Region growing using superpixels with learned shape prior

Jiří Borovec[a,*], **Jan Kybic**[a], **Akihiro Sugimoto**[b]
[a]Biomedical Imaging Algorithms Group, CMP, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, 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 speedup. Second, our method uses learned statistical shape properties which 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.

Keywords: segmentation, region growing, superpixels, shape prior, ray features, Graph cuts, Drosophila ovary .

*Jiří Borovec, jiri.borovec@fel.cvut.cz

1 Introduction

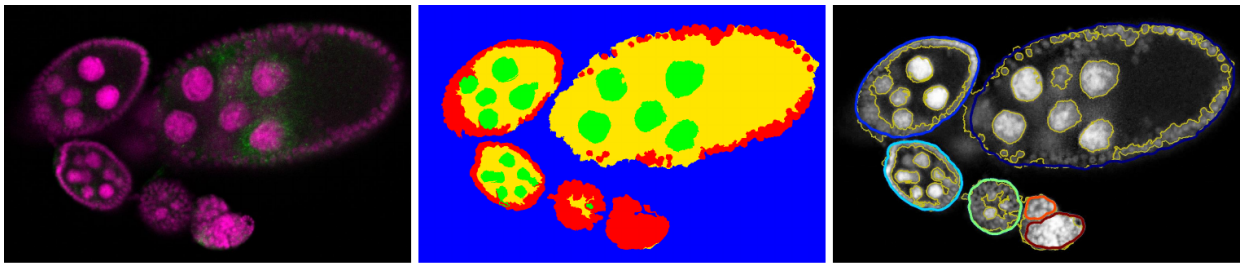
Image segmentation is one of the fundamental image analysis tasks,[1–4] consisting of dividing an image into multiple regions (or classes). Our target application is segmenting individual eggs in microscopy images of Drosophila ovaries (see Figure 1(a), 1(b)), which is one of the key steps in the image 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 multi-class segmentation[7,8] (see Figure 1(b)). The standard approach is to post-process 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 seen in 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 Figure 2(b)). This improves the segmentation speed by several orders of magnitude while the superpixels preserve the object boundaries. We are using the 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



(a) (b) (c)

Figure 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 4-class segmentation (thin yellow contour) with individual eggs marked manually (wide color lines), superimposed over the cell anatomy channel in gray.

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 are 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 Figure 1(b)) as input (see Figure 2(a)) for the region growing — the annotation of individual eggs is shown in Figure 1(c). In this way the region growing can correct imperfections of the preliminary segmentation as we will present later in Section 3.

Second, we incorporate a shape model based on the so-called ray features,[15,16] to guide the region growing towards plausible shapes, using our a priori knowledge.[1] 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]

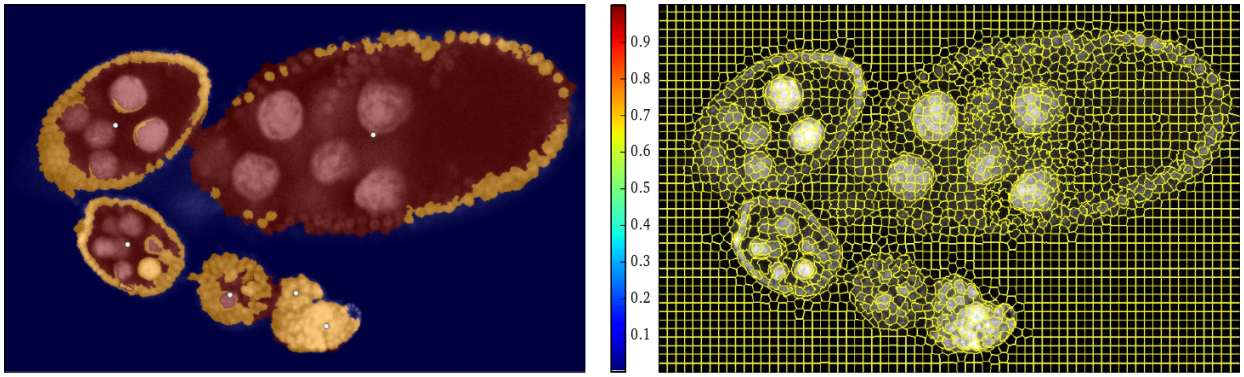
Third, our method segments several objects simultaneously, ensuring that they do not overlap. One iteration of the growing process is formulated as an energy minimization problem with a Markov random field (MRF) regularization and solved either greedily or using Graph Cuts. Since the number of boundary 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 resources demanding.

1.2 Other related methods

Ye[24] uses 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 shape model;[30] it is also possible to choose between multiple models.[31] These methods alternate between estimating the pose parameters and refining the segmentations and can converge to a sub-optimal solution if the pose estimation is incorrect. The number of pose hypotheses that can be

[1] We see that in our application, the eggs are approximately oval.



(a) (b)

Figure 2 Sample Drosophila ovary image with multiple eggs. *(a)* Probability map obtained from the preliminary segmentation shown in Figure 1 representing the likelihood for each superpixels being an egg; *(b)* SLIC superpixels.

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 can be also 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 multi-object 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 a 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 relays on correct boundaries prediction. Another, instance segmentation[51] solves the task at once 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 Section 2, including superpixels (Section 2.1), shape models Section 2.3, optimization (Section 2.6). We continue by experiments in Section 3 and conclude in Section 4.

2 Methods

Given an input image containing multiple non-overlapping 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 (Section 2.1) and for each of them calculate the appearance-based object probability. The regions corresponding to objects are then grown (Section 2.6) using

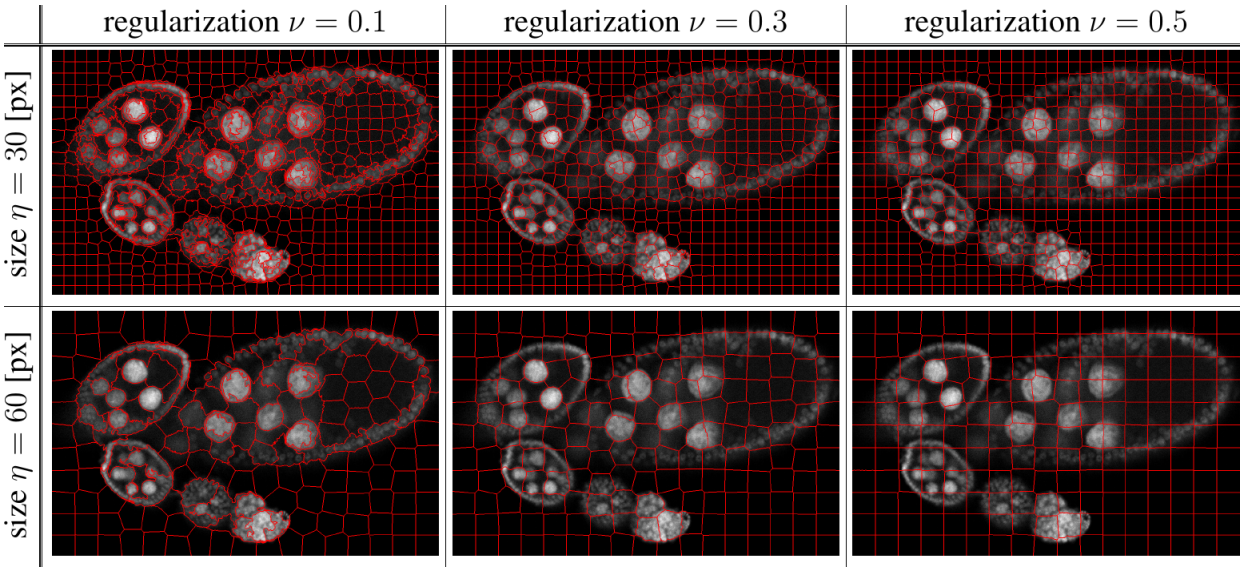


Figure 3 The influence of the SLIC parameters: superpixel size η and regularization ν.

the appearance (Section 2.2) and shape (Section 2.3) models. The final segmentation is represented by a function $g : S \to \{0, 1, \ldots, K\}$, which assigns each superpixel $s \in S$ to one of the objects (if $g(s) \neq 0$) or to the background (if $g(s) = 0$).

2.1 Superpixel clustering

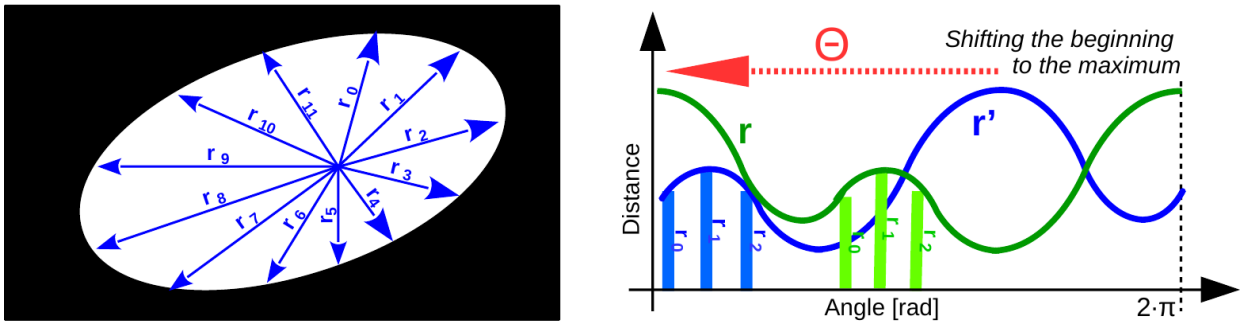
We use the Simple Linear Iterative Clustering[12] (SLIC) algorithm to calculate a set of superpixels S which are compact both in space and color. The SLIC algorithm is an adaptation of a widely-used k-means clustering algorithm. It uses a combined color and spatial distance $D = d_c + \frac{\xi}{\eta^2} \cdot d_s$, where d_c is an Euclidean distance in the CIELAB color space, and d_s is an 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 which 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

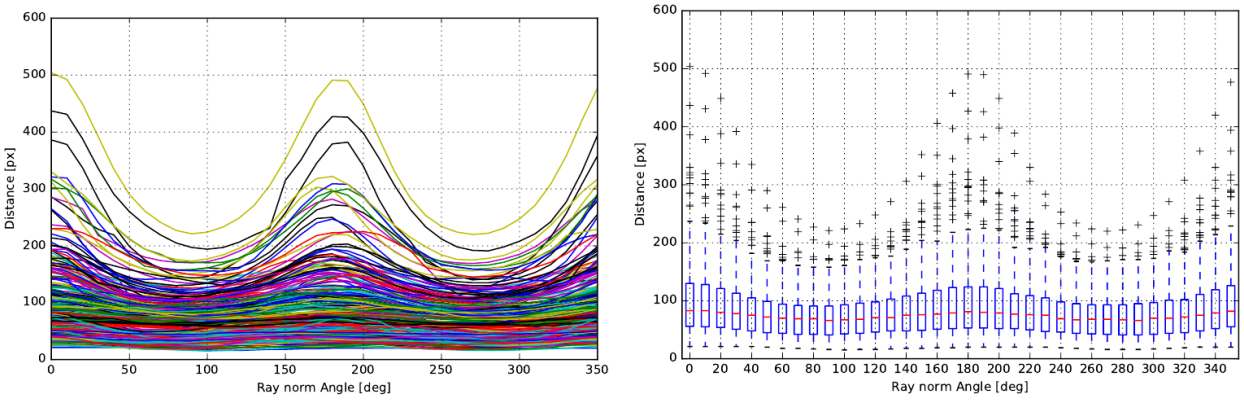
$$P_y\big(g(s)|y_s\big) = \begin{cases} P_y(y_s) & \text{for } g(s) \neq 0 \\ 1 - P_y(y_s) & \text{for } g(s) = 0 \end{cases}$$

For our application, we take advantage of the fact that we already have a good preliminary segmentation method which 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



(a) (b)

Figure 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, $\boldsymbol{r}'$ (in blue) and $\boldsymbol{r}$ (in green) respectively.



(a) (b)

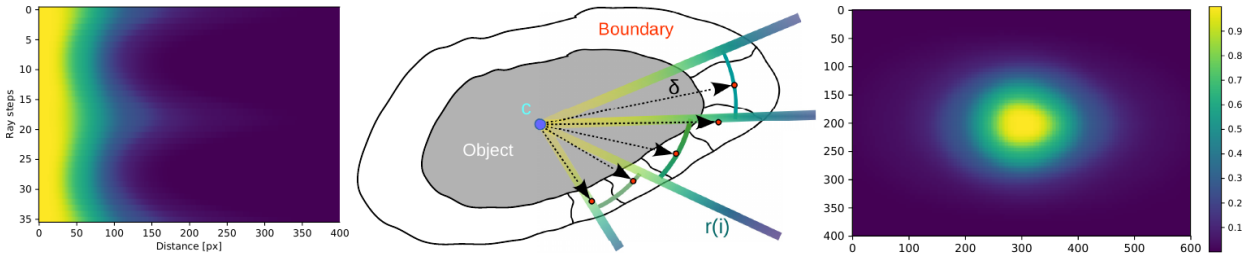
Figure 5 *(a)* Visualisation of 250 egg shapes represented by the distance vectors $\boldsymbol{r}$ and *(b)* their element-wise box and whisker plots for each $\boldsymbol{r}(i)$ with angular step 10^o.

$\{1,\ldots,4\}$, 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 Figure 1(b) for an example of the preliminary segmentation, and Figure 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 $\boldsymbol{c}$ and the so-called ray features[15,53] $\boldsymbol{r}'$, the distances from $\boldsymbol{c}$ to the region boundary in a set of N predefined directions (see Figure 4). To ensure rotation invariance, the distance vector $\boldsymbol{r}' = \{r_0,\ldots,r_{N-1}\}$ is circularly shifted to obtain a rotational normalized vector $\boldsymbol{r}(i) = \boldsymbol{r}'\big((i-\Theta) \bmod N\big)$ such that it starts with the maximum element, $\boldsymbol{r}(0) = \max_i \boldsymbol{r}'(i)$.

As an example, the ray feature vectors $\boldsymbol{r}$ with $N = 36$ (see Figure 5(a)) and the whisker plots for each $\boldsymbol{r}(i)$ independently are shown in Figure 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.



(a) (b) (c)

Figure 6 *(a)* Statistical shape model represented as the inverted cumulative probability of ray distance distributions in polar coordinates $i \in N$ and $\delta < \boldsymbol{r}(i)$; *(b)* its mapping (and interpolation) to the Euclidean space using superpixels; and *(c)* the resulting spatial prior q for a single object with zero orientation $\Theta = 0$.

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

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

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

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 (EM) algorithm,[54] while the M weights $\boldsymbol{w}$ are estimated for each object independently.

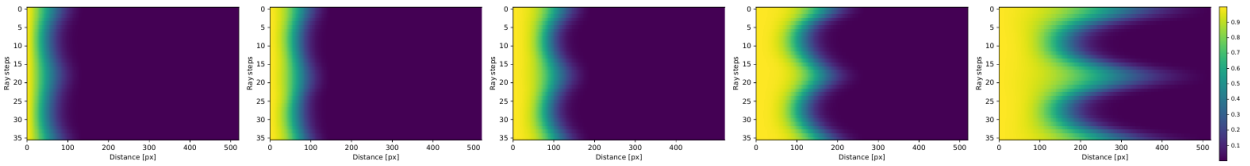
2.4 Shape prior

During region growing, we need to calculate the shape prior $P_m\big(g(s) = k|\boldsymbol{m}_k\big) = q(s, \boldsymbol{m}_k)$ that a given superpixel $s \in S$ belongs to an object k, where $\boldsymbol{m}_k = \big[\boldsymbol{c}, \boldsymbol{r}, \Theta, \boldsymbol{w}\big]$ is the shape parameter vector described below. We first calculate the center of gravity $\boldsymbol{c}$ of the region, then calculate the ray features to obtain the shifted distance vector $\boldsymbol{r}$ and orientation angle Θ. Finally, the GMM weights $\boldsymbol{w}$ are obtained by maximum-likelihood fitting of $\boldsymbol{r}$ to the model (1).

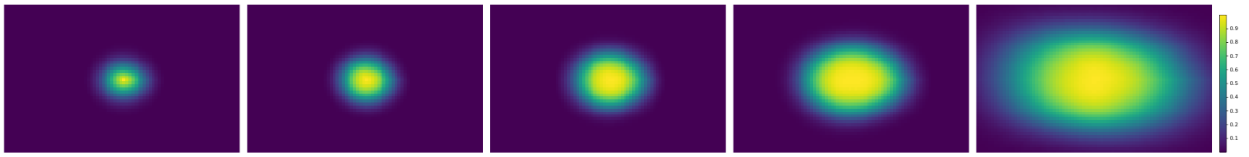
The particularity of using shape models in the region growing framework is that the shape model needs to allow also intermediate shapes, i.e. shapes which can be grown into likely objects. In other words, the shape described by $\boldsymbol{m}_k$ is not necessarily the shape of the object to be segmented but it may be smaller. Let us denote δ the distance of s from the center of gravity $\boldsymbol{c}$ and let $r = \boldsymbol{r}(i)$ be the corresponding ray along the line from $\boldsymbol{c}$ to s. As $\rho(r)$ from (1) is the density of the boundary being at distance r, we see that $q(s, \boldsymbol{m}_k)$ is the cumulative probability of finding the boundary at a distance $\delta < r$, which leads to

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

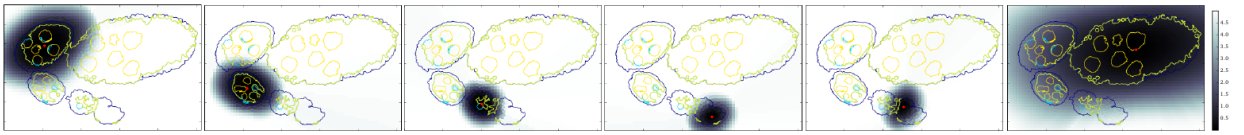
which is easy to evaluate using the cumulative probability density of the GMM. For this calcula-



(a)



(b)



(c)

Figure 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 object with thin contours presenting levels of appearance probability P_y.

tion, superpixels are represented by their centers. The parameters $\mu_{i,j}$, $\sigma_{i,j}$ are interpolated from neighboring rays using linear interpolation in angles (see Figure 6).

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

$$P_m\big(g(s) = k \mid \boldsymbol{M}\big) = \begin{cases} q(s, \boldsymbol{m}_k) & \text{for } k > 0 \\ \prod_l \big(1 - q(s, \boldsymbol{m}_l)\big) & \text{for } k = 0 \end{cases} \tag{2}$$

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

2.5 Variational formulation

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

$$P\big(g(s) \mid y, \boldsymbol{M}\big) = \frac{1}{Z(\boldsymbol{M}, y)} P_y(g \mid y) P_m(g \mid \boldsymbol{M}) P_R(g) \tag{3}$$

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

$$P_y(g \mid y) = \prod_{i \in \Omega} P_y\Big(g\big(s(i)\big) \mid y\big(s(i)\big)\Big) = \prod_{s \in S} P_y\big(g(s) \mid y(s)\big)^{|\Omega_s|} \tag{4}$$

$$P_m(g \mid \boldsymbol{M}) = \prod_{i \in \Omega} P_m\Big(g\big(s(i)\big) \mid \boldsymbol{M}\Big) = \prod_{s \in S} P_m\Big(g(s) \mid \boldsymbol{M}\Big)^{|\Omega_s|} \tag{5}$$

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

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

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

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

where $\mathcal{N}_S$ is set of all neighbouring superpixels along the object boundaries, $D_s(k) = -\log P_y\big(k \mid y(s)\big)$ is the data term (described in Section 2.2), $V_s(k) = -\log P_m\big(k \mid M\big)$ is the shape term (described in Section 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 it the following structure

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

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 $\omega_1 = -\log(0.1)$, $\omega_2 = -\log(0.03)$.

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:

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

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 (9). We alternate two steps: (1) update the shape parameters $\boldsymbol{M}$ for fixed labels g and (2) optimizing

Algorithm 1: Region growing.

Input: S: superpixels, y: superpixel descriptors, $\boldsymbol{c}_k$: initial object centers, $\boldsymbol{M}$: 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 centre 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 (9) wrt g by changing s ∈ ∂S_k using the greedy or Graph Cut algorithms;
end
```

the labels g for $\boldsymbol{M}$ fixed (see Algorithm 1). The initial object labeling g is derived from user provided initial object centers $\boldsymbol{c}_k$. 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 $\boldsymbol{M}$ is straightforward and quick — for all superpixels S_k currently assigned to object k, we calculate their center of gravity $\boldsymbol{c}$, the ray distances $\boldsymbol{r}$, the angle Θ of the longest ray, and the weights $\boldsymbol{w}$ as described in Section 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 to change 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 Figure 8. We have considered four optimization strategies for the superpixel labels.

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 improvement ΔE_s^k is positive, it is switched to the object label k, the model $\boldsymbol{m}_k$ is updated, and also the energy improvement ΔE of all superpixels neighbouring 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 cant 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 approach attempts to find optimal labels $g(s)$ for superpixels from ∂S, the remaining labels are fixed. We create a graph from the superpixels $\bar{S}$ with edges connecting

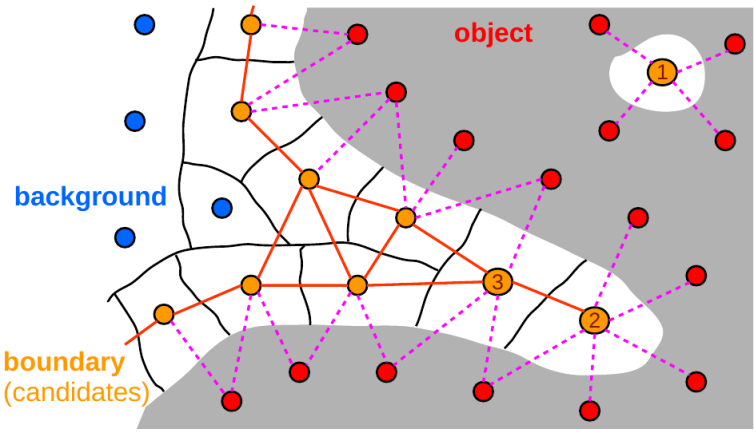


Figure 8 Creating a graph from ∂S_k on the boundary of object S_k. We connect all candidates of being objects neighboring superpixels ∂S_k (orange). For purposes of compactness, we also connect the neighboring object S_k (red) superpixels. This configuration imply pairwise penalty and impose the object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

neighbors. We set the potentials from (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 optimising this graph problem the standard $\alpha\beta$-swap Graph Cut algorithm is used.[55]

Binary Graph Cut. As a simplification, 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 Figure 8). The modified unary and binary energy terms are obtained by a restriction of the general formulation (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 Figure 9 and results in Figure 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 15000 images) of microscopy images of Drosophila ovaries, containing egg chambers at various stages of development.[56,57] For 72 images, containing approximately 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 Figure 1(a)) is used.

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

For all experiments we 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

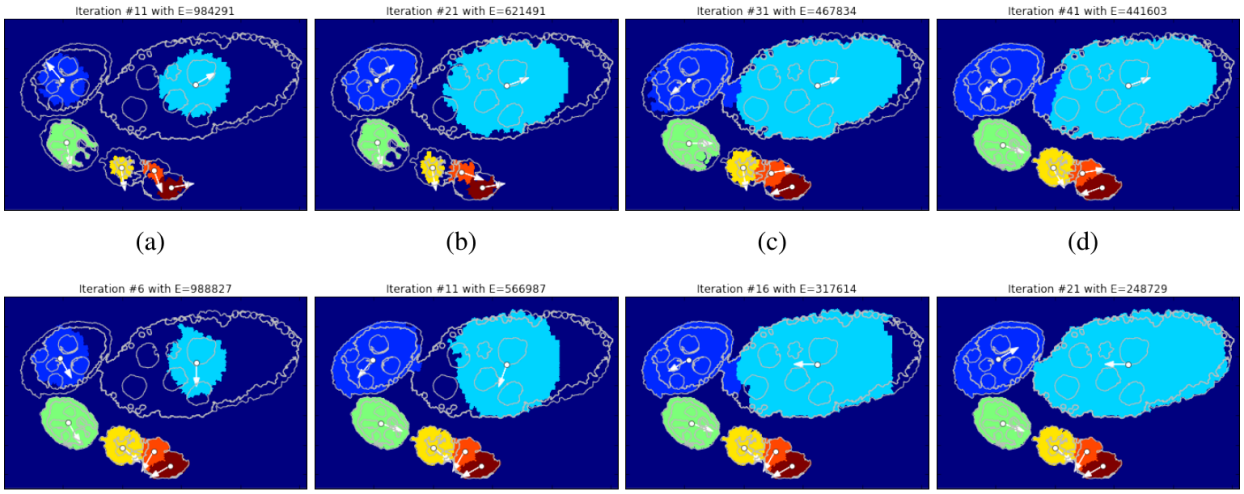


Figure 9 Several iteration steps of the greedy region growing *(a–d)*, and of the multiclass Graph Cut optimisation with object label swapping*(e–h)* algorithms, Each color region corresponds to an individual object k, the thin white contours are levels of the appearance probabilities P_y. White dots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

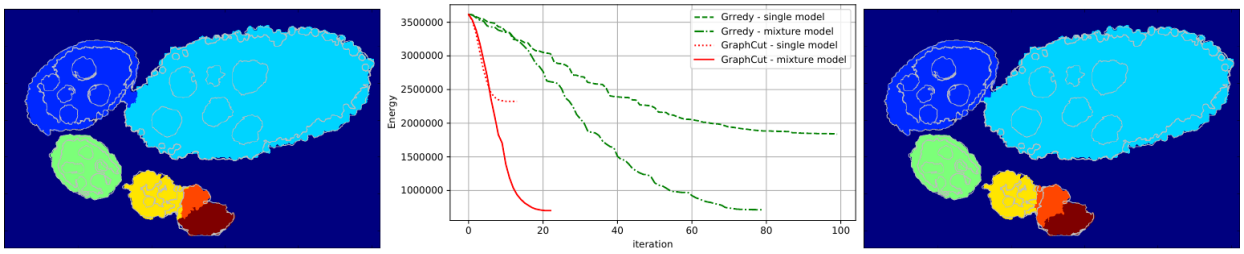


Figure 10 Example of region growing segmentation results applying greedy approach *(a)* and Graph Cuts *(c)*. We also show the energy evolution during iterations *(b)*. Each color region corresponds to an object, the thin white contours are leves of the appearance probabilities P_y.

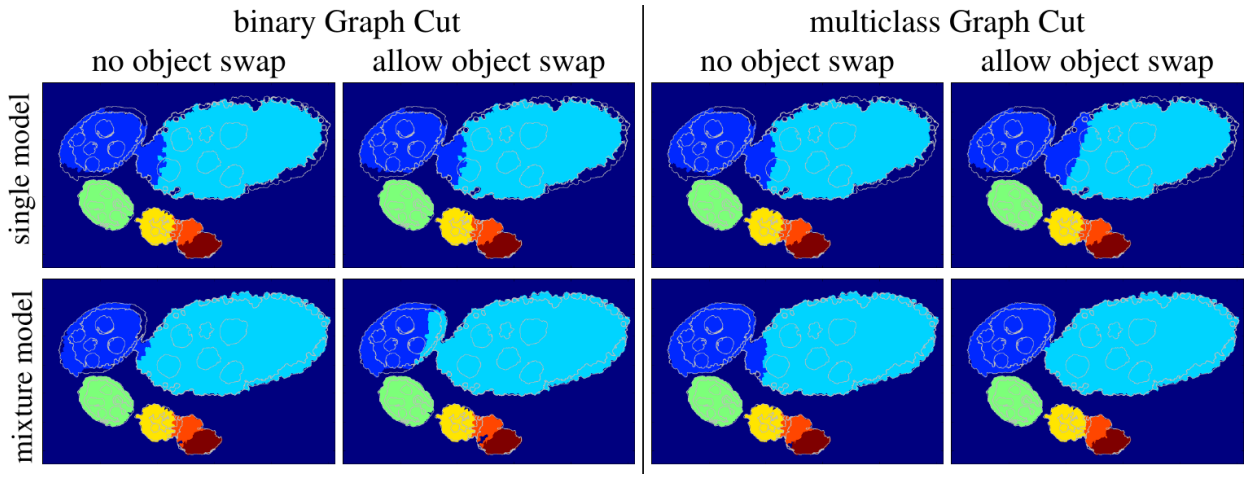


Figure 11 Resulting segmentation for several different variants of our method: single Gaussian model (top row) versus GMM (bottom row), the binary and multiclass Graph Cut on the left and right half respectively. Colored regions represents individual objects and white levels the contours or segmentation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

method	model	obj. 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
GC	single	no	0.6426	0.9317	0.9317	0.9317	0.9317
GC	single	yes	0.6220	0.9284	0.9284	0.9284	0.9284
GC	mixture	no	0.7360	0.9573	0.9573	0.9573	0.9573
GC	mixture	yes	0.7544	0.9568	0.9568	0.9568	0.9568

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

size 1000×1000 pixels was about 1 second. 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 $\gamma = 5$ gives the best results for our images. We set the threshold for a shift to 20 pixels (superpixel size), rotation 10° degrees, and volume change to 5%. These values allow to reach 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 vs greedy approach, GMM (with $M = 15$) vs single Gaussian (assuming GMM with $M = 1$) , allowing object label swapping. Quantitative results are shown in Table 1. It confirms our expectation (see Figure 11) that it is best to use a GMM with 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 are very similar. Speaking about the second criterion - processing time, the Graph Cut is faster in terms of a number of the iterations (see Figure 10), but each iteration is little longer. The total processing time of Graph Cut approach is about 9 seconds compare to Greedy which takes about 72 seconds per image. We experimented

superpixel size		10	15	20	25	30	35	40
greedy	time [s]	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 [s]	94	41	21	9	7	6	5
	Jaccard	0.756	0.755	0.754	0.754	0.753	0.748	0.743

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

	Jaccard	accuracy	F_1 score	precision	recall	time [s]
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

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

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 fewer candidates to evaluate.

We also experimented the dependence of the resulting segmentation on the position of the initial centers $\boldsymbol{c}_k$. 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 Figure 12.

3.2 Baseline methods

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

Watershed segmentation[59–61] 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 centres is then used to identify individual objects. We also tested some morphological operations such as opening, before applying 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.

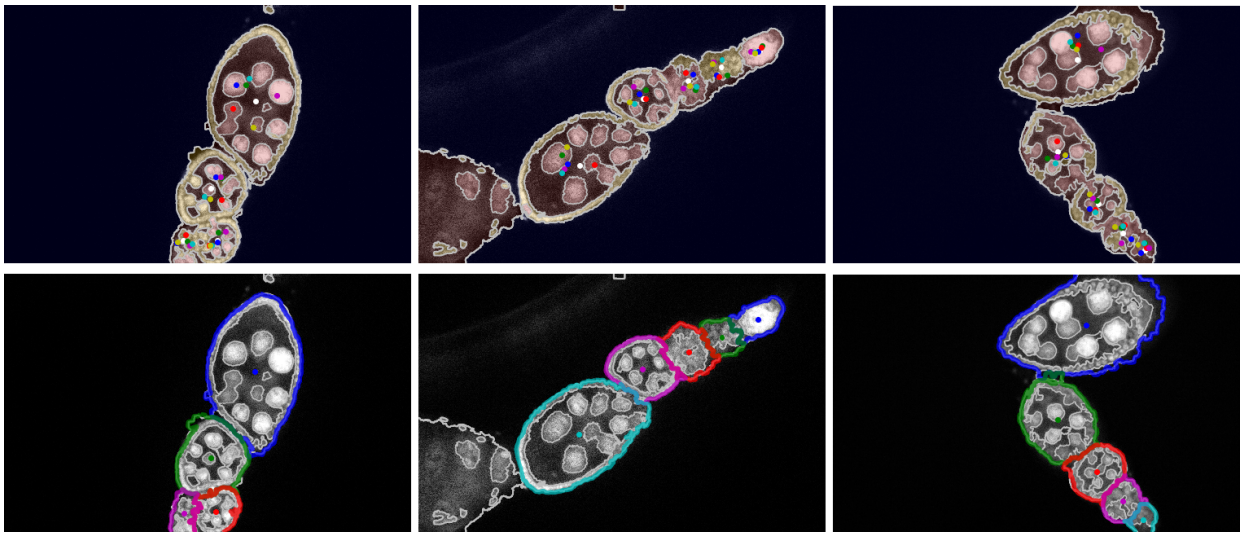


Figure 12 The impact of quality of initial center selection *(top row)* on the final segmentation *(bottom row)*. Each set of initial centres (colored equally for all eggs) was obtained by adding random displacement regarding particular egg size. For all initialization the region growing converged to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

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

Pixel-level Graph Cuts[63] 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, $D_i(k) = -\log P_y\big(k \mid y(s(i))\big)$ for all pixels $i \in \Omega$ and standard pairwise regularization for Potts model. Pixels from a small region around the provided initial object centers $\boldsymbol{c}_k$ are forced to class k.

Superpixel-level Graph Cut[24] works similarly as above except that we assign classes k to superpixels, not pixels. The energy function $E(g)$ from (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. The Table 3 presents the quantitative results. We can say that proposed method performed better than the other methods in all comparable metrics.

Example segmentation results are shown in Figures 14. We can see that the comparable methods usually fail to properly distinguish touching eggs. Also, they are frequently merging two eggs

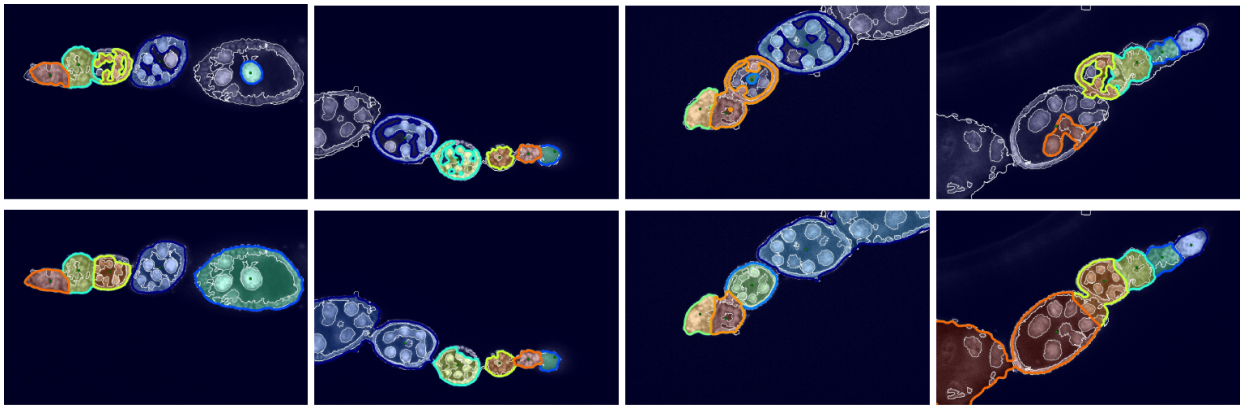


Figure 13 Examples of resulting segmentation using Morphological snakes on input images directly *(top row)* and on appearance probabilities P_y *(bottom row)*. The manual annotation for these images is presented in Figure 14.

together even if the second egg is not contained an initial seed which can happen in real-world application.[8]

4 Conclusion

We presented a new region growing segmentation technique. It is fast thanks to using superpixels, and it is also robust thanks to handling the growing with 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.

Source code

The implementation of the proposed method together with other tested methods will be available on `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*, Cengage Learning, 3 ed. (2007).

2 D. L. Pham, C. Xu, and J. L. Prince, "A Survey of Current Methods in Medical Image Segmentation," *In Annual Review of Biomedical Engineering* **2**, 315–338 (2000).

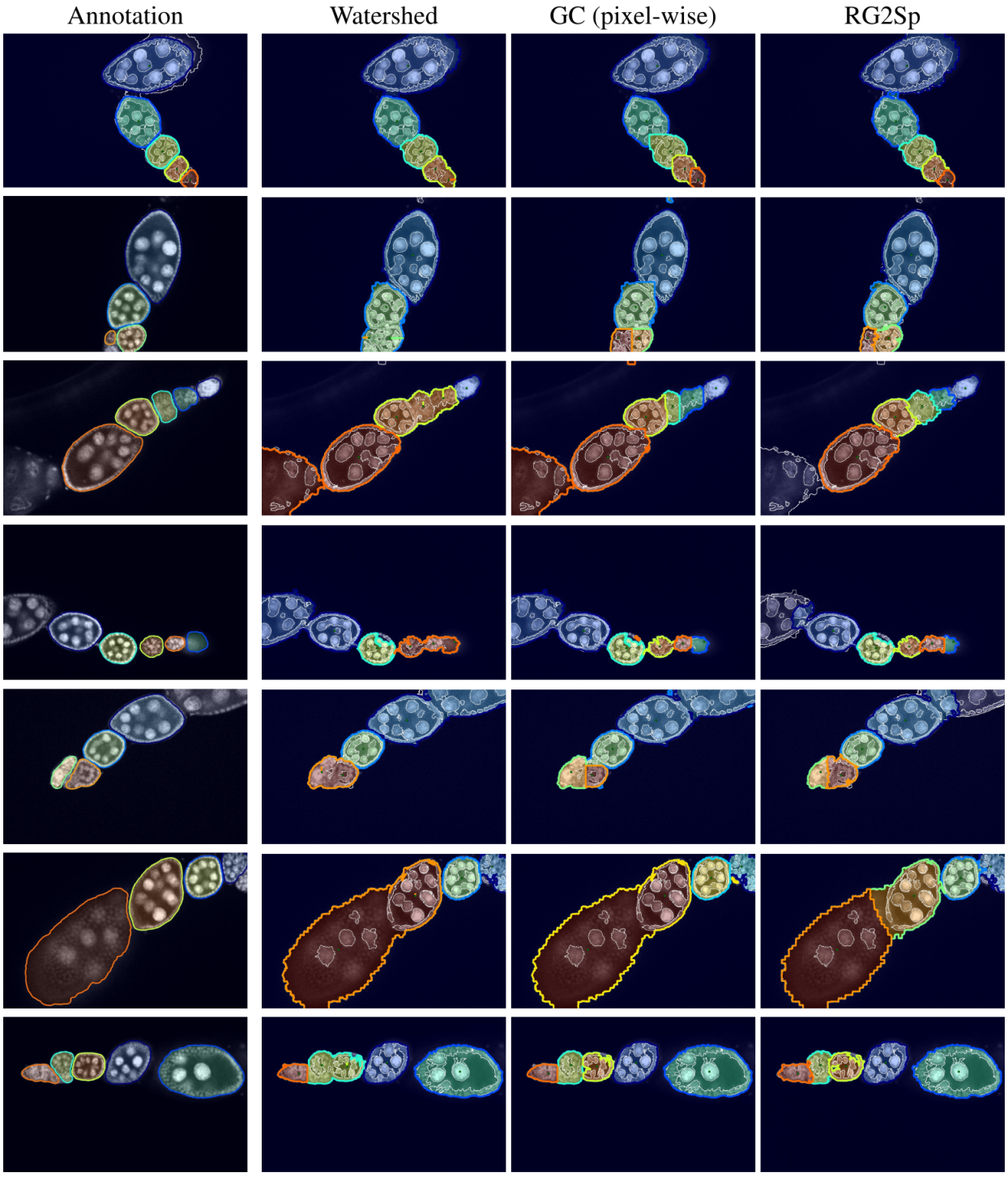


Figure 14 Each row represents a microscopy image segmented by an expert ('Annotation') and the three automatic methods — from left to right: watershed, Graph Cut on pixels and region growing. The expert annotation is shown overlaid on the input image. The segmentation result are shown overlaid over the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

3 K.-P. Wong, *Medical Image Segmentation: Methods and Applications in Functional Imaging*, Springer (2005).

4 A. Elnakib, G. Gimel'farb, J. Suri, *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, R. Acharya, A. F. Laine, *et al.*, Eds., 1–39, Springer New York (2011).

5 P. Tomancak, A. Beaton, R. Weiszmann, *et al.*, "Systematic determination of patterns of gene expression during Drosophila embryogenesis.," *Genome biology* **3**(12), RESEARCH0088 (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*, 555–569 (2016).

7 R. Nava and J. Kybic, "Supertexton-based segmentation in early Drosophila oogenesis," in *Proceedings - International Conference on Image Processing, ICIP*, **2015-Decem**, 2656–2659 (2015).

8 J. Borovec, J. Kybic, and R. Nava, "Detection and localization of Drosophila egg chambers in microscopy images," in *8th International Workshop on Machine Learning in Medical Imaging*, Springer, (Quebec) (2017).

9 S. W. Zucker, "Region growing: Childhood and adolescence.," *Computer Graphics and Image Processing* **3**(5), 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, T. Grenier, J. L. Rose, *et al.*, "Region growing: When simplicity meets theory - region growing revisited in feature space and variational framework," in *Communications in Computer and Information Science*, **359 CCIS**, 426–444 (2013).

12 R. Achanta and A. Shaji, "SLIC Superpixels Compared to State-of-the-art Superpixel Methods," *Pattern Analysis and Machine Intelligence, IEEE* **34**(11), 2274 – 2282 (2012).

13 D. Stutz, A. Hermans, and B. Leibe, "Superpixels: An Evaluation of the State-of-the-Art," *Computer Vision and Image Understanding* **abs/1612.0**(35), – (2016).

14 P. Buyssens, I. Gardin, S. Ruan, *et al.*, "Eikonal-based region growing for efficient clustering," *Image and Vision Computing* **32**(12), 1045 – 1054 (2014).

15 K. Smith and A. Carleton, "Fast ray features for learning irregular shapes," in *Computer Vision*, 397 – 404 (2009).

16 K.-M. Lee and W. N. Street, "Learning shapes for automatic image segmentation," in *Proc. INFORMS-KORMS Conference*, 1461–1468 (2000).

17 J. L. Rose, C. Revol-Muller, M. Almajdub, *et al.*, "Shape prior integrated in an automated 3D region growing method," in *Proceedings - International Conference on Image Processing, ICIP*, **1** (2007).

18 J. L. Rose, C. Revol-Muller, J. B. Langlois, *et al.*, "3D region growing integrating adaptive shape prior," in *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Proceedings, ISBI*, 967–970 (2008).

19 J. L. Rose, C. Revol-Muller, D. Charpigny, *et al.*, "Shape prior criterion based on tchebichef moments in variational region growing," in *Proceedings - International Conference on Image Processing, ICIP*, 1081–1084 (2009).

20 A. Quispe and C. Petitjean, "Shape prior based image segmentation using manifold learning," in *5th International Conference on Image Processing, Theory, Tools and Applications 2015, IPTA 2015*, 137–142 (2015).

21 P. Etyngier, F. Segonne, and R. Keriven, "Shape Priors using Manifold Learning Techniques," in *2007 IEEE 11th International Conference on Computer Vision*, 1–8 (2007).

22 O. Moolan-Feroze, M. Mirmehdi, M. Hamilton, *et al.*, "Segmentation of the right ventricle using diffusion maps and Markov random fields," in *Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*, **8673 LNCS**(PART 1), 682–689 (2014).

23 Y. Li, J. Sun, C.-K. Tang, *et al.*, "Lazy snapping," *ACM Trans. Graph.* **23**, 303–308 (2004).

24 X. Ye, G. Beddoe, and G. Slabaugh, "Automatic graph cut segmentation of lesions in CT using mean shift superpixels," *International Journal of Biomedical Imaging* **2010** (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 Transactions on Image Processing* **19**, 2157 – 2170 (2010).

27 M. P. Kumar, P. H. S. Torr, and A. Zisserman, "OBJ CUT," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego*, **1**, 18–25 (2005).

28 D. Freedman and T. Zhang, "Interactive graph cut based segmentation with shape priors," in *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, **I**, 755–762 (2005).

29 N. Vu and B. S. Manjunath, "Shape prior segmentation of multiple objects with graph cuts," in *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, (2008).

30 O. Veksler, "Star Shape Prior for Graph-Cut Image Segmentation," in *Computer Vision–ECCV*, 454—-467, Springer (2008).

31 K. Nakagomi, A. Shimizu, H. Kobatake, *et al.*, "Multi-shape graph cuts with neighbor prior constraints and its application to lung segmentation from a chest CT volume," *Medical Image Analysis* **17**(1), 62–77 (2013).

32 T. Schoenemann and D. Cremers, "Globally optimal image segmentation with an elastic shape prior," in *Proceedings of the IEEE International Conference on Computer Vision*, (2007).

33 A. Delong and Y. Boykov, "Globally optimal segmentation of multi-region objects," in *Proceedings of the IEEE International Conference on Computer Vision*, 285–292 (2009).

34 J. Ulen, P. Strandmark, and F. Kahl, "An efficient optimization framework for multi-region segmentation based on lagrangian duality," *IEEE Transactions on Medical Imaging* **32**(2), 178–188 (2013).

35 H. N. Isack, O. Veksler, I. Oguz, *et al.*, "Efficient optimization for hierarchically-structured interacting segments (hints).," in *Proceedings of CVPR*, (2017).

36 H. Lu, Y. Li, Y. Wang, *et al.*, "Active Contours Model for Image Segmentation: A Review," in *The Proceedings of the 1st International Conference on Industrial Application Engineering 2013*, 104–111 (2013).

37 K. Zhang, L. Zhang, H. Song, *et al.*, "Active contours with selective local or global segmentation: A new formulation and level set method," *Image and Vision Computing* **28**(4), 668–676 (2010).

38 O. Dzyubachyk, W. A. van Cappellen, J. Essers, *et al.*, "Advanced Level-Set-Based Cell Tracking in Time-Lapse Fluorescence Microscopy," *IEEE Transactions on Medical Imaging* **29**(3), 1–16 (2010).

39 L. Vese and T. Chan, "A multiphase level set framework for image segmentation using the Mumford and Shah model," *International Journal of Computer Vision* **50**(3), 271–293 (2002).

40 B. C. Lucas, M. Kazhdan, and R. H. Taylor, "Multi-object geodesic active contours (mogac)," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012: 15th International Conference, Nice, France, October 1-5, 2012, Proceedings, Part II*, N. Ayache, H. Delingette, P. Golland, *et al.*, Eds., 404–412, Springer Berlin Heidelberg, (Berlin, Heidelberg) (2012).

41 N. Paragios and R. Deriche, "Coupled geodesic active regions for image segmentation: A level set approach," in *Computer Vision — ECCV 2000: 6th European Conference on Computer Vision Dublin, Ireland, June 26–July 1, 2000 Proceedings, Part II*, D. Vernon, Ed., 224–240, Springer Berlin Heidelberg, (Berlin, Heidelberg) (2000).

42 T. F. Cootes, C. J. Taylor, D. H. Cooper, *et al.*, "Active shape models—their training and application," *Comput. Vis. Image Underst.* **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 *CVPR*, 1316–1323, IEEE Computer Society (2000).

44 A. Tsai, A. J. Yezzi, W. M. W. III, *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 Techn.* **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 *International Conference on Pattern Recognition*, (ICPR), 1427–1430 (2012).

47 C. Molnar, I. H. Jermyn, Z. Kato, *et al.*, "Accurate Morphology Preserving Segmentation of Overlapping Cells based on Active Contours," *Scientific Reports* **6**(1), 1–10 (2016).

48 Y. Shi and W. C. Karl, "Real-time tracking using level sets," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on* **2**, 34–41 (2005).

49 J. Kybic and J. Krátký, "Discrete curvature calculation for fast level set segmentation," in *ICIP: International Conference on Image Processing*, 4404, Institute of Electrical and Electronics Engineers, (445 Hoes Lane, Piscataway, NJ, U.S.A.) (2009). electronic version.

50 O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III*, N. Navab, J. Hornegger, W. M. Wells, *et al.*, Eds., 234–241, Springer International Publishing, (Cham) (2015).

51 B. Romera-Paredes and P. H. S. Torr, "Recurrent instance segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, **9910 LNCS**, 312–329 (2016).

52 J. Borovec and J. Kybic, "jSLIC : superpixels in ImageJ," in *Computer Vision Winter Workshop*, Z. Kunbelova and J. Heller, Eds., 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," *Medical Imaging, IEEE* **31**(2), 474 – 486 (2012).
54 G. Xuan and W. Zhang, "EM algorithms of Gaussian mixture model and hidden Markov model," *Image Processing, 2001.* **1**, 145–148 (2001).
55 Y. Boykov and O. Veksler, "Fast approximate energy minimization via graph cuts," *Pattern Analysis and Machine Intelligence, IEEE* **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, Q. Xu, Q. Xie, *et al.*, "Automatic stage identification of Drosophila egg chamber based on DAPI images," *Scientific Reports* **6**(November 2015), 18850 (2016).
58 O. Koyejo, P. Ravikumar, N. Natarajan, *et al.*, "Consistent Multilabel Classification," in *Advances in Neural Information Processing Systems*, 3321—-3329 (2015).
59 S. Beucher, "The Watershed Transformation Applied to Image Segmentation," in *Proceedings of the 10th Pfefferkorn Conference on Signal and Image Processing in Microscopy and Microanalysis*, (March), 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 *Proceedings. Second International Conference on Creating, Connecting and Collaborating through Computing*, **2**, 111–116 (2004).
61 X. Ji, Y. Li, J. Cheng, *et al.*, "Cell image segmentation based on an improved watershed algorithm," in *Proceedings - 2015 8th International Congress on Image and Signal Processing, CISP 2015*, 433–437 (2016).
62 P. Marquez-Neila, L. Baumela, and L. Alvarez, "A morphological approach to curvature-based evolution of curves and surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36**(1), 2–17 (2013).
63 Y. Boykov, "Graph cuts and efficient nd image segmentation," *International Journal of Computer Vision* **70**, 109–131 (2006).

Biography

Jiří Borovec was awarded an MSc double-degree in intelligent systems by Université Paul Sabatier (UPS), France, and the Technical University of Liberec (TUL), Czechia in 2011. He is now pursuing a PhD degree 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 degree from the Czech Technical University in Prague, Czech Republic and a Ph.D. from EPFL, Switzerland, in 1998 and 2001, respectively. He held a postdoc position at INRIA, France, in 2002–2003. Since 2003 he is at Czech Technical University in Prague, becoming a full Professor in 2015 and currently serving as a research group leader.

Akihiro Sugimoto received a Ph.D. in mathematical engineering from the University of Tokyo, Japan. After working for Hitachi, ATR, and Kyoto University, he joined in 2002 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.