

Project 3: Segmentation of Color Image Using Multi-Objective Evaluation Algorithm

Per Magnus Veierland
permve@stud.ntnu.no

March 23, 2017

IMAGE SEGMENTATION

The described system implemented in C++ performs the task of image segmentation through Multi-Objective Evolutionary Algorithm (MOEA) using the NSGA-II algorithm [1] with the three objective functions; *overall deviation*, *edge value*, and *connectivity measure*. Each individual chromosome in the population encodes a set of graphs where each separate graph describes an image segment. The encoding is capable of encoding any possible non-intersecting and non-overlapping image segmentation. Evolution shapes the individuals in the population according to the provided objective functions to find the set of graphs which describes the best segmentation of the input image.

To allow for efficient experimentation and evaluation, all benchmark images were scaled while respecting their aspect ratio such that their maximum dimension is 128 pixels.

The *Boost Generic Image* library was used together with *libjpeg* to handle images, and the *Qt* library was used to render solution output images.

CHROMOSOME REPRESENTATION

The implementation uses a chromosome representation (see Figure 1) inspired by the graph-structure described in [2]. The genotype of an individual is a fixed-length sequence with length equal to the number of pixels in the input image. Each cell in the genotype contains one out of five possible values; *left*, *up*, *down*, *right*, or *none*. This cell value describes how the graph node representing the input image pixel at the index of the cell is connected to its neighbors. Each graph node can connect to either a single of its four cardinal neighbors, or to itself. If a graph node at an edge of the image plane points in an outwards direction, it is treated as having the value *none*. This means that all possible chromosome permutations are valid.

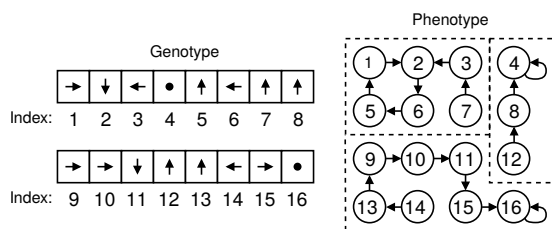


Figure 1: Image segmentation chromosome for a 4 × 4 pixels image. Each number represents an index in the genotype which corresponds to the pixel index in the two-dimensional input image. The dotted sections indicate the resulting image segments.

CREATION

Initial genotype sequences are generated by constructing a Minimum Spanning Tree (MST) from the input image. This is done to provide a good starting point in the chromosome search space for image segmentation. The input image is treated as a graph where each pixel is a node connected to each of its cardinal neighbors. The weight of each edge is given by the Euclidean distance δ_{RGB} between the two neighbors in RGB color space:

$$\delta_{\text{RGB}} = \sqrt{\Delta R^2 + \Delta G^2 + \Delta B^2} \quad (1)$$

From the initial image graph, *Prim's algorithm* [3] is applied from a random starting point to create a MST. Since a random starting point is used, a different MST is constructed as the basis of the genotype for each initial individual.

DEVELOPMENT

Evaluating the chromosome of an individual requires developing the genotype into a phenotype. The development process compiles the graph described by the genotype into the image segmentation description. During the development process the directions of edges in the graph described by the genotype is ignored. Starting with the first pixel node in the graph, all directly or indirectly connected pixel nodes are assigned to the same segment. This process is continued until all pixel nodes have been assigned to a segment.

EVALUATION

After compiling the image segmentation description during development, the *overall deviation*, *edge value*, and *connectivity measure* objective functions are evaluated to produce the objective values for the individual.

SELECTION

The MOEA implementation uses a tournament selection strategy for parent selection. To select a single parent, a random sample of k individuals are drawn from the population P_t . With a probability of ϵ , a random individual is selected from the group of k individuals, and with a probability of $1 - \epsilon$; the lowest ranked individual in the group of k individuals is selected according to the *crowded-comparison operator* $<_n$ which is described in NSGA-II [1]. This procedure is performed twice to select two parents for each mating which results in two offspring.

GENETIC OPERATORS

As part of the evolution process the genetic operators crossover and mutation are used. During inheritance, a single-point crossover operation combines the genotype sequences of two parent individuals to produce two children genotype sequences.

Crossover is controlled by a crossover rate, which is a probability between $[0, 1]$ determining whether to apply the crossover operator to produce two children genotypes from two selected parent's genotypes, or to use the parent genotypes unmodified by crossover.

When applied, the mutation operator selects a random gene in a genotype sequence and sets it to a new value which is uniformly randomly selected from the set $\{left, up, down, right, none\}$.

Mutation is controlled by a mutation rate, which is a probability between $[0, 1]$ determining whether to mutate an individual's genotype.

PARAMETERS

Table 1 lists all parameters found using trial and error which were used for all image segmentation experiments.

Parameter	Value
Population	200
Generations	250
Crossover rate	1.0
Mutation rate	0.05
Tournament group size	10
Tournament randomness (ϵ)	0.1

Table 1: MOEA parameter values

IMPLEMENTATION NOTE

To eliminate a bias introduced by the crowding-distance-assignment function, it was found necessary to shuffle the last front added to the next population. Otherwise, if only part of the last front is added to the next population, having the front sorted by one of the objective functions creates a significant bias towards the lower end of the last objective function sorted for. This in turn can lead to a clustering of solutions on one of the ends of the population front.

As the true Pareto front is unknown to the implementation for the problem given, f_m^{\max} and f_m^{\min} are estimated for each objective function m by tracking the maximum and minimum values encountered throughout an evolutionary run for each objective.

REFERENCES

- [1] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.
- [2] Shinichi Shirakawa and Tomoharu Nagao. Evolutionary image segmentation based on multiobjective clustering. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 2466–2473. IEEE, 2009.
- [3] Robert Clay Prim. Shortest connection networks and some generalizations. *Bell Labs Technical Journal*, 36(6):1389–1401, 1957.

SOLUTIONS

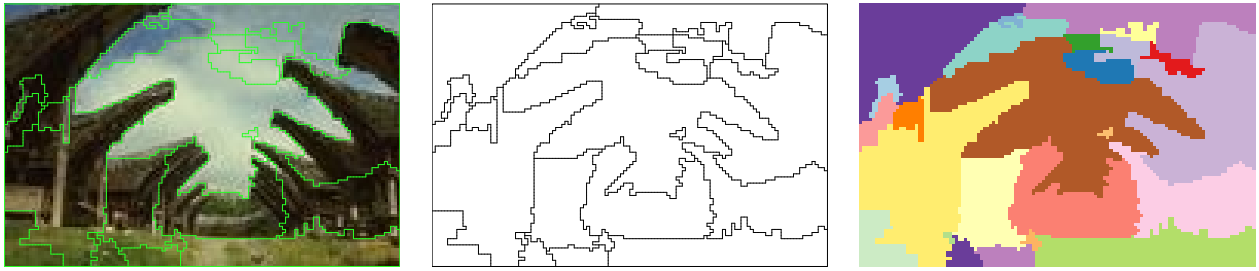


Figure 2: Segmentation of test image #3 using objective functions *overall deviation* and *edge value*.
Segment count: 25 *Overall deviation:* 379206.5 *Edge value:* 203872.7

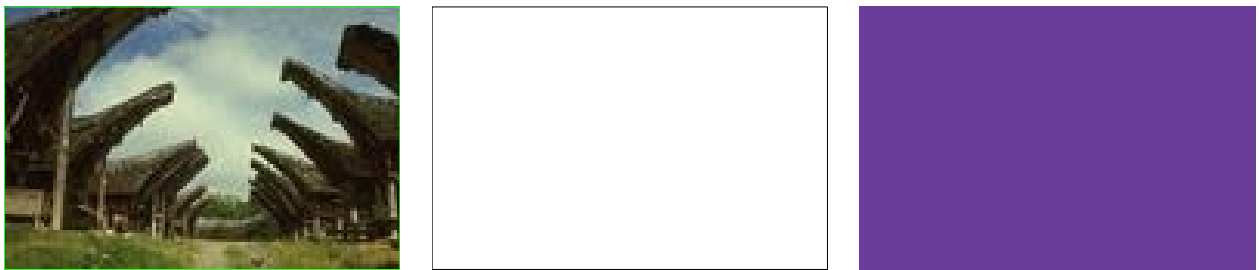


Figure 3: Segmentation of test image #3 using objective functions *edge value* and *connectivity measure*.
Segment count: 1 *Edge value:* 0.0 *Connectivity measure:* 0.0

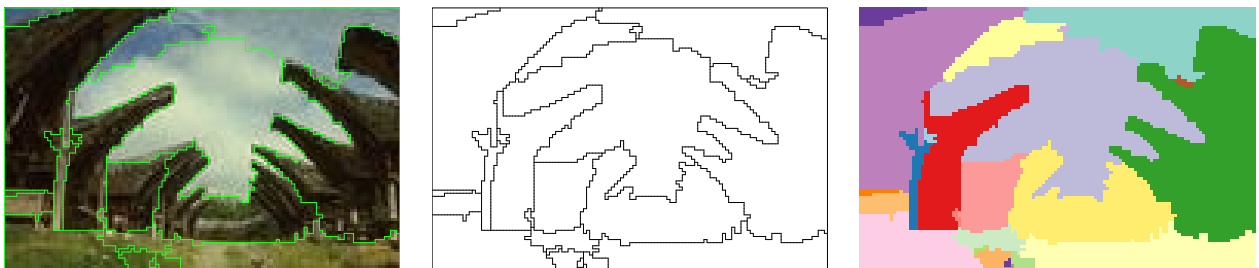


Figure 4: Segmentation of test image #3 using objective functions *overall deviation* and *connectivity measure*.
Segment count: 25 *Overall deviation:* 339663.8 *Connectivity measure:* 2087.2

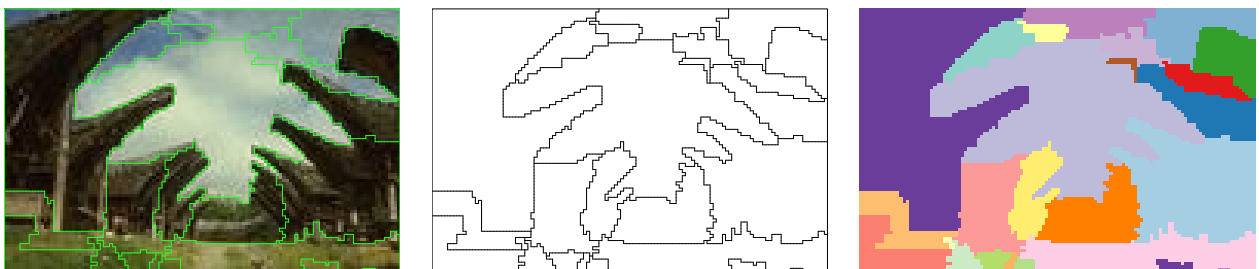


Figure 5: Segmentation of test image #3 using objective functions *overall deviation*, *edge value*, and *connectivity measure*.
Segment count: 25 *Overall deviation:* 345576.5 *Edge value:* 206196.8 *Connectivity measure:* 2213.0

PARETO FRONTS

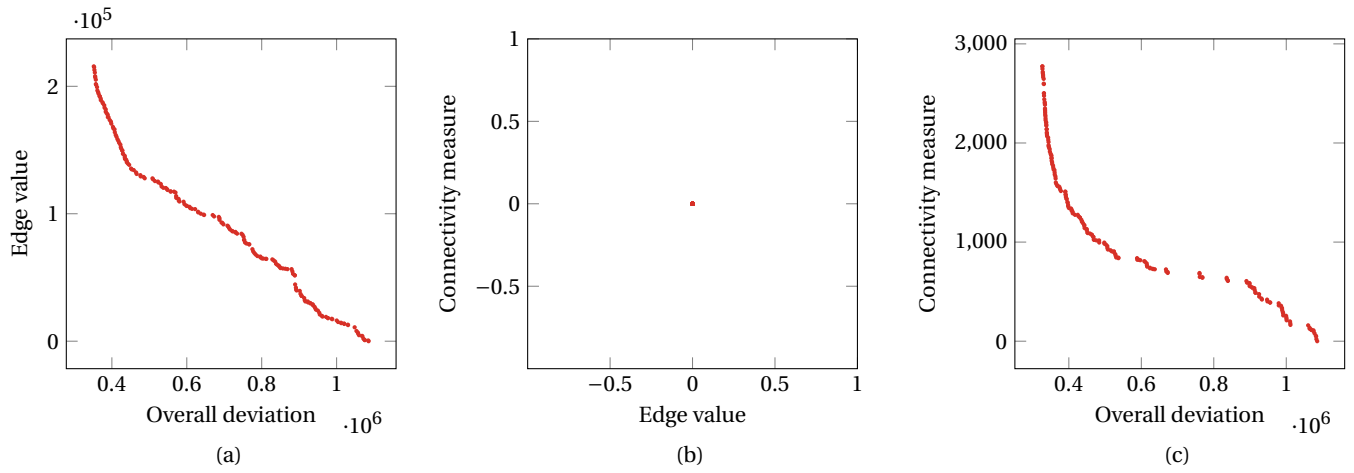


Figure 6: Plots showing the Pareto front resulting from multi-objective NSGA-2 evolution using combinations of two objective functions; *overall deviation* and *edge value* (a), *edge value* and *connectivity measure* (b), and *overall deviation* and *connectivity measure* (c). All final individuals belongs to the first non-dominated front.

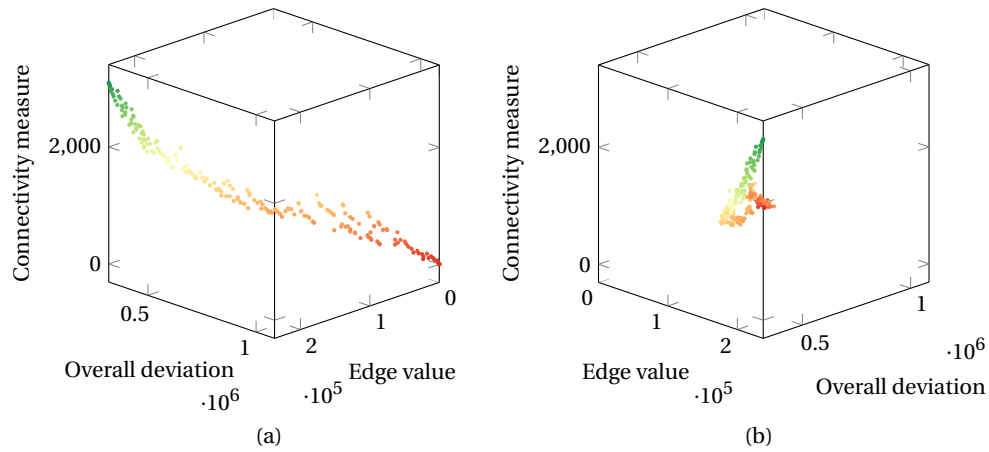


Figure 7: Plots showing the Pareto front resulting from multi-objective NSGA-2 evolution using the three objective functions *overall deviation*, *edge value*, and *connectivity measure*. The colors reflect the *connectivity measure* and is meant to improve the readability of the plot. All final individuals belongs to the first non-dominated front.

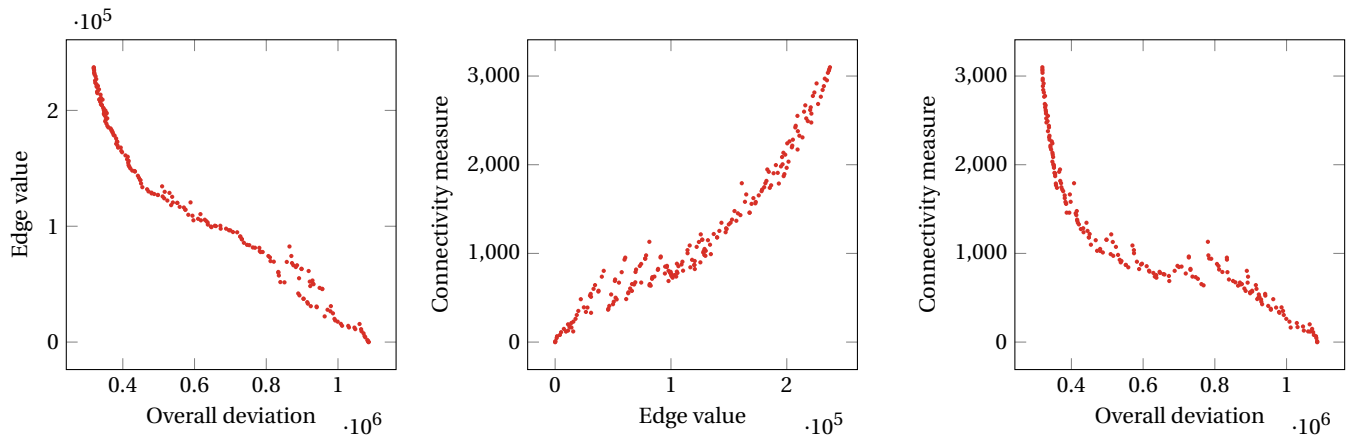


Figure 8: Plots showing the same data as Figure 6 in three separate two-dimensional views.