Received: November 25, 2021. Revised: January 6, 2022.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Evolutionary Optimization of Geometrical Image Contour Detection

Salwa Khalid Abdulateef[1]*

[1]*Department of computer science, Computer Science and Mathematics College. Tikrit University, Iraq*
* Corresponding author's Email: khalid.salwa@tu.edu.iq

Abstract: The image vectorization based on geometrical contour representation is an active research topic. It involves approximating the contour of the objects inside the image using geometrical definition or estimation. Typical approach of estimating the parameters of the geometrical model is fitting-based estimation. However, the noise that exists in the image causes degradation in the performance. In order to overcome this degradation, we propose integrating evolutionary based optimization. In this article, the design of the operators of the genetic optimization for improving the contour detection results of segments-based estimation of the contour is proposed. The operator includes both crossover and mutation. The evaluation shows that Structural Similarity Index Metrics (SSIM) measure has increased after applying it from 0.64 to 0.77 on UEC Food 100 Dataset and from 0.54 to 0.66 after applying it to UEC Food 256 Dataset.

Keywords: Evolutionary approach, Optimization, Contour detection, Geometrical model.

1. Introduction

The process of splitting a digital image into numerous segments or pixels is known as image vectorization (IV) [1]. The goal of IV is to provide a simple representation of the image that has a strong meaning and can be easily examined [2, 3]. IV can also assign a label to each pixel in the image, and pixels with the same label will have similar properties [4]. IV has the capacity to locate objects and boundaries in images, such as lines and curves. A part of the functions of IV is the boundary detection or the contour splitting [5].

The IV produces a series of segments that can be used to cover the entire image. IV has gotten a lot of interest in recent decades since it may be used to solve a variety of problems, including object detection [6], occlusion boundary estimate in motion or stereo systems [7], image compression [8], picture editing [9] and more. Color IV is a different type of IV that divides a chromatic image into homogenous and related sections based on color, texture, and their combination [10].

Optimization is an essential process for increasing the performance of many algorithms [11]. One category of the optimization algorithms is the meta-heuristic searching optimization [12]. It operates based on generating random solutions and improving using heuristics until reaching the optimal solution. Meta-heuristic algorithms are categorized under two categories: the first one is evolutionary [13, 14], and the second one is swarm [15]. The concept of evolutionary algorithms is to develop the solutions based on mating process that generates new generation from current one based on crossover and mutation [16]. The concept of swarm algorithms is to develop the solutions based on moving the solutions toward a best solution using moving operator. In both evolutionary and swarm algorithm mutation is used to enable an avoidance of local minima [17]. In this article we are interested in improving the contour detection by developing evolutionary algorithm. The design of the solution, the objective function and the operators of genetic is done for this purpose. The genetic optimization (GA) is incorporated inside a geometrical based contour detection named orientation map. We select GA because of its effectiveness in performing optimization based on evolutionary searching. In addition, the crossover and mutation in GA are subject to modification based on the application need. The following is how the rest of the article is organized. The literature review is included in section 2. Next, section 3 provides the methodology. Afterwards, the experimental evaluation is given in section 4. Lastly, section 5 concludes with a conclusion and recommendations for future work.

2. Contributions

This article provides several contributions, we present them as follows:

1. It uses straight lines based contour fitting as a mathematical model for objects vectorization in images. The algorithm provides a simple but efficient vectorization.
2. It uses genetic algorithm for improving the effectiveness of the provided algorithm by using the slops of the straight lines as decision variables in the chromosome design.
3. It suggests as an objective function for the genetic the sum of distances among each corrupted line segment and the two adjacent line segments in the object contour.
4. It provides a thorough evaluation based on comparing the image after vectorization and the original raster images using Structural Similarity Index Metrics (SSIM).

3. Literature survey

The literature contains numerous approaches for contour vectorization. A contour vectorization comprising of two primary phases has been devised by [18]. The first is to use the intensity step to extract the image's level set contours, which defines the intensity fluctuations between two consecutive intensity levels. After that, polygonalization algorithms can extract and pick the geometric information. The vectorial picture reconstruction from these contours is then carried out. A method for vectorization of such inputs specifically designed to coincide with human perception was proposed in the work of [19]. The method is based on a simultaneous spline fitting and corner detection method that combines learning metric with perception-driven algorithmic discontinuity analysis to approximate human perception of boundary discontinuities on raster inputs. Local cues provided by the learned metric are balanced with global cues produced by balancing simplicity and continuity expectations in the final technique. The method connects the corners using simple, continuous curves that capture input regularities, given the finalised set of corners. An algorithm for generating a PDF containing vectorized text symbols and drawing images from scanned documents was developed by [20]. Segmentation of the document into text/drawing areas and backdrop, translation of symbols to sequences of lines and bezier curves, and storage of the compressed background and foreground in PDF format are all part of the multi-stage approach. Vectorization technique based on trainable COSFIRE filters for the detection of junction points and subsequent tracing of line pathways to construct a topology graph as a representation of the sketched object shape has been created in the study [21]. A vectorization approach that uses mesh colours as a vector primitive for image vectorization was published in the work of [22]. Mesh colours have been found to improve rendering performance and texture detail. They also enable a simpler and more efficient construction of meshes of curved triangular patches, which are constrained in this case by the picture feature extraction algorithm, due to their flexibility. In the work of [23] object shape detection using a technique based on hough transform introduced. The algorithm performs hough transform for circle fitting for circle model where the goal is to detect the center and the radius of the best circle. Each circle is corresponding to one berry. Some researchers have incorporated optimization approaches for correction of contour detection. The authors in [24] have been used a computational model to detected orientation of the contour in the objects and eliminate the insignificant elements. The orientation map provides an analytical representation of the image information instead of pixel representation. This application has some problems with outliers and discontinuity. It is meant by outliers any additional short line to the object contour while it is meant by discontinuity any corrupted short line because of wrong parameters. In [25], the researchers used a genetic algorithm to solve combination optimization problems of contour features that were extracted using a contour feature with a new kind of gene encoding based on graph structure, while [26] employed a graph-based genetic representation and improved genetic procedures to combine short line segments into lengthy contour lines. This is a simplified version of the approach. Short line segments are incorporated into large line segments using this GA-method; however, it does solve the fragments that occur between the lines. In the work of [27] using deleting, joining, and fitting, a method

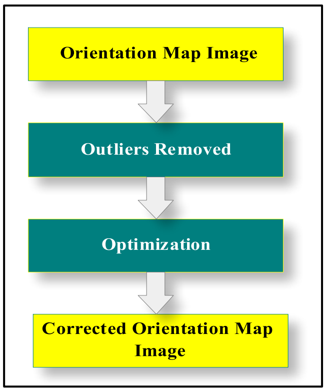


Figure. 1 Steps of an enhanced orientation map

for obtaining closed object outlines was developed. To begin, pre-processing processes are used to generate a set of contour segments from an image. Second, an eight-neighborhood discriminant is recommended for identifying and removing non-target curve portions. Finally, to link the fractured curve segments to produce a closed object contour, a connection algorithm based on closeness and continuity of closed contours is proposed. Finally, to make the closed object contour more consistent with the item's genuine contour, a B-spline curve-fitting approach is presented. The authors put their method to the test using a variety of images. There are still some limitations for the method, such as the appropriate values of some parameters should be present through experiments.

Overall, it is found that vectorization of images is a research topic and it has been performed by various mathematical and geometric models. However, least work has exploited the usage of optimization algorithm for obtaining effective and efficient vectorization. Lacking of optimization of causes degradation in the performance and sensitivity to the parameters.

4. Methodology

This section presents the developed methodology for this article. It starts with presenting the orientation map as input. Next, we provide the outliers removal. Afterwards, the optimization is presented. Lastly, we get the corrected orientation map image as output. A diagram of the developed methodology is Fig. 1.

4.1 Problem definition

Assuming that we have a pixel image that contains one or multiple objectives. The goal is convert the control of the objects inside the image to geometrical representation that enables restoring the content of the image with minimum differences between the original image and the geometrically described image. For this goal, we use a geometrical model named orientation map.

4.2 Orientation map

The orientation map [24] is a geometrical based representation using fitting approach. The fitting is performed on each small segment of the contour by estimating two parameters, namely, the angle and the offset. A vector of the two parameters for each small segment is presented to the evolutionary based optimization algorithm that is elaborated in the subsequent phases.

4.3 Outliers removal

Outliers need to be removed before feeding the image to the remaining blocks of recognition. In removing the first type of outliers from the orientation map image, the image is checked first to find short lines for different nodes that shape the line segments. Line segment below the threshold in terms of number of pixels requirement are then eliminated from the image. The image is converted to small lines according to the pixels that exist inside the sub-window. Before converting any sub-window to small lines, the number of pixels inside the sub-window is examined. Then, every set of connected small lines segments are converted to straight lines in case the number is more than the pre-defined threshold N_{pre}.

A search process within adjacent subgraphs is performed to find closed contours and remove outliers of the second type. A contour is defined as “closed” if the distance d between two adjacent nodes in the contour is less than the threshold of connectivity T_{con}. If d is greater than T_{con}, then the nodes are removed. Hence, the unnecessary lines attached to the contour are removed. The procedure for this method is presented in algorithm 1. The algorithm indicates that the goal of the function find “ConnectedGraphs” is to find the graph representation of the orientation map image. The graph representation is given as list of graphs with each graph a connected one.

4.4 Genetic-based optimization on orientation map

Wrong orientation is defined as a discontinuity (wrong slop of lines). A GA has been introduced to address the problem of wrong orientations. GA is a type of meta-heuristic optimization approach. Meta-heuristic refers to a nature-inspired approach

Algorithm 1 Removing outliers from the output of the original orientation map [24]

```
Define: $N_{pre}$      //Pre chosen number
        $d$            //Distance between two adjacent nodes
        $T_{con}$      //Threshold of connectivity
Input:  $Im$           //Orientation map image
Output: Corrected Orient map Image
Begin
    GraphsList=Find_ConnectedGraphs(Im)   // This function extracts connected graphs and the number
                                             of nodes and copies them in GraphList
    For i=1:length(GraphList)
        If(GraphList(i).Number< $N_{pre}$)        //Type 1 of error
            GraphList(i).Delete()
        Endif
    EndFor
    For i=1:length(GraphList)
        G= GraphsList(i);
        For j=1: G.Number
            d=Dist(GraphList(j), GraphList(j+1));
            If(d) > $T_{con}$)                     //Type 2 of error
                GraphList.Add(G(1:j))
                GraphList.Add(G(j+1:end))
            End If
        EndFor
    EndFor
End
```

applied to find solutions for hard combinatorial optimization problems [28]. Therefore, the GA optimization approach is used to search the solution space for the best solution among wide range of candidate solutions.

In the context of the orientation map, the sub-windows were converted into representations of straight lines. Sub-window refers to a small portion of the image. However, the slope and offset of the straight line is not the best in terms of the global border representation of the object. Changing the slope and the offset will result in a more fitting line to the border of the object of the item. However, many possibilities of changing the lines to obtain the best representation of the borders of the items exist. Searching for the best possibilities is a type of optimization problem that can be solved using optimization approaches. In the next subsections, the genetic chromosome encoding, objective function, crossover, and mutation for the genetic optimization in the present study are presented.

4.4.1. Genetic chromosome encoding

In a GA, a population of solutions or individuals is encoded in the form of chromosomes as shown in the Eq. (1). Individuals are initialized randomly from a large search space. This population is evolved from one generation to the other using the genetic processes of crossover, mutation and selection. The selection operation is based on an evolutionary function also known as objective function. The operations of selection, crossover and mutation are applied continually for individuals until the termination condition is satisfied or a fixed number of generations are reached (300 generation as shown in the present experiments). The termination condition is set when the fitness value is no longer improving.

In the case of wrong orientations, the corrupted lines responsible for the discontinuity were first identified. Then, the slopes of the corrupted lines were assigned as chromosomes for the GA (encoding equation) as Eq. (1).

$$Chromosome = (k_1, k_2, \ldots, k_m) \tag{1}$$

Where m is the number of corrupted line segments, k_1 the slope of the first corrupted line segment, k_2 slope of the second corrupted line segment, and k_m is the slope of the m^{th} corrupted line segment.

4.4.2. Objective function

The objective function is defined as the sum of distances among each corrupted line segment and the two adjacent line segments in the object contour as in Eq. (2).

$$Fitnessvalue = \sum_{i=1}^{m} distSum_i \tag{2}$$

Algorithm 2 solution space and objective function

```
Input   Graph G
        Population Size Ps
        Number of Iteration N
Output  Corrected Graph Gc
Begin
1: P=GenerateInitialPopulation()//P = Si, i=1,2, . Ps,Si=K(1),K(2),.K(G.NumberofNodes)where Si represents the solution
2: For i=1:N
3:     For each Si from P        // calculating the fitness value  for each solution in the population
4:         FitnessValue=0      // initialization of the fitness value with zero
5:         For j=1:G.NumberofNodes-1   //  for each node of thecorresponding solution
6:             FitnessValue= FitnessValue+MinDist(G(j),G(j+1),S(j),S(j+1))
7:         EndFor
8:         S(i).AddFitness(FitnessValue)     // collecting the fittness values of the solution
9:     EndFor
10:    Go to 2
11: EndFor
End
```

Algorithm 3 Distance function

```
Function Distance=MinDist(NodeA,NodeB,KA,KB)
    NodeAc=ChangeSlope(NodeA,KA)             // change the slope of the straight line of the node
    NodeBc=ChangeSlope(NodeB,KB)             // change the slope of the straight line of the node
    Distance=CalDistance(NodeAc,NodeBc)      //the minimum distance between the two lines
End

Function ChangeSlope(Node,m)
    Node.slope=m
End
```

Where distSum is defined as the sum of the distance between the corrupted line segment and the two adjacent line segments, i denotes the index and m is the number of nodes. The algorithm 2 presents the procedure for calculating the fitness values of the pool of solutions denoted by population *Ps*.

4.4.3. Crossover operator

The goal of the crossover is to combine two solutions from the pool of solutions to produce an offspring. The process is repeated until the generated number of solutions obtained is equal to the desired number of offspring in the new generation. Typically, a crossover is performed in a random manner without customization to the application nature. However, in this study, it can develop an application-dependent crossover that makes it more efficient in finding the solution. The combination of two solutions is based on the genes that represent the slopes of the straight lines, which increases its effectively in finding the optimal solution that corrects the orientation map. Another difference is that the crossover is based on either a one-point crossover or two-points, which adds more possibilities for generating better offspring. This method was selected because it performed an efficient search.

The crossover is presented in Fig. 2 and shows two solutions each of which is composed of m segments as an example: $S_A = (s_1{}^a, s_2{}^a \dots s_m{}^a)$, $S_B = (s_1{}^b, s_2{}^b \dots s_m{}^b)$. Each segment is connected to the next segment through a connection point as

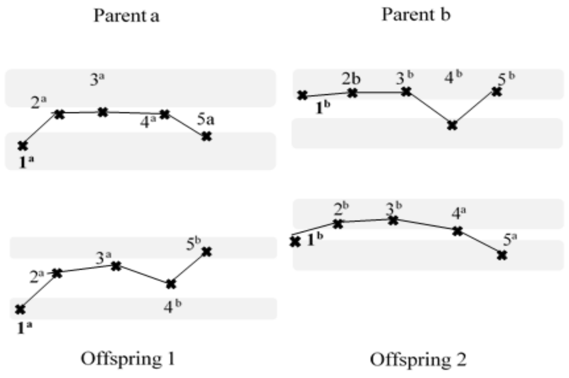


Figure. 2 One-point crossover

Algorithm 4 Crossover algorithm

```
Function [C1,C2]=Crossover(P1,P2)
    M=Length(P1);                      // length of the  chromosome
     k=generateRandom(2,M-1)           // generate the point of cut
     C1a=extract(P1,1,k)               // extract the first part of the first  parent
     C1b=extract(P2,k+1,M)             // extract the first part of  the second parent
     C2a=extract(P1,1,k)               // extract the second part of  the first parent
     C2b=extract(P2,k+1,M)             // extract the second part of the second part
     C1=[C1a C2b]                      // combining the first child
     C2=[C2a C1b]                      // combining the second child
End

Function C=extract(P,Starting,Ending)
   C=zeros(1, Ending-Starting+1)     //initialize the subpart  of the chromosome
   For i=Starting:Ending
      C(i)=P(i)                      // assign the subpart of the chromosome that starts with starting and ends with Ending
   EndFor
End
```

$p_1{}^a = cp(s_1{}^a s_2{}^a)$, where $p_1{}^a$ denotes the connection point between segments $s_1{}^a$ and $s_2{}^a$. Similarly, $p_i{}^a = cp(s_i{}^a s_{i+1}{}^a)$ denotes the connection point between segments $s_i{}^a$ and $s_{i+1}{}^a$. For five segments, the number of points as $p_i{}^a, i \in \{1,2,\dots m-1\}$, where a and b are parents.

Each of the solutions represents one parent. The goal of the crossover is to combine the two parents and produce a number of children, which is referred to as offspring.

The algorithm applies for a single point crossover and for two-points crossovers. For a single point crossover, a random integer number *k ∈{2…m-1}* that indicates the point of the cutting in the crossover operation is generated. Let us assume that this number is generated as k, which means the two parents will be decomposed into two parts:

$S_A{}^1 = (s_1{}^a, s_2{}^a, \dots, s_k{}^a)$, $S_A{}^2 = (s_{k+1}{}^a, \dots s_m{}^a)$, $S_B{}^1 = \left(s_1{}^b, s_2{}^b, \dots, s_m{}^b\right)$, $S_b{}^2 = \left(s_{k+1}{}^b, \dots, s_m{}^b\right)$, see Eq. (3) and Eq. (4).

$$C_1 = \left(s_1{}^a, s_2{}^a, \dots, s_k{}^a, s_{k+1}{}^b, \dots, s_m{}^b\right) \tag{3}$$

$$C_2 = \left(s_1{}^b, s_2{}^b, \dots, s_k{}^b, s_{k+1}{}^a, \dots, s_m{}^a\right) \tag{4}$$

The crossover operation is denoted as *$\oplus$^1*, thus, for a single point crossover as Eq. (5).

$$S_A \oplus^1 S_B = \{C_1, C_2\} \tag{5}$$

The same crossover operation can be done using a two-points crossover, in such case Eq. (6).

$$S_A \oplus^2 S_B = \{C_1, C_2, C_3, C_4\} \tag{6}$$

The pseudo code of the crossover algorithm is presented in algorithm 4. The algorithm starts by generating a random number that represents the partitioning point. Once the number is generated, the parts of the offspring are extracted using the function called extract (P, Starting, Ending). The function copies only the genes corresponding to the indices between the starting and the ending to combine a possible part of the new child. Once the parts are generated, they will be combined because each child includes two parts, one from each parent while preserving the constraint that the length of the child is the same as the length of the parent.

In the classical crossover, the value of the chromosome represents any number while in the present study, it represents a slope. Four possible cases of the offspring can be observed when one-point crossover is used and nine possible cases when two-point crossover is used.

4.4.4. Mutation

The goal of the mutation is to select the percentage of solutions in the generation and to change the selected parts of their genes to add more exploration in the solution space, thereby enabling the searching algorithm to have both an exploration and exploitation. In this study, it can be assumed that there is a solution $S_A = (s_1{}^a, s_2{}^a \dots s_m{}^a)$. This solution can be mutated by selecting k genes and changing them according to the probability density function as in Eq. (7).

$$M(s_i{}^a) = x, where\ x \sim N(s_i{}^a, \sigma) \tag{7}$$

Where N indicates to normal distribution, σ

Algorithm 5 Mutation algorithm

```
M=Mutation(C)
 Start
 Define SIGMA                // standard deviation of the mutation
 M=Length(C)                 // length of the chromosome that will be mutated
 k=generateRandom(1,M) // index of the gene that will be changed inside the chromosome,  random is uniform
                             distribution
 C(k)=generateRandomNormal(C(k),SIGMA)   // generate the new gene according to normal distribution
 End
```

denotes standard deviation, the higher value of σ implies more exploration, M denotes mutation operation and x is the value of the gene after mutation.

In the present study, the mutation is performed using the probabilistic approach that changes the genes according to normal distribution and preserves the main shape of the contour, thereby providing more opportunities for the defects inside it to be improved. In the traditional mutation, the changed value of the gene can be replaced with any new value inside the valid range. While in this study, the changed value is replaced by a random number that follows a normal distribution with the old one being the expected value and standard deviation is that which was determined previously. This step prevents the corruption of the contour with big changes in the slopes of the segments because the goal is to perform tiny changes to correct the contour. The pseudo code of the mutation is presented in algorithm 5.

5. Experimental evaluation and results

The setting of the GA parameters is introduced in table 1, before the experimental results are shown. The selection of the parameters was performed using a trial and error method.

Intensive evaluation of the proposed algorithm was undertaken, and various types of images were used for evaluation as presented in the datasets. This experiment used two datasets, UEC100 images [29] and UEC256 [30].

Table 1. Configuration of parameters for GA

Parameters	Value
Chromosome type	Continues
Chromosome	variable
Population size	50
Crossover probability	0.8
Selection method	Roulette wheel
Mutation probability	0.1

Structural similarity index metrics (SSIM) is used to measure the similarity between two images, ground truth a and segmented b of common size [31] as Eq. (8).

$$SSIM(a,b) = \frac{(2\mu_a\mu_b + C_1)(2\sigma_{ab} + C_2)}{(\mu_a^2+\mu_b^2 + C_1)(\sigma_a^2+\sigma_b^2 + C_2)} \tag{8}$$

Where: μ denotes to an average of a and b; σ_a^2 refers to variance of a, σ_b^2 is variance of b, σ_{ab} denotes to the covariance a and b and C_1 and C_2 are two positive constants to avoid a null denominator.

The proposed algorithm was applied on two common datasets, namely, UEC food 100 and UEC food 256. Examples of the visual results of connected food items from UEC food 100 dataset are presented in table 2. The images contain different types of food with different color, shape, texture, size and arrangement well as various plates, background, and illuminations. The images were captured from various perspectives. The table shows a raw image in the first column, ground truth produces in second column. The third column shows the orientation map image. Image after removing outliers and discontinuity can show in fourth and fifth column. Removing outliers and discontinuity are the results of the proposed method. In this study, it is noticed that some results of the orientation map have failed in identifying the boundary of some items like the white items (rice) in images No. 2. The reason is the similarity in color to the plate color. Considering that orientation map is used only to assist contour detection, it is adequate to detection the spot in the plate where the items exits even if some items were not detected because the optimization process of energy in the active contour will play the role in identifying the rest of items. Also, it can be seen in images 1 and 3, the algorithm cannot detect internal boundaries between items. Another observation is that the removing of outliers in second column and

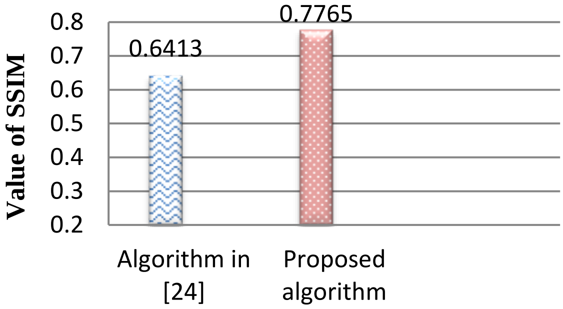


Figure. 3 SSIM results of the proposed method and method in [24] for UEC Food 100

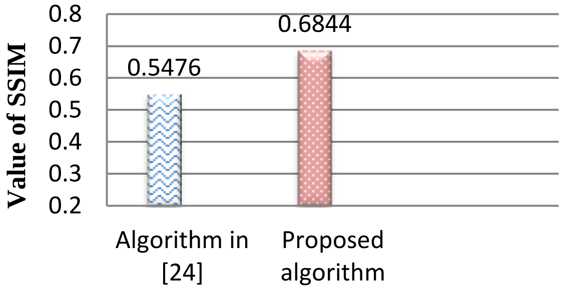
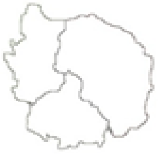
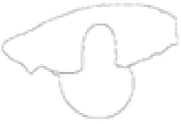


Figure. 4 SSIM results of the proposed method and method in [24] for UEC Food 256 dataset

Table 2. Visual comparison between the proposed algorithm and the algorithm [24] on UEC food 100 dataset

Raw image	Ground truth image	Orientation map image [24]	Proposed algorithm	
			Post processing	Evolutionary enhanced
				
				
				

removing discontinuity in the third column has played a significant role in correcting the boundaries of the detected food items.

Observing, Fig. 3 which shows the SSIM measure between the orientation map in [24] and the ground truth in one side, and between the orientation map after enhancement (proposed method) and the ground truth on the other side. It is obvious that the enhancement has increased the SSIM from 0.64 to 0.77 which is a good improvement. For UEC food 256 dataset, as can be seen in table 3, results of 3 images of different connected items visually. It is presents ground truth of raw image, then result of orientation map image [24]. Also, presents the image after removing outliers and discontinuity. From the table it is noticed that orientation map was acceptable separated between connected food items as images 1 and 2, while, in another image hasn't achieved. However, the results indicate the enhanced orientation map method is more reliable and higher in robustness.

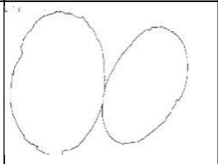
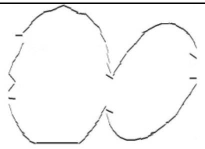
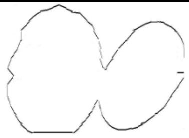
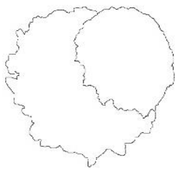
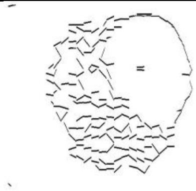
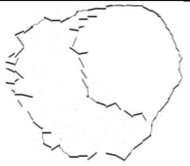
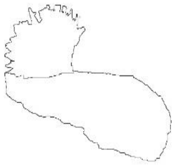
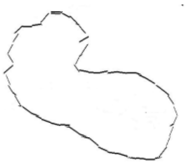
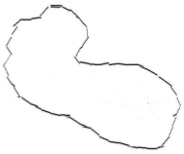
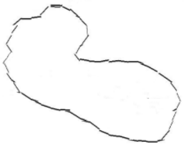
Fig. 4 presents the SSIM measure between the orientation map in [24] and proposed method. It is obvious that the best enhancement of the SSIM from 0.54 to 0.68 which is a good improvement.

6. Conclusion and future works

In this article, an evolutionary algorithm has been designed for enabling optimization of geometrical based contour detection. The algorithm has been applied to segments-based contour estimation for food images. The design has included the chromosome representation, objective function calculation, and two genetic namely, crossover and mutation. The evaluation was

Received: November 25, 2021. Revised: January 6, 2022.

Table 3. Visual comparison between the proposed algorithm and the algorithm [24] on UEC food 100 dataset

Raw image	Ground truth image	Orientation map image [24]	Proposed algorithm	
			Post-processing	Evolutionary enhanced
				
				
				

conducted on two food datasets, namely, UEC food 100 and UEC food 256 dataset. The evaluation has shown an increase of SSIM measure from 0.54 to 0.66 and from 0.64 to 0.77 for UEC food 256 and UEC food 100 respectively. Future work is to generalize this algorithm on other types of geometrical based contour detection.

Conflict of interest

The authors declare no conflict of interest.

Author contributions

Salwa Khalid Abdulateef has contributed to the design and implementation of the research by using Matlab. In addition, result analysis and comparison, preparing and editing draft. Also, providing the necessary data and approved the final version.

References

[1] B. Kim, O. Wang, A. C. Öztireli, and M. Gross, “Semantic segmentation for line drawing vectorization using neural networks”, *Computer Graphics Forum*, Vol. 37, No. 2, pp. 329-338, 2018.

[2] I. Iosifescu, A. Tsorlini, and L. Hurni, “Towards a comprehensive methodology for automatic vectorization of raster historical maps”, *e-Perimetron*, Vol. 11, No. 2, pp. 57-76, 2016.

[3] V. Egiazarian, O. Voynov, A. Artemov, D. Volkhonskiy, A. Safin, M. Taktasheva, and E. Burnaev, “Deep vectorization of technical drawings”, In: *Proc. of European Conf. on Computer Vision*, pp. 582-598, 2020.

[4] Q. X. Meng, W. Y. Xu, H. L. Wang, X. Y. Zhuang, W. C. Xie, and T. Rabczuk, “DigiSim—an open source software package for heterogeneous material modeling based on digital image processing”, *Advances in Engineering Software*, Vol. 148, p. 102836, 2020.

[5] K. Rajan, H. O. Brinkhaus, A. Zielesny, and C. Steinbeck, “A review of optical chemical structure recognition tools”, *Journal of Cheminformatics*, Vol. 12, No. 1, pp. 1-13, 2020.

[6] Y. Wang, S. Zorzi and K. Bittner, “Machine-learned 3D Building Vectorization from Satellite Imagery”, In: *Proc of the IEEE/CVF Conf.on Computer Vision and Pattern Recognition*, pp. 1072-1081, 2021.

[7] M. Rosen, K. W. Godin, and N. Raghuvanshi, “Interactive sound propagation for dynamic scenes using 2D wave simulation”, *Computer Graphics Forum*, Vol. 39, No. 8, pp. 39-46, 2020.

[8] S. M. Hilles, “Sofm And Vector Quantization For Image Compression By Component”, In: *Proc. of International Conference on Smart*

Computing and Electronic Enterprise (ICSCEE), pp. 1-6, 2018.

[9] S. A. Mihelic, W. A. Sikora, A. M. Hassan, M. R. Williamson, T. A. Jones, and A. K. Dunn, “Segmentation-Less, Automated, Vascular Vectorization”, *PLoS Computational Biology*, Vol. 17, No. 10, e1009451, 2021.

[10] S. Lu, W. Jiang, X. Ding, C. S. Kaplan, X. Jin, F. Gao, and J. Chen, “Depth-aware image vectorization and editing”, *The Visual Computer*, Vol. 35, No. 6, pp. 1027-1039, 2019.

[11] D. M. Vargas, J. M. V. Kinani, and J. D. J. Rubio, “Color-based image segmentation by means of a robust intuitionistic fuzzy c-means algorithm”, *International Journal of Fuzzy Systems*, Vol. 22, pp. 901–916, 2020.

[12] H. N. N. A. Sammarraie and D. N. Jawawi, “Multiple black hole inspired meta-heuristic searching optimization for combinatorial testing”, *IEEE Access*, Vol. 8, pp. 33406-33418, 2020.

[13] S. A. N. Alexandropoulos, C. K. Aridas, S. B. Kotsiantis, and M. N. Vrahatis, “Multi-objective evolutionary optimization algorithms for machine learning: A recent survey”, In: *Approximation and Optimization*, pp. 35-55, Springer, Cham.2019.

[14] W. Deng, H. Chen, and H. Li, “A novel hybrid intelligence algorithm for solving combinatorial optimization problems”, *Journal of Computing Science and Engineering*, Vol. 8, No. 4, pp. 199-206, 2014.

[15] F. A. Zeidabadi, M. Dehghani, and O. P. Malik, “RSLBO: Random Selected Leader Based Optimizer”, *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 5, pp. 529-538, 2021, doi: 10.22266/ijies2021.1031.46

[16] D. Sedighizadeh, E. Masehian, M. Sedighizadeh, and H. Akbaripour, “GEPSO: A new generalized particle swarm optimization algorithm”, *Mathematics and Computers in Simulation*, Vol. 179, pp. 194-212, 2021.

[17] T. Nakane, N. Bold, H. Sun, X. Lu, T. Akashi, and C. Zhang, “Application of evolutionary and swarm optimization in computer vision: a literature survey”, *IPSJ Transactions on Computer Vision and Applications*, Vol. 12, No. 3, pp. 1-34, 2020.

[18] B. Kerautret, P. Ngo, Y. Kenmochi, and A. Vacavant, “Greyscale image vectorization from geometric digital contour representations”, In: *International Conference on Discrete Geometry for Computer Imagery*, Vienna, Austria, pp. 319-331, 2017.

[19] S. Hoshyari, E. A. Dominici, A. Sheffer, N. Carr, Z. Wang, D. Ceylan, and I. C. Shen, “Perception-driven semi-structured boundary vectorization”, *ACM Transactions on Graphics (TOG)*, Vol. 37, No. 4, pp. 1-14, 2018.

[20] I. V. Safonov, I. V. Kurilin, M. N. Rychagov, and E. V. Tolstaya, “Scanned Text Vectorization”, In: *Document Image Processing for Scanning and Printing, Springer*, pp. 181-203, 2019.

[21] A. Bonnici, D. Bugeja, and G. Azzopardi, “Vectorisation of sketches with shadows and shading using COSFIRE filters”, In: *Proc. of the ACM Symposium on Document Engineering*, New York, USA, pp. 1-10, 2018.

[22] G. Hettinga, J. Echevarria, and J. Kosinka, “Efficient Image Vectorisation Using Mesh Colours”, In: *Italian Chapter Conference 2021: Smart Tools and Apps in Graphics. Eurographics Association*, 2021.

[23] E. A. M. Bracamontes, M. E. M. Rosas, M. M. M. Velasco, H. L. M. Reyes, J. R. M. Sandoval, and H. C. D. Avila, “Implementation of hough transform for fruit image segmentation”, *Procedia Engine*ering, Vol. 35, pp. 230–239, 2012.

[24] H. Wei and Y. Ren, “A mathematical model of retinal ganglion cells and its applications in image representation”, *Neural Processing Letters*, Vol. 38, No. 2, pp. 205–226, 2013.

[25] H. Wei, X. S. Tang, and H. Liu, “A genetic algorithm (GA)-based method for the combinatorial optimization in contour formation”, *Applied Intelligence*, Vol. 43, No. 1, pp. 112-131, 2015.

[26] H. Wei, L. Hang, and T. Fuyu, “A GA-based solution for the combination optimization in the contour formation”, In: *IEEE 25th International Conference on Tools with Artificial Intelligence*, Herndon, VA, USA, pp. 292–299, 2013.

[27] F. Gao, M. Wang, Y. Cai, and S. Lu, “Extracting closed object contour in the image: remove, connect and fit”, *Pattern Analysis and Applications*, pp. 1–14, 2018.

[28] B. Ravinder and M. Lakpathi, “Improve the capacitor voltage balancing control for three-level boosting PFC”, *International Journal of Advanced Technology and Innovative Research*, Vol. 7, No. 8, pp. 1361-1368, 2015.

[29] Y. Matsuda, H. Hoashi, and K. Yanai, “Recognition of multiple-food images by detecting candidate regions”, In: *IEEE International Conference on Multimedia and*

Expo, Melbourne, Australia, pp. 25-30. July, 2012.

[30] Y. Kawano and K. Yanai, “Automatic expansion of a food image dataset leveraging existing categories with domain adaptation”, In: *European Conference on Computer Vision*, pp. 3–17, 2014.

[31] A. Borji, “Pros and cons of gan evaluation measures”, *Computer Vision and Image Understanding*, Vol. 179, pp. 41-65, 2019.