Journal of
Electronic Imaging

JElectronicImaging.org

Region growing using superpixels with learned shape prior

Jiří Borovec
Jan Kybic
Akihiro Sugimoto

Jiří Borovec, Jan Kybic, Akihiro Sugimoto, "Region growing using superpixels with learned shape prior," *J. Electron. Imaging* **26**(6), 061611 (2017), doi: 10.1117/1.JEI.26.6.061611.

Region growing using superpixels with learned shape prior

Jiří Borovec,[a,*] Jan Kybic,[a] and Akihiro Sugimoto[b]
[a]Czech Technical University in Prague, Faculty of Electrical Engineering, Department of Cybernetics, Biomedical Imaging Algorithms Group, Prague, Czech Republic
[b]National Institute of Informatics, Tokyo, Japan

Abstract. Region growing is a classical image segmentation method based on hierarchical region aggregation using local similarity rules. Our proposed method differs from classical region growing in three important aspects. First, it works on the level of superpixels instead of pixels, which leads to a substantial speed-up. Second, our method uses learned statistical shape properties that encourage plausible shapes. In particular, we use ray features to describe the object boundary. Third, our method can segment multiple objects and ensure that the segmentations do not overlap. The problem is represented as an energy minimization and is solved either greedily or iteratively using graph cuts. We demonstrate the performance of the proposed method and compare it with alternative approaches on the task of segmenting individual eggs in microscopy images of Drosophila ovaries. *© 2017 SPIE and IS&T* [DOI: 10.1117/1.JEI.26.6.061611]

Keywords: segmentation; region growing; superpixels; shape prior; ray features; graph cuts; Drosophila ovary.

Paper 170330SS received Apr. 22, 2017; accepted for publication Oct. 20, 2017; published online Nov. 16, 2017.

1 Introduction

Image segmentation is one of the fundamental image analysis tasks,[1–4] consisting of dividing an image into multiple regions (or classes). Our target application is segmenting individual eggs in microscopy images of Drosophila ovaries [see Figs 1(a) and 1(b)], which is one of the key steps in the image processing pipeline to automatically and robustly analyze many thousands of images needed to study the gene expressions governing Drosophila oogenesis.[5,6] This problem has several challenging aspects. First, the objects (eggs) are highly structured and cannot be easily distinguished by texture or intensity. Second, there are several eggs in the image, often touching, with unclear boundaries between them. Third, the algorithm should be fast, as there is a high number of images to be processed. Note also that identifying individual eggs with mutually similar appearance is more challenging than a standard binary or multiclass segmentation[7,8] [see Fig. 1(b)]. The standard approach is to postprocess the foreground/background segmentation using mathematical morphology and connected component analysis but in this case it turned out not to be sufficiently robust as you can see in the final experiments.

1.1 *Proposed Method*

The proposed method combines three existing techniques—region growing, superpixels, and shape modeling. Region growing[1,9–11] is one of the classical image segmentation approaches, which starts from "seeds," often individual pixels, and repeatedly joins them with their neighbors according to rules designed to encourage the homogeneity of the regions. It is simple to implement and has been used successfully in many applications. The novelty of our approach is threefold: first, we grow the regions based on superpixels instead of pixels, where superpixels are small compact homogeneous groups of pixels, which can be calculated quickly and just once [see Fig. 2(b)]. This improves the segmentation speed by several orders of magnitude while the superpixels preserve the object boundaries. We are using the simple linear iterative clustering (SLIC) superpixels,[12] which we found to be a good trade-off between speed and quality, but other superpixel types can also be used.[13] Note that region growing by superpixels, i.e., representing regions using superpixels, is very different from calculating superpixels by region growing.[14] Another particularity of our approach is the image features used. It would be possible to use classical texture or color features directly. However, for robustness and speed we first use these features to assign each superpixel to one of four biologically meaningful classes[7] and use this preliminary segmentation [see Fig. 1(b)] as input [see Fig. 2(a)] for the region growing—the annotation of individual eggs is shown in Fig. 1(c). In this way, the region growing can correct imperfections of the preliminary segmentation as we will present later in Sec. 3. Second, we incorporate a shape model based on the so-called ray features,[15,16] to guide the region growing toward plausible shapes, using our *a priori* knowledge. (We see that in our application, the eggs are approximately oval.) The ray features are transformed to provide rotation invariance.[8] Scale invariance could be also easily achieved but in our case we do not require it, since object size is an important attribute. Multiple alternative models can be used in parallel. Previously, shape models for region growing were described for example by a distance map[17,18] or moments.[19] We build a probability model over the shape features using histogramming, other options include PCA, or manifold learning.[20–22] Third, our method segments several objects simultaneously, ensuring that they do not overlap. One iteration of the

*Address all correspondence to: Jirí Borovec, E-mail: jiri.borovec@fel.cvut.cz

1017-9909/2017/$25.00 © 2017 SPIE and IS&T

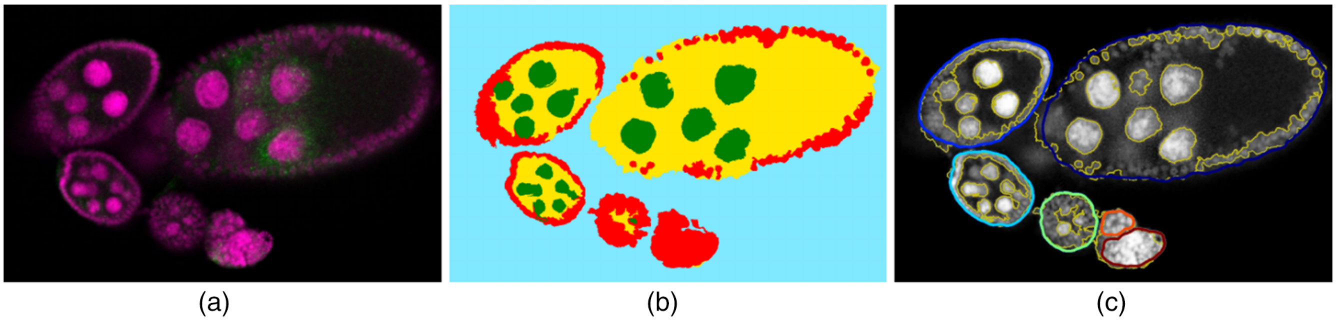


Fig. 1 (a) Fluorescence microscopy image of a Drosophila ovary with cell anatomy in magenta and gene expression in green, (b) preliminary texture-based four-class superpixel-level segmentation, (c) the boundary of the initial four-class segmentation (thin yellow contour) with individual eggs marked manually (wide color lines), superimposed over the cell anatomy channel in gray.

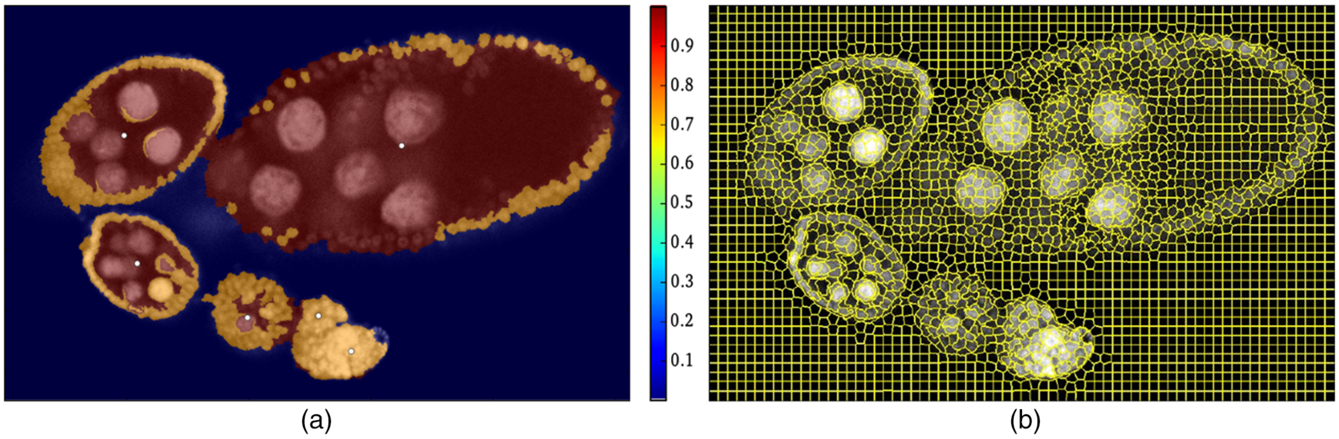


Fig. 2 Sample Drosophila ovary image with multiple eggs. (a) Probability map obtained from the preliminary segmentation shown in Fig. 1 representing the likelihood for each superpixels being an egg and approximate egg centres marked by the white dot; (b) SLIC superpixels.

growing process is formulated as an energy minimization problem with a Markov random field (MRF) regularization and solved either greedily or using graph cuts. Since the number of boundary superpixels in a given iteration is small, the procedure is very fast. In contrast, applying graph cuts to all superpixels[8,23] is much more time and resource demanding.

1.2 *Other Related Methods*

Ye et al.[24] used mean-shift superpixels and graph cuts, while the MRF optimization can also be solved by Gibbs sampling or simulated annealing.[25,26] Unlike the present work, neither of these methods can handle multiple interacting objects and incorporate shape models. As one major alternative, graph cuts can be combined at the pixel-level with a shape model such as the layered pictorial structures,[27] the distance functions to a shape template,[28,29] or the star-shaped model;[30] it is also possible to choose among multiple models.[31] These methods alternate between estimating the pose parameters and refining the segmentations and can converge to a suboptimal solution if the pose estimation is incorrect. The number of pose hypotheses that can be simultaneously considered is limited for computational reasons. Global optimization with respect to the shape model parameters is possible but very computationally expensive.[32] Graph cuts also can be augmented by constraints on class distances,[33] one region being inside another.[34,35] All these methods are slower than applying graph cuts on superpixels. Region growing is similar to active contours,[1,36] which can be interpreted as region boundaries, use region-based criteria,[37] and are also often used in biomedical imaging, for example, for cell segmentation and tracking.[38] Active contours can be used to segment multiple objects using, e.g., multiphase level sets[39] or multiobject active contours.[40] Objects may be allowed to overlap or separation between objects can be enforced.[40,41] Shape priors can be integrated using the usual alternative optimization of pose and segmentation;[42–45] specialized methods exist for simple shapes such as circles.[46,47] Active contours can provide subpixel accuracy but their computational complexity is often very high, although fast discrete methods exist.[48,49]

Finally, there seems to be great promise in deep learning methods using convolutional neural networks, such as U-net[50] requiring a small number of training examples but it assumes reasonable homogeneous objects and produces only binary segmentation where individual object separation relies on correct boundaries prediction. Instance segmentation method[51] solves the task at hand but it requires a large amount of training data with detailed pixel-level annotation, which is usually expensive to obtain in biomedical imaging due to high time demands to medical experts, and also it is currently not available for our application.

1.3 *Structure of This Paper*

The rest of this paper is structured as follows: the method is described in Sec. 2, including superpixels (Sec. 2.1), shape models (Sec. 2.3), and optimization (Sec. 2.6). We continue by experiments in Sec. 3 and conclude in Sec. 4.

2 Methods

Given an input image containing multiple nonoverlapping but possibly touching objects, a seed point for each object, and a shape and appearance model, we shall segment these objects as follows: we group pixels into superpixels S (Sec. 2.1) and for each of them calculate the appearance-based object probability. The regions corresponding to objects are then grown (Sec. 2.6) using the appearance (Sec. 2.2) and shape (Sec. 2.3) models. The final segmentation is represented by a function $g: S \rightarrow \{0, 1, \dots, K\}$, which assigns each superpixel $s \in S$ to one of the objects [if $g(s) = 0$] or to the background [if $g(s) = 0$].

2.1 *Superpixel Clustering*

We use the SLIC[12] algorithm to calculate a set of superpixels S that are compact both in space and color. The SLIC algorithm is an adaptation of a widely used k-means clustering algorithm. It uses a combined color and spatial distance $D = d_c + \frac{\xi}{\eta^2} \cdot d_s$, where d_c is a Euclidean distance in the CIELAB color space, and d_s is a Euclidean spatial distance measured in pixels. The superpixel centers are initially placed on a grid with spacing η, which determines the number of superpixels and their size. The user-provided weight ξ controls the trade-off between spatial compactness and color homogeneity. We use instead a regularization parameter $\nu \in (0,1)$, with $\xi^2 = \eta^3 \nu^2$, which we found easier to choose.[52] Figure 3 shows the impact of the SLIC parameters. We have also tested the parameter-free adaptive version SLICO,[12] but it performed worse on our data.

2.2 *Appearance Model*

For each superpixel $s \in S$, we calculate a descriptor y_s that represents the appearance of s through its texture or color properties. Given y_s, we use the appearance model to calculate the probability $P_y(y_s)$ that a superpixel s belongs to an object. For notational convenience, we shall write

$$P_y(g(s)|y_s) = \begin{cases} P_y(y_s) & \text{for } g(s) \neq 0 \\ 1 - P_y(y_s) & \text{for } g(s) = 0 \end{cases}.$$

For our application, we take advantage of the fact that we already have a good preliminary segmentation method that can assign superpixels into four biologically meaningful classes (cytoplasm, follicle cells, nurse cells, and background) based on texture and color features, and a random forest classifier with graph cuts regularization.[7,8] Our descriptor y_s is therefore simply an integer $\{1, \dots, 4\}$, representing one of the four classes. The probability $P_y(y_s)$ of a superpixel belonging to an egg given the preliminary segmentation can be estimated from labeled training data. See Fig. 1(b) for an example of the preliminary segmentation and Fig. 2(a) for an example of the probability map P_y.

2.3 *Shape Model*

The purpose of the shape model is to determine the likelihood of a particular shape being the desired object (in our case, an egg). Given a region (the reference segmentation during model-learning or an intermediate step of the region growing during model fitting), we calculate its center of gravity $\mathbf{c}$ and the so-called ray features[15,53] $\mathbf{r}'$, the distances from $\mathbf{c}$ to the region boundary in a set of N predefined directions (see Fig. 4). To ensure rotation invariance, the distance vector $\mathbf{r}' = \{r_0, \dots, r_{N-1}\}$ is circularly shifted to obtain a rotational normalized vector $\mathbf{r}(i) = \mathbf{r}'((i - \Theta) \bmod N)$ such that it starts with the maximum element, $\mathbf{r}(0) = \max_i \mathbf{r}'(i)$. As an example, the ray feature vectors $\mathbf{r}$ with $N = 36$ [see Fig. 5(a)] and the whisker plots for each $\mathbf{r}(i)$

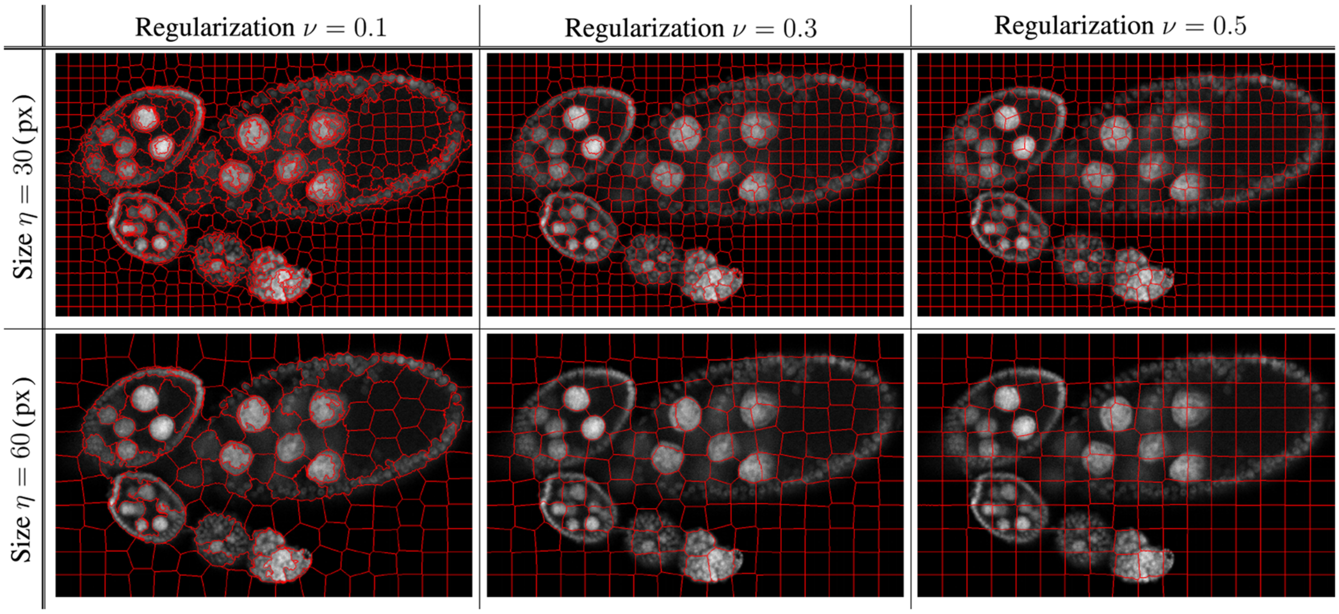

Regularization $\nu = 0.1$
Regularization $\nu = 0.3$
Regularization $\nu = 0.5$
Size $\eta = 30$ (px)
Size $\eta = 60$ (px)

Fig. 3 The influence of the SLIC parameters: superpixel size η and regularization ν.

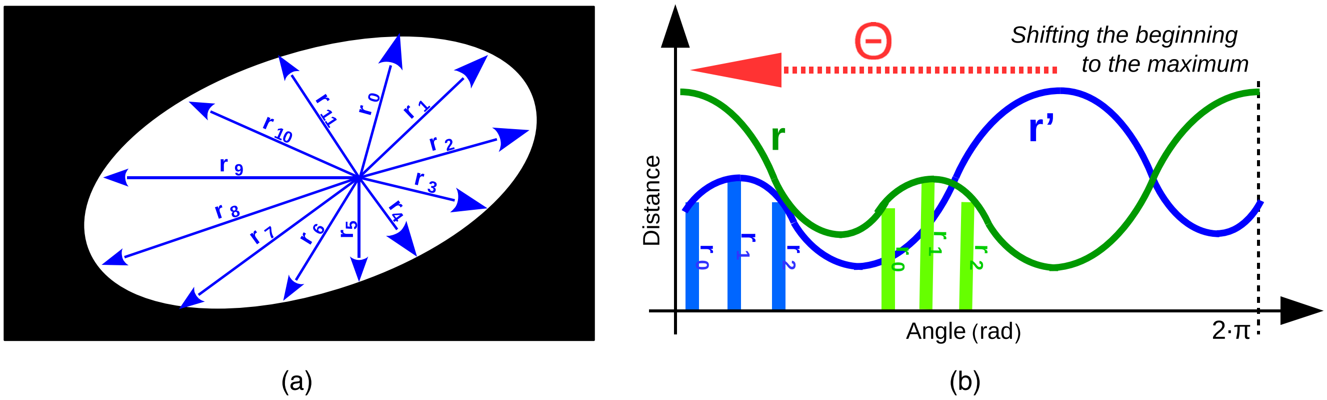


Fig. 4 (a) A shape is described by ray features, distances from the center to the boundaries in predefined directions. (b) The original and shifted distance vectors, $\mathbf{r}'$ (in blue) and $\mathbf{r}$ (in green), respectively.

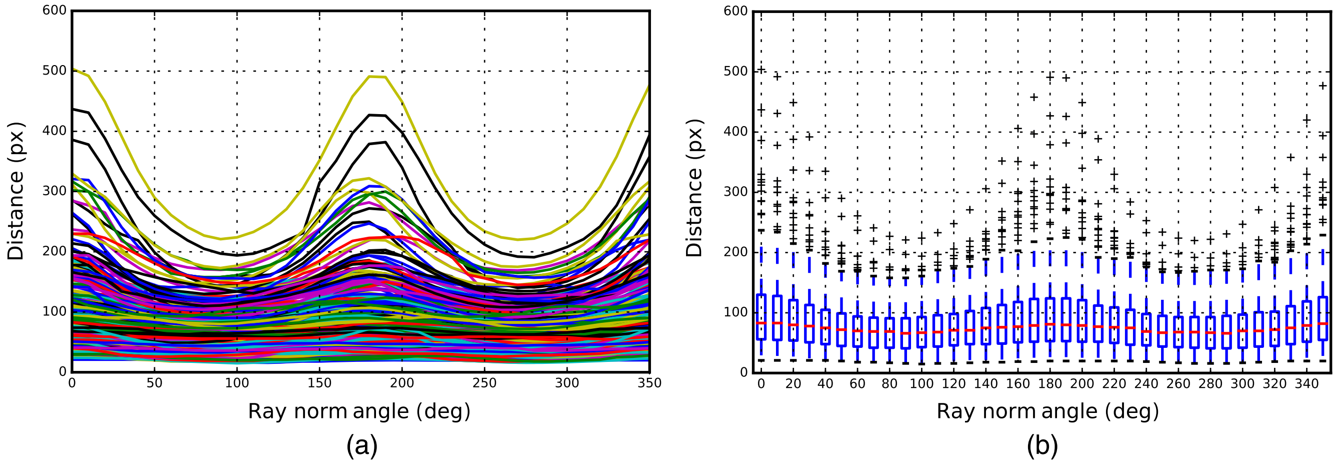


Fig. 5 (a) Visualization of 250 egg shapes represented by the distance vectors $\mathbf{r}$ and (b) their element-wise box and whisker plots for each $\mathbf{r}(i)$ with an angular step of 10 deg.

independently are shown in Fig. 5(b) for a set of 250 Drosophila eggs. Invariance to scale can be achieved by another normalization but it is not suitable for our application.

We have chosen to describe the probability density of the ray distance vector $\mathbf{r}$ by a simple Gaussian mixture model (GMM) with M components over all vectors $\mathbf{r}$ assuming diagonal coverings matrix for each Gaussian

$$p_r(\mathbf{r}) = \rho(\mathbf{r}) = \sum_{j=1}^{M} w_j f_j(\mathbf{r}), \tag{1}$$

with $f_j(\mathbf{r}(i)) = \frac{1}{\sigma_{i,j}\sqrt{2\pi}} \exp\left(-\frac{(\mathbf{r}(i)-\mu_{i,j})^2}{2\sigma_{i,j}^2}\right)$ and $\sum_j^M w_j = 1$.

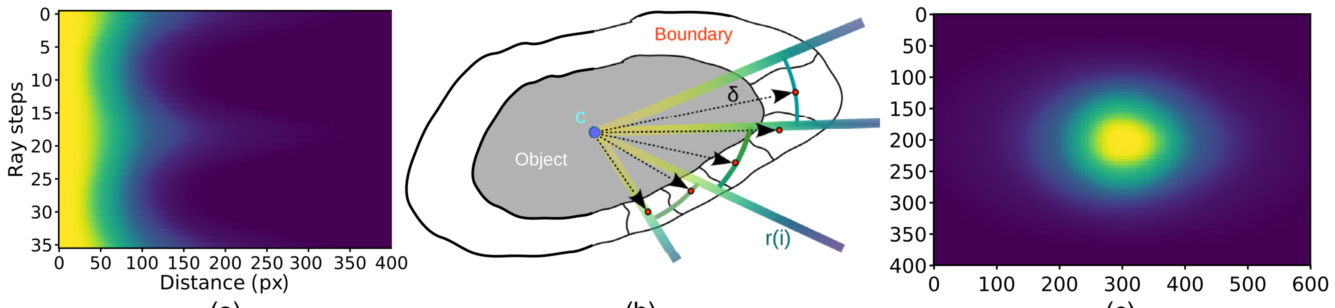
The GMM components may represent different egg development stages or significant shape variations. There are $2NM$ model parameters $(\mu_{i,j}, \sigma_{i,j})$ to be estimated from the training data with the expectation-maximization algorithm,[54] while the M weights $\mathbf{w}$ are estimated for each object independently.

2.4 *Shape Prior*

During region growing, we need to calculate the shape prior $P_m(g(s) = k|\mathbf{m}_k) = q(s, \mathbf{m}_k)$ that a given superpixel $s \in S$ belongs to an object k, where $\mathbf{m}_k = [\mathbf{c}, \mathbf{r}, \Theta, \mathbf{w}]$ is the shape parameter vector described below. We first calculate the center of gravity $\mathbf{c}$ of the region, then calculate the ray features to obtain the shifted distance vector $\mathbf{r}$ and orientation angle Θ. Finally, the GMM weights $\mathbf{w}$ are obtained by maximum-likelihood fitting of $\mathbf{r}$ to the model Eq. (1). The particularity of using shape models in the region growing framework is that the shape model needs to allow also intermediate shapes, i.e., shapes that can be grown into likely objects. In other words, the shape described by $\mathbf{m}_k$ is not necessarily the shape of the object to be segmented but it may be smaller. Let us denote δ the distance of s from the center of gravity $\mathbf{c}$ and let $r = \mathbf{r}(i)$ be the corresponding ray along the line from $\mathbf{c}$ to s. As $\rho(r)$ from Eq. (1) is the density of the boundary being at distance r, we see that $q(s, \mathbf{m}_k)$ is the cumulative probability of finding the boundary at a distance $\delta < r$, which leads to

$$q(s, \mathbf{m}_k) = \int_{\delta}^{\infty} \rho(r)\mathrm{d}r = 1 - \int_{0}^{\delta} \rho(r)\mathrm{d}r,$$

which is easy to evaluate using the cumulative probability density of the GMM. For this calculation, superpixels are represented by their centers. The parameters $\mu_{i,j}$ and $\sigma_{i,j}$ are interpolated from neighboring rays using linear interpolation in angle (see Fig. 6).



(a) (b) (c)

Fig. 6 (a) Statistical shape model represented as the inverted cumulative probability of ray distance distributions in polar coordinates $i \in N$ and $\delta < \mathbf{r}(i)$ where i is the ray index (angle) and δ is a distance; (b) its mapping (and interpolation) to the Euclidean space using superpixels; and (c) the resulting spatial prior q for a single object with standard orientation $\Theta = 0$ (see Fig. 4).

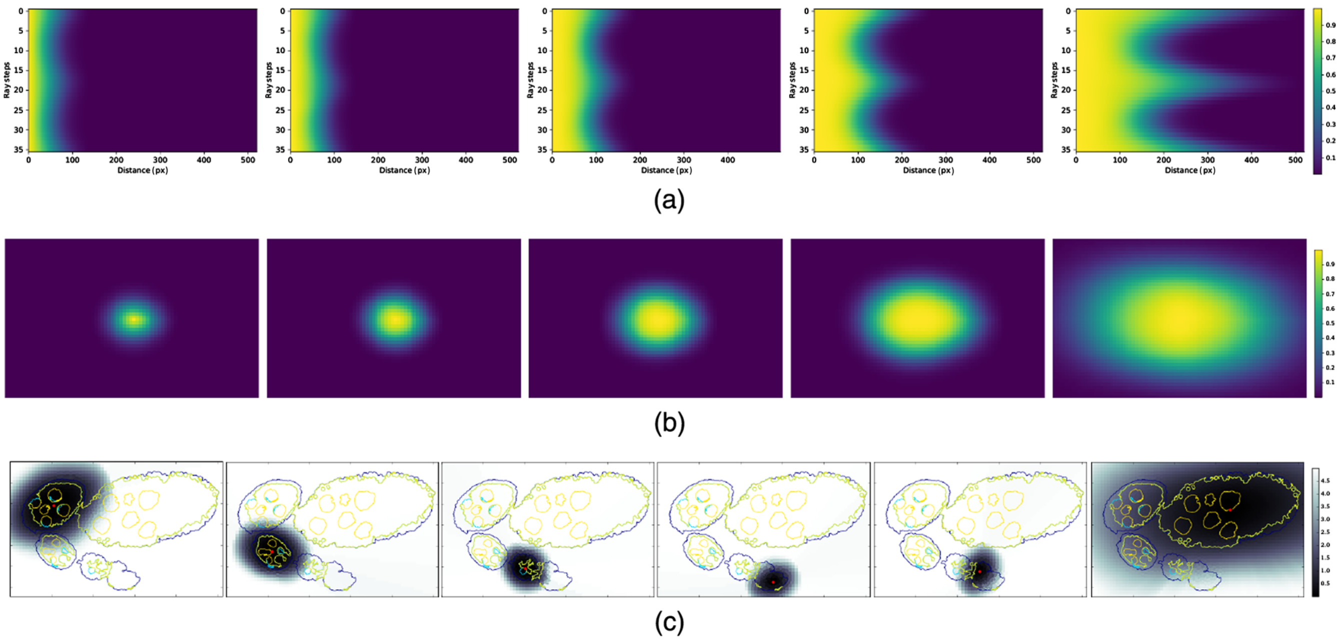


Fig. 7 (a) Inverted cumulative probabilities of ray distances for $M = 5$ components of the GMM; (b) the spatial shape prior q corresponding to each component; and (c) shape cost of fitted models to each of the segmented objects [see Fig. 1(c)] with thin contours presenting levels of appearance probability P_y.

The background probability (for $k = 0$) can be calculated as a complement. Given the estimated parameters of all regions $\mathbf{M} = (\mathbf{m}_1, \dots, \mathbf{m}_K)$, we get

$$P_m\left(g(s) = k|\mathbf{M}\right) = \begin{cases} q(s, \mathbf{m}_k) & \text{for } k > 0 \\ \prod_l (1 - q(s, \mathbf{m}_l)) & \text{for } k = 0 \cdot \end{cases} \quad (2)$$

An example of shape priors for the $M = 5$ GMM components is shown in Fig. 7.

2.5 *Variational Formulation*

The optimal segmentation g^* is found by maximizing the *a posteriori* probability $P(g|y, \mathbf{M})$, where y represents the descriptors of all superpixels. We assume that it can be factorized into appearance, shape, and regularization terms as follows:

$$P(g(s)|y, \mathbf{M}) = \frac{1}{Z(\mathbf{M}, y)} \cdot P_y(g|y) \cdot P_m(g|\mathbf{M}) \cdot P_R(g), \quad (3)$$

where Z is the normalization factor. The appearance and shape terms P_y and P_m, respectively, are expanded assuming independent pixels as follows:

$$P_y(g|y) = \prod_{i\in\Omega} P_y(g(s(i))|y(s(i))) = \prod_{s\in S} P_y(g(s)|y(s))^{|\Omega_s|}, \quad (4)$$

$$P_m(g|\mathbf{M}) = \prod_{i\in\Omega} P_m(g(s(i))|\mathbf{M}) = \prod\nolimits_{s\in S} P_m(g(s)|\mathbf{M})^{|\Omega_s|}, \quad (5)$$

where Ω_s are pixels belonging to a superpixel s and $|\Omega_s|$ is the superpixel size. The neighborhood regularization prior P_R is assumed to factorize as

$$P_R(g) = \prod_{(u,v)\in\mathcal{N}_S} H(g(u), g(v)), \tag{6}$$

where the product is over neighboring superpixels (u, v) and H is chosen such that it encourages them to belong to the same class, see below.

Taking the negative log-likelihood leads to energy minimization $g^* = \arg\min_g E(g)$ with

$$E(g) = \sum_{s\in S} |\Omega_s| [D_s(g(s)) + V_s(g(s))] + \sum_{(u,v)\in\mathcal{N}_S} B(g(u), g(v)), \tag{7}$$

where $\mathcal{N}_S$ is set of all neighboring superpixels along the object boundaries, $D_s(k) = -\log P_y(k|y(s))$ is the data term (described in Sec. 2.2), $V_s(k) = -\log P_m(k|M)$ is the shape term (described in Sec. 2.3).

It remains to define the neighborhood term $B(k, l) = -\log H(k, l)$. The matrix B can be learned from labeled training data. To simplify the task, we shall impose on it the following structure:

$$B(k, l) = \begin{cases} \omega_0 & \text{for } k = l \\ \omega_1 & \text{for } \min (k, l) = 0, k = l\,, \\ \omega_2 & \text{otherwise} \end{cases} \tag{8}$$

where ω_1 and ω_2 represent penalties for an object superpixel touching a background or another object, respectively; ω_0 can be calculated from the partitioning of unity condition $\sum_{k,l} H(k, l) = 1$. In our case, we obtain approximately $\omega_1 = -\log(0.1)$ and $\omega_2 = -\log(0.03)$.

To compensate for model imperfections, it turns out to be useful to add multiplicative coefficients β and γ to modify the relative strength of the three terms

$$E'(g) = \sum_{s\in S} |\Omega_s| [D_s(g(s)) + \beta V_s(g(s))] + \sum_{(u,v)\in\mathcal{N}_S} \gamma B(g(u), g(v)). \tag{9}$$

It can be solved by a standard graph cut method.[55]

2.6 *Region Growing*

We use an iterative approach to find a labeling g minimizing the global energy E in Eq. (9). We alternate two steps: (1) update the shape parameters $\mathbf{M}$ for fixed labels g and (2) optimizing the labels g for $\mathbf{M}$ fixed; see Algorithm 1. The initial object labeling g is derived from user-provided initial object centers c_k. The objects start as small as possible (one superpixel) and grow. For our application, object centers can be obtained automatically using a random forest classifier, neighborhood label histograms, ray features, and density-based spatial clustering.[8]

Updating $\mathbf{M}$ is straightforward and quick—for all superpixels S_k currently assigned to object k, we calculate their center of gravity $\mathbf{c}$, the ray distances $\mathbf{r}$, the angle Θ of the longest ray, and the weights $\mathbf{w}$ as described in Sec. 2.4.

Let us now consider how to update the superpixel labels g. For speed-up, simplification, and in the spirit of region growing, we only allow changing a label of superpixels ∂S_k neighboring an object S_k from the previous iteration and only to a label k. This has the important property that the objects remain compact (simply connected), see Fig. 8. We have considered four optimization strategies for the superpixel labels.

Algorithm 1 Region growing.

```
Input S: superpixels, y: superpixel descriptors, c_k: initial object
      centers, M: statistical shape model
Output: object segmentation g
compute data cost D;
from object centers c_k set initial segmentation g and model shape
parameters m_k;
compute shape cost V;
while not converged do
    update object pose parameters c_k and Θ_k;
    if significant change of center c_k position, orientation Θ_k and
    object area then
        update remaining object shape parameters m_k;
        update shape costs V for all s close to k;
    end
    find superpixels ∂S_k on the external object boundary of k;
    optimize Eq. (9) wrt g by changing s ∈ ∂S_k using the greedy or
    Graph Cut algorithms;
end
```

Greedy approach: We define a priority queue containing background superpixels s from $\partial S = \cup_k \partial S_k$ sorted by the energy improvement ΔE_s^k obtained by switching s to object k, which is a neighbor of s. A superpixel s is removed from the top of the queue if the energy

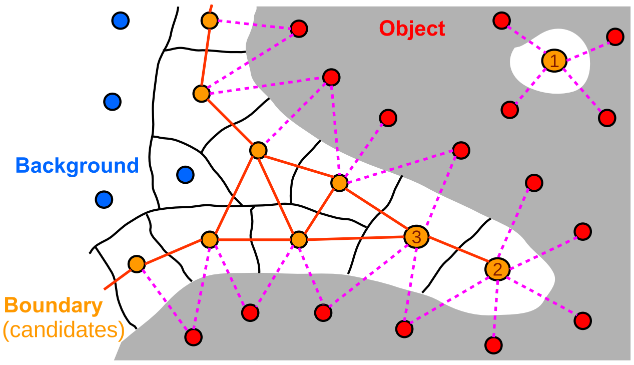


Fig. 8 Creating a graph from ∂S_k on the boundary of object S_k. We connect all candidates of being objects neighboring superpixels ∂S_k (orange). For purposes of compactness, we also connect the neighboring object S_k (red) superpixels. This configuration imply pairwise penalty and impose the object compactness, see e.g., $s \in \{1,2\}$.

improvement ΔE_s^k is positive, it is switched to the object label k, the model $\mathbf{m}_k$ is updated, and also the energy improvement ΔE of all superpixels neighboring with object k. The convergence can be accelerated by processing several best superpixels from the top of priority queue at once. This threshold can be a fixed number of superpixels or relative energy improvement, switching s where $\Delta E_s^k > \epsilon E$. Note, the condition $\Delta E_s^k > 0$ still holds and once assigned s to an object k in single iteration cannot be assigned later to another object l. Regarding the optimality and growth strategy, this number of assigned s in single iteration should be small.

Multiclass graph cut: This approach attempts to find optimal labels $g(s)$ for superpixels from ∂S, and the remaining labels are fixed. We create a graph from the superpixels $\bar{S}$ with edges connecting neighbors. We set the potentials from Eq. (9). A superpixel may only get a value of one of its neighboring object or a background. Other changes are forbidden by setting the corresponding unary potential to ∞. For optimizing this graph problem, the standard $\alpha\beta$-swap graph cut algorithm is used.[55]

Binary graph cut: As a simplification, the binary graph cut considers that a background superpixel $s \in \partial S_k$ neighboring with a single object k can either remain background or be switched to the label of its neighbor (see Fig. 8). The modified unary and binary energy terms are obtained by a restriction of the general formulation Eq. (9) to the two possibilities for each s. The advantage of this formulation is that finding a global minimum is guaranteed and can be done quickly. We perform this binary graph cut sequentially on all objects and so it is convenient to allow swapping object labels, description follows.

Swapping object labels If two objects k and l touch during the optimization process, we found it useful to allow the superpixels on the boundary to exchange labels, thus shifting the border to reflect the shape models, even after the two objects have touched (see iterations in Fig. 9 and results in Fig. 10). It is implemented by adding these boundary pixels to the set ∂S.

3 Experiments

The experiments are performed on a large dataset (containing more than 15,000 images) of microscopy images of Drosophila ovaries, containing egg chambers at various

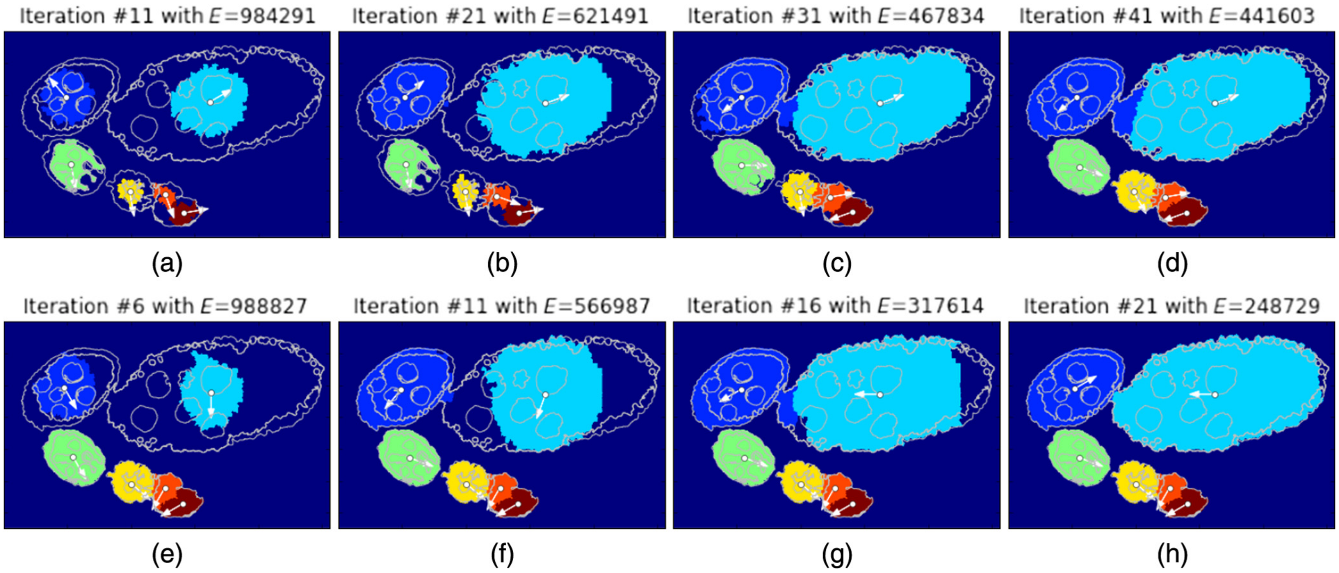

Iteration #11 with E=984291
Iteration #21 with E=621491
Iteration #31 with E=467834
Iteration #41 with E=441603
(a)
(b)
(c)
(d)
Iteration #6 with E=988827
Iteration #11 with E=566987
Iteration #16 with E=317614
Iteration #21 with E=248729
(e)
(f)
(g)
(h)

Fig. 9 (a–d) Several iteration steps of the greedy region growing, and (e–h) of the multiclass graph cut optimization with object label swapping algorithms. Each color region corresponds to an individual object k, the thin white contours are levels of the appearance probabilities P_y. White dots represent the centers $\mathbf{c}_k$ of mass and white arrows the principal orientations Θ_k for each object.

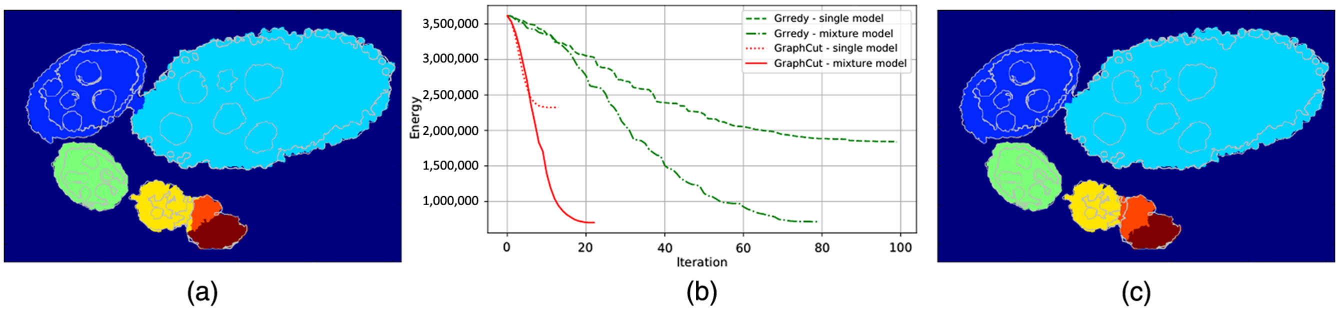


Fig. 10 Example of region growing segmentation results applying (a) greedy approach and (c) graph cuts. (b) We also show the energy evolution during iterations. Each color region corresponds to an object, the thin white contours are levels of the appearance probabilities P_y.

stages of development.[56,57] For 72 images, containing ∼250 eggs, we have a full pixel-level manual segmentation. The images have two channels but only the cell anatomy channel [shown in magenta in Fig. 1(a)] is used.

For evaluation of the segmentation performances, we used the standard measures[58]—F_1-score, accuracy, precision, and recall. We also use the Jaccard index—computed on the binary object/background results. For all experiments, we

Table 1 Quantitative evaluation of the segmentation quality for several configurations of our region growing method.

Configurations							
Method	Model	Object swap	Jaccard	Accuracy	F_1 score	Precision	Recall
Greedy	Single	No	0.6433	0.9324	0.9324	0.9324	0.9324
Greedy	Single	Yes	0.6367	0.9299	0.9299	0.9299	0.9299
Greedy	Mixture	No	0.7377	0.9583	0.9583	0.9583	0.9583
Greedy	Mixture	Yes	0.7527	0.9577	0.9577	0.9577	0.9577
Graph cut	Single	No	0.6426	0.9317	0.9317	0.9317	0.9317
Graph cut	Single	Yes	0.6220	0.9284	0.9284	0.9284	0.9284
Graph cut	Mixture	No	0.7360	0.9573	0.9573	0.9573	0.9573
Graph cut	Mixture	Yes	0.7544	0.9568	0.9568	0.9568	0.9568

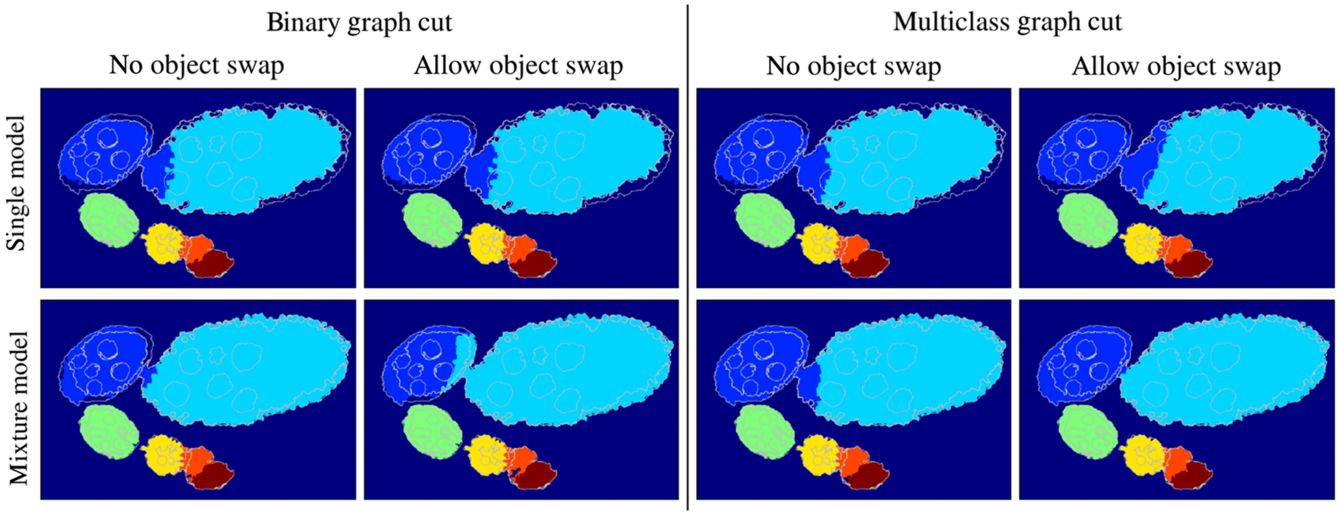


Fig. 11 Resulting segmentation for several different variants of our method: single Gaussian model (top row) versus GMM (bottom row), the binary and multiclass graph cut on the left and right half, respectively. Colored regions represent individual objects and white levels the contours or segmentation Y.

Table 2 Dependency of running time on superpixels sizes (respectively number of superpixels) with regularization $\nu = 0.3$. Note, the code has not been yet optimized for speed.

		Superpixel size [pixels]						
		10	15	20	25	30	35	40
Greedy	Time [seconds]	1468	225	98	72	38	32	27
	Jaccard	0.755	0.754	0.753	0.753	0.752	0.746	0.741
Graph cut	Time [seconds]	94	41	21	9	7	6	5
	Jaccard	0.756	0.755	0.754	0.754	0.753	0.748	0.743

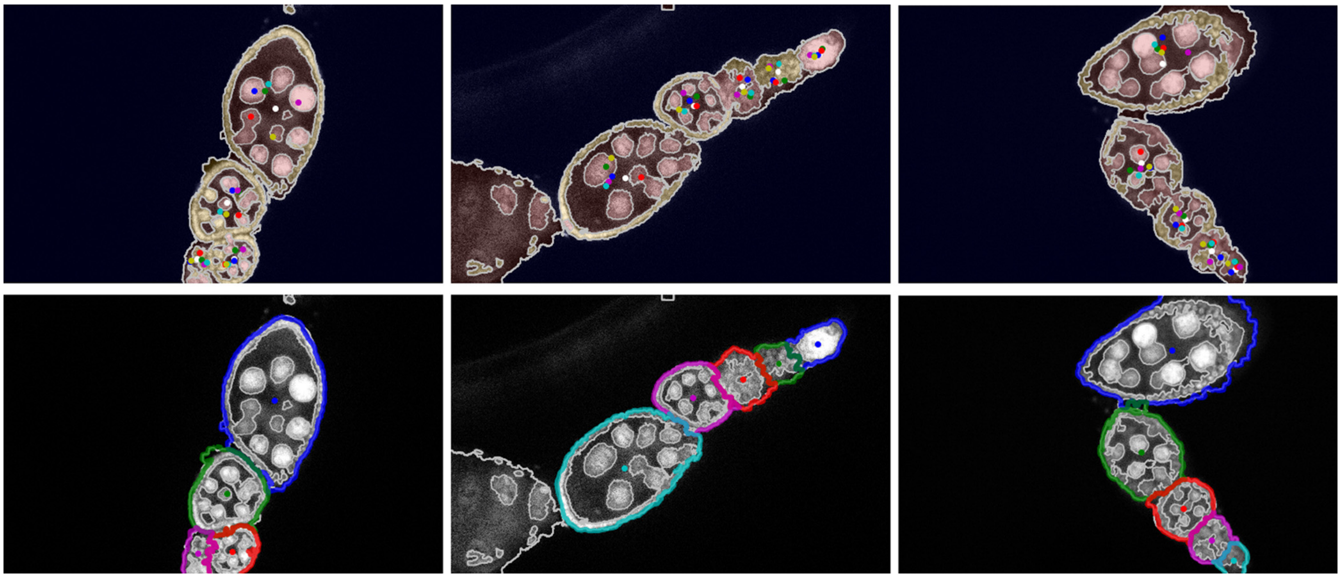


Fig. 12 The impact of quality of initial center selection (top row) on the final segmentation (bottom row). Each set of initial centers (colored equally for all eggs) was obtained by adding random displacement regarding particular egg size. For all initialization, the region growing converged to the same segmentation.

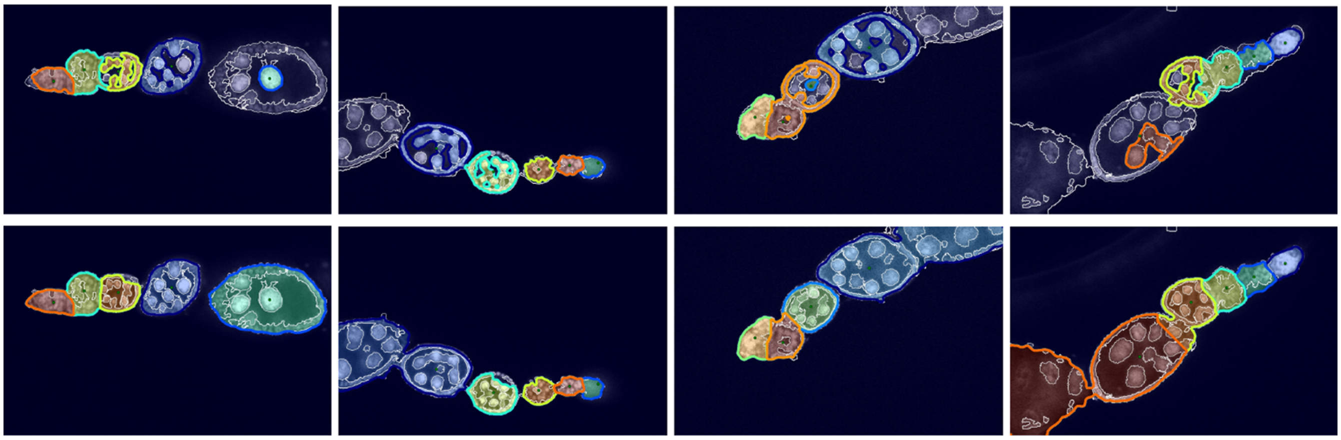


Fig. 13 Examples of resulting segmentation using morphological snakes on input images directly (top row) and on appearance probabilities P_y (bottom row). The manual annotation for these images is presented in Fig. 14.

Table 3 Quantitative comparison of the proposed region growing method (RG2Sp) with other baseline methods.

	Jaccard	Accuracy	F_1 score	Precision	Recall	Time [seconds]
Watershed	0.5705	0.9246	0.9246	0.9246	0.9246	5
Watershed (w. morph.)	0.5705	0.9270	0.9198	0.9136	0.9327	7
Morph. snakes (image)	0.4251	0.8769	0.8070	0.9053	0.7987	784
Morph. snakes (P_y)	0.6494	0.8812	0.8812	0.8812	0.8812	968
Graph cut (pixel-level)	0.7143	0.9204	0.9204	0.9204	0.9204	15
Graph cut (superpixels)	0.3164	0.8643	0.8643	0.8643	0.8643	3
RG2Sp (greedy)	0.7527	0.9577	0.9577	0.9577	0.9577	72
RG2Sp (graph cut)	0.7544	0.9568	0.9568	0.9568	0.9568	9

used the following SLIC parameters: superpixel size $\eta = 20$ pixels and regularization $\nu = 0.3$. The average running time to calculate the superpixels for our images of size 1000×1000 pixels was about 1 seconds. The proposed method has a few parameter to set—the coefficient β and γ, and the update thresholds in Algorithm 1. Experimentally, we found that setting $\beta = 2$ and $\gamma = 5$ give the best results for our images. We set the threshold for a shift to 20 pixels

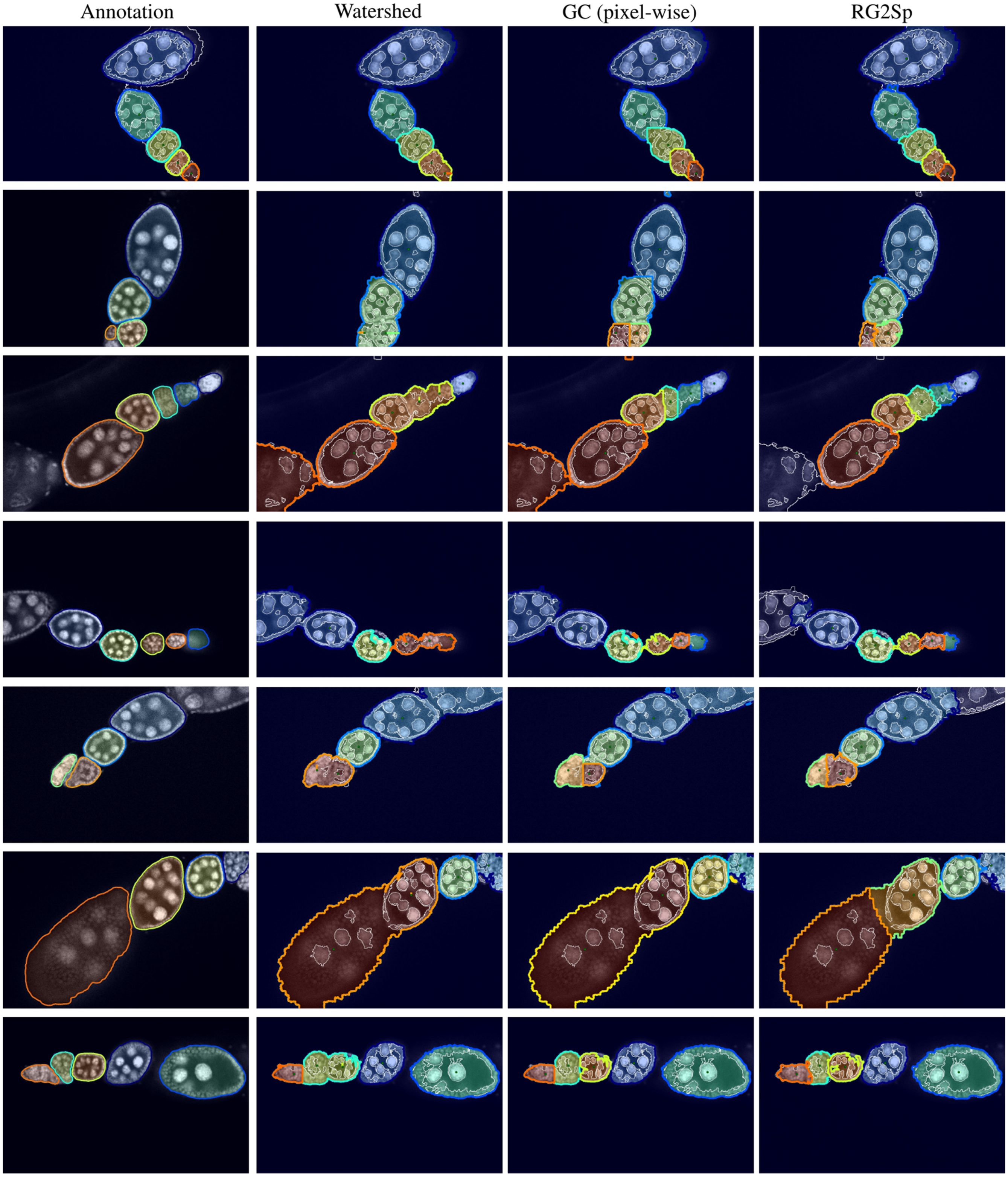


Fig. 14 Each row represents a microscopy image segmented by an expert (annotation) and the three automatic methods—from left to right: watershed, graph cut on pixels, and region growing. The expert annotation is shown overlaid on the input image. The segmentation results are shown overlaid over the input image with the preliminary four-class segmentation contours shown as thin white lines.

(superpixel size), rotation 10 deg, and volume change to 5%. These values allow reaching the same segmentation quality as with updating V in every iteration, about twice as fast.

3.1 *Comparison of Region Growing Variants*

We compare the different variants of our segmentation method: using graph cut versus greedy approach, GMM (with $M = 15$) versus single Gaussian (assuming GMM with $M = 1$), allowing object label swapping. Quantitative results are shown in Table 1. It confirms our expectation (see Fig. 11) that it is best to use a GMM with a multiclass graph cut and label swapping with respect to Jaccard index which well reflects our visual observation.

Let us discuss the behavior of the graph cut and greedy region growing algorithms. The resulting segmentation of both graph cut and greedy is very similar. Speaking about the second criterion—processing time, the graph cut is faster in terms of a number of the iterations (see Fig. 10), but each iteration is a little longer. The total processing time of the graph cut approach is about 9 seconds compared to greedy which takes about 72 seconds per image. We experimented with superpixel sizes and observed that they do not have a large influence on the segmentation quality, but they have a significant impact on the processing time, see Table 2. Region growing speeds up with larger superpixels and consequently there are fewer candidates to evaluate.

We also experimented with the dependence of the resulting segmentation on the position of the initial centers c_k. We found that our method is very robust to the initialization—for a center initialization up to 1/2 distance between the true center and the object boundary, we obtained visually equivalent results, see Fig. 12.

3.2 *Baseline Methods*

We apply all methods on the results of the preliminary four-class segmentation Y [see an example in Fig. 1(b)] as there is no method that can well segment individual eggs in our microscopy images of Drosophila ovaries directly. For comparison we chose such methods to cover a wide range of segmentation approaches that can be potentially used for this task—object segmentation, as we discussed in Sec. 1.

Watershed segmentation:[59–61] This is widely used for separating touching objects. We start from the binarized segmentation[7,8] and apply the distance transform to calculate the distance of each pixel to the background. The watershed algorithm starting from initial centers is then used to identify individual objects. We also tested some morphological operations such as opening, before applying the watershed to see the improvement of the egg separation. It then turned out that selecting a universal structure element (SE) for all images is not reasonable because of (i) the large variance in egg size and (ii) connection thickness in between two eggs—a small SE does not always split neighboring eggs and a large SE may suppress the appearance of small eggs. We remark that in the experiments, we used morphological opening with the circular SE with 25 pixels in diameter.

Morphological snakes:[62] We used multiple morphological snakes with smoothing 3 and $\lambda_{1,2} = 1$ initialized from the circular region around the center with diameter 20 pixels which are approximately the size of used superpixels evolving in parallel. We also adopted a restriction that individual snakes cannot overlap. We apply the multisnakes approach on the input image directly and also on appearance probabilities P_y, see Fig. 13. The snakes on raw images frequently struggle with handling internal egg structure, on the other hand, snakes on P_y have difficulty separating touching eggs.

Pixel-level graph cuts:[63] This optimizes an energy function similar to the previous method but at the pixel-level. The data term is distributed from superpixels to pixels in a straightforward way, $D_i(k) = -\log P_y(k|y(s(i)))$ for all pixels $i \in \Omega$ and standard pairwise regularization for the Potts model. Pixels from a small region around the provided initial object centers c_k are forced to class k.

Superpixel-level graph cut:[24] This works similarly as above except that we assign classes k to superpixels, not pixels. The energy function $E(g)$ from Eq. (7) can be used directly, again without the shape cost V and without employing region growing.

3.3 *Comparison with Baseline Methods*

In the final experiment, we compare the performance of our selected method, i.e., region growing with a GMM, multiclass graph cut, and label swapping, and compare it with alternative baseline methods. Table 3 presents the quantitative results. We can say that the proposed method performed better than the other methods in all comparable metrics.

Example segmentation results are shown in Fig. 14. We can see that the comparable methods usually fail to properly distinguish touching eggs. Also, they are frequently merging two eggs together even if the second egg does not contain an initial seed which can happen in real-world application.[8]

4 Conclusion

We presented a new region growing segmentation technique. It is fast due to using superpixels, and it is also robust due to handling the growing with a graph cut and a ray feature-based shape model. It can handle touching objects as well as objects with only partly visible boundaries. Our method is developed with a specific application in mind where we have shown it to perform better than the baseline methods. However, it can be easily generalized and applied to other domains, whenever a set of objects with known shapes is to be segmented.

5 Source Code

The implementation of the proposed method together with other tested methods will be available in a Github repository: http://github.com/Borda/pyImSegm.

Disclosures

The authors have no relevant financial interests in this article and no potential conflicts of interest to disclose.

Acknowledgments

This work was supported by the Czech Science Foundation project 14-21421S and by the Grant Agency of the Czech Technical University in Prague under the grant SGS15/

154/OHK3/2T/13. A part of this work was carried out under the NII International Internship Program.

References

1. M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, 3rd ed., Thomson Engineering, Toronto (2007).
2. D. L. Pham, C. Xu, and J. L. Prince, "A survey of current methods in medical image segmentation," *Ann. Rev. Biomed. Eng.* **2**, 315–337 (2000).
3. K.-P. Wong, "Medical image segmentation: methods and applications in functional imaging," in *Handbook of Biomedical Image Analysis*, D. Wilson and S. Laxminarayan, Eds., pp. 111–182, Springer, Boston, Massachusetts (2005).
4. A. Elnakib et al., "Medical image segmentation: a brief survey," in *Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies*, A. S. El-Baz et al., Eds., pp. 1–39, Springer, New York (2011).
5. P. Tomancak et al., "Systematic determination of patterns of gene expression during Drosophila embryogenesis," *Genome Biol.* **3**(12), research0088.1 (2002).
6. J. Borovec and J. Kybic, "Binary pattern dictionary learning for gene expression representation in drosophila imaginal discs," in *Mathematical and Computational Methods in Biomedical Imaging and Image Analysis (MCBMIIA) Workshop at ACCV*, pp. 555–569 (2016).
7. R. Nava and J. Kybic, "Supertexton-based segmentation in early Drosophila oogenesis," in *Proc. Int. Conf. on Image Processing (ICIP)*, pp. 2656–2659 (2015).
8. J. Borovec, J. Kybic, and R. Nava, "Detection and localization of Drosophila egg chambers in microscopy images," in *8th Int. Workshop on Machine Learning in Medical Imaging*, Springer, Quebec (2017).
9. S. W. Zucker, "Region growing: childhood and adolescence," *Comput. Graphics Image Process.* **5**(3), 382–399 (1976).
10. R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Anal. Mach. Intell.* **16**(6), 641–647 (1994).
11. C. Revol-Muller et al., "Region growing: when simplicity meets theory—region growing revisited in feature space and variational framework," *Commun. Comput. Inf. Sci.* **359**, 426–444 (2013).
12. R. Achanta and A. Shaji, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(11), 2274–2282 (2012).
13. D. Stutz, A. Hermans, and B. Leibe, "Superpixels: an evaluation of the state-of-the-art," *Comput. Vision Image Understanding* (2017).
14. P. Buyssens et al., "Eikonal-based region growing for efficient clustering," *Image Vision Comput.* **32**(12), 1045–1054 (2014).
15. K. Smith and A. Carleton, "Fast ray features for learning irregular shapes," in *IEEE 12th Int. Conf. on Computer Vision*, pp. 397–404 (2009).
16. K.-M. Lee and W. N. Street, "Learning shapes for automatic image segmentation," in *Proc. INFORMS-KORMS Conf.*, pp. 1461–1468 (2000).
17. J. L. Rose et al., "Shape prior integrated in an automated 3D region growing method," in *Proc. Int. Conf. on Image Processing (ICIP)*, Vol. **1** (2007).
18. J. L. Rose et al., "3D region growing integrating adaptive shape prior," in *5th IEEE Int. Symp. on Biomedical Imaging: From Nano to Macro*, pp. 967–970 (2008).
19. J. L. Rose et al., "Shape prior criterion based on Tchebichef moments in variational region growing," in *Proc. Int. Conf. on Image Processing (ICIP)*, pp. 1081–1084 (2009).
20. A. Quispe and C. Petitjean, "Shape prior based image segmentation using manifold learning," in *5th Int. Conf. on Image Processing, Theory, Tools and Applications (IPTA)*, pp. 137–142 (2015).
21. P. Etyngier, F. Segonne, and R. Keriven, "Shape priors using manifold learning techniques," in *IEEE 11th Int. Conf. on Computer Vision*, pp. 1–8 (2007).
22. O. Moolan-Feroze et al., "Segmentation of the right ventricle using diffusion maps and Markov random fields," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*, Vol. **8673**, pp. 682–689 (2014).
23. Y. Li et al., "Lazy snapping," *ACM Trans. Graphics* **23**, 303–308 (2004).
24. X. Ye, G. Beddoe, and G. Slabaugh, "Automatic graph cut segmentation of lesions in CT using mean shift superpixels," *Int. J. Biomed. Imaging* **2010**, 1–14 (2010).
25. Q. Yu and D. A. Clausi, "IRGS: image segmentation using edge penalties and region growing," *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(12), 2126–2139 (2008).
26. K. Qin and D. A. Clausi, "Multivariate image segmentation using semantic region growing with adaptive edge penalty," *IEEE Trans. Image Process.* **19**, 2157–2170 (2010).
27. M. P. Kumar, P. H. S. Torr, and A. Zisserman, "OBJ CUT," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, Vol. **1**, pp. 18–25 (2005).
28. D. Freedman and T. Zhang, "Interactive graph cut based segmentation with shape priors," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, Vol. **1**, pp. 755–762 (2005).
29. N. Vu and B. S. Manjunath, "Shape prior segmentation of multiple objects with graph cuts," in *26th IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8 (2008).
30. O. Veksler, "Star shape prior for graph-cut image segmentation," in *European Conf. on Computer Vision (ECCV)*, pp. 454–467, Springer (2008).
31. K. Nakagomi et al., "Multi-shape graph cuts with neighbor prior constraints and its application to lung segmentation from a chest CT volume," *Med. Image Anal.* **17**(1), 62–77 (2013).
32. T. Schoenemann and D. Cremers, "Globally optimal image segmentation with an elastic shape prior," in *Proc. of the IEEE Int. Conf. on Computer Vision* (2007).
33. A. Delong and Y. Boykov, "Globally optimal segmentation of multi-region objects," in *Proc. of the IEEE Int. Conf. on Computer Vision*, pp. 285–292 (2009).
34. J. Ulen, P. Strandmark, and F. Kahl, "An efficient optimization framework for multi-region segmentation based on Lagrangian duality," *IEEE Trans. Med. Imaging* **32**(2), 178–188 (2013).
35. H. N. Isack et al., "Efficient optimization for hierarchically-structured interacting segments (HINTS)," in *Proc. of CVPR* (2017).
36. H. Lu et al., "Active contours model for image segmentation: a review," in *Proc. of the 1st Int. Conf. on Industrial Application Engineering*, pp. 104–111 (2013).
37. K. Zhang et al., "Active contours with selective local or global segmentation: a new formulation and level set method," *Image Vision Comput.* **28**(4), 668–676 (2010).
38. O. Dzyubachyk et al., "Advanced level-set-based cell tracking in time-lapse fluorescence microscopy," *IEEE Trans. Med. Imaging* **29**(3), 852–867 (2010).
39. L. Vese and T. Chan, "A multiphase level set framework for image segmentation using the Mumford and Shah model," *Int. J. Comput. Vision* **50**(3), 271–293 (2002).
40. B. C. Lucas, M. Kazhdan, and R. H. Taylor, "Multi-object geodesic active contours (MOGAC)," in *15th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 404–412, Springer, Berlin Heidelberg (2012).
41. N. Paragios and R. Deriche, "Coupled geodesic active regions for image segmentation: a level set approach," in *6th European Conf. on Computer Vision*, pp. 224–240, Springer, Berlin Heidelberg (2000).
42. T. F. Cootes et al., "Active shape models—their training and application," *Comput. Vision Image Understanding* **61**(1), 38–59 (1995).
43. M. E. Leventon, W. E. L. Grimson, and O. D. Faugeras, "Statistical shape influence in geodesic active contours," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 316–323, IEEE Computer Society (2000).
44. A. Tsai et al., "A shape-based approach to the segmentation of medical imagery using level sets," *IEEE Trans. Med. Imaging* **22**(2), 137–154 (2003).
45. M. Gastaud, M. Barlaud, and G. Aubert, "Combining shape prior and statistical features for active contour segmentation," *IEEE Trans. Circuits Syst. Video Technol.* **14**(5), 726–734 (2004).
46. C. Molnar, Z. Kato, and I. Jermyn, "A multi-layer phase field model for extracting multiple near-circular objects," in *Int. Conf. on Pattern Recognition (ICPR)*, pp. 1427–1430 (2012).
47. C. Molnar et al., "Accurate morphology preserving segmentation of overlapping cells based on active contours," *Sci. Rep.* **6**(1), 1–10 (2016).
48. Y. Shi and W. C. Karl, "Real-time tracking using level sets," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Vol. **2**, pp. 34–41 (2005).
49. J. Kybic and J. Krátký, "Discrete curvature calculation for fast level set segmentation," in *Int. Conf. on Image Processing (ICIP)*, pp. 3017–3020, IEEE, Piscataway, New Jersey, electronic version (2009).
50. O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer International Publishing, Cham (2015).
51. B. Romera-Paredes and P. H. S. Torr, "Recurrent instance segmentation," *Lect. Notes Comput. Sci.* **9910**, 312–329 (2016).
52. J. Borovec and J. Kybic, "jSLIC: superpixels in ImageJ," in *Computer Vision Winter Workshop*, Z. Kunbelova and J. Heller, Eds., pp. 14–18, Czech Society for Cybernetics and Informatics, Praha (2014).
53. A. Lucchi, K. Smith, and R. Achanta, "Supervoxel-based segmentation of mitochondria in EM image stacks with learned shape features," *IEEE Trans. Med. Imaging* **31**(2), 474–486 (2012).
54. G. Xuan and W. Zhang, "EM algorithms of Gaussian mixture model and hidden Markov model," in *Proc. Int. Conf. on Image Processing*, Vol. **1**, pp. 145–148 (2001).
55. Y. Boykov and O. Veksler, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(11), 1222–1239 (2001).
56. D. A. Baker and S. Russell, "Gene expression during Drosophila melanogaster egg development before and after reproductive diapause," *BMC Genomics* **10**, 242 (2009).

57. D. Jia et al., "Automatic stage identification of Drosophila egg chamber based on DAPI images," *Sci. Rep.* **6**, 18850 (2016).
58. O. Koyejo et al., "Consistent multilabel classification," in *Advances in Neural Information Processing Systems*, pp. 3321–3329 (2015).
59. S. Beucher, "The watershed transformation applied to image segmentation," in *Proc. of the 10th Pfefferkorn Conf. on Signal and Image Processing in Microscopy and Microanalysis*, pp. 299–314 (1992).
60. Q. C. Q. Chen, X. Y. X. Yang, and E. Petriu, "Watershed segmentation for binary images with different distance transforms," in *Proc. Second Int. Conf. on Creating, Connecting and Collaborating through Computing*, Vol. **2**, pp. 111–116 (2004).
61. X. Ji et al., "Cell image segmentation based on an improved watershed algorithm," in *Proc. 8th Int. Congress on Image and Signal Processing (CISP)*, pp. 433–437 (2016).
62. P. Marquez-Neila, L. Baumela, and L. Alvarez, "A morphological approach to curvature-based evolution of curves and surfaces," *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(1), 2–17 (2014).
63. Y. Boykov, "Graph cuts and efficient N-D image segmentation," *Int. J. Comput. Vision* **70**, 109–131 (2006).

Jiří Borovec was awarded an MSc double-degree in intelligent systems by the Université Paul Sabatier, France, and the Technical University of Liberec, Czechia, in 2011. He is now pursuing a PhD at the Center of Machine Perception, which is a part of the Department of Cybernetics of the Faculty of Electrical Engineering of the Czech Technical University (CTU) in Prague.

Jan Kybic received a master's degree from the CTU in Prague, Czech Republic, and a PhD from EPFL, Switzerland, in 1998 and 2001, respectively. He held a postdoc position at INRIA, France, in 2002 to 2003. Since 2003, he has been at CTU in Prague, becoming a full professor in 2015 and currently serving as a research group leader.

Akihiro Sugimoto received a PhD in mathematical engineering from the University of Tokyo, Japan. After working for Hitachi, ATR, and Kyoto University, in 2002 he joined the National Institute of Informatics, Tokyo, Japan, where he is currently a full professor. From 2006 to 2007, he was a visiting professor at the University of Paris-Est Marne-la-Vallee, France. His current main research interests include discrete mathematics, optimization algorithm, vision geometry, and modeling of human vision.