

AUTOMATIC COLOR IMAGE SEGMENTATION USING DEFORMABLE NET

Khaled M. Shaaban and Nagwa M. Omar

Electrical Engineering Department, Assiut University, Assiut, Egypt

(Received November 19, 2006 Accepted February 24, 2007)

This paper proposes the use of deformation as a foundation for color image segmentation. Instead of deforming a single contour, typically used with other deformable contour methods, the proposed technique deforms a planner net. This net consists of a group of vertices connected by edges without crossing each other. The connected edges form polygons that represent the segmented regions boundaries. During the deformation process, the algorithm changes the location and the number of vertices as well as the number of polygons to enhance the segmentation fit. The deformation forces are generated based upon the homogeneity of the color distribution within the regions. The algorithm is completely autonomous and does not require any training or pre-knowledge about the image contents. The experimental results demonstrated the capability of the algorithm to segment color images from arbitrary sources within reasonable time. Furthermore, the compact mathematical representation of the resulting boundaries could be of value for further image analysis.

KEYWORDS: *Color Image Segmentation, Entire Image Segmentation, Object Segmentation, Region Based Segmentation, Edge Based Segmentation, Deformable Contour Methods.*

1. INTRODUCTION

In image analysis, the term segmentation is usually used to indicate two separate operations. One operation can be called *entire image segmentation*. It is intended to partition the whole image into regions [1-12]. The other operation which is sometimes called segmentation is *object segmentation*. It aims for obtaining the contour of a specific object or objects in the image [13-29]. There are two concepts usually used to implement both directions of segmentation. The first depends upon the gradient information around the edges of the objects. Therefore, it is frequently called *edge based segmentation* [13-15, 17-29]. Local gradient makes it sensitive to noise, which affects its convergence speed and quality. The other segmentation concept relies upon the homogeneity of one or more of the characteristics (brightness, texture, or color) within the regions in the image. Since these algorithms are affected by all the regions' pixels, they are usually named *region based segmentation* [1-12, 16]. The whole aspect of the region based segmentation makes it less sensitive to noise but it is typically more expensive. Examples of the region based segmentation include: region

growing [11], split and merge [12], histogram thresholding [10], random field [1], and clustering [4-6].

Classical edge based segmentation is realized in two steps. First, all edges in the image are detected using some low-level edge detection method [28, 29]. Second, an edge linking strategy is used to order edges according to continuity and connectivity. These methods have difficulties in handling noisy edge points, overlapping and broken boundaries, as well as complex shapes [17]. As an attempt to solve some of the above problems, energy minimization methods called *Deformable Contour Methods* (DCMs) were proposed [13-27].

DCMs use elastic curves (contours) that deform under the influence of internal and external forces. The internal forces impose the contour smoothness during deformation. The external forces attract the contour to the object boundary. The DCMs algorithms try to minimize the integration of these forces around the contour. Parametric deformable models [13, 23, 25-27], Geometric deformable models [18] and Prototype-based deformable models [14-17, 19-22, 24] are popular examples. Due to the locality of the search, DCMs suffer from a convergence problem. They require more prior knowledge or manually guided initial contours. For example, model-based DCMs [24, 26] were proposed to integrate the prior object shape information in the object extraction process. However, the algorithm could not solve the problem of variation due to object geometric transformations. Other model-based DCMs [22, 24] require complex searching algorithms, or need heavy training [20, 21]. Due to this convergence problem, DCMs are used for object segmentation and had not been tested for entire image segmentation.

Although, color images contain additional information than gray level ones, the segmentation of such images is more complicated. Surveys of color image segmentation could be found in [7, 30]. The added difficulty of segmenting color images is mainly due to the lack of a good definition of color that matches the human perception of this concept. Another reason is the mathematical complexity associated with color distance measurement in the 3D-representation of color. This complexity makes the qualification of the pixel to join a segment, harder to evaluate. For these reasons, segmenting color images is still subject to research [1-12, 16, 20, 21, 30].

This paper proposes a segmentation technique for color images using a Deformable Contour strategy. We call this technique Region-based Deformable Net (RbDN) technique. Instead of the single contour usually used with DCMs, the proposed technique deforms a planner net. This net consists of elastic polygons of an arbitrary number of line segments that cover the entire image. The location and the shape of the polygons change during the deformation to minimize an energy-function. The function minimization moves the polygons' corners until they fit the regions' boundaries. This energy function is the

integration of forces that are generated on the common line between every adjacent polygon pair. The forces are generated based on the color of the pixels surrounding these lines and the average color of the polygons. Because of this energy minimization strategy, the proposed technique belongs to the family of Deformable Contours. RbDN technique is also region based because it segments the whole image into regions depending upon the homogeneity of the color distribution within each region. Consequently, the main contribution of the proposed system is the original use of deformation for entire image segmentation without any prior information, user interference or training.

The rest of this paper is organized as follows: Section 2 provides a description for the proposed net structure. Section 3 describes the net deformation. Section 4 describes the necessary net maintenance operations. Section 5 shows some of the experimental results. Section 6 concludes this work.

2. NET STRUCTURE

In order to fully understand the RbDN technique, a mathematical formalism is needed. The net is simply a plane graph, $Net = (V, E)$, that consists of a group of vertices, V , connected by edges, E . Each vertex, $v_i \in V(Net)$, is represented by a point in the Euclidian plane, $v_i(x, y)$, where x and y are Euclidian distances from a reference at the upper left corner of the Net . Each edge, $e_i \in E(Net)$, is represented by a line segment that connects two vertices, $e(v_i, v_j)$, i.e. $E \subseteq [V]^2$. For the rest of this work the term edge will be used to represent this defined mathematical meaning and will not be used to indicate a point with high value of gradient in the image. Nontrivial network covers a limited area of the Euclidian plane that is referred to as Q .

The plane graph has a unique characteristic: it can be sketched on a piece of paper in such a way that no edges meet in a point other than the common ends (the vertices). The following few restrictions are added to the general definition of the planer graph to form the definition of the Net :

- The Net has vertices at the corners of Q , to identify the Net extent. These vertices are connected with edges to surround Q . These edges form the outer boundary of the Net .
- The set of edges, $E(Net)$, could be partitioned into subsets, such that each subset, p_k , represents a **polygon** within Q . The edges within each polygon are ordered such that the interior of the polygon is always on the right hand side of the edges. Note that, each edge contributes in exactly two polygons

except the edges at the outer boundary of the *Net*. The sequence of edges, $\{e_1, e_2, \dots, e_f \mid e_i \in p_k\}$, could be represented by an ordered set of vertices. Therefore, we can rewrite the polygon as $p_k = \{v_1, v_2, \dots, v_f\}$ which signify that, each pair (v_i, v_{i+1}) is an edge in, p_k . The pair (v_f, v_1) represents the last edge in the polygon, p_k . Each polygon covers an area of Q that we call, $A(p_k) \subseteq Q$. These areas are not mutually exclusive, that as $A(p_i) \cap A(p_j)$ does not necessary represent a zero area. A polygon can contain another polygon within its area.

- Except for very special networks, there is a large number of ways in which a network can be partitioned into polygons. A unique partitioning is to use polygons with the smallest possible area. That is to minimize the overlapping of polygons.

Therefore, the *Net* represents a way to partition the space, Q , into set of polygons, $P(\text{Net})$. In other words the polygons resample the pieces of a puzzle that when fitted together form the full area, Q . At this point we need to refine the notation of the net to be, $\text{Net} = (V, E, P)$.

Given a real life image, I , and a $\text{Net} = (V, E, P)$ with extent, Q , that has the exact same dimension of the image, we can overlay the *Net* over the image. Each Polygon of the *Net*, $p_k \in P(\text{Net})$, or the difference of two or more polygons represents a segment of the image. Therefore, we can consider the *Net* as a formal mathematical notation to represent a segmentation of an image. This mathematical representation is necessary to introduce the concept of deformation to the process of image segmentation. One can easily imagine the process of deformation as the process of adjusting the location of the vertices (the corners of the polygons) to coincide the segments in the image. The mathematical description of the segments as a *Net*, provides the language to describe the different deformation operations like, inserting a new vertex into a polygon or merging two polygons to form a single larger one. Table (1) provides the definitions that describe the *Net* structure as well as the operations that could be applied to it:

The general structure of the proposed net is illustrated through simple example shown in Figure (1). As shown in this figure the image under segmentation has three regions R_1 , R_2 and R_3 . The first region, R_1 , is represented by one polygon, $p_1 = \{v_1, v_5, v_6, v_7\}$, while R_2 is represented by two polygons $p_2 = \{v_2, v_3, v_4, v_7, v_6, v_5\}$ and $p_3 = \{v_8, v_9, \dots, v_{23}\}$, the area of $R_2 = A(p_2 - p_3)$, the third region, R_3 , is represented by p_3 .

Table (1): Summary of Net symbols and definitions

Symbol	Description
$Net = (V, E, P)$	The <i>Net</i> defined by the vertices, edges and polygons.
$V(Net) = \{v_1, v_2 \dots, v_n\}$	The set of all the vertices in the <i>Net</i> . Each vertex, $v_i \in V(Net)$, is represented by a point in the Euclidian plane, $v_i(x, y)$.
$E(Net) = \{e_1, e_2 \dots, e_m\}$	The set of all the edges in the <i>Net</i> . Each edge, $e_i \in E(Net)$, is represented by a line segment that connects two vertices, $e(v_i, v_j)$.
$P(Net) = \{p_1, p_2 \dots, p_o\}$	The set of all polygons in the <i>Net</i> .
$p_k = \{v_1, v_2, \dots, v_f\}$	The polygon k of the <i>Net</i> , i.e., $p_k \in P(Net)$. Note that the index of the vertex in p_k not necessary equals to its index in the <i>Net</i> .
Q	The area in the Euclidian plane covered by the <i>Net</i> .
$A(p_k)$	The subset of Q that is covered by the polygon p_k .
$C(p_k)$	The average color vector of the images pixels that are covered by the polygon, p_k .
$\zeta_k = \{p_1, p_2, \dots, p_r\}$	The set of all polygons in which v_k is a member.
$L(v_k) = (x, y)$	The Euclidian location of the vertex, v_k .
$L(v_k) + (\Delta x, \Delta y) \rightarrow L(v_k)$	The operation of moving the location of the vertex v_k . For example, $L(v_k(x, y)) + (\Delta x, \Delta y) \rightarrow L(v_k(x + \Delta x, y + \Delta y))$
$p_k + i$	The process of inserting a new vertex at location i to a polygon k . This insertion requires the update of the indices of all vertices starting from $i+1$. For example $\{v_1, v_2 \dots, v_f\} + i = \{v_1, v_2 \dots, v_i, \dots, v_{f+1}\}$.
$p_k - i$	The process of deleting the vertex at location i from a polygon k . This deletion requires re-indexing the vertices.
$p_i + p_j \rightarrow p_i$	The operation of merging two polygons forming a single larger one. It requires re-indexing the polygons.
$A(p_i + p_j)$	The subset of Q that is covered by the polygon which results from merging p_i and p_j .
$A(p_i - p_j)$	The subset of Q that is enclosed between p_i and p_j .

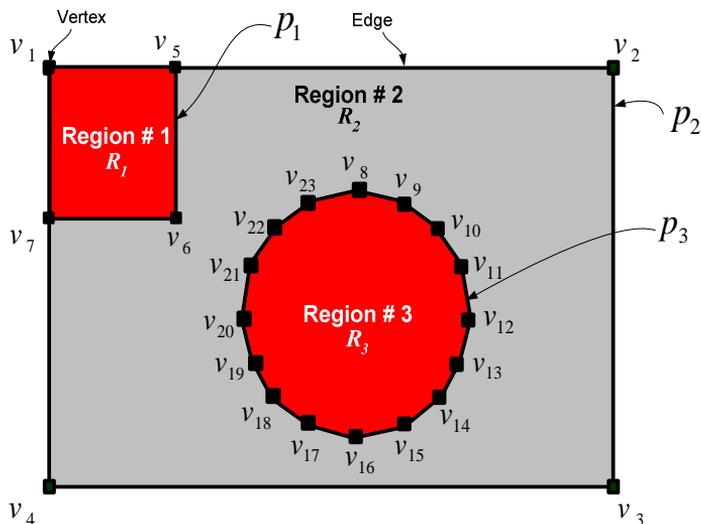


Figure 1: Segmentation example clarifies the net structure

3. NET DEFORMATION

The proposed net is automatically initialized to fully cover the real life image, I . That is the corner vertices that define Q should coincide the image corners. The proposed net can have arbitrary initial structure but we choose the simple one illustrated in Figure (2). As shown in this figure the net extent, Q , is partitioned into equaled size squares.

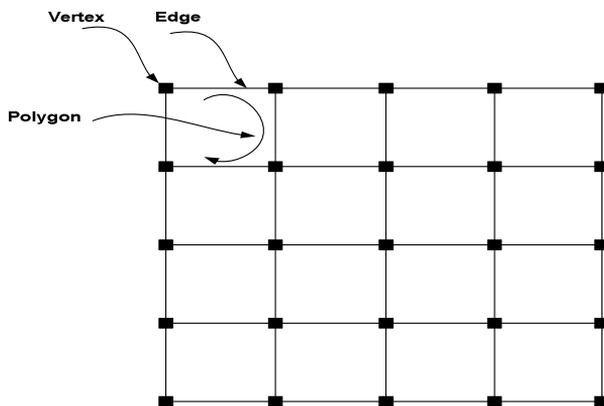


Figure 2: The initial shape of the proposed net, equaled size squares

The net deforms under the effect of forces generated around the common edge between every adjacent polygon pair. The average color of each polygon in the pair and the color of the pixels around the common edge, generate these

deformation forces. Each polygon searches a thin area outside its boundary for pixels with color that are close to its average color. If considerable number of such pixels is found, the polygon attempts to inflate itself to include these pixels. We call these thin areas the sensitivity regions. Naturally the forces of the neighboring polygon oppose this inflation and the system settles at the equilibrium of all these forces.

The left hand side (outside) of every edge in each polygon contains two non overlapped sensitivity regions as shown in Figure (3).

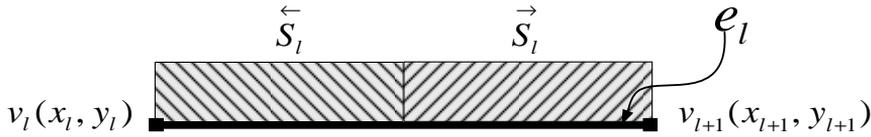


Figure 3: Edge, e_l , surrounded by two sensitivity regions. Left and right sensitivity regions are represented by \overleftarrow{S}_l and \overrightarrow{S}_l respectively

For the edge, e_l , these sensitivity regions are denoted \overleftarrow{S}_l and \overrightarrow{S}_l . Each sensitivity region is a rectangular area having a height of w and width equals to half of edge length. To understand how the forces are generated consider the arrangement shown in Figure (4).

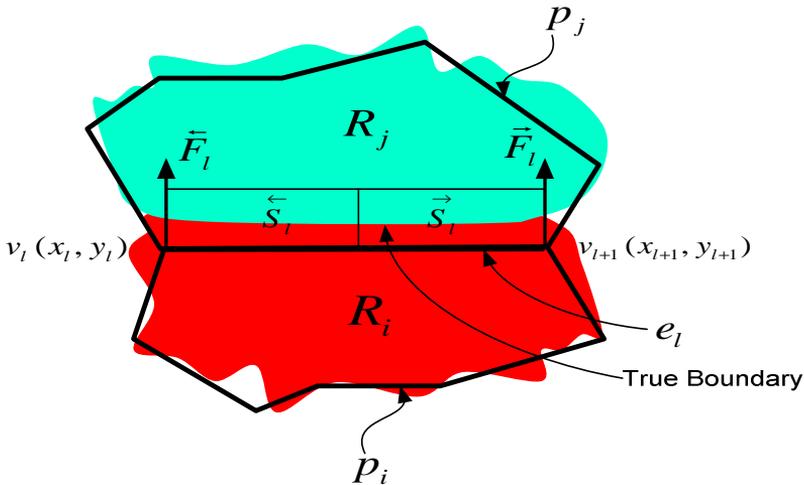


Figure 4: A part of the proposed net shows forces affect edge e_l from the point of view of p_i .

In the figure, there are two adjacent regions having different colors, R_i and R_j , and two polygons, p_i and p_j , that are not aligned over the regions. The two polygons cover image areas, $A(p_i)$ and $A(p_j)$ and their respective colors

averages are represented by $C(p_i)$ and $C(p_j)$. The edge separating the two polygons does not coincide with the true boundary separating the two regions forming alignment disparity. From the point of view of p_i , this disparity is measured by the number of pixels within each of its sensitivity regions \overleftarrow{S}_i and \overrightarrow{S}_i that satisfy the following conditions:

1. The pixel ρ is located within the area of the neighboring polygon, $\rho \in A(p_j)$.
2. The color distance between the pixel color and its current polygon color is large, $ColorDist (C(\rho), C(p_j)) > \eta$. That is, the pixel should not belong to this region based on the color distance.
3. The distance between the pixel color and the neighboring polygon color is small, $ColorDist (C(\rho), C(p_i)) \leq \eta$.

Where,

$C(\rho)$: The color vector of the pixel ρ .

$ColorDist(C_1, C_2)$: A measurement of color dissimilarity between two color vectors, C_1 and C_2 .

η : The color distance threshold.

We denote such alignment disparity measure $H(\overleftarrow{S}_i)$ and $H(\overrightarrow{S}_i)$ respectively.

A small value of $H(\overleftarrow{S}_i)$ and $H(\overrightarrow{S}_i)$ represents a good fit of the edge e_i . The deviation from this state leads to the deformation forces:

$$\vec{F}_i = \frac{H(\overleftarrow{S}_i)}{2\lambda} \quad (1)$$

$$\vec{F}_i = \frac{H(\overrightarrow{S}_i)}{2\lambda} \quad (2)$$

Where, λ : The length of the edge

From the point of view of p_j (not shown in the figure), there is no color mismatch under its sensitivity regions and thus no opposing forces.

In general any vertex v_k is a member in a set of polygons ζ_k . In each polygon, this vertex connects exactly two edges each generates forces that affect its position. Thus, the number of forces that affect the vertex v_k is $\psi_k = 2\zeta_k$, see Figure (5). These forces are arbitrary oriented and are treated as real forces. They are added as vectors to generate the total force, F_k^T , that affecting the vertex v_k ,

$$F_k^T = \sum_{i \in \psi_k} F_i \tag{3}$$

F_k^T could be decomposed into two components one in the x direction that we denote F_k^{Tx} and the other in the y direction that we denote F_k^{Ty} . These components are the best estimation of the position change needed to enhance the fit of the polygon edge over the region boundary, that is:

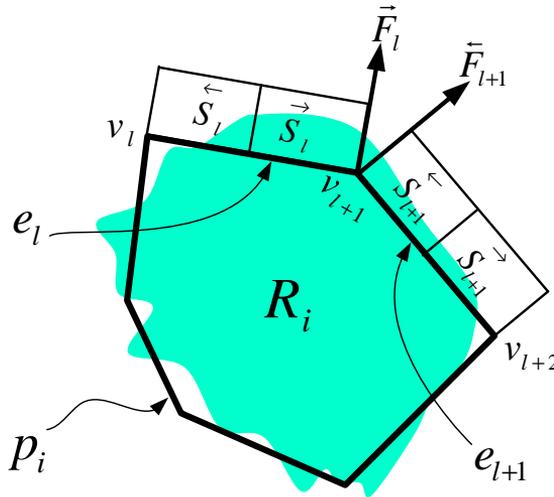


Figure 5: A part of the proposed net shows forces affect vertex v_{l+1} due to its existence in p_i

$$\Delta x_k = F_k^{Tx} : \text{The total deviation of the vertex } v_k \text{ in the } x \text{ direction.} \tag{4}$$

$$\Delta y_k = F_k^{Ty} : \text{The total deviation of the vertex } v_k \text{ in the } y \text{ direction.} \tag{5}$$

Therefore the position update rule could be written as:

$$L(v_k) + (\Delta x_k, \Delta y_k) \rightarrow L(v_k) \tag{6}$$

A complete round of vertices adjustment forms a single deformation cycle. Usually more than one cycle is needed to get good results.

4. NET MAINTENANCE

During the deformation process situation that requires special treatment may arise. The system periodically checks and handles these situations to keep the net simple. The most import situations and the way to handle them are as follows:

Polygon merge: If during the deformation, two neighboring polygons with almost the same average region colors emerge, they should be merged in order to reduce the overall number of the polygons. Assume that these two polygons colors averages are represented by $C(p_i)$ and $C(p_j)$ and if $ColorDist(C(p_i), C(p_j)) \leq \eta$ then p_i and p_j should be merged, $p_i + p_j \rightarrow p_i$.

There is another type of polygon merging that depends on the polygon size. Polygons with very small area (smaller than 200 pixels) are merged to one of its neighbors. The neighbor to be merged with is the one with minimum color distance (to the polygon to be deleted) regardless of the magnitude of that distance.

Vertex deletion: There are three states that require deleting a vertex in order to minimize the overall number of vertices. These states are:

1. Two edges that almost lie on the same line, see Figure (6-a, 6-b). As shown in this figure, v_{q+1} should be deleted, $p_i = p_i - (q + 1)$.
2. Small length edges that have a negligible effect on the net shape see Figure (6-c, 6-d). As shown in this figure, v_{s+2} should be deleted, $p_i = p_i - (s + 2)$.
3. Spike (thorn) edges, the edges which enclose small angel, see Figure (6-e, 6-f). As shown in this figure, v_{d+1} should be deleted, $p_i = p_i - (d + 1)$.

Vertex Insertion: Since there is no prior knowledge about the regions' shapes, the optimum number of vertices for each specific polygon is not known. Therefore, and during the deformation process a polygon with less than adequate number of vertices may arise. The solution for such case is the vertex insertion operation. Figure (7) shows an edge, e , that need vertex insertion to enhance its fit. As shown in the Figure, the two alignment disparity measure of this edge from the point of view of the polygon p_i are $H(\overleftarrow{S}_i)$ and $H(\overrightarrow{S}_i)$ and from point of view of p_j are $H(\overleftarrow{S}_j)$ and $H(\overrightarrow{S}_j)$. In this arrangement the force

due to $H(\overleftarrow{S}_i)$ is balanced with the force due to $H(\overrightarrow{S}_i)$ and the force due to $H(\overrightarrow{S}_i)$ is balanced with the force due to $H(\overleftarrow{S}_i)$. That is, the overall forces affecting e are small but the quality of the fit is not good. This special balance state could be easily detected by observing that the overall small forces are not accompanied with small value of its alignment disparity measures. If any of the measures is above a specific limit, δ , then there is a need for a new vertex. The insertion operation is performed by breaking the edge e into two edges then re-indexing the vertices in the polygon. If the new vertex will be inserted at location k in p_i , then we denote this insertion operation as $p_i = p_i + k$.

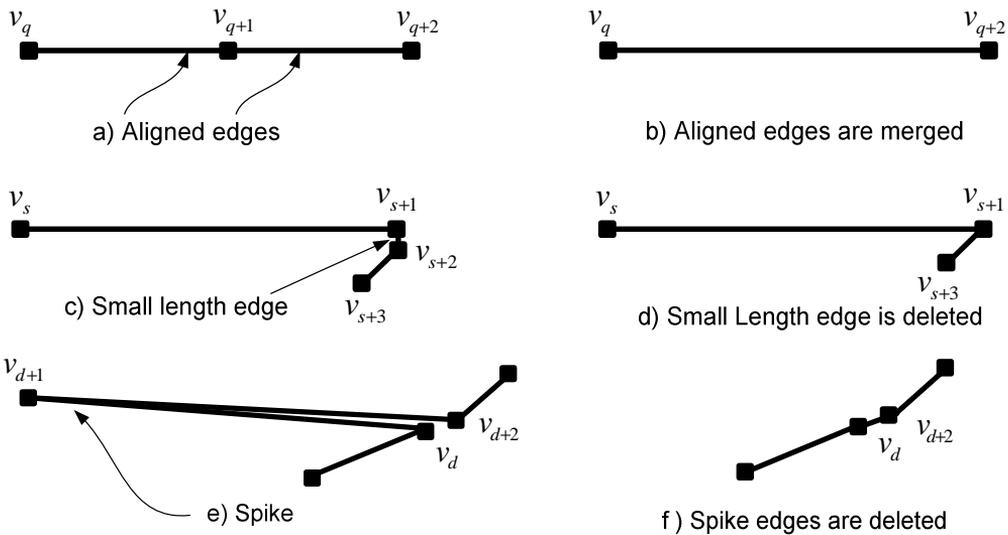


Figure 6: Vertex deletion example, the left column shows unnecessary edges in a certain polygon, p_i , a) Aligned edges, c) Small length edge and e) Spike Edges. The right column shows the results after deleting the unnecessary edges.

The net deformation and the maintenance cycles are repeated periodically until a good fit is reached. Stopping the iteration process depends upon the maximum displacement over of all the vertices in the net. If this displacement is under a specific preset value the algorithm stops.

Finally the proposed algorithm steps can be summarized as follows:

1. Initialize the deformable net (Section 3)
2. Calculate the polygons average colors
3. For $i=0$ to Number of iterations

- Use equation (3) to calculate the total forces that affecting the vertices then use equation(6) to update the vertices positions (Section 3)
 - Calculate the polygons average colors
 - if ((i % Maintenance Period)=0)
 - Merge the neighboring polygons with almost the same average region colors (Section 4)
 - Delete the unnecessary vertices (Section 4)
 - Improve the fitting accuracy by inserting new vertices (if needed), (Section 4)
- End if

End For

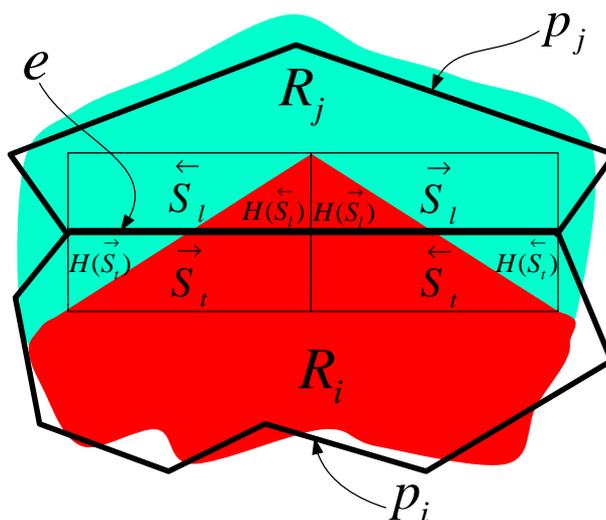


Figure 7: Condition at which a vertex should be inserted

The algorithm complexity depends basically on the first two steps in the loop as well as the number of iterations. The complexity of the first step in the loop is proportional to the number of vertices, n , $O(n)$. The second step complexity is proportional to the number of the image pixels, u , $O(u)$. The maintenance step has no significant effect in the overall complexity. The number of iterations is proportional to the number of vertices, n . Then the overall complexity of the algorithm is $O(n^2, un)$.

5. EXPERIMENTAL RESULTS

Unfortunately, there is no quantitative measure to evaluate the fitting quality [1-30] of segmentation algorithms. Therefore the evaluation is usually based on human judgment. For the evaluation purpose we developed our own set of color

images from different sources, taken under diverse environmental parameters and with different quality. The set consists of more than 200 bitmap images of different size. Samples from this set are shown in the following examples.

All the examples shown in this section share a common set of operation parameters. These parameters are chosen by experimental trial. The used parameters are:

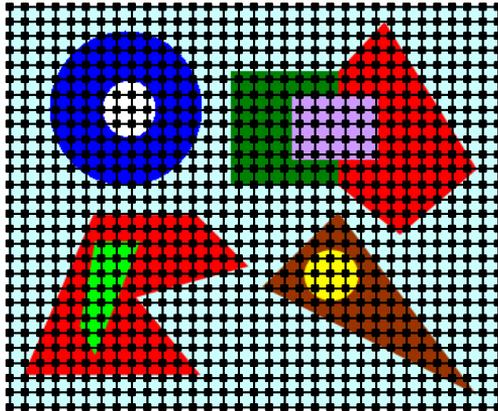
- The initial net has equal squares of 10x10 pixels.
- The regions with size less than 200 pixels are deleted.
- The edges having length smaller than 10 pixels are deleted.
- The edges that enclose an angle less than 10 degrees are deleted.
- The edges having an angle difference smaller than 5 degrees are merged.

The first example uses a simple geometric shapes image shown in Figure (8). The initial net has large number of squares as shown in Figure (8-a). Figure (8-b) shows the result of merging similar color adjacent regions after one step which highly reduces the number of the polygons. Figure (8-b,c,d) shows the improvement in the fitting as the number of iterations increases. The number of the vertices is vastly reduced due to the vertex deletion process; see (8-b, c, d). The final net as shown in Figure (8-d) fits the objects boundaries with good quality and reasonable number of vertices.

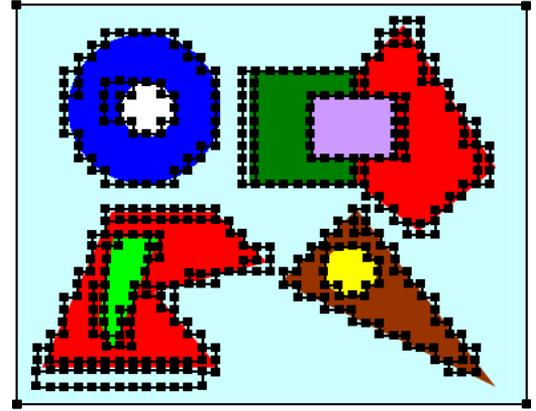
Figure (9) shows the results of using the proposed technique to segment a selected group of images. The resulted nets highly fit the objects boundaries. As shown in the last image (the kid), the image objects can be highly recognized through the resulted net. A summary of the parameters and the numerical results of this experiment are illustrated in Table (2). As shown in this Table, the number of the resulted regions and the resulted vertices are suitable for further analysis. The execution time is affected by the image size, the number of the polygons, the number of the vertices and the number of iterations.

Table (2): Summary of the experimental results for the images in Fig. 9

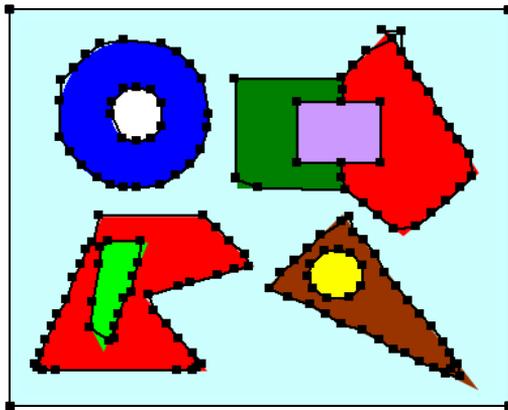
Image	Size	Number of resulted regions	Number of resulted vertices	Number of iterations	Execution time in sec.
Rose	310x210	3	71	5	0.15
Horse	320x240	8	75	10	0.5
Peppers	420x310	15	145	10	0.8
Kid	260x330	6	60	10	0.49



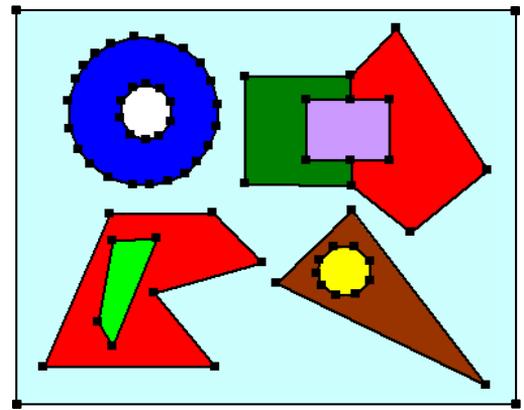
a) The initial net



b) The segmentation result after one iteration



c) The segmentation result after three iterations



d) The final segmentation result after 10 iterations

Figure 8: Simple Image (Basic shapes). a) The initial net, b) The segmentation result after 1 iteration, c) The segmentation result after 3 iterations, d) The final segmentation result

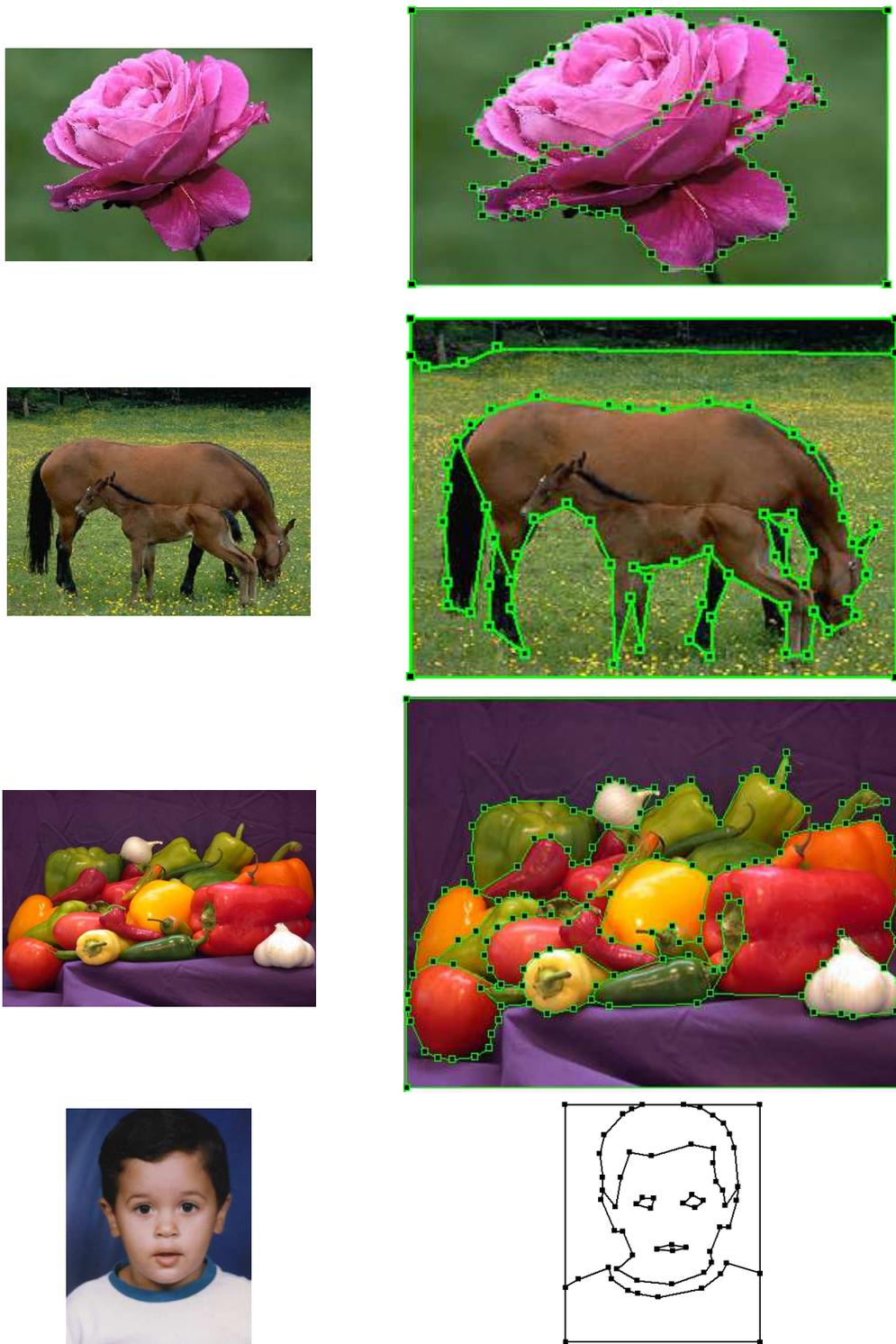


Figure 9: Example illustrates the technique quality. Left column represents the test image set (Rose, Horse, Peppers and Kid). Right column represents the final results

Table (3): Comparing RbDN with Makrogiannis and others [4, 5]

Image	Size	Number of resulted regions		Execution time in sec.	
		RbDN Technique	Makrogiannis and others	RbDN Technique	Makrogiannis and others
House	256x256	15	140	0.275	1.4365
Parrots	256x256	20	100	0.332	0.8125
Peppers	256x256	16	60	0.3797	0.884
Tree	256x256	27	80	0.3758	0.988

To evaluate the overall performance and the convergence speed of the proposed algorithm we compared our work to the original work of Makrogiannis and others [4, 5]. Makrogiannis tried to incorporate the principles of region-based segmentation and cluster-analysis approaches to achieve efficient segmentation results. The results of this comparison are illustrated in Table (3). As shown from the table, RbDN technique gives good results in shorter time also it resulted in smaller number of regions. Segmenting the images into smaller number of regions is of significant value for further image analysis. Furthermore, the compact mathematical representation of the objects boundaries in the form of a net is superior to representing the resulting image in the form of labeled image. The compact representation as net directly gives topological and other information about the regions and the shape of the boundary between them. Such information is harder to evaluate directly through other segmentation techniques (up to our knowledge).

As shown from these experiments, the results provided good segmentation quality in a reasonable time. Also the compact mathematical representation of the regions' boundaries could be useful for processing towards scene understanding.

6. CONCLUSIONS

This work proposes using Deformable Contours for color image segmentation. It aims to segment the entire image into regions not to detect a certain object or group of objects in the image. We call this technique Region-based Deformable Net (RbDN) technique. RbDN deforms a planner net that has an arbitrary initial structure. The net consists of a group of vertices connected by edges. The edges are connected without crossing each others forming polygons (closed contours). The polygons represent the segmented regions boundaries. The deformation process is based on the homogeneity of the color distribution within each

region. The proposed technique does not require prior knowledge or user interference or training.

The segmentation process starts by positioning the initial net over the image such that the four corner vertices of the net coincidence with the image corners. The deformation process iteratively changes the number and the location of both the vertices and the polygons. These changes are due to forces generated on the common edges between polygons. These forces are generated based upon the color of the pixels within a thin layer surrounding these edges and the average color of the polygons.

The algorithm performs periodical maintenance-cycle to enhance the general structure of the net. The operations performed during this maintenance cycle include:

- Searching for polygon with less than adequate fit quality and insert more vertices to provide the flexibility to enhance the fit.
- Also the algorithm deletes the unnecessary edges to maintain a reasonable size of the net.
- It merges similar color adjacent regions to make the number of regions smaller.

The experimental results showed the quality of proposed technique. The segmentation time of the process is smaller compared with the time reported with other color image segmentation techniques. The proposed technique can be used effectively in real time applications such as robotics applications. RbDN technique can be used effectively to segment video sequences. The final contour which results from the deformation process of a frame can be used as a good estimation of the initial contour for the next frame. The sequential segmentation should considerably shorten the segmentation time for subsequent frames.

REFERENCES

1. Y. Xia, D. Feng, and R. Zhao, "Adaptive Segmentation of Textured Images by Using the Coupled Markov Random Field Model", *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3559- 3566, 2006.
2. Q. Luo, and T. M. Khoshgoftaar, "Unsupervised Multiscale Color Image Segmentation Based on MDL Principle", *IEEE Transactions on Image Processing*, vol. 15, no. 9, pp. 2755-2761, 2006.
3. J. S. Cardoso, and L. Corte-Real, "Toward a Generic Evaluation of Image Segmentation", *IEEE Transactions on Image Processing*, vol. 14, no. 11, pp. 1773-1782, 2005.

4. S. Makrogiannis, G. Economou, and S. Fotopoulos, "A Region Dissimilarity Relation That Combines Feature-Space and Spatial Information for Color Image Segmentation", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 35, no. 1, pp. 44-53, 2005.
5. S. Makrogiannis, G. Economou, S. Fotopoulos, and N. G. Bourbakis, "Segmentation of Color Images Using Multiscale Clustering and Graph Theoretic Region Synthesis", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 35, no. 2, pp. 224-238, 2005 .
6. D. Guo, and X. Ming, "Color Clustering and Learning for Image Segmentation Based on Neural Networks", *IEEE Transactions on Neural Networks*, vol. 16, no. 4, pp. 925-936, 2005.
7. J. Verges, "Color Constancy and Image Segmentation Techniques for Applications to Mobile Robotics", PhD, University of Catalonia, Spain, 2005.
8. L. Salgado, N. Garcia, J. M. Menendez, and E. Rendon, "Efficient Image Segmentation for Region-Based Motion Estimation and Compensation", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 7, pp. 1029-1039, 2000.
9. J. Bruce, T. Balch, and M. Veloso, "Fast and Inexpensive Color Image Segmentation for Interactive Robots", *Proc. of IEEE Inter. Conf. on Intelligent Robots and Systems*, vol. 3, pp. 2061- 2066, 2000.
10. M. Celenk, and M. Uijt de Haag, "Optimal Thresholding for Color Images", *Proc. of the SPIE -The Int'l Soc. for Optical Eng., Nonlinear Image Processing IX*, San Cose, CA, vol. 3304, pp. 250-259, 1998.
11. A. Tremeau, and N. Borel, "A Region Growing and Merging Algorithm to Color Segmentation", *Pattern Recognition*, vol. 30, no. 7, pp. 1191-1203, 1997.
12. K. Saarinen, "Color Image Segmentation by a Watershed Algorithm and Region Adjacency Graph Processing", *Proc. of IEEE Inter. Conf. on Image Processing (ICIP'94)*, Austin, TX, vol. 3, pp. 1021-1025, 1994.
13. M. Sakalli, K.-M. Lam, and Y. Hong, "A Faster Converging Snake Algorithm to Locate Object Boundaries", *IEEE Transactions on Image Processing*, vol. 15, no. 5, pp. 1182-1191, 2006.
14. Ch. Chang, "Deformable Shape Finding with Models Based on Kernel Methods", *IEEE Transactions on Image Processing*, vol. 15, no. 9, pp. 2743-2754, 2006.
15. G. Foresti, and F. Pellegrino, "Automatic Visual Recognition of Deformable Objects for Grasping and Manipulation", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 34, no. 3, pp. 325-333, 2004.
16. K. Shaaban, "Model Deformation Using Hit or Miss Operation", *Journal of Engineering Science, Assiut University, Egypt*, vol. 32, no. 1, pp. 471-484, 2004.
17. L. He, "A Comparison of Deformable Contour Methods and Model Based

- Approach Using Skeleton for Shape Recovery from Images", PhD, Elec. Dep., Computer Eng. and Computer Science, College of Engineering, University of Cincinnati, 2003.
18. X. Han, C. Xu, and J. L. Prince, "A Topology Preserving Level Set Method for Geometric Deformable Models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 755-768, 2003.
 19. B. Ginneken, A. F. Frangi, J. J. Staal, B. M. Romeny, and M. A. Viergever, "Active Shape Model Segmentation with Optimal Features", *IEEE Transactions on Medical Imaging*, vol. 21, no. 8, pp. 924-933, 2002.
 20. S. Sclaroff and L. Liu, "Deformable Shape Detection and Description via Model-Based Region Grouping", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 5, pp. 475-489, 2001.
 21. L. Liu, and S. Sclaroff, "Region Segmentation via Deformable Model-Guided Split and Merge", Technical Report, Computer Science Department Boston University, 2001.
 22. Y. Tang, L. He, X. Wang, and W. Wee, "A Model Based Contour Searching Method", *Proc. of IEEE International Symposium on Bio-Informatics & Biomedical Engineering*, pp. 347-354, 2000.
 23. C. Xu, and J. L. Prince, "Snakes, Shapes, and Gradient Vector Flow", *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359-369, 1998.
 24. A. K. Jain, Y. Zhong, and S. Lakshmanan, "Object Matching Using Deformable Templates", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 3, pp. 267-278, 1996.
 25. S. Lobregt, and M. A. Viergever, "A Discrete Dynamic Contour Model", *IEEE Transactions on Medical Imaging*, vol. 14, no.1, pp. 12-24, 1995.
 26. L. H. Staib and J. S. Duncan, "Boundary Finding with Parametrically Deformable Models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1061-1075, 1992.
 27. M. Kass, A. Witkin and D. Terzopoulos, "Snake: Active Contour Models", *Int. J. Computer Vision*, vol. 1, no. 4, pp. 321-331, 1987.
 28. J. F. Canny, "A Computational Approach to Edge-Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, 1986.
 29. D. Marr and E. Hildreth, "Theory of Edge Detection", *Proc. Royal Society, London*, vol. B207, pp. 187-217, 1980.
 30. L. Lucchese and S. K. Mitra, "Color Image Segmentation: A State of The Art Survey", *Proc. of the Indian National Science Academy (INSA-A)*, vol. 67, no. 2, pp. 207-221, 2001.

شبكة مُتَشَكِّلة لتجزئة الصور الملونة آلياً

يقدم هذا البحث تقنية جديدة لتجزئة الصور الملونة إلى مناطقها بطريقة آلية مبنية على نظرية التشكل للمنحنيات والنظم المصممة بهذه التقنية لا تحتاج إلى تدريب أو معرفة مسبقة بمكونات الصورة. و تعتمد هذه التقنية على شبكة متشكلة للحصول على الحدود لكل مناطق الصورة ، ويعتبر استخدام التشكل لتجزئة الكاملة للصورة هو الاختلاف الجوهرى بين التقنية المقترحة وطرق التشكل الحالية ، حيث أن الطرق الحالية يتم فيها عملية التشكل لمنحنى واحد للحصول على الحدود الخارجية لكائن محدد وليس لاستنتاج الحدود لكل أجزاء الصورة ، وغالبا ما تحتاج طرق التشكل الحالية لتدخل من المستخدم لوضع تصور مبدئي لموقع أو شكل الكائن بالصورة أو لتدريب النظام للتعرف على عدد محدود من الكائنات المتوقعة مما يتطلب معرفة مسبقة بمكونات الصورة والذي قد لا يكون متوفراً في كثير من الأحيان.

والشبكة المستخدمة مكونة من مجموعة من رؤوس المضلعات موصلة معا بخطوط غير متقاطعة ومتقابلة فقط عند هذه الرؤوس و تشغل الشبكة مساحة من المستوى الإقليدى محدودة بخطوط خارجية موصلة للرؤوس الممتلئة لأركان هذه المساحة، وكل مضلع من هذه الشبكة يمثّل رياضياً بمجموعة من الرؤوس المرتبة بحيث تكون المضلعات دائماً على يمين الخطوط الموصلة لهذه الرؤوس، وتشغل هذه المضلعات مساحات متباينة من المستوى الإقليدى ، واتحاد هذه المضلعات يكوّن المساحة الكلية للشبكة وهي مساوية لمساحة الصورة المراد تجزئتها وعند بسط الشبكة على الصورة تمثّل كل منطقة من مناطقها بمضلع واحد أو بالفرق بين عدد من المضلعات، ويعتبر التمثيل الرياضي المحكم للحدود لكل أجزاء الصورة كشبكة متشكلة له قيمة فعالة في التطبيقات القائمة على تحليل الصور.

يقوم الخوارزم المقترح بتحسين انطباق المضلعات على الحدود الحقيقية لأجزاء الصورة عن طريق تغيير أعداد المضلعات وأعداد ومواقع رؤوسها ويتم تغيير مواقع هذه الرؤوس خلال عملية التشكل عن طريق قوى يتم توليدها حول الخطوط المشتركة بين المضلعات بناء على تجانس توزيع اللون في مناطق الصورة بحيث يبحث كل مضلع في شريحة رفيعة حول خطوطه على وحدات تكوين الصورة (البكسل) الأقرب له في اللون من قربها للون المضلعات المجاورة له، وعند تواجد عدد كافي من هذه الوحدات يتمدد المضلع ليحتويها، كما يعمل الخوارزم أيضا على تحسين شكل الشبكة دورياً كالتالي ، أولاً : إضافة رؤوس إلى المضلعات التي لا تتخذ الشكل الأمثل لجزء الصورة الممتلئة له وذلك لتحسين انطباقها عليه ، ثانياً: التخلص من رؤوس المضلعات الزائدة للحصول على الشكل الأبسط للشبكة ، ثالثاً: تقليل عدد المضلعات بدمج ذات الألوان شديدة التقارب معا.

وقد أثبتت التجارب مدى كفاءة وسرعة الخوارزم المقترح في تجزئة صور ملونة متنوعة المصدر .