Image segmentation & BPDL in microscopy

Jiri Borovec
@ CMP, FEE, CTU in Prague

15.03.2018
https://www.researchgate.net/publication/323120618
Resources

- **Publications:**
Overview

1. Problem statement
2. Used datasets
3. Instance segmentation
   a. Structure (tissue) segmentation on superpixels
   b. Center detection and ellipse fitting
   c. Region growing on superpixels
4. Binary Pattern Dictionary Learning
5. Conclusion & Future work
Motivation

Why Drosophila?

- High gene similarity with mammals (~ 93%)
- Short life cycles -> gene evaluation in generations

Automatic image processing:

- Analyses require thousands of images to be processed
- Very time consuming for experts
Schema of Drosophila development

(a) Embryo
- cellularization, gastrulation, germ band extension, dorsal closure, myoblast fusion

(b) Larva
- CNS, PNS, NMJs, imaginal discs, salivary gland

(c) Pupa
- imaginal discs, SOP, retina, histoblast

(d) Adult
- eye, brain, heart, testis, ovary

(e) 3rd instar larva
- discs
- Antenna
- Eye
- Leg
- Wing
- Haltere
- Genital

(f) ovary
- ovarirole
- follicle cells
- stalk cells
- germarium
- stage 3-5
- stage 8
- stage 9
- stage 10
- stage 14

(g) Anterior → Posterior
Sample images

Drosophila ovaries

Gene activations

Imaginal disk

Tissue
Drosophila Datasets

Just a few annotations in biomedical imaging is common issue…

- Noisy images with local deformations
- Low image variance (patterns)

<table>
<thead>
<tr>
<th></th>
<th>Domain</th>
<th>Semantic annot.</th>
<th>Instance annot.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ovaries</td>
<td>2.5D</td>
<td>72</td>
<td>250</td>
</tr>
<tr>
<td>Imaginal discs</td>
<td>2D</td>
<td>15</td>
<td>-</td>
</tr>
</tbody>
</table>
Drosophila ovary - semantic segmentation
Drosophila imaginal discs - semantic segmentation
Notations

Image related

Ω  set of pixels (image plane)
I  input image function $I : \Omega \rightarrow \mathbb{R}^m$
L  set of labels

Superpixels

η  original regularization
ξ  proposed regularization
ν  initial superpixel size
s  superpixel
S  set of superpixels $s \in S$
D  weighted distance
d_{(M,E,T)}  Manhattan, Euclidean and Tchebychev distances
Ω_s  pixels belonging to a superpixel $s$, $\Omega_s \subseteq \Omega$

Segmentation & Region growing

$y_\Omega$  pixel-wise segmentation function $y_\Omega : \Omega \rightarrow \mathcal{L}$
$Y_\Omega$  ordered set of pixel-wise segmentation $Y_\Omega = y_\Omega (\Omega)$
y  superpixel segmentation function $y : S \rightarrow \mathcal{L}$ with abbrev. for $y_s = y(s)$
Y  ordered set of superpixel segmentation $Y = y(S)$
x  feature vector
X  set of features $x_s \in X$ for all superpixels
d_{(M,E,T)}  Manhattan, Euclidean and Tchebychev distances
U, B  unary (data) and binary (pairwise) term for GC respectively
g  image descriptor
r  vector of ray distances
c  vector of object centers
m  statistical shape model $m = [c, r, \Theta, w]$
M  mixture of shape model
w  vector with model weights
q  cumulative probability of spatial prior

Dictionary Learning

g  image appearance (binary association to a class)
G  set of images $g \in G$
y_\Omega, Y_\Omega  atlas (binary patterns, segmentation)
w  vector with binary weights
W  matrix with binary weights $w \in W$
Superpixels - Imaginal discs

<table>
<thead>
<tr>
<th></th>
<th>regularization $\xi = 0.1$</th>
<th>regularization $\xi = 0.2$</th>
<th>regularization $\xi = 0.3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>size $v = 30$ [px]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>size $v = 60$ [px]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
</tbody>
</table>

Too small & broken

Not reflecting object boundary
## Superpixels - Drosophila ovary

<table>
<thead>
<tr>
<th>size $v$</th>
<th>regularization $\xi = 0.1$</th>
<th>regularization $\xi = 0.3$</th>
<th>regularization $\xi = 0.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v = 30$</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
<tr>
<td>$v = 60$</td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
</tbody>
</table>
Overview

1. Problem statement
2. Used datasets
3. Instance segmentation
   a. Structure (tissue) segmentation on superpixels
   b. Center detection and ellipse fitting
   c. Region growing on superpixels
4. Binary Pattern Dictionary Learning
5. Conclusion & Future work
Supervised and unsupervised segmentation using superpixels, model estimation, and Graph Cut

Annotation

Training classifier

Labeling
(major vote)

Labels (superpixels)

Classifier

Model estimation

Input image

Superpixels

SLIC

Feature extraction

Color features, $x^c$

Texture features, $x^t$

GMM

Soft predict.

Edge weights, $w_{ij}$

Neighborhood relations

Segmentation

Unary potentials, $U$

Graph Cut

Graph

$p(x|y = k)$
Segmentation method overview

Image segmentation method consisting of the following steps:

1. Computation of superpixels - SLIC
2. Extraction of superpixel-based descriptors:
   a. Color - mean, median, energy, STD
   b. Texture - Leung-Malik filter bank
3. Calculating image-based class probabilities:
   a. Supervised - Random Forest, k-NN, Adaboost, ...
   b. Unsupervised - Gaussian Mixture Model
4. Spatial regularized superpixel classification using Graph Cut
   a. Edge weights - color, features, model
Problem formulation

Formulation (standard) as

\[ Y^* = \arg \max_Y P(Y|X) = \arg \max_Y \frac{p(X|Y) \cdot P(Y)}{p(X)} \]

\[ P(Y) = \prod_{s \in S} h(y_s) \cdot \prod_{(i,j) \in N \subseteq S^2} R(y_i, y_j) \]

\[ Y^* = \arg \max_Y \prod_{i \in S} (p(x_i|y_i) \cdot h(y_i)) \cdot \prod_{(i,j) \in N} R(y_i, y_j) \]

Energy minimisation

\[ Y^* = \arg \min_Y \sum_s - \log \left( p(x_s|y_s) \cdot h(y_s) \right) \cdot \frac{1}{U_s(y_s)} + \sum_{(i,j) \in N} - \log R(y_i, y_j) \cdot \beta_{w_{i,j}} B(y_i, y_j) \]
Superpixel features

input image  $N = 15$ pixels  $N = 30$ pixels  $N = 45$ pixels

Superpixel color means

Leven-Malik filter bank
Influence of superpixel parameters

Low feature representativeness

Optimum (~compromise)

Low separability
Graph Cut - Edge weights

Spatial

Color

Features

Models

weighted edges

segmentation
Segmentation results with SOA (F1-score)

<table>
<thead>
<tr>
<th>Method</th>
<th>imaginal disc</th>
<th>ovary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel-wise</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supervised</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weka</td>
<td>0.6923</td>
<td>0.5800</td>
</tr>
<tr>
<td>Weka &amp; GC(0, 100)</td>
<td>0.6887</td>
<td>0.5810</td>
</tr>
<tr>
<td>Weka &amp; GC(1, 50)</td>
<td>0.6887</td>
<td>0.5965</td>
</tr>
<tr>
<td>Weka &amp; GC(10, 50)</td>
<td>0.6887</td>
<td>0.1395</td>
</tr>
<tr>
<td>Weka &amp; GC(50, 100)</td>
<td>0.6850</td>
<td>0.6007</td>
</tr>
<tr>
<td>ideal segm. $Y_A$</td>
<td>0.9696</td>
<td>0.9067</td>
</tr>
<tr>
<td>Supertextons</td>
<td>-</td>
<td>0.7488</td>
</tr>
<tr>
<td>our RF</td>
<td>0.8181</td>
<td>0.8201</td>
</tr>
<tr>
<td>our RF &amp; GC</td>
<td><strong>0.8229</strong></td>
<td><strong>0.8600</strong></td>
</tr>
<tr>
<td>Superpixels</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unsuper.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>our GMM</td>
<td>0.7542</td>
<td>0.5967</td>
</tr>
<tr>
<td>our GMM &amp; GC</td>
<td>0.7644</td>
<td>0.6039</td>
</tr>
<tr>
<td>our GMM [gr]</td>
<td>0.7301</td>
<td>0.6009</td>
</tr>
<tr>
<td>our GMM [gr] &amp; GC</td>
<td>0.7564</td>
<td>0.6083</td>
</tr>
</tbody>
</table>
Advantage of using Graph Cut
Supervised vs Unsupervised
Overview

1. Problem statement
2. Used datasets
3. Instance segmentation
   a. Structure (tissue) segmentation on superpixels
   b. Center detection and ellipse fitting
   c. Region growing on superpixels
4. Binary Pattern Dictionary Learning
5. Conclusion & Future work
Detection and localization of Drosophila egg chambers in microscopy images

Center detections - illustration
Schema

1. Extract pixel features
   a. Label histogram
   b. Ray features

2. Train classifier

3. Group center candidates

4. Ellipse fitting
Features for center detection

- Label histogram

- Ray features
Classification & Grouping

Train classifier on 3 zones

- Negative
- No not case
- Positive

Clustering of center candidates with DBSCAN
Ellipse fitting

Fitting ellipses to boundary points to maximise foreground labels inside ellipse
Ellipse fitting

Formulate as Maximize likelihood

\[ h_\bullet = - \log P_\bullet \]

\[ \sum_{i \in \Omega} h_B(Y_i) = \sum_{i \in \Omega_F} h_B(Y_i) + \sum_{i \in \Omega \setminus \Omega_F} h_B(Y_i) \]

\[ \begin{aligned}
\arg\max_{F,B} & \prod_{i \in \Omega_F} P_F(Y_i) \cdot \prod_{i \in \Omega \setminus \Omega_F} P_B(y_i) \\
\log & \arg\max_{F,B} \sum_{i \in \Omega_F} h_F(Y_i) + \sum_{i \in \Omega} h_B(Y_i) - \sum_{i \in \Omega_F} h_B(Y_i) \\
\sum_{i \in \Omega} h_B(Y_i) & \text{ is constant} \\
\log & \arg\min_{F,B} \sum_{i \in \Omega_F} h_F(Y_i) - h_B(Y_i) 
\end{aligned} \]
Ellipse pruning

Ellipse approximation eliminate multiple center detection in single egg

<table>
<thead>
<tr>
<th>Egg chambers</th>
<th>Stage 1</th>
<th>Stage 2</th>
<th>Stage 3</th>
<th>Stage 4</th>
<th>Stage 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>number</td>
<td>921</td>
<td>1403</td>
<td>865</td>
<td>834</td>
<td>836</td>
</tr>
<tr>
<td>false negatives</td>
<td>306 (33%)</td>
<td>158 (11%)</td>
<td>6 (0.7%)</td>
<td>1 (0.1%)</td>
<td>0 (0.0%)</td>
</tr>
<tr>
<td>multiple detections (MD)</td>
<td>37 (4.0%)</td>
<td>31 (2.2%)</td>
<td>109 (12%)</td>
<td>80 (9.6%)</td>
<td>90 (11%)</td>
</tr>
<tr>
<td>MD after ellipse fitting</td>
<td>18 (2.0%)</td>
<td>13 (0.9%)</td>
<td>27 (3.1%)</td>
<td>20 (2.4%)</td>
<td>30 (3.6%)</td>
</tr>
<tr>
<td>false positives</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>43 (0.9%)</td>
</tr>
</tbody>
</table>
Overview

1. Problem statement
2. Used datasets
3. Instance segmentation
   a. Structure (tissue) segmentation on superpixels
   b. Center detection and ellipse fitting
   c. Region growing on superpixels
4. Binary Pattern Dictionary Learning
5. Conclusion & Future work
Region growing - variational framework

Formulated as:

\[ P(y_s \mid y, M) = \frac{1}{Z(M, y)} P_g(g \mid y) P_m(g \mid M) P_R(g) \]

- **Appearance model**: \( P_g(g \mid y) \)
- **Pairwise term**: \( P_R(g) = \prod_{(u,v) \in \mathcal{N}_S} H(y(u), y(v)) \)
- **Mixture model**: \( P_m(g \mid M) \)

![Region growing - variational framework](image_url)
Region growing - variational framework

Where:

\[
P_g(g \mid y) = \prod_{i \in \Omega} P_g\left(g(s(i)) \mid y(s(i))\right) = \prod_{s \in S} P_g(y_s \mid y(s))_{|\Omega_s|}
\]

\[
P_m(g \mid M) = \prod_{i \in \Omega} P_m\left(g(s(i)) \mid M\right) = \prod_{s \in S} P_m(y_s \mid M)_{|\Omega_s|}
\]

\[
P_R(g) = \prod_{(u, v) \in \mathcal{N}_S} H(y(u), y(v))
\]

Resolves in energy minimisation:

\[
E'(g) = \sum_{s \in S} |\Omega_s| \left[D_s(y_s) + \beta V_s(y_s)\right] + \gamma \sum_{(u, v) \in \mathcal{N}_S} B(y(u), y(v))
\]

38
Appearance model

Associating a probability for each pixel (superpixel) whether it belongs to an object or not

\[
P_g(y(s)|g_s) = \begin{cases} 
P_g(g_s) & \text{for } y(s) \neq 0 \\
1 - P_g(g_s) & \text{for } y(s) = 0 
\end{cases}
\]
Measured shapes

Measured egg shapes as ray features from training examples (~250 eggs)
Shape model & prior

Gaussian model from all measured Ray features

Prior is represented as integral of probabilities

\[ q(s, m_k) = \int_{\delta}^{\infty} \rho(r) \, dr = 1 - \int_{0}^{\delta} \rho(r) \, dr \]
Mixture of Gaussian models

\[ P_m(y_s = k \mid M) = \begin{cases} q(s, m_k) & \text{for } k > 0 \\ \Pi_l (1 - q(s, m_l)) & \text{for } k = 0 \end{cases} \]
RG - optimisation

- Iterative approach on object boundaries
- Alternating: region growing & update shape prior
- Strategies:
  - Greedy
  - Graph Cut
    - Binary
    - Multi-class
  - Object swapping
Algorithm 1: Region growing.

Input: $S$: superpixels, $g$: superpixel descriptors, $c_k$: initial object centers, $M$: mixture of statistical shape models

Output: object segmentation $\mathbf{y}$

1. compute data cost $D$;
2. from object centers $c_k$ set initial segmentation $\mathbf{y}$ and model shape parameters $m_k$;
3. compute shape cost $V$;
4. while not converged do
   5. update object pose parameters $c_k$ and $\Theta_k$;
   6. if significant change of center $c_k$ position, orientation $\Theta_k$ and object area then
      7. update remaining object shape parameters $m_k$;
      8. update shape costs $V$ for all $s$ close to $k$;
   end
5. find superpixels $\partial S_k$ on the external object boundary of $k$;
6. optimize (7.10) wrt $\mathbf{y}$ by changing $s \in \partial S_k$ using the greedy or Graph Cut algorithms;
end
Result compared to SOA

High Jaccard index with reasonable processing time

<table>
<thead>
<tr>
<th>Method</th>
<th>Jaccard</th>
<th>Accuracy</th>
<th>$F_1$ score</th>
<th>Precision</th>
<th>Recall</th>
<th>Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Watershed</td>
<td>0.5705</td>
<td>0.9246</td>
<td>0.9246</td>
<td>0.9246</td>
<td>0.9246</td>
<td>5</td>
</tr>
<tr>
<td>Watershed (w. morph.)</td>
<td>0.5705</td>
<td>0.9270</td>
<td>0.9198</td>
<td>0.9136</td>
<td>0.9327</td>
<td>7</td>
</tr>
<tr>
<td>Morph. snakes (image)</td>
<td>0.4251</td>
<td>0.8769</td>
<td>0.8070</td>
<td>0.9053</td>
<td>0.7987</td>
<td>784</td>
</tr>
<tr>
<td>Morph. snakes ($P_y$)</td>
<td>0.6494</td>
<td>0.8812</td>
<td>0.8812</td>
<td>0.8812</td>
<td>0.8812</td>
<td>968</td>
</tr>
<tr>
<td>Graph Cut (pixel-level)</td>
<td>0.7143</td>
<td>0.9204</td>
<td>0.9204</td>
<td>0.9204</td>
<td>0.9204</td>
<td>15</td>
</tr>
<tr>
<td>Graph Cut (superpixels)</td>
<td>0.3164</td>
<td>0.8643</td>
<td>0.8643</td>
<td>0.8643</td>
<td>0.8643</td>
<td>3</td>
</tr>
<tr>
<td>RG2Sp (greedy)</td>
<td>0.7527</td>
<td>0.9577</td>
<td>0.9577</td>
<td>0.9577</td>
<td>0.9577</td>
<td>72</td>
</tr>
<tr>
<td>RG2Sp (Graph Cut)</td>
<td>0.7544</td>
<td>0.9568</td>
<td>0.9568</td>
<td>0.9568</td>
<td>0.9568</td>
<td>9</td>
</tr>
</tbody>
</table>
Overview

1. Problem statement
2. Used datasets
3. Instance segmentation
   a. Structure (tissue) segmentation on superpixels
   b. Center detection and ellipse fitting
   c. Region growing on superpixels
4. Binary Pattern Dictionary Learning
5. Conclusion & Future work
Binary Pattern Dictionary Learning of gene expressions

Pipeline: segmentation - registration - BPDL

Input images → Gene expressions → Imaginal discs → Segmented images

Gene expressions: Transform

Aligned expressions: Input binary images → Atlas

Weights:

<table>
<thead>
<tr>
<th></th>
<th>P_1</th>
<th>P_2</th>
<th>P_3</th>
<th>...</th>
<th>P_K</th>
</tr>
</thead>
<tbody>
<tr>
<td>img_1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>img_2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>img_N</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>...</td>
<td>1</td>
</tr>
</tbody>
</table>

Segmentation → Registration → BPDL
Decomposition methods

Formulation

$$\min_{Y,W} \|X - Y \cdot W\|^2$$

- Standard approaches
  - Non Negative Matrix Factorisation
  - Fast Independent Component Analysis
  - Sparse Principal Component Analyses
  - Dictionary Learning with Matching pursuit
Formulation

- Image representation

\[ \hat{g} = \sum_{l \in L} w_l \cdot [y = l] \]

- Similarity measure (Hamming distance)

\[ F(g, y, w) = \sum_{i \in \Omega} [g_i \neq \hat{g}_i] \]

- Regularize neighbouring pixels

\[ H(y) = \sum_{i,j \in \Omega, \ i \neq j, \ d(i,j) = 1} [y_i \neq y_j] \]

- Optimization criterion

\[ y^*, w^* = \arg \min_{y, w} \frac{1}{N} \sum_n F(g^n, y, w^n) + \beta \cdot H(y) \]
Alternating minimization

- Update weights - maximise overlap

\[ w_l = \left[ P(g, y, l) \geq \sigma \right] \quad \text{where } \sigma = 1 \]

and \( P(g, y, l) = \frac{\sum_{i \in \Omega, y_i = l} [g_i = 1]}{\sum_{i \in \Omega, y_i = l} [g_i \neq 1]} = \frac{||[y = l]||}{\sum_{i \in \Omega, y_i = l} (1 - g_i)} - 1 \)

- Update atlas

\[
\frac{1}{N} \sum_{i \in \Omega} \sum_{n} \left| g_i^s - \sum_{l \in L} w_l^s \cdot [y = l] \right| + \sum_{i,j \in \Omega, i \neq j, \ d(i,j)=1} [y_i \neq y_j]
\]

\[ U_i(y_i) \]
Algorithm

Algorithm 1 General schema of BPDL algorithm.

1: initialise atlas $y$
2: while not converged do
3: update weights $w \in W$
4: reinitialise empty patterns in $y^*$
5: update atlas $y^*$ via Graph Cut
6: end while

*It makes the algorithm more robust to initialisation.*
Synthetic datasets
Synthetic images

Simulates real images
Comparison on synth. images

<table>
<thead>
<tr>
<th>datasets</th>
<th>NMF</th>
<th>FastICA</th>
<th>sPCA</th>
<th>DL</th>
<th>BPDL</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>(size $64 \times 64$ px, 13 patterns)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pure</td>
<td>ARS</td>
<td>1.0</td>
<td>1.0</td>
<td>0.992</td>
<td>0.995</td>
</tr>
<tr>
<td></td>
<td>diff.</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0298</td>
<td>0.019</td>
</tr>
<tr>
<td></td>
<td>time</td>
<td>2.333</td>
<td>340.32</td>
<td>18.291</td>
<td>737.47</td>
</tr>
<tr>
<td>deform</td>
<td>ARS</td>
<td>0.785</td>
<td>0.948</td>
<td>0.780</td>
<td>0.779</td>
</tr>
<tr>
<td></td>
<td>diff.</td>
<td>0.017</td>
<td>0.004</td>
<td>0.029</td>
<td>0.033</td>
</tr>
<tr>
<td></td>
<td>time [s]</td>
<td>4.001</td>
<td>312.18</td>
<td>15.000</td>
<td>700.03</td>
</tr>
<tr>
<td>D&amp;N</td>
<td>ARS</td>
<td>0.091</td>
<td>0.878</td>
<td>0.009</td>
<td>0.0727</td>
</tr>
<tr>
<td></td>
<td>diff.</td>
<td>0.048</td>
<td>0.010</td>
<td>0.061</td>
<td>0.0499</td>
</tr>
<tr>
<td></td>
<td>time [s]</td>
<td>4.490</td>
<td>439.04</td>
<td>11.420</td>
<td>697.599</td>
</tr>
</tbody>
</table>
Comparison on synth. images

<table>
<thead>
<tr>
<th>datasets</th>
<th>NMF</th>
<th>FastICA</th>
<th>sPCA</th>
<th>DL</th>
<th>BPDL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>v2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>pure</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARS</td>
<td>1.0</td>
<td>1.0</td>
<td>0.989</td>
<td>1.0</td>
<td>0.999</td>
</tr>
<tr>
<td>diff.</td>
<td>0.0</td>
<td>0.0</td>
<td>0.037</td>
<td>0.0</td>
<td>0.005</td>
</tr>
<tr>
<td>time [s]</td>
<td>82.329</td>
<td>5533.4</td>
<td>460.82</td>
<td>14786.</td>
<td>88.260</td>
</tr>
<tr>
<td><strong>deform</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARS</td>
<td>0.818</td>
<td>0.846</td>
<td>0.801</td>
<td>0.807</td>
<td>0.970</td>
</tr>
<tr>
<td>diff.</td>
<td>0.019</td>
<td>0.015</td>
<td>0.056</td>
<td>0.046</td>
<td>0.004</td>
</tr>
<tr>
<td>time [s]</td>
<td>144.10</td>
<td>5683.2</td>
<td>477.47</td>
<td>13619.</td>
<td>165.22</td>
</tr>
<tr>
<td><strong>D&amp;N</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARS</td>
<td>0.120</td>
<td>0.612</td>
<td>0.024</td>
<td>0.144</td>
<td>0.877</td>
</tr>
<tr>
<td>diff.</td>
<td>0.036</td>
<td>0.036</td>
<td>0.092</td>
<td>0.039</td>
<td>0.013</td>
</tr>
<tr>
<td>time [s]</td>
<td>77.399</td>
<td>6912.9</td>
<td>485.44</td>
<td>13729.</td>
<td>289.51</td>
</tr>
</tbody>
</table>

(size 128 × 128 px, 23 patterns)
Input segmented imaginal discs
Visualise results on imaginal discs

Gene & atlas

Used patterns

Pattern activation
Extracted Atlases for Imaginal discs
Results on imaginal discs

No ground truth atlases…

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of patterns $K$</th>
<th>Time [min]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>NMF</td>
<td>0.0939</td>
<td>0.0823</td>
</tr>
<tr>
<td>FastICA</td>
<td>0.1197</td>
<td>0.0779</td>
</tr>
<tr>
<td>sPCA</td>
<td>0.0476</td>
<td>0.0413</td>
</tr>
<tr>
<td>DL</td>
<td>0.0939</td>
<td>0.0648</td>
</tr>
<tr>
<td>BPDL</td>
<td>0.0467</td>
<td>0.0395</td>
</tr>
</tbody>
</table>
Parameter selection

Graph Cut regularization $\beta$

<table>
<thead>
<tr>
<th>nb. $K$</th>
<th>0.0</th>
<th>0.001</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

& More discs

(a) (b) (c) (d)
Overview

1. Problem statement
2. Used datasets
3. Instance segmentation
   a. Structure (tissue) segmentation on superpixels
   b. Center detection and ellipse fitting
   c. Region growing on superpixels
4. Binary Pattern Dictionary Learning
5. Conclusion & Future work
Conclusion

- Presented four image processing methods:
  - Image segmentation on superpixels
  - Center detection on segmented images
  - Region growing with shape prior
  - Binary Pattern Dictionary Learning
- Implementation:
  - http://borda.github.com/pyImSegm
  - http://borda.github.com/pyBPDL

- Future work
  - Complete image analysis pipeline in 2.5D
  - Try instance segmentation with CNNs
  - ...