IT 3708: Project 3

Segmentation of Color Image Using Multi-Objective Evaluation Algorithm (MOEA)

Lab Goals

- Implement multi-objective evaluation algorithm (MOEA) to color image segmentation.
- Compare the performance of your implemented algorithm(s) on several benchmark problems.
- Test and analyze effects of single objective vs multi-objective optimization.

Groups Allowed? Yes. For this project you may work alone or in groups of two.

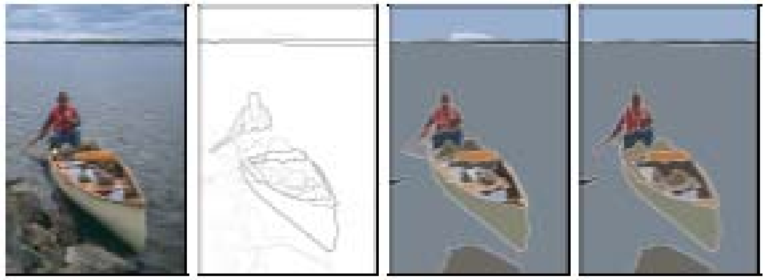
Deadline: March 23, 2017 (Thursday) at 07: 59 AM.

Assignment Details

In a conventional sense, image segmentation is the partitioning of an image into non-intersecting regions (***set of pixels***), where pixels within a region are similar according to some uniformity predicate, and dissimilar between neighboring regions. Image segmentation is a fundamental process in many image, video, and computer vision applications, where the segmented regions have two properties: (1) homogeneity within a region, i.e., the texture, color, or intensity of each pixel in a region should be similar to the other, and (2) heterogeneity between the regions, i.e., the texture, color, or intensity of the pixels in one region should be distinct from the pixels in another region. Fig. 1 presents several examples of image segmentation.

(a)

(b)



(c)

Fig. 1: Image segmentation. Rightmost figures are original images, and others are several segmentation examples.

From Fig. 1, you can find that the same image can be segmented differently, however, **the main goal is to partition an image into separate regions of pixels, which ideally correspond to different real-world objects**. Segmentation is generally the first stage in any attempt to analyze or interpret an image automatically. It bridges the gap between low-level image processing and high-level image processing. Some kinds of segmentation technique will be found in any application involving the detection, recognition, and measurement of objects in images.

In this project, you will implement **a segmentation technique for color image using multi-objective evolutionary algorithm (MOEA)**. You can select any one of the following three MOEAs as your choice: (i) NSGA-II, (ii) SPEA, (iii) PAES. To analyze the effects of multi-objective optimization, you will segment color images not only optimizing multiple segmentation criteria (objectives) simultaneously, **but also optimizing each criterion individually**. It implies that you need to implement simple GA also.

Problem Formulation:

The first step in formulating the evolutionary-based approach is to convert the phenotype (image) into genotype (chromosome) which will be processed during the evolutionary process. In most cases, pixels are stored as corresponding color values (RGB or CIE L*a*b as the color space [1]). However, **you are free to choose your own representation** (chromosome: transformation from genotype to phenotype). It is necessary mentioning that chromosome representation will guide rest of the evolutionary process. Therefore, you need to be very careful in choosing chromosome representation. There are several alternatives for chromosome representation for image segmentation. Some of them are graph-representation/tree-representation; you can store all the pixels or you can store only the segment. **Be noted that these are only few examples.** There are many existing representation techniques. As mentioned earlier, **you can choose as per your requirement, even you can propose a new one**.

You need to simultaneously optimize three objectives (segmentation criteria) using your chosen MOEA. The objectives are: (i) overall deviation, (ii) edge value, and (iii) the connectivity measure.

- The *overall deviation* is a measure of the similarity of pixels in the same segment, as defined in Equation (1). It expresses the compactness of segments by giving the overall summed distances between the pixels and the center value of the corresponding segment they belong to.

$$\text{overall-deviation}(C) = \sum_{C_k \in C} \sum_{i \in C_k} dist(i, \mu_k) \tag{1}$$

Where C is the set of all segments, μ_k is the centroid of the pixels in the segment C_k, and *dist( )* is the distance function. **Overall deviation should be minimized**. Minimizing overall deviation roughly increases the number of segments. You need to define the distance function *dist( )* as Euclidean distance, and use either RGB or CIE L*a*b* [1] as the color space. The distance function using the RGB color space is defied in equation (2).

$$\delta_{RGB} = \sqrt{\Delta R^2 + \Delta G^2 + \Delta B^2} \tag{2}$$

The distance function using the CIE L*a*b* color space is defined in equation (3). The CIE L*a*b* color space has a uniform chromaticity scale.

$$\delta_{L*a*b} = \sqrt{\Delta L*^2 + \Delta a*^2 + \Delta B*^2} \tag{3}$$

- The second objective, the *edge value*, evaluates the overall summed distances on boundaries between the segments. This value is a measure of the difference in the boundary between the segments. **It is also minimized**, which roughly decreases the number of segments. It is defined as

$$\mathrm{Edge}(C) = -\sum_{i=1}^{N}\left(\sum_{j \in F_i} x_{i,j}\right),$$

$$\text{where,} \quad x_{c,s} = \begin{cases} dist(c,s) & if \nexists C_k : c,s \in C_k \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where N is the number of pixels, F_i indicates the four neighboring pixels of pixel i, and *dist( )* is the distance function as described in the above objective.

- The third objective, the *connectivity measure*, is defined in Equation (5). This objective evaluates the degree to which neighboring pixels have been placed in the same segment, as follows.

$$\mathrm{Conn}(C) = \sum_{i=1}^{N}\left(\sum_{j=1}^{L} x_{i,nn(j)}\right),$$

$$\text{where,} \quad x_{c,s} = \begin{cases} \frac{1}{j} & if \nexists C_k : c,s \in C_k \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where N is the number of pixels in a segment, $nn(j)$ is the j-th nearest neighbor of the i, L is a parameter determining the number of neighbors that contribute to the connectivity measure (you can use any value for L. However, if you use L=4, it will be similar to F_i used in edge value). As an objective, **the connectivity measure will also be minimized**. This objective (Eq. 5) uses a ***gradually decreasing*** penalty ($1/j$). For example, if 3 (any 3) of your total nearest neighbors (L) follows the condition ($if \nexists C_k : c,s \in C_k$) i.e. not in the same segment, then $x_{c,s} = 1 + 1/2 + 1/3$. Similarly, if 2 (any 2) of your total nearest neighbors (L) follows the condition, then $x_{c,s} = 1 + 1/2$.

By minimizing these three objectives, a variety of different segmentations for the same image are generated. In other words, your method will return a range of solutions that have different numbers of **trade-off (Pareto-optimal) segments**.

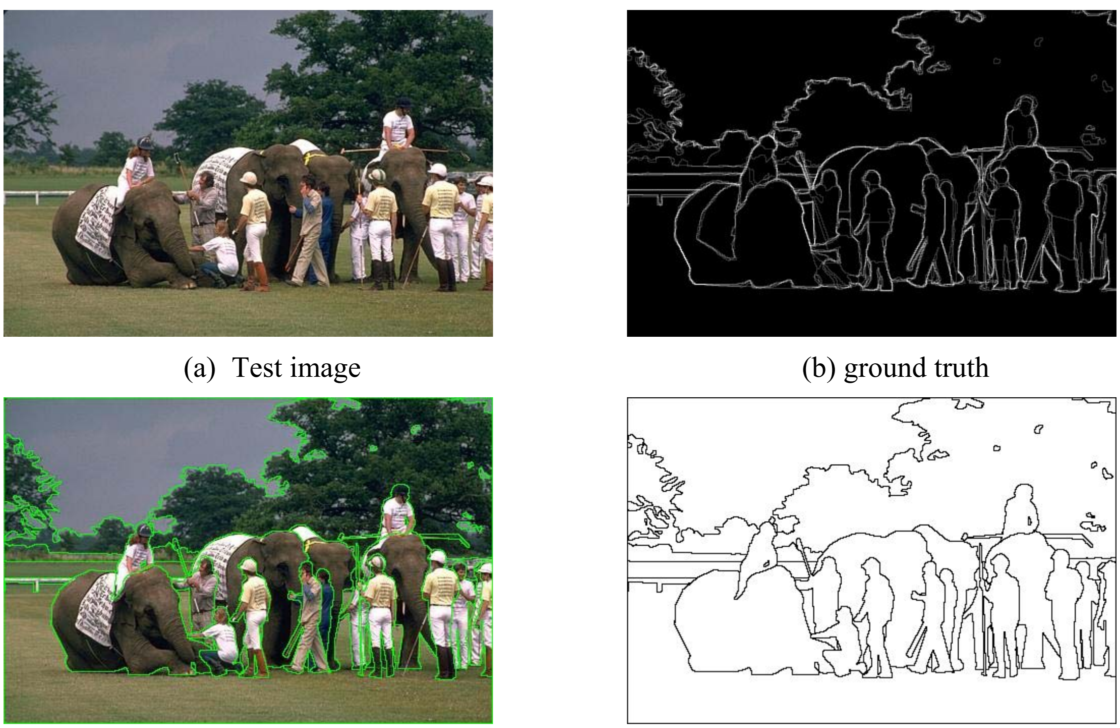
As mentioned earlier, you also need to implement a simple single objective GA that will optimize each objective separately. As a result, you will also get three extra segmentation solutions. By comparing these segmentations with the trade-off (Pareto-optimal) segmentation produced by the MOEA, you can analyze the effects of multi-objective evolutionary optimization.

Things To Do

The 30 points total for this project is 30 of the 100 points available for this course. The 30 points will be distributed on two parts: (*i*) demo and (*ii*) report. **The demo can give you a maximum of 25 points and the report can give you a maximum of 05 points**.

To test your code, we uploaded 08 benchmark images and their solutions. The solutions contain *ground-truth solution*[1] and several other segmented solutions of the same image. These solutions are considered "good" segmentation examples considering several performance metrics.

You need to produce two types of segmentation solutions for each image. Fig. 2 presents the requirements. As shown in the figure, for the original image (Fig. 2a), you need to produce two types of segmentation as shown in Fig 2c and 2d. The ground truth is also presented in Fig. 2b. As you can assume, one image can be segmented differently with numbers of different segments. For example, Fig. 3 presents different reasonable "good" solution for the original image presented in Fig. 2a.



(a) Test image

(b) ground truth

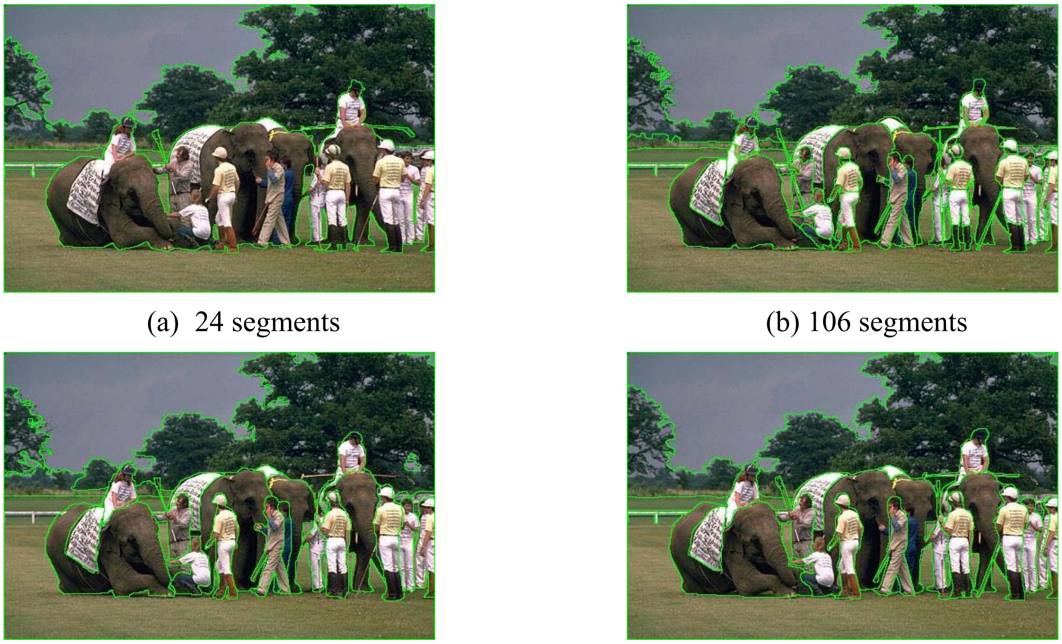
(b) Required solution type - 1

(d) Required solution type - 2

Fig. 2: Examples for your requirement

[1] Ground truth solution: the segmented solution of the test image that is known to be much more accurate solution. Basically, it is created by segmenting the image by lots of human subjects (including both experts and non-experts). Then the ground-truth is created considering all the segmented solutions.

In this project, you need to implement segmentation approach for color image optimizing single and multiple objectives using any one of the three MOEAs mentioned earlier. You need to follow the following experimental steps.



(a) 24 segments

(b) 106 segments

(b) 54 segments

(d) 42 segments

Fig. 3: Different segmentation of the same image

Step -1: Implement the segmentation approach using MOEA optimizing two of the three objectives separately.

1 (a) optimizing overall deviation and edge value.

1 (b) optimizing edge value and the connectivity measure.

1 (c) optimizing overall deviation and the connectivity measure.

Step -2: Implement the segmentation approach using MOEA optimizing all the three objectives simultaneously.

For Step-1, you need to show the final Pareto-optimal solutions (final segmentations) along with respective objective values and segmentation number by optimizing two of the three objectives separately. If the number of final Pareto-optimal segmentations is more than 5, you can show any 5 among them. For presenting segmented image(s), you need to show both types (for example, Fig. 2a and 2b) of segmentation for the same solution.

You need to repeat the same procedure for Step – 1a, 1b and 1c.

For Step-2, Since you will optimize all the three objectives simultaneously, you need to present the final Pareto-optimal front consisting of all three objectives[2]. Then again, you need to show the final Pareto-optimal solutions (segmentations) along with respective three objective values and segmentation number

[2] The Pareto-front consisting of three objectives will be different from what we observe as usually. However, it is a simple extension with another axis. Along with the usual *x* and *y* axes, you need to add the third (*z*) axis for the extra (third) objective.

for each of the Pareto-optimal solution produced at the last generation. Similarly, if the number of final Pareto-optimal solutions is more than 5, you can show any 5 among them.

(a) Demo (25p):

There will be a demo session where you will show us the running code and we will verify that it works. If you work in a group, **both group members need to attend the demo session together**.

In the demo session, you need to describe how you designed and implemented your GA. Also, you need to test your code by segmenting 01 (one) test image which will be supplied during the demo. Note that, **you must run your code and show us all the requirements within 30 (thirty) minutes**. The point distribution for the demo is as follows:

(1) Step – 1: You need to optimize two objectives simultaneously for three times, as mentioned above. (15p = 5 x 3)

- Show the Pareto-optimal segmentations along with (i) respective objective value, and (ii) segmentation number for all the final Pareto-optimal segmentation solutions (if the number of final Pareto-optimal solutions is more than 5, show any 5). Like Step-1, for presenting segmented images, you need to show both types (as shown in Fig. 2a and 2b). (1)
- Does any of your Pareto-optimal segmentation find the known optimal number of segmentation? Since there are several benchmark solutions of the same test images, it is sufficient if your segmentation number matches with any of the segmentation number of the available benchmark solutions. (3p)
  - If your value is within ±2 of any of benchmark solutions, you will get full points.
  - If your value is within ±4 of any of benchmark solutions, you will get 2.5 point.
  - If your value is within ±5 of any of benchmark solutions, you will get 2 point.
  - If your value is within ±6 of any of benchmark solutions, you will get 1.5 point.
  - If your value is within ±7 of any of benchmark solutions, you will get 1 point.
  - Otherwise, you will get 0.
- Present the Pareto-front consisting of the two segmentation criteria (objectives). (1p)

(2) Step – 2: Segmentation by optimizing all three objectives simultaneously. (9p)

- Show the segmented Pareto-optimal solutions along with (i) respective objective value, and (ii) segmentation number for all the final Pareto-optimal segmentation solutions (In case of more than 5 Pareto-optimal solutions, show any 5). As earlier, for presenting segmented images, you need to show both types (as shown in Fig. 2a and 2b). (4)
- Does any of your Pareto-optimal segmentations find the known optimal number of segmentation? Like Step-2, it is sufficient to compare with any segmentation number of the available benchmark solutions for the test image. (4p)
  - If your value is within ±2 of any of benchmark solutions, you will get full points.
  - If your value is within ±3 of any of benchmark solutions, you will get 3.5 point.
  - If your value is within ±5 of any of benchmark solutions, you will get 2.5 point.
  - If your value is within ±6 of any of benchmark solutions, you will get 2 point.

- If your value is within ±7 of any of benchmark solutions, you will get 1.5 point.
- Otherwise, you will get 0.

- Present the final Pareto-optimal front consisting of all three objectives. (1p)

(4) Describing your MOEA implementation. (1p)

Your system need to be an integrated one, i.e. after terminating your EA/MOEA, it will automatically show the segmentation solution(s) of both types and the corresponding Pareto-front.

(b) Report (05p):

You should write a report answering the points below. Your report must not exceed 04 (Four) pages in total. **Over length reports will result in points being deducted from your final score**. Print on both sides of the sheet, preferably. **Bring a hard copy of your report to the demo session**. If you work in a group, you only need to submit one single report on behalf of the group.

1. How does your MOEA find segmentation? (0.5p)
2. Describe your implementation, only for MOEA optimizing three objectives. Your description should include every step of your implementation including representation and evolutionary operators. **Using figure(s) for chromosome is a must**. (2p)
3. For any of the test images, provide one of the segmentation examples for Pareto-optimal (for two and three objectives) solutions. You need to use the same test image for all categories (3 two objectives, 1 three objectives). You need to show both types (as Fig. 2a and 2b) of segmentation for each solution. (1p)
4. Plot Pareto-fronts (3 two objectives, 1 three objectives) for the same solutions those you will use in Point-3. (1p)
5. Mention the values for every parameter of your chosen MOEA. (0.5p)

Delivery

You should deliver your report + a zip file of your code on *itslearning*. The submission system will be closed at 07:59 AM on March 23, 2017.

If you work in a group, you only need to deliver once on *itslearning* (but both group members must the registered as part of the submission on *itslearning*!). **Both group members need to attend the demo session**.

You must attend the demo on the scheduled demo date which has been declared on *itslearning***.** Since the demo dates were declared at the beginning of the semester**, no early or late demo will be entertained** except for extreme emergency like sickness with medical certificate, job interview, attending funeral or the like**.** Traveling or holidays will not be considered as emergency situation.

References:

[1] Gunther Wyszecki and W. S. Stiles. Color Science: Concepts and Methods, Quantitative Data and Formulae, 2nd Edition. Wiley Interscience, 2000.