Master Thesis Project Specification

Image processing to detect worms

Javier Fernández

March 25, 2010

1 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. This involves processing the raw images to gather and organize data that would allow to analyze and manipulate the information accurately and faster.

The purpose of this project is to detect C.elegans worms (larva) in liquid media. It would be implemented as part of the image analysis and data processing software *Endrov*. The project can be extended to allow tracking (using microscope XY stage) of worms moving on plates if time allows.

2 Task description

The general objective is to obtain an algorithm that receives images of worms in a liquid culture as input and outputs fitted shapes of worms. This will involve two main stages: the implementation stage and the fine-tune and benchmarking stage.

2.1 Implementation stage

In this stage will be studied the previous attempts and approaches on the different algorithms that have to be implemented in order to fulfil the project main objective of fitting the worms shape from an initial microscope image.

The implementation would be done as a filter for *Endrov*, an image analysis and data processing software developed at Karolinska Institute. This open source project provides a wide suite of tools that allows to manipulate biological data with agility, making more efficient the study of samples.

The main activities that the project involve are the following:

- Rasterizing and maybe tesselating general polygons.

- Finding a suitable thresholding algorithm. This will allow to obtain a binary image from the initial sample, dividing the images into object pixels and background pixels. This will facilitate the manipulation and filtering of the data.
- Finding a good shape descriptor. This involves researching on different past attempts on shape fitting, and designing a shape descriptor that fits the current problem. There are many published studies on shape fitting for many different problems, but just few focused on worms.
- Doing the math required to use a normal continuous optimization algorithm which does not require differentials.
- Optimizing code, both data structures and constant time factor.

2.2 Benchmarking and fine tuning

This stage goal is to analyze and test the accuracy of the implemented approach. The main activities are:

- Benchmarking algorithm with expert annotated images.
- Fine tuning the code to suit better the specification and improve unnacurate results.

3 Procedure

3.1 Technical fields addresed

- Numerical Optimization (use off-the shelf algorithm)
- Image processing (thresholding, distance transsformation, skeletonization)
- Computer Graphics (rendering)
- Code Optimization
- Algorithms, data structures
- Interpolation (maybe splines)

3.2 Programming language and environment

- Java on Eclipse. (Linux environment)
- Working with a large source code with GIT version control (*Endrov*)

3.3 Documentation

- Weekly report highlighting problems addressed, new solutions and problems found, references contribution and links.

4 Time Schedule

- Finding a good thresholding algorithm, 2w
- Rasterizer and polygon ROI, 2w
- Implement shape descriptor, 4w
- Implement optimizer (use a library if possible), 3w
- Misc helper image processing fuctions, 1w
- Fine-tune and benchmark algorithm. Some way of guessing initial shape, 4w
- Report writing and occasional meetings, 3w. (The project is being written in parallel with every task. There are 3 weeks more for completing the report and refining).

4.1 Delimitations

- The implementations tasks are straight forward. The minimum objective is the full implementation.
- The benchmarking and fine tuning step can fail entirely.

Minimum objective is an attempt and if it does not work, documentation of what the problems were and suggestions for further work.