Structural Description of Binary Images: An Evolutionary Approach

Róbert Ványi

Extraction of some structural information from a binary image may be a very difficult task. In this paper a method using evolutionary algorithms is outlined. The idea is to evolve a structural description that describes the given image as good as possible. However the construction of such an algorithm may also not be trivial.

When one tries to solve a problem with some kind of evolutionary algorithms there are some building blocks of the method which make up the whole algorithm. The evolutionary algorithms use a population, that is a set of possible solutions. From this population some individuals are selected using some kind of goodness measures, called fitness function. Finally from these selected individuals new solutions are produced using evolutionary operators to make up a new population.

First the representation form of these solutions has to be decided. From this usually follow the set of the used evolutionary operators, but there are always some possibilities to choose from. A very important part is the fitness function. This namely influences the convergence speed of the algorithm and it is calculated many times. Finally one have to be careful with the implementation, since the evolutionary algorithms are slow by nature, and therefore each calculation must be as fast as possible. These building blocks are considered in the following sections.

Solution representation: The representation of the solutions depends on the information one would like to extract from the image. Usually this is some kind of picture generating languages, like postscript or turtle graphics. The description can be, however more complex, one can use grammars to produce such descriptions. These higher level descriptions may involve Chomsky or other grammars, like Lindenmayer systems or Collage grammars.

Evolutionary operators: A picture description language usually contains parametrized commands. So a solution is a sequence of parametric symbols. For higher level descriptions the solutions are sets of strings, that is representations of the rewriting rules. It is possible to evolve such structures, but it yields a simpler algorithm and better solutions when the parameters are extracted from the command sequences and evolved separately.

This method yields a *two-level algorithm*. On the upper level the rough structure description is evolved using genetic programming and variable length strings or trees. A goodness measure is however not possible on this level, therefore for each individual the best possible parameter set is approximated using evolution strategies and real vectors on the lower level.

Fitness function: To calculate the fitness of an individual, it has to be interpreted usually yielding a bitmap image. This can then be compared with the given image. One way of this is a pixel by pixel comparison. An easy method is calculating the quadratic error, that is the sum of the squares of pixel value differences for all pixels. There are however better methods involving some distance information. One can also use other types of similarity measurements. For example the Hausdorff distance, which results a very good metric and a very bad calculation time.

Efficient calculation: As the most CPU time is usually consumed by the fitness calculation, to make a fast algorithm this has to be made fast. This can be achieved by the so called *efficient fitness functions*, defined by the author. They can be calculated in linear time with respect to the number of drawn (that is nonzero) pixels. It was shown, that fitness functions that use any information from the given image but only local information from the generated images, can be computed efficiently.