Master thesis project – Image processing to detect worms

Supervisor: Johan Henriksson, Karolinska institutet, Department of biosciences and nutrition
johan.henriksson@ki.se

Background

Images of biological samples are no longer just overview pictures; they are measurements. To turn images into manageable data the computer has to be able to make sense of them. The purpose of this project is to detect *C.elegans* worms (larva) in liquid media. The project can be extended to allow tracking (using microscope XY stage) of worms moving on plates if time allows.

General Approach

- Read literature on past attempts
- Test algorithms. The following are likely good candidates:
  - Thresholding
  - Distance transform, morphology
  - Shape-fitting, energy formulation
- Comparison with expert annotated images

Detailed plan

This is based on how we currently think we can solve the problem and our strategy.

Specific results that are to be obtained and its possible application

- Shape definition plugin for Endrov
- Polygon ROI and rasterizer
- Algorithm with
  - Input: Images of worms in liquid culture
  - Output: Fitted shapes of worms

Activities that the project involve

- Working with a large source code with GIT version control
- Java programming
- Finding a suitable thresholding algorithm
- Doing the math required to use a normal continuous optimization algorithm which does not require differentials
- Rasterizing and maybe tessellating general polygons
- Finding a good shape descriptor
- Optimizing code, both data structures and constant time factor
- Benchmarking algorithm with hand-annotated images
- If time allows, see if algorithm also can be used to track worms on agar plates

Points of interest that must be studied or analyzed during the project development

Should look at other shape-fitting algorithms. Benchmark to fine-tune fitting parameters.

Estimated time

1. Finding a good thresholding algorithm, 3w
2. Rasterizer and polygon ROI, 3w
3. Implement shape descriptor, 3w
4. Implement optimizer (use a library if possible), 2-4w
5. Misc helper image processing functions, 1w
6. Fine-tune and benchmark algorithm. Some way of guessing initial shape, 7w

Literature study included in each step.
Step 1-5 are straight-forward. Minimum objective is the full implementation.
Step 6 can fail entirely; minimum objective is an attempt and if it does not work, documentation of what the problems are and suggestions for further work.

Required resources

Computer and test images made available by KI.

Technical fields that are addressed during the project development

- Java programming
- Numerical optimization (use off-the-shelf algorithm)
- Image processing (thresholding)
- Computer graphics (rendering)
- Code optimization
- Algorithms, data structures (simple ones, only if code too slow)
- Interpolation (maybe splines)