Segmentation of Images using Level Set Analysis

Partha Pratim Mohanta
Electronics and Comm. Sc. Unit
Indian Statistical Institute, Calcutta
E-mail: ppmohanta@isical.ac.in.

Arjun Sen
National Institute of Management Calcutta
Email: sen_arjun@gettos.com.

Dipti Prasad Mukherjee
Electronics and Comm. Sc. Unit
Indian Statistical Institute, Calcutta
E-mail: dipti@isical.ac.in.

Abstract

We propose an image segmentation technique using level set analysis. Image level set is the binary decomposition of a gray level image. Connected components in the level set, less than a pre-defined size are removed from the level set. Based on level set topology an exposed connected component is defined in the level set. These exposed connected components are merged based on a proximity value derived between the neighboring components. This proximity measure is a function of intensity difference and the shared boundary length between image regions described by the connected components present in the level set. The result obtained using the proposed method is shown and compared with a similar process along with the accuracy measure of the segmentation.

1. Introduction

We are proposing an integrated image level set based approach for image segmentation. The image level set is the hierarchical binary decomposition of the intensity image. Level set analysis was used to represent contrast invariant image features [1]. The binary decomposition generates connected components at every level set. A preset area threshold controls the size of the connected components at every image level set. In this case the segmentation is achieved after appropriate merging of the connected components generated at different image level sets. The merging process is preceded by the definition of an exposed component which is present in a particular level set but not included in level sets higher than the current one. The size of the segmented region is dictated by the area threshold set during level set analysis. The merging of components is based on a proximity measure that describes the affinity between components within the image level set. This proximity measure helps in region merging for effective image segmentation.

The level sets describe a unique representation of the image satisfying properties like causality and edge localization [2]. The proposed segmentation approach based on image level set therefore satisfies these important properties as well. In [2] image segmentation is achieved by sequential processing of area morphological operation followed by clustering using fuzzy *c*-means approach. In this case the component merging is achieved as we proceed along the level set stack of any intensity image. Also, we do not need any a priori knowledge of number of clusters present in the image.

In [3], local pixel variation is utilized for image segmentation by maximizing intra-region homogeneity. For the current implementation, we have utilized level set components to calculate local intensity variation. In addition, the common boundary length between two segments is considered as a factor for possible region merging.

In the next section, we first present the appropriate definitions and properties of image level set necessary to describe our algorithm. This is followed by the proposed segmentation algorithm in Section 3. The results obtained using the proposed algorithm is given in Section 4 followed by conclusion.

2. Image Level Set Representation

In this section we first define the generation and properties of level set that are necessary for binary component merging.

Let Ψ be the set of all possible intensity values for an intensity image **I** with *m* rows and *n* columns. Typically, $\Psi = \{0,1,2,\cdots,255\}$ for an 8-bit intensity image. Let $Z_m = \{1,2,\cdots,m\}$ and $Z_n = \{1,2,\cdots,n\}$. Let λ be a real function $\ni \lambda : Z_m \times Z_n \rightarrow \Psi$. In other words, the function λ represents the spatial information of the image. If $z \in Z_m \times Z_n$ be any element of image **I** and the corresponding intensity value $\zeta \in \Psi$, then we can define $\lambda(z) = \zeta$.

2. 1 Level set and its properties

We now define the level sets and some of its properties.

Definition: For $\zeta \in \Psi$, consider the following sets $\omega_{\zeta \in \Psi} = \{x \in \Psi : x \geq \zeta\}$ and

$\theta_{\zeta\in\Psi} = \{z \in Z_m \times Z_n : \exists y \in \omega_{\zeta\in\Psi} \ni \lambda(z) = y\}$.

Consider the function $L_\zeta : Z_m \times Z_n \to \{0,1\} \ni$

$$L_\zeta(z) = \begin{cases} 1 & \text{if } z \in \theta_{\zeta\in\Psi} \\ 0 & 0 \text{ otherwise} \end{cases}$$

The function L_ζ may be perceived as a matrix containing only 1's and 0's corresponding to the row and column positions defined in L_ζ. The matrix representation of L_ζ will no doubt yield a binary image. This binary image is called a *level set* and ζ is the corresponding *level*. A trivial statement $\forall \zeta \in \Psi$ $\exists$ L_ζ that guarantees $|\Psi|$ level sets in the image λ.

Now we prove some of the basic properties of the level sets.

Lemma 1: $\forall z \in Z_m \times Z_n, \forall \alpha, \beta \in \Psi : \alpha > \beta$ then $L_\alpha(z) = 1 \Rightarrow L_\beta(z) = 1$ but not conversely.

Proof: Let $\alpha, \beta \in \Psi$ be any elements such that $\alpha > \beta$.
Let $z \in Z_m \times Z_n$ be any element $\ni L_\alpha(z) = 1$.
We have $\omega_{\alpha\in\Psi} = \{x \in \Psi : x \geq \alpha\}$ and $\omega_{\beta\in\Psi} = \{x \in \Psi : x \geq \beta\}$. Let $u \in \omega_{\alpha\in\Psi}$ be any element.
Now, $u \geq \alpha \Rightarrow u > \beta \Rightarrow u \in \omega_{\beta\in\Psi}$. Thus $\omega_{\alpha\in\Psi} \subset \omega_{\beta\in\Psi}$ (since $\beta < \alpha$ and $\beta \notin \omega_{\alpha\in\Psi}$ but $\beta \in \omega_{\beta\in\Psi}$). We also have
$\theta_{\alpha\in\Psi} = \{\eta \in Z_m \times Z_n : \exists y \in \omega_{\alpha\in\Psi} \ni \lambda(\eta) = y\}$ and $\theta_{\beta\in\Psi} = \{\eta \in Z_m \times Z_n : \exists y \in \omega_{\beta\in\Psi} \ni \lambda(\eta) = y\}$.
Let $v \in \theta_{\alpha\in\Psi}$ be any element. Thus,
$\exists y \in \omega_{\alpha\in\Psi} \ni \lambda(v) = y$. But by (1) $y \in \omega_{\beta\in\Psi}$. So,
$\exists y \in \omega_{\beta\in\Psi} \ni \lambda(v) = y \Rightarrow v \in \theta_{\beta\in\Psi}$
Therefore, $v \in \theta_{\alpha\in\Psi} \Rightarrow v \in \theta_{\beta\in\Psi}$. Hence $\theta_{\alpha\in\Psi} \subseteq \theta_{\beta\in\Psi}$. Now $L_\alpha(z) = 1$ iff $z \in \theta_{\alpha\in\Psi}$ and $z \in \theta_{\alpha\in\Psi} \subseteq \theta_{\beta\in\Psi} \Rightarrow z \in \theta_{\beta\in\Psi} \Leftrightarrow L_\beta(z) = 1$.
Hence, $L_\alpha(z) = 1 \Rightarrow L_\beta(z) = 1$.

Conversely, if $q \in \theta_{\alpha\in\Psi}$ but $q \notin \theta_{\beta\in\Psi}$,
$\exists y \in \omega_{\alpha\in\Psi} \ni \lambda(q) = y$ and $y \in \omega_{\alpha\in\Psi} \subset \omega_{\beta\in\Psi}$.
Therefore, $q \in \theta_{\beta\in\Psi}$ is a contradiction to the assumption. $\theta_{\alpha\in\Psi} \subseteq \theta_{\beta\in\Psi}$ but $\theta_{\beta\in\Psi} \not\subset \theta_{\alpha\in\Psi}$. Thus, $z \in \theta_{\beta\in\Psi} \not\Rightarrow z \in \theta_{\alpha\in\Psi}$. Hence, $L_\beta(z) = 1 \not\Rightarrow L_\alpha(z) = 1$. Hence proved.

Lemma 2: $\forall z \in Z_m \times Z_n, \forall \alpha, \beta \in \Psi : \alpha > \beta$ then $L_\beta(z) = 0 \Rightarrow L_\alpha(z) = 0$ but not conversely.

Proof: From the definition of level set we have
$L_\beta(z) = 0$ iff $z \notin \theta_{\beta\in\Psi}$.
And from Lemma 1, $\alpha > \beta \Rightarrow \theta_{\alpha\in\Psi} \subseteq \theta_{\beta\in\Psi}$.
So, if $z \notin \theta_{\beta\in\Psi}$ then $z \notin \theta_{\alpha\in\Psi}$.
Also $z \notin \theta_{\alpha\in\Psi} \Leftrightarrow L_\alpha(z) = 0$.
Thus, $L_\beta(z) = 0 \Rightarrow L_\alpha(z) = 0$
Conversely, if $\theta_{\alpha\in\Psi} \subset \theta_{\beta\in\Psi}$ then $\exists$ at least one $z \in \theta_{\beta\in\Psi}$ but $z \notin \theta_{\alpha\in\Psi}$.
And $L_\beta(z) = 1$ iff $z \in \theta_{\beta\in\Psi}$ but $L_\alpha(z) = 0$ iff $z \notin \theta_{\alpha\in\Psi}$. Hence $L_\alpha(z) = 0 \not\Rightarrow L_\beta(z) = 0$
The only case when $L_\alpha(z) = 0 \Rightarrow L_\beta(z) = 0$ is that if $\theta_{\alpha\in\Psi} = \theta_{\beta\in\Psi}$. Thus $L_\alpha(z) = 0 \not\Rightarrow L_\beta(z) = 0$ in general. Hence proved.

Lemma 3: $\forall z \in Z_m \times Z_n, \lambda(z) = \alpha \in \Psi,$ then $L_\alpha(z) = 1$ and $L_{(\alpha+1)}(z) = 0$.

Proof: Let $z \in Z_m \times Z_n$ be any element and $\lambda(z) = \alpha$.
Thus we have $\alpha \in \omega_{\alpha\in\Psi}$ and $z \in \theta_{\alpha\in\Psi}$.
Also $z \in \theta_{\alpha\in\Psi} \Leftrightarrow L_\alpha(z) = 1$.
$(\therefore \theta_{\alpha\in\Psi} = \{\eta \in Z_m \times Z_n : \exists y \in \omega_{\alpha\in\Psi} \ni \lambda(\eta) = y\})$. Now we consider $\omega_{(\alpha+1)\in\Psi}$, $\alpha < \alpha + 1 \Rightarrow \alpha \notin \omega_{(\alpha+1)\in\Psi}$.
Given $\lambda(z) = \alpha \in \Psi,$ there cannot exist any y such that:
$\nexists\ y \in \omega_{(\alpha+1)\in\Psi} : \lambda(z) = y$.
So, $z \notin \theta_{(\alpha+1)\in\Psi} \Leftrightarrow L_{(\alpha+1)}(z) = 0$. Hence proved.

Theorem 1. The total pixel-wise sum of all the level sets is equal to the image.
The theorem can be restated as to prove that:
$\forall z \in Z_m \times Z_n, \sum_{\zeta\in\Psi} L_\zeta(z) = \lambda(z)$.
Proof: Let $\Psi = \{\zeta_1, \zeta_2, ..., \zeta_{|\Psi|}\}$.
Let $z \in Z_m \times Z_n$, $j \in \{1,2,........,|\Psi|\}$ be any elements $\ni \lambda(z) = \zeta_j$.
Consider $\sum_{\zeta\in\Psi} L_\zeta(z) = L_{\zeta_1}(z) + L_{\zeta_2}(z) + ... +$
$L_{\zeta_j}(z) + L_{\zeta_{j+1}}(z) + ... + L_{\zeta_{|\Psi|}}(z)$.
From Lemma 3, $L_{\zeta_j}(z) = 1$ and $L_{\zeta_{j+1}}(z) = 0$.
From Lemma 1,

$L_{\zeta_{j-1}}(z)=1 \Rightarrow L_{\zeta_{j-2}}(z)=1 \Rightarrow ... \Rightarrow L_{\zeta_2}(z)=1$

$\Rightarrow L_{\zeta_1}(z)=1.$

From Lemma 2,

$L_{\zeta_{j+1}}(z)=0 \Rightarrow L_{\zeta_{j+2}}(z)=0 \Rightarrow ... \Rightarrow L_{\zeta_{|\Psi|-1}}(z)=0$

$\Rightarrow L_{\zeta_{|\Psi|}}(z)=0.$

$\sum_{\zeta\in\Psi} 1+1+1+........+1 \,(\zeta_j \text{ times})$

$+0+0+..........+0 \,(|\Psi|-\zeta_j \text{ times})$

$\Rightarrow \sum_{\zeta\in\Psi} L_\zeta(z)=\zeta_j=\lambda(z)$.

Hence proved.

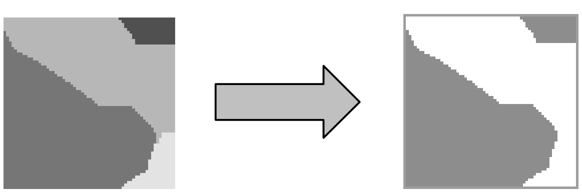
Thus we have established that the collection of level sets of an image λ can be summed to recreate the image. The ordered collection of level sets can be defined as a function:

$L_\lambda : \Psi \to \{L_\alpha : \alpha\in\Psi\}$ where $L_\lambda(\alpha)=L_\alpha \quad \forall\alpha\in\Psi$.

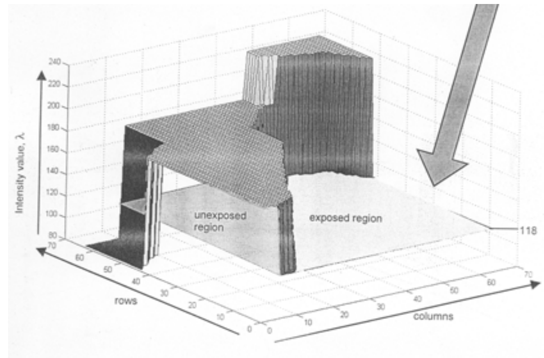
Now for an arbitrary level ζ, the corresponding level set L_ζ can be partitioned into a set of components by performing connected component analysis. Each component set contains elements of $\theta_{\zeta\in\Psi}$ which satisfy one of the connectivity constraints (4N or 8N connectivity) with other elements of the component.

Following the definition of level set it is obvious that for any two arbitrary levels, the level set corresponding to the lower level always contains the higher level set. For example, consider two levels α and β such that $\alpha,\beta\in\Psi$ and $\alpha>\beta$, the lower level set L_β always contains the higher level set L_α. Now if C_α and C_β be connected components within the level sets L_α and L_β, then either C_α and C_β are spatially disjoint or $C_\alpha \subseteq C_\beta$. But they can never intersect each other such that $C_\beta \subset C_\alpha$.

Thus since components of one level set are always contained in (or equal to) some component in a lower intensity level set we can characterise visible segments as the difference of components on consecutive level sets. These visible segments are often termed as *exposed components*. For a typical image Fig. 1(a) with its level set at intensity 118 (shown in Fig. 1(b)), the exposed component at level 118 is shown in Fig. 1(c).



(a) (b)



(c)

Fig. 1: (a) original image (b) level set at intensity 118 (c) graphical representation of exposed component.

In the next section we use the morphological and intensity features of binary components of the level sets to achieve image segmentation.

3. Proposed Method

So far we have discussed the level sets and their properties that are very useful towards the implementation of our segmentation algorithm. Utilizing these properties, our goal is to identify the relevant segments in the image without any prior knowledge of the number of segments present in the image.

We would like to control the size of the segmented region. Let *s* be the minimum area that a segment is permitted at the completion of the entire segmentation process. As level sets are binary images, so we can treat the 1's as foreground and 0's as background in the level set. We filter the foreground connected component of size less than *s* and replace them by background pixels. Identical operation is carried out for connected component of background pixels as they are replaced by foreground ones. In other words, this operation assigns 0s to the pixels of foreground components with size less than s and in the reverse process, assigns 1s to the background components with area less than s.

3.1 Component analysis

The level sets are now devoid of any foreground or background components of area less than the pre-specified size. The larger regions in this filtered image are much more homogeneous compared to the original image but there may exists some regions having the area less than the preset limit. The above fact can be explained clearly through the following example. Consider two consecutive intensities α and β such that $\alpha>\beta$ and the respective level sets L_α and L_β.

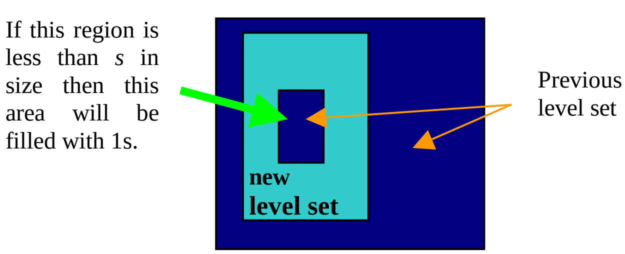
Let C_α and C_β be the connected components with areas $S(C_\alpha)$ and $S(C_\beta)$ respectively such that $C_\alpha \in L_\alpha, C_\beta \in L_\beta$. Naturally $C_\alpha \subset C_\beta$ and in the filtered image (i.e., the image which is found after summing all the level sets) the existence of C_α depends on its size. Now, after area based filtering, if $S(C_\alpha)$ permits C_α to survive then the intermediate region $\Delta S = S(C_\beta) - S(C_\alpha)$ also survives in the filtered image. But it is not assured that the region ΔS always exceeds the preset area limit *s*. Alternatively, C_α does not survive if $S(C_\alpha) < s$ and then the object containing connected component C_β in the original image is more homogeneous in the filtered image.

So to get a well segmented image we need to introduce a component merging process. Component merging essentially involves the identification of the immediate neighbours of a component under scrutiny. For identifying the most probable candidate for merging a *proximity value* is assigned to each neighbour. Among the various merging factors the intensity difference and the shared boundary length are are used for merging of two neighbouring components. These features are evaluated on the original image segment delineated by the topology of the component in the level set. The proximity of a component and its neighbour is defined as:

$$p = \left[B^{\frac{1}{n}} \right] \times \left[\frac{1}{\sqrt{2\pi}(\mu + 3\sigma)} e^{\left[\frac{\delta^2}{2(\mu+3\sigma)^2} \right]} \right],$$

where, B is the shared boundary length, δ is the intensity difference, μ and σ are the mean and standard deviation respectively of the global intensity difference and n is a positive integer typically greater than 2. The evaluation of μ and σ are given in the section 3.2.

In this approach each area filtered level set passes through the component merging process. This merging process is carried out on exposed components, which are bounded by neighbouring exposed components. An exposed component will always be revealed on the boundary of the lower level set provided that area filter has already been applied on the higher level set. The following figures clearly define the above fact.



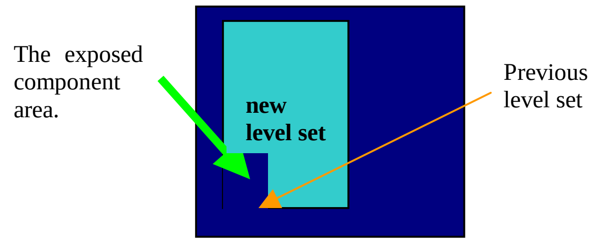


Fig. 2: Exposed components in two different situations.

As soon as a component is identified as an exposed component, the neighbours of this component are scrutinised and a relevant proximity value is assigned to each of the neighbour to find out the nearest neighbour. The nearest neighbour exists either in the upper level set or in the lower level set (but not in the same level set because that would violate the connectivity constraints imposed on each component). An exposed component is promoted or demoted to its nearest neighbour's level depending on whether the nearest neighbor exists in the upper or lower level set. The promotion is implemented by converting the 0's belonging to exposed component to 1's in each of the level sets lying between the nearest neighbour level set and the exposed component level set (including nearest neighbour level set). The demotion process is realized by converting the 1's belonging to exposed component to 0's in each of the level sets lying between the nearest neighbour level set and the exposed component level set (including exposed component level set).

It is trivial to prove and follows from Lemma 3 and Theorem 1 that the process of promotion and demotion ensures that the sum of the level sets will result in a valid intensity image. The complete algorithm is presented in the next section.

3.2 The algorithm

The implementation of the proposed method is done through the following steps:

1. Read gray scale intensity image into matrix λ.
2. An *absolute difference image*, $\Delta\lambda$, is computed from horizontal and vertical pixel intensity differences ($\Delta\lambda_h$ and $\Delta\lambda_v$ respectively). The Euclidean difference of magnitude of $\Delta\lambda_h$ and $\Delta\lambda_v$ yields $\Delta\lambda$:

   $$\Delta\lambda = \sqrt{|\Delta\lambda_h|^2 + |\Delta\lambda_v|^2}$$

   Significant $\Delta\lambda$ value is considered as the edge condition ρ. The components for those $\Delta\lambda$

values are greater than ρ , are never considered for merging. For the current implementation the top 20 percentile values of $\Delta\lambda$ are considered as edge condition.

3. The mean μ_λ and standard deviation σ_λ of the range of $\Delta\lambda$ are calculated to evaluate proximity value *p*.
4. The following is executed for each of the level sets encountered at each intensity value, $\alpha \in \Psi$.
5. The level set L_α for current intensity value α is passed through the area filtering process. L_α is now devoid of any 1-region or 0-regoin having an area less than *s*.
6. L_α is placed upon the stack of other level sets L_λ and the each resulting component, with area less than *s* is tested for merging:
7. Neighbours of the current component are selected. If all the neighbours are exposed components, the algorithm proceeds as follows (else the next component is chosen).
8. The proximity value is computed for each neighbor and the neighbor with the highest such value is the *nearest neighbour*.
9. The current component is *promoted* or *demoted* to its nearest neighbor's level set according as its intensity value is less or greater than the intensity value of the nearest neighbour.
10. The next component is selected and the process continues from step 9.
11. If the area of a resulting component is larger than *s*, then it is not guaranteed to merge with any other component. Then if the intensity difference between the current component and the nearest component is less than ρ , the current component is *promoted* or *demoted* according as step 11.
12. $\sum_{\alpha \in \Psi} L_\lambda(\alpha)$ yields the segmented image.

In the next section we present the result of application of this algorithm on intensity images.

4. Results

The algorithm is developed and executed in Matlab 5.1 and applied on number of intensity images, three of which are presented in this paper. The algorithm described in [4] is also applied on those images and the segmentation result is compared with the result obtained in our method. The method in [4] needs thresholds values for color quantization, image scale and region merging whereas our method requires definition of only the minimum segment size. Note that for all the results in method [4] we have used default threshold values for all the parameters. The original images along with the segmented outputs using both the methods are shown. The comparison for segmentation accuracy is done with respect to the ground truth image.

The performance of the proposed algorithm is measured and compared in two different ways. The classification method measures the level of correctness of classifying the segments with respect to the ground truth. The steps involves in this measurement are as follows For the segmented image S and the ground truth image T , select each segment s_i in the ground truth image and search for the largest segment k of S contained (partly or wholly) in s_i . The ratio of area (k) and area (s_i) expressed as a percentage is the classification level for the segment s_i .

In the second measure, coefficient of variation (CV) is calculated from for a particular segment in any of the output images from the corresponding pixel values in original image. Thus, CV of a segment provides a measure of internal variation of pixel values within that segment. Hence for the present analysis, a segmentation method is good if it gives consistently lower coefficient of variation than the other method.

Fig. 3(a) is the "blood cells" image containing different blood cells where area filtering is performed for the object size 200. Fig. 3(b) and 3(c) are the segmented images in our method and in method [4] respectively. Due to the size constraint the small objects of the original image are absent in the Fig. 3(b) although those still survive in the Fig. 3(c). For classification accuracy measure, we have considered the small objects as the part of the background region. Table 1 shows the segment wise percentage of correct classification and coefficient of variation for both the output images. The performance of the proposed method as described in Table 1 is better than the method [4].

A "galaxy" image is shown in Fig. 4(a) where 150 is the minimum object size for area filter. Though the segments are not very clear in the original image, number of segments in both the output image is same as the ground truth image. Fig. 4(b) is the segmented output in the proposed method whereas Fig. 4(c) describes the output using method [4]. Table 2 shows the comparative study of correct classification and coefficient of variation of the segments present in both the output images. It is clear from the Table 2 that our method is superior than the method described in [4].

Fig. 5(a) describes an image of a "planet". The minimum object size is taken as 100. The segmented images corresponding to the proposed method and the method [4] are shown in Fig. 5(b) and 5(c) respectively. Fig. 5(c) clearly defines that the method [4] fails to segment the "planet" image whereas the output of the proposed method is very close to the expected result.

The correctness for classifying different segments present in the output images and the coefficient of variations of those segments are presented in Table 3. It is clear from the Table 3 that the performance of our algorithm is much better than method [4]. Table 3 shows that the classification of segment1 is 100% in the output image of method [4], but it is clear from fig. 5(c) that segment1 is over segmented. Also method [4] is unable to detect segment2 and segment4. Note that “Saturn” image contains five segments, which is same as the number of segments present in the output image of the proposed method whereas the output image of the method [4] has only three segments. Hence in all sense our method is superior to the method [4].

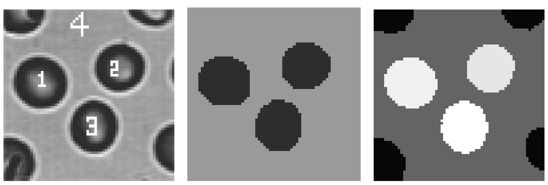


Fig. 3(a) Fig. 3(b) Fig. 3(c)

Fig 3: (a) Original “blood cell” image (b) Segmentation by the proposed method. (c) Segmentation using [4].

Table 1: Percentage of classification and coefficient of variation “blood cell” image.

Segments	Classification (%)		CV(%)	
	By our method	By [4]	By our method	By [4]
Segment1	72.21	70.96	45.29	46.91
Segment2	65.46	66.49	45.94	47.38
Segment3	68.04	69.32	46.98	48.87
Segment4	100.00	96.74	25.61	28.40

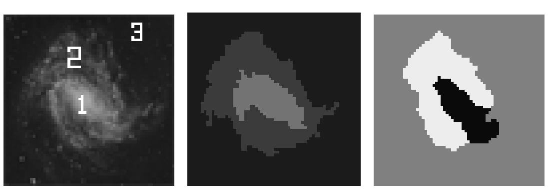


Fig. 4(a) Fig. 4(b) Fig. 4(c)

Fig 4: (a) Original “galaxy” image (b) Segmentation by the proposed method. (c) Segmentation using [4].

Table 2: Percentage of classification and coefficient of variation for “galaxy” image.

Segments	Classification (%)		CV (%)	
	By our method	By [4]	By our method	By [4]
Segment1	90.11	82.41	15.71	16.99
Segment2	71.46	61.59	19.72	23.36
Segment3	94.04	92.79	18.68	22.86

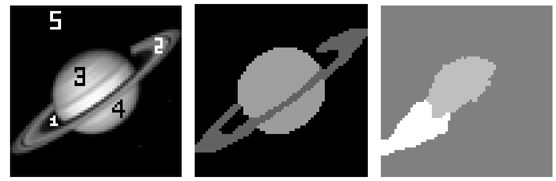


Fig. 5(a) Fig. 5(b) Fig. 5(c)

Fig 5: (a) Original “planet” image (b) Segmentation by the proposed method. (c) Segmentation using [4].

Table 3: Percentage of classification and coefficient of variation error for “Saturn” image.

Segments	Classification (%)		CV (%)	
	By our method	By [4]	By our method	By [4]
Segment1	88.89	100.00	99.36	97.57
Segment2	66.67	61.64	29.35	61.78
Segment3	86.15	62.56	33.31	37.58
Segment4	79.59	63.29	23.45	53.27
Segment5	99.45	98.67	23.52	42.84

5. Conclusion

We have described an efficient method of image segmentation using the inherent property of the image pixels. The size filter has removed insignificant image components. Such definition of size may be useful for identifying objects of certain scale in remote sensing and biomedical applications. We have studied some properties of the image level sets. Currently we are trying to incorporate shape based image features in the proposed proximity measures so that images containing specific shapes may be extracted. These would be important for applications like content based image retrieval.

References

[1] P. Monasse and F. guichard, “Fast Computation of a Contrast-Invariant Image Repesentation,” *IEEE Trans. IP*, vol. 9, no. 5, 2000, 860-872.

[2] S.T. Acton and D.P. Mukherjee, “Scale Space Classification using Area Morphology,” *IEEE Trans. IP*, vol. 9, no. 4, 2000, 623-635.

[3] F. Felzenszwalb and D.P. Huttenlocher, “Image Segmentation using Local Variation,” *CVPR’ 98*, Santa Barbara, CA, June 1998.

[4] Y. Deng, B.S. Manjunath and H. Shin, “Color Image Segmentation,” *CVPR ’99*, vol. 2, Fort Collins, CO, June 1999, pp. 446-51.