Spatiotemporal Approach for Tracking Using Rough Entropy and Frame Subtraction

B. Uma Shankar and Debarati Chakraborty

Machine Intelligence Unit,
Indian Statistical Institute, Kolkata 700 108, India
uma@isical.ac.in, debarati.earth@gmail.com

Abstract. We present here an approach for video image segmentation where spatial segmentation is based on rough sets and granular computing and temporal segmentation is done by consecutive frame subtraction. Then the intersection of the temporal segmentation and spatial segmentation for the same frame is analyzed in RGB feature space. The estimated statistics of the intersecting regions is used for the object reconstruction and tracking.

Keywords: Segmentation, rough entropy, rough sets, video tracking.

1 Introduction

In computer vision, detection and tracking of moving object is very important task. The application of object tracking in video sequences has been studied over the years. In this task there are many types of uncertainties and ambiguities which is making this task a difficult problem. Over the years researchers have been trying to improve the accuracy and speed in detection and tracking [5,11,13]. Granular computing is a young, though rapidly expanding, and important area of research. Granular computing is the process of dealing with the information granules which is collection of some points similar in some respect and dealing with which proves to be effective for human cognition. In case of image and video processing, proper recognition of each part of the image from the available information is a well known problem where partitioning (segmentation or classification) of data set plays a very important role. Understanding of an image depends on efficient partitioning. So, granular computing can be useful for image processing. According to Butenkov [2] the most important problem regarding granulation of an image is different kind of input information. Here we propose to use Rough entropy as proposed by Pal *et al.* [7], incorporating some modification for video image segmentation and object detection for tracking in video image sequences.

In the present article we have proposed an approach for video image segmentation based on granular computing and rough sets. The contributions of the article are as follows : (i) We propose a method for detection of granule adaptively. Here we decompose the image into homogeneous granules using quadtree decomposition, which takes into account the spatially connectedness and gray level similarity, both. (ii) Then a general form of rough entropy function is defined, which can

be computed with the estimated parameter (base of the logarithm) for the function for a given image. This function can be modified according to the choice of the user and the amount of noise present in the image. (iii) After having spatial segmented image frame, we have used a frame subtraction method to collect the temporal information of the frame [13]. Then we have considered the intersection of the spatial and temporal segmented regions in RGB feature space and calculated the mean and maximum deviation of the intersecting region (region/object statistics). Then we reconstruct the object in the current frame by using this statistics within a predefined tracker, thus tracked the required object. (iv) As here rough entropy based segmentation is applied on video sequence, we have the advantage of exploiting the threshold in the previous frame for the searching in the limited range over a window around the threshold in the previous frame, for detection of threshold in the current frame. Thus our proposed method is fast, efficient, accurate and different from the existing methods.

2 Rough Sets for Image Segmentation

In the process of image segmentation normally we have to take care of noise present in the image. Due to which there is always ambiguities in gray level as well as in spatial. These ambiguities lead to uncertainty in image segmentation. To demarcate the objects of interest in the image properly, we segment the image into some homogeneous regions. These regions put together normally provides the object of interest. Theory of Rough sets [8] has recently become a popular mathematical framework for granular computing. The focus of rough set theory is on the ambiguity caused by limited discernibility of objects in the domain of discourse [4,6]. Its key concepts are those of object ‘indiscernibility’ and ‘set approximation’. The set approximation capability of rough sets is exploited in this investigation [7,12] to formulate an entropy measure, called rough entropy, quantifying the uncertainty in an object-background image. This has been done by defining an image as a collection of pixels and the equivalence relation induced partition as pixels lying within each non-overlapping (homogeneous in some respect) windows over the image. With this definition the roughness of various transforms (or partitions) of the image can be computed using image granules. Maximization of the said rough entropy measure minimizes the uncertainty arising from vagueness of the boundary region of the object and ambiguity caused by gray level. Therefore, for a given granulation the threshold for object-background classification can be obtained through its maximization. This has been successfully demonstrated in [3,7,12]. Here we propose a method of selecting the appropriate granules based on the local gray level distribution. The granules are obtained by quadtree decomposition of the image frame [3]. Therefore the granule sizes will not be same. We define the size of the i^{th} granule as: $Size_of_granule_i = m_i \times n_i$. Incorporating these changes in the algorithm for computation of Rough entropy is presented below. The Rough entropy (as defined in [7,12]) has been modified to make it more general and presented in equation (1).

$$RE_T = -\frac{BASE}{2} \left[R_{O_T} \log_{(BASE)} (R_{O_T}) + R_{B_T} \log_{(BASE)} (R_{B_T}) \right] \quad (1)$$

Here, we consider the R_{O_T} as the roughness of the object and R_{B_T} is the roughness of the background with respect to the threshold “T”. Therefore the combined (object and background) entropy computed will be RE_T . In the present experiment we choose the *BASE* as “10” which we found to be suitable for most of the images.

To detect the object, we describe the method of object enhancement/ extraction based on the principle of minimizing the roughness of both object and background regions, i.e., maximizing RE_T . One can compute for every T the RE_T of the image, representing the background and object regions $(0, \dots, T)$ and $(T + 1, \dots, L - 1)$, respectively, and select the one for which RE_T is maximum. In other words, select $T^* = \arg \max_T RE_T$, as the optimum threshold to provide the object background segmentation.

We have taken the range of the data within a granule (or window) as the measure for the quadtree decomposition. This range should be less than a predefined threshold (say, $RT = \text{half of the interquartile range}$), which helps in finding the granules that are homogenous in gray level as well as spatially connected. Incorporating proposed changes in the algorithm for computation of the Rough entropy, assuming that the object is in the upper side of the histogram, is given below [3].

Algorithm for Threshold Selection: Following is the algorithm for efficient implementation of the aforesaid methodology for selecting T^* as defined above. Let *max_gray* and *min_gray* be the maximum and minimum gray level values of the image, respectively. Let the i^{th} granule (*granule_i*) represent a window of $m_i \times n_i$ pixels. Let total number of granules be *total_no_granule* and the size of the i^{th} granule is *Size_of_granule_i* = $m_i \times n_i$. Initialize: Four integer arrays namely *object_lower*, *object_upper*, *background_lower*, *background_upper* each of size (*max_gray* - *min_gray* + 1) to zero.

Step 1: for $i = 1$ to *total_no_granule*

max_granule_i = maximum gray value of pixels in *granule_i*

min_granule_i = minimum gray value of pixels in *granule_i*

(a) for $\text{max_granule}_i \leq j \leq \text{max_gray}$

object_lower(j) = *object_lower*(j) + *Size_of_granule_i*

(b) for $\text{min_granule}_i \leq j \leq \text{max_gray}$

object_upper(j) = *object_upper*(j) + *Size_of_granule_i*

(c) for $\text{min_gray} \leq j \leq \text{min_granule}_i$

background_lower(j) = *background_lower*(j) + *Size_of_granule_i*

(d) for $\text{min_gray} \leq j \leq \text{max_granule}_i$

background_upper(j) = *background_upper*(j) + *Size_of_granule_i*

Step 2: for $l = \text{min_gray}$ to *max_gray*

object_roughness(l) = $1 - \lfloor \text{object_lower}(l) / \text{object_upper}(l) \rfloor$

background_roughness(l) = $1 - \lfloor \text{background_lower}(l) / \text{background_upper}(l) \rfloor$

IF *object_roughness*(l) $\leq \frac{1}{(\text{BASE})}$, then *object_entropy*(l) = 1,

ELSE *object_entropy*(l) = $\lfloor \text{object_roughness}(l) \log_{\text{BASE}} (\text{object_roughness}(l)) \rfloor$,

AND

IF *background_roughness*(l) $\leq \frac{1}{(\text{BASE})}$, then *background_entropy*(l) = 1,

ELSE $background_entropy(l) =$

$$[background_roughness(l) \log_{BASE}(background_roughness(l))]$$

$$Rough_entropy(l) = - [\frac{BASE}{2}] \times [object_entropy(l) + background_entropy(l)].$$

Step 3: $Threshold(optimal) = \arg \max_l [rough_entropy(l)]$. $\diamond$

Remark: Given the max_gray and min_gray values, the computation of rough entropy (and hence the algorithm) requires only a single scan of pixels in the image, since $max_granule_i$ and $min_granule_i$ are computed exactly once for each granule. Therefore the computational complexity of the algorithm is same as that of histogram computation and scanning throughout all the gray level ranged from min_gray to max_gray is required only for the first incoming frame, after that we will scan only through those gray levels which are in the neighborhood of the optimum threshold value in the previous frame.

3 Detection and Tracking in Video Sequence

A video normally consists of a sequence of color images or gray level images. In the present work we have used both. We converted the color image to gray level image for spatial segmentation. Then perform the tracking by acquiring some temporal information and merging it with the spatial one. After that using the color information the object statistics is estimated in the RGB space. This object statistics helped in reconstruction of object in the RGB space, hence the detected and tracked properly and accurately.

The above proposed segmentation methods works with gray level images. Therefore, for implementing it on color images we convert the color image into a gray level image by using the relation $Y = 0.3R + 0.59G + 0.11B$ (see [10]), where R , G & B has its usual meaning and Y is the obtained gray level image. After conversion of color images into gray level the above proposed rough entropy for object segmentation is applied. After that we have performed a temporal segmentation approach to gather more information and reconstruct the object properly. This segmentation is done by estimating the change between two consecutive frames. Here we subtracted the current frame from the previous frame (i.e., if the object is in upper side of the histogram and the reverse one if the object is in the lower side). Then the noise in the difference image is eliminated by estimating a suitable threshold. The pixels greater than the threshold value will be treated as the part of the object in the temporal segmented image. After performing both the segmentations (i.e., spatial and temporal), we have combined the two results to reconstruct the object properly.

Target Localization and Tracking: At first reference model from the reference frame is marked. After having the two (spatial and temporal) segmented images of the current frame, we have taken the intersection of the two segments to get the object region which is common to both. The statistics of this common region is estimated in the RGB feature space. Here, we used the mean(mn) and maximum deviation (max_dev) of this intersecting region in the RGB feature space. Note, here the common region should have at least 5% of the object region, otherwise we can skip and take the next frame as the current frame. After

having these information, we need to place a tracker on the current frame within which the pixels will be scanned. We have considered a rectangular tracker of size 1.25 times in both the direction (i.e., in width and length) of the size of the object in the previous frame, with location of center is the same as that was in the previous frame. That is we assume that after changing the location from the previous frame to the current frame, the whole object should be within the area considered by the tracker. Then we start to scan each pixel within the tracker, which belong to either spatial or temporal segmented region, and if the difference between a pixel and the mean (mn) in RGB feature space (i.e., all the three features) are less than the max_dev then that pixel will be considered as the part of the object otherwise as background.

Algorithm for Tracking: The object detection and tracking starts with selection of the tracker [5,11,13]. So we first decide the tracker. Here we have considered a rectangle tracker, which completely covers the object that will be tracked. Then the following are the steps to detect the object and tracking of the object. We first select the first (or reference frame), then we will segment it into regions and mark the object of interest.

- Step 1:* Convert the input color image (I) to gray level image (Y).
- Step 2:* If necessary, enhance the image for better contrast.
- Step 3:* Apply the Rough entropy based image segmentation. (Note: Here, the advantage of considering a window around the threshold in the previous frame is exploited for segmentation of video images in the current frame.)
- Step 4:* Do a temporal segmentation by subtraction between two consecutive frames and eliminate the noise from subtracted image by thresholding.
- Step 5:* The intersection of the two segmented images are considered in RGB feature space and the mean (mn) and maximum deviation(max_dev) of those points are calculated in the same feature space.
- Step 6:* Then design a rectangular tracker which is 1.25 times (in both directions) of the object size in the reference frame (i.e., in the previous frame) and place the center of the tracker at the location which is the center of the tracker in the previous frame.
- Step 7:* For every pixel (x, y) within the tracker, which belong to either spatial or temporal segmented region, do the following
 $IF |f_R(x, y) - mn| < max_dev_R$ and $|f_G(x, y) - mn| < max_dev_G$ and $|f_B(x, y) - mn| < max_dev_B$ then $F(x, y) = I(x, y)$ *ELSE* $F(x, y) = 0$
(here, F is the detected object image).
- Step 8:* Repeat the Steps 1 to 7, for all the frame in the video sequence for tracking of the moving object.

4 Results and Discussion

Here, we have considered two video sequences for demonstration of the proposed approach. These are *Baggage Detection Sequence* of *i-LIDS dataset* supplied to *AVSS-2007* [1] and *Surveillance Scenario Sequence* data used in *PETS-2000* [9].

Fig. 1. The *Baggage Detection Sequence* from AVSS-2007, the results on frame nos. 1297, 1298, 1313 and 1314 are presented here: (a) Original images, (b) After converting to gray level, (c) Spatial Segmented regions, (d) Temporal Segmented regions, (e) Reconstructed object, and (f) Tracked object in the color images.

Fig. 2. The *Surveillance Scenario Sequence* from PETS-2000, the results on frame nos. 137, 141, 145 and 150 are presented here: (a) Original images, (b) After converting to gray level, (c) After enhancing by histogram equalization, (d) Spatial Segmented regions, (e) Temporal Segmented Regions, (f) Reconstructed Object, and (g) Tracked object in the color images.

The sample images for the two video sequences are shown in Figs. 1(a) and 2(a), respectively. After converting into gray level the two sets of sample images are presented in Figs. 1(b) and 2(b), respectively. The *Surveillance scenario Sequence* images are enhanced (histogram equalized) for better contrast, before segmentation, the sample images are presented in Fig. 2(c). And after spatial segmentation of the two sequences the results are presented in Figs. 1(c) and 2(d), respectively.

The proposed object detection and tracking is implemented and tested on various video sequences. For the demonstration of the results, we considered two sequences and four frames from each sequence and presented with the intermediate results obtained during various steps of processing. This provides the complete picture of the proposed methodology. The results obtained after temporal segmentation is shown in Fig. 1(d) and Fig. 2(e), respectively. The reconstructed object is shown in Fig. 1(e) and Fig. 2(f). These obtained color images along with

the tracker are presented in the Figs. 1(f) and 2(g), respectively, for the two sequences. From the demonstrated results it can be seen that the moving objects are detected and tracked properly. The enhancement has helped in detection of object (car) accurately, in Fig. 2. However in the *Baggage Detection Sequence* the available contrast was sufficient to detect the moving person properly. We compared with another thresholding based method and found to be performing better, but results can not be presented here due to limitation on length of the paper.

5 Conclusions

Here we have proposed rough entropy and frame difference based spatiotemporal segmentation for object detection and tracking in a video sequence. We proposed a method of image granulation using quad-tree decomposition and a modification in the rough entropy function for object and background segmentation and then combined it with the results obtained after frame subtraction based temporal segmentation to reconstruct the moving object in the video sequence. The rough entropy based thresholding in window around the threshold detected in the previous frame makes it efficient. The proposed method has been tested on various video sequences, and the initial experimental results are very satisfying. The proposed method uses spatial as well as temporal information and gives an accurate and efficient algorithm.

References

1. AVSS-2007: Fourth IEEE Int. Conf. Adv. Video & Signal Based Surveillance (2007)
2. Butenkov, S.A.: Granular computing in image processing and understanding. In: Proc. IASTED Int. Conf. Artificial Intelligence and Applns, pp. 811–816 (2004)
3. Chakraborty, D., Shankar, B.U.: Rough entropy based object segmenatation and tracking in video images. Tech. Rep. MIU/TR/-02/10, MIU, ISI (2010)
4. Hassanien, A.E., et al.: Rough sets and near sets in medical imaging: A review. IEEE Trans. on Information Technology in Biomedicine 13(6), 955–968 (2008)
5. Maggio, E., Cavallaro, A.: Video Tracking - Theory and Practice. Wiley, Chichester (2010)
6. Pal, S.K., Peters, J.F. (eds.): Rough Fuzzy Image Analysis: Foundations and Methodologies. Chapman & Hall/CRC (2010)
7. Pal, S.K., Uma Shankar, B., Mitra, P.: Granular computing, rough entropy and object extraction. Pattern Recognition Letters 26(16), 2509–2517 (2005)
8. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Norwell (1992)
9. PETS-2000: IEEE Int. WS Perfor. Evaluation of Tracking and Surveillance (2000)
10. Pratt, W.K.: Digital Image Processing. John Wiley & Sons, New York (1991)
11. Tekalp, A.M.: Digital Video Processing. Prentice Hall, New Jersey (1995)
12. Shankar, B.U.: Novel classification and segmentation techniques with application to remotely sensed images. In: Peters, J.F., Skowron, A., Marek, V.W., Orłowska, E., Słowiński, R., Ziarko, W.P. (eds.) Transactions on Rough Sets VII. LNCS, vol. 4400, pp. 295–380. Springer, Heidelberg (2007)
13. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Computing Surveys 38(4), 1264–1291 (2006)