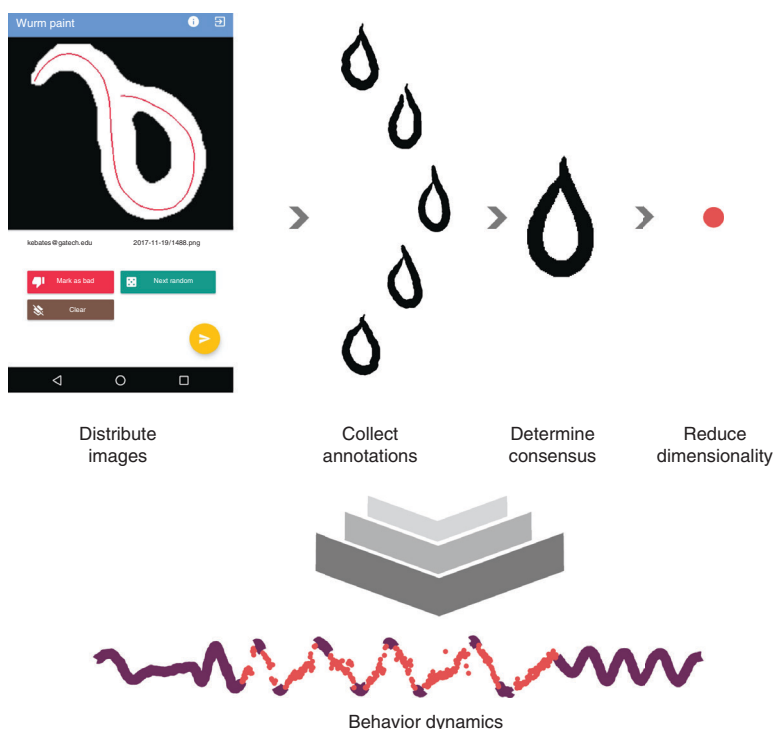
KEYWORDS:

behavior • *Caenorhabditis elegans* • image analysis • root systems • smartphone app • software • stem cell aggregates

¹Interdisciplinary Program in Bioengineering, Georgia Institute of Technology, Atlanta, GA, USA; ²School of Chemical & Biomolecular Engineering, Georgia Institute of Technology, Atlanta, GA, USA; ³School of Computer Science, Georgia Institute of Technology, Atlanta, GA, USA; ⁴School of Physics, Georgia Institute of Technology, Atlanta, GA, USA; *Author for correspondence: hang.lu@gatech.edu; [†]These authors contributed equally

BioTechniques 66: 269-275 (June 2019)
10.2144/btn-2019-0010

GRAPHICAL ABSTRACT



The accelerating ease of collecting very large image data sets (terabytes to petabytes) has led to a shift in scientific bottlenecks from image collection to image analysis across many disciplines, including connectomics [1–3], cell lineage tracing [4], and ethology [5–8]. Although highly specialized computational pipelines are emerging to address this new bottleneck, these pipelines require significant effort to develop, are computationally expensive and not error-free, and may still rely on human image annotation to establish ground truths. The widespread dependence on human image annotation or correction is likely to continue, and yet tools for image annotation, especially at large scales, often do not meet the needs of researchers.

Specifically, tools for quantitative annotation of images are hindered by a

trade-off between speed, accuracy and versatility. Some automated tools require extensive tuning or parameter optimization prior to annotation to enhance accuracy, and many image processing pipelines are not well-suited for heterogeneous image sets. In addition, many tools for human annotation limit the way users can define image features of interest, for example, via rectangles, polygons or circles [9]. Annotation speed is limited by the complexity of annotation software and, ultimately, how quickly annotators can mark phenotypes accurately [10]. Equally critical for efficient annotation of large datasets is ease in distributing annotation tasks, as well as broadness in settings or locations where users can annotate. To serve the greatest number of researchers effectively, tools for large-scale image annotation should be generalizable, fast and accurate.

► Here we report a highly versatile, fast and quantitative method for image annotation. Features of interest of an arbitrary image can be annotated simply from users' finger- or stylus-tracings (Supplementary Movie 1). We demonstrate the use of a simple and intuitive smartphone- and tablet-based app to annotate complex body postures in *Caenorhabditis elegans*, morphology of stem cell aggregates, and root growth of *Oryza sativa* (rice) and *Zea mays* (corn). We crowd-sourced annotations of over 16,000 nematode images, 500 stem cell aggregate images, and 900 root images, with a total of over 30,000 user annotations (Figure 1A–E).

MATERIALS & METHODS

Development of image annotation app

The app is written in Java, utilizing Android Studio to package the app for Android phones. Briefly, the app loads images from an online database managed on Firebase to the user's Android device, on which users draw their best annotation. The structure of our database is shown in Figure 2. The user then has the option to clear their annotation and try annotating again before uploading, to report the image as something they are unable to annotate, or to load a new random image from the cloud database. Once the user is satisfied with their annotation, they upload their annotated image (as well as vectors of the annotation's trajectory) and are immediately presented with another image from the image set. The app conforms to material design and focuses on clean user interfaces for better usability and a smoother drawing experience. During beta-testing, we recruited six users, four of whom work with *C. elegans* on a daily basis, with the other two having some general knowledge of the worm. Although we collected more annotations from other users, including children, only the annotations from these six users were used in our behavioral analysis of *C. elegans*, and they make up >90% of the total annotations collected.

User annotation speed

We collect timestamps when users upload images and drawing vectors with a resolution of 1 s, based on the user's device's time. To determine a conservative average user annotation speed, we grouped all annotations

by user and computed the time between each upload for that user. All inter-upload times were pooled. Because inter-upload times could range from a few seconds to days depending on the user's usage frequency, we imposed an upper threshold of 30 s for worm image annotations and an upper threshold of 90 s for root image annotations to determine the average user annotation speed.

Worm tracking

We built upon an existing worm tracker [11] for our initial image analysis and to identify movie frames where worms were partially self-occluded (i.e., ambiguous). A subset of these frames was uploaded to our database for annotation. Using the generative algorithm included in the existing worm tracker to predict worm posture for occluded shapes, we optimized parameters for our data set and found predicted worm postures for several full videos from which we had drawn ambiguous postures for our database. The worm tracker uses MATLAB software (we used MATLAB version 2017a). To evaluate the time required to process individual frames using this worm tracker, we used MATLAB to measure how long the point-swarm (PS) optimization (generation of alternative posture predictions) required for each ambiguous frame. This step took an average of 776.2 ± 5.2 s/ambiguous frame (95% CI; $n = 66$) with parallel processing (a local pool consisting of four cores). After this generative step, a progressive optimizing interpolation (POI) step evaluates the alternative posture predictions to determine which makes sense in the context of the worm postures in the surrounding frames. For this step, we timed the total time until a solution was generated. For a movie with 444 ambiguous frames, this step required 155.5 s/ambiguous frame (equivalent to the time required for >20 human annotations). Combined, the PS and POI tracking steps required on average 931.7 s/ambiguous frame, or the equivalent of 133 human annotations. The computers used were Dell Precision Tower model 5810 with 32 GB RAM and Intel Xeon CPU (model E5-1620 v4, 3.5 GHz).

Similarity score calculation

In order to compare two worm annotations, or a worm annotation to ground truth, we matched 100 points between two worm

midlines and computed the Euclidean distance between each pair, summing all of these distances and normalizing the distance by 75% of the width of that particular worm at each of the 100 matched points. In some cases for earlier versions of the app, we reconstructed occluded (ambiguous) worm shapes from user drawings as opposed to user vectors (Supplementary Figure 1A). Mathematically this is represented as:

$$1 - \sum_{i=1}^{100} \frac{\| \text{midline}_1 - \text{midline}_2 \|_2}{0.75 \cdot \text{worm thickness}_i}$$

Consensus generation

To construct consensus midlines from user annotations, we noted that even for pairs of reconstructed midlines that were below a zero similarity score, users were making essentially the same annotation. To identify a threshold similarity score below which we could consider two annotations to be from distinct groups, we modeled the distribution of similarity scores from user–user comparisons (Figure 3C) as a mixture of gaussians. The primary mode was centered at -0.068 and the secondary mode was centered at -3.260 (Supplementary Figure 1B). To ensure that most generally similar annotations were grouped together, we computed a threshold two standard deviations below the primary mode, a similarity score value of -0.809. We found that several other methods of identifying this similarity score threshold identified thresholds that ranged from slightly positive to slightly negative. These methods included the Otsu thresholding method on user–user similarity scores and searching for the lowest threshold of the user–user similarity scores for which the Wilcoxon rank-sum test failed to reject the null hypothesis that the user–user similarity scores and user–ground truth similarity scores were drawn from the same distribution at the 5% significance level.

Having identified a reasonable threshold, we generated consensus contours. During this process, we used the projections of worm backbones into the space of the first five eigenvectors. We identified and removed annotations whose eigenvector projections were outside of the range of *C. elegans* posture space. Then, for each source image, we identified all annotations of the source image and removed any remaining outlier annotations of that image, where an outlier is a value more than three absolute devia-

tions away from the median. We computed similarity scores for all pairs of annotations and used our previously identified threshold to identify pairs of images that were very similar to one another. We further grouped these pairs into larger groups of similar annotations and identified the group of similar annotations with the largest number of members. For example, if image pairs (1, 2), (2, 3) and (5, 6) all have similarity scores above our threshold, we take the union of all pairs that contain images 1, 2 and 3 and, separately, the union of all pairs that contain images 5 and 6. If more images belong to the first union set than the second, we use the first set to calculate a consensus contour by finding the centroid of this group of contours in the 5D space of posture projections.

RESULTS & DISCUSSION

Our app is indiscriminate to the nature of images or annotations. Worm images on our database were derived from brightfield and darkfield microscope configurations, solid and liquid imaging environments, and included both processed and binarized images (Figure 1B). Stem cell aggregate images on our database were derived from phase images of both live and fixed aggregates grown in tissue culture plates as well as aggregates grown in microfluidic devices [12] (Figure 1C). For both nematode and stem cell aggregate applications, users are presented with randomized images from the full dataset and draw a single contour. This generic annotation scheme could also be used to trace individual cells or features of developing embryos (such as *Drosophila melanogaster*, *Xenopus* or zebrafish), to name a few. To allow users to annotate video frames in a pre-defined order (e.g., when temporal context is critical to annotation) and in cases where an image contains multiple features of interest, we created a second version of the app that presents uploaded images in order and allows users to draw as many contours as needed. We used this app version to annotate rice and corn root systems (Figure 1D–E). We expect that these two versions of the app could serve many other image annotation problems equally well with little to no changes of the source code.

The app is extremely easy for annotators to use. By using smartphones as the basis

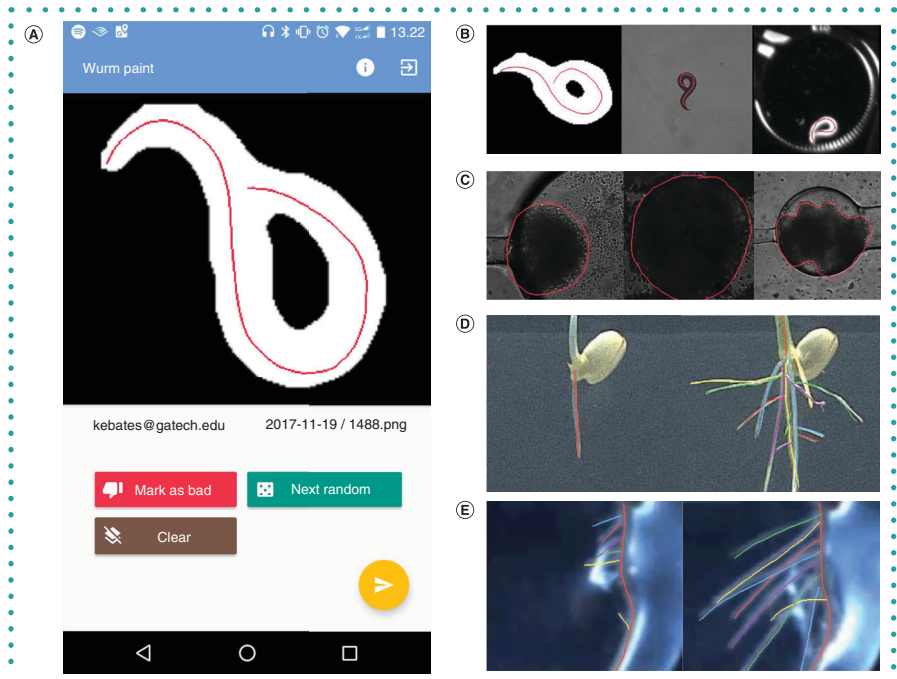


Figure 1. Versatile smartphone annotation of images. (A) Screen capture of Android interface of worm tracing app, 'Worm Paint'. See Supplemental Movie 1 for a video of the app in use. (B) User annotations of worm posture in binary image, grayscale brightfield image and grayscale darkfield image. (C) User annotations of stem cell aggregate morphology using app with same source code as Worm Paint. (D) User annotations of rice root structure. Left-hand images temporally precede right-hand images. (E) User annotations of corn root structure. Left-hand images temporally precede right-hand images.

for our image annotation system, users need only draw with a finger or stylus, as compared to the greater difficulty of drawing with a computer mouse or, as in ImageJ, drawing piecewise lines. The interface itself is simple and intuitive compared to popular image annotation and analysis tools. We had 7–12 year-olds use the worm tracing app, and found that it was simple enough for them to use without help after a brief explanation (Figure 3A). Although the quality of children's annotations was far more variable than annotations by adults, many of the children's annotations were of indistinguishable quality compared to those of adults and annotations inconsistent with other user's annotations were easy to identify.

We sought to demonstrate that our app enabled fast annotation. For two of our applications, we quantified the time between image uploads of individual users as a conservative estimate of time per annotation. For worm tracing, which always required a single user-drawn contour, the average annotation time was

7 ± 0 s/image (95% CI) and for root tracing, which often required multiple contours per image, the average annotation time was 14 ± 1 s/image (95% CI). To benchmark user annotation speed in our app, we annotated worm images using ImageJ [13], which routinely required more time. In addition to the importance of individual users' speed, overall speed is dependent on how many users can annotate in parallel. Smartphone-based annotation not only allows us to easily distribute image annotation tasks as narrowly (a single expert) or broadly (general public) as desired, it also expands geospatial locations and settings where users can annotate [14].

We assessed the ability of users to trace known shapes accurately. We did this by comparing averaged hand-drawn worm postures to computationally generated ground truth postures. For worms with unambiguous postures, we matched points along the averaged hand-drawn worm midlines with points along the corresponding ground truth midline and summed the Euclidean norm of all point pairs. To ►



Figure 2. App database structure. Top-level (left) and expanded (right) structure of the root tracing app database. All apps have similarly structured databases. 'bad_images' contains mapping to user-reported images. 'master_upload' defines which source image sets are live on the app, as well as the number of images in each source set. User feedback is stored in the 'ratings' structure. Finally, 'uploads' maps user annotations (with user id, image name, and date and time of annotation) to the source image. In newer app versions available on our Github, we also save line trajectories at the bottom of the 'uploads' structure. To initialize the app, only the 'master_upload' structure is needed.

► determine an overall similarity between any two worm midlines, we reasoned that an acceptably similar midline should lie within the center three-quarters of the worm's total width at any given point. We therefore normalized similarity scores so that a score of one indicated identical midlines, any positive score indicated that the midlines were on average less than three-quarters of the worm width apart, and negative similarity scores indicated that midlines were further than three-quarters of the worm width apart (Figure 3B). Most averaged annotations of unambiguous postures had similarity scores above zero when compared to their corresponding ground truth midline, including data collected from non-expert annotators (Figure 3C). We concluded that the annotation accuracy was sufficiently high for tracing worms.

To further demonstrate a practical application of our app, we focused on using annotations of ambiguous *C. elegans* postures to reconstruct the dynamics of worm behavior. Ambiguous postures result from segmentation errors or, more frequently, the worm partially occluding itself, for example during stereotyped Ω - or δ -turns. A major advantage of using human annotators is the ability to quickly generate varied predictions for images that humans and algorithms alike struggle to find a ground truth for. *C. elegans* postures are often simplistic and sinusoidal, but ~7% of the worms' behavior results in postures that are impossible to segment using current tools. One approach relies on computationally expensive optimization to attempt a quantitative posture description [11,15]. Although accurate in most instances, this state-of-the-art strategy for predicting

ambiguous nematode posture requires on average 931.7 s ($n = 66$) per video frame. Based on our average worm annotation time, users can make predictions approximately 130-fold faster than this computational strategy. User predictions for individual ambiguous images varied, but could typically be grouped into several distinct shapes, indicating that there were often only a few reasonable predictions for each ambiguous posture (Figure 4A). To characterize this variability quantitatively, we calculated pairwise similarity scores comparing different annotations of the same image for >500 source images and found that similarity scores peaked between zero and one and had a left-skewed distribution with a significant tail (Figure 4B). This is consistent with our observation that although there is significant variability in user annotations, users are frequently in agreement with one another, suggesting the utility of a consensus-based approach in identifying a best solution. The ease and speed of generating viable predictions based on human intelligence with the app gives it particular advantage in analyzing images where a single 'correct' solution is non-existent and several solutions have high likelihood.

To resolve the ambiguities in our postural data set, we used annotations to create a consensus prediction for ambiguous images (Figure 4C). For each source image, we first eliminated annotations that were outliers or that created shapes outside of *C. elegans* postural space, then used pairwise similarity scores to identify groups of similar annotations. We chose the group containing the most individual annotations, and averaged annotations in this group to come to a consensus contour. We compared these disambiguated annotations to predictions generated by the state-of-the-art computational method and found that the mode of the similarity score distribution was -1, indicating that although consensus contours had somewhat reduced accuracy, they overall agreed well with computational predictions (Supplementary Figure 1B). Further, for frames where initial segmentation failed, users could correctly annotate grayscale source images, while computational predictions were erroneous.

C. elegans is a powerful model organism with a large suite of tools for genetic manip-

ulation [16,17]. These tools, along with a fully mapped nervous system [18], have enabled researchers to identify molecular mechanisms and individual genes associated with behavioral phenotypes [19–21]. However, quantitative analysis of some of the most complex behaviors, large-angle turns that commonly include ambiguous postures, remains difficult, and gaps in quantifiable behavior prevent dynamic posture analysis altogether. Using our consensus worm contours, we recreated the postural repertoire and behavioral dynamics of *C. elegans*. We sought to answer how significantly complex worm postures affect the overall shape space of *C. elegans*. To answer this, we calculated the first four principle components of *C. elegans*' shape space [22] ('eigenworms') using either unambiguous results alone or both unambiguous results and consensus contours (Figure 5A, Supplementary Figure 1C). Consistent with prior reports, we found that the first four principle components were very similar with or without ambiguous postures [11]. Interestingly, the fractional variance of the worm's posture space captured by these eigenworms is greater when ambiguous postures are included (Figure 5B). Lastly, we recreated complete timeseries of the first four eigenworm amplitudes for individual worms using the consensus contours (Figure 5C). These traces fill in the gaps left by ambiguous shapes and outperform the computational prediction in some cases where the worm is tightly coiled (Supplementary Movie 2). In addition to adding to our knowledge of *C. elegans* behavioral dynamics purely through image annotation, this app can help improve existing posture prediction algorithms by using these results.

Our app-based annotation scheme allows researchers from any field to quickly and easily annotate complex images in quantitative ways. Here, we demonstrated its flexibility and speed in annotating rice root growth and structure, stem cell aggregate morphology and complex worm postures, where we showed that the app is ~130-fold faster than state-of-the-art posture optimization techniques. We expect that the app will be useful as an alternative to creating complex and bespoke computational image processing pipelines, as a way to complement and augment existing compu-

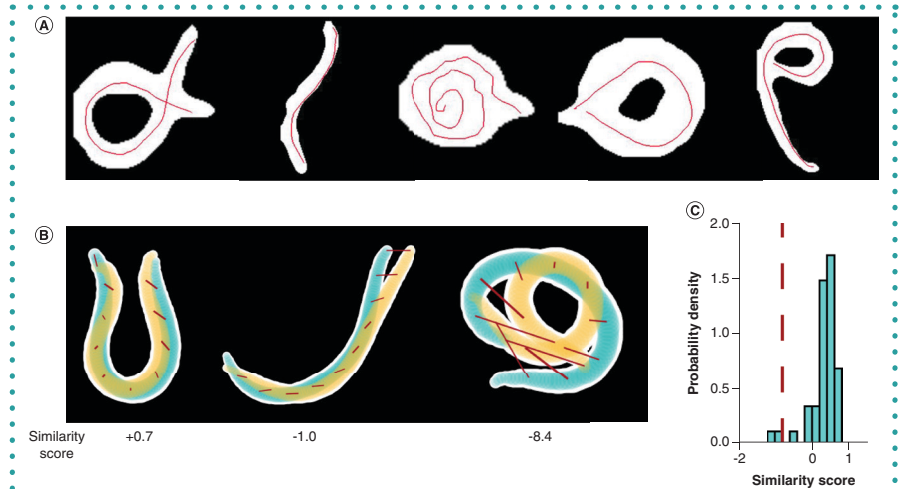


Figure 3. Intuitive and accurate annotation of images. (A) Annotations by 7–12 year-olds. (B) Sketch of similarity score calculations. For each panel, two worm contours (white overlaid with yellow or blue) are reconstructed based on ground truth or annotated midlines. 100 points along the midline are matched, and the Euclidean distance between each pair is computed. Here we show this at ten points along the backbone (red lines). Yellow and blue highlight the center three-quarters of the worm's width. (C) Probability density of similarity scores for unambiguous posture solutions compared to averaged user annotations of the same unambiguous postures. The dashed red line indicates the threshold we use to calculate consensus contours. $n = 44$.

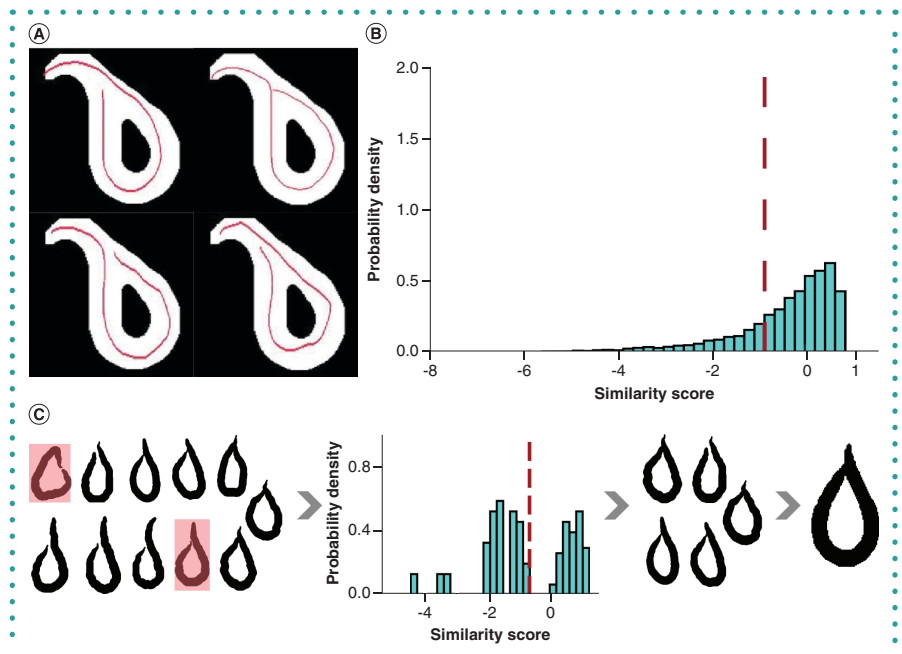


Figure 4. Generating accurate consensus predictions for ambiguous images. (A) Example set of ambiguous images. (B) Probability density of similarity scores for comparison between different user annotations of the same ambiguous posture. The dotted red line indicates the threshold we use to calculate consensus contours. The threshold was determined by modeling probability density as a mixture of two Gaussians (see Methods); $n = 26,098$. (C) Illustration of consensus generation scheme.

tational pipelines, and as a simple way to generate consensus ground truths towards improving machine-learning algorithms for image processing.

FUTURE PERSPECTIVE

As image datasets become larger and the demand for more nuanced analysis of complex image features becomes ►

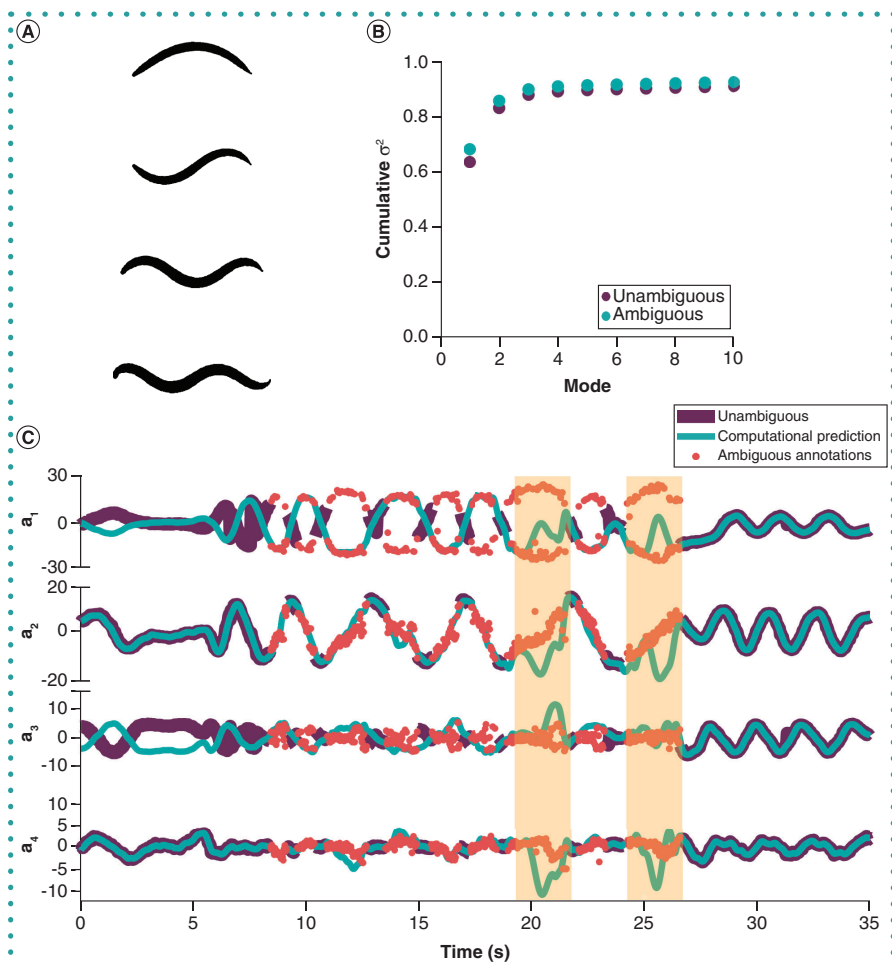


Figure 5. Reconstruction of continuous behavior dynamics from annotations. (A) Representation of first four principle components of worm posture ('eigenworms') including both unambiguous and ambiguous postures from four annotated videos in our dataset (37,784 frames). As reported in other works, our eigenworms both with and without ambiguous postures were very similar (see Supplementary Figure 2C). (B) Cumulative variance captured by each additional eigenworm ('mode') for both unambiguous images only and both unambiguous and ambiguous images together. (C) Traces of amplitude of first four eigenworms in time for an individual worm. Dark purple lines are amplitudes calculated for unambiguous postures via image processing. Gaps in purple lines correspond with frames containing ambiguous postures that are usually associated with reorientation of the worm. Blue lines are computational predictions for the full video, including ambiguous postures. Red dots represent consensus contours for individual frames found using app user annotations. Yellow highlighted regions are time points where computational predictions do not match consensus predictions (highlighted red in Supplementary Movie 2). The top inset image is the unphysical computational prediction at the timepoint corresponding to the grey dashed line. The bottom inset image is the consensus contour prediction generated from the app at the same timepoint. For the amplitudes of the first eigenworm in particular, the red dots follow two opposing sinusoidal contours simultaneously, one contour representing the opposite head orientation of the worm compared to the other contour.

► commonplace, human annotation will remain important in establishing ground truths and correcting outputs from automated analysis pipelines. We will continue to need tools for large-scale human annotation to accomplish these tasks. Here, we demonstrated an efficient method for collecting complex annotations; the time needed for an individual annotation is

shorter than a comparable annotation using other software. Thus, this is a useful tool even when the total number of annotations needed is modestly scaled. For applications that require much greater scale, recruitment of more annotators may be necessary. Several methods for recruiting annotators include paying annotators a small amount of money for each annotation by integrating

with Amazon Turk [9], gamifying the app [23], or creating a citizen science effort [1]. We envision that this type of accurately human-curated images will support (by providing ground truth) and complement (in rare and unanticipated scenarios) machine-learning approaches as they become dominant in image-based analyses for many fields of scientific inquiry.

AUTHOR CONTRIBUTIONS

KB and HL conceptualized the project and SJ developed the app interface and database. KB developed image processing software and analyzed data. KB, MJ, SC, SJ, EJH, and HL participated in data curation. DG and EM contributed root images. KB and HL wrote the manuscript and developed analysis metrics. All authors participated in reviewing and editing the manuscript.

FINANCIAL & COMPETING INTERESTS DISCLOSURE

KB received funding from the National Institute of General Medical Sciences (US NIGMS), award number F31 GM123662. HL received funding from the National Institute of General Medical Sciences (US NIGMS), award number R01GM088333 and the US National Science Foundation STC EBICS 0939511. Some *C. elegans* strains were provided by the CGC, which is funded by NIH Office of Research Infrastructure Programs (P40 OD010440). The authors have no other relevant affiliations or financial involvement with any organization or entity with a financial interest in or financial conflict with the subject matter or materials discussed in the manuscript apart from those disclosed.

No writing assistance was utilized in the production of this manuscript.

ACKNOWLEDGMENTS

The authors thank many beta-testers of the app, including Morningside Elementary School in Atlanta (GA, USA), and Ruth Bates for their extensive annotation using the app. The authors also thank Philip Benfey for supplying seeds.

ETHICAL DISCLOSURE

App annotations used in this study were collected from six beta-testers, including both *C. elegans* experts and non-experts recruited from Georgia Tech and the author's acquaintances. All six beta-testers are either

authors on this paper or acknowledged in the acknowledgments. Several Morningside Elementary School students tested the usability of the app by drawing worm shapes with their finger. Combined, students annotated approximately 30 worm images, none of which were used in worm posture analysis. All children who annotated using the app did so with the verbal consent of their parents, and no demographic or other information was collected from them.

Annotations from users of our example app published on Google Play are not included in this study, but we inform users on our Google Play site, on our app information site (<https://sites.google.com/view/wurm/app-privacy-policy>), and within the app itself what information we collect: user emails so that they may establish an account; annotations they produce; and timestamps of when each annotation is updated. No demographic information is collected from users and we do not contact users via their email or share their email addresses.

OPEN ACCESS

This work is licensed under the Attribution-NonCommercial-NoDerivatives 4.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>

SUPPLEMENTARY DATA

Supplementary data accompany this paper. To view these please visit the journal website at: www.future-science.com/doi/suppl/10.2144/btn-2019-0010

REFERENCES

- Kim JS, Greene MJ, Zlateski A *et al.* Space-time wiring specificity supports direction selectivity in the retina. *Nature* 509(7500), 331–336 (2014).
- Boergens KM, Berning M, Bocklisch T *et al.* webKnossos: efficient online 3D data annotation for connectomics. *Nat. Methods* 14(7), 691–694 (2017).
- Helmstaedter M. Cellular-resolution connectomics: challenges of dense neural circuit reconstruction. *Nat. Methods* 10(6), 501–507 (2013).
- Amat F, Lemon W, Mossing DP *et al.* Fast, accurate reconstruction of cell lineages from large-scale fluorescence microscopy data. *Nat. Methods* 11(9), 951–958 (2014).
- de Chaumont F, Coura RD-S, Serreau P *et al.* Computerized video analysis of social interactions in mice. *Nat. Methods* 9(4), 410–417 (2012).
- Kabra M, Robie AA, Rivera-Alba M, Branson S, Branson K. JAABA: interactive machine learning for automatic annotation of animal behavior. *Nat. Methods* 10(1), 64–67 (2013).
- Tungtur SK, Nishimune N, Radel J, Nishimune H. Mouse behavior tracker: an economical method for tracking behavior in home cages. *Biotechniques* 63(5), (2017).
- Pottash AE, McKay R, Virgile CR, Ueda H, Bentley WE. TumbleScore: Run and tumble analysis for low frame-rate motility videos. *Biotechniques* 62(1), (2017).
- Hughes AJ, Mornin JD, Biswas SK *et al.* Quanti.us: a tool for rapid, flexible, crowd-based annotation of images. *Nat. Methods* 15(8), 587–590 (2018).
- Carpenter AE, Kamentsky L, Eliceiri KW. A call for bio-imaging software usability. *Nat. Methods* 9(7), 666–70 (2012).
- Broekmans OD, Rodgers JB, Ryu WS, Stephens GJ. Resolving coiled shapes reveals new reorientation behaviors in *C. elegans*. *Elife* 5, 1077–1084 (2016).
- Jackson-Holmes EL, McDevitt TC, Lu H. A microfluidic trap array for longitudinal monitoring and multi-modal phenotypic analysis of individual stem cell aggregates. *Lab Chip* 17(21), 3634–3642 (2017).
- Schneider CA, Rasband WS, Eliceiri KW. NIH Image to ImageJ: 25 years of image analysis. *Nat. Methods* 9(7), 671–675 (2012).
- Rainie L, Zickuhr K. Americans' Views on Mobile Etiquette. www.pewresearch.org.
- Yemini E, Jucikas T, Grundy LJ, Brown AEX, Schafer WR. A database of *C. elegans* behavioral phenotypes. *Nat. Methods* 10(9), 877–879 (2013).
- Fire A, Xu S, Montgomery MK, Kostas SA, Driver SE, Mello CC. Potent and specific genetic interference by double-stranded RNA in *Caenorhabditis elegans*. *Nature* 391(6669), 806–11 (1998).
- Friedland AE, Tzur YB, Esvelt KM, Colaiácovo MP, Church GM, Calarco JA. Heritable genome editing in *C. elegans* via a CRISPR-Cas9 system. *Nat. Methods* 10(8), 741–743 (2013).
- White JG, Southgate E, Thomson JN, Brenner S. The structure of the nervous system of the nematode *Caenorhabditis elegans*. *Philos. Trans. R. Soc. Lond. B. Biol. Sci.* 314(1165), 1–340 (1986).
- Nelson LS, Rosoff ML, Li C. Disruption of a neuropeptide gene, *flp-1*, causes multiple behavioral defects in *Caenorhabditis elegans*. *Science* 281(5383), 1686–90 (1998).
- Mendel J, Korswagen H, Liu K *et al.* Participation of the protein Go in multiple aspects of behavior in *C. elegans*. *Science* 80, 267(5204), 1652–1655 (1995).
- de Bono M, Bargmann CI. Natural variation in a neuropeptide Y receptor homolog modifies social behavior and food response in *C. elegans*. *Cell* 94(5), 679–689 (1998).
- Stephens GJ, Johnson-Kerner B, Bialek W, Ryu WS. Dimensionality and dynamics in the behavior of *C. elegans*. *PLoS Comput. Biol.* 4(4), e1000028 (2008).
- Cooper S, Khatib F, Treuille A *et al.* Predicting protein structures with a multiplayer online game. *Nature* 466(7307), 756–760 (2010).

