Active Graph Matching for Automatic Joint Segmentation and Annotation of *C. elegans*

****** **********[1], ******* ***[1], ******* ******[2], and **** *****[1]

[1] **********
[2] **********

Abstract. In this work we present a novel technique we term *active graph matching*, which integrates the popular active shape model into a sparse graph matching problem. In this way we are able to combine the benefits of a global, statistical deformation model with the benefits of a local deformation model in form of a second-order random field. Despite the optimization being NP-hard we present a new iterative, global energy minimization technique which achieves empirically good results. This enables us to exceed state-of-the art results for the task of matching nuclei in 3D microscopic images of *C. elegans*. Furthermore by adding an additional pre-processing step in the form of the generalized Hough-transform, we are able to jointly segment and annotate a large set of nuclei in a fully automatic fashion for the first time.

1 Introduction and Related Work

A frequently used model organism in developmental biology is the worm *C. elegans*.[1] Since *C. elegans* is highly stereotypical it is well suited for comparative developmental studies. A common and time consuming problem in such studies is the segmentation and annotation (labeling) of cell nuclei with their unique biological names in 3-dimensional microscopy images [3,4,5,6].

In this work we present two automated methods to help solve this problem. The first, semi-automatic method utilizes hand-segmented nuclei and automatically achieves the annotation of each nucleus by matching a statistical atlas to the given segmentation. The second method goes a step further and solves the full problem of joint nuclei segmentation and annotation. We are not the first to work on automatic annotation of hand-segmented nuclei in *C. elegans*: Long et al. [5,6] find a one-to-one mapping of a static atlas based on nucleus position and shape, but their approach is agnostic to covariances between nucleus movements. To this end, we suggest to use a global point distribution model, a.k.a. *active shape model* [7], to capture the global deformations seen in 3D *C. elegans*. However, since such global deformation models cannot deal well with local deformations, we borrow advanced graph matching methods from the computer vision literature [8]. The idea is not only to match individual points but entire neighborhoods between a source and a target dataset. Technically this is achieved by matching nodes and edges in adequately built graph models. Matching costs for edges can thereby express various forms of local deformations and have shown to be practically relevant in many cases, e.g. in [9,10,11]. See [12] and references therein for an introduction to graph matching.

[1] Here we work exclusively with *disentangled, straightened* images of *L1 larvae*. Disentangling and straightening are not topics of this paper. See e.g. [1,2]

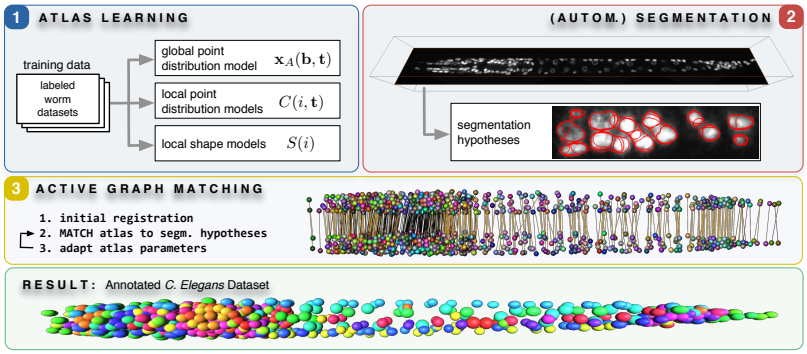

Fig. 1. A sketch of the proposed pipeline. First, a statistical atlas is learned from annotated data. New, straightened images are then segmented automatically. Subsequently, the body axes of atlas and segmentation are aligned. *Active Shape Matching* then alternate between graph matching and optimization of atlas parameters.

Our goal is to find a matching between nuclei in our atlas and a given target dataset which is optimal with respect to both a global and a local deformation model. Our main technical contribution is a strong and global optimization technique for this problem. We show that the gap between the best solution and a lower bound is empirically small. To the best of our knowledge there is only one very recent work [12] which has presented a related idea for combining graph matching with a global deformation model. However, active shape models are not considered, and the graph matching is solved differently, without getting any lower bound. Most importantly, no experimental results show advantages of combining a global and local deformation model.

To summarize, our main contributions are three-fold: (*i*) A new model we term *active graph matching* with an associated global optimization technique. (*ii*) An experimental validation that such a complex model can be optimized successfully for annotating nuclei in *C. elegans* L1 larvae (*iii*) An extension of this procedure to jointly segment and match nuclei in a fully automatic fashion. Results show that our *semi-automatic* method considerably outperforms the state-of-the-art, and our *full pipeline* is, to our knowledge the first fully automatic method ever described for this problem. Finally, a small contribution is the idea of *in-painting* missing information into the training set.

2 Method

The pipeline presented in this paper is sketched in Figure 1. It starts with the *training phase*. Given manually segmented and annotated datasets (cf. Section 2), we build a *statistical atlas* of *C. elegans*. Section 2.1 describes the global and local properties of the worm that we learn from training data. Given the atlas, the *test phase* (red part of Figure 1) runs our automatic segmentation and annotation pipeline on a new image of *C. elegans*. First, a set of segmentation hypotheses is generated (cf. Section 2.2). Based on these hypotheses, the *body axes* of the worm are determined, and aligned with the body axes of the atlas by means of an affine transformation (cf. Section 2.3). This alignment serves as initial transformation for *Active Graph Matching* (cf. Section 2.3). The

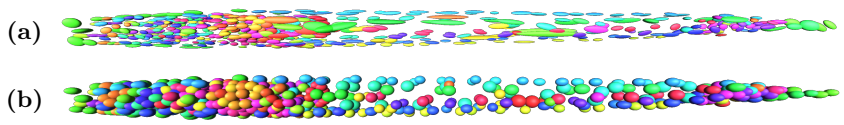

Fig. 2. Two types of local statistical models in our *C. elegans* atlas. **(a)** Local point distribution models per nucleus: Respective covariance matrices $C(i, I)$ represented as ellipses. **(b)** Average shape $s_A(i)$ of each nucleus in the atlas. Not that the local shape models also contain the respective covariances. Figures show the *in-painted* atlas.

best matching hypotheses are determined by minimizing an objective function that encodes the problem of matching our statistical atlas to the segmentation hypotheses. We optimize the objective by a novel, iterative closest points (ICP)-like method that alternates between graph matching and adaptation of global transformation and deformation parameters. The matching thus selects a subset of segmentation hypotheses, while simultaneously annotating them with their biological names.

Datasets. We work on a set of *training worms* that was also used in [6][1]. One training worm contains manual segmentations of 357 nuclei that can be distinguished by eye in one individual *C. elegans* L1 larva in confocal microscopy image data. All of these nuclei were manually annotated with their biological names. We refer the reader to [6] for more details on the dataset. We denote the number of training worms as n_W, and the number of annotated nuclei as n_A. Per worm, index i, with $1 \leq i \leq n_A$, represents the i-th of all annotated nuclei. Nuclei are sorted consistently by their biological names in all training worms.

2.1 Atlas

Our statistical atlas of *C. elegans* consists of (i) a global point distribution model [7] that captures modes of variation of the *point clouds* given by the set of nuclei center points of each worm (cf. supplementary movie), (ii) local point distribution models describing the variability of individual nucleus locations (cf. Figure 2a), (iii) local shape models capturing the variablity of the shape of each nucleus (cf. Figure 2b), and (iv) local offset distribution models describing the variablilty of differences of any two individual nucleus locations.

Global point distribution model. From the set of training worms we extract locations of nuclei center points. We denote the center point location of nucleus $i \leq n_A$ in training worm $w \leq n_W$ as $x(i, w) \in \mathbf{R}^3$, the concatenation of all locations of training worm w as $\mathbf{x}_w := (\ldots, x(i, w)^T, \ldots)^T \in \mathbf{R}^{3n_A}$, and the matrix assembled from all training vectors as $\mathcal{X} := (\ldots, \mathbf{x}_w, \ldots)$. From the set of corresponding training vectors, $\{\mathbf{x}_w \in \mathbf{R}^{3n_A} | 1 \leq w \leq n_W\}$, we build a point distribution model [7] of nuclei locations. Therefore we align all training vectors via Procrustes analysis, and then perform principle component analysis, yielding the eigenvectors $\mathbf{p}_k \in \mathbf{R}^{3n_A}$ of the covariance matrix $(1/(n_W - 1))(\mathcal{X} - \bar{\mathcal{X}})(\mathcal{X} - \bar{\mathcal{X}})^T$, where $\bar{\mathcal{X}}$ denotes a matrix with the average nuclei location vector $\bar{\mathbf{x}}_A := (1/n_W) \sum_w \mathbf{x}_w$ in every column. We denote the matrix

[1] We thank Hanchuan Peng, Fuhui Long, Xiao Liu and Stuart Kim for providing the set of images and training data of C. Elegans L1 larvae from their 2009 work [6].

assembled from the eigenvectors as $P := (\ldots, \mathbf{p}_w, \ldots) \in \mathbf{R}^{3n_A \times 3\min(n_W, n_A)-1}$. The point distribution model can then be formulated as $\mathbf{x}_A(\mathbf{b}, \mathbf{t}) := \mathbf{t}(\bar{\mathbf{x}}_A + P \cdot \mathbf{b})$ with $\mathbf{b}$ denoting a vector of global *shape parameters*, and $\mathbf{t} : \mathbf{R}^3 \to \mathbf{R}^3$ an affine transformation. $\mathbf{t}$ consists of a rotation, scale and shear matrix $R \in \mathbf{R}^{3\times3}$, and an offset vector $\mathbf{o}$. Per nucleus i, the model reads $x_A(i, \mathbf{b}, \mathbf{t}) := R(\bar{x}_A(i) + P(i) \cdot \mathbf{b}) + \mathbf{o}$, with $\bar{x}_A(i)$ denoting the average location of nucleus i, and $P(i)$ denoting rows $3i-2$, $3i-1$ and $3i$ of the eigenvector matrix P. Each training vector can be represented by some global shape parameter vector: $\exists \mathbf{b}_w : \bar{\mathbf{x}}_A + P \cdot \mathbf{b}_w = \mathbf{x}_w$. In our global shape model we confine the shape parameters to the min/max respective values that appear in training data.

Local models. Let $\mathcal{X}(i)$ denote the matrix that contains all training locations of nucleus i, namely $x(i, w)$ for all $w \leq n_W$, as its columns. Let $\bar{\mathcal{X}}(i)$ denote the matrix that repeatedly contains the average nucleus location $\bar{x}_A(i)$ in n_W columns. Our local point distribution model is then encoded in the covariance matrix per nucleus location: $C(i, R) = (1/n_W)R(\mathcal{X}(i) - \bar{\mathcal{X}}(i))(\mathcal{X}(i) - \bar{\mathcal{X}}(i))^T R^T \in \mathbf{R}^{3\times3}$. This matrix allows us to measure the distance of some point $x \in \mathbf{R}$ to location i in an instance of the atlas: $\text{locDiff}(x, i, \mathbf{b}, \mathbf{t}) := (x - x_A(i, \mathbf{b}, \mathbf{t}))^T \cdot C(i, R)^{-1} \cdot (x - x_A(i, \mathbf{b}, \mathbf{t}))$.

We describe the shape of an individual nucleus by means of the radii of an ellipsoid fit to the nucleus volume, sorted by value. We denote the shape of nucleus i in training worm w as $s(i, w) \in \mathbf{R}^3$. From the training data, we derive the average shape per nucleus, $s_A(i)$, as well as the respective covariance matrix $S(i)$. Thus we can measure the distance of some shape s to the shape of atlas nucleus i as $\text{shapeDiff}(s, i) := (s - s_A(i))^T \cdot S(i)^{-1} \cdot (s - s_A(i))$.

In addition to nucleus-individual statistics, we also perform statistics on offset vectors between any two nuclei: Let $d(i, j, w) := x(i, w) - x(j, w)$ denote a training offset vector. We retrieve the average offset $\bar{d}_A(i, j) := (1/n_W) \sum_w d(i, j, w)$ as well as the respective covariance matrix $D(i, j, R)$. Let $d_A(i, j, \mathbf{b}, R)$ denote an offset vector in an instance of the global point distribution model. Then, we can measure the distance of some offset d w.r.t. nuclei i and j in the atlas: $\text{offsetDiff}(d, i, j, \mathbf{b}, R) := (d - d_A(i, j, \mathbf{b}, R))^T \cdot D(i, j, R)^{-1} \cdot (d - d_A(i, j, \mathbf{b}, R))$. The covariance of offsets furthermore enables us to measure how tightly two nuclei locations correlate: We use the determinant of the covariance matrix $D(i, j, R)$ as a respective "neighborhood"-measure. Thus we can define a k-neighborhood on the atlas, denoted as $\mathcal{N}_k \subset \{(i, j) | 1 \leq i, j \leq n_A\}$.

Inpainting. Nuclei that are "missing" in our 357-nuclei-atlas, mainly in the brain region, pose a severe challenge to the annotation problem: The atlas region posterior to the brain can freely match to target nuclei within the brain, taking the whole posterior body part with them. Therefore we *inpaint* the missing 201 nuclei into the training worms by taking one complete manual segmentation as reference and warping it to all the other training point clouds by means of Thin Plate Spline Warps. This yields synthetically completed point clouds for all training worms. We inpaint the missing nuclei shapes by assigning the shape of the closest not-annotated nucleus in the respective manual segmentation.

2.2 Segmentation Hypotheses

Nuclei segmentation is based on the Generalized Hough Transform (GHT) [13]. We use a triangular surface mesh of an ellipsoid as a template. We run GHT

multiple times with a range of differently scaled and oriented templates. Each run returns a score $\in \{0,1\}$ at each image voxel that measures how well the template fits the image gradient. We select the highest scoring n positions for each scale/orientation of the template. A segmentation hypothesis is simply the template put at the respective position, scale and orientation. To avoid duplicates we omit positions at which the template overlaps with higher-scoring positions already selected for this scale/orientation. This procedure results in an over-segmentation of the image, i.e. more hypotheses than nuclei. Hypotheses from different GHT runs are, in general, nested. This way we reduce the risk of not detecting nuclei. Figure 1 shows exemplary segmentation hypotheses.

2.3 Active Graph Matching

Objective. Let n_T denote the number of nuclei segmentation hypotheses. An *assignment* $a_{i,j} \in \{0,1\}$ encodes whether atlas index $i \leq n_A$ is assigned to target index $j \leq n_T$. We denote the matrix of assignments as $\mathcal{A} := (a_{i,j})_{i=1,j=1}^{n_A,n_T}$. A bipartite matching is a matrix $\mathcal{A}$ which satisfies $\forall i \leq n_A : \sum_{j=1}^{n_T} a_{i,j} \leq 1$ and $\forall j \leq n_T : \sum_{i=1}^{n_A} a_{i,j} \leq 1$. I.e., an atlas nucleus can be matched to at most one target nucleus, and vice-versa. We define the energy of matching the atlas to the target with affine transformation $\mathbf{t}$, shape parameters $\mathbf{b}$, and matching $\mathcal{A}$ as

$$E(\mathcal{A},\mathbf{b},\mathbf{t}) := \sum_{i\leq n_A, j\leq n_T} \phi(i,k,\mathbf{b},\mathbf{t}) \cdot a_{i,k} + \sum_{(i,j)\in\mathcal{N}, k,l\leq n_T} \psi(i,j,k,l,\mathbf{b},\mathbf{t}) \cdot a_{i,k} \cdot a_{j,l} \tag{1}$$

where $\mathcal{N}$ is the neighborhood relation we defined on the atlas, cf. Section 2.1. *Unary potentials* $\phi(i,k,\mathbf{b},\mathbf{t})$ encode the cost per assignment $a_{i,k}$. We define

$$\phi(i,k,\mathbf{b},\mathbf{t}) := \lambda_1 \cdot \text{locDiff}(x_T(k), i, \mathbf{b}, \mathbf{t}) + \lambda_2 \cdot \text{shapeDiff}(s_T(k), i) + \lambda_3 \cdot \text{cost}(k) + \lambda_4 \cdot \text{c} \tag{2}$$

where $x_T(k) \in \mathbf{R}^3$ is the center point location of the k-th hypothesis, $k \leq n_T$, $s_T(k) \in \mathbf{R}^3$ is the target shape descriptor, $\text{cost}(k)$ is inversely proportional to the GHT score and hence encodes how well the image data supports the k-th hypothesis, and c is a negative constant that serves as an incentive to make matches. Terms get relative weights by positive constants λ. *Binary potentials* $\psi(i,j,k,l,\mathbf{b},\mathbf{t})$ encode the cost per pair of assignments, $a_{i,k}, a_{j,l}$. We define

$$\psi(i,j,k,l,\mathbf{b},\mathbf{t}) := \lambda_5 \cdot \text{offsetDiff}(d_T(k,l), i, j, \mathbf{b}, \mathbf{t}) \tag{3}$$

where $d_T(k,l)$ denotes the offset between target nuclei k,l, namely $x_T(k) - x_T(l)$.

Optimization. *Initial Atlas Parameters:* The global shape parameters $\mathbf{b}$ are initialized to zero. We automatically determine an initial affine transformation of the atlas: The first eigenvector of the point cloud given by all segmentation hypotheses is aligned with the anterior-postierior (AP) axis of the atlas such that the centers of gravity line up. To determine the correct rotation around the AP axis we exploit the fact that nuclei are distributed asymmetrically along the dorso-ventral axis, while symmetrically along the left-right axis.

Optimal Matching: For fixed $\mathbf{b},\mathbf{t}$ we minimize (1) w.r.t. the matching $\mathcal{A}$ with the Dual-Decomposition-based method of Torresani et al. [8]. In practice, considering all entries of $\mathcal{A}$ is infeasible. Hence we only consider assignments $a_{i,k}$ for which $\text{locDiff}(x_T(k), i, \mathbf{b}, \mathbf{t})$ falls below a fixed threshold.

	ActiveGM	**ActiveIGM**	ActiveHungarian	Long et al.
Synthetic	95/94(7)	-	93/88(12)	-
SemiAuto	92/90(8)	**93/92**(7)	79/77(9)	*/86(*)
Automatic	86/82(12)	**86/83**(11)	62/60(12)	**

Table 1. Evaluation of annotation accuracy on 30 worms. Measures: median/mean(std), all in %. See text for description of scenarios (rows) and algorithms (columns). *Results presented as plot in [6], but numbers not given. **Results presented as plot in [6], but error measure not described and numbers not given.

Optimization of Atlas Parameters: For a fixed matching, the objective is the sum of squared residuals of an overdetermined system of equations which is linear in all atlas parameters, as described in the following. The optimal parameters can then be determined by the standard least squares method. In the objective, only the terms locDiff and offsetDiff depend on atlas parameters. For locDiff each matched nucleus i entails three equations, namely

$$S \cdot R^{-1} x_T(k) - S \cdot \bar{x}_A(i) - S \cdot P(i)\mathbf{b} + S \cdot \mathbf{o} = (0,0,0)^T \tag{4}$$

where S satisfies $S^T \cdot S = C(i,I)^{-1}$. Such an S exists in case $C(i,I)$ is symmetric and positive definite, which is the case in our practical setting. The equations are linear in the entries of R^{-1}, $\mathbf{b}$, and $\mathbf{o}$, respectively. For offsetDiff, each pair of matched neighbors i, j entails the following three equations:

$$G \cdot R^{-1}(x_T(k) - x_T(l)) - G \cdot (\bar{x}_A(i) - \bar{x}_A(j)) - G \cdot (P(i) - P(j))(b) = (0,0,0)^T \tag{5}$$

where G satisfies $G^T \cdot G = D(i)^{-1}$. Analogous to S, such a G exists in our practical setting. Overall we have, in practice, far more equations than parameters. Hence we can solve for R, $\mathbf{o}$ and $\mathbf{b}$ with the method of least squares.

3 Results and Discussion

We run our method in a leave-one-out fashion on the 30 datasets used for atlas training (cf. Section 2). We consider three different scenarios. (1, *Synthetic*): An idealized scenario, where we match the 357-nuclei atlas to the corresponding 357 target nucleus segmentations. (Therefore the 357 corresponding target nuclei have to be selected by hand from the manual segmentation.) (2, *SemiAuto*): We match the atlas to manual segmentations of all nuclei present in respective target datasets. (3, *Automatic*): We run fully automatic, simultaneous segmentation and annotation as described in Section 2. In each of the scenarios, we run our algorithm in three different ways: (1, *ActiveGM*) with the 357-nuclei atlas, (2, *ActiveIGM*) with the inpainted 558-nuclei atlas, and (3, *ActiveHungarian*) without binary potentials. We run ActiveIGM only for the real-world scenarios. We run ActiveHungarian without in-painting in the synthetic scenario, and with in-painting in the real-world scenarios.

Parameters of our method are the scalar weights $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ in (1), the number m of global deformation parameters in $\mathbf{b}$ to consider, and the number of iterations to perform. For all scenarios and algorithms we set: $\lambda_2 := 1$, $c := -150$, $m := 2$, and perform three iterations where only R and $\mathbf{t}$ are optimized, followed

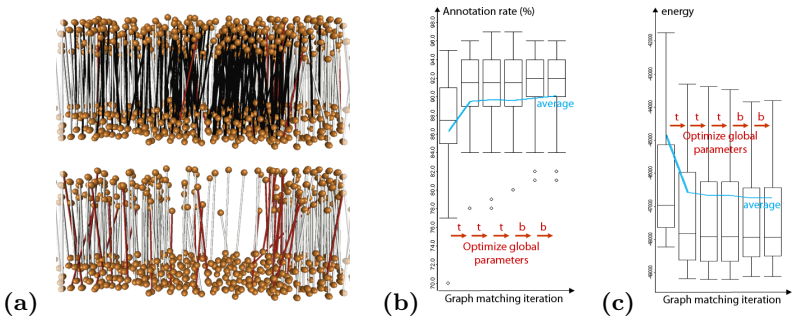

Fig. 3. (a) Close-up to matching results in the head of an exemplary worm. Top: inpainted atlas; bottom: partial atlas. Inpainting leads to better matching performance. White lines: correct annotation; black lines: inpainted nuclei, no ground truth available; red lines: annotation errors (fewer on top). **(b)** Evolution of annotation accuracy for semi-automatic matching scenario. X-axis: matching iteration. Y-axis: fraction of correctly annotated nuclei. **(c)** Evolution of the respective matching energy.

by three iterations where we optimize only **b**. As for the weight of locDiff, we set $\lambda_1 := 0$ for ActiveGM, and $\lambda_1 := 1$ for ActiveHungarian. As for the weight of offsetDiff, we set $\lambda_4 := 1$ for ActiveGM, and $\lambda_4 := 0$ for ActiveHungarian (since Hungarian matching cannot take into account binary potentials). As target cost weight we set $\lambda_3 := 0$ in all but the fully automatic scenario. Only there we have a confidence value for each segmentation hypothesis and set $\lambda_3 := 10$.

Table 3 lists the resulting annotation accuracy in percent for all the scenarios and algorithms described above. For reference we also include the result of Long et al. [6]. For the scenarios working on manual segmentations, we count the fraction of nuclei that are correctly annotated. For the fully automatic scenario, we count the fraction of matched segmentation hypotheses whose center points lie within the respective ground truth nucleus, or are at most one average nucleus radius apart from the respective ground truth center point.

Neglecting location differences in the 2nd order energy, i.e. $l_1 := 0$ for ActiveGM, yields considerably better annotation accuracy. We argue that this is due to the respective much more flexible local deformation model. Note that locDiff $= 0$ means that the objective is invariant w.r.t. **o**. However, in practice we still need **o** for selecting the assignments we consider in the matching problem (cf. Section 2.3), hence we always derive it via locDiff.

Discussion. For the task of annotating manual segmentations of nuclei, we achieve an average annotation rate of 92% with ActiveIGM, thereby considerably outperforming the result of Long et al. who report an average of 86%. Note that Long et al. make use of an additional image channel that our method does not need. For the task of fully automatic segmentation and annotation, we achieve a median/average annotation rate of 86/83%, which approaches the rate that Long et al. achieve in the much simpler partly manual scenario.

Employing 2nd order graph matching instead of just the Hungarian algorithm makes a huge difference: ActiveHungarian works relatively well *only* in the synthetic scenario, while the inferiority as compared to ActiveGM increases as the matching problem gets more sophisticated: In the order of complexity of

the matching problem (top to bottom in Table 3), ActiveGM is on average 6%, 15%, and 23% better than ActiveHungarian, respectively.

The benefit of using the in-painted atlas instead of the 357-nuclei-atlas is shown in Figure 3(a). The impact of running matching and parameter optimization iteratively is shown in Figure 3(b,c) in terms of the evolution of the annotation rate and the respective value of the objective. While the matching problem is solved approximately up to a duality gap of about $2c$ in the fully automtatic scenario, lower bounds are tight in the synthetic and the semi-automatic scenario, i.e. here we find the globally optimal matching.

Conclusion. We have presented *active graph matching*, a method that combines active shape models with graph matching in one objective and provides an approach for global optimization. With this method we do not only outperform the current state of the art in annotating manual segmentations of nuclei in *C. elegans* L1 larvae, but furthermore define the state of the art in solving both segmentation and annotation simultaneously in a fully automatic fashion. We hypothesize that our method will be highly beneficial for the equally relevant task of nuclei annotation in later stages of *C. elegans* development, where nuclei are more numerous and more densely packed, and hence methods that do not consider covariances of nuclei locations and offsets are bound to fail.

References

1. TR Raviv, V Ljosa, AL Conery, FM Ausubel, AE Carpenter, P Golland, and C Wählby. Morphology-Guided Graph Search for Untangling Objects: C. elegans Analysis. In *MICCAI 2010*, pages 634–641, 2010.
2. H Peng, F Long, X Liu, S K Kim, and E Myers. Straightening Caenorhabditis Elegans Images. *Bioinformatics*, 24(2):234–242, January 2008.
3. J E Sulston and H R Horvitz. Post-embryonic Cell Lineages of the Nematode, Caenorhabditis Elegans. *Developmental biology*, 56(1):110–156, March 1977.
4. M Sarov et al. A Genome-Scale Resource for In Vivo Tag-Based Protein Function Exploration in C. Elegans. *Cell*, 150(4):855–866, August 2012.
5. SJ Aerni, X Liu, CB Do, SS Gross, A Nguyen, SD Guo, F Long, H Peng, SS Kim, and S Batzoglou. Automated Cellular Annotation for High-resolution Images of Adult C. Elegans. *Bioinformatics*, 29(13):i18–26, July 2013.
6. F Long, H Peng, X Liu, S K Kim, and E Myers. A 3D Digital Atlas of C. Elegans and its Application to Single-Cell Analyses. *Nature Methods*, 6:667672, 2009.
7. T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active Shape Models - Their Training and Application. 61(1):38–59, 1995.
8. L. Torresani, V. Kolmogorov, and C. Rother. A dual decomposition approach to feature correspondence. *IEEE Transactions*, 35(2):259–271, Feb 2013.
9. O Duchenne, A Joulin, and J Ponce. A graph-matching kernel for object categorization. In *ICCV 2011*, pages 1792–1799, 2011.
10. J Hays, M Leordeanu, A A Efros, and Y Liu. Discovering Texture Regularity as a Higher-Order Correspondence Problem. *ECCV 2006*, 3952:522–535, 2006.
11. H Jiang, S X Yu, and D R Martin. Linear Scale and Rotation Invariant Matching. *IEEE Transactions*, November 2010.
12. Feng Zhou and F. De La Torre. Deformable graph matching. In *CVPR 2013*, pages 2922–2929, June 2013.
13. K. Khoshelham. Extending Generalized Hough Transform to Detect 3D Objects in Laser Range Data. In *ISPRS Workshop Laser Scanning*, page 206, 2007.