

Fall 2011  
COSC 5P71  
Instructor: B. Ross

## Assignment 3: Computer Vision Part II

**Due date:** 12:00 noon Tuesday December 6.  
**Lates:** 12:00 noon Friday, December 9. (-25%)  
**Goal:** Enhancing your computer vision system.

**Hand in:** A report about your experiments. It must use an IEEE or similar format. See earlier assignments for content requirements of your report. Also include printed and electronic copies of your report, source and parameter files (may be included as appendices in your report). Include electronic copies of training and testing images. Further requirements are described at the end of this assignment.

Please choose **only one** of the following:

### A. Multi-objective evaluation:

- a) Use Pareto ranking to assign fitness ranks. Three dimensions are:
- (i) percentage true positives;
  - (ii) percentage true negatives.

In other words, where assignment 2 combined these values into one overall score, here you will keep them independent. Also include...

- (iii) number nodes in program (where smaller is considered better!).

b) Use Pareto ranking to evaluate multi-classification, in which your program identifies 2 or more features in the image. One Pareto parameter will retain the rank for the score of one object to be classified. For example, for Mars, it could be the interiors of craters. Then include one or more additional training features, such as crater rims, outside plains, mountains, etc. Each of these objectives will be another dimension of the Pareto vector. Of course, new training data will need to be prepared.

Note that fitness performance graphs aren't possible with Pareto ranks. However, you can plot a population scatter graph for 2 dimensions.

### B. Fitness sharing

Fitness sharing is a form of dynamic weight allocation during the scoring of training examples. Imagine that you split the "fitness reward" with all the training examples. Whenever an example (a single training pixel) is incorrectly identified, you update a table (one entry per training example). Then at the end of the generation, you redistribute the fitness based on the number of programs each training pixel "fooled". The pixels that fooled more programs are worth more, and those programs should be assigned a higher score for that pixel. This is equivalent to calculating example weights. You might do this separately for positive and negative examples, or perhaps with all examples together. The advantage of fitness sharing is that tricky portions of the image will be automatically stressed during evaluation, and given a higher value to programs that can handle them.

To implement fitness sharing, you must perform some bookkeeping on the score table that keeps track of which examples are correctly identified, and by which program (identified with

ID perhaps). Then at the end of a generation, the table is examined to figure out the value of each example, and what score to add to the programs that correctly identified the example.

This fitness computation is done between generations on the newly made population, before it is to be used for the next generation. It is easier to implement with generational GA's. Steady-state is not recommended here for fitness sharing.

### **C. Multi-phase evolution**

Your best solution from assignment 2 probably has trouble with certain situations... edges of craters on Mars, or mountains on Mars. For this experiment, take the output from your best solution in assignment 2, and use it to train a new genetic program that will correct its errors. Hence the program could conceivably have the same language as that in assignment 2, with the addition of the output (true/false) mask from the assignment 2 program. However, you will probably want to specialize the language and pre-filtered data for this question. For example, perhaps expanding the area of filters may be prudent.

You should still have positive and negative examples, all of which are "true" pixels on assignment 2's output. You must still tally the scores as done in assignment 2, and test it with testing data. A new colour coded output image should show how well the program corrects mistakes, hopefully without introducing more new errors!

#### **Other Requirements:**

1. Your report should have the style of a scientific research paper, for example, a conference paper or technical report. Use the IEEE format (or similar format).
2. Your experiments should compare at least 2 different variables – one is a set of runs from from assignment 2, while another is one of the high-level enhancements (A, B, C) above. For example, you can compare fitness sharing vs no fitness sharing; or multi-objective Pareto ranking vs normal fitness scoring (from assignment 2). If you are comparing with results from assignment 2, be sure to keep all the independent variables identical between the experiments.

You will be required to apply a statistical significance test (t-tests) to your results, when comparing experiments. Ideally, you should do 20 runs or more, to make stronger t-test results. The test to do is to see if the improved experiment is better than the old assignment 2 run, with a given significance (95%). See the web site for a tutorial on statistical significance testing with Excel.

3. An in-depth discussion of the results is important. Did your new techniques used for this assignment make a positive difference in the results? How significant was the improvement? Thoroughly discuss this in your report!