Computing and Informatics, Vol. 37, 2018, 717–736, doi: 10.4149/cai_2018_3_717

DEFORMABLE OBJECT TRACKING USING CLUSTERING AND PARTICLE FILTER

Muhammad Aasim Rafique, Moongu Jeon*

School of Information and Communications
Gwangju Institute of Science and Technology (GIST)
Gwangju, Republic of Korea
e-mail: {arafique, mgjeon}@gist.ac.kr

Malik Tahir Hassan

University of Management and Technology
Lahore, Pakistan
e-mail: tahir.hassan@umt.edu.pk

Abstract. Visual tracking of a deformable object is a challenging problem, as the target object frequently changes its attributes like shape, posture, color and so on. In this work, we propose a model-free tracker using clustering to track a target object which poses deformations and rotations. Clustering is applied to segment the tracked object into several independent components and the discriminative parts are tracked to locate the object. The proposed technique segments the target object into independent components using data clustering techniques and then tracks by finding corresponding clusters. Particle filters method is incorporated to improve the accuracy of the proposed technique. Experiments are carried out with several standard data sets, and results demonstrate comparable performance to the state-of-the-art visual tracking methods.

Keywords: Visual object tracking, data clustering, object segmentation, cluster correspondence

* Corresponding author

1 INTRODUCTION

Visual object tracking (VOT) has numerous applications in surveillance, intelligent transportation systems, sports broadcasting, robotics and so on. Single object tracking is a base case and, usually, extended to track multiple objects in a scene. Video analysis is affected by the video quality, scene environment attributes (illumination, noise, shadow and jitter), spatio-temporal attributes and behavioral change of intrinsic properties of a target object (such as shape, color and size). The uncertain behavior of the intrinsic properties over the length of a video sequence is deformation of the object.

Single object tracking has defined a work-flow for general VOT cases. The general strategy considers detection of the target region (target object), representation of the target object and activity of the target object. An effective VOT technique is efficient and wise combination of the aforementioned. A brief description of each of the stated component work-flow is worth mentioning. The target region is selected as a preliminary geometric shape such as a quadrilateral or ellipse; these shapes provide with a benefit of handling few parameters and a disadvantage of redundant information besides the target object. Specific contours are used to avoid the ineffective data for demarcation of the target object precisely, but it burdens with many parameters to track along. Adaptive selection of the target region can be an effective way to track deformed objects, but it might come with an associated computational cost.

Second challenge is representation of the target object. The simplest representation is the pixels of target regions as color values. The basic RGB color values are vulnerable to the underlying challenges of video analysis, thus they are not invariant representation of the target object. Histogram of colors is an effective representation for alternating color change in the target region. Expensive features which are invariant to color, rotations and motion are some other choices for the target object representation. Well known features are edges, HAAR-like features, SIFT and SURF features, HoG features, etc. Motion representation of the target object relates its motion within vicinity of the current locality. However, a super fast object can disturb the tracking results drastically. Alternatively, one can model the motion from the initial frames of the video which can, later, be used to predict the location of the target object. Probabilistic Gaussian motion model, Kalman filters and particle filters, optical flow trackers, etc., are commonly used motion models.

In the end, prediction of the target object is required to conclude one step of tracking. Prediction may be as simple as template matching and can extend to complex sophisticated discriminative classifiers. Tracking of deformable objects is a situation, where the object alternatively changes color, shape and scale with motion. These variations make it hard to track deformable objects optimally, as a general case. In this work, we propose a spatio-temporal representation of target object and an optimal method to model the activity of the target object. The spatial representation of the target object is segmented using data clustering techniques, and the temporal representation is given by solving the clustering correspondence problem. Moreover, particle filtering technique is used to model the activity of the target object.

This paper is organized as follows. Section 2 presents the relevant literature survey. In Section 3, we explain our proposed method. Section 4 articulates the evaluation setup used to cross examine and describe our experiments to test the performance of our proposed method, and presents the obtained results with discussion. Conclusion and possible future directions are briefed in Section 5.

2 RELATED WORK

The single object tracking problem is an active research area, and persuasive literature is available for study. Comprehensive evaluations and contemplative discussions, with summaries of most of the interesting techniques, are aggregated in literature for interested users [26, 28, 25]. Another recent review evaluating the single object tracking techniques on different video sequences will be helpful for survey [17]. In addition to the aforementioned references, it will be beneficial to discuss recent progress in the single object tracking domain. Structure-preserving object tracker (SPOT) [30, 29] uses online structured SVM to learn the spatial constraints of different parts of the objects, and it predicts from the candidate windows for object tracking. Lucas-Kanade algorithm [19] is extended as an optimization problem in [23] where the object's pixels and the background segmentation are optimized by applying likelihood of a Bayesian framework. Incremental subspace learning and Fisher discriminant analysis techniques are combined, and a graph based combination is proposed to effectively capture the dynamic appearance of the target object and differentiate it from the background [32]. Another graph inspired technique used graph cut method for object segmentation, and it improved the object tracking results, reported in [31].

Since there are plenty of techniques employing variety of strategies to approach the single object tracking problem a rational thought is to discuss the pertinent literature which follows henceforth. Mean shift is used to find best candidate windows for the target object from the next frame by matching histograms discrimination information from the Bhattacharya coefficients [4]. The target region is divided into static segments of 20×20 pixel values, and each segment is associated with a separate Kalman filter in [22]. Later, the object tracking is performed using template matching. A likely idea is to divide the target object in fragments of fixed size and use the color histogram of these fragments to compare the probable matches from candidate segments with Earth Movers Distance (EMD) [2] to track. A recent work in similar regards is the representation of the segmented target object by a superpixel per segment. A superpixel is defined by the center of mass and average HSV-values [24], and EMD is used for comparisons. The target object state is sampled using particle filter for the segments. Key-points are used with hierarchical clustering techniques for deformable object tracking in [21].

Deformable object tracking has been aimed by many researches from general to specific cases. As discussed in Section 1, the challenges put forth by change of shape, occlusion, motion activities, and so on, recognized the deformable object tracking as a standalone task. A nonlinear model with implicit representation of the target object by contours and defining generative dynamical model for the motion is presented in early literature [13]. The boundary element method is applied with a deformable template to model the displacements, and the template is registered to the image by energy minimization of the force field [8]. Later, the idea is extended with the use of canny edge detector for occlusion [9]. An optical flow equation applied on the whole image with constraints on the elastic deformation is discussed in [12]. Deformable objects are tracked using a sliding window particle filter, where the change in an object's shape is captured using a modified technique of principle component analysis [16].

Dynamic graphs are employed in tracking to represent the geometrical structure of the target and the candidate object as nodes, and their interaction is denoted by edges; Markov random field and spectral clustering is used to solve the target and the candidate graph matching [3]. A recent work used the weightless neural networks for tracking the deformable objects to a success [27]. [18] discussed a path based tracking which overcame the limitation of core reliance on the initialization by intelligently selecting the correct patches. [5] proposed use of hyper-graph for guessing correspondence in deformable object in successive multiple frames, which helped in long-term occlusions and intense deformations. Fusion of the data from multiple sensors used with a multiple Kalman filters tracking technique to improve visual tracking is presented in [15].

In comparison to existing techniques, we propose the use of clustering, an unsupervised technique, to segment the target object into parts, and use these parts wisely to track the object. We keep with us the discriminative parts of the reference (target) object, and estimate the location of matching parts in the vicinity of the object in the previous frame. Moreover, particle filtering is incorporated into the method to make it more robust to the tracking challenges. We shall discuss the formal details of our methodology in coming sections.

3 OUR METHODOLOGY

Formally defining the single object tracking problem: given a sequence of N images $I_1, I_2, \ldots, I_N$, and an initializing bounding box ground truth region $b_g = b_1$ in I_1 containing the object to be tracked, we aim at predicting the bounding boxes $b_2, \ldots, b_N$ that contain the target object in remaining frames of the sequence $I_2, \ldots, I_N$, respectively. The detail of our clustering and particle filter based tracking method TUC (tracking using clustering) is provided in the remainder of this section. We call the target object to be tracked as *tracked object* or *reference object*, and the estimated object as the *predicted object* alternatively.

3.1 Clustering for Object Segmentation

Data clustering discovers groups of similar patterns in data and its application for image segmentation is quite intuitive. In our first step, we obtain k segments of b_g, the initially provided ground truth region in I_1, using k-means clustering method. K-means is chosen for its efficiency and simplicity. Note that although clustering is expensive for large data yet applying it to a usually small region like b_g is not computationally expensive. These k segments of b_g become the reference segments that will be compared with the segments of test regions in next frames to estimate the tracked object's location.

3.1.1 Number of Clusters

Number of clusters k is an input parameter for k-means. We tested different values for k and empirically fixed it to 15 being a good tradeoff between accuracy and efficiency. Figure 1 shows the segments of an object discovered using different values of k.

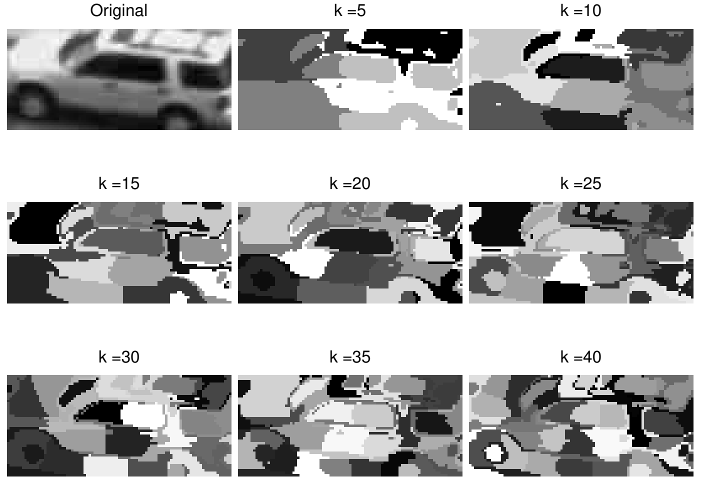

Figure 1. Segments of the object using different number of clusters, $k \in \{5, 10, \dots, 40\}$

3.1.2 Feature Selection

Feature selection can be regarded as the most important part in any computer vision, machine learning and pattern recognition algorithm in general, and in a tracking method in particular. We segment the object using pixel location, gray intensity, and x- and y-directional gradient values. The separation of salient segments in Figure 1 justifies the suitability of using these features.

3.2 Selecting Discriminative Segments

In practice, the target object's neighborhood may contain textures that are similar to the target object itself and can hinder the tracker's accuracy. Considering the fact that the far regions has less to add to this obstruction, we select the segments of the reference object that are most discriminative from the immediate background. We take four neighboring regions up, down, left and right of the object having same size as the object, and segment each of these regions with same k value (Figure 2). The segments of the reference object b_g that have high similarity with the segments from neighboring regions are removed and not used as reference segments. Thus, we obtain the set of most discriminative segments of the reference object, S_g. We removed the top 25 % most similar segments to the background in our experimentation.

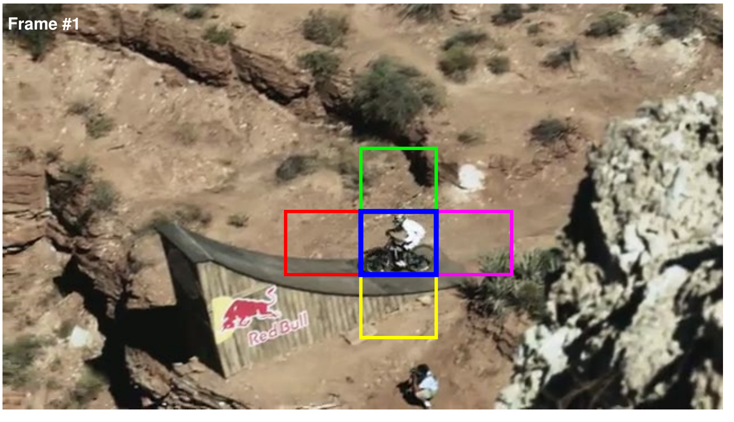

Figure 2. The four background boxes around the tracked object are shown that are used to calculate discriminative segments of the object

3.3 Object Tracking Using Segments

Once we have the discriminative segments of the reference object from the frame I_1, the next step is to locate and track the object in subsequent frames $I_2, \ldots, I_N$. For this, we pick the region b_n in frame I_n where $n \in \{2, 3, \ldots, N\}$ in sequence, using the immediate previous frame's region information, i.e., the location, width and height of the bounding box in frame I_{n-1}. A realistic assumption which will be relieved later is that the object is not moving too fast from I_{n-1} to I_n, and we get some part of the object in b_n to estimate the object's location in I_n. However, such fast motion situations are handled by incorporating particle filter in our method. Detail of using particle filter is presented in Section 3.5.

Thus clustering is applied on region b_n to obtain the set of k segments S_n, and these segments in S_n are then compared with the reference segments in S_g. This comparison, however, demands to solve the segments correspondence problem, which is discussed in detail in Section 3.4. The segments correspondence problem enables us to compute the amount of translation between two corresponding segments by using centroids of the segments. As different pairs of corresponding segments suggest different translation values, we take the median of these translation values and predict the translated location of the bounding box in I_n. Hence, the change in locations of the corresponding segments in S_n and S_g helps us estimate the distance the object has traveled.

3.4 Finding Corresponding Segments

Finding correct corresponding segments in the set of current segments S_n and the set of reference segments S_g is of key importance in our method, and we are able to solve this correspondence problem pretty accurately. Different regional properties of the segments are compared to calculate their similarity. These regional properties include area, eccentricity, Euler number, mean intensity and normalized intensity range of a segment. Area is the number of pixels in a region. Area is computed as actual number of pixels in a segment. Eccentricity specifies the eccentricity of the ellipse that has the same second-moments as the region, analogically it represents how circular the region is. Eccentricity is computed as a ratio of the distance between the foci of the ellipse and its major axis length. A line segment has 1 eccentricity and a circle has 0 eccentricity. Euler number specifies the number of objects in the region minus the number of holes in those objects. Mean intensity is the average intensity value of a region, and normalized intensity range of a region is defined as:

$$\frac{(\mathit{MaxIntensity} - \mathit{MinIntensity})}{255}.$$

Euclidean distances between each pair of segments is calculated based on these regional properties.

$$\mathrm{dist}(s_i, s_j) = \sqrt{\sum (u_i - v_j)^2}, \quad \forall s_i \in S_n, s_j \in S_g, \tag{1}$$

where u_i and v_j represent the vectors of the regional properties of segments s_i and s_j, respectively.

In addition to this distance calculation of regions, overlap of each pair of segments is also computed using Jaccard index as follows:

$$o(s_i, s_j) = \frac{|s_i \cap s_j|}{|s_i \cup s_j|}, \quad \forall s_i \in S_n, s_j \in S_g. \tag{2}$$

Finally, the similarity of two segments $s_i \in S_n$ and $s_j \in S_g$ is computed as:

$$\mathrm{sim}(s_i, s_j) = \alpha \cdot o(s_i, s_j) + \beta \cdot \frac{1}{\mathrm{dist}(s_i, s_j)}. \tag{3}$$

We fixed α and β values to be 0.25 and 0.75, respectively, based on empirical results. Section 4.5 shows the impact of various combinations of α and β value on the quantified results.

The correspondence solving method returns matching segments in S_n and S_g along with the confidence weights based on similarities of the corresponding segments. Since there exist low similarity pairs of segments, we pick the top 75 % of the segment matches based on these confidence weights, and use them for tracking.

3.5 Incorporating Particle Filter

Particle filtering is used to approximate the intractable distributions for sample generation techniques. It starts by generating a random set of particles and it estimates states and observations for the next time step. It overcomes the limitation of unnormalized and non-gaussian distributions and generate samples using the weighted previous observations. It is interesting to initialize the particles and weights updating strategy, what is a domain specific gimmick.

We incorporate particle filtering into our clustering based tracking method to improve its robustness and to behave well with less accurate clustering. P particles are sampled from a 2-d Gaussian distribution centered at the center of the target object in previous frame, with covariance matrix V. Initial weight to every particle is assigned based on two measures. First, the sum of distances of a particle p to all the centers of the reference object's segments c_i^g; call it w_p^d. Second, the correlation of the reference window b_g and the same sized window centered at the particle b_p; call it w_p^r.

$$w_p^d = \sum_{i=1}^{k} \mathrm{dist}(p, c_i^g), \tag{4}$$

$$w_p^r = \mathrm{corr}(b_p, b_g), \tag{5}$$

w_p^d and w_p^r are normalized by their total sum values and then combined to find initial weight of the particle p as:

$$w_p = \frac{1}{w_p^d} \cdot \exp(w_p^r), \tag{6}$$

w_p is normalized to sum to 1. The estimate for object's motion in current frame is computed using clustering as described in the previous steps of this section. Next step is to move every particle using this estimated amount of motion. Instead of using the single motion value, we sample P motion values from a 2-d Gaussian distribution centered at the estimated amount of motion, and having covariance V_d. Updated weights are calculated again using particles' distance from reference centers and correlation with the reference window. Finally, particle with the maximum weight is picked as center of the target object's new location. Figure 3 gives a small demo of our tracking method by showing the object, the estimated bounding box and the particles.

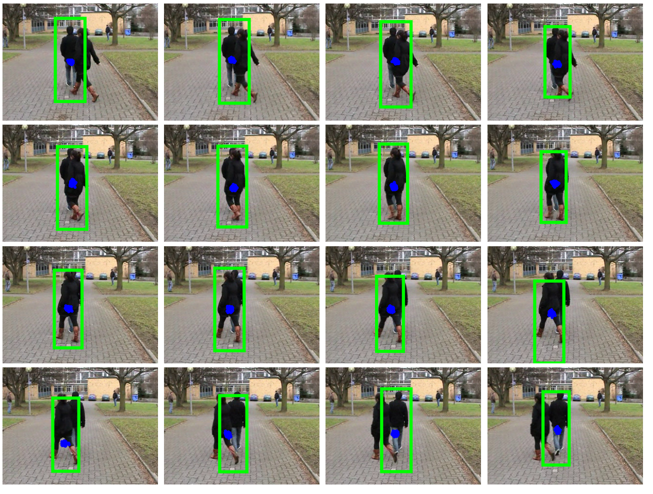

Figure 3. Tracked object (the person moving straight) and particles are shown in 16 consecutive frames from top-left to bottom-right (person crossing data set). Successful occlusion handling is also visible.

3.6 Scale Estimation

The estimation of change in scale of the tracked object is assisted by the nature of our clustering based procedure. Corresponding segments or clusters of the true object b_g and the predicted object b_n are identified, as described in Section 3.4, and the sizes of these corresponding segments in S_g and S_n are compared. The ratio of their sizes gives an estimate of the scale-change factor δ_{scale}. As different corresponding segments give different estimate values, δ_{scale} is set to be the median of these values.

$$\delta_{scale} = \text{median}\left(\frac{|s_i^c|}{|s_j^c|}\right), \quad \forall s_i^c \in S_n, s_j^c \in S_g. \tag{7}$$

The superscript c indicates that these are the corresponding segments of the current and ground truth segments, S_n and S_g, respectively. $|.|$ is the size of the segment calculated as count of pixels in the segment, also known as area. δ_{scale} is used to get the updated width w_n and height h_n of the predicted bounding box b_n.

$$[w_n, h_n] = \sqrt{\delta_{scale}} \cdot [w_g, h_g], \tag{8}$$

where w_g and h_g are the width and height of the ground truth bounding box b_g, respectively.

The steps of our methodology are summarized in Algorithm 1.

Algorithm 1 TUC – *T*racking *U*sing *C*lustering

Require: $I_1, \ldots, I_N$ {image sequence}, b_1 {bounding box in I_1}
$k \leftarrow 15$ {initialize number of clusters}
$P \leftarrow 200$ {initialize number of particles}
$F_1 \leftarrow \mathit{computeFeatures}(b_1)$ {features of ground truth b_g}
$S_g \leftarrow \mathit{kmeans}(F_1, k)$ {segments of the object b_g}
$S_g^d \leftarrow \mathit{findDiscriminativeSegments}(S_g, I_1)$ {discriminative reference segments, Section 3.2}
for n = 2 to N **do**
  $P_{xy} \leftarrow \mathit{generateParticles}(c_n^0, V, P)$ {Section 3.5 and Equation (11)}
  $w_p \leftarrow \mathit{assignWeights}(P_{xy})$
  $P_{xy} \leftarrow \mathit{resample}(P_{xy}, w_p)$
  $F_n \leftarrow \mathit{computeFeatures}(b_n^0)$ {b_n^0 is the box in current frame using previous frame's box information}
  $S_n \leftarrow \mathit{kmeans}(F_n, k)$
  $MATCHES \leftarrow \mathit{findCorrespondingSegments}(S_n, S_g^d)$ {Section 3.4}
  $t_{xy} \leftarrow \mathit{estimateTranslation}(MATCHES)$ {Section 3.3}
  $t' \leftarrow \mathit{generateRandomSpeeds}(t_{xy}, V_d, P)${Section 3.5 and Equation (12)}
  $P_{xy} \leftarrow P_{xy} + t_{xy} + t'$ {move the particles with estimated and random speeds}
  $w_p \leftarrow \mathit{assignWeights}(P_{xy})$
  $c_n \leftarrow \max(w_p, P_{xy})$ {estimated center of the object}
  $b_n^0 \leftarrow \mathit{boundingBox}(c_n)$
  $\delta_{scale} \leftarrow \mathit{estimateScale}(b_n^0, b_g)$ {Section 3.6}
  $b_n \leftarrow \mathit{scale}(b_n^0, \delta_{scale})$
  return b_n {predicted bounding box in I_n}
end for

4 EXPERIMENTAL EVALUATION

We compare our method with state-of-the-art tracking methods on standard data sets using a popular evaluation measure. Our experimental setup and obtained results are discussed in this section.

4.1 Data Sets

Experimental evaluation of our tracking method is carried out on nine standard publicly available data sets.[1] The video sequences in these data sets contain different visual tracking challenges like deformation, in-plane rotation, out-of-plane rotation, scale change, occlusions, etc. Figure 4 shows the first frames of these video sequences and the target object to be tracked.

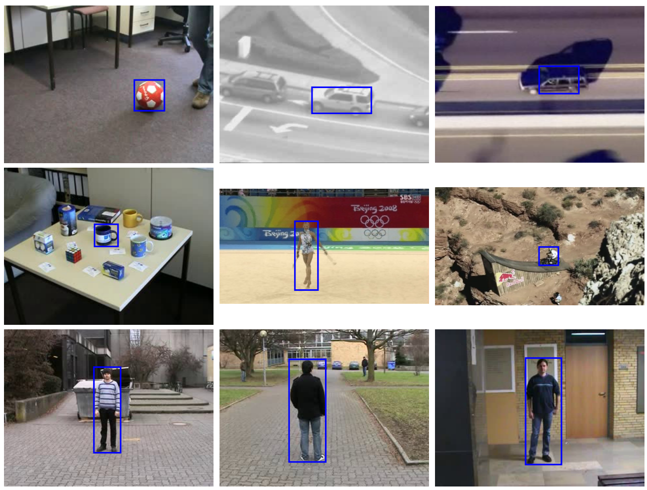

Figure 4. First frame and the ground truth bounding box are shown for each of the nine video sequences used in experimental evaluation. The video sequences from top-left to bottom-right are ball, car2, car chase, cup on table, gym, mountain bike, person, person crossing and person occlusion.

4.2 Evaluation Measure

Many measures exist in literature for quantitative evaluation of tracking methods. The center-error measure expresses the distance between the centroid of the predicted box and the centroid of the ground truth. This measure is not bounded and ignores the scale and the aspect ratio of the bounding boxes. We have selected the commonly used overlap measure:

$$o(b_n, b_g) = \frac{|b_n \cap b_g|}{|b_n \cup b_g|}, \tag{9}$$

where b_n refers to the predicted bounding box and b_g refers to the ground truth bounding box. This measure is bounded between 0 and 1, penalizes translation

[1] `http://www.gnebehay.com/cmt/`

and scale alterations, and is popularly known to be a better indicator for per-frame success [20].

In order to find an overall score for a sequence, a threshold τ is applied on Equation (9) to find true positives (TP). True positive rate (or recall) is then reported for all sequences.

$$\text{recall} = \frac{TP}{TP + FN}. \tag{10}$$

The value of recall gives the percentage of frames that are tracked correctly, i.e. when $o \geq \tau$.

Results are computed for three different values of τ, i.e., 0.25, 0.50 and 0.75. These threshold values are suggested by [20] with an interpretation as low, medium and high requirements on accuracy.

4.3 Comparison Methods

A comparison of our approach is performed with the state-of-the-art tracking approaches. The comparison methods include CMT (Consensus-based Matching and Tracking [20, 21]), STRUCK (Structured output Tracking [10]), TLD (Tracking-Learning-Detection [14]), LM (LearnMatch [11]), FT (Fragments-based Tracking [2]), HT (HoughTrack [6]) and SB (Semi-supervised online Boosting [7]).

4.4 Parameters Setting

Required parameters of our method were set once and then used for all of the data sets consistently. The setting was guided by initial experimental results.

The number of clusters parameter k which becomes the number of tracked segments is set to be 15. Number of particles P is set to be 200. Covariance matrix V for initial random Gaussian particles is set to be

$$V = \begin{bmatrix} 7 & 1 \\ 1 & 7 \end{bmatrix}, \tag{11}$$

and covariance matrix for random motions of the particles V_d is set to be

$$V_d = \begin{bmatrix} 2 & 1.5 \\ 1.5 & 2 \end{bmatrix}. \tag{12}$$

Covariance matrix V is used to generate initial random Gaussian particles. The shape of the target object (width and height of the bounding box) and the dominant direction of motion can help in determining this spread to be more in one direction or other (we fixed to 1 and 7 in our experiments). V controls the spread of particles and can be learned through some initial frames or adapted incrementally (not done in the current work). In the case of V_d, the covariance matrix of random motions, the values are small and almost identical for both horizontal and vertical directions (1.5 and 2).

Sequence	τ	**CMT**	**STR**	**TLD**	**FT**	**LM**	**HT**	**SB**	**TUC**
	0.25	**0.98**	0.30	0.40	0.31	0.14	0.15	0.30	0.90
ball	0.50	0.57	0.15	0.28	0.19	0.12	0.11	0.28	**0.58**
	0.75	**0.19**	0.10	**0.19**	0.13	0.09	0.10	0.12	0.15
	0.25	0.90	0.81	**1.00**	0.04	0.46	0.59	0.72	0.98
car2	0.50	0.88	0.47	**1.00**	0.04	0.36	0.47	0.72	0.94
	0.75	0.64	0.11	**0.95**	0.03	0.17	0.00	0.70	0.72
	0.25	0.30	0.08	0.16	0.04	0.00	0.04	0.08	**0.32**
carchase	0.50	**0.20**	0.03	0.15	0.03	0.00	0.04	0.08	0.13
	0.75	**0.07**	0.02	0.06	0.02	0.00	0.00	0.05	0.04
	0.25	0.83	**1.00**	0.89	**1.00**	0.68	**1.00**	0.47	**1.00**
cup on table	0.50	0.81	0.92	0.64	0.88	0.54	**1.00**	0.47	0.98
	0.75	**0.61**	0.35	0.06	0.40	0.31	0.48	0.34	0.53
	0.25	0.93	**1.00**	0.76	0.24	0.10	0.30	0.61	**1.00**
gym	0.50	0.86	**0.93**	0.32	0.22	0.05	0.00	0.58	0.89
	0.75	0.22	0.3	0.08	0.12	0.02	0.00	0.22	**0.36**
	0.25	0.99	0.99	0.37	0.65	0.11	0.99	0.20	**1.00**
mount-bike	0.50	**0.98**	0.93	0.36	0.63	0.08	0.40	0.17	0.88
	0.75	**0.48**	0.23	0.16	0.18	0.04	0.03	0.08	0.27
	0.25	0.95	**1.00**	0.92	**1.00**	0.75	0.49	0.52	**1.00**
person	0.50	0.82	0.95	0.71	0.95	0.67	0.00	0.52	**0.99**
	0.75	0.49	0.50	0.25	0.54	0.31	0.00	0.40	**0.57**
	0.25	0.76	0.51	0.86	0.88	0.80	0.18	**0.96**	0.87
person-cro	0.50	0.70	0.42	0.70	0.66	0.75	0.10	**0.91**	0.78
	0.75	**0.58**	0.12	0.10	0.15	0.42	0.04	0.16	0.13
	0.25	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.99	**1.00**
person-occ	0.50	0.94	0.91	0.87	0.91	**0.95**	0.93	0.91	0.92
	0.75	**0.82**	0.80	0.58	0.80	0.82	0.44	0.80	0.80
	0.25	*0.85*	*0.74*	*0.71*	*0.57*	*0.45*	*0.53*	*0.54*	***0.90***
Average	0.50	*0.75*	*0.63*	*0.56*	*0.50*	*0.39*	*0.34*	*0.52*	***0.79***
	0.75	***0.46***	*0.28*	*0.27*	*0.26*	*0.24*	*0.12*	*0.32*	*0.40*

Table 1. Comparison of our method (last column) with existing methods on 9 video sequences. Recall results are reported for 0.25, 0.50 and 0.75 threshold (τ) values of overlap with the ground truth. The top recall values are highlighted in bold and average values are presented in italic typeface.

4.5 Results and Discussion

Figure 5 shows results of our tracking method obtained using clustering alone, and after incorporating particle filtering and discriminative segments. Improvement in results is visible when particle filtering is added to the simple clustering based tracking. Removing ambiguous segments and keeping discriminative segments only, further improves the tracking accuracy.

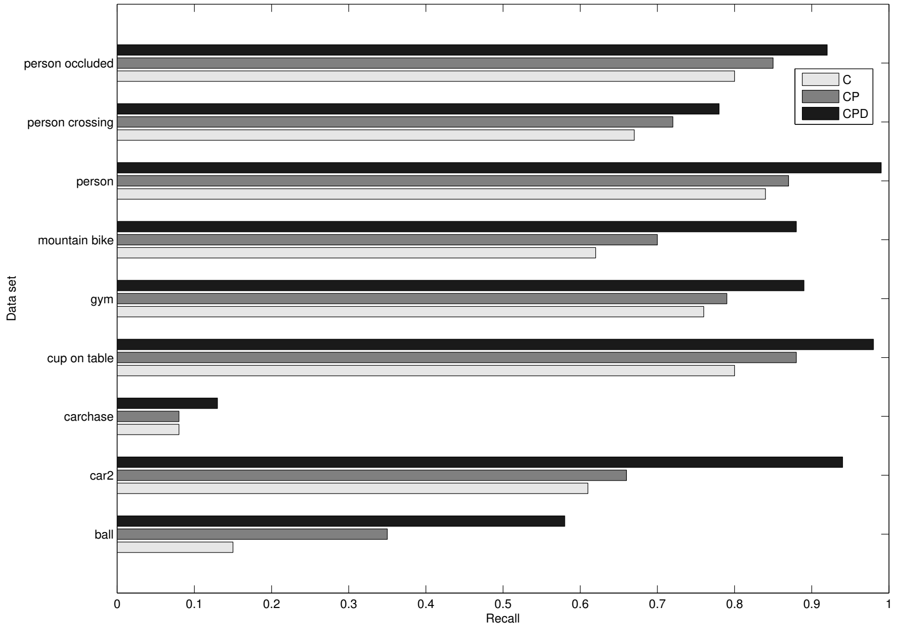

Figure 5. Comparison of our tracking method using clustering (C), and after incorporating particle filtering (CP) and discriminative segments (CPD). Overlap threshold $\tau = 0.50$. Combined method, i.e., clustering with particle filtering and discriminative segments (CPD) achieves the best performance.

Table 1 presents the comparison of our proposed method (TUC) with existing methods. Recall values for seven comparison methods are taken from [20]. Our method attains the highest average value for low and medium accuracy requirements, i.e., when overlap with the ground truth bounding box is greater than or equal to 0.25 and 0.50 threshold (τ) values, respectively. For high accuracy requirement, i.e., when τ is 0.75, our method achieves the second highest average value as CMT gets on the top. This slightly lower performance of our method in this case is attributable to the randomness involved in the method causing atremble movements of the bounding box sometimes. Note that this randomness, on the other hand, helps in keeping track of the object in other scenarios (low and medium accuracy requirements) where other methods show lower performance. After TUC and CMT, the next best results are achieved by STR and TLD.

Eminent performance of our method is clearly observable on sequences staging deformable objects (e.g., gym and person). Taking discriminative and using top 75 % parts of the object that match the reference model helps in achieving these high quality results, particularly for videos having deforming objects. We fixed the parameters for our method, e.g. variance (as described in Section 4.4), for all presented experiments. Adapting these parameters intelligently based on the object and its environment information in a sequence can further improve the overall results, and can make the method more generic in the future.

Figure 6 demonstrates the qualitative results of our method compared with the selected techniques. The figure gives the frame number and each frame shows the tracked object in the boundary from all 228 frames of the mountain-bike sequence.

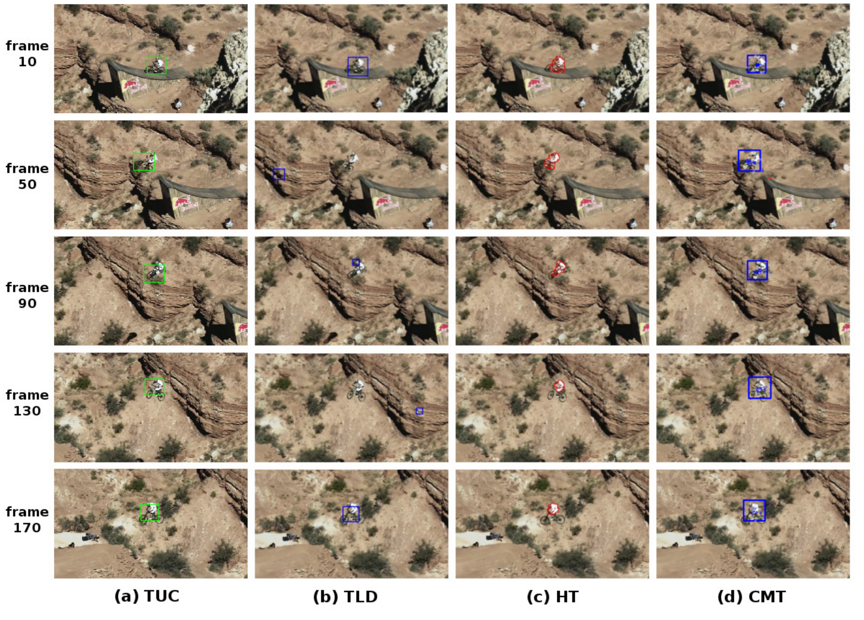

Figure 6. Qualitative results of our proposed tracking method compared with TLD, HT and CMT techniques (mountain-bike data set). Left column gives the frame number.

As discussed earlier, some of the values of control parameters are selected empirically, based on the best combination of correctness and efficient. Table 2 shows the recall values computed while trying different combinations of numbers of clusters and numbers of particles value. It is evident that after a certain number of clusters the segments become too sparse to track. Table 3 shows the recall values computed while trying different combinations of α and β value.

Currently, k-means clustering has been applied for object's segmentation. In the future, other clustering methods (e.g. density based) can be tested. In addition, more features and key-points detection and description methods can be explored to further improve the performance and to handle full occlusions more effectively. The method can also be extended to update the reference model at run-time and to generalize this technique to perform better in all cases. Super-pixel algorithm [1] (i.e. SLIC) can also be used for a fine and quick construction of the segmentation of the target object, as it is faster and more memory efficient. Moreover, some control parameters in this work are selected empirically, what we have considered

	Number of Clusters				
Number of Particles	5	10	15	20	25
100	0.52	0.7	0.92	0.85	0.85
150	0.65	0.79	0.81	0.59	0.97
200	0.6	0.85	0.98	0.82	0.87
250	0.58	0.83	0.94	0.84	0.86
300	0.56	0.68	0.89	0.72	0.83

Table 2. Recall values for combinations of number of clusters and number of particles experimented with the car2 video

	Alpha			
Beta	0.25	0.5	0.75	1
0.25	0.70	0.68	0.85	0.84
0.5	0.73	0.88	0.67	0.80
0.75	0.97	0.83	0.79	0.75
1	0.74	0.80	0.75	0.81

Table 3. Recall values for combinations of alpha and beta value experimented with the car2 video

as sufficient for the scope of this work. An adaptive parameter learning technique can be introduced for the further experimentation and extension of this work.

5 CONCLUSION

In this paper, we have proposed a single object tracking method, Tracking Using Clustering (TUC) by employing data clustering and particle filter. TUC outperforms state-of-the-art tracking methods in deformable object tracking while achieving competitive performance in general. Data clustering is applied to segment the target object into several unstructured parts. To reduce ambiguity, discriminative parts of the object are selected by removing its segments similar to the neighboring background segments. Particle filtering is employed to improve the accuracy and robustness of our method and overcome the lacking caused by the randomness inherited by data clustering methods. Experimental results on nine standard data sets demonstrate the effectiveness of our approach.

Acknowledgments

This work was in part supported by the Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. B0101-15-0525, Development of global multi-target tracking and event prediction techniques based on real-time large-scale video analysis), and by the National Strategic Project-Fine particle of the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (MSIT), and the Ministry of Environment (ME), and the Ministry of Health and Welfare (MOHW) (NRF-2017M3D8A1092022).

REFERENCES

[1] Achanta, R.—Shaji, A.—Smith, K.—Lucchi, A.—Fua, P.—Süsstrunk, S.: SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, 2012, No. 11, pp. 2274–2282, doi: 10.1109/TPAMI.2012.120.

[2] Adam, A.—Rivlin, E.—Shimshoni, I.: Robust Fragments-Based Tracking Using the Integral Histogram. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06), June 2006, Vol. 1, pp. 798–805, doi: 10.1109/CVPR.2006.256.

[3] Cai, Z.—Wen, L.—Lei, Z.—Vasconcelos, N.—Li, S. Z.: Robust Deformable and Occluded Object Tracking with Dynamic Graph. IEEE Transactions on Image Processing, Vol. 23, 2014, No. 12, pp. 5497–5509, doi: 10.1109/TIP.2014.2364919.

[4] Comaniciu, D.—Ramesh, V.—Meer, P.: Real-Time Tracking of Non-Rigid Objects Using Mean Shift. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000), 2000, Vol. 2, pp. 142–149, doi: 10.1109/CVPR.2000.854761.

[5] Du, D.—Qi, H.—Li, W.—Wen, L.—Huang, Q.—Lyu, S.: Online Deformable Object Tracking Based on Structure-Aware Hyper-Graph. IEEE Transactions on Image Processing, Vol. 25, 2016, No. 8, pp. 3572–3584, doi: 10.1109/TIP.2016.2570556.

[6] Godec, M.—Roth, P. M.—Bischof, H.: Hough-Based Tracking of Non-Rigid Objects. Computer Vision and Image Understanding, Vol. 117, 2013, No. 10, pp. 1245–1256, doi: 10.1016/j.cviu.2012.11.005.

[7] Grabner, H.—Leistner, C.—Bischof, H.: Semi-Supervised On-Line Boosting for Robust Tracking. In: Forsyth, D., Torr, P., Zisserman, A. (Eds.): Computer Vision (ECCV 2008). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 5302, 2008, pp. 234–247.

[8] Greminger, M. A.—Nelson, B. J.: Deformable Object Tracking Using the Boundary Element Method. Proceedings of 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003, Vol. 1, pp. I-289–I-294, doi: 10.1109/CVPR.2003.1211366.

[9] Greminger, M. A.—Nelson, B. J.: A Deformable Object Tracking Algorithm Based on the Boundary Element Method That Is Robust to Occlusions and Spurious Edges. International Journal of Computer Vision, Vol. 78, 2008, No. 1, pp. 29–45, doi: 10.1007/s11263-007-0076-6.

[10] Hare, S.—Saffari, A.—Torr, P. H. S.: Struck: Structured Output Tracking with Kernels. 2011 IEEE International Conference on Computer Vision (ICCV), 2011, pp. 263–270, doi: 10.1109/ICCV.2011.6126251.

[11] Hare, S.—Saffari, A.—Torr, P. H. S.: Efficient Online Structured Output Learning for Keypoint-Based Object Tracking. 2012 IEEE Conference on

Computer Vision and Pattern Recognition (CVPR), 2012, pp. 1894–1901, doi: 10.1109/CVPR.2012.6247889.

[12] Hilsmann, A.—Eisert, P.: Deformable Object Tracking Using Optical Flow Constraints. 4th European Conference on Visual Media Production (IETCVMP), 2007, pp. 1–8.

[13] Jackson, J. D.—Yezzi, A. J.—Soatto, S.: Tracking Deformable Moving Objects Under Severe Occlusions. 43rd IEEE Conference on Decision and Control (CDC), 2004, Vol. 3, pp. 2990–2995, doi: 10.1109/CDC.2004.1428922.

[14] Kalal, Z.—Mikolajczyk, K.—Matas, J.: Tracking-Learning-Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, 2012, No. 7, pp. 1409–1422.

[15] Kim, D. Y.—Jeon, M.: Data Fusion of Radar and Image Measurements for Multi-Object Tracking via Kalman Filtering. Information Sciences, Vol. 278, 2014, pp. 641–652, doi: 10.1016/j.ins.2014.03.080.

[16] Kim, D. Y.—Yang, E.—Jeon, M.—Shin, V.: Robust Auxiliary Particle Filter with an Adaptive Appearance Model for Visual Tracking. In: Kimmel, R., Klette, R., Sugimoto, A. (Eds.): Computer Vision (ACCV 2010). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6494, 2010, pp. 718–731.

[17] Kristan, M.—Pflugfelder, R.—Leonardis, A.—Matas, J.—Čehovin, L.—Nebehay, G.—Vojíř, T.—Fernández, G.—Lukežič, A.—Dimitriev, A.—Petrosino, A.—Saffari, A.—Li, B.—Han, B.—Heng, C.—Garcia, C.—Pangeršič, D.—Häger, G.—Khan, F. S.—Oven, F.—Possegger, H.—Bischof, H.—Nam, H.—Zhu, J.—Li, J.—Choi, J. Y.—Choi, J.-W.—Henriques, J. F.—van de Weijer, J.—Batista, J.—Lebeda, K.—Öfjäll, K.—Yi, K. M.—Qin, L.—Wen, L.—Maresca, M. E.—Danelljan, M.—Felsberg, M.—Cheng, M.-M.—Torr, P.—Huang, Q.—Bowden, R.—Hare, S.—Lim, S. Y.—Hong, S.—Liao, S.—Hadfield, S.—Li, S. Z.—Duffner, S.—Golodetz, S.—Mauthner, T.—Vineet, V.—Lin, W.—Li, Y.—Qi, Y.—Lei, Z.—Niu, Z. H.: The Visual Object Tracking VOT2014 Challenge Results. In: Agapito, L., Bronstein, M. M., Rother, C. (Eds.): Computer Vision Workshops (ECCV 2014). Springer International Publishing, Lecture Notes in Computer Science, Vol. 8926, 2015, pp. 191–217.

[18] Li, Y.—Zhu, J.—Hoi, S. C. H.: Reliable Patch Trackers: Robust Visual Tracking by Exploiting Reliable Patches. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015, pp. 353–361, doi: 10.1109/CVPR.2015.7298632.

[19] Lucas, B. D.—Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81), Vol. 2, San Francisco, CA, USA, 1981, Morgan Kaufmann Publishers Inc., pp. 674–679.

[20] Nebehay, G.—Pflugfelder, R.: Consensus-Based Matching and Tracking of Keypoints for Object Tracking. 2014 IEEE Winter Conference on Applications of Computer Vision (WACV), 2014, pp. 862–869, doi: 10.1109/WACV.2014.6836013.

[21] Nebehay, G.—Pflugfelder, R.: Clustering of Static-Adaptive Correspondences for Deformable Object Tracking. Proceedings of the 2015 IEEE Conference on

Computer Vision and Pattern Recognition (CVPR), 2015, pp. 2784–2791, doi: 10.1109/CVPR.2015.7298895.

[22] Nguyen, H. T.—Smeulders, A. W. M.: Fast Occluded Object Tracking by a Robust Appearance Filter. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, 2004, No. 8, pp. 1099–1104.

[23] Oron, S.—Bar-Hillel, A.—Avidan, S.: Extended Lucas-Kanade Tracking. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (Eds.): Computer Vision (ECCV 2014). Springer International Publishing, Lecture Notes in Computer Science, Vol. 8693, 2014, pp. 142–156.

[24] Oron, S.—Bar-Hillel, A.—Levi, D.—Avidan, S.: Locally Orderless Tracking. International Journal of Computer Vision, Vol. 111, 2015, No. 2, pp. 213–228.

[25] Ristic, B.—Hernandez, M. L.: Tracking Systems. Radar Conference (RADAR '08), IEEE, 2008, pp. 1–2.

[26] Smeulders, A. W. M.—Chu, D. M.—Cucchiara, R.—Calderara, S.—Dehghan, A.—Shah, M.: Visual Tracking: An Experimental Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 36, 2014, No. 7, pp. 1442–1468.

[27] Staffa, M.—Rossi, S.—Giordano, M.—De Gregorio, M.—Siciliano, B.: Segmentation Performance in Tracking Deformable Objects via WNNS. 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 2462–2467, doi: 10.1109/ICRA.2015.7139528.

[28] Yilmaz, A.—Javed, O.—Shah, M.: Object Tracking: A Survey. ACM Computing Surveys (CSUR), Vol. 38, 2006, No. 4, Article No. 13, doi: 10.1145/1177352.1177355.

[29] Zhang, L.—van der Maaten, L.: Structure Preserving Object Tracking. 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 1838–1845, doi: 10.1109/CVPR.2013.240.

[30] Zhang, L.—van der Maaten, L.: Preserving Structure in Model-Free Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 36, 2014, No. 4, pp. 756–769.

[31] Zhang, M.—Kang, B.: An Improved Method of Tracking and Counting Moving Objects Using Graph Cuts. In: Wong, W. E. (Ed.): Proceedings of the 4^{th} International Conference on Computer Engineering and Networks. Springer International Publishing, Lecture Notes in Electrical Engineering, Vol. 355, 2015, pp. 583–590.

[32] Zhang, X.—Hu, W.—Chen, S.—Maybank, S.: Graph-Embedding-Based Learning for Robust Object Tracking. IEEE Transactions on Industrial Electronics, Vol. 61, 2014, No. 2, pp. 1072–1084.

Muhammad Aasim Rafique received his M.Sc. degree in computer science from Quaid-e-Azam University, Islamabad, Pakistan. He then received his M.Sc. degree in computer science from Lahore University of Management and Sciences, Lahore, Pakistan in 2008. He received his Ph.D. degree from School of Electrical Engineering and Computer Sciences, GIST, Gwangju, Republic of Korea, in 2018 (when submitted this article he was Ph.D. student at GIST). He is now working as Assistant Professor at Quaid-e-Azam University, Islamabad, Pakistan. His research interests are artificial neural networks, their application in machine learning and computer vision.

Moongu Jeon received his B.Sc. degree in architectural engineering from the Korea University, Seoul, Korea, in 1988 and his M.Sc. and Ph.D. degrees in computer science and scientific computation from the University of Minnesota, Minneapolis, MN, USA, in 1999 and 2001, respectively. As Postgraduate Researcher, he worked on optimal control problems at the University of California at Santa Barbara, Santa Barbara, CA, USA, in 2001–2003, and then moved to the National Research Council of Canada, where he worked on the sparse representation of high-dimensional data and the level set methods for image processing until July 2005. In 2005, he joined the Gwangju Institute of Science and Technology, Gwangju, Korea, where he is currently Full Professor at the School of Electrical Engineering and Computer Science. His current research interests are in machine learning, computer vision, and intelligent transportation systems.

Malik Tahir Hassan received his M.Sc. and Ph.D. degrees in computer science from Lahore University of Management Sciences (LUMS), Lahore, Pakistan. He worked at Gwangju Institute of Science and Tehcnology (GIST), Gwangju, South Korea, as a Post-Doc fellow. Currently, he is working as Assistant Professor at School of Systems and Technology (SST) at University of Management and Technology (UMT), Lahore, Pakistan. His research interests include pattern recognition, text mining, recommender systems and autonomic computing.