Journal of AL-Qadisiyah for computer science and mathematics
Vol.9 No.1 Year 2017

Abnormality Detection using K-means Data Stream Clustering Algorithm in Intelligent Surveillance System

Abdulamir A. Karim

Department of Computer Science
University of Technolog
Ameer_aldelphi@yahoo.com

Narjis Mezaal Shati

Department of Computer Science
University of Al-mustansiryah
College of of Science
nareen_manar@yahoo.com

Recived : 12\3\2017 **Revised : //** **Accepted : 22\3\2017**

Abstract

In this research work a k-Means clustering technique utilized in a new data stream clustering method used in abnormal detection system. This system implies the use of a set of features (such as: distance, direction, x-coordinate, y-coordinate) extracted from set of pairs of interest point that obtained using HARRIS or FAST detector from the frames of video clips in two publically available datasets, the first UCSD pedestrian dataset (ped1 and ped2 datasets), and the second VIRAT video dataset. The results indicated that using HARRIS detector achieved detection rate 1% with 6% false alarms by using UCSD (Ped1) dataset, 10.75% detection Rate with 10% false alarm rate by using UCSD (Ped2) dataset, and 5% detection rate with 40% false alarms by using VIRAT dataset. While for FAST detector, the achieved detection rates are 0.5%, 10.75%, and 4.08% while the false alarm rates are 5%, 10.50%, and 45.92% by using UCSD (Ped1), UCSD (Ped2), and VIRAT datasets respectively.

Keywords: abnormal behavior detection; anomaly behavior detection; crowd scene; video analyses; intelligent surveillance system

Journal of AL-Qadisiyah for computer science and mathematics
Vol.9 No.1 Year 2017

1. Introduction

The paper addresses the problem of ensuring security so a lot of protection means have emerged. One of the most important security systems is video surveillance system. As the society security demands are increased along time, so that the number of the security cameras does. Such increasing leads to a lot of tedious and drudge labors for the human operators who take care for monitoring this increased number of the cameras. This led to need to a robust suspicious behavior detection for video surveillance systems because current surveillance systems cannot effectively prevent crime. One of the constraint is that most of the current systems rely heavily on human operators who have physical limitations in the form of fatigue, or loss of concentration when monitoring multiple screens for long periods of time. These limitations could be alleviated by enhancing these systems to automatically flag any suspicious behavior that may occur in a scene. Such capability when added to the current systems requires a mechanism to not only capture but also understand human motion and behavior. Capturing and understanding human motion and behavior is still a challenging problem [1, 2].

Computer vision areas including intelligent surveillance system has been investigated in a variety of application domains. These application domains include public or private areas for the sake of security and safety such as banks, government premises, railway stations, and house [3].

Some presented researches involves using of a number of image processing and computer vision methods [4, 5, 6] to detect individual behavior other using statistical method to detect crowed action [1, 7, 8, 9].

2. Abnormal Behavior Detection

Behavior can be analyzed on two different levels: the individual behavior, and the crowd behavior. The first is trying to detect the actions of an individual human, and the second is using statistical methods to detect "crowd actions". Recognizing individual behavior requires detailed information about the people present in the scene. This leads to the following steps (Motion Detection, Object Classification, Tracking, Behavior Analysis) [10].

The reason for attempting to analyze crowd behavior instead of individual is that it can be very difficult to get useful information when trying to recognize the actions of individuals in a crowded scene. This is because segmentation of individual objects become difficult due to frequent occlusions (people partially or completely covering other people in the scene). The motion detection step mentioned above would produce large blobs containing many of the people in the scene, which makes analysis of individual behavior impossible. Analyzing the motion of the crowd as a whole is often enough to determine that something unusual is going on, by detecting a disturbance in the typical flow based on motion flow detection [10], as shown in figure (1), or by using suspicious behavior detection. Suspicious behavior is defined as the behavior, which is only considered as suspicious [11].

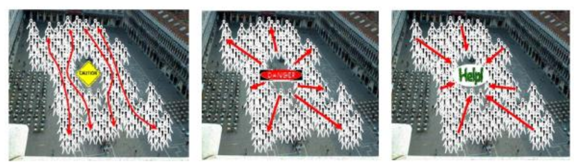


Figure (1): Extracting security information from observing crowd motion [10]. (a) Something is unusual (b) Something is dangerous (c) Something is attracting a crowd

2.1 Human Behavior Features

Human behavior features are divided into two major parts: Human global motion and Human local motion features.

Human Global Motion features utilize the information on a subject location at various times (tracking information). It is useful when human limb movements are not observable. The information extracted from human trajectories, such as: positional information of human trajectories (Commonly occupied region), human trajectory shape (Human walking path shape) [12].

Human Local Motion features are useful when human limb movements can be discerned in video feeds. Based on the degree of not-rigidity of the objects, human motion are classified as rigid or not rigid motions [13].

Human motion approaches classified basing on with/ without prior knowledge about the object shape as follows: Model based approaches (using prior knowledge about the shape of an objects) [14], and Appearance based approaches (Building body representation in a bottom-up fashion by first detecting appropriate features in a sequence of frames). Appearance based approaches do not require a specific object model but are sensitive to noise. These approaches are classified into three types: Flow based approaches, Spatial-temporal shape template base approach, and Interest point based approaches [15, 16].

The interest points extracted by interest point detectors. It is extension of key point's concept for object detection in images. One of the advantage of using interest points is that they can be used to search an action contained in the short query video over a large resolution without using background subtraction and tracking of the object [16, 17]. In this paper interest point features are used HARRIS or FAST detector, for more information about these detectors see [18].

2.2 Data Stream Clustering and K-Means Clustering

A data stream model is defined as an ordered sequence of points $x_1, \ldots, x_n$ where ($n \approx \infty$). The sequence has to be read in order and once or a small number of times [19].

The data stream model requires decisions to be made before all the data becomes available. This model is similar to online models. So these models need algorithm (data stream algorithm). These algorithms are allowed to take action after a group of points arrives [19].

K-Means clustering is a partitional clustering approach. Each cluster is represented by one prototype object, and a new data sample is assigned to the nearest prototype and therefore to that cluster, for more information see [20].

Data stream and online clustering approaches are similar in that both of them require decisions made before all data are available. But these models are not identical because online algorithm can access the first i data point (with its previous i decisions) when reacting to the $(i+1)^{th}$ point, the amount of memory available to data stream algorithm is bounded by a function of the input size (sublinear function used). In addition, a data stream approach do not need to take action after the arrival of each point (after a group of points) [11].

Data stream clustering approach stores and processes large-scale data efficiently because it provides summarizations of the past data, see [21, 22, 23, 24, 25, and 26] for more information. In this work, k-mean clustering algorithm is utilized to form new data stream algorithm.

3. Design of K-means Data Stream Clustering Algorithm in Intelligent Surveillance Systems

Anomaly behavior detection systems utilizing k-means algorithm in data stream clustering algorithm are created were trained and applied to the same datasets. Their performance were measured using (Detection Rate (DR), False Alarm Rate (FAR), Recall (R), Precision (P), The Coverage Test (CT)), for more detail about these measurements see [27, 28].

3.1 The Proposed Systems Layouts

The diagram for the proposed anomaly detection system is shown in figure (2).

3.2 Video Dataset

The entire set of the selected video dataset is divided into two sets of videos: (i) a training video set is used for the developments of a classifier, and (ii) a test video set contains videos used to measure the anomaly behavior detection performance of the classifier. The selected video datasets are two publicly available datasets. The first is the UCSD dataset; which contains two pedestrians' dataset pedestran1 (UCSD ped1) and pedestran2 (UCSD ped2); and the second is VARAT dataset.

3.3 Preprocessing

Preprocessing steps are implemented to make the data of both training and test sets more suitable and easier to analyses. Converting to gray preprocessing applied on both training and test datasets. In this step, the frames of video dataset are converted to gray images.

3.4 Feature Generation

Context information (interest point information) such as interest point pairs and (distance, direction, x-coordinate and y-coordinate) of these pairs are extracted from the training and test datasets. This done in two phases: feature detection and feature matching.

3.4.1 Feature Detection and Matching

By utilizing interest point detector algorithms (HARRIS, FAST) to extract interest points from the current frame and previous frame (according to previous frame threshold PFT) and then matching these interest points are from both frames to obtain list of pairs of interest points. Algorithms (1 and 2) show the implemented steps for utilizing the HARRIS, and FAST detectors respectively. After construction of interest point pairs list, an extraction process will be done to estimate list of features (distance, direction, x-coordinate, and y-coordinate) from the corresponding pairs, see figure (3). This illustrated in algorithm (3).

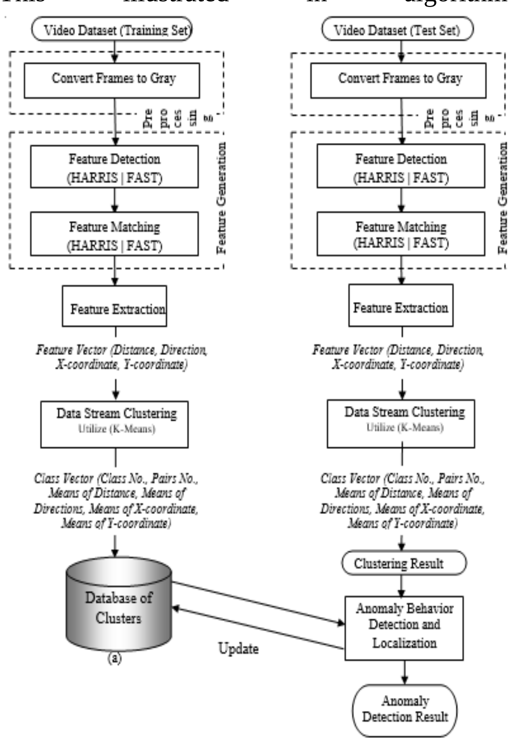


Figure (2) Diagram of the proposed systems: (a) Enrolment phase (b) Anomaly detection and localization phase

Algorithm (1)// Detection and matching interest points using HARRIS

Input: Frames_List, PFT (Previous Frame Threshold), No. of Frames

Output: LOP (List of interest point pairs)

```
For n = 0 to Frame_List.count
    Detect interest point using HARRIS corner:
    Detector (Frames_List [n], Interest_Points_List [n])
    If n > PFT then
        Matching (Frames_List [n], Frames_List [n-PFT], Interest_Points_List [n], Interest_Points_List [n-PFT]), LOP)
        For i = 0 to LOP.count
            Compute (LOP[i].distance) using Euclidean distance between the two points of the pair (LOP[i])
            If LOP[i].distance < Low_Threshold or LOP[i].distance > Max_Threshold
            Then Remove (LOP[i])
            End If
        End For
    End If
End For
End.
```

Algorithm (2)// Detection and matching interest points using FAST

Input: Frames_List, PFT (Previous Frame Threshold), No. of Frames

Output: LOP (List of interest point pairs)

For n = 0 **to** Frame_List.count

Detect interest point using FAST corner: Detector (Frames_List [n], Interest_Points_List [n])

If n > PFT **then**

Matching (Frames_List [n], Frames_List [n-PFT], Interest_Points_List [n], Interest_Points_List [n-PFT]), LOP)

For i = 0 **to** LOP.count

Compute (LOP[i].distance) using Euclidean distance between the two points of the pair (LOP[i])

If LOP[i].distance < Low_Threshold or LOP[i].distance > Max_Threshold

Then Remove (LOP[i])

End If

End For

End If

End For

End.

The distance is computed by using Euclidean distance between the two points (P1(x1,y1) and P2(x2,y2)) of each pair in list of pairs. Then calculate direction by finding the angle of inclination, as shown in the following equation and illustrated in algorithm (4).

$$Dir = \begin{cases} 0 & if\ (x_1 - x_2 > 0)\ and\ (y_1 - y_2 = 0) \\ 90 & if\ (x_1 - x_2 = 0)\ and\ (y_1 - y_2 > 0) \\ 180 & if\ (x_1 - x_2 < 0)\ and\ (y_1 - y_2 = 0) \\ 270 & if\ (x_1 - x_2 = 0)\ and\ (y_1 - y_2 < 0) \\ \tan^{-1}(y_1 - y_2)/(x_1 - x_2) & Otherwize \end{cases} \quad ,\ldots(1)$$

After that, compute the x-coordinate and y-coordinate for the pair by finding the mid-point between the first and the second point of each pair, as show in the following equations.

$$x = (x_1 + x_2)\ /\ 2 \quad ,\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

$$y = (y_1 + y_2)\ /\ 2 \quad ,\ldots\ldots\ldots\ldots\ldots\ldots(3)$$

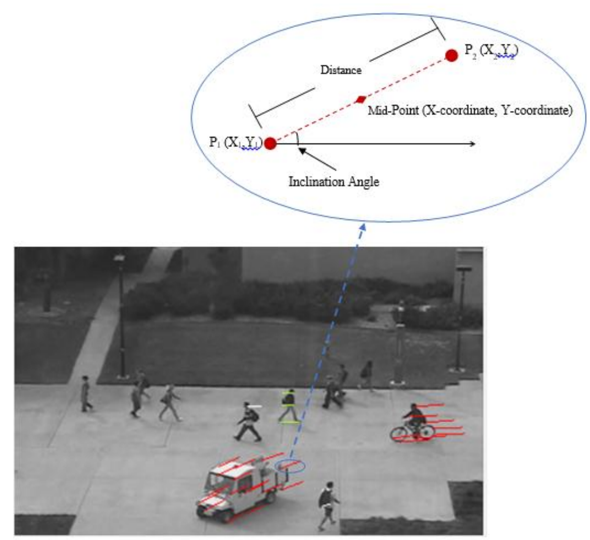


Figure (3) Illustration of distance, inclination angle, x-coordinate, and y-coordinate for two points (P_1 and P_2)

Algorithm (3)// Compute distance, direction, x-coordinate, y-coordinate from interest point pairs

Input: List of interest point pairs (LOP)

Output: LOF (List of feature vector (distance, direction, x-coordinate, y-coordinate))

For i=0 **to** LOP.count

Compute (LOF[i].dis) using Euclidean distance between $P_1(x_1, y_1)$ and P_2 (x_2, y_2) in one pair

Compute (Direction) using algorithm (4),

Compute (x-coordinate, y-coordinate),

Set LOF[i].x ← (LOP[i,1].x + LOP[i,2].x)/2

Set LOF[i].y ← (LOP[i,1].y + LOP[i,2].y)/2

End For

End.

Algorithm (4)// Compute direction

Input: two interest point P_1 (x_1, y_1) & P_2 (x_2, y_2)

Output: D (Direction)

If $(x_1 - x_2) = 0$ **then**

If $(y_1 - y_2) > 0$ **then**

Set D ←90

Else If $(y_1 - y_2) < 0$ **then**

Set D ←270

Else If $(y_1 - y_2) = 0$ **then**

If $(x_1 - x_2) > 0$ **then**

Set D ← 0

Else

Set D ←180

Else

Set D ← $\tan^{-1}(y_1 - y_2)/(x_1 - x_2)$

End If

End.

3.5 Proposed Data Stream Clustering

In this paper, a data stream model is defined as four features of each pairs of interest points, which are obtained from the previous section. The proposed algorithm is used to make decision after a group of these features arrived and then these decisions (past data of clusters) are summarized and organized in a database; that gives a summarization of all clusters in limited memory size.

In this paper K-Means Clustering technique utilized in data stream clustering algorithm, this illustrated in algorithm (5). First of all read the training (in training phase) or testing (in test phase) video clips from datasets to obtain list of video frames and number of frames, and then convert it to gray. Apply interest point detector (HARRIS or FAST) on the gray frames to extract the list of interest pairs by using (1 or 2) algorithms. After that extract list of interest point features such as: distance, direction, x-coordinate, y-coordinate, using (3) algorithm. The next step is clustering these extracted lists of interest point features by utilizing k-means clustering algorithm in data stream clustering stage, this is done based on distance, see algorithm (6), or direction, as in algorithm (12), or coordinates using algorithm (15). Cluster database is created if the clustering process is done in enrolment phase, as shown in algorithm (18), otherwise it will be anomaly detection and localization phase, so the anomaly event and its location are detected using the previously created database.

Algorithm (5)// First Proposed Data Stream Clustering System
Input: Video data set (Train or Test) **Output:** Database_Clusters (database of clusters)

Step$_1$: Read Video
Step$_2$: Convert color frame to gray frame using equation (3.1)
Step$_3$: Detection and matching interest points using (HARRIS | FAST) by using algorithms (1, 2) respectively
Step$_4$: Compute distance, direction, x-coordinate, y-coordinate from interest point pairs using algorithm (3)
Step$_5$: Clustering list of features using K-Means by algorithms (6|12|15)
Step$_6$: **If** training phase **then**
Create Database_Clusters using algorithm (18)
Else
Detection and localization anomaly behavior using algorithm (19)
End If
End.

In algorithm (6), K-Means clustering algorithm employed in data stream clustering algorithm based on distance. Initial K centroids will be chosen from the list of pairs features. For all pairs compute the closest centroid according to distance by calculating the absolute difference between the pair distance and the centroid distance, and then choose the centroid having minimum distance to have this pair, as seen in algorithm (7).

Algorithm (6)// Clustering list of features using K-Means based on Distance
Input: LOF **Output:** LOF, Centroids_list
Step$_1$: Choose initial k centroids randomly from LOF named Centroids_list **Step$_2$:** For each pair determine which centroid (Centroids_list) closest to it using algorithm (7). **Step$_3$:** Eliminate pairs from class according to distance direction, or coordinate using algorithms (8, 9, and 10) **Step$_4$:** Compute new centroids of distance using algorithm (11) **Step$_5$:** Compare new centroids of distance with previous centroids; if it changed, go to step$_2$. **End.**

Algorithm (7)// Determine closest centroid of distance to each pair
Input: LOF, Centroids_list **Output:** LOF
For i = 0 **to** LOF.count **For** j = 0 **to** Centroids_list.count Compute (Dic_dis[j]) using absolute difference between (LOF[i].dis, Centroids_list [j].dis) **End For** Put in (LOF[i].cluster) the index of minimum value in (Dic_dis) **End For** **End.**

After that, each pair in the clusters will be examined to eliminate each pair which has distance, direction, and coordinate more than distance threshold, direction threshold, and coordinate threshold, as illustrated in algorithms (8, 9, and 10).

Algorithm (8)// Eliminate pairs according to distance
Input: LOF, Centroids_list, E_dis_th (threshold value of distance using in elimination process) **Output:** LOF, Centroids_list
For each pair in LOF **Set** C ← pair.cluster Compute (Dic_dis using absolute difference between (pair, Centroids_list.[c]). **If** Dis_dis > E_dis_th **then Set** pair.cluster ← 0 **End for** **End.**

Algorithm (9)// Eliminate pairs according to direction
Input: LOF, Centroids_list, E_dir_th (threshold value of direction using in elimination process) **Output:** LOF, Centroids_list
For each pair in LOF **Set** C ← pair.cluster Compute (Dic_dir using absolute difference between (pair, Centroids_list.[c]). **If** Dis_dir > E_dir_th **then Set** pair.cluster ← 0 **End for** **End.**

Algorithm (10)// Eliminate pairs according to coordinate
Input: LOF, Centroids_list, E_coord_th (threshold value of coordinates using in elimination process) **Output:** LOF, Centroids_list
For each pair in LOF **Set** C ← pair.cluster Compute (Dic_c using Euclidean distance between (pair, Centroids_list.[c]) **If** Dis_c > E_coord_th **then Set** pair.cluster ← 0 **End for** **End.**

Next step, new centroids based on distance will be computed by taking the average of distance for all pairs in the cluster by using algorithm (11).

Algorithm (11)// Compute new centroid of distance
Input: LOF **Output:** Centroids_list (list of K centroid)

For i = 0 **to** LOF.count
    Set Cluster ← LOF[i].cluster
    Set Centroids_list [cluster].dis←Centroids_list [cluster].dis + LOF[i].dis
    Set Counter ← counter+1
End For
For i=0 **to** Centroids_list.count
    Set Centroids_list [j].dis ← Centroids_list [j].dis/ counter
End For
End.

Finally, compare the new centroids with old ones, in case there is a change in it, then return to computing closest centroids for each pair again, and so on until no change is found. Figure (4) illustrates the process of the first proposed data stream clustering algorithm based on distance.

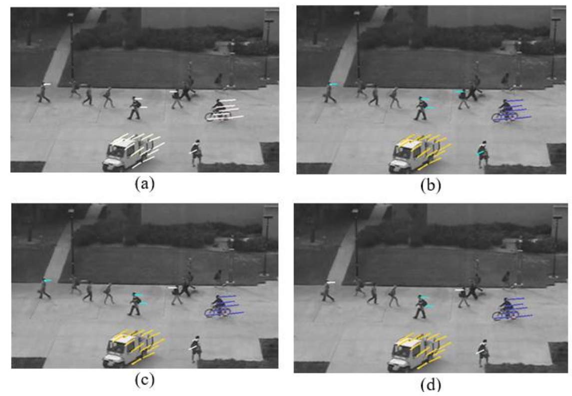


Figure (4): Illustrates proposed data stream clustering based on distance: (a) Samples frame with interest pairs, (b) K-means clustering for distance using k=3, (c) Eliminate each cluster according direction, (d) Eliminate each cluster according coordinates.

In algorithm (12), also k-means clustering algorithm is utilized in data stream clustering based on direction. First of all, K centroids will be chosen initially from the list of pairs features. Then the closest center for all pairs is determined based on direction, as shown in algorithm (13), by computing the absolute difference between the pair direction and the centroid direction in order to elect the centroid that has the minimum absolute difference to implicate in that centroid class.

Algorithm (12)// Clustering list of features using K-Means based on Direction

Input: LOF
Output: LOF, Centroids_list

Step$_1$: Choose initial k centroids randomly from LOF named Centroids_list
Step$_2$: For each pair determine which centroid (Centroids_list) closest to it using algorithm (13).
Step$_3$: Eliminate pairs from class according to distance, direction, or coordinate using algorithms (8, 9, and 10)
Step$_4$: Compute new centroids of direction using algorithm (14)
Step$_5$: Compare new centroids of direction with previous centroids, if it changed go to step$_2$.
End.

Algorithm (13)// Determine closest centroid of direction to each pair

Input: LOF, Centroids_list

Output: LOF

For i = 0 **to** LOF.count

  For j = 0 **to** Centroids_list.count

    Compute (Dic_dir[j] using absolute difference between (LOF[i].dir, Centroids_list [j].dir).

  End For

  Put in (LOF[i].cluster) the index of minimum value in (Dic_dir)

End For

End.

Then remove each pair having; distance, direction, and coordinates more than distance threshold, direction threshold, and coordinate threshold. This removing is illustrated in algorithm (8, 9, and 10). After that, the average of direction for all pairs owned by the cluster will be calculated to obtain new direction centroids, as illustrated in algorithm (14). Finally, a comparison between the new centroids and the previous ones will be done, if there is a change then the computation of the cluster centroids will be done again for each pair and so, on until no change is found. These processes shown in figure (5).

Algorithm (14)// Compute new centroid of direction

Input: LOF

Output: Centroids_list (list of K centroid)

For i = 0 **to** LOF.count

  Set Cluster ← LOF[i].cluster

  Set Centroids_list [cluster].dir←Centroids_list [cluster].dir + LOF[i].dir

  Set Counter ← counter + 1

End For

For i=0 **to** Centroids_list.count

  Set Centroids_list [j].dir ← Centroids_list [j].dir/ counter

End For

End.

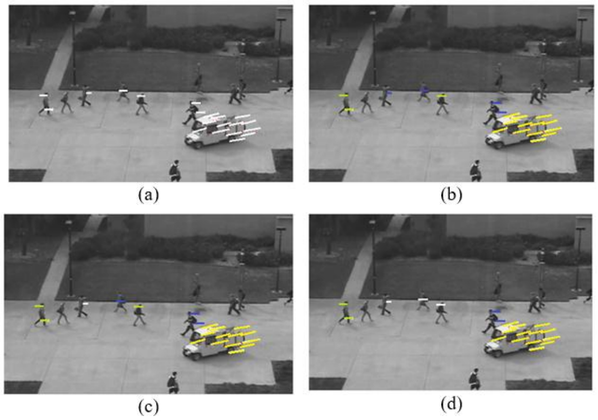


Figure (5): Illustrates proposed data stream clustering based on direction: (a) Sample frame with interest pairs, (b) K-means clustering for direction using k=3, (c) Eliminate each cluster according distance, (d) Eliminate each cluster according to coordinates.

In algorithm (15), k-means clustering algorithm is employed in data stream clustering based on coordinate. Start by choosing initial k centroids from list of pair features, then calculate the closest centroids based on coordinate for all pairs by using algorithm (16), where computing the Euclidian distance between the pair coordinate and all centroids coordinate, and then the centroids that own the closest distance will be owned this pair.

Algorithm (15)// Clustering list of features using K-means based on coordinate

Input: LOF
Output: LOF, Centroids_list

Step$_1$: Choose initial K centroids randomly from LOF named Centroids_list
Step$_2$: For each pair determine which centroid (Centroids_list) closest to it using algorithm (16).
Step$_3$: Eliminate pairs from class according to distance, direction, or coordinate using algorithms (8, 9, and 10)
Step$_4$: Compute new centroids of coordinate using algorithm (17)
Step$_5$: Compare new centroids of coordinate with previous centroids, if it changed go to step$_2$.
End.

Algorithm (16)// Determine closest centroid of coordinate to each pair

Input: LOF, Centroids_list

Output: LOF

For i = 0 **to** LOF.count

For j = 0 **to** Centroids_list.count

Compute (Dic_c[j] using Euclidean distance betwee (LOF[i].coordinate, Centroids_list [j].coordinate)

End For

Put in (LOF[i].cluster) the index of minimum value in (Dic_c)

End For

End.

For the next step, eliminate each pair that has distance, direction, coordinates more than distance threshold, direction threshold, and coordinate threshold, as shown in algorithms (8, 9, and 10).

After the elimination, a new coordinate centroids will be computed which represents the average of coordinates for all pairs in this cluster, as shown in algorithm (17).

Algorithm (17)// Compute new centroid of coordinate

Input: LOF

Output: Centroids_list (list of K centroid)

For i = 0 **to** LOF.count

Set Cluster ← LOF[i].cluster

Set Centroids_list [cluster].x ← Centroids_list [cluster].x + LOF[i].x

Set Centroids_list [cluster].y ← Centroids_list [cluster].y + LOF[i].y

Set Counter ← counter + 1

End For

For i=0 **to** Centroids_list.count

Set Centroids_list [j].x ← Centroids_list [j].x/ counter

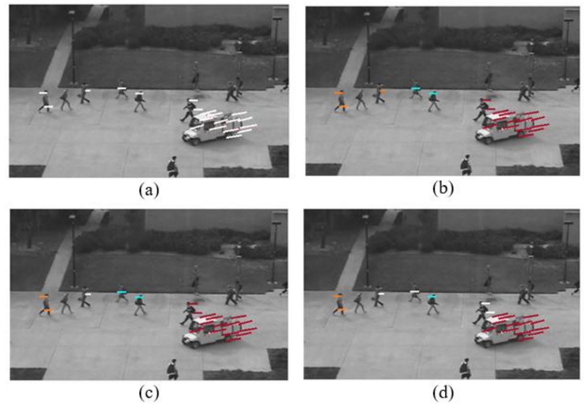
Set Centroids_list [j].y ← Centroids_list [j].y/ counter

End For

End.

Finally compare the new centroids with the previous ones, if there is a change go again to compute closest centroids for each pair and so on until no change happened. This is illustrated in figure (6).

In training phase the database of normal clusters will be generated through searching the cluster-database to add new normal cluster if it is not found in database or increase frequency if it is found, as illustrated in (18) algorithm.



(a) (b) (c) (d)

Figure (6): Illustrates proposed data stream clustering based on coordinates: (a) Sample frame with interest pairs, (b) K-means clustering for direction using k=3, (c) Eliminate each cluster according distance, (d) Eliminate each cluster according to direction.

Algorithm (18)// Training

Input: Centroids_list

Output: Database_Clusters (database of clusters)

For i = 0 **to** Centroids_list.count

For each record in Database_Clusters

If record is equal to Centroids_list [i] according threshold

Then

Set Record. Frequnce ← Record. Frequnce+1

Else

Database_Clusters.add (Centroids_list [i])

End If

End For

End For

End.

Journal of AL-Qadisiyah for computer science and mathematics
Vol.9 No.1 Year 2017

3.6 Anomaly Behavior Detection and Localization

After completing the construction of cluster database from passing the training video dataset, it will be used in anomaly behavior detection in order to detect anomaly behavior in test video dataset and localize its position; algorithm (19) describes the anomaly behavior detection and localization. In this algorithm, a search process will be done between the class obtained from the proposed data stream clustering algorithm in test phase and the database of clusters that is obtained from the training phase to find if this class is in the database. So it will be detected as anomaly event and its location in the frame is marked by using the coordinate from the clustering data stream process. Figure (7) shows anomaly detection in some frames.

Algorithm (19)// Anomaly behavior detection and localization

Input: Test video dataset, Database_Clusters (database of clusters)

Output: Anomaly_behavior_location

For i=0 **to** centroids_list.count

If record in Database_Clusters where record equal to centroids_list[i] **then**

Set Record.Frequnce ← Record. Frequnce +1

Else

Mark the position of anomaly behavior

End If

End For

End.

Figure (7) Sample of frames illustrating anomaly detection

4. Test Result

The proposed approach is tested on two datasets. The UCSD pedestrian dataset it contains video sequences from two pedestrian (Ped1 and Ped2) walkways where abnormal events occur. The dataset contains different crowd densities, and the anomalous patterns are the presence of non-pedestrians on a walkway (bicyclists, skaters, small carts, and people in wheelchairs). The second dataset is VIRAT Video Dataset was designed by Kitware Company to be more realistic, natural and challenging for video surveillance domains than existing action recognition datasets in terms of its resolution, background clutter, diversity in scenes, and human activity/event categories. Data are collected in natural scenes showing people performing normal actions in standard contexts, with uncontrolled and cluttered backgrounds. The key characteristics of these datasets are summarized in table (1).

Table (2) presents the values of the used data stream clustering parameters. The experiment was repeated 15 times, and the anomaly detection test results of UCSD (Ped1) video clips sets using features extracted from the interest point pairs as an input to the anomaly behavior detection system in both method of feature extraction using HARRIS or FAST detectors are shown in table (3).

The detection test results by using UCSD (ped2) dataset are presented in table (4), and the results of VIRAT dataset are presented in table (5).

Table (1) Key characteristics of training and testing video clips in datasets

			UCSD (Ped1)	UCSD (Ped2)	VIRAT
Entire dataset	training	Number of video clips	34	16	6
		Frame Dimension	238×158	360×240	426×240
		Frame rate	15 frames/second	15 frames/second	15 frames/second
	test	Number of video clips	36	12	8
		Frame Dimension	238×158	360×240	426×240
		Frame rate	15 frames/second	15 frames/second	15 frames/second
Anomaly Behavio Event	test	Number of anomaly events	1460	987	335

Table (2): The proposed data stream clustering parameters

Parameter	**Value**
Previous frame threshold (PFT)	3
Distance threshold	2
Direction threshold	1
Coordinate threshold	25
No of Cluster K	3

Table (3): Results of the anomaly behavior detection system by using UCSD (Ped1) dataset (PFT=3; Distance threshold=2; Direction threshold=1; Coordinate threshold=25; k=3; Repetition=15)

	Expt.	Training Time (ms/ frame)	Testing Time (ms/ frame)	DR (%)	FAR (%)	R	P	CT
HARRIS	Min	1.76	1.81	0.75	5.50	0.01	0.10	0.01
	Max	1.76	2.15	1.00	5.00	0.01	0.17	0.02
	Average	1.76	2.25	1.00	6.00	0.01	0.14	0.02
FAST	Min	2.12	3.12	0.25	3.75	0.00	0.06	0.00
	Max	2.12	2.85	0.75	6.00	0.01	0.11	0.01
	Average	2.12	3.04	0.50	5.00	0.00	0.04	0.01

Table (4): Results of the anomaly behavior detection system by using UCSD (Ped2) dataset (PFT=3; Distance threshold=2; Direction threshold=1; Coordinate threshold=25; k=3; Repetition=15)

	Expt.	Training Time (ms/ frame)	Testing Time (ms/ frame)	DR (%)	FAR (%)	R	P	CT
HARRIS	Min	1.99	2.01	10.50	10.50	0.11	0.50	0.17
	Max	1.99	2.11	11.00	10.25	0.11	0.52	0.18
	Average	1.99	2.02	10.75	10.00	0.11	0.52	0.18
FAST	Min	2.13	3.20	10.00	10.00	0.10	0.50	0.17
	Max	2.13	2.94	10.76	10.10	0.11	0.52	0.18
	Average	2.13	2.95	10.75	10.50	0.11	0.51	0.18

Table (5) Results of the first anomaly behavior detection system by using VIRAT dataset (PFT=3; Distance threshold=2; Direction threshold=1; Coordinate threshold=25; k=3; Repetition=15)

	Expt.	Training Time (ms/ frame)	Testing Time (ms/ frame)	DR (%)	FAR (%)	R	P	CT
HARRIS	Min	2.98	3.52	4.00	45.00	0.04	0.08	0.05
	Max	2.98	3.36	6.00	43.00	0.06	0.12	0.08
	Average	2.98	3.54	5.00	40.00	0.05	0.11	0.07
FAST	Min	2.87	3.80	3.06	51.02	0.03	0.06	0.04
	Max	2.87	3.16	5.10	48.98	0.05	0.09	0.07
	Average	2.87	3.42	4.08	45.92	0.04	0.08	0.05

In these group of tests, the results indicate that the DRs= (1%, 10.75%, 5%) with FARs= (6%, 10%, 40%) in using HARRIS detector. For FAST detector the results obtained are (0.50%, 10.75%, 4.08%) as Detection rates with (5%, 10.50%, 45.92%) as a false alarms for the three dataset respectively. And the effects of R,P,CT are shown in figures (8, 9).

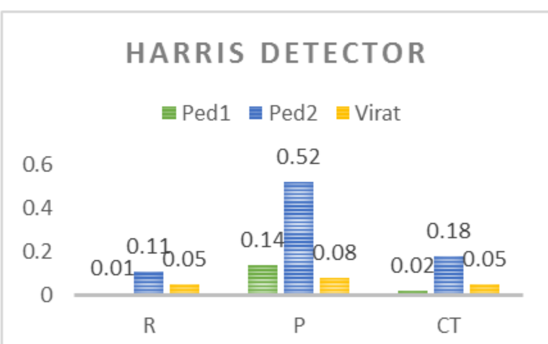


Figure (8): The bar chart result of proposed anomaly detection system using HARRIS detector.

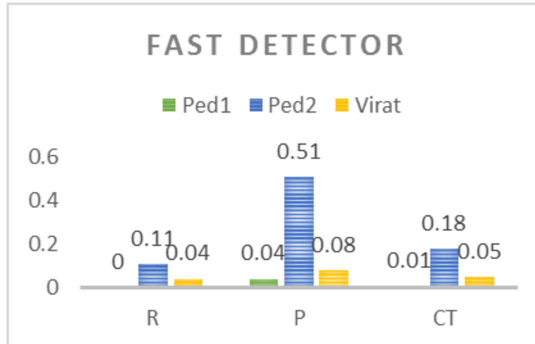


Figure (9): The bar chart result of proposed anomaly detection system using FAST detector.

5. Conclusions

The performance of the suggested approach that uses the k-mean data stream clustering algorithm with features extracted from the interest point pairs as input features to the approach indicates that the recognition performance had not improved. This is due to the fact that the data stream modeling algorithm is single-pass algorithm that need a decisions to be made with little amount of data and in this approach the k-means utilized and k-means need all data available before clustering. So this algorithm suffer from the stability/ plasticity problem; that mean the clustering results change from frame to frame as new data are entered. Further work is needed to increase the detection rate and decrease the false alarm rate.

References

Roshtkhari, M. J.; "Visual Event Description in Videos"; PhD. Thesis; McGill University; Faculty of Graduate Studies and Research; Department of Electrical and Computer Engineering; Montreal; 2014.

Pal, S. K.; Petrosino, E.; and Maddalena L.; "Handbook of Soft Computing for Video Surveillance"; CRC press; 2012.

Qian, H.; Wu, X.; and Xu, Y.; "Intelligent Surveillance Systems"; Springer Publishing Company, Inc.; 2011.

Castro, J. L.; and Et al.; "Intelligent surveillance system with integration of heterogeneous information for intrusion detection"; Expert Systems with Applications; ELSEVIER; Vol. 38; PP.11182–11192; 2011.

Tang, Y.; Ma, B.; and Yan, H.; "Intelligent Video Surveillance System for Elderly People Living Alone Based on ODVS"; Advances in Internet of Things; SciRes; Vol. 3; PP. 44-52; 2013.

Khodabandeh, M.; and Et al.; "Discovering Human Interactions in Videos with Limited Data Labeling"; IEEE; 978-1-4673-6759; 2015.

Lee, D.G.; and Et al.; "Motion influence map for unusual human activity detection and localization in crowded scenes"; IEEE; 1051-8215; 2015.

Biswas, S.; and Gupta, V.; " Abnormality detection in crowd videos by tracking sparse components"; Springer; Machine Vision and Applications; 10.1007/s00138-016-0800-8; Verlag Berlin Heidelberg; 2016.

Al-Nawashi, M.; Al-Hazaimeh, O. M.; and Saraee, M.; " A novel framework for intelligent surveillance system based on abnormal human activity detection in academic environments"; Neural Comput & Applic ; Springer; 2016.

Alan, J. L.; "Intelligent video surveillance in crowds"; Online white paper; http://www.objectvideo.com/objects/pdf/whitepapers/FlowControl.pdf; 2004.

Wiliem, A.; "Robust suspicious behaviour detection for smart surveillance systems"; PhD. Dissertation; Queensland University of Technology; Faculty of Built Environment and Engineering; School of Engineering System; Brisbane; Australia; 2010.

Morris, B. T., and Trivedi, M.M.; "A survey of vision based trajectory learning and analysis for surveillance"; 2008.

Aggarwal, J. K.; and Sangho, P.; "Human motion: modeling and recognition of actions and interactions"; in 3D Data Processing, Visualization and Transmission. 3DPVT. Proceedings. 2nd International Symposium; pp. 640-647; 2004.

Niyogi, S. A.; and Adelson, E. H.; "Analyzing and recognizing walking figures in xyt"; technical report, M.I.T. Media Lab Vision and Modeling Group Technical Report; No. 223; 1994.

Bobick, A.; and Davis J.; "The recognition of human movement using temporal templates"; Pattern Analysis and Machine Intelligence, IEEE Transactions; Vol. 23; No. 3; pp. 257-267; 2001.

Dollar, P.; and Et al.; "Behavior recognition via sparse spatio-temporal features"; in Visual Surveillance and Performance Evaluation of Tracking and Surveillance, IEEE International Workshop; pp. 65-72; 2005.

Shechtman, E.; and Irani, M.; "Space-time behavior-based correlation or how to tell if two underlying motion fields are similar without computing them"; Pattern Analysis and Machine Intelligence, IEEE Transactions; Vol. 29; No. 11; pp. 2045-2056; 2007.

Szeliski, R.; "Computer Vision: Algorithms and Applications"; Springer; 2010.

Guha, S.; and Et al.; "Clustering data streams: Theory and practice"; Knowledge and Data Engineering, IEEE Transactions; Vol. 15; No. 3; pp. 515-528; 2003.

Dougherty, G.; "Pattern recognition and classification: An introduction"; Springer; 2013.

O'Callaghan, L.; and Et al.; "Streaming-data algorithms for high-quality clustering"; in Data Engineering, Proceedings. 18th International Conference; pp. 685-694; 2002.

Aggarwal, C. C.; and Et al.; "A framework for clustering evolving data streams"; in the 29th VLDB Conference; Germany; pp. 81-92; 2003.

Aggarwal, C. C.; and Yu, P. S.; "A framework for clustering uncertain data streams"; in Data Engineering. ICDE. IEEE 24th International Conference; pp. 150-159; 2008.

Cheng, L.; and Et al.; "An online discriminative approach to background subtraction"; In IEEE International Conference on Advanced Video and Signal based Surveillance; pp. 2; 2006.

Lühr, S.; and Lazarescu, M.; "Incremental clustering of dynamic data streams using connectivity based representative points"; Data and Knowledge Engineering; Vol. 68; No. 1; pp. 1-27; 2009.

Wiliem, A.; and Et al.; "Adaptive unsupervised learning of human actions"; in The 3rd International Conference on Imaging for Crime Detection and Prevention; Kingston University, London; 2009.

Shati, N. M.; "2D-objects detection using back propagation neural networks"; MSc. Thesis; Al-Mustansiriah University; College of Science; 2006.

Ali, A. E.; "Multi object tracking using machine learning technique"; PhD. Dissertation; University of Technology; Department of Computer Science; 2015.

Journal of AL-Qadisiyah for computer science and mathematics
Vol.9 No.1 Year 2017

تمييز الحالات غير الطبيعية باستخدام خوارزمية (K-means Data Stream Clustering) في أنظمة المراقبة الذكية.

عبد الأمير عبد الله كريم
الجامعة التكنولوجية
قسم علوم الحاسوب
ameer_aldelphi@yahoo.com

نرجس مزعل شاتي
الجامعة المستنصرية
كلية العلوم
قسم علوم الحاسوب
nareen_manar@yahoo.com

المستخلص :

في هذا البحث استخدمت طريقة جديدة لعنقدة دفق البيانات موظفا فيها تقنية الـ (K-Means clustering) لانجاز نظام تمييز الحالات غير الطبيعية. استخدم هذا النظام مجموعة من الخصائص (مثل: المسافة والاتجاة واحداثيات المحور والسيني والصادي) المستخلصة من مجموعة ازواج نقاط الاهتمام باستخدام (HARRIS detectors) او الـ FAST detectors) على الـ (frames) التي تم الحصول عليها من مجاميع بيانات عالمية وهي (UCSD pedestrian) وتتكون من قاعدتين(ped1) و(ped2) واما الثانية فهي (VIRAT).

اشارت النتائج الى ان استخدام الـ (HARRIS detectors) تم انجاز نسبة التمييز ١% مع نسبة خطأ بالتمييز ٦% باستخدام (Ped_1)، و ١٠,٧٥% نسبة تمييز مع نسبة خطأ ١٠% باستخدام (Ped_2)، اما عند استخدام (VIRAT) فأن نسبة التمييز كانت ٥% مع ٤٠% نسبة خطأ بالتمييز. استخدام الـ (FAST detectors)، كانت

نسبة التمييز (٠,٥%، ١٠,٧٥%، ٤.٠٨%) مع نسبة خطأ في التمييز (٥ %، ١٠,٥٠%، ٤٥,٩٢%) باستخدام الـ Ped_1، Ped_2، VIRAT وعلى التتابع.

الكلمات المفتاحية : تمييز السلوك غير الطبيعي، تمييز السلوك الشاذ، المشاهد المزدحمة، تحليل اشرطة الفيديو، أنظمة المراقبة الذكية.