

Multiple Objects Segmentation with Fuzzy Rule-Base Trained Topology Adaptive Active Membrane

Sitansu Kumar Das^{*}
ECS Unit, Indian Statistical
Institute
Kolkata, India
sitansukumardas@gmail.com

Sanjoy Kumar Saha
CSE Department, Jadavpur
University
Kolkata, India
sks_ju@yahoo.co.in

Dipti Prasad Mukherjee
ECS Unit, Indian Statistical
Institute
Kolkata, India
dipti@isical.ac.in

ABSTRACT

Segmentation of multiple objects in a scene from the single initialization of an active membrane is always advantageous compared to separate initialization of active contour for segmenting each of the multiple objects. This proposal, however, is not robust in segmenting poorly-contrasted touching objects especially when pixel groups belonging to a single object can have spectral signatures similar to the background pixels. In this paper we have used fuzzy rule based learning scheme to record the spectral signature of the objects and background and spatial information of the topology of an active membrane segmenting the objects. The learning scheme helps in splitting the active membrane for segmenting multiple objects and integrates the topology adaptive property of the active membrane with its architecture and evolution. The evolution of this membrane is tested in a challenging application domain of estimation of sizes of oil sand.

Keywords: Active membrane, topology adaptive parametric model, fuzzy-rule base.

1. INTRODUCTION

The topology adaptive active membrane [3] segments images of multiple objects from the single initialization of the membrane placed on top of the image intensity surface. This is superior to topology adaptive active contour [4] because segmentation using active membrane is independent of initialization. Also, since the active membrane evolution is effected in both image plane and also along image intensity plane, it can segment images of touching objects which is not possible through [4] or through straightforward implementation of level sets [1] [2].

However, the active membrane evolution fails to segment images of multiple objects when objects and background have similar spectral characteristics due to uneven surface of the objects or non-uniform lighting condition. Therefore, a

^{*}Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICVGIP '10, December 12-15, 2010, Chennai, India

Copyright 2010 ACM 978-1-4503-0060-5/10/12 ...\$10.00.

need exists to learn or correlate the spectral characteristics of objects and background in a scene with their spatial positions or class label. The objective of this paper is to design a classifier framework to label object and background class in a scene and integrate this classifier output with the active membrane evolution.

The active membrane is modeled using finite elements and contains a set of vertices and links (joining two consecutive vertices). Once the membrane is placed on top of an image intensity surface, the links of the membrane can lie completely within a cluster representing object(s) or within background. In addition the links may lie at the boundary between the object and background clusters. The PDE governing the active membrane evolution should iteratively tear the links lying at the boundary of object and background clusters and should gradually delete the links contained in the background pixel clusters. Finally, after convergence, the remaining links should represent the object(s). The output of the classifier should help in identifying the position of the link with respect to object, background or their boundary pixel clusters.

The classifier is designed in the form of a set of fuzzy rules [5] which get its initial training from a known set of images and their ground truths. The fuzzy rules are derived in the form of a set of features of a link of the active membrane and the corresponding spatial locations of the link, either within the objects or background or at their borders in the training image. Each of the class labels representing objects, background or their boundary, is represented using a Gaussian function. The advantage of the proposed approach is that the parameters of these three Gaussian functions are updated simultaneously using PDE considering membership of each of the link to all the classes rather than taking a hard decision of accepting a particular link to a specific class. This also conforms to the basic premise that the spectral characteristics of these three classes overlap to a great extent, segmentation of which is otherwise almost impossible.

In the next section, we present the basic design of the active membrane, followed by the basic architecture of fuzzy rule based training approach in Section 3. The result of the proposed approach including comparison to competing methods are presented in Section 4 followed by the conclusions.

2. DESIGN OF THE ACTIVE MEMBRANE

An image $I(x, y) : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is considered as a 3D surface where an image pixel value is the height of the surface

along z -axis at an image location (x, y) . The tool for segmentation is a flexible active membrane. This membrane is guided by the internal energy that controls the stiffness of the membrane and an external energy that guides the membrane towards objects seen in the image. The membrane can be defined as a geometric mapping from material (parametric) co-ordinate domain to 3D Euclidean space \mathbb{R}^3 at a particular time instance $t \in [0, \infty)$: $M : (r, s, t) \rightarrow V(r, s, t) = (X(r, s, t), Y(r, s, t))$. The bivariate material co-ordinates are $(r, s) \in [0, 1]$ and $V(r, s, t)$ represents membrane at time t .

Evolution and tearing of membrane need discrete representation of membrane. We assume that there exists an imaginary grid over the membrane. The grid tessellates the membrane into a mesh of small rectangular elements. The grid points over the membrane are represented in a matrix referred as grid matrix. If the membrane is tessellated into a mesh of $m \times n$ rectangular elements, then there exists $(m + 1)$ rows and $(n + 1)$ columns in the grid matrix. The discrete vertices of the elements are represented by $V = (V_{1,1}, V_{2,1}, \dots, V_{i,j}, \dots, V_{m+1,n+1})$ where $V_{i,j} = (X_{i,j}, Y_{i,j})$ denotes the co-ordinate of the (i, j) th vertex. The connection between two neighbouring vertices is defined as link. We first place the membrane covering the entire image. Then we design a mechanism to delete the part of the membrane that belongs to background region of the image and tears those links at the junction of two objects or object and boundary. As noted earlier this deleting and tearing is effected using a classifier which is described next.

2.1 Design and training of classifier

Given a set of training images and their corresponding ground truths including the active membrane grid matrix defined on the ground truths, we label the active membrane links as 0, 1 and 2 corresponding to locations of these links within object, background and their boundaries respectively. Given a link, we specify two image intensity based features across horizontal and vertical directions starting from the middle of the link. Let us describe these features using an example.

Fig. 1(a) is a 6-element, 12-vertices active membrane with 8 horizontal and 9 vertical links. Consider the link AB of Fig. 1(a) and consider the horizontal and vertical directions HH' and VV' starting middle point D of AB . A set of equi-spaced points are sampled along HH' and VV' for the entire span of the active membrane. These sampled points are shown using crossed square boxes in Fig. 1(a). For a set of points S_0, S_1, S_2 etc. as shown in Fig. 1(a), we check for the intensity distribution patterns. Average of pixel intensities between S_0 and S_1 is assigned to S_1 , S_1 and S_2 is assigned to S_2 and so on. In case of monotonically increasing $S_0 < S_1 < S_2 \dots$, the difference between initial and final sampled point is a candidate feature in between DH' . In case the intensities of sampled points are not monotonically increasing, the candidate feature value is taken as zero. Note that for every link centre point like D , we can get four such candidate feature values testing increasing intensities along DH, DH', DV and DV' . The maximum of these four feature values is taken as one of the features of the link. This feature is an indication of local minima of an image point and averaging the pixel values in between the sampled points averages the noise to certain extent.

For the next feature, we calculate the maximum range of

monotonically increasing or decreasing intensity along the sampled points S_1, S_1 and S_2 etc. The minimum of these ranges along four directions DH, DH', DV and DV' is the second feature of the corresponding link, in this case AB . This feature is an indicator of the thickness of a probable edge when a link is in between the object and the background pixel clusters.

Therefore, the training module consists of a feature set (f_1, f_2) of each link as noted above and for every pair of features (f_1, f_2) , there exists a label to the link 0, 1 or 2 depending on the location of the link within object, background or their boundaries respectively. In order to estimate the cluster parameters corresponding to clusters of links within object, background or their boundaries, we apply 3-clusters fuzzy c -means clustering on $N \times 3$ training data set for N number of links and 3 columns corresponding to two features and class label column for each link. On convergence, the fuzzy c -means clustering provides a partitioned vector $U \in [0, 1]$ of size $3 \times N$ where r th column of U provides the fuzzy indices of r th link to the three possible clusters. The cluster centers are defined as, $C = \{C_{e,p}, e = 1, 2, 3 \text{ and } p = 1, 2, 3\}$ where $C_{e,p}$ represents p th co-ordinate of e th cluster center in 3D space. The closeness of the p th feature of r th link $f_{r,p}$ with respect to the p th co-ordinate of e th cluster is defined by Gaussian function,

$$G(f_{r,p}, C_{e,p}, \sigma_{e,p}) = e^{-\frac{(f_{r,p} - C_{e,p})^2}{(\sigma_{e,p})^2}} \quad (1)$$

where $\sigma_{e,p} = \frac{1}{N_e} \sum_{r=1}^{N_e} (f_{r,p} - C_{e,p})^2$. The closeness of features of r th link with respect to e th cluster center is defined by,

$$\psi_{r,e} = \prod_{p=1}^P G(f_{r,p}, C_{e,p}, \sigma_{e,p}). \quad (2)$$

Now if features of r th link belong to e th cluster then output label ϕ for e th cluster is found from the linear regression of input features $f_{r,p}, p = 1, \dots, P$,

$$\phi_{r,e} = W_{e,0} + \sum_{p=1}^P W_{e,p} f_{r,p}, \quad (3)$$

where W is the regression coefficient of e th cluster. Since, a particular r th link has membership to each of the three clusters, the final label of that particular link is given by weighting the label of the link with its membership value.

$$\bar{N}(r) = \sum_{e=1}^K \phi_{r,e} \psi_{r,e}. \quad (4)$$

In case of training, the value of $\bar{N}(r)$ calculated from (4) may not be identical with $N(r)$ obtained from the training images. Therefore, the total error for all the link can be defined as,

$$E = \sum_{r=1}^N \|\bar{N}(r) - N(r)\|^2 \quad (5)$$

The Gaussian parameters $C_{e,p}, \sigma_{e,p}$ and regression coefficients $W_{e,p}$ for e th cluster are tuned by minimizing the error of (5) iteratively with the help of gradient decent method,

$$C_{e,p}(t+1) = C_{e,p}(t) + \eta_C \frac{\partial E}{\partial C_{e,p}} \Big|_{C_{e,p}=C_{e,p}(t)} \quad (6)$$

$$\sigma_{e,p}(t+1) = \sigma_{e,p}(t) + \eta_\sigma \frac{\partial E}{\partial \sigma_{e,p}} \Big|_{\sigma_{e,p}=\sigma_{e,p}(t)} \quad (7)$$

$$W_{e,p}(t+1) = W_{e,p}(t) - \eta_W \frac{\partial E}{\partial W_{e,p}} \Big|_{W_{e,p}=W_{e,p}(t)} \quad (8)$$

where t denotes the iteration number and η_C, η_σ and η_W are the learning parameters. We stop the iteration when (5) is less than a small preset number ϵ . The overall training algorithm is given as follows,

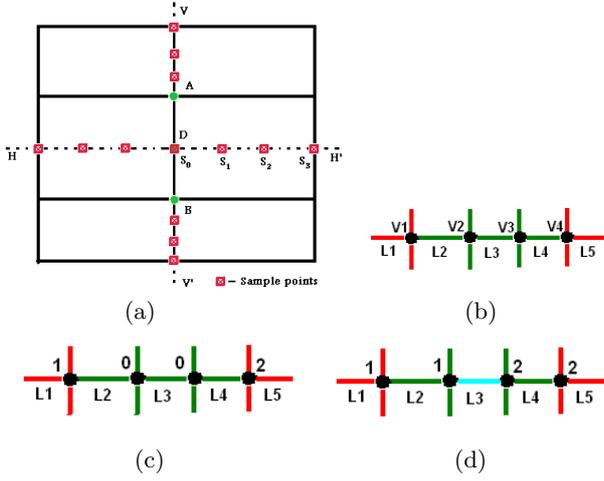


Figure 1: (a): Sampled points for a vertical link. (b)-(d): Link deletion between two conjoint objects.

```

1   Input:  $f, W_{e,l,p}$  randomly chosen;
2   Compute  $C = C_{e,p}$ , for given  $K$ ;
3   do{
4     for  $r = 1$  to  $N$ {
5       for  $e = 1$  to  $K$ {
6         for  $p = 1$  to  $P$ {
7           Compute  $\sigma_{e,p}, G(f_{r,p}, C_{e,p}, \sigma_{e,p}), \psi_{r,e}$ ;
8           }/* end: for  $r = 1$  to  $N$  */
9           Compute  $\phi_{r,e}, \bar{N}(r)$ ;
10          }/* end: for  $e = 1$  to  $K$  */
11          Compute  $E, \frac{\partial E}{\partial C_{e,p}}, \frac{\partial E}{\partial \sigma_{e,p}}, \frac{\partial E}{\partial W_{e,l,p}}$ ;
12          }/* end: for  $p = 1$  to  $P$  */
13          Tune  $C_{e,p}, \sigma_{e,p}, W_{e,l,p}$  with (6)-(8);
14        while( $E < \epsilon$ ).

```

In the next section we present the scheme for evolution of the membrane and how the learning using classifier helps in image segmentation.

2.2 Membrane evolution and link deletion

The active membrane evolves minimizing the energy [3],

$$E(V) = \int_0^1 \int_0^1 \left[\alpha \left(\left\| \frac{\partial V}{\partial r} \right\|^2 + \left\| \frac{\partial V}{\partial s} \right\|^2 \right) + \beta P(V) \right] dr ds. \quad (9)$$

The first two terms in the right hand side are the internal energy terms dependent on the amount of deformation of the membrane. The last term is the external energy term dependent on the image characteristics. The weights α and β depend on the local image intensity. The external energy $P(V)$ depends on the image gradient and also on the difference of the height of the local membrane and the image intensity surface. Minimization of $E(V)$ is done by

$$\frac{\partial V}{\partial t} + \frac{\partial E(V)}{\partial V} = 0, \quad (10)$$

and with the help of Finite Element Method (FEM) we get [3],

$$\begin{aligned} \alpha \left(I \left(X_{i,j}^{t-1}, Y_{i,j}^{t-1} \right) \right) \left(4X_{i,j}^t - X_{i-1,j}^t - X_{i+1,j}^t - X_{i,j-1}^t - X_{i,j+1}^t \right) &= \beta \left(I \left(X_{i,j}^{t-1}, Y_{i,j}^{t-1} \right) \right) \frac{\partial P}{\partial X}, \text{ and} \\ \alpha \left(I \left(X_{i,j}^{t-1}, Y_{i,j}^{t-1} \right) \right) \left(4Y_{i,j}^t - Y_{i-1,j}^t - Y_{i+1,j}^t - Y_{i,j-1}^t - Y_{i,j+1}^t \right) &= \beta \left(I \left(X_{i,j}^{t-1}, Y_{i,j}^{t-1} \right) \right) \frac{\partial P}{\partial Y}. \end{aligned} \quad (11)$$

The FEM vertices are indexed using (i, j) and the corresponding image pixel value at time t is given by $I(X_{i,j}^t, Y_{i,j}^t)$. The external energy components along different axes are defined as,

$$\begin{aligned} \frac{\partial P(V_{i,j})}{\partial X_{i,j}} &= - \sum_{\eta} \sum_{k=0}^q N(k) \nabla_x |\nabla I|, \text{ and} \\ \frac{\partial P(V_{i,j})}{\partial Y_{i,j}} &= - \sum_{\eta} \sum_{k=0}^q N(k) \nabla_y |\nabla I|. \end{aligned} \quad (12)$$

In (12) the gradient of the Gaussian convolved image is $|\nabla I|$. Instead of taking straightforward image gradient, we have utilized the gradient vector flow [7] of Gaussian convolved image gradients $GVF(|\nabla I|)$ for the external forces in x and y directions. The external energy at one particular vertex is calculated after summing external energies of its $4N$ vertices specified by domain η . The weight of the external energy at the k th point out of total q number of discrete points in the inter-vertex distance is $N(k)$.

The weights $\alpha(I)$ (and similarly $\beta(I)$) are taken as linear function,

$$\alpha(I(X, Y)) = \frac{I(X, Y)}{\text{Range}(I)} \times \alpha_{high}, \text{ and} \quad (13)$$

$$\beta(I(X, Y)) = \frac{\text{Range}(I) - I(X, Y)}{\text{Range}(I)} \times \beta_{high}, \quad (14)$$

where $(\alpha_{high}, \beta_{high})$ are set experimentally and $\text{Range}(I)$ is the range of intensity of image I . Rewriting (11) and simplifying the notation, we ultimately get the active membrane evolution equation [3],

$$\begin{aligned} \frac{V^t - V^{t-1}}{\Delta t} + \alpha \cdot *AV^t &= \beta \cdot *Fv \Rightarrow \\ (I + \Delta t \times \alpha \cdot *A)V^t &= V^{t-1} + \Delta t \times \beta \cdot *Fv, \end{aligned} \quad (15)$$

where A is a $t_v \times t_v$ square stiffness matrix. The total number of FEM vertices in the membrane is t_v which is equal to $(m+1)(n+1)$. V is $t_v \times 2$ position matrix. The elements of each row of V give the position vector of a FEM vertex of the membrane. Fv is the $t_v \times 2$ force matrix at the membrane vertices. The elements of each row of Fv give the force vector of a FEM vertex of the membrane. The operation ' \cdot ' denotes element wise multiplication. We assume that we have a priori estimation V^{t-1} at iteration $(t-1)$ for the current iteration t . I and Δt in (15) denote identity matrix and time step respectively.

In [3] for each iteration we compare the inter-vertex Euclidean distance between a pair of neighbouring vertices (i, j) and (k, l) with the $D_{i,j}^{k,l}$ given as,

$$D_{i,j}^{k,l} = g_D + \frac{I(X_{i,j}, Y_{i,j}) + I(X_{k,l}, Y_{k,l})}{2 \times \text{Range}(I)} \times (D_{high} - g_D), \quad (16)$$

where, g_D is the initial inter-vertex distance of the membrane and D_{high} are application dependent preset constants. Due to membrane evolution if distance between vertices (i, j) th and (k, l) becomes greater than $D_{i,j}^{k,l}$, then we delete the link between the vertices (i, j) and (k, l) . As a result each of the two diagonal elements of A due to $A_{i,j}$ and $A_{k,l}$ rows are reduced by one and $A[A_{i,j}, A_{k,l}]$ and $A[A_{k,l}, A_{i,j}]$ elements are changed to zero. In deletion process, if (i, j) th vertex of the grid matrix is deleted then the row and column corresponding to $A_{i,j}$ is deleted and A is reduced in size by one both in row and column directions. Deletion of a link deletes the two elements sharing the link. Also, deletion of two elements on the two sides of a link deletes the link. As A is a symmetric matrix the value of $A[A_{i,j}, A_{k,l}]$ and $A[A_{k,l}, A_{i,j}]$ are always identical and changing it to 0 we

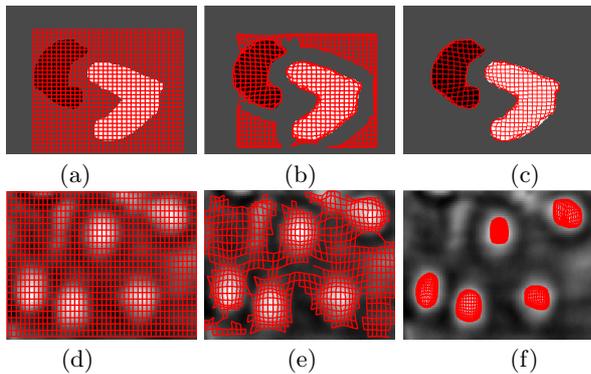


Figure 2: (a): Initial membrane on synthetic objects. (b): Intermediate stages of membrane evolution after 20 iterations. (c): Final segmentation using topology adaptive membrane after 150 iterations. (d): Multiple white blood cells with initial active membrane. (e): After 15 iterations of the membrane. (f): After segmentation using 180 iterations.

can delete the link or connectivity between vertices (i, j) and (k, l) . We stop the evolution of membrane when for same number of membrane vertices in previous and current iteration $\|V^t - V^{t-1}\| < \epsilon$ where ϵ is a very small preset number. So once the membrane evolution stops, we get different membrane pieces representing deformable objects present in the scene. An example of such membrane evolution is shown in Fig. 2, where we take $D_{high} = 3g_d$, $g_d = 3$, $\alpha_{high} = 0.03$, and $\beta_{high} = 0.01$.

However, in the proposed method given an image frame, first features (f_1, f_2) are evaluated as described in section 2.1. Then using (3) and (4), labels of each of the links of the membrane is evaluated either as object link, background link or a boundary link. Next, the background links are deleted from the membrane following the links and element deletion procedure described in the previous paragraph. Next the active membrane is evolved following (15) with the hard constraint in the stiffness matrix as object links do not change its shape while boundary links are deformed based on gradient vector flow in order to capture the shape of the objects.

The proposal however faces problem when the images of objects to be segmented are touching each other. In that case a set of boundary links exists between two sets of object links unlike the normal situation where background links also exist between sets of boundary links encompassing one or more objects. We discuss such situation in the next section.

2.3 Boundary Link Deletion

For touching object segmentation, we take the help of diffusion mechanism where vertex labels of objects are propagated in the neighbourhood to determine boundary of touching objects. We can explain this with the help of an example.

Let us mark both vertices of all boundary links with tag 0 and mark vertices of background links with a large number, say L . All vertices of object links are marked with a unique tag representing the connected component to which this particular object link belongs. That is, say, if there are two objects or connected components, vertices of these

connected components (which are also the vertices of object links) may be marked with tags 1 and 2 respectively. For the current problem, only tag 0 vertices exist between tag 1 and tag 2 vertices but no tag L vertices exist between tag 1 and tag 2 vertices. We need to find out those tag 0 vertices along which the membrane could be split to identify the boundary between two touching objects. This could have been easier in case there exists tag L vertices between tag 1 and tag 2 vertices. In that case deleting tag L vertices straightaway separates membrane for each of the connected component.

The proposed diffusion algorithm is explained with the help of Figs. 1(b)-(d). In the figure, a set of boundary links ($L2, L3, L4$) are connected to two object links ($L1$ of object 1 and $L5$ of object 2). Therefore, vertices $v1$ and $v4$ of Fig. 1(b) are tagged 1 and 2 respectively as shown in Fig. 1(c). By design vertices $v2$ and $v3$ are vertices of boundary links. Now we update each tag 0 vertex with tag of its nonzero-tagged neighbourhood vertices iteratively. So $v2$ is changed to 1 and $v3$ is changed to 2 as shown in Fig. 1(d). This process continues till there is no further change in the tag value. While updating the tag value of a vertex its $4N$ neighbours are searched in anti-clockwise direction. At a particular iteration, if a tag 0 vertex is surrounded by tag 0 vertices in $4N$, its tag is not changed in that particular iteration. This entire process can be given an algorithmic form.

v_t : tag value of a vertex.

adj_graph : Graph storing $4N$ vertices of each vertex of an element.

```

1  pre_sum=Sum(v_t); /* Evaluate sum of all v_t */
2  do{
3    sum = 0;
4    for count=1 to total_vertex{
5      if v_t(count,1)==0{
6        for count1=1 to 4{
7          temp=adj_graph(count,count1);
8          if v_t(temp,1)≠(0 or L){
9            v_t(count,1)=v_t(temp,1);
10           sum=sum+v_t;
11           break;}
12          } /* end: for count1=1 to 4 */
13         } /* end: if v_t(count,1)==0 */
14        } /* end: for count=1 to total_vertex */
15    }while(sum ≠ pre_sum)

```

Once the above diffusion algorithm changes the tag of all vertices, all links having different tags at their vertices, are deleted to segment the connected component approximating each of the objects. The segmented connected components are then evolved using (15) for accurate segmentation and extraction of multiple objects in the scene.

In the next section we present the segmentation result of our proposed methodology.

3. RESULTS

We have implemented the proposed methodology to detect oil sand rock pieces seen on a dirt-filled conveyor belt. These rock pieces are crushed for extracting crude oil and estimation of sizes of the rock pieces is important indicator for estimation of the crude oil production. Clearly, the con-

Table 1: Calculated $C_{e,p}$ and $\sigma_{e,p}$ from (6) and (7) respectively.

	$C_{e,p}$			$\sigma_{e,p}$		
	$e=1$	$e=2$	$e=3$	$e=1$	$e=2$	$e=3$
$p=1$	5.6	6.1	104.7	25.4	27.1	16.6
$p=2$	1.5	72.1	77.6	12.2	15.4	17.2

Table 2: Calculated $W_{e,p}$ from (8)

	$p=0$	$p=1$	$p=2$
$e=1$	0.667	-0.067	-0.278
$e=2$	0.667	0.111	.0093
$e=3$	0.333	0.0032	0.0043

trasts of these images make it a difficult candidate for image segmentation.

We first take 5 oil sand rock piece images along with their ground truths as training images sampled randomly from a set of 100 oil sand rock piece images. Two typical ground truth images of Figs. 3(a) and 4(a) are shown in Figs. 3(b) and 4(b), respectively. We place active membranes on the images, calculate features f_1 , f_2 and label the links of the membrane as defined in Section 2.1. For the current set of images, the training algorithm described in Section 2.1 calculates $C_{e,p}$, $\sigma_{e,p}$ and $W_{e,p}$ taking ϵ , η_C , η_σ , η_W equal to 0.001 as given in Tables 1 and 2, respectively. We use the leave-one-out cross validation to compute the parameters in Tables 1 and 2.

A set of test images having two, three and four rock pieces are shown in Figs. 3(a), (g) and (l), respectively. We first initialize active membrane as shown in Figs. 3(c), (h) and (m). The results of the use of trained parameters of Tables 1, and 2 on the links of Figs. 3(c), (h) and (m) are shown in Figs. 3(d), (i) and (n), respectively. In Figs. 3(d), (i) and (n) the red links denote background links, while the green links denote boundary and object links. We take label value less than 0.5 as 0, greater than 1.5 as 2 and take label value between 0.5 to 1.5 as 1 for marking background, boundary and object links. After deleting all the background links (marked red) the membrane is split into different segments as shown in Figs. 3(e), (j) and (o). Finally, the segments of Figs. 3(e), (j) and (o) are evolved following (15) to accurately identify objects as shown in the Figs. 3(f), (k) and (p).

In Fig. 4(a), we have taken a test image having two conjoint rock pieces. Initialization of active membrane is shown in Fig. 4(c). The result of the use of trained parameters of Tables 1 and 2 on the links of Fig. 4(c) is shown in Fig. 4(d). Here, the blue, green and red links are background, boundary and object links respectively. The magenta links of Fig. 4(e) are the result of vertex tag propagation algorithm of Section 2.3. These links have different vertex tags for their end vertices. After deleting all the background links (marked blue) and magenta coloured links, the membrane is split into two segments as shown in Fig. 4(f). Finally, the segments of Fig. 4(g) are evolved following (15) to accurately identify the objects as shown in the Fig. 4(f). The test result for another conjoint rock pieces is shown in Figs. 4(h)-(i).

In all the above examples, we take $\alpha_{high} = 0.03$, and $\beta_{high} = 0.01$. The entire approach was implemented in Mat-

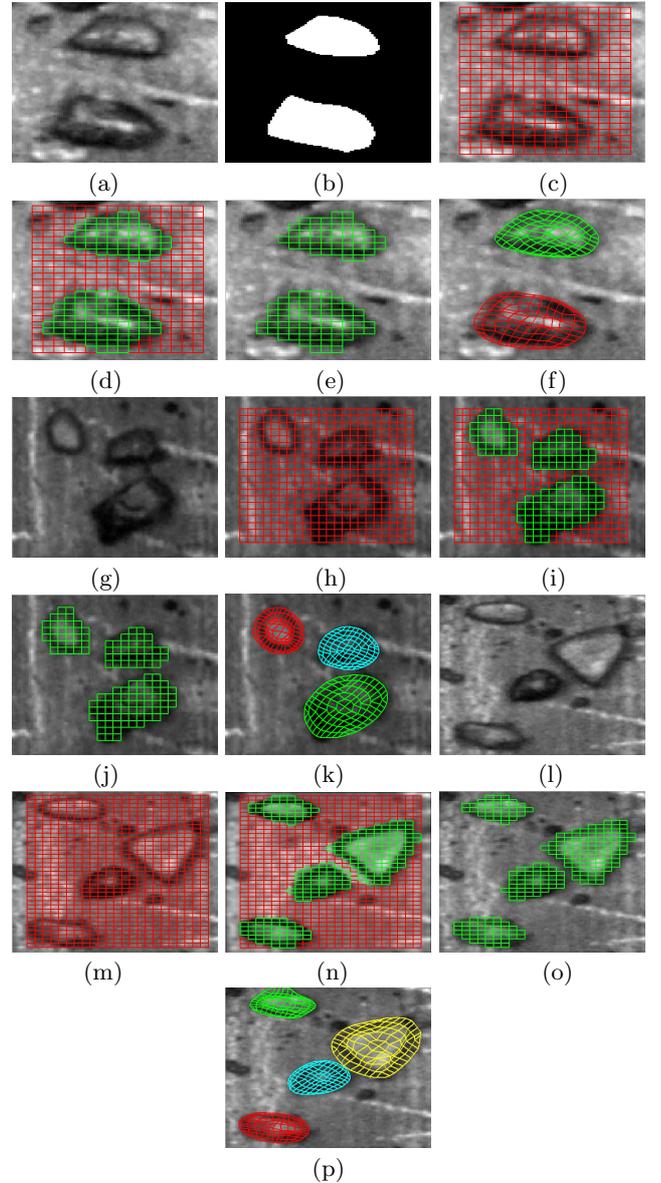


Figure 3: (a), (g), (l): Rock images, (b): Ground truth image of conjoint rock pieces of Fig. 3(a), (c), (h), (m): Initial Active membrane, (d), (i), (n): Link type identification by trained Fuzzy classifier, (e), (j), (o): Rough estimated objects after deletion of the boundary links, (f), (k), (p): Final results after membrane evolution (50 iterations).

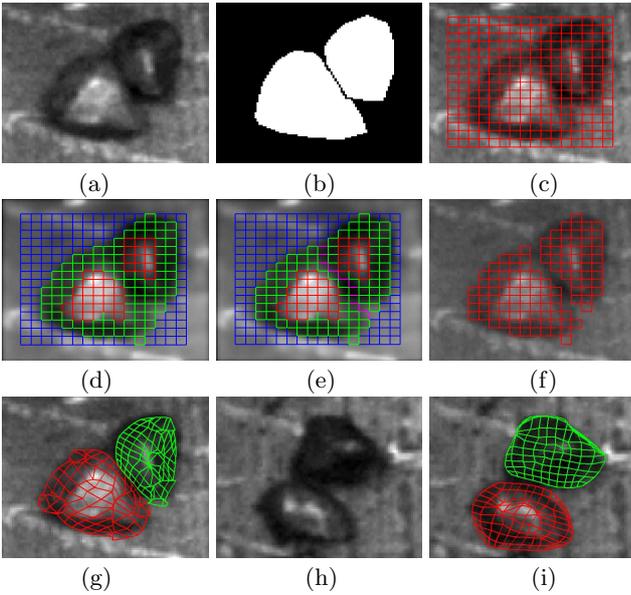


Figure 4: (a): Two conjoint rock pieces, (b): Ground truth image of conjoint rock pieces of Fig. 4(a), (c): Initial Active membrane, (d): Link type identification by trained Fuzzy classifier, (e): Boundary links detection by diffusion mechanism, (f): Rough estimated objects after deletion of the boundary links, (g): Final results after membrane evolution (70 iterations). (h): Two conjoint rock pieces, (i): Final results after membrane evolution (70 iterations).

lab 7 in Pentium 4, 2.1 GHz PC. In the next section we compare our method with straightforward implementation of level sets [1], topology adaptive snake [4] and topology adaptive active membrane [3].

3.1 Comparison

As stated in introduction, the object in Figs. 4(a) is an image of multiple conjoint objects where objects and background have similar spectral characteristics. So, the active membrane implementation without training [3] completely fails to segment these objects as shown in Figs. 5(a)-(b). The level set implementation [1] [2] or topology adaptive active contour [4] cannot identify them correctly as shown in Fig. 5(c)-(j). In these cases either the spectral characteristics confuses the evolved curve or the outer edge cannot see the edges internal to the outer contour.

In order to establish the segmentation accuracy, we have taken the help of segmentation score metric reported in [6]. This score metric is a form of local intersection-over-union of pixel areas whereby both errors at the pixel level and object level are penalized. The score metric is defined as,

$$\psi(A, B) = \sum_j^m \left[\sum_i^n \left(\frac{|A_j \cap B_i|}{|A_j \cup B_i|} \frac{B_i}{\bigcup_{|A_j \cap B_i| \neq 0} B_i} \right) \frac{A_j}{\bigcup_j A_j} \right] \quad (17)$$

where A_j is a connected component in image A and B_i is a connected component in image B . Since, $\psi(A, B) \neq \psi(B, A)$, $\min(\psi(A, B), \psi(B, A))$ is taken as the conservative measure of the score metric.

In Table 3 we show the segmentation score metric and

Table 3: Segmentation score metric and number of detected object by proposed and its competitive methods.

	Score metric	Object detected
Proposed method	88.1	2
Active membrane [3]	60.7	1
Active contour [4]	45.3	1
Level set [1]	41.2	1
Multiphase level set [2]	39.8	5

number of detected object by the proposed and its competitive methods. Table 3 shows, as expected, the proposed approach performs much better than the topology adaptive active membrane [3], level set implementation [1] [2], and topology Adaptive Active Contour [4].

4. CONCLUSIONS

We have developed a fuzzy-rule base training scheme for active membrane evolution. With appropriate train, it can identify portions of active membrane describing separate objects in a scene, especially when the objects are touching each other. Our goal is to use this technique in a video where we can track low contrast moving objects, that often touches or occludes each other.

5. REFERENCES

- [1] T. F. Chan and A. Vase. Active contours without edges. *IEEE Trans. on Image Processing*, 10(2):266–278, 2001.
- [2] T. F. Chan and A. Vase. A multiphase levelset framework for image segmentation using mumford and shah model. *International Journal of Computer Vision*, 50(3):271–293, 2002.
- [3] S. K. Das and D. P. Mukherjee. Topology adaptive active membrane. In *Lecture Notes in Computer Science*, volume 4815, pages 95–102, 2007.
- [4] T. McInerney and D. Terzopoulous. T-snake: Topology adaptive snakes. *Medical Image Analysis*, 4:73–91, 2000.
- [5] N. R. Pal, V. K. Eluri, and G. K. Mondal. Fuzzy logic approaches to structure. preserving dimensionality reduction. *IEEE Trans. Fuzzy Syst.*, 10(3):277–287, 2002.
- [6] M. Polak, H. Zhang, and M. Pi. An evaluation of metric for image segmentation of multiple objects. *Image and Vision Computing*, 27:1223–1227, 2009.
- [7] C. Xu and J. Prince. Snakes, shapes, gradient vector flow. *IEEE Trans. on Image Processing*, 7(3):359–369, 1998.

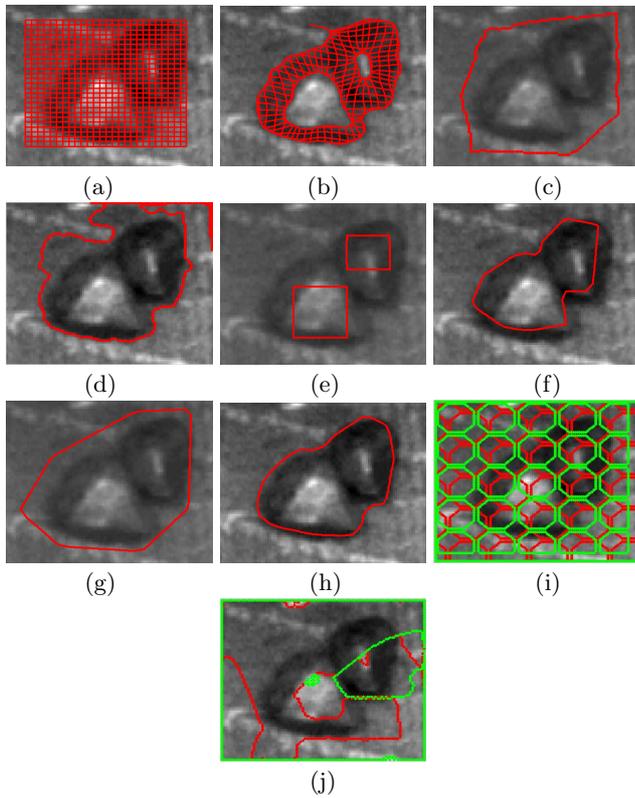


Figure 5: (a): Initial Active membrane [3] on conjoint objects (of Fig. 4(a)), (b): Segmentation of two conjoint rock pieces by [3]. (c): Initial Level set [1] on conjoint objects (of Fig. 4(a)), (d): Segmentation of two conjoint rock pieces by [1]. (e), (g): Initial Active contour [4] conjoint objects (of Fig. 4(a)), (f), (h): Segmentation of two conjoint rock pieces by [4], (i): Initial Level set [2] on conjoint objects (of Fig. 4(a)), (j): Segmentation of two conjoint rock pieces by [2].