Granular Flow Graph, Adaptive Rule Generation and Tracking

Sankar Kumar Pal, *Life Fellow, IEEE*, and Debarati Bhunia Chakraborty

***Abstract*—A new method of adaptive rule generation in granular computing framework is described based on rough rule base and granular flow graph, and applied for video tracking. In the process, several new concepts and operations are introduced, and methodologies formulated with superior performance. The flow graph enables in defining an intelligent technique for rule base adaptation where its characteristics in mapping the relevance of attributes and rules in decision-making system are exploited. Two new features, namely, expected flow graph and mutual dependency between flow graphs are defined to make the flow graph applicable in the tasks of both training and validation. All these techniques are performed in neighborhood granular level. A way of forming spatio-temporal 3-D granules of arbitrary shape and size is introduced. The rough flow graph-based adaptive granular rule-based system, thus produced for unsupervised video tracking, is capable of handling the uncertainties and incompleteness in frames, able to overcome the incompleteness in information that arises without initial manual interactions and in providing superior performance and gaining in computation time. The cases of partial overlapping and detecting the unpredictable changes are handled efficiently. It is shown that the neighborhood granulation provides a balanced tradeoff between speed and accuracy as compared to pixel level computation. The quantitative indices used for evaluating the performance of tracking do not require any information on ground truth as in the other methods. Superiority of the algorithm to nonadaptive and other recent ones is demonstrated extensively.**

***Index Terms*—Adaptive rule-base, flow graph, granular computing, Internet of Things (IoT), neighborhood rough sets (NRSs), soft computing, unsupervised video tracking.**

I. Introduction

THEORY of rough sets, as explained by Pawlak [1] dealt with uncertainties or incompleteness of knowledge arising from the limited discernibility of objects in the domain of discourse. Its key concepts are those of object "indiscernibility" and "set approximation." These characteristics made the theory useful in several areas of pattern recognition and machine learning, e.g., feature reduction and selection [2], [3], image processing [4]–[6], data mining, and knowledge discovery [3], [7]. In image processing rough set is recently used to deal with the uncertainties arising from grayness and spatial ambiguities the concept of set approximation [4]–[6], [8]. Although rough set has been used in image processing its merit in video processing has not yet been explored.

Rough sets is widely used in granular computing. Granulation is a basic step of human cognition system and hence a part of natural computing [9], [10]. This concept was first introduced by Zadeh [11]. Granules may be of different types, e.g., crisp granules, fuzzy granules, rough-fuzzy granules, and neighborhood granules. The predefined parameters in all the existing granulation approaches are those related to the sizes or shapes of the granules. But, natural granulation is arbitrary and it does not have any fixed shape or size. In this paper, an attempt of forming such granules over videos has been made. The similarities in both color and temporal feature spaces are considered during this formulation which are applied for faster video processing.

Tracking of moving objects from video sequences is considered to be an important task in computer vision. This has several applications, such as in surveillance, gesture recognition, event detection, and Internet of Things. Several approaches for tracking have been reported under various modes, e.g., supervised, partially supervised (with initial manual interactions) and unsupervised (without manual interactions). Most of the approaches are partially supervised [12]–[20]. The unsupervised methods are mostly based on background estimation [21], [22]. A few unsupervised approaches for tracking are recently proposed with statistical modeling of object motion [20], [23], [24]. Further, video sequences have characteristics like, changes in shapes/sizes of moving object(s), change in motion of the object(s), change in the number of object(s), and occurance of overlapping. Since the available information is not always complete, this makes the tasks of prediction and estimation difficult while tracking. Problems related to these characteristics have been recently addressed and several partially supervised approaches are proposed using statistical modeling, local features extraction, etc. [13], [14], [16], [17]. But, the initial object(s) need to be labeled manually in all those approaches. The method developed in this paper for detecting continuously moving objects in static background is unsupervised and does not need manual labeling. This method also does not consider background approximation like the other unsupervised techniques. The merits of rough set theories are exploited here for object modeling and tracking. The initial object models are roughly estimated with their lower and upper approximations. Then the rough rule base is effectively used to deal with

Manuscript received March 11, 2016; revised May 31, 2016 and July 20, 2016; accepted August 10, 2016. Date of publication August 26, 2016; date of current version November 15, 2017. This paper was recommended by Associate Editor Y. Yang.

The authors are with the Center for Soft Computing Research, Indian Statistical Institute, Kolkata 700108, India (e-mail: sankar@isical.ac.in; debarati.earth@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2016.2600271

the incompleteness and uncertainties in the information system and to have gain in computation time.

Video tracking is a decision-making process over time. Incorporation of rough rule-base to this application will only be effective with proper updation procedure. The updation of all of the attributes in every frame is time consuming and reduces the merit of using rule-base. The concept of flow graph proves to be useful for an intelligent adaptive rule-base generation. Flow graph was introduced by Pawlak [25] which is a directed acyclic graph used to map information flow. This concept is used in video processing to accumulate information from multiple cameras [26] for behavior modeling, but not for tracking. The main advantage of this graph over rule-base is that it can map the information flow between the attributes and hence can show the significance of each individual attribute in an information system. The adaptation can be performed according to the changing significance of the attributes. One way of using flow graph for training and testing is shown in this paper. Two new features (expected flow graph and mutual dependency between flow graphs) are introduced in this process. The entire decision-making is performed on granular level, therefore this can be called a granular rough flow graph.

Different types of sensors have been used to track moving objects in static background. Some such examples are, multiple cameras [27], PTZ camera [28], and Kinect sensor [29], [30]. In this paper, the data obtained from RGB and depth (D) sensors are used. That is, this algorithm works with RGB-D information or information sensed from both RGB and D sensors together. The processing of RGB and D information of a particular frame are performed separately and the decision making is done by considering both the outputs. It is obvious that the depth of moving object(s) in a sequence always has discrete (higher or lower) values than that of its background if it is not occluded. The datasets that are used in this paper are sensed either by infra red (IR) sensor [31] (different types of surveillance) or by Kinect [32] sensor (different types of hand movements).

The basic concept of rough rule-base in tracking was formulated in our earlier work [33], but that was a partially supervised approach and no adaptation technique was there. Whereas the unsupervised methodology proposed in this paper is adaptive. It can be broadly divided into three parts: 1) training; 2) testing; and 3) updation. Initial P number of frames are given as the input for training, i.e., for the object/background labeling and formation of rough rule-base. The current and its previous frames are the input in testing phase. The 3-D temporal and spatio-color granules are formed over these frames and the object-background separation in the current frame is performed with the rule-base. The rule-base gets updated with the help of flow graph afterwards. The study made in this chapter has the following novelties. It is shown how: 1) spatio-temporal information can be used to form 3-D neighborhood granules in videos with natural partitioning; 2) neighborhood granular rough rule-base can be formed on 3-D granulated space to reduce the computational complexity; 3) granular rough flow graph can be defined for rule-base adaptation; 4) flow graph can be used for the tasks of testing and validation of the rule-base by introducing two new features, namely, expected flow graph and mutual dependency; and 5) the indices for evaluating the performance of tracking (not requiring ground truth information) can be formulated with IR sensed data and 3-D granules. All these features characterize the task of unsupervised tracking of moving object(s) with static background, and are proved to be effective in the scenario with no initial manual labeling.

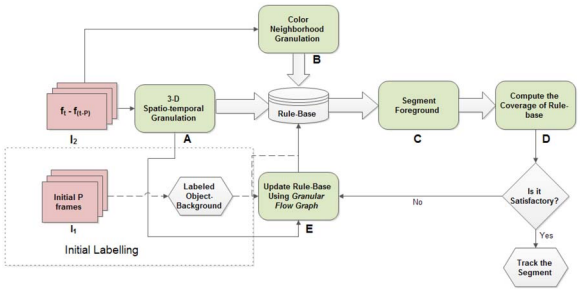

Fig. 1. Block diagram of the tracking method.

A brief discussion on neighborhood rough sets (NRSs) [34], [35] which is a new variant of Pawlak's rough set [1], and its relevance to video tracking were described in our earlier work [33]. The proposed method for unsupervised tracking and the concept of 3-D neighborhood granules are described in Section II. The method of rule-base adaptation using flow graphs is explained in Section III. This includes definitions of the new features, namely, expected flow graph and mutual dependency between flow graphs. A brief introduction to flow graph and its relevance in tracking is also provided. Indices for tracking evaluation are formulated in Section IV. Section V describes the effectiveness of all those features experimentally along with suitable comparisons. Section VI concludes this paper.

II. Neighborhood Granular Rough Rule-Base for Unsupervised Tracking

Block diagram of the proposed methodology for video tracking is shown in Fig. 1. The initial P frames are given to the rule-base for initial labeling, marked as I_1 in the figure. The current frame (f_t) and its previous P frames (f_{t-P}) in a video, marked as I_2 are the input for its processing. The spatio-color and spatio-temporal granules are formed over these frames, shown in blocks "A" and "B," respectively. The decision making regarding the object-background separation in f_t is performed according to the rule-base. The foreground segmentation is then done (block "C") depending on the output of the rule-base. The coverage of the rule base is checked (block "D") afterwards over the segmented output. If the coverage is satisfactory, then the foreground segment will be tracked, otherwise the rule-base will get updated with the granular flow graph (block "E").

The details of the tasks in blocks A and B, I_1, rule-base, and blocks C and D are described in this section. The block E is described in Section III.

A. Formation of Neighborhood Granules in Videos

As discussed earlier granular computing is effective to reduce the computational time of a system. But it may also cause loss of information. Hence, meaningful granulation is very important for granular computing. Neighborhood granules incorporates neighbor information during granule formation. We are looking for a meaningful granulation in images and videos. A way of forming spatio-color natural granulation over a still image was introduced in our earlier work [33]. Here, two new types of granules are defined to deal with video sequences. They incorporate spatio-temporal nearness and color nearness. The way of forming 3-D spatio-temporal neighborhood granules over videos is described here. This incorporates the temporal nearness which is expected to contain the information on the trajectory of moving object(s) as the third dimension of the granules. The color nearness of these 3-D granules is then considered and color neighborhood granules are formed accordingly.

1) 3-D Neighborhood Granules: Temporal domain, which is the third dimension of a video stream, makes it different from still images. The processing of information is one of the most important tasks for decision making in videos. Here, a new approach of extracting this information is introduced where the highest importance is given to the current frame among the previous P number of frames under consideration instead of giving equal importance to all of them like the other approaches. All the changed information are computed with respect to the current frame. If the current frame (f_t) is of size $M \times N$ and its previous P frames ($f_{t-p} : p = 1 \cdots P$) are considered, then the changed information between f_t and F_{t-p} (denoted as τ_p) is computed according to (1). The matrix (τ_p) is of size $M \times N$ and there are P such matrices

$$\tau_p = \left| f_t - f_{t-p} \right| \quad \forall p \in P. \tag{1}$$

The median of $\tau_p : p = 1, \ldots, P$ (τ_{med}) is a matrix of size $M \times N$ and is computed as

$$\tau_{\text{med}} = \text{Median}(\tau_1, \ldots, \tau_P). \tag{2}$$

The spatio-temporal granules are formed considering values of the points as τ over the spatial as well as the temporal domains, resulting in 3-D granules. Let x_i be the position of a pixel in the current (tth) frame, then the 3-D granules around it are formed according to

$$\aleph_{\text{sp-tmp}}(x_i) = \bigcup x_j \in U \tag{3}$$

where x_i and x_j are binary connected over $|\tau(x_j) - \tau(x_i)| < Thr_t$ and $x_j \in f_p : p = t, \ldots, t - P$.

The physical interpretation of these 3-D granules is that they contain all the similar changed regions with respect to the current frame in a single granule. In this way a single granule is expected to contain a single moving object region along with its trajectory.

The value of P depends on the speed of the moving object(s) from frame to frame. Further, the spatio-color and spatio-temporal granules formed in our earlier work [33] were 2-D granules, where the nearness was considered in 2-D spatial space. Whereas, here the granules are 3-D which are formed considering spatial as well as temporal space.

2) Color Neighborhood Granules: These granules are formed by incorporating the color nearness in a color histogram of a video frame. In a video sequence the moving object models do not usually have the same color values throughout the sequence, rather they deviate. This phenomenon is considered during the granule formation. Besides, the computation with these granules instead of individual color levels is expected to be much faster. These granules are formed with the RGB-D values of the points present in each $\aleph_{\text{sp-tmp}}$. Let c_i be the color value of a point in $\aleph_{\text{sp-tmp}}$. Then the granule around it is defined as

$$\aleph_{clr}(c_i) = \left\{ c_j \in \aleph_{\text{sp-tmp}} : \triangle \left(c_i, c_j \right) \leq Thr_c \right\}. \tag{4}$$

The method of formation of color and spatio-temporal granules, thus explained in this section, is now applied in videos for the rule base generation and adaptation. This adaptive rule-base, formed over the granulated space, is used for tracking input frames.

B. Rough Rule-Base and Unsupervised Tracking

A method of partially supervised tracking based on generating a rule base was described in our earlier work [33]. Here, an approach for unsupervised tracking is introduced using the same rules. This unsupervised method involves some observations for acquiring features (see Section II-C) and then making some decisions using them. The conditional attributes of the decision-making system are the 3-D spatial-temporal and color granules. The main differences of this rule-base designed for unsupervised tracking from that of the partial supervised one [33] are as follows.

1) No initially labeled data is provided, the labeling is done based on the observations.
2) No separate temporal and spatial features are given to the rule-base as the conditional attributes; instead the 3-D spatio-temporal granules are the inputs.
3) 2-D temporal granules formed with the values τ_{med} (2) considering the spatial nearness constitute the decision space, unlike the spatio-color granules in [33].

The descriptions of object and background characterizing their different states and the corresponding new rule-base are the same as in [33]. The descriptions are as follows.

1) An ideal background granule.
2) A background granule with RGB-D features same as that of an object.
3) An object granule that moves slower than estimated.
4) A background granule containing similar color as that of the object in the previous frame.
5) An object granule with changing depth (movement toward or away from camera) along with its movement.
6) An object granule moving only within the object area (object to object granule).
7) A background granule moving within the region of interest with changes in RGB values (object to background granule).
8) An object granule that starts moving from the current frame onward/ newly appeared object granule.
9) A noisy background granule.

TABLE I
RULE-BASE WITH 3-D NEIGHBORHOOD GRANULES

U	$\aleph_{sp-tmp}$	$\aleph_{RGB}$	$\aleph_D$	$Decision$
$\aleph_1$	NB	NB	NB	B
$\aleph_2$	NB	Be	Be	B
$\aleph_3$	PB	Be	Be	O
$\aleph_4$	PB	Be	Be	B
$\aleph_5$	Be	Be	PB	O
$\aleph_6$	CC	Be	Be	O
$\aleph_7$	NB	NB	Be	B
$\aleph_8$	Be	NB	NB	B
$\aleph_9$	Be	NB	NB	O
$\aleph_{10}$	PB	PB	CC	O
$\aleph_{11}$	Be	Be	Be	O
$\aleph_{12}$	Be	Be	Be	O

10) An object granule that moves in similar RGB-D background.
11) An object granule which moves faster than estimated.
12) An ideal object granule.

The corresponding new rule-base thus generated by the proposed method incorporating 3-D spatio-temporal granules and color granules is shown in Table I. Here, there are four attributes for each features: 1) belongs (Be); 2) does not belong (NB); 3) partially belongs (PB); and 4) contained in (CC). Rules 1 and 12 represent the ideal cases, and the remaining ambiguous.

One may note that the rules that are not relevant with respect to the problem of object tracking have been eliminated during the formation of this rule-base. These aforesaid 12 rules characterize the situations that occur in a sequence often. It is proved that the set of these rules gives more than 90% coverage. In this way, 12 most relevant rules are considered out of $4^3 = 64$ probable combinations. For example, the rule [CC NB NB] does not have any significance in this scenario, as no granule contained by the spatio-temporal region can have totally different color granules than that of its estimated object region.

It is expected that if the number of conditional features gets reduced, the computational complexity will also get reduced simultaneously in this unsupervised task. The task of unsupervised tracking will be faster in this way.

The rule-base (Table I) is formed using the initial object/background model and the observations over P number of initial frames. The method of acquiring the conditional features based on the said observations is described in the next section.

C. Acquiring Features for Initial Unsupervised Labeling

In case of online processing the object-background can not always be labeled manually. Here, one way of object-background estimation is defined. This is different from the conventional unsupervised approaches based on background modeling. Since the computation for background estimation and accordingly the updation processes need to be performed all over the frames, this could be more time-consuming. In our method, we have adopted object estimation, rather than background estimation. Here, all the continuous moving elements in a sequence are treated as the object(s) and the rest part as the background. The common moving regions of τ_p over P frames may be considered to estimate the lower approximation of the objects ($\underline{O}$), that is

$$\underline{O} = \bigcap \{\tau_p \quad \forall p \in P\}. \tag{5}$$

The upper approximation of the object(s) ($\overline{O}$) may similarly be modeled over the union of the changed regions of $\underline{O}$ and τ_P (difference between tth and $(t-P)$th frames). That is

$$\overline{O} = \bigcup_w \{\underline{O}, \tau_P\}. \tag{6}$$

The values of the features that are CC the set $\underline{O}$ are treated as the core values of the object model, whereas those in the set $\{\overline{O} - \underline{O}\}$, i.e., the boundary region, determine the extent to which the values in the object model are allowed.

The values of the attributes in the rule base that are acquired from the first P frames are as follows.

1) Spatio-temporal features in RGB-D space: values of frame difference according to (2).
2) Color features in RGB-D space: RGB-D values present in $\overline{O}$.

Therefore, the conditional features in granular level are: 1) 3-D spatio-temporal granules ($\aleph_{\text{sp-tmp}}$) according to (3) over the region $\overline{O}$ and 2) color/ RGB-D granules ($\aleph_{\text{rgb}}$) and ($\aleph_D$) formed over the color values of $\overline{O}$.

It can be observed that the incorporation of neighborhood granules into the rule-base makes it more robust. However, the information CC a rule-base needs to be updated with time. The requirement of updation is checked by computing the coverage of the rule base. The coverage [36] of a rule-base is the measure of how many granules in the dataset get classified with respect to the total number of granules. As the rule-base in Table I can not overcome the inconsistency between rules 8 and 9, the granules following these two rules remain unclassified. If the coverage shows that there are several unclassified granules in the rule-base, then the updation of the rule-base is carried out. For example, let there be a moving object appear newly in a video scene. For example, let a moving object appear newly in a video scene. This phenomenon will be characterized by rule 8 of Table I. Therefore, the granules present in that object region will remain unclassified resulting in a lower coverage value of the rule-base. The updation process will then be fired.

III. ADAPTIVE RULE-BASE: GRANULAR FLOW GRAPH APPROACH

A. Flow Graph

Information flow graph was introduced [25] to model the information flow in a system. A flow graph is a directed, acyclic and finite graph represented as $G = (N, \mathcal{B}, \varphi)$. N represents the total number (no.) of nodes, $\mathcal{B} \subseteq N \times N$ represents a set of directed branches, $\varphi : \mathcal{B} \to R^+$ is a flow function and R^+ is the set of non-negative reals. Each node of a flow graph represents an attribute of the information system. The input and output represented as n_i and n_o of a node n are the sets: $n_i = m \in N : (m, n) \in \mathcal{B}$ and $n_o = m \in N : (n, m) \in \mathcal{B}$.

The inflow and outflow of that node n are computed as

$$\varphi_+(n) = \sum\nolimits_{m \in n_i} \varphi(m, n) \tag{7a}$$

$$\varphi_-(n) = \sum\nolimits_{m \in n_o} \varphi(n, m) \tag{7b}$$

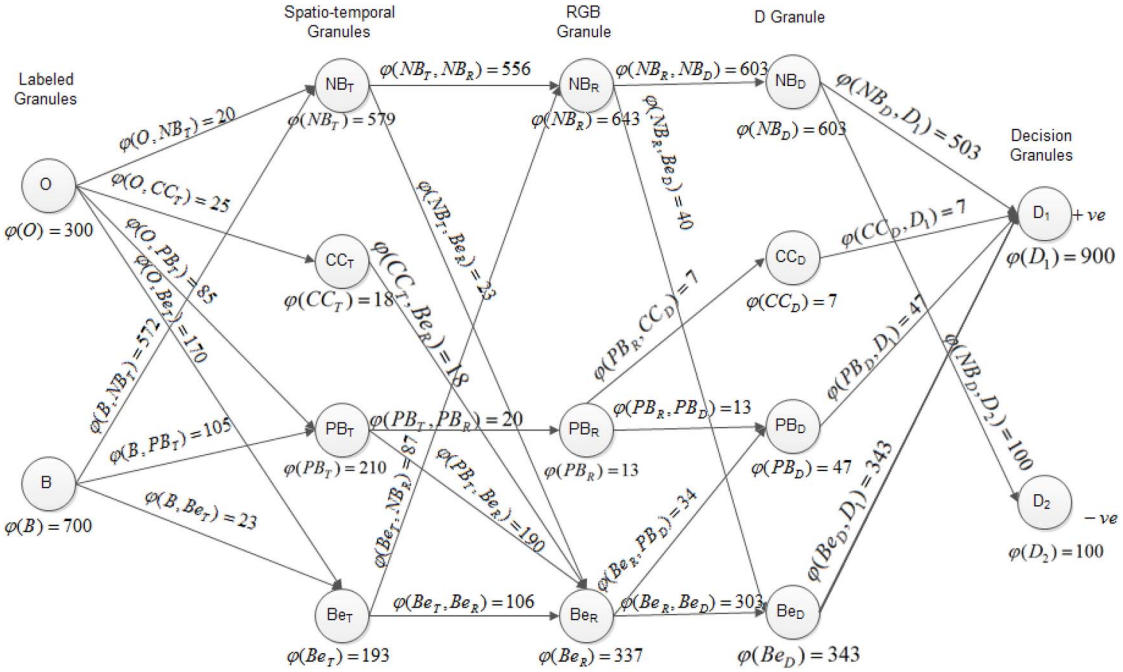

Fig. 2. Flow graph model according to Table I.

where $\varphi(m,n)$ is the flow function characterizing the flow from nodes m to n. The inflow and outflow of an internal node of a graph are supposed to be the same, and so of the graph G. Let $\varphi(G)$ represent the through-flow of the graph G.

The normalized flow graph is represented as $G=(N,\mathcal{B},\sigma)$ where N and $\mathcal{B}$ are the same as earlier, and

$$\sigma(m,n)=\frac{\varphi(m,n)}{\varphi(G)}. \tag{8}$$

The certainty (cer) and coverage (cov) are two major factors associated with every branch of a flow graph. These are defined for the branch (m,n) as

$$\text{cer}(m,n)=\frac{\sigma(m,n)}{\sigma(m)} \tag{9a}$$

$$\text{cov}(m,n)=\frac{\sigma(m,n)}{\sigma(n)} \tag{9b}$$

where $\sigma(m)$ and $\sigma(n)$ are the normalized weights of the nodes m and n, respectively, and $\sigma(m,n)$ is the normalized weight of the branch (m,n). One may note that certainty and covariance are only the two features explored by Pawlak [25] while defining flow graph.

B. Flow Graph in Video: Relevance and Issues

There are several scenarios in a video sequence when the initially defined rule-base can not give much effective results, rather gives many false positive (FP) or false negative (FN) outputs. That is, the importance of a certain rule in a system may increase or decrease with time. For example, new moving object(s) may appear into a sequence, or moving object(s) may get stopped or disappeared from the sequence. The new appearance of object(s) needs new sets of conditional attributes, whereas, disappearance of object(s) needs deletion of the respective sets of the conditional attributes. These can be done with proper updation of the rule-base. Flow graph shows the relationship between the attributes (nodes) and the relevance of the rules (branches) generated from those attributes. Therefore, flow graph has the ability to detect the deviation in the value of a particular branch. The updation of only the attributes associated with the branch can be an effective technique for rule-base adaptation. In this way the sudden or unpredictable changes in a videos can be detected and the corresponding set of rules can be updated. It may be noted that the updation is mostly required when there are multiple moving objects in a sequence.

The updation of the conditional features in every frame due to appearance/disappearance of object(s), and the change in shape, alignment and color of the object(s) is quite time consuming and hence less convenient. Therefore, an intelligent rule-adaptation process is defined here. The underlying idea behind this process is that the attributes should get updated only when required, instead of every frame. Here, all of the conditional attributes do not get updated simultaneously, rather it is done according to the requirement. This is decided with the help of the flow graph.

In the proposed approach the relevance of a certain rule or attribute in an information system is mapped with the help of flow graph. When some deviation in the values of any branch/node is detected, the updation of only the attributes associated with those branches is expected to give the desired results. The flow graph that is modeled here is in granular level and the information flow is designed with the neighborhood granules. Therefore, it is named as "granular flow graph."

In this process, a training flow graph is to be designed first based on the labeled information. The foreground regions of

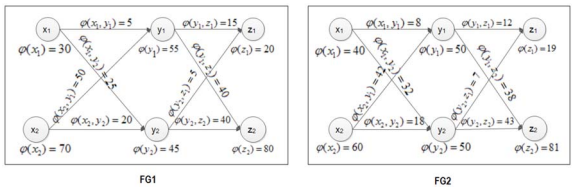

Fig. 3. Training flow graph (FG1) and test flow graph (FG2).

the training frame are the input to the training flow graph. The initial values of its branches and nodes are set to those, as obtained from the rule-base in Table I.

An example of training flow graph according to training data set of 2*b* sequence (described in Section V-A) is shown in Fig. 2. Initially there are 300 object granules and 700 background granules represented as $\varphi(O) = 300$ and $\varphi(B) = 700$, respectively. The branch $\varphi(O, \mathrm{NB}_T) = 20$ indicates that only 20 granules out of those 300 object granules do not belong to the set of the temporal values (NB_T). The values assigned to the rest of the branches represent the similar characteristics. The output shows the correctness in the classification. For example, $D1_{+\mathrm{ve}} = 910$ means that 910 granules out of the 1000 input granules are correctly classified. The performance of the rule-base is measured over each input frame with the help of training flow graph.

The test flow graph is generated afterwards over each input frame (current frame). It is compared with the training flow graph with respect to node and branch parameters. The test flow graph is designed with the assumption that all of the spatio-temporal neighborhood granules with higher gray level values (i.e., expected moving object) are the foreground granules. Hence, these granules are given as the input to the test flow graph. The entire test flow graph, corresponding to an input frame, does not get formed at a time, rather the values are assigned to every node and branch after checking their deviations from the corresponding training values. If the deviation in a certain node or branch is more than the expected, the associated sets of attributes get updated. The updation is done based on the spatio-temporal information in the current frame and this is applied from the next frame onward.

In case of videos, the number of foreground-background granules can not remain the same over the sequence, rather these are expected to change from frame to frame. The comparison between two flow graphs (training and testing) for updation of the latter is fair as long as the input data set distribution among the classes for those graphs is of same ratio. But, the ratio can not remain the same if the distribution of data sets among the classes gets changed. In Fig. 3, such a case is shown with two simple example flow graphs. There are two input classes with data distributions 3 : 7 and 4 : 6. Let FG1 be the training graph and FG2 be the test graph. The overall output accuracies for both are almost the same. But, if the normalized weights in every branch and node of the two graphs are compared then a large deviation can be found. It will lead to false decision. Therefore, comparison of these two flow graphs is unfair for updation of the test flow graph FG2 and updation of the rule-base for these cases is not justifiable.

The concepts of expected flow graph and mutual dependency between flow graphs are introduced here to deal with the aforesaid situation and to make the flow graph applicable to verification tasks. These are described in the following sections.

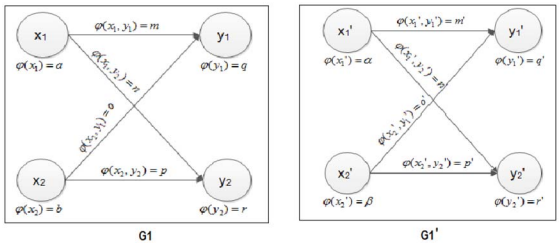

Fig. 4. $G1$ and $G1'$: flow graphs with different data distributions.

C. Expected Flow Graph

Expected flow graph characterizes how the training flow graph is expected to be modeled if the data distributions among the classes get changed. Two simple flow graphs are shown in Fig. 4 for elaboration.

The graph $G1$ is the normalized training graph with two input classes with data distribution of $a : b$. $G1'$ is the expected model of $G1$ if the data distribution gets changed to $\alpha : \beta$. The respective values of the branch parameters, viz, m, n, o, p get changed to m', n', o', p' as

$$m' = \frac{m\alpha}{a} = \frac{m}{a} \times \alpha = \mathrm{cer}(x_1, y_1) \times \alpha \tag{10a}$$

$$n' = \frac{n\alpha}{a} = \frac{n}{a} \times \alpha = \mathrm{cer}(x_1, y_2) \times \alpha \tag{10b}$$

$$o' = \frac{o\beta}{b} = \frac{o}{b} \times \beta = \mathrm{cer}(x_2, y_1) \times \beta \tag{10c}$$

$$p' = \frac{p\beta}{b} = \frac{p}{b} \times \beta = \mathrm{cer}(x_2, y_2) \times \beta \tag{10d}$$

cer(.) is defined as in (9).

The expected values of the nodes in G_1' are

$$q' = m' + o' = \mathrm{cer}(x_1, y_1) \times \alpha + \mathrm{cer}(x_2, y_1) \times \beta \tag{11a}$$

$$r' = n' + p' = \mathrm{cer}(x_1, y_2) \times \alpha + \mathrm{cer}(x_2, y_2) \times \beta. \tag{11b}$$

For example, the expected normalized flow graph of FG1 in Fig. 3 with data distribution, changed to 3 : 7 from 4 : 6 (which is the case of FG2), is shown in Fig. 5. Let it be named as FG1′. Here, the values of the nodes and branches are computed as per (11) and (10), respectively.

D. Mutual Dependency Between Flow Graphs

The mutual dependency between two flow graphs represents how much reliable the decision of a graph would be if it was taken by the another one. That is, if a decision is reliable with $Graph_1$ how much its reliability could be with $Graph_2$. Here, mutual dependency between two flow graphs means mutual dependency between the respective nodes and branches of these graphs. This can be computed by dividing the graph parameter values of the two normalized flow graphs with similar nodes and branches, but the weights can be different.

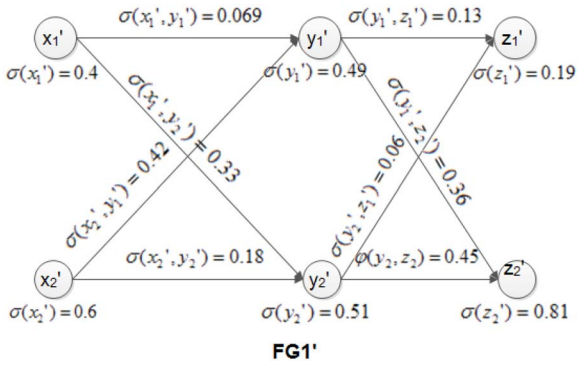

Fig. 5. FG1': expected flow graph of FG1 in Fig. 3 with different distributions.

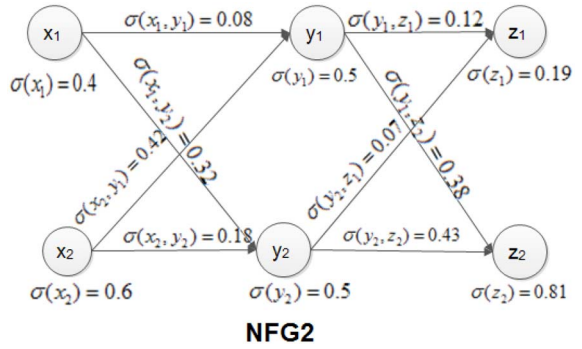

Fig. 6. NFG2: normalized flow graph of FG2.

Let $\text{NFG}_1 = (N_1, \mathcal{B}_1, \sigma_1)$ and $\text{NFG}_2 = (N_2, \mathcal{B}_2, \sigma_2)$ be two normalized flow graphs. The mutual dependency between them can be calculated if and only if:

1) $N_1 \equiv N_2$;
2) $\mathcal{B}_1 \equiv \mathcal{B}_2$;
3) $\sigma_1 \neq \sigma_2$;
4) $\sigma(N_{1i_1}) : \sigma(N_{1i_2}) : \ldots : \sigma(N_{1i_C}) = \sigma(N_{2i_1}) : \sigma(N_{2i_2}) : \ldots : \sigma(N_{2i_C})$, where N_{1i} and N_{2i} are the input nodes of NFG_1 and NFG_2, respectively, and C is the total number of input classes.

Mutual dependency between two of them is defined as

$$\mu \equiv \frac{\text{NFG}_1}{\text{NFG}_2}. \tag{12}$$

The mutual dependency between the respective nodes and branches of the two flow graphs under consideration is computed by diving the graph parameter values with (12).

If the two graphs have same weights, all the nodes and branches of their mutual dependency graph will have the value unity. That is, same decision-making can be performed from both of the graphs. The more deviation from one (unity) reflects less reliability of the respective node(s) or branch(s). The normalized flow graph of FG2, denoted as NFG2, is shown in Fig. 6.

It can be noticed that the two graphs NFG2 and FG1′ (Fig. 5) satisfy the aforesaid conditions for computing the mutual dependency. The computed mutual dependency between them is shown in the graph in Fig. 7. It can be observed that the values of the branches (x_1, y_1), (x_1, y_2), (y_2, z_1), and (y_2, z_2) deviate more from the ideal value of

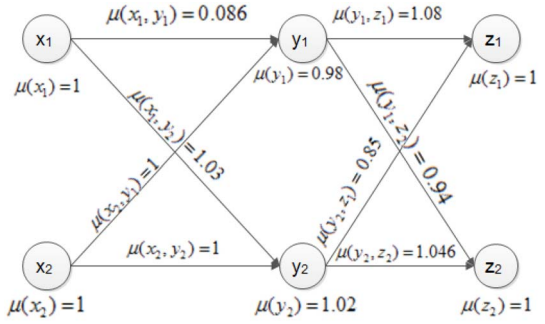

Fig. 7. Mutual dependency graph between FG1' and NFG2.

one (1). Hence, the nodes (y_1 and y_2) representing the conditional attributes and the corresponding rules which establish the above connections need updation. The threshold which detects when the updation is required based on the aforesaid derivation is application dependent.

These two concepts, expected flow graph and mutual dependency between flow graphs, are used in the decision-making process of the proposed algorithm for tracking. The expected flow graph and the training flow graph are designed according to the data distribution in the current frame. The mutual dependency between the normalized flow graph of current (input) frame and expected training flow graph is then measured. The values where the deviations are either higher or lower than unity, the respective set of attributes needs to get updated.

IV. Tracking Indices

The video data that we used here were acquired by color (RGB) as well as depth (D) sensors. The depth values were acquired with Kinect sensors. The moving objects in a sequence are expected to have distinct values in this D-feature space as those move in front of their corresponding background, i.e., nearer to the sensor, if not get occluded. The IR sensor works based on the reflected light with IR wavelength. So, objects closer to the sensor produce brighter regions. But, in case of Kinect sensors the objects nearer to the sensors have less depth values resulting in darker regions. These characteristics are used here while formulating the indices for evaluating the quality of tracking. Two indices are defined by using the merits of D-values. They are independent of ground truth or the trajectory of objects, unlike those in [37] and [38]. Moreover, these indices involve both Kinect and IR sensed data and the proposed 3-D neighborhood granules (Section II-A1), unlike the earlier work [33] using only Kinect data and 1-D spatio-color granules. The video sequences, either acquired by Kinect sensors where the depth value is obtained or by IR sensors where the reflected IR wave value is obtained, and the foreground segmented sequence (FS) are the inputs that are given for the evaluation of FS.

A. E_D Index: Foreground Segmentation

This index incorporates the edge information. In case of IR sensor the moving object region is brighter resulting in higher gray level which implies that if a neighborhood of an edge pixel has brighter or same gray level, then the pixel will

belong to the object; otherwise background. The situation will be the reverse in case of Kinect sensor.

The false positive (FP_{ep}) and false negative (FN_{ep}) values for an edge pixel E_{ep} with an window of $[x_i]$, $i \in w \times w$ around it are computed as

$$FP_{ep} = \left|\left\{ep_i : i \in w \times w \text{ and } ep_i < E_{ep}, \text{ if } ep_i \in FS\right\}\right| \tag{13a}$$

$$FN_{ep} = \left|\left\{ep_i : i \in w \times w \text{ and } ep_i \geq E_{ep} \text{ if } ep_i \notin FS\right\}\right| \tag{13b}$$

where $|.|$ represents the cardinality of the set. The E_D index incorporating the values obtained from (13) is defined as

$$E_D = \frac{\sum_{ep \in EP} \frac{FN_{ep}+FP_{ep}}{w \times w}}{ep}. \tag{14}$$

Kinect sensor maps the depth in the reverse way, i.e., the edge-neighborhood pixels within moving segments have same or lower values than that of E_{ep}. In this way, (13) defining FP and FN gets changed with the conditions $ep_i > Eep_i$ and $ep_i \leq E_{ep}$, and E_D index is calculated accordingly. Lower value of this index implies higher accuracy.

B. $\aleph_I$ Index: Misclassification

This index is computed with the information on the 3-D spatio-temporal neighborhood granules used for tracking. Its expression is given exploiting the characteristics of IR sensor. That is, the FS would always have brighter values than its background implying that the third dimension (representing different values) of the 3-D object granules should always contain positive values.

As we are dealing with neighborhood granules the maximum deviations in the values of the points within a granule can be Thr_t (3). But the amount of deviation may differ within a set (object/background) if it contains several granules. This deviation should not be much in ideal cases until there occur some misclassifications. This index $\aleph_I$ reflects those cases. Let the object(s) (O_I) and background (B_I) sets over $\aleph_{sp-tmp}$ be

$$O_I = \left\{\tau_{med}(\aleph_{O_i}) \quad \forall \quad \aleph_{O_i} \in FS\right\} \tag{15a}$$

$$B_I = \left\{\tau_{med}(\aleph_{B_i}) \quad \forall \quad \aleph_{B_i} \notin FS\right\} \tag{15b}$$

where $\aleph_{O_i}$ and $\aleph_{B_i}$ represent the ith object and background granules, respectively, and τ_{med} is computed according to (2). The amount of scattering (in terms of statistical mean and maximum deviation) in each of the object (d_o) or background (d_b) set is then computed as

$$d_o = \frac{\text{mean}(O_I)}{\text{maxdev}(O_I)} \tag{16a}$$

$$d_b = \frac{\text{mean}(B_I)}{\text{maxdev}(B_I)}. \tag{16b}$$

In (16), d_o is supposed to be higher as the object granules have higher values and d_b lower. Therefore, lower value of d_o indicates that some part(s) of the background may get included in the object set, resulting in the case of over-tracking. On the other hand, higher d_b indicates incorporation of object granules in background, i.e., under-tracking. The $\aleph_I$ index, defined accordingly incorporating these phenomena, is

$$\aleph_I = \frac{d_b}{d_o}. \tag{17}$$

Therefore, higher the value of $\aleph_I$ is, less accurate is the tracking. One may note that the formation of E_D (14) deals only with the accuracy of foreground segmentation, i.e., how accurately the foreground is segmented. On the other hand, $\aleph_I$ deals with the misclassification of both foreground and background.

V. Results and Discussions

In the present section, the effectiveness of the method of tracking multiple objects, as described before, is demonstrated experimentally. The experiment is carried out with different types of video sequences and more than 2500 frames in total. The experiment is characterized by establishing the effectiveness of the following.

1) Rough rule base for unsupervised video tracking, both in granular and pixel levels.
2) New neighborhood granular rough rule base in tracking over crisp granular and pixel level rule bases.
3) Adaptive rule-base with rough flow graph.
4) Proposed indices for evaluating the tracking quality.
5) Visual and quantitative performance of the proposed method with respect to other popular methods.

A. Results of Tracking

The proposed method of unsupervised tracking is implemented with several video sequences obtained by Kinect sensor and IR sensor. The datasets are obtained from CGD-2011 [39] and thermal and visible imagery [31]. All of the frames of the sequences are of size 240×320 pixels. There are many sequences with different types of movement of human hands which are sensed by Kinect sensor and several surveillance scenarios with movement of people obtained from IR sensor. The results over a few such types of sequences (viz, *M*_2, *M*_4, *M*_9 from CGD-2011, and 2*b* and 5*b* from thermal and visible imagery) are shown here, as examples, where the value of P (the number of previous frames) is chosen as 6. The initial value of Thr_t (3) is assigned as $0.2 \times \max(\tau_{med})$, with an assumption that at least 20% pixels of a frame will be within the moving object. However, if the number of 3-D granules gets changed from frame to frame and only a few granules remain same over all the frames, then Thr_t will get decreased by 0.5 times. It will be increased by 1.5 times in the reverse scenario, i.e., when there will be very few moving granules detected in each frame. Thr_c (4) is initially assigned as 5, according to Weber's law which states that the naked eye can not detect less than five to six consecutive gray level changes.

1) Comparisons Between Neighborhood Granular Level and Pixel Level Methods: The comparative study is performed between the proposed neighborhood granular level (NGrRB) and a pixel level (PRB) rule-base methods. The results are

TABLE II
CPU TIME AND ACCURACY OF NGRRB AND PRB

Sequence	Method	Avg. CPU Time / Frame	Max Time	Min Time	CD
M_2	$NGrRB$	0.381	0.631	0.192	3.24
M_2	PRB	0.331	0.394	0.213	6.11
$2b$	$NGrRB$	0.228	0.366	0.163	3.52
$2b$	PRB	0.186	0.267	0.155	7.16

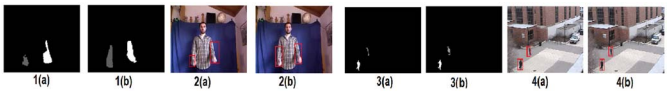

Fig. 8. Visual comparison between the techniques (a) PRB and (b) NGrRB over frame no. 12 of M_2 sequence (1, 2) and frame no. 415 from 5b sequence (3, 4).

TABLE III
CPU TIME AND ACCURACY OF NGRRB, RU, AND NRBFG

Sequence	Method	Time	$Coverage$
M_9	$NGrRB$	0.222	72%
M_9	RU	0.836	97%
M_9	$NRBFG$	0.335	95%
M_4	$NGrRB$	0.308	69%
M_4	RU	0.798	96%
M_4	$NRBFG$	0.369	94%

Fig. 9. Visual comparison between the techniques (a) NGrRB and (b) NRBFG over frame no. 30 of M_9 sequence (1, 2) and frame no. 17 from M_4 sequence (3, 4).

shown in Table II. The metrices over which the comparison is made are: 1) accuracy which is measured based on the distance between the centroids (CD) of the ground truth and the obtained foreground segment of the respective frames and 2) time [average CPU processing time (in seconds) for each frame]. The ground truth that are used here is manually annotated in every frame. If the CD is less it means more accuracy. The experimental results over two video sequences are shown here. The sequence M_2 contains a hand movement scenario and the sequence 5*b* contains a surveillance scenario.

It is seen from Table II that the pixel level method is a bit faster but is less accurate. These results lead to the theoretical conclusion that the PRB provides less indiscernibility than that of NGrRB, thereby producing inferior performance.

Note that the granules that are considered here are overlapping granules with the attribute set of higher cardinality and hence the computation with these granules makes the process little slower compared to PRB. But, it does not contradict the preliminary assumption that granulation makes a process faster. Crisp granulation with same cardinality as that of PRB obviously makes the process faster, but at the cost of accuracy.

The two visual results of frame number 12 from M_2 sequence and frame no. 425 from 5b sequence, as shown in Fig. 8, depict that PRB can not detect the total object when it is moving slower than expected and treat some parts as noise. These were the cases characterized by rules 2 and 3 (Section II-B), and this indiscernibility could not be overcome in [33]. In the first scenario (Fig. 8.1 and 8.2) some parts of the hands are moving slower, whereas in the second scenario (Fig. 8.3 and 8.4) one object (human) is moving slower. Neither, both the hands, nor both the persons can totally be detected as the foreground segment by PRB [see Fig. 8.1(a) and 8.3(a)], i.e., misclassification occurs. The detection and tracking are more accurate with NGrRB [see Fig. 8.1(b) and 8.3(b)], as expected.

2) Comparisons Between With Flow Graph and Without Flow Graph: The effect of rule-base updation with flow graph is discussed here. Without flow graph, there may be two actions, the rule base either will not get updated (NGrRB) or the entire rule base will get updated in every frame (RU). Here, the comparative studies of these instances with the proposed flow graph based method (NRBFG) are shown in Table III. The time (avg CPU time/frame in seconds) and rule-base coverage are the metrices considered in this table. The two datasets over which the results are shown in Fig. 9 represent two types of hand movements where NGrRB fails.

In case of sequence M_9 the left hand of the person was moving initially, and the right hand starts its movement from 30th frame onward. As expected, the movement of the right hand can not be classified by NGrRB [Fig. 9.1(a) and 9.1(b)] according to the inconsistency between the rules 8 and 9 as in Table I which results in less coverage (as shown in Table III). However, it is successfully tracked by NRBFG [see Fig. 9.1(b) and 9.2(b)]. The similar situation occurs in case of M_4 sequence, where the right hand was moving initially and it gets stopped at 17th frame, and the left hand starts its movement. Nothing can be tracked by NGrRB [see Fig. 9.3(a) and 9.4(a)] in this scenario, whereas the moving left hand can be successfully tracked by NRBFG [see Fig. 9.3(b) and 9.4(b)]. It can be noticed from Table III that RU gives good results but is more time consuming as expected, whereas, NRBFG keeps a well balance between the time and accuracy.

3) Comparisons With Other Methods: Comparative studies of the proposed NRBFG method is done with five recent tracking methods, namely, sequential partial filtering on graphs (SPG) [13], PLS-tracking [14], continuous energy minimization-based multiple target tracking (CEMT) [17], sparse representation-based local appearance learning for tracking (LLAS) [20], and Grassmann Manifolds-based online tracking method GMOT [15]. The SPG method is useful for surveillance tracking, PLS is useful for partial occlusion handling, CEMT and LLAS are effective both in case of multiple target tracking and overlapping handling, and GMOT is effective in handling changing shapes of object(s). All of them are partially supervised methods, i.e., initial manual labeling is required for all of the methods. The experiments are conducted over different types of sequences with static background as described in Section V-A. The visual and quantitative results for four of such sequences are given here. Their descriptions are as follows. The sequences M_9 and M_7 represent two hand movement videos sensed by Kinect sensor. In M_9 the left hand of the person was moving initially with changing shapes and sizes and the right hand starts to move from 30th frame onward (see Fig. 10.1). In M_7 sequence both of the

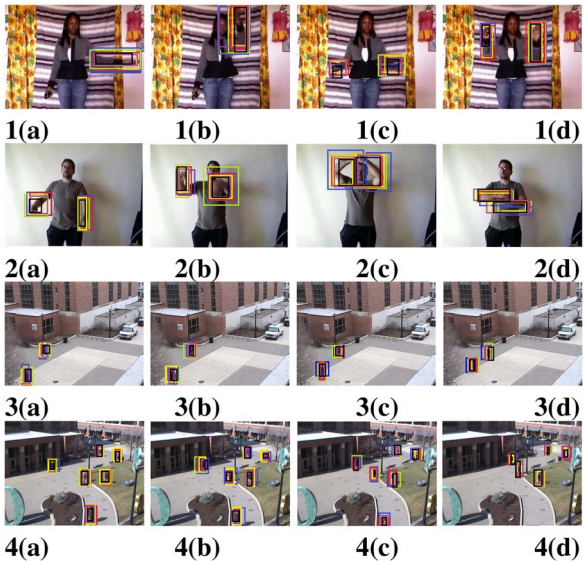

Fig. 10. Tracking results for frame nos. (1) 13, 22, 32, 39 from M_9 sequence (2) 11, 23, 30, 42 from M_7 sequence (3) 370, 440, 471, 510 from $5b$ sequence and (4) 44, 72, 132, 250 from 2b sequence.

TABLE IV
CPU TIME AND ACCURACY OF NRBFG, PLS, SPG, CEMT, LLAS, AND GMOT

Sequence	Metric	NRBFG	PLS	SPG	CEMT	LLAS	GMOT
M_9	CD	4.78	31.42	7.06	4.23	4.42	4.75
M_9	Time	0.325	0.275	0.390	0.352	0.348	0.365
M_7	CD	5.54	7.21	6.86	4.94	4.88	6.21
M_7	Time	0.258	0.323	0.356	0.283	0.265	0.289
$5b$	CD	4.23	6.32	6.22	3.67	3.92	4.11
$5b$	Time	0.308	0.341	0.395	0.325	0.371	0.398
$2b$	CD	5.52	7.02	7.42	4.25	4.12	6.22
$2b$	Time	0.386	0.441	0.495	0.422	0.451	0.462

hands of the person are moving, getting overlapped, and partially occluded to each other (see Fig. 10.2). The sequences $5b$ and $2b$ are two surveillance scenarios obtained from IR sensor. In $5b$, two persons are moving with different speeds (see Fig. 10.3). In the sequence $2b$ six persons are moving in different directions and speeds, one of them is moving in similar colored background and is hardly separable in RGB feature space (see Fig. 10.4). In Fig. 10, the red tracker shows the results obtained by NRBFG, the green tracker shows the results of PLS, the blue tracker shows the results of SPG, the yellow tracker shows the results obtained by CEMT, the magenta tracker shows the results obtained by LLAS, and the black tracker shows the results of GMOT. It can be noticed that all of the sequences are successfully tracked by NRBFG. The overlapping scenarios are better handled by PLS, but it fails to detect new appearance of object in the sequence [Fig. 10(c) and (d)]. The tracking results for SPG are satisfactory for all of the four sequences, but the tracker fails to cover the entire moving objects, whereas both CEMT and LLAS give very satisfactory results even with multiple moving elements.

Quantitative comparisons are shown in Table IV with the same metrices as in Table II for the aforesaid four sequences. The quantitative values of the CD reflect well the visual results. It can be noticed from the table that the computation time is the lowest for the proposed method NRBFG with accuracy better than PLS, SPG, and GMOT, but slightly worse

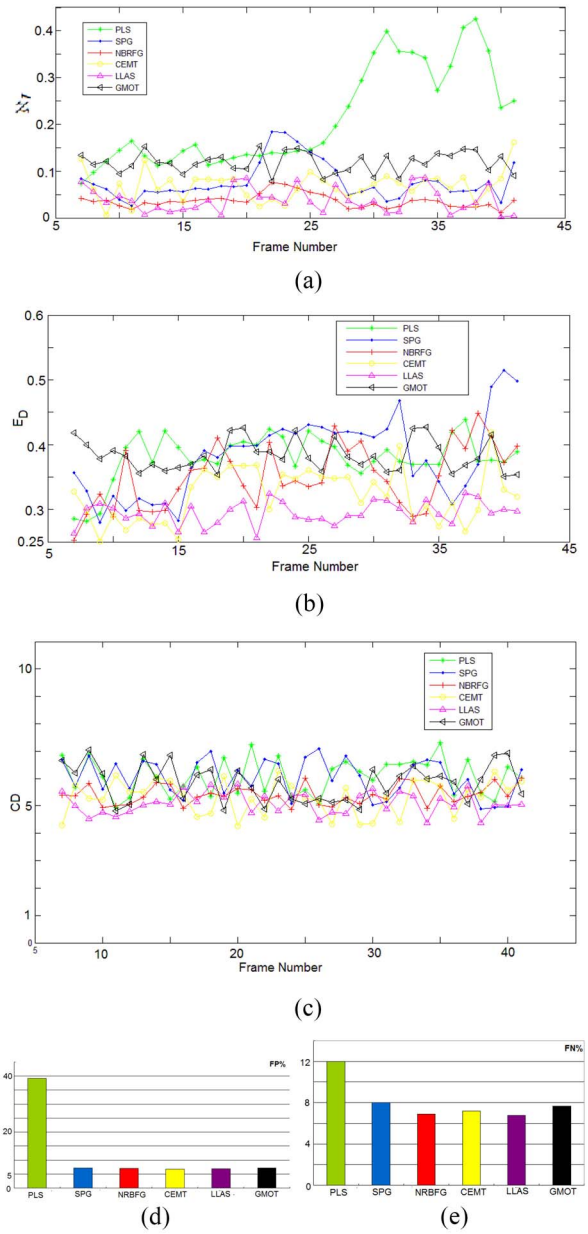

Fig. 11. Quantitative comparisons over M_9 Sequence using (a) $\aleph_I$ index, (b) E_D index, (c) CD, (d) average % of FP pixels, and (e) average % of FN pixels.

than CEMT and LLAS. However, NRBFG requires no manual interactions, i.e., unsupervised.

B. Validation of the Proposed Measures

The effectiveness of the proposed $\aleph_I$ (17) and E_D (14) indices is shown along with their validation with the popular ground truth dependent existing indices, namely, CD, FP pixels, and FN pixels. The CD value for every frame is shown here, whereas the for FP and FN average values are given. The values of these indices obtained for tracking of M_9 sequence with the five methods [SPG (blue, ".−"), PLS (green, "∗−"), NRBFG (red, "+−"), CEMT (yellow, "o−"), LLAS (magenta, "△−")], and GMOT (black, "> −") are shown graphically in Fig. 11. The comparative visual results for frame nos. 13, 22, 32, and 39 are

shown in Fig. 10.1 as example. In case of fame no. 13, [Fig. 10.1(a)], the performances for PLS, NRBFG, LLAS, and CEMT are almost equally good. These are well reflected by $\aleph_I$ and E_D indices [Fig. 11(a) and (b)]. Similar nature can be seen with the corresponding values of CD. In case of frame no. 22 [Fig. 10.1(b)], SPG results in over-tracking by including some parts of the head which NB to the object, and hence increases the inter–intra granular deviations in the object set resulting in higher $\aleph_I$ index as well as higher value of CD. The inclusion of a part of the head leads to increase in FP_{ep} and FN_{ep} values (13), and the E_D index. In the cases of next two frames 32 and 39 [Fig. 10.1(c) and (d)] PLS can not detect one of the moving objects, which results in higher intra granular deviation in background set, and high $\aleph_I$ index (CD is also high for this frame). The performance of the other two methods is satisfactory for these two frames, though SPG can not cover the entire objects resulting in higher $\aleph_I$ and E_D values (same is reflected by CD) compared to that of LLAS, NRBFG, and CEMT. E_D index cannot reflect the mistracking for PLS as it takes care only of the FSs and their corresponding background. These indices for the two frames show that less over-tracking or under-tracking occurs in the FSs of SPG compared to those of PLS, NRBFG, LLAS, GMOT, and CEMT, i.e., the foreground part which gets segmented is correct for SPG.

The average values (in terms of %) of FP pixels (the background pixels classified as the foreground) and FN pixels (foreground pixels classified as background) for this sequence are shown in Fig. 11(d) and (e), respectively. It can be noticed that CEMT provides the least FP value, whereas LLAS provides the least FN value. The proposed NBRFG provides satisfactory results in terms of both. PLS gives high FN values compared to the other methods. This is expected, as this method fails to detect another moving hand in the sequence *M_9*. This misclassification was also reflected by $\aleph_I$. All these demonstrate the effectiveness of the proposed measures.

VI. Conclusion

The problem of adaptive rule generation for unsupervised video tracking in granular computing frame work has been addressed. Concepts of rough flow graph and NRS are used. Several new ideas, definitions, methodologies are introduced in this process. This paper demonstrates how the concept of rough-rule base can be used over 3-D granulated space for unsupervised tracking or tracking without manual interactions, how to formulate a method of updating the rough rule-base using rough flow graph, and introduces new few features like expected flow graph and mutual dependency between flow graphs. This also includes a method of forming 3-D natural granules, and formulating quantitative indices for tracking based on these granules and IR sensed data.

Rough flow graph provides a method of intelligent adaptation of the rule-base and the method NRBFG based on this is proven to be very effective both in terms of accuracy (as compared to NGrRB) and time (as compared to RU), as expected. Two new features, namely, expected flow graph and mutual dependency between flow graphs, defined over the flow graph, make it applicable for testing and validation of the rules. NRBFG provides the best tracking results compared to the other two nonadaptive methods (PRB and NGrRB), however, it takes maximum time among them. Further, incorporation of flow graph for rule-base adaptation enables detection of newly appeared objects in a sequence which is treated as noise in the nonadaptive method NGrRB.

The comparative studies with the five recent partially supervised methods show that PLS has superior ability in handling occlusion, GMOT can handle the variations in object shapes better, SPG can detect the unpredictable changes in videos more accurately, whereas CEMT and LLAS are effective in handling both of the cases. On the other hand, all these problems can be successfully handled by the proposed unsupervised NRBFG with a well balance between time and accuracy.

Granular level computation using 3-D spatio-temporal neighborhood granules is superior to pixel level computation in increasing the discrimination ability of the rule base as well as in reducing the computation time. Indices based on these granules (without ground truth information) are seen to reflect well the quality of unsupervised tracking. These further strengthen the findings in the nonadaptive method [33] for partially supervised tracking with 2-D spatio-color granules.

The present investigation has significance in other areas of decision science where indiscernibility is prominent. The proposed algorithm for unsupervised tracking can be extended for more challenging data sets, e.g., with moving camera. The granular adaptive rough rule base could be highly effective in dealing with large scale video data due to its characteristics of reducing computation time without affecting the accuracy much.

We have used here the NRS as a model of uncertainty handling. The characteristic difference among NRS, fuzzy rough sets (FRS), and composite rough sets (CRSs) are as follows. CRS is mainly suitable for dealing with incompleteness, whereas NRS and FRS are more useful for handling ambiguities arising from overlapping characteristics of classes. FRS needs to define membership function for characterizing the overlapping granules, while NRS does not need to define the size of the granules.

Acknowledgment

The authors would like to acknowledge Prof. D. K. Das, Jadavpur University for his support. S. K. Pal would also like to acknowledge the J. C. Bose National Fellowship and DAE-BRNS Senior Scientist Scheme (RRF) of the Government of India.

References

[1] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data*. Norwell, MA, USA: Kluwer Academic, 1992.

[2] R. W. Świrniaski, "Rough sets methods in feature reduction and classification," *Int. J. Appl. Math. Comput. Sci.*, vol. 11, no. 3, pp. 565–582, 2001.

[3] J. Komorouski, Z. Pawlak, L. Polkowski, and A. Skowron, "Rough sets: A tutorial," in *Rough Fuzzy Hybridization: A New Trend in Decision-Making*, S. K. Pal and A. Skowron, Eds. Singapore: Springer, 1999, pp. 3–98.

[4] D. Sen and S. K. Pal, "Generalized rough sets, entropy, and image ambiguity measures," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 1, pp. 117–128, Feb. 2009.
[5] S. K. Pal, B. U. Shankar, and P. Mitra, "Granular computing, rough entropy and object extraction," *Pattern Recognit. Lett.*, vol. 26, no. 16, pp. 2509–2517, 2005.
[6] D. Sen and S. K. Pal, "Histogram thresholding using fuzzy and rough measures of association error," *IEEE Trans. Image Process.*, vol. 18, no. 4, pp. 879–888, Apr. 2009.
[7] A. Albanese, S. K. Pal, and A. Petrosino, "Rough sets, kernel set, and spatiotemporal outlier detection," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 194–207, Jan. 2014.
[8] A. E. Hassanien, A. Abraham, J. F. Peters, G. Schaefer, and C. Henry, "Rough sets and near sets in medical imaging: A review," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 6, pp. 955–968, Nov. 2009.
[9] S. K. Pal and S. K. Meher, "Natural computing: A problem solving paradigm with granular information processing," *Appl. Soft Comput.*, vol. 13, no. 9, pp. 3944–3955, 2013.
[10] J. T. Yao, A. V. Vasilakos, and W. Pedrycz, "Granular computing: Perspectives and challenges," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1977–1989, Dec. 2013.
[11] L. A. Zadeh, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets Syst.*, vol. 90, no. 2, pp. 111–127, 1997.
[12] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surveys*, vol. 38, no. 4, pp. 1264–1291, 2006.
[13] P. Pan and D. Schonfeld, "Video tracking based on sequential particle filtering on graphs," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1641–1651, Jun. 2011.
[14] Q. Wang, F. Chen, W. Xu, and M.-H. Yang, "Object tracking via partial least squares analysis," *IEEE Trans. Image Process.*, vol. 21, no. 10, pp. 4454–4465, Oct. 2012.
[15] Z. H. Khan and I. Y.-H. Gu, "Nonlinear dynamic model for visual object tracking on Grassmann manifolds with partial occlusion handling," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 2005–2019, Dec. 2013.
[16] M. A. A. Aziz, J. Niu, X. Zhao, and X. Li, "Efficient and robust learning for sustainable and reacquisition-enabled hand tracking," *IEEE Trans. Cybern.*, vol. 46, no. 4, pp. 945–958, Apr. 2016.
[17] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 58–72, Jan. 2014.
[18] K. Zhang, L. Zhang, and M.-H. Yang, "Fast compressive tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 2002–2015, Oct. 2014.
[19] F. Pernici and A. D. Bimbo, "Object tracking by oversampling local features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2538–2551, Dec. 2014.
[20] T. Bai, Y.-F. Li, and X. Zhou, "Learning local appearances with sparse representation for robust and fast visual tracking," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 663–675, Apr. 2015.
[21] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1778–1792, Nov. 2005.
[22] B. Dey and M. K. Kundu, "Robust background subtraction for network surveillance in H.264 streaming video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 10, pp. 1695–1703, Oct. 2013.
[23] Y. Yin, D. Xu, X. Wang, and M. Bai, "Online state-based structured SVM combined with incremental PCA for robust visual tracking," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1988–2000, Sep. 2015.
[24] X. Zhou, X. Li, and W. Hu, "Learning a superpixel-driven speed function for level set tracking," *IEEE Trans. Cybern.*, vol. 46, no. 7, pp. 1498–1510, Jul. 2016.
[25] Z. Pawlak, *Flow Graphs and Data Mining*. Heidelberg, Germany: Springer-Verlag, 2005.
[26] K. Lisowski and A. Czyzewski, "Modelling object behaviour in a video surveillance system using Pawlak's flowgraph," in *Proc. 7th Int. Conf. (MCSS)*, Kraków, Poland, 2014, pp. 122–136.
[27] C.-M. Huang and L.-C. Fu, "Multitarget visual tracking based effective surveillance with cooperation of multiple active cameras," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 1, pp. 234–247, Feb. 2011.
[28] N. Liu, H. Wu, and L. Lin, "Hierarchical ensemble of background models for PTZ-based video surveillance," *IEEE Trans. Cybern.*, vol. 45, no. 1, pp. 89–102, Jan. 2015.
[29] H. P. H. Shum, E. S. L. Ho, Y. Jiang, and S. Takagi, "Real-time posture reconstruction for microsoft kinect," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1357–1369, Oct. 2013.
[30] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1318–1334, Oct. 2013.
[31] J. W. Davis and V. Sharma, "Background-subtraction using contour-based fusion of thermal and visible imagery," *Comput. Vis. Image Understand.*, vol. 106, nos. 2–3, pp. 162–182, 2007.
[32] C. Sinthanayothin, N. Wongwaen, and W. Bholsithi, "Skeleton tracking using kinect sensor and displaying in 3D virtual scene," *Int. J. Adv. Comput. Technol.*, vol. 4, no. 11, pp. 213–223, 2012.
[33] D. B. Chakraborty and S. K. Pal, "Neighborhood granules and rough rule-base in tracking," *Nat. Comput.*, pp. 1–12, 2015, doi: 10.1007/s11047-015-9493-6.
[34] Q. Hu, D. Yu, J. Liu, and C. Wu, "Neighborhood rough set based heterogeneous feature subset selection," *Inf. Sci.*, vol. 178, no. 18, pp. 3577–3594, 2008.
[35] Y. Du, Q. Hu, P. Zhu, and P. Ma, "Rule learning for classification based on neighborhood covering reduction," *Inf. Sci.*, vol. 181, no. 24, pp. 5457–5467, 2011.
[36] S. K. Pal, S. Mitra, and P. Mitra, "Rough-fuzzy MLP: Modular evolution, rule generation, and evaluation," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 1, pp. 14–25, Jan./Feb. 2003.
[37] T. Nawaz, F. Poiesi, and A. Cavallaro, "Measures of effective video tracking," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 376–388, Jan. 2014.
[38] E. Maggio and A. Cavallaro, *Video Tracking—Theory and Practice*. Chichester, U.K.: Wiley, 2010.
[39] ChaLearn Gesture Dataset (CGD 2011), ChaLearn, California, 2011.

Sankar Kumar Pal (M'80–SM'84–F'93–LF'15) received the Ph.D. degrees from Calcutta University, Kolkata, India, and Imperial College London, London, U.K.

He joined the Indian Statistical Institute, Kolkata, in 1975, as a CSIR Senior Research Fellow, where he became a Full Professor in 1987, a Distinguished Scientist in 1998, and the Director in 2005. He is a Raja Ramanna Fellow of the Government of India and a Former Indian National Academy of Engineering Chair Professor. He founded the Machine Intelligence Unit and the Center for Soft Computing Research with the Institute in Calcutta which are enjoying international recognition. He was with the University of California at Berkeley, Berkeley, CA, USA, and the University of Maryland at College Park, College Park, MD, USA, the NASA JSC, Houston, TX, USA, and the U.S. Naval Research Laboratory, Washington, DC, USA. He held several visiting positions in Italy, Poland, Hong Kong, and Australian Universities. He has co-authored 19 books and over 400 research publications in the areas of pattern recognition and machine learning, image processing, data mining, Web intelligence, soft computing, bioinformatics, and cognitive machines.

Prof. Pal was a recipient of several national and international awards, including the most coveted S. S. Bhatnagar Prize and the Padma Shri by the President of India, the G. D. Birla Award, and the P. C. Mahalanobis Birth Centenary Gold Medal of Indian Science Congress Association by the Prime Minister of India for lifetime achievement, the NASA Tech Brief Award in USA, and the Khwarizmi International Award from the President of Iran. He has been serving as a Distinguished Visitor of the IEEE Computer Society since 1987. He is/was on the editorial boards of 22 journals, including some IEEE TRANSACTIONS. He visited about 40 countries as a keynote/plenary/invited speaker. He is a fellow of The World Academy of Sciences for the Advancement of Science in Developing Countries, International Association for Pattern Recognition, International Fuzzy Systems Association, International Rough Set Society, and all four National Academies for science/engineering in India.

Debarati Bhunia Chakraborty received the B.Tech. degree in electronics and communication engineering from the Siliguri Institute of Technology, Chamta, India, in 2008, the M.Tech. degree from the National Institute of Technology at Rourkela, Rourkela, India, in 2010, and the Ph.D. degree from Jadavpur University, Kolkata, India, in 2016.

She is currently a Visiting Scholar with the Indian Statistical Institute, Kolkata. Her current research interests include video processing, rough sets, granular computing, soft computing, and computer vision.

Mrs. Chakraborty was a recipient of the Prestigious Young Scientist Award from Indian Science Congress Association in 2012.