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...
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
integrated 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, \( \{e_1, e_2, \cdots, e_f \mid e_i \in p_k \} \), could be represented by an ordered set of vertices. Therefore, we can rewrite the polygon as \( p_k = \{v_1, v_2, \cdots, v_f \} \) 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, \( P(Net) \). 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, \( Net = (V, E, P) \).

Given a real life image, \( I \), and a \( Net = (V, E, P) \) 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, \( p_k \in P(Net) \), 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 \( p_3 = \{v_8, v_9, \ldots, v_{23}\} \), 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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Net = (V, E, P)$</td>
<td>The $Net$ defined by the vertices, edges and polygons.</td>
</tr>
<tr>
<td>$V(Net) = {v_1, v_2, \ldots, v_n}$</td>
<td>The set of all the vertices in the $Net$. Each vertex, $v_i \in V(Net)$, is represented by a point in the Euclidean plane, $v_i(x, y)$.</td>
</tr>
<tr>
<td>$E(Net) = {e_1, e_2, \ldots, e_m}$</td>
<td>The set of all the edges in the $Net$. Each edge, $e_i \in E(Net)$, is represented by a line segment that connects two vertices, $e(v_i, v_j)$.</td>
</tr>
<tr>
<td>$P(Net) = {p_1, p_2, \ldots, p_o}$</td>
<td>The set of all polygons in the $Net$.</td>
</tr>
<tr>
<td>$p_k = {v_1, v_2, \ldots, v_f}$</td>
<td>The polygon $k$ of the $Net$, 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 $Net$.</td>
</tr>
<tr>
<td>$Q$</td>
<td>The area in the Euclidean plane covered by the $Net$.</td>
</tr>
<tr>
<td>$A(p_k)$</td>
<td>The subset of $Q$ that is covered by the polygon $p_k$.</td>
</tr>
<tr>
<td>$C(p_k)$</td>
<td>The average color vector of the images pixels that are covered by the polygon, $p_k$.</td>
</tr>
<tr>
<td>$\mathcal{F}_k = {p_1, p_2, \ldots, p_r}$</td>
<td>The set of all polygons in which $v_k$ is a member.</td>
</tr>
<tr>
<td>$L(v_k) = (x, y)$</td>
<td>The Euclidean location of the vertex, $v_k$.</td>
</tr>
<tr>
<td>$L(v_k) + (\Delta x, \Delta y) \rightarrow L(v_k)$</td>
<td>The operation of moving the location of the vertex $v_k$. For example, $L(v_k(x, y)) + (\Delta x, \Delta y) \rightarrow L(v_k(x + \Delta x, y + \Delta y))$.</td>
</tr>
<tr>
<td>$p_k + i$</td>
<td>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, ${v_1, v_2, \ldots, v_f} + i = {v_1, v_2, \ldots, v_i, \ldots, v_{f+1}}$.</td>
</tr>
<tr>
<td>$p_k - i$</td>
<td>The process of deleting the vertex at location $i$ from a polygon $k$. This deletion requires re-indexing the vertices.</td>
</tr>
<tr>
<td>$p_i + p_j \rightarrow p_i$</td>
<td>The operation of merging two polygons forming a single larger one. It requires re-indexing the polygons.</td>
</tr>
<tr>
<td>$A(p_i + p_j)$</td>
<td>The subset of $Q$ that is covered by the polygon which results from merging $p_i$ and $p_j$.</td>
</tr>
<tr>
<td>$A(p_i - p_j)$</td>
<td>The subset of $Q$ that is enclosed between $p_i$ and $p_j$.</td>
</tr>
</tbody>
</table>
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_i, surrounded by two sensitivity regions. Left and right sensitivity regions are represented by \( \vec{S}_i \) and \( \vec{S}_i \) respectively.](image)

For the edge, \( e_i \), these sensitivity regions are denoted \( \vec{S}_i \) and \( \vec{S}_i \). 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 \( e_i \) from the point of view of \( p_i \).](image)

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 $\vec{S}_i$ and $\vec{S}_i$ that satisfy the following conditions:

1. The pixel $\rho$ 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, $\text{ColorDist} (C(\rho), C(p_j)) > \eta$. 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, $\text{ColorDist} (C(\rho), C(p_j)) \leq \eta$.

Where,

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

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

$\eta$: The color distance threshold.

We denote such alignment disparity measure $H(\vec{S}_i)$ and $H(\vec{S}_i)$ respectively. A small value of $H(\vec{S}_i)$ and $H(\vec{S}_i)$ represents a good fit of the edge $e_i$. The deviation from this state leads to the deformation forces:

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

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

Where, $\lambda$: 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 $\zeta_k$. 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 $v_k$,

$$F_k^T = \sum_{i \in \psi_k} F_i$$  \hspace{1cm} (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:

$$\Delta x_k = F_k^{Tx} \hspace{0.5cm} \text{: The total deviation of the vertex } v_k \text{ in the } x \text{ direction.} \hspace{1cm} (4)$$

$$\Delta y_k = F_k^{Ty} \hspace{0.5cm} \text{: The total deviation of the vertex } v_k \text{ in the } y \text{ direction.} \hspace{1cm} (5)$$

Therefore the position update rule could be written as:

$$L(v_k) + (\Delta x_k, \Delta y_k) \rightarrow L(v_k)$$  \hspace{1cm} (6)

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

Figure 5: A part of the proposed net shows forces affect vertex $v_{l+1}$ due to its existence in $p_i$.
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 \( \text{ColorDist}(C(p_i), C(p_j)) \leq \eta \) 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 \( H(S_i) \) and \( H(S_j) \) and from point of view of \( p_j \) are \( H(S_i) \) and \( H(S_j) \). In this arrangement the force
due to $H(S_i)$ is balanced with the force due to $H(S_i)$ and the force due to $H(S_i)$ is balanced with the force due to $H(S_i)$. 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, $\delta$, 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 $p_i = p_i + k$.

$$q_{v_1} + q_{v_2} + q_{v_3} + q_{v_4} + q_{v_5} + q_{v_6} + q_{v_7} + q_{v_8} + q_{v_9} + q_{v_{10}} + q_{v_{11}} + q_{v_{12}} + q_{v_{13}} + q_{v_{14}} + q_{v_{15}}$$

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 angel less than 10 degrees are deleted.
- The edges having an angel 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): Summery of the experimental results for the images in Fig. 9**

<table>
<thead>
<tr>
<th>Image</th>
<th>Size</th>
<th>Number of resulted regions</th>
<th>Number of resulted vertices</th>
<th>Number of iterations</th>
<th>Execution time in sec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rose</td>
<td>310x210</td>
<td>3</td>
<td>71</td>
<td>5</td>
<td>0.15</td>
</tr>
<tr>
<td>Horse</td>
<td>320x240</td>
<td>8</td>
<td>75</td>
<td>10</td>
<td>0.5</td>
</tr>
<tr>
<td>Peppers</td>
<td>420x310</td>
<td>15</td>
<td>145</td>
<td>10</td>
<td>0.8</td>
</tr>
<tr>
<td>Kid</td>
<td>260x330</td>
<td>6</td>
<td>60</td>
<td>10</td>
<td>0.49</td>
</tr>
</tbody>
</table>
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]

<table>
<thead>
<tr>
<th>Image</th>
<th>Size</th>
<th>Number of resulted regions</th>
<th>Execution time in sec.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RbDN Technique</td>
<td>Makrogiannis and others</td>
<td>RbDN Technique</td>
</tr>
<tr>
<td>House</td>
<td>256x256</td>
<td>15</td>
<td>140</td>
</tr>
<tr>
<td>Parrots</td>
<td>256x256</td>
<td>20</td>
<td>100</td>
</tr>
<tr>
<td>Peppers</td>
<td>256x256</td>
<td>16</td>
<td>60</td>
</tr>
<tr>
<td>Tree</td>
<td>256x256</td>
<td>27</td>
<td>80</td>
</tr>
</tbody>
</table>

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 coincide 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


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
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
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-
23. C. Xu, and J. L. Prince, "Snakes, Shapes, and Gradient Vector Flow", IEEE
Deformable Templates", IEEE Transactions on Pattern Analysis and
25. S. Lobregt, and M. A. Viergever, "A Discrete Dynamic Contour Model",
Deformable Models", IEEE Transactions on Pattern Analysis and Machine
27. M. Kass, A. Witkin and D. Terzopoulos, "Snake: Active Contour Models",
Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 6,
شبكة متشكلة لتجزئة الصور الملونة آلياً

يقدم هذا البحث تقنية جديدة لتجزئة الصور الملونة إلى مناطقها بطريقة آلية مبنية على نظرية الشكل للمنحنى والنموذج المصمم بهذه التقنية لا تحتاج إلى تدريب أو معرفة مسبقة بمكونات الصورة. وتعتمد هذه التقنية على شبكة متشكلة للحصول على الحدود لكل مناطق الصورة، ويعتبر استخدام الشكل للتجزئة الكاملة للصورة هو الاختلاف الجوهري بين التقنية المقدّرة وطريقة التشحكم الحالية، حيث أن الطرق الحالية يتم فيها عملية التشكل لمنحنى واحد للحصول على الحدود الخارجية لكاتب محدد وليس استنتاج الحدود لكل أجزاء الصورة، وغالباً ما تحتاج طرق التشحكم الحالية لتدخل المستخدم لوضع تصور مبديء لموقع أو شكل الكائن بالصورة أو لتدريب النظام للتعريف على عدد محدود من الكائنات المتوقعة بما يتطلب معرفة مسبقة بمكونات الصورة والذي قد لا يكون متوفراً في كثير من الأحيان.

والشبكة المستخدمة مكونة من مجموعة من رؤوس المضلعات موضعًا معا بخطوط غير متقطعة ومتقابلة فقط عند هذه الرؤوس وتشغل الشبكة مساحة من المستوى الإقليدي محدودة بخطوط خارجية موصلة للرؤوس المماثلة لأركان هذه المساحة، وكل مضلع من هذه الشبكة يمثل ضياً رياضياً بمجموعة من الرؤوس المرتبة بحيث تكون المضلاع دائما على بين الخطوط الموصولة لهذه الرؤوس، وتشغل كل مجموعة من المضلاعات مساحة متباينة من المستوى الإقليدي، واتحاد هذه المضلاعات يكون المساحة الكلية للشبكة وهي مساحة لمساحة الصورة المراد تجزئتها وعدن بسط الشبكة على الصورة تمثل كل منطقة من مناطقها بمضلع واحد أو بالفرق بين عدد من المضلاعات، ويعتبر التمثيل الرياضي المحكم للحدود لكل أجزاء الصورة كشبكة متشكلة له قيمة فعالة في التطبيقات القائمة على تحليل الصور.

يقوم الخوارزم المفترض بتحسين انتطاق المضلاعات العدد الحقيقية لأجزاء الصورة عبر طريق تغيير أعداد المضلاعات وأعداد ومواقع رؤوسها ويتغير موقع هذه الرؤوس خلال عملية التشكل عن طريق قوى يتم توليدها حول الخطوط المشتركة بين المضلاعات بناء على تجاسي توزيع اللون في مناطق الصورة بحيث يبحث كل مضلع في رقبة رفيعة حول خطوطه على وحدات تكوين الصورة (البكسل). الأقرب له في اللون من قربها لللون المنطلقات المجردة له، وعند توأج عدد كافي من هذه الوحدات يتمدد المضلع ليحيوبيها، كما يعمل الخوارزم أيضاً على تحسين شكل الشبكة دورياً كالتالي، أولًا: إضافة رؤوس إلى المضلاعات التي لا تتخذ الشكل الأمثل لجزء الصورة المماثلة له وذلك لتحسين انطباقها عليه، ثانياً: التخلص من رؤوس المضلاعات الزائدة للحصول على الشكل الأبتض للشبكة، ثالثاً: تقليم عدد المضلاعات بدمج ذات الألوان شديدة التقارب معاً.

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