Evolutionary Image Segmentation Based on Multiobjective Clustering

Shinichi Shirakawa and Tomoharu Nagao

***Abstract*— In the fields of image processing and recognition, image segmentation is an important basic technique in which an image is partitioned into multiple regions (sets of pixels). In this paper, we propose a method for evolutionary image segmentation based on multiobjective clustering. In this method, two objectives, *overall deviation* and *edge value*, are optimized simultaneously using a multiobjective evolutionary algorithm. These objectives are important factors for image segmentation. The proposed method finds various solutions (image segmentation results) by the use of an evolutionary process. We apply the proposed method to several image segmentation problems and confirm that various solutions are obtained. In addition, we use a simple heuristic method to select one solution from the original Pareto solutions and show that a good image segmentation result is selected.**

I. Introduction

Image processing and recognition technologies are becoming increasingly important. In the fields of image processing and recognition, image segmentation is a basic and important technique in which an image is partitioned into multiple regions (sets of pixels). Each of the pixels in a region is similar to the other with respect to some characteristic or computed property, such as color, intensity, or texture. Recently, many image segmentation methods have been proposed and their effectiveness have been demonstrated [1], [2]. Typical examples include clustering methods, histogram-based methods, region growing methods [3], and graph partitioning methods [4], [5]. In clustering methods, the k-means algorithm is the most popular technique, and is used to partition an image into k clusters. The quality of the solution using k-means depends on the initial set of clusters and the value of k. Region growing methods examine the neighboring pixels of initial "seed points" and determine whether the pixel should be added to the seed point.

The minimum spanning tree (MST) technique is also used in image segmentation [6], [7]. Each pixel is a node in a graph. The weight of an edge is a measure of the similarity between pixels. The image is partitioned into segments by removing the edges whose weight is more than the threshold.

Evolutionary computation (EC) techniques are also applied to image segmentation problems. Several algorithms using genetic algorithms (GAs) have been proposed [8], [9]. Poli applied genetic programming (GP) [10] to construct pixel-classification-based segmentation algorithms [11].

S. Shirakawa and T. Nagao are with the Department of Information Media and Environment, Graduate School of Environment and Information Sciences, Yokohama National University, 79-7 Tokiwadai, Hodogaya-ku, Yokohama, Kanagawa 240-8501, Japan `shirakawa-shinichi-bg@ynu.ac.jp`, `nagao@ynu.ac.jp`

The multiobjective clustering algorithm, multi-objective clustering with automatic k-determination (MOCK), was developed by Handl and Knowles [12], [13], and various improvements and applications have been investigated [14]–[18]. This algorithm optimizes two complementary objectives based on cluster compactness and connectedness using a multiobjective evolutionary algorithm (MOEA) [19], [20]. MOCK has been applied mainly to artificial datasets. MOCK is reported to find partitions better than other conventional clustering algorithms, such as the k-means method.

In this paper, we propose a method for evolutionary image segmentation based on multiobjective clustering. In the proposed method, two objectives, *overall deviation* and *edge value*, are optimized simultaneously using an MOEA. These objectives are important factors in image segmentation. The proposed method finds various solutions (image segmentation results) through an evolutionary process. We apply the proposed method to several image segmentation problems and confirm that various solutions are obtained. Since there is no general solution to image segmentation problems, image segmentation techniques often have to be combined with domain knowledge to solve such problems effectively for a problem domain. Therefore, we believe that it is significant to obtain various types of image segmentation results simultaneously. In addition, we use a simple heuristic method to select one solution from the original Pareto solutions and show that a good image segmentation result is selected.

The next section of this paper presents an overview of MOCK. In Section 3, we describe our proposed method for evolutionary image segmentation based on multiobjective clustering. Next, in Section 4, we apply the proposed method to image segmentation problems and show several experimental results. Finally, in Section 5, we describe our conclusions and future work.

II. Multiobjective Clustering with Automatic K-determination (MOCK) [12], [13]

MOCK is a recent clustering algorithms based on an MOEA. MOCK optimizes two clustering objectives, *overall deviation* and *connectivity*, which reflect two fundamentally different aspects of a good clustering solution. Therefore, MOCK returns not just one solution, but an entire set of solutions. The first objective, overall deviation, is defined in equation (1). This equation described the overall summed distances between data items and their corresponding cluster center.

$$Dev(C) = \sum_{C_k \in C} \sum_{i \in C_k} \delta(i, \mu_k) \quad (1)$$

where C is the set of all clusters, μ_k is the centroid of cluster C_k, and $\delta()$ is the distance function. Overall deviation should be minimized. Minimizing overall deviation increases the number of clusters.

The second objective, connectivity, is defined in equation (2). This objective evaluates the degree to which neighboring data points have been placed in the same cluster.

$$Conn(C) = \sum_{i=1}^{N} \sum_{j=1}^{L} x_{i,nn_i(j)},$$
$$\text{where} \quad x_{r,s} = \begin{cases} \frac{1}{j} & if \nexists C_k : r, s \in C_k \\ 0 & otherwize \end{cases} \quad (2)$$

where N is the number of data points, $nn_i(j)$ is the jth-nearest neighbor of datum i, and L is a parameter determining the number of neighbors that contribute to the connectivity measure. Connectivity is also minimized. By minimizing these two objectives, a variety of different solutions are generated in MOCK.

Each individual in MOCK can be represented as a graph. Each datum is represented as a node, and an edge between two nodes indicates that these data are in the same cluster. Each individual has an N-length gene (N is the number of data points). The range of the gene value is $[1, N]$. When the ith gene value is j, data item i links to data item j. The decoding of this representation requires the identification of all subgraphs. Thus, the number of clusters is determined automatically in the decoding step.

In the MOCK algorithm, MST is used for the initialization of individuals. MST is the shortest tree that contains all nodes in the graph and has no loops. The shortest tree means that the total cost of all edges is the minimum. MOCK uses Prim's algorithm to create the MST. In the initialization of the ith individual in the population, the $(i-1)$ long links are removed from the MST individual. In this manner, MOCK creates high-quality initial solutions.

In addition, MOCK has a scheme for determining the number of clusters automatically (automatically selecting the best solution). This scheme is based on the Gap statistic [21]. In this scheme, random data are created based on the principal component of original data, and these data are divided into clusters by MOCK. The solutions of the random and original data are different. The appropriate number of clusters is determined using these solutions of random and original data.

III. Evolutionary Image Segmentation Based on Multiobjective Clustering

In this section, we describe the proposed method for evolutionary image segmentation based on multiobjective clustering.

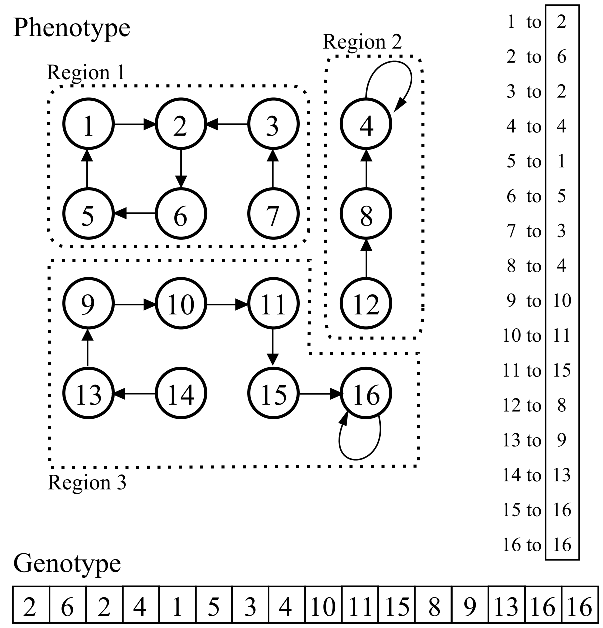


Fig. 1. Example of the phenotype (graph representation) and genotype (string of integers) of an individual in the proposed method. The graph corresponds to an image plane.

A. Representation of individuals

Figure 1 illustrates the phenotype and genotype of an individual in the proposed method. The representation of individuals is a graph structure. The phenotype corresponds to an image plane. Each pixel in the image is represented as a node, and an edge between two nodes indicates that these pixels are in the same region (cluster). Each node can connect only with either itself or the four neighboring pixels. This graph representation indicates image segmentation. Each individual has an N-length gene (N is the number of pixels). When the ith gene value is j, pixels i and j are in the same region. For instance, nodes (pixels) numbered 1, 2, 3, 5, 6, and 7 in Figure 1 are in the same region. The decoding of this representation requires identification of all subgraphs. Thus, the number of regions is determined automatically in the decoding step. Because of the constraint on connections, regions far from each other on the image plane do not appear as the same region. A very large search space can be limited using this constraint, and it also reduces the computational time required to create the MST. We use an MOEA to optimize image segmentation represented by the graph structure.

In the proposed algorithm, MST under the constraint on node connections is used for the initialization of individuals. We use Prim's algorithm, as used in MOCK, to create the MST. We assume to the initial individuals, MST is used as it is (do not remove the links from MST). Therefore, initial individuals have the potential for creating high-quality solutions.

B. Multiobjective evolutionary algorithm

We adopt an evolutionary method to obtain the optimum structure. The genotype of the proposed method is a linear string, as in MOCK. Therefore, the method can use a typical simple genetic operator. In this study, we use *uniform crossover* and *mutation* as the genetic operators. In uniform crossover, randomly selected genes are swapped between two parents, which generate offsprings. In mutation, randomly selected genes with a mutation rate P_m are randomly changed under the constraints on connections.

We use Strength Pareto Evolutionary Algorithm 2 (SPEA2) [22] as the MOEA technique. SPEA2 assigns fitness values to individuals according to dominance and density criteria. The selection of individuals is also based on these criteria. It maintains two populations, a current population and an archive population. In each generation, the nondominated solutions in the current population and the archive are copied into the archive of the next generation. Selection occurs from both the population and archive. The procedure of the SPEA2 algorithm is as follows:

1) **Initialization:** Generate an initial population P_0 and create the empty archive population $\bar{P}_0 = \emptyset$. Set the generation counter $t = 0$.
2) **Fitness assignment:** Calculate fitness values of individuals in P_t and $\bar{P}_t$.
3) **Environmental selection:** Copy all nondominated individuals in P_t and $\bar{P}_t$ to $\bar{P}_{t+1}$. If the size of $\bar{P}_{t+1}$ exceeds $\bar{N}$ (archive size), reduce $\bar{P}_{t+1}$ by means of the truncation operator; otherwise, if the size of $\bar{P}_{t+1}$ is less than $\bar{N}$, fill $\bar{P}_{t+1}$ with dominated individuals in P_t and $\bar{P}_t$.
4) **Termination:** If a certain specified condition is satisfied, return nondominated individuals in $\bar{P}_{t+1}$ as the outputs of the system.
5) **Mating selection:** Select parents from $\bar{P}_{t+1}$ based on binary tournament selection.
6) **Variation:** Apply crossover and mutation operators to the parents and set P_{t+1} to the resulting population. Increment generation counter ($t = t + 1$) and go to step 2.

In the fitness assignment step, a strength value $S(i)$ and a fitness $F(i)$ are calculated for each individual i in the population P_t and archive $\bar{P}_t$. $S(i)$ shows the number of dominated solutions by the ith individual. $F(i)$ is the sum of the strength values of its dominators. In the environmental selection step, the nondominated individuals whose fitness is 0 are placed in $\bar{P}_{t+1}$. If the number of such individuals is greater than the archive size, the archive is truncated using a special operator. In this operator, the individual that has the minimum distance to another individual is chosen at each stage. If the nondominated individuals cannot fill the archive exactly, the best dominated individuals are copied to $\bar{P}_{t+1}$.

C. Objective functions

In the proposed method, two objectives, *overall deviation* and *edge value*, are optimized simultaneously using an MOEA. These objectives are important factors of image segmentation.

Overall deviation is defined in equation (3). This equation gives the overall summed distances between the pixels and the center value of the corresponding region (cluster) they belong to. Overall deviation is a measure of the similarity of pixels in the same region.

$$Dev(R) = \sum_{R_k \in R} \sum_{i \in R_k} \delta(i, \mu_k) \tag{3}$$

where R is the set of all regions, μ_k is the centroid of the pixels in the region R_k, and $\delta()$ is the distance function. Overall deviation should be minimized. Minimizing overall deviation roughly increases the number of regions (clusters). We calculate the distance function $\delta()$ defined as a Euclidean distance, and use either RGB or CIE L*a*b* [23] as the color space. The distance function using the RGB color space is defined in equation (4).

$$\delta_{RGB} = \sqrt{\Delta R^2 + \Delta G^2 + \Delta B^2} \tag{4}$$

The distance function using the CIE L*a*b* color space is defined in equation (5). The CIE L*a*b* color space has a uniform chromaticity scale.

$$\delta_{L*a*b*} = \sqrt{\Delta L*^2 + \Delta a*^2 + \Delta b*^2} \tag{5}$$

The second objective, the edge value, is defined in equation (6). This objective evaluates the average distance on boundaries between the regions. This value is a measure of the difference in the boundary between the regions.

$$Edge(R) = -\frac{1}{B} \sum_{i=1}^{N} \sum_{j \in F_i} x_{i,j},$$
$$\text{where } x_{r,s} = \begin{cases} \delta(r,s) & if \nexists R_k : r, s \in R_k \\ 0 & otherwize \end{cases} \tag{6}$$

where N is the number of pixels, F_i indicates the four neighboring pixels of pixel i, B is the number of boundary pixels, and $\delta()$ is the distance function. This edge value is also minimized, which roughly decreases the number of regions (clusters). By minimizing these two objectives, a variety of different image segmentations are generated. In other words, the proposed method returns a range of solutions that have different numbers of regions.

D. Solution selection method

The main objective of this study is to obtain a variety of image segmentation results. However, it is necessary to obtain one solution suitable for a circumstances. In MOCK, the number of clusters is determined based on the Gap statistic. However, its determination scheme requires random data clustering in several trials. Therefore, the computational time for determining the number of clusters is high [17].

In the proposed approach, we use a simple heuristic method to select one solution and find the “knee” of the original Pareto solutions. The most interesting solutions of

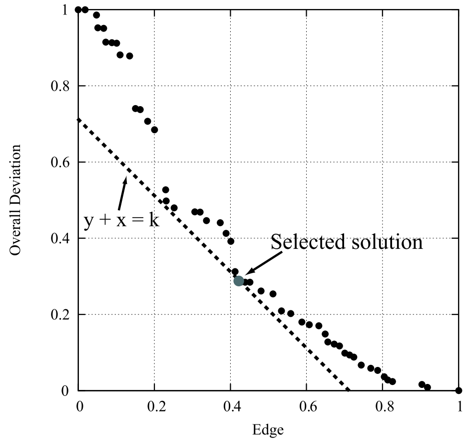


Fig. 2. Example of the selection scheme in the proposed method.

TABLE I
SETTINGS OF PARAMETERS USED IN THE EXPERIMENTS.

Parameters	*Values*
Gene length	image size (96 × 96[pixels] or 128 × 96[pixels])
Color space	RGB or CIE L*a*b*
MOEA model	SPEA2
Number of generations	300
Population size	50
Archive size	50
Crossover	Uniform crossover
Crossover rate P_c	0.7
Mutation rate P_m	0.0001
Constraints	$1 <$ region num < 50 region size > 100 [pixels]

the Pareto-optimal front are those where a small improvement in one objective would lead to a large deterioration in at least one other objective. These solutions are sometimes also called knees [24].

Our selection scheme is as follows:

1) Normalize the fitness value of all the solutions to the range of $[0.0, 1.0]$ using the original Pareto solutions.
2) Calculate the sum of "normalized $Dev(R)$" and "normalized $Edge(R)$", and select the solution with the minimal value as the best solution.

In other words, the solution that has the minimal distance from the line $y = -x$ is selected as the adequate solution in this selection scheme. An example of our selection scheme is shown in Figure 2.

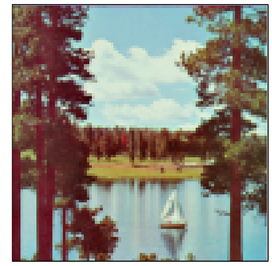
IV. EXPERIMENTS AND RESULTS

In this section, we apply the proposed method to image segmentation problems, and confirm that a variety of image segmentation results are obtained. In addition, we select one solution from the original Pareto solutions.

A. Settings of the experiments

The parameters used by the proposed method are shown in Table I. We use either RGB or CIE L*a*b* as the color space. We constrain the number of pixels in one cluster to be less than 100 to prevent the existence of very small regions (clusters). The original color images used in the experiments are shown in Figure 3. In these experiments, we use four images; (a) pepper, (b) sailboat, (c) terra, and (d) paprika.

(a) pepper



(b) sailboat

(c) terra

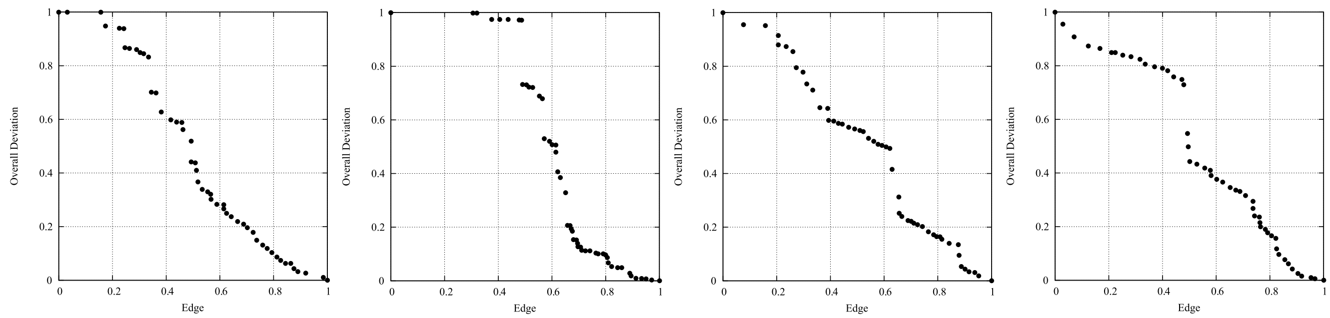
(d) paprika

Fig. 3. Original color images used in the experiments.

B. Results and discussion

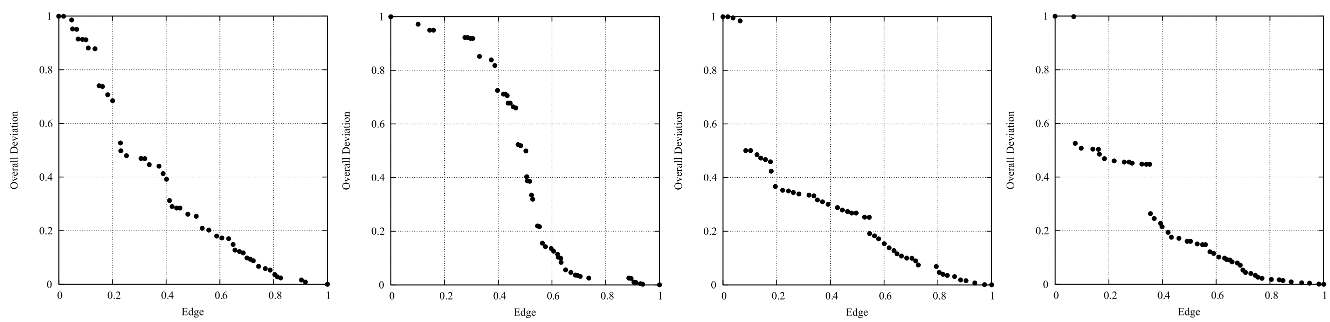
The obtained Pareto solutions by the proposed method are shown in Figures 4 and 5. The values of the axes are normalized. Figure 4 shows the obtained Pareto solutions using the RGB color space, and Figure 5 shows those using CIE L*a*b* color space. The obtained solutions of each image have different Pareto front. The solutions obtained with the RGB and CIE L*a*b* color spaces are also different. Although the shapes of the Pareto front are different, a wide range of solutions is found in all cases. Therefore, we conclude that the evolutionary algorithm is efficient. We use SPEA2 as the MOEA model in the experiments. It would be interesting to analyze the behaviour of NAGA-II [25] and other MOEA models. Nebro et al. show that the convergence speed of SPEA2 is slowest for test problems [26].

Another objective of this study is to verify whether various



(a) pepper (b) sailboat (c) terra (d) paprika

Fig. 4. Obtained Pareto solutions by the proposed method using the RGB color space. The horizontal axis represents normalized edge values, and the vertical axis represents normalized overall deviation.



(a) pepper (b) sailboat (c) terra (d) paprika

Fig. 5. Obtained Pareto solutions by the proposed method using the CIE L*a*b* color space. The horizontal axis represents normalized edge values, and the vertical axis represents normalized overall deviation.

image segmentation results are obtained. Figure 6 shows examples of the image segmentation results in the case of the "pepper" image using the RGB color space. The black lines in the images indicate the boundaries of the regions. The results show that various image segmentations (various numbers and shapes of regions) are obtained. Also, the image segmentations are considered to be relatively good.

Next, we select one solution from the Pareto front using the selection scheme. Figure 7 shows the selected solutions. In general, these image segmentations are considered good results. In the case of "paprika" the three peppers are partitioned into a single region when the CIE L*a*b* color space is used. Thus, the best solution is not selected in this case.

The results of image segmentation using the k-means method, which is one of the clustering methods, are shown in Figure 8. In this case, we apply the k-means method to the "paprika" image using the RGB color space. In this method, the user has to choose an appropriate number of clusters. Image segmentation is overdone when there are many clusters. Also, the k-means method assigns regions separated on the image plane to the same region. It is confirmed that the proposed method obtains good solutions compared with the k-means method.

In the experiment, we only compared the proposed method with k-means method. We might have to compare the proposed method with other methods such as graph partitioning based methods [4], [5] and ensemble based methods [27].

V. Conclusions and Future Work

In this paper, we proposed evolutionary image segmentation based on multiobjective clustering. In the proposed method, two objectives, *overall deviation* and *edge value*, are optimized simultaneously using an MOEA. These objectives are important factors in image segmentation. The proposed method finds various solutions (image segmentation results) by an evolutionary process. We applied the proposed method to several image segmentation problems and confirmed that various solutions are obtained. In addition, one solution was selected from the original Pareto solutions by a simple heuristic method. We also show that a good image segmentation result is selected.

In future work, we will apply the proposed method to other types of images, such as medical images. Moreover, we plan to examine more appropriate objectives for image segmentation.

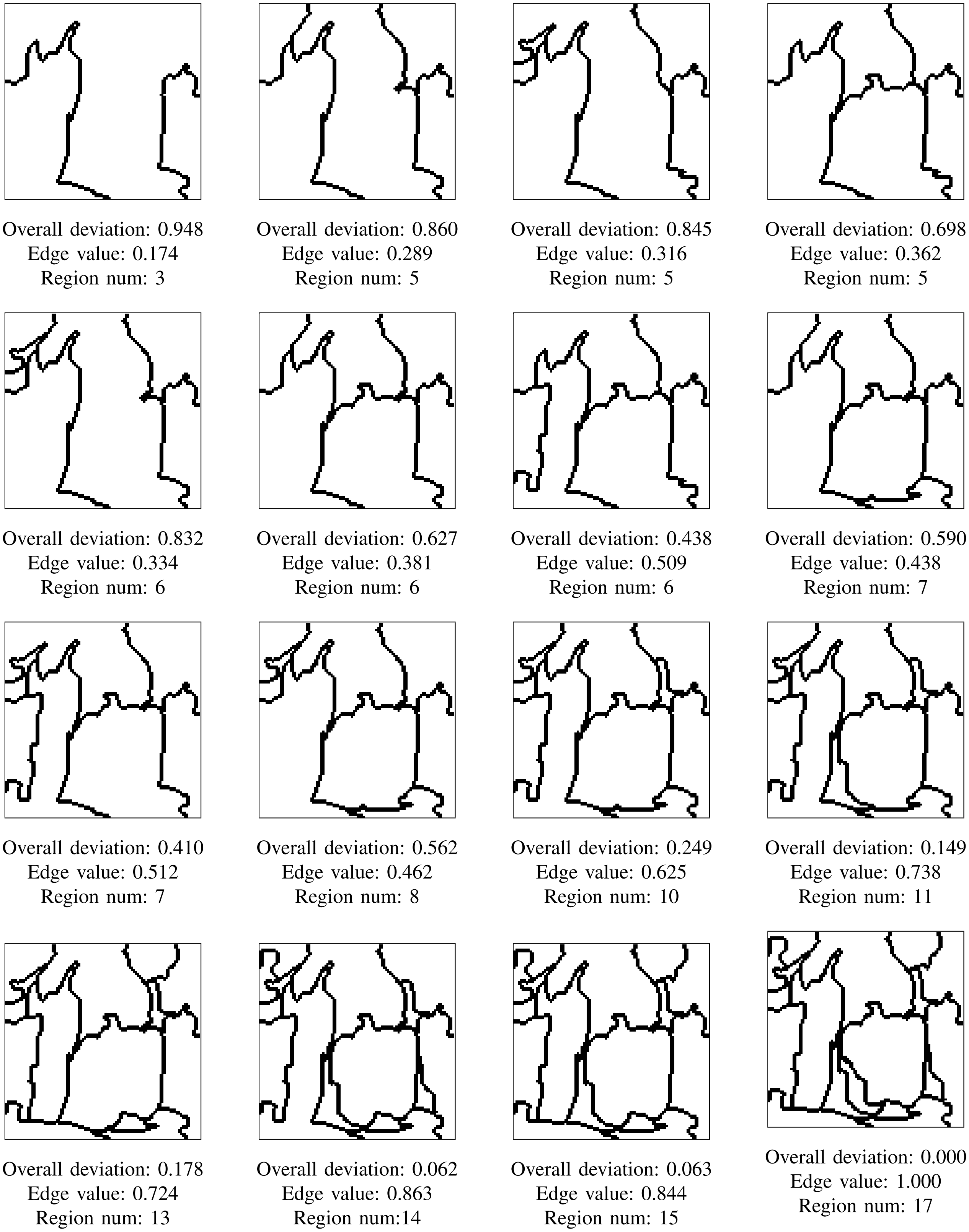
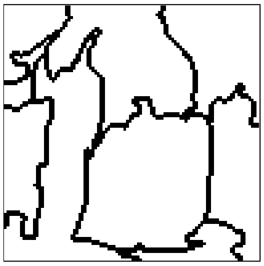
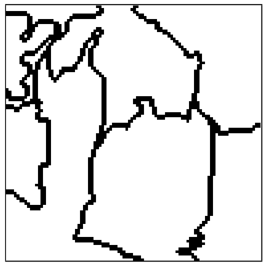


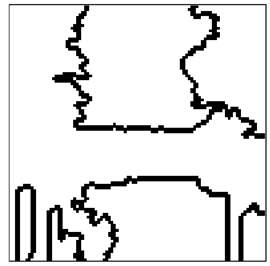
Fig. 6. Examples of the image segmentation results in the case of the "pepper" image using the RGB color space. The black lines indicate the boundaries of the regions. Each result also shows the normalized overall deviation, normalized edge value, and number of regions.



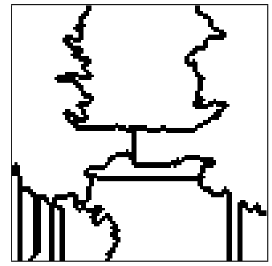
pepper
Color space: RGB
Overall deviation: 0.302
Edge value: 0.567
Region num: 9



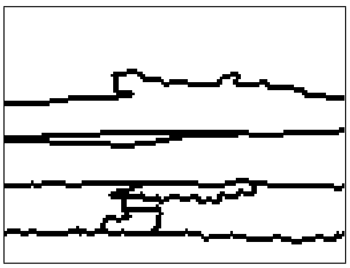
pepper
Color space: CIE L*a*b*
Overall deviation: 0.290
Edge value: 0.422
Region num: 9



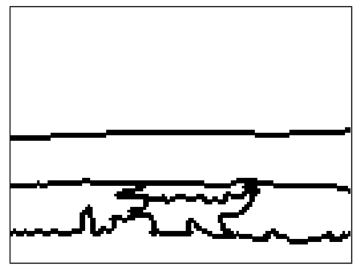
sailboat
Color space: RGB
Overall deviation: 0.114
Edge value: 0.710
Region num: 7



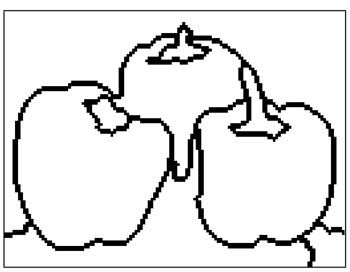
sailboat
Color space: CIE L*a*b*
Overall deviation: 0.056
Edge value: 0.651
Region num: 10



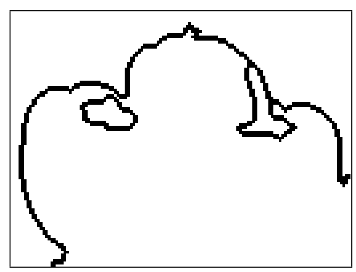
terra
Color space: RGB
Overall deviation: 0.240
Edge value: 0.666
Region num: 9



terra
Color space: CIE L*a*b*
Overall deviation: 0.366
Edge value: 0.195
Region num: 7



paprika
Color space: RGB
Overall deviation: 0.042
Edge value: 0.883
Region num: 9



paprika
Color space: CIE L*a*b*
Overall deviation: 0.526
Edge value: 0.076
Region num: 4

Fig. 7. Selected image segmentation results obtained by the proposed method. The black lines indicate the boundaries of the regions. The normalized overall deviation, normalized edge value, and number of regions are also given for each selected solution.

REFERENCES

[1] John C. Russ. *The Image Processing Handbook, Fifth Edition*. CRC Press, 2006.

[2] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2002.

[3] A. Tremuau and N. Borel. A region growing and merging algorithm to color segmentation. *Pattern Recognition*, 30(7):1191–1203, 1997.

[4] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 731–737, 1997.

[5] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[6] M. Suk and T. H. Cho. Segmentation of images using minimum spanning tree. In *Applications of Digital Images Processing V, Proc. SPIE 397*, pages 180–185, 1983.

[7] Ying Xu and Edward C. Uberbacher. 2D image segmentation using minimum spanning trees. *Image and Vision Computing*, 15(1):47–57, 1997.

[8] B. Bhanu, Sungkee Lee, and J. Ming. Adaptive image segmentation using a genetic algorithm. *IEEE Transactions on Systems, Man and*

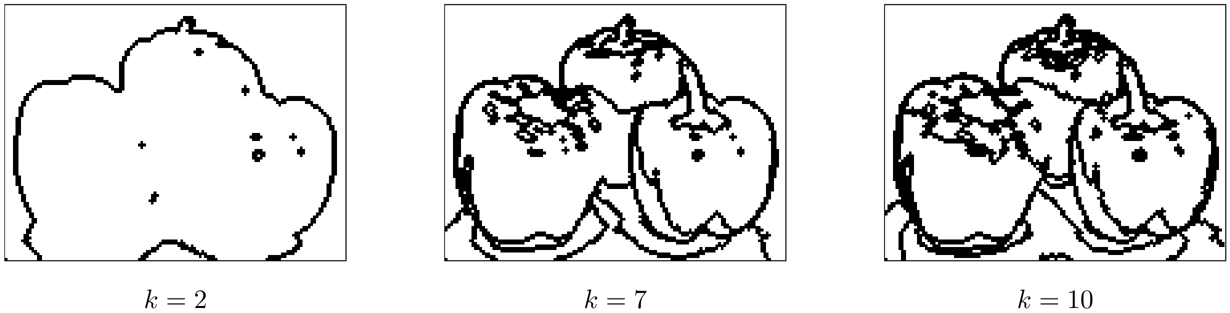


Fig. 8. Image segmentation results in the case of "paprika" image using k-means method ($k = 2, 7, 10$). RGB is used as the color space. The black lines in the images indicate the boundaries of the regions.

Cybernetics, 25(12):1543–1567, 1995.

[9] Motohide Yoshimura and Shunichiro Oe. Evolutionary segmentation of texture image using genetic algorithms towards automatic decision of optimum number of segmentation areas. *Pattern Recognition*, 32(12):2041–2054, 1999.

[10] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.

[11] Riccardo Poli. Genetic programming for image analysis. In John R. Koza, David E. Goldberg, David B. Fogel, and Rick L. Riolo, editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*, pages 363–368, Stanford University, CA, USA, 28–31July 1996. MIT Press.

[12] Julia Handle and Joshua Knowles. Multiobjective clustering with automatic determination of the number of clusters. Technical Report TR-COMPSYSBIO-2004-02, Department of chemistry, UMIST, Manchester, August 2004.

[13] Julia Handle and Joshua Knowles. An evolutionary approach to multiobjective clustering. *IEEE Transactions on Evolutionary Computation*, 11(1):56–76, 2007.

[14] Julia Handle and Joshua Knowles. Improvements to the scalability of multiobjective clustering. In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC 2005)*, pages 2372–2379. IEEE Press, 2005.

[15] Julia Handle and Joshua Knowles. Multiobjective clustering around medoids. In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC 2005)*, pages 632–639. IEEE Press, 2005.

[16] Julia Handle and Joshua Knowles. An investigation of representations and operators for evolutionary data clustering with a variable number of clusters. In *Proceedings of the 9th International Conference on Parallel Problem Solving from Nature (PPSN IX)*, pages 839–849, 2006.

[17] Nobukazu Matake, Tomoyuki Hiroyasu, Mitsunori Miki, and Tomoharu Senda. Multiobjective clustering with automatic k-determination for large-scale data. In *Proceedings of the Genetic and Evolutionary Computation Conference 2007 (GECCO 2007)*, pages 861–868, London, England, July7–11 2007.

[18] Gul Nildem Demir, A. Sima Uyar, and Sule Oguducu. Graph-based sequence clustering through multiobjective evolutionary algorithms for web recommender systems. In *Proceedings of the Genetic and Evolutionary Computation Conference 2007 (GECCO 2007)*, pages 1943–1950, London, England, July7–11 2007.

[19] Kalyanmoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 2001.

[20] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edition*. Springer, 2007.

[21] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a dataset via the Gap statistic. Technical report, Stanford University, 2000.

[22] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Zurich, Switzerland, May 2001.

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

[24] Jürgen Branke, Kalyanmoy Deb, Henning Dierolf, and Matthias Osswald. Finding knees in multi-objective optimization. In *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, pages 722–731, 2004.

[25] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[26] Antonio Jesús Nebro, Juan José Durillo, Carlos A. Coello Coello, Francisco Luna, and Enrique Alba. A study of convergence speed in multi-objective metaheuristics. In *Proceedings of the 10th international conference on Parallel Problem Solving from Nature (PPSN X)*, pages 763–772. Springer-Verlag, 2008.

[27] Yuan Jiang and Zhi-Hua Zhou. SOM ensemble-based image segmentation. *Neural Processing Letters*, 20(3):171–178, 2004.

CODE

The python implementation is available at `https://github.com/shirakawas/mock-segmentation`.

ERRATA

Equation (6)

- *In the original paper:*
  The second objective, the edge value, is defined in equation (6). This objective evaluates the overall summed distances on boundaries between the regions. This value is a measure of the difference in the boundary between the regions.

$$Edge(R) = -\sum_{i=1}^{N} \sum_{j \in F_i} x_{i,j},$$
$$\text{where} \;\; x_{r,s} = \begin{cases} \delta(r,s) & if \nexists R_k : r, s \in R_k \\ 0 & otherwize \end{cases}$$

  where N is the number of pixels, F_i indicates the four neighboring pixels of pixel i, and $\delta()$ is the distance function.
- *In this revised paper (corrected version):*
  The second objective, the edge value, is defined in equation (6). This objective evaluates the average distance on boundaries between the regions. This value is a measure of the difference in the boundary between the regions.

$$Edge(R) = -\frac{1}{B}\sum_{i=1}^{N} \sum_{j \in F_i} x_{i,j},$$
$$\text{where} \;\; x_{r,s} = \begin{cases} \delta(r,s) & if \nexists R_k : r, s \in R_k \\ 0 & otherwize \end{cases}$$

  where N is the number of pixels, F_i indicates the four neighboring pixels of pixel i, B is the number of boundary pixels, and $\delta()$ is the distance function.