EXTRACTING SOCIAL NETWORK GROUPS FROM VIDEO DATA USING MOTION SIMILARITY AND NETWORK CLUSTERING

A Thesis Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Kinjal Haresh Kotadia

May 2018

EXTRACTING SOCIAL NETWORK GROUPS FROM VIDEO DATA USING MOTION SIMILARITY AND NETWORK CLUSTERING

Kinjal Haresh Kotadia

APPROVED:

Dr. Shishir Shah

Dr. Christoph Eick

Dr. Xuging Wu

Dean, College of Natural Sciences and Mathematics

ACKNOWLEDGEMENTS

My deepest gratitude goes to Dr. Shishir Shah, Ph.D., for his guidance, support, patience and advice throughout this endeavor. My special thanks go to everyone in my family and my friends for their constant encouragement and total support in my attainment of this goal.

EXTRACTING SOCIAL NETWORK GROUPS FROM VIDEO DATA USING MOTION SIMILARITY AND NETWORK CLUSTERING

An Abstract of a Thesis

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Kinjal Haresh Kotadia

May 2018

ABSTRACT

Detecting Social Network Groups from Video data acquired from surveillance cameras is a challenging problem currently being addressed by the Data Mining and Computer Vision Communities. As a part of continuing research in this area, a new graph-based post analysis approach is developed to process data obtained from the state-of-the-art Detection and Tracking systems to extract the various social groups present in it. The process of extracting social network groups is primarily divided into two tasks. The first task consists of finding a method to compute a graph that connects all the people present in the video. Motion similarity between the tracks of the people on the ground plane is used as a metric to compute the weights on the edges of the graph. The second task is to cut the graph to form groups which is done by creating a minimal spanning tree and cutting the edges with least weights. The number of cuts to be made depends on the number of groups that are present in the video. To deal with the problem of unknown number of groups, the parameter of consistency of within cluster distances is exploited and the number of groups is decided by the finding the elbow point in the plot. The method shows promising results with UCLA Courtyard Dataset Videos and Simulation systems. This work can be regarded as one of the many approaches to solve the problem of “Detecting Social Networks from Video Data” which tend to exhibit decent outcomes.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 SECTION 1

Analyzing data and extracting useful information from it is an extremely crucial task. With the availability of extensive technology and the ease of access to multimedia devices, it can safely be stated that nowadays data is not restricted only to text data. A huge amount of data is present in multimedia format specially videos. Mining information from videos is one of the challenging tasks for the Computer Vision Community. Computer Vision algorithms like crowd segmentation [1,2], people detection and tracking, [3] and people recognition [4,5] have reached the state-of-the-art stages and are now in use to solve challenging real-world problems. They open doors to new range of problems that extend beyond traditional capabilities. Crowd analysis [6], behavior detection [9], action detection [10] are a few to mention.

Human beings are social animals and have a basic need to communicate with each other to survive. Natural human tendency compels every human being to be a part of a social group. The English Dictionary defines Social Network as “A network of social interactions and personal relationships.” Every individual usually is a part of multiple social network groups. People have connections with their family members, friends, colleagues, classmates, and other acquaintances. Besides that, they also interact with people they don’t know on a personal level like the shopkeeper across the street, fellow pedestrians, community helpers, or people travelling on the same bus. Understanding such connections, especially in places like prisons or in a public crime scene, can prove to be a topic of interest. Also, these connections can be used for analysis in many other fields of applications including security, medical, surveillance, and military purposes.

1.2 MOTIVATION

Facial-Recognition Technology have been in existence since the 1960's. However, advances in Machine-Learning Technology and Video Analytics have improved accuracy to a point that a new set of applications are becoming viable which can exploit the capabilities of the existing system and open new doors for innovation. Surveillance and security is one of the major application industries where computer Vision and Machine Learning is increasingly proving to be a boon. If there exists a system that could analyze videos, that would detect all kinds of physical crimes including something as small as pickpocketing or shoplifting as well as something huge like a terrorist attack, it cannot be said enough how useful it would prove to be. With the right kind of facial recognition technology and right information such physical crime activities could be prevented from occurring. Facial-Recognition technologies are by themselves sufficient to solve smaller crimes. However, monitoring and detecting imminent terrorist activities would require a much more advanced system that could not only recognize a suspect in videos but also detect groups that a person is a part of and give insights about other potential suspects. A machine that can identify all the acquaintances of the terrorist, can be a valuable addition to the security and surveillance industry. It can also be used in other industries like medicine for finding the source of a contagious infection. If a machine can find out a common link in the interactions made by two infected patients, then there could be a chance of the link being the source of infection and necessary steps can be taken to prevent the disease from spreading.

This thesis topic was inspired due to these reasons. The need to create a system that would make the task of monitoring more convenient coupled with the need to mine useful information that is in video format, birthed the idea of a system that could extract social

network groups from video data. Some technologies on which this system is proposed to be built are explained in the next section.

1.3 BACKGROUND

This thesis is based on the results of previous technologies like human detection and human tracking. How these systems work is explained in the following sections.

1.3.1 Human Detection

Detecting human beings in videos is a crucial and challenging task in the field of research. The task includes differentiating people from other things in the video like cars, animals, traffic lights, then keeping track of them continuously. Various state-of-the-art detection systems exists which perform these tasks with high accuracy [11]. These systems use background subtraction and PCA algorithms for detection. Other ways use supervised Machine-Learning techniques to train a model using various features that differentiate people from other things in a scene. These features generally are shape oriented like a Histogram of Gradients or Local Binary Patterns.

1.3.2 Human Tracking

Human tracking has always been an interest, due to its importance in many areas, from security, surveillance, defense, as well as for mobile robotics applications like human following. There exists a large number of tracking algorithms like Kalman-Filters, Mean-Shift Filter, and Partial Least Square method. Comparisons of tracking techniques have been made [7] and results have been documented. Tracking systems assign an ID to every detected individual and then extract the position of that individual throughout the course of the video.

1.4 PREVIOUS WORK

Uncovering groups or communities is an important task that has been worked on. There are algorithms that find communities from a network. Modularity maximization, Hierarchical clustering, Statistical Inference are some of the methods. Hierarchical Clustering is a method which uses a similarity metric to quantify the similarity between the nodes and clusters them hierarchically by merging the closet clusters. This clustering could either be top down or bottom up. Modularity maximization forms clusters and uses the modularity to measure the effectiveness of the community. This thesis presents the use of the minimum-cut algorithm. This widely used algorithm is explained in detail in a later section. A unique modularity cut algorithm is proposed [8] , which is an Eigen-based approach for discovering community and leadership structure in an estimated social network. There are other approaches to this problem which have different applications like coherent motion detection using Collective Density Clustering [12], Measuring Crowd Collectiveness [13], and Group Motion Graphs [14].

Creating a network from available data with appropriate metrics is another necessary task for community detection. The metric of similarity to be used is another key factor to create a social network graph. This metric forms the edges of the graph whose nodes are different people. A paper called “Understanding Graph Sampling Algorithms for Social Network Analysis” [15], makes comparisons of various state-of-the-art graph sampling algorithms and evaluates their performances on large-scale social-network datasets.

CHAPTER 2: METHODS

2.1 SECTION 2

As discussed earlier in section 1.2.2, Human Detection and Tracking systems provide the tracking information or the position of a person throughout the length of a video. It detects people even under occlusions and returns the coordinates of the bounding boxes around the person. Ideal tracking data for this thesis requires ground plane co-ordinates of the positions. Identified humans with a unique ID and corresponding ground plane tracking data are the two main requirements of the proposed system. The methods proposed assume these results are reliably obtained and they are used as the raw dataset for the executing the experiments. With improving accuracy of these results, it can safely be assumed that the accuracy of the proposed system will improve as well.

The method design proposed in this thesis can be explained by dividing the method into three sections. The first part is computing a social network graph which connects all the people in the video to each other. The weights on the graph edges are calculated by a metric of motion similarity. The second part consists of dividing this social graph into different groups. This is done by creating a minimal spanning tree and making cuts depending upon the number of clusters required. The third part is deciding the ideal number of clusters, which is performed by using the elbow method. These three techniques put together in a pipeline creates an algorithm that performs the task of extracting social groups from videos. Each individual method is explained in detailed in the following sections.

2.2 COMPUTING SOCIAL GRAPH

A Social Graph can be defined as a set of nodes connected to each other by weighted edges. A Social Graph G is represented as G= (V, E), where V is the set of vertices or nodes and E is the set of edges. In this experiment, the graph nodes represent each person and the edge represents the connection between two people. Connection can be calculated by measuring the amount of interaction, the amount of time spent together, similarity in their actions or body language between two people. In some scenarios, two people might be together for a very short time in a visual context, but they could be best friends which can be concluded from the affection they show towards each other for that short time. On the other hand, two people could be standing together for hours and not know each other. For example, people standing in a long queue for hours could visually appear as two people standing close to each other for hours, but, they are just strangers who just happen to be waiting in that queue. So, constructing a social graph that accurately represents the real social connections is a task that is non-trivial. There are several difficult scenarios and to tackle each of them is a challenging task since every scenario is distinctively different from each other. In this thesis, the method used to compute the Social Graph was originally proposed in [8] and it uses the concept of Motion Similarity to calculate the edges of the graph.

The assumption that recognition is already done gives the nodes of the graphs. The number of nodes in the graph is known because a post analysis of the video is done. Hence, all the people recognized by the recognition system form the nodes of the graph. Every node is connected to every other node initially. Hence, it is a fully connected graph. To calculate the edge weights, initially the tracks of two people are calculated. Suppose, there are N individuals in the video, then a vector Tracks of size N is created such that,

for n ∈ N, Tracks [n]= $\{x_1, x_2, x_3, \ldots\ldots, x_t\}$

Where x_i = (x, y, t) where x and y are the ground plane co-ordinates of n at time frame t.

For, two people x and y, given a pair of tracks X_x and X_y which overlap temporally between (t_0, t_t) such that

t_0 = max ($X_{x,0}$, $X_{y,0}$) and t_t = min ($X_{x,t}$, $X_{y,t}$), the connection between then C_{xy} is calculated by the following formula.

$$\mathrm{Dmn} = \exp\left(-\frac{\Sigma_{t=t_0}^{t_t} \left|\left|x_m^t - x_n^t\right|\right|^2}{2\sigma^2(\mathrm{t_t} - \mathrm{t_0})}\right) \qquad \text{Equation (1)}$$

σ is the scaling factor that controls the influence of the variations between the track locations. Changing the value of sigma changes the distance value that is considered to be 'far' between two nodes. Summation is performed only for the values of t where both x and y are present. For example, on a timeline of t=0 to t=100 x and y are present in the video sequence for the highlighted parts, then the summation calculations and variable values will be as follows:

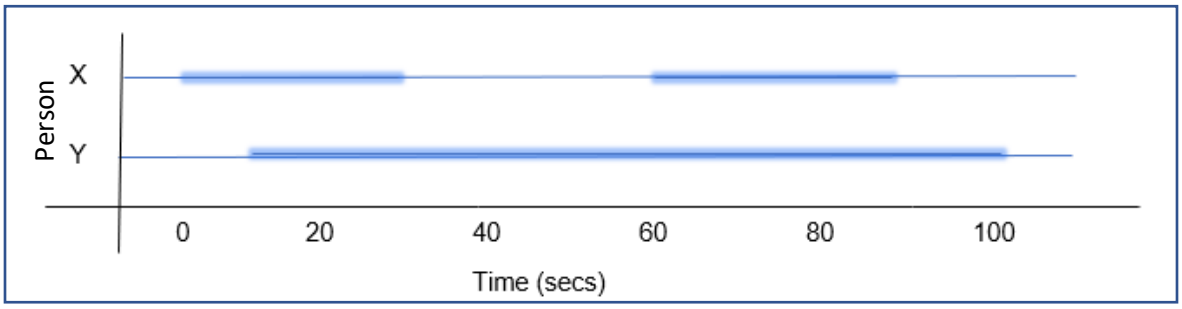


Figure 2-1. Shows the highlighted time frames for which the person is present in the video.

In Figure 2-1, t_0 = max (0,10) = 10, t_t = min (90,100) = 90. Hence, the summations happen over the range of 10 to 90. But instead of summing over the entire time frame of (10,90),

only the instances from 10 to 30 and 60 to 90 will be used for the calculation because only for those time periods both X and Y are present in the video. Instead of subtracting t_0 from t_t in the denominator of the equation (1), the calculation of the term would be (30 - 10) + (90 - 60) = 50 i.e. the equation would be normalized by the number of terms in the summation in the numerator.

For each x and y, the connection is calculated, and a fully connected graph is generated using these values as the edges weights. The best way to represent this structure is creating a N*N matrix, for N members in the video, such that the value at the i^{th}, j^{th} cell is the C_{ij} value calculated as discussed above. The values obtained are small and need to be normalized for further processing. In order to normalize the small values, the logarithm of all the values were calculated. Then for each row, the maximum value of that row was subtracted from each value. After subtracting, the exponent of the values was computed. These calculations make the original values a little larger. The values can be normalized by dividing by the summation of the row. Diagonal values were omitted during the process of normalization. This implies that the social graph does not contain self-associations. The above process can be represented in the equation as follows:

For each row, R of the network matrix,

1. LL = Log(R)
2. M = max(LL)
3. LL = LL - M
4. R = exp (LL)
5. R = R / sum(R)

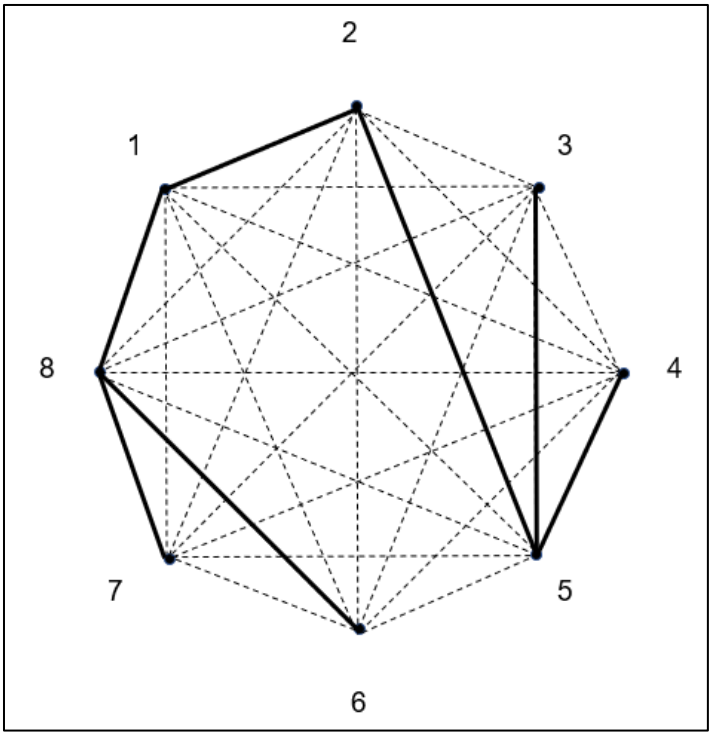
These normalized values give a better representation of the edge weights and can be efficiently used for calculations. A fully connected graph is constructed where each node is connected to every other node and the associations or the edge weights correspond to the similarity of motion of the two nodes as well their proximity.

2.2 DIVIDING THE GRAPH INTO SOCIAL GROUPS

After obtaining a fully connected graph with weighted edges, the next task is to make graph cuts to form groups. To do that, first a minimal spanning tree is computed. A minimal spanning tree can be defined as a tree that is obtained by visiting all the nodes of the graph at least once such that the cost of path is minimum. In this case, all the nodes in the minimal spanning tree are connected in such a way that all the strong edges are preserved. That means each node is connected to the tree by its strongest edge.

Now, the next step is to truncate one of the edges to obtain two groups. The weakest edge in the minimal spanning tree is selected for pruning which results in two different groups. If the number of groups formed is n then n-1 edges are pruned. For example, if following is the minimum spanning tree, one edge could be cut to form two clusters as shown. Suppose the edge that connects node 2 and 5 has the minimum weight and is selected for pruning. Then, the two clusters formed can be seen in the Figure 2-2 (C) below.

A



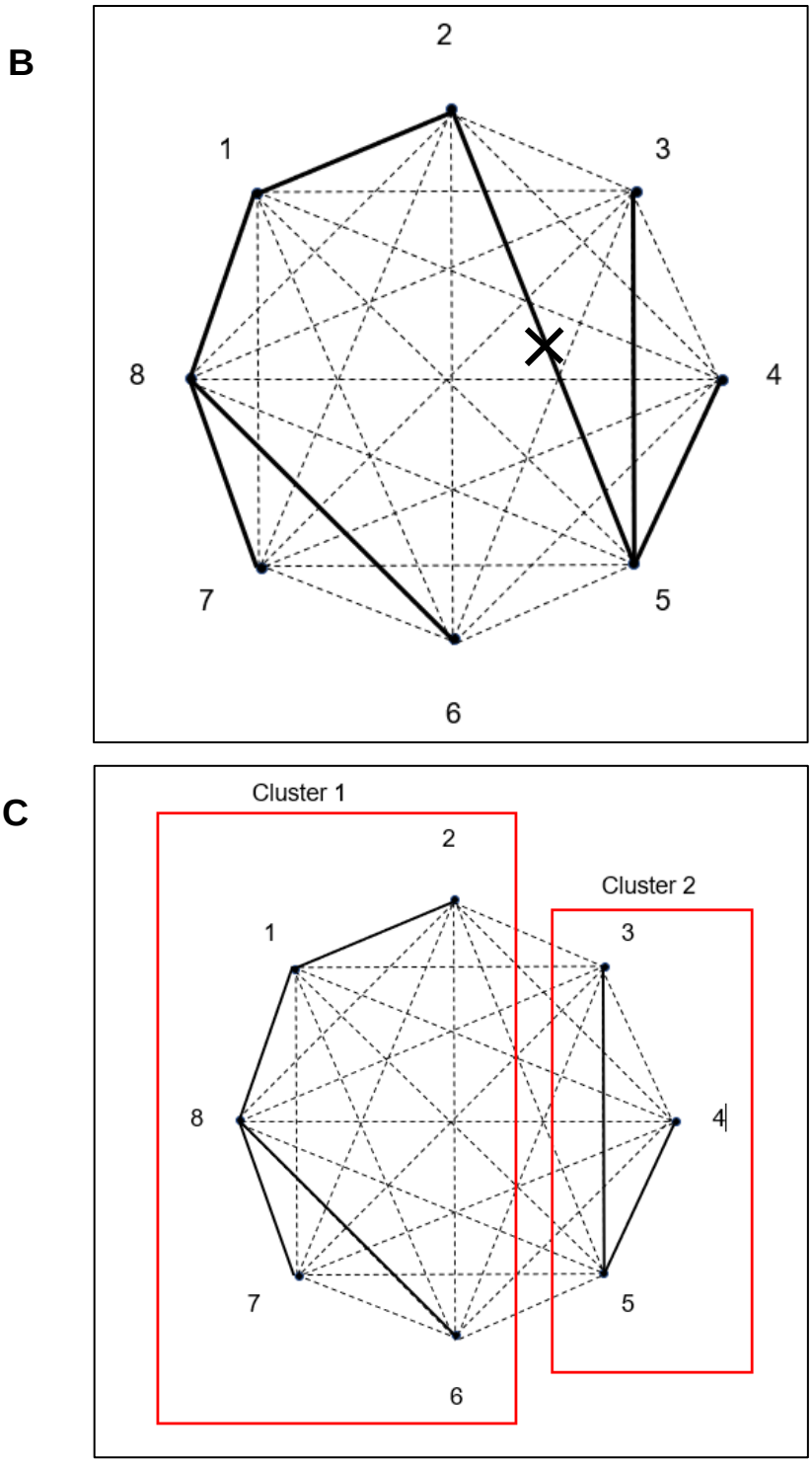


Figure 2-2. A minimal spanning tree (A), the edge that is going to be pruned (B) and the resulting clusters formed due to pruning one edge (C).

2.3 DECIDING THE OPTIMUM NUMBER OF GROUPS

To find the ideal number of groups present in the video, a technique called the Elbow method is used. This method uses the variation of within-cluster distances to find the ideal number of clusters. The parameter value converges after a point in the graph and that point is known as the elbow point. The corresponding value of the number of cluster at the elbow point is used as the ideal number of clusters. The elbow point is the maximum value of the third differences of the within-cluster distances. Figure 2-3 below shows a graph of the number of clusters vs the summation of the within-cluster variance. It can be seen from the graph that the elbow point is when the number of clusters is equal to 5 and the within cluster variance value converges after that point.

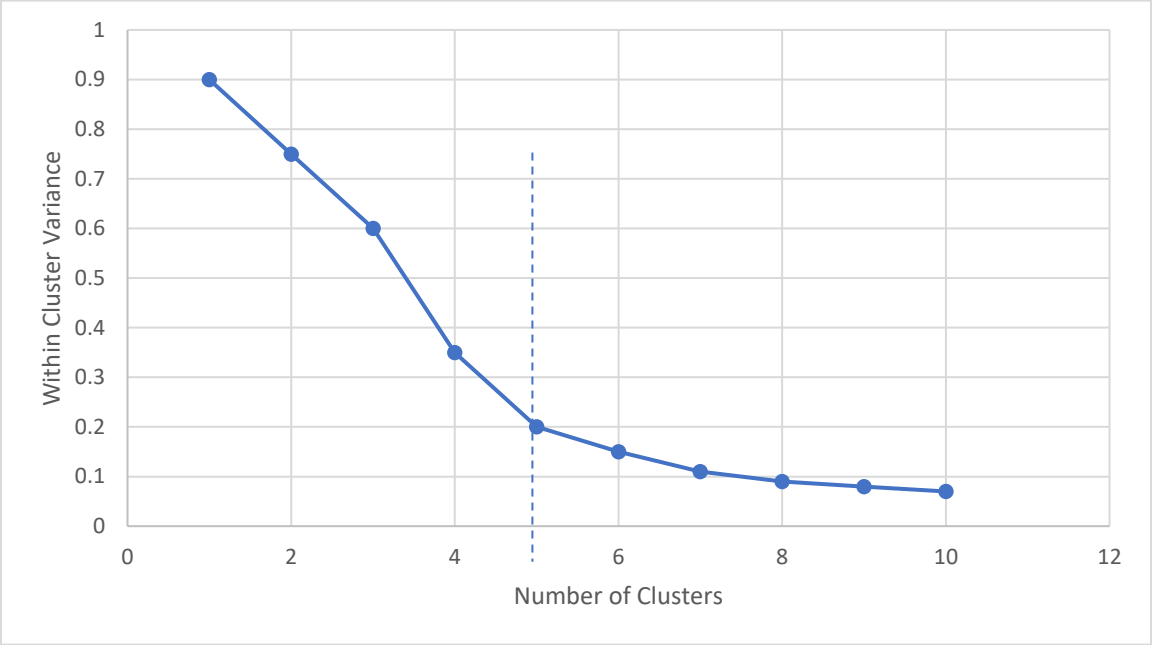


Figure 2-3. Graph of within-cluster variance vs number of clusters depicting the elbow point.

As mentioned earlier, to find the elbow point, third differences are used. For example, as seen in the Table 2-1, the maximum value in third differences corresponds to the number of clusters 5 which is the elbow point.

Table 2-1. Table Showing an example calculation for 1st, 2nd and 3rd differences. The elbow point can be seen highlighted in red color

Number of Clusters	1	2	3	4	5	6	7	8	9	10
Within Cluster Variance	0.9	0.75	0.6	0.35	0.2	0.15	0.11	0.09	0.08	0.07
1st Difference		-0.15	-0.15	-0.25	-0.15	-0.05	-0.04	-0.02	-0.01	-0.01
2nd Difference			0	-0.1	0.1	0.1	0.01	0.02	0.01	0
3rd Difference				-0.1	**0.2**	0	-0.09	0.01	-0.01	-0.01

The limitation of using 3rd differences lies in the assumption that there are at least 4 groups present in the video. If the number of groups are less than 4, then values of second differences can also be used to make an inference with the assumption that there are at least three groups in the video. But third differences tend to provide more accurate results.

After generating the minimum-spanning tree, first the within-cluster variance of grouping everything in one cluster was calculated. Then, the weakest link was cut, and two clusters were formed. The summation of within cluster variance was calculated again. In the original minimum-spanning tree two cuts were made to form three clusters and so on. This process was repeated for the number of clusters from 1 to 2*N/3 where N is the number of subjects in the video. The number 2*N/ 3 is a parameter obtained by selecting the value which provided the best results when experiments were run for different values of the same parameter.

CHAPTER 3: EXPERIMENTS

3.1 SIMULATIONS

The algorithm was tested on various simulations that imitate real-life scenarios. Different parameters modified to create different simulations were the number of people or data points in the simulation video, number of groups present, the pattern in which they were moving, and the time frame of the video. Each of the simulations are explained as follows with their corresponding results and inferences.

3.1.1 Simulation 1

There were four groups in a 100*100 image in the four corners of the image. Each group had five members each. For every frame of the video, for every person, a direction was selected by using a random generator and then that member moved in that direction by 1 pixel. If the move to be done resulted in an overflow through the boundary, the position of the member was reset to 10 pixels away from the crossed boundary, and the process was repeated for the next frame. The simulation video was run for 600 frames. The Figure 3-1 below shows a screenshot of every 50th frame of the simulation starting from frame 0.

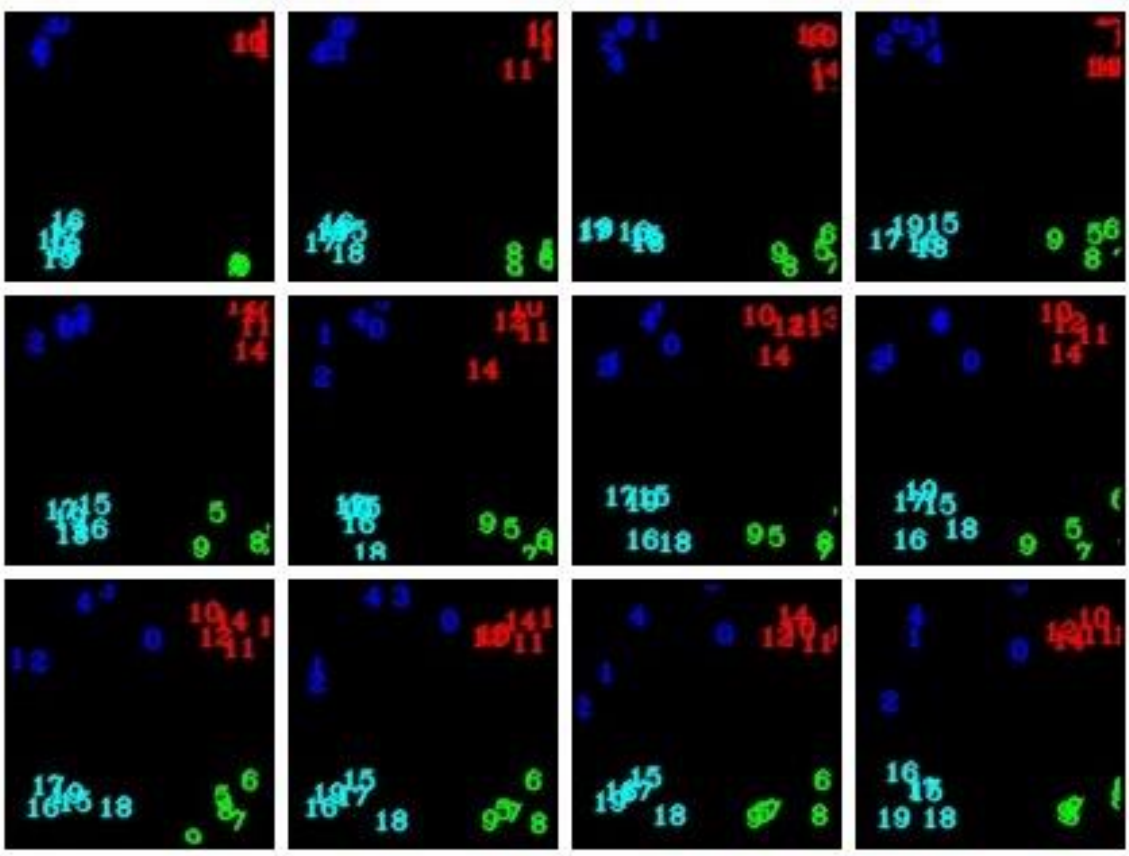


Figure 3-1. An image showing screenshots of every 50th frame starting from frame 0 for simulation video 1.

Initially, the four groups are well defined. However, during the later frames, the groups tend to separate, specially the group on the top left corner becomes dispersed with member 1 and member 2 being close to each other while the remaining three members 0, 3, 4 separate from the original group. The remaining three groups remain well established for most of the simulation video. The results of the groups formed by the algorithm are as follows:

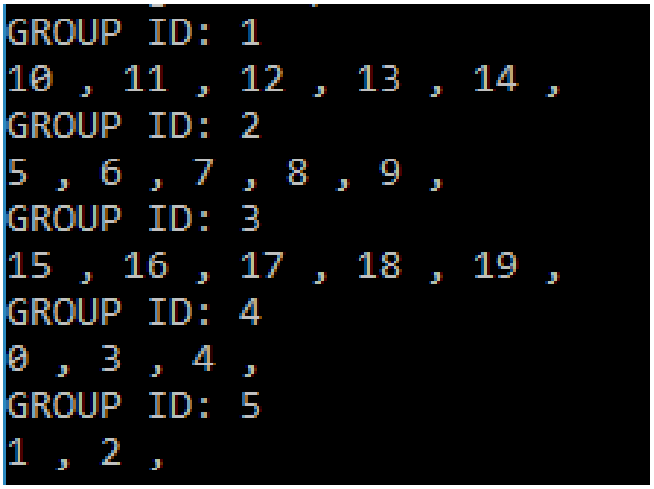


Figure 3-2. Grouping results for simulation video 1.

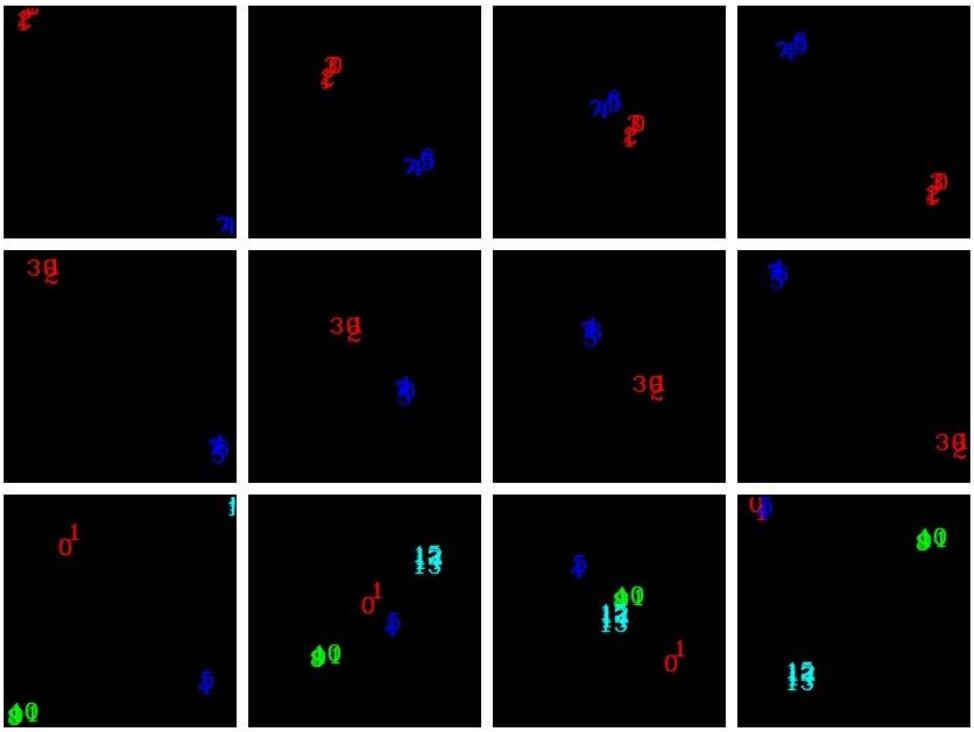
The ideal case results should be four groups, but as explained above the simulation breaks the group at the top left into two groups and so does the algorithm.

3.1.2 Simulation 2

This simulation tested scenarios were some groups were partially present and situations were some groups were present for only a certain amount of time. There were sixteen members in a 200*200 image size. The simulation started with two groups each of size, four groups kept moving along the diagonal for 400 frames. When the group members reach the corner, they start moving in the opposite direction along the same diagonal. For, the next 200 frames, two members from each of the existing groups exited the image, while the remaining two members of each group kept repeating their motion. Also, for the latter 200 frames, two new groups each of size 4 came in the video and were moving along the other diagonal. The Simulation Video ran for 600 frames and a screenshot of every 50^{th} frame starting from frame 0 can be seen in the Figure 3-3(A)

There are two groups which have members 0, 1, 2, 3 and 4, 5, 6, 7, respectively. In the latter half of the video, only 0, 1, and 4, 5, are present in the video. Also, we see two new groups which have members 8, 9, 10, 11 and 12, 13, 14, 15, respectively. The numbers in the images overlap but a better representation with points can be seen in Figure 3-3(B). Four yellow and pink dots are seen in the initial frames while in the latter frames there are only two.

A



B

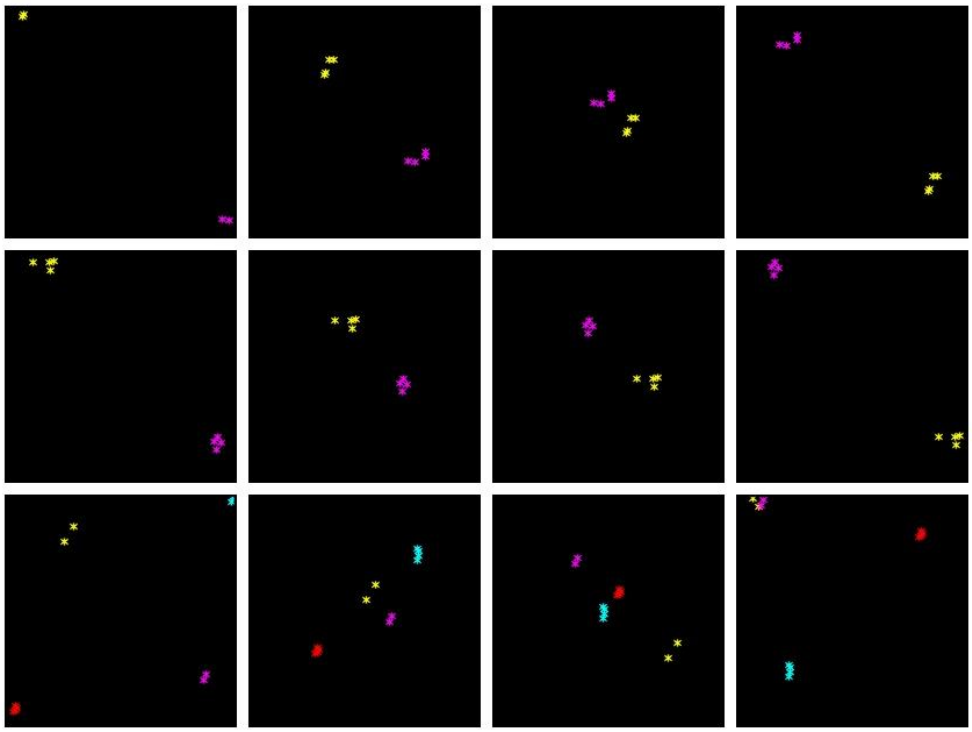


Figure 3-3. An Image showing screenshots of every 50th frame starting from frame 0 with number ids for each person (A) and a different representation of same scenario with color coded groups (B) for simulation 2.

The results of the simulation the shows the presence of four groups in the video. 2, 3 and 6, 7 associate with 0, 1, and 4, 5, respectively even if they are not present throughout the video.

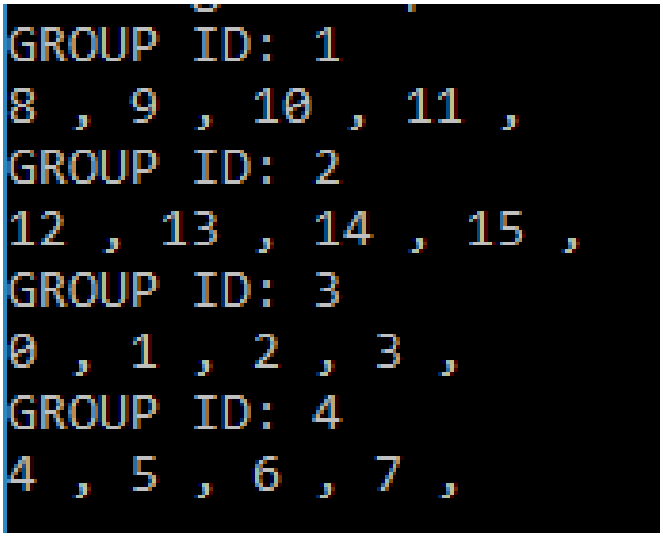


Figure 3-4. Grouping results for simulation video 2.

3.1.3 Simulation 3

There were 20 members in this simulation which were divided into six groups. The simulation image size was 200*200 pixels. Four groups out of six groups present had four members each and remaining two groups had two members each. The four larger groups started from the four corners of the image and moved clockwise by one pixel in every frame along the edge of the image. The initial position of the two smaller groups was in the center of the image and they moved along the diagonal of the image in opposite directions by one pixel. Once they hit the corner, they were transported back to the center of the image and they repeated their movement along the diagonal. This simulation had different overlapping groups for brief moments. The video ran for 1000 frames and analysis was done. Position data collected for 1000 frames. Figure 3-5 shows the simulation video screenshots of different frames during the video.

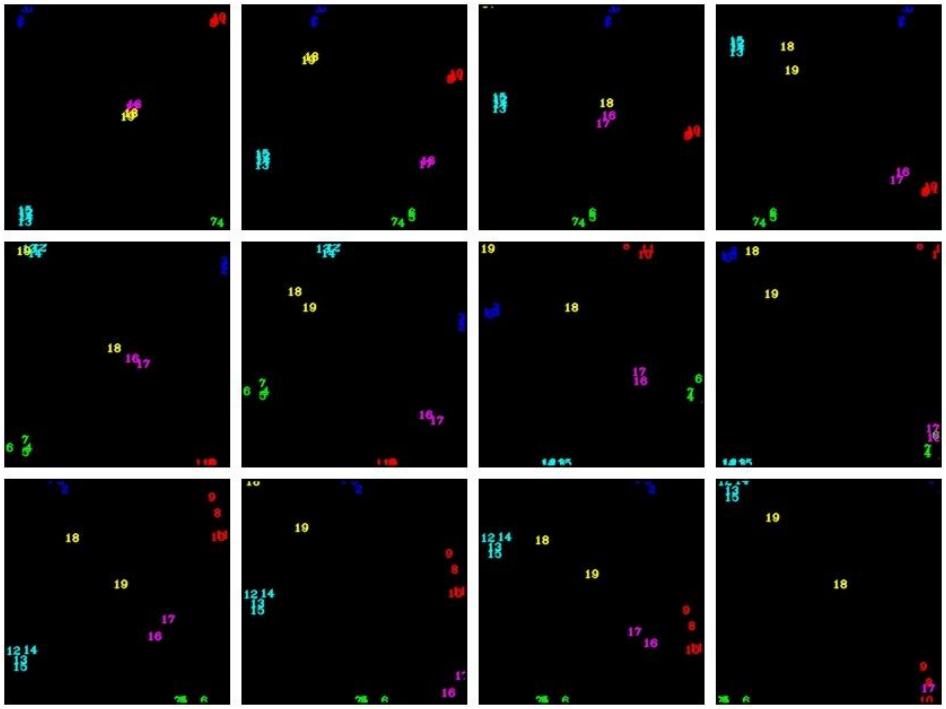


Figure 3-5. An image showing screenshots at frame instances 0, 50, 100, 150, 200, 250, 700, 750, 800, 850, 900, and 950 respectively for the simulation video 3.

At the 0th frame, we can see six well-defined groups. In the 50th frame, the larger four groups that initiated in the corners of the image drifted clockwise. The two smaller groups whose members were 16, 17 and 18, 19 started moving away from the center of the image along the diagonal towards the corner of the image. The groups overlap for multiple brief moments, especially when the smaller groups reach the corner of the image like in the frames 700 and 950. Throughout the video there were six separate groups each with their own motion. Members 18 and 19 drifted apart a few pixels as seen in the latter frames, however they had the same motion throughout the video. The action imitated a scenario where member 19 kept following member 18. The grouping results of this experiment showed seven different groups as seen in the image. There was a false positive detection of the group. The group with members 8, 9, 10, 11, split into two groups 8, 9, and 10, 11. This was because the elbow point indicated a higher third difference of within cluster variance for seven groups instead of six which led to the algorithm generating false positives.

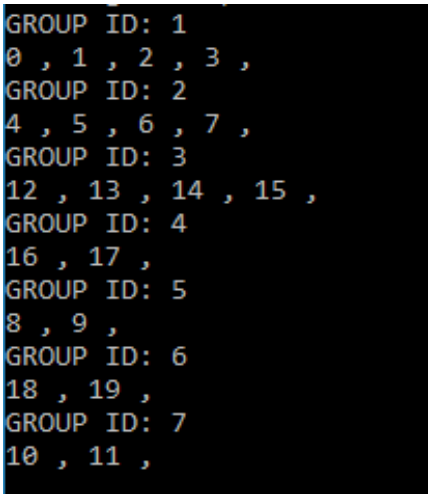


Figure 3-6. Grouping results for the simulation video 3.

3.1.4 Simulation 4

This simulation scenario replicates a person as a part of multiple groups which is a realistic and an intuitive possibility. There were four groups of five members each in an image size of 200*200 pixels. The four groups are well-defined in the four corners of the image. There was a member who spent some time with a particular group for 200 frames and then spent 300 frames with another group. The four groups had members from 0 to 19 with 5 consecutive numbers being part of a particular group. Member 20 was a part of the group with members 15, 16, 17, 18, 19, for the first 200 frames with is the group in the bottom left corner of the image. For the remaining 300 frames member 20 was a part of the group with members 10, 11, 12, 13, 14, the group on the top right corner of the image. Two screenshots in the Figure 3-7 below show member 20 being a part of two different groups at time instances 100 and 340 respectively. In order to make the screenshot legible, other members of the group were represented by a dot.

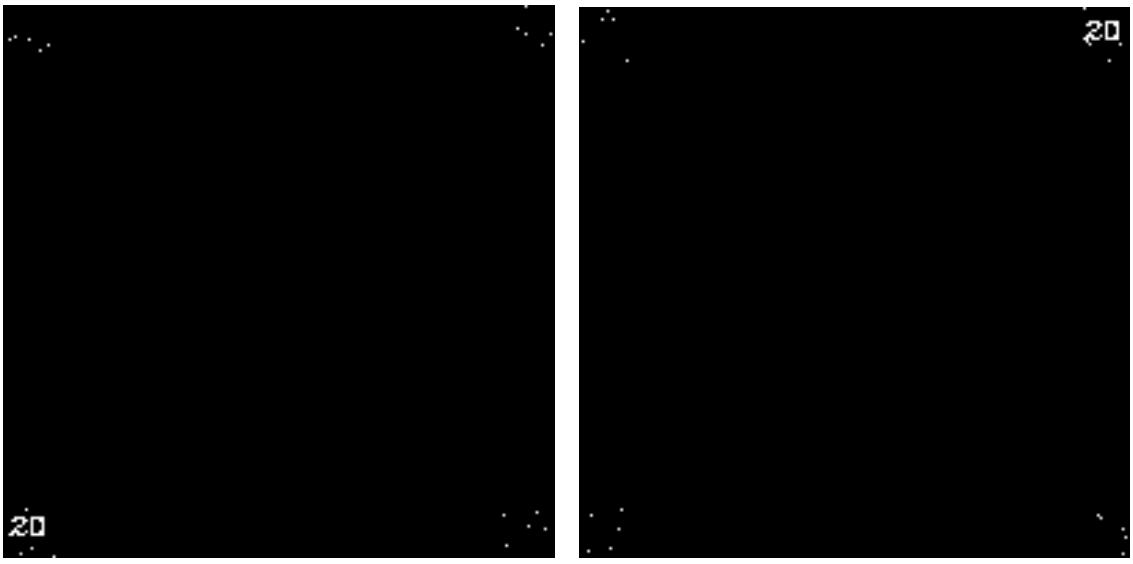


Figure 3-7. Images showing screenshots of simulation video 4 taken at frames 100 and 340.

The results of this simulation show expected results grouping member 20 with the group on the top right. This is the group with which the member, in question, spent more time. The following image shows the results obtained from processing simulation video 4 with the proposed algorithm.

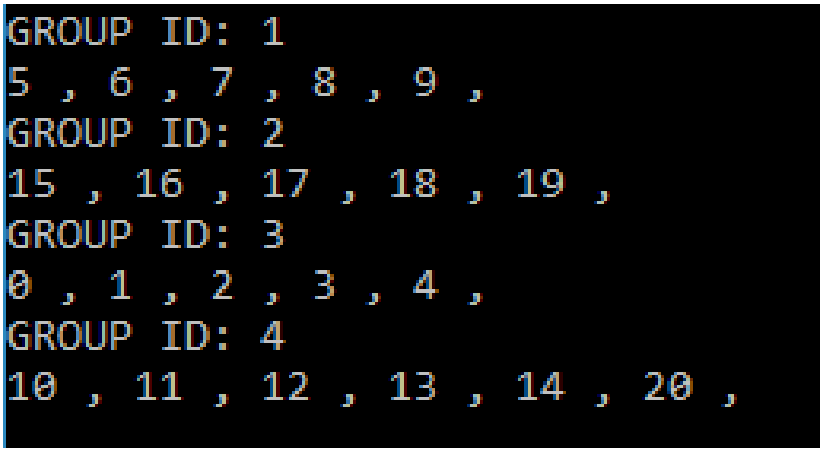


Figure 3-8. Grouping results for simulation video 3.

3.2 DATASET EXPERIMENTS

3.2.1 Data

Apart from simulations, this algorithm was tested on a public dataset obtained from University of California, Los Angeles called the UCLA courtyard dataset. This dataset is available for free to researchers at academic institutions (universities, schools, and government research labs) for non-commercial purposes. It contains high-resolution videos of various, co-occurring activities taking place in a courtyard on the UCLA campus. It has annotations indicating various groups present in the scene as well as positions of each individual along with their unique ID's. These positions are four co-ordinates of the bounding box surrounding an individual. The groups are indicted with a bounding box as well. Members of the groups can be found by extracting all the ID's whose individual bounding boxes lie within the group bounding box, either partially or completely. There are six different videos each having about 30,000 frames on average. For experimentation purposes, only parts of these videos were used.

The dataset was created for the purpose of crowd behavior analysis, which means it was used to find out whether people are sitting together, walking together, or standing in a queue. So, in some cases the ground truth might group the entire crowd standing in the queue one group, but the algorithm proposed might divide it into two different groups. Another important assumption made while implementing the algorithm was the videos gives the ground-plane position of people in the video. This dataset has videos taken from a higher angle, but they are not ground-plane values. This affects the grouping results to some extent.

3.2.2 Performance Measures

The results are interpreted mainly by three parameters:

1. Number of hits – The number of groups correctly identified with each group having the exact same members as indicated in the ground truth.
2. Number of misses – The total number of groups that exist in the ground truth but were not identified as a separate group by the algorithm.
3. Number of false positives – The total number of groups that were not present in the ground truth but were identified as groups by the algorithm.

Accuracy can be evaluated by calculating precision and recall. The equations for precision and recall are as follows:

Precision = (Number of hits) / (Number of hits + Number of false positives)

Recall = (Number of hits) / (Number of hits + Number of misses)

3.2.3 Parameter Selection

There are three parameter values which need to be determined for the algorithm to run.

1. Length of the section to be analyzed at a time -This algorithms results depends upon the number of frames analyzed to create the network. When more and more frames are added, the network becomes complex with all the edges having weights and grouping becomes difficult. For this reason, the video needs to be divided into smaller sections and each section needs be analyzed separately. This gives an idea about how the relationships of an individual flourished throughout the video. An individual could be a part of one group for some time and then other group. Running the algorithm on the entire video in this case would place that person with

either of the two groups. Instead, if two sections were analyzed separately, then the person could be grouped with the respective groups.

The results for a few iterations are shown in the Table 3-1.

Table 3-1. Table showing results for different number of frames

Number of frames	50	100	200	400	600	800
Number of Groups in the Ground Truth	15	15	16	16	19	19
Number of Groups According to the algorithm	17	17	17	18	18	16
Number of hits	12	12	15	14	14	14
Number of misses	0	0	0	1	4	6
Number of false positives	2	3	1	2	2	2
Precision	0.86	0.8	0.94	0.87	0.87	0.87
Recall	1	1	1	0.93	0.78	0.7

The Precision – Recall values for 200 frames was the best and hence that value was selected as the parameter value for the number of frames to perform all future experiments.

2. The value of sigma was discussed in Equation (1). After various trials of performing experiments with different values of sigma, the algorithm produced optimum results when this value was 5 for the UCLA courtyard dataset.
3. While finding the optimum number of clusters, the number of clusters corresponding to the elbow point of the graph were selected. This is done by dividing the graph in one cluster and finding the within-cluster variance, then dividing it into two clusters and finding the sum of the within-cluster variances of the two clusters and so on. The number of times this loop should be run is an important parameter which was decided by performing various experiments and selecting the best parameter value which was 2*N/3 as discussed in Section 2.3

3.2.4 Software Development

A small implementation of an interface was created which enabled the user to query the results. It displayed an image showing snippets of all the people present in the video. After the user clicks on a person, the software displayed all the people who were friends with the person or all the members of the group the selected person was a part of. When the user clicks on one of the friends, the software displays the part of the video for which the two people selected were together. The software marks the two people in question with two black 'X's so that they can be seen together. Figure 3-9 (A) shows all the people present in the video and 3-9 (B) shows the friends of the person clicked on. The final outputs are shown in the following section.

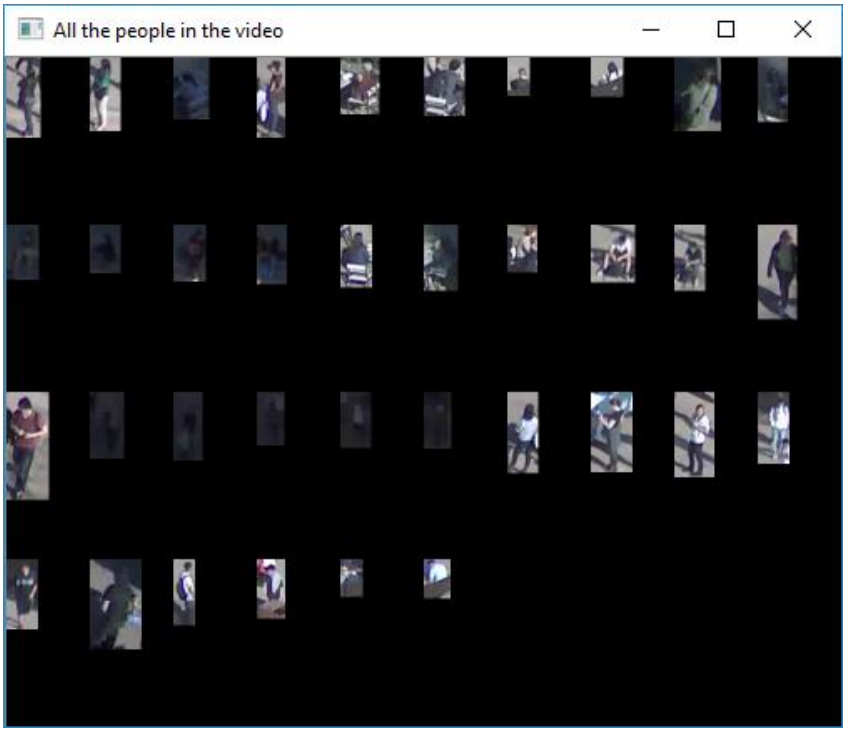


Figure 3-9 (A). – An image showing all the people present in the video.

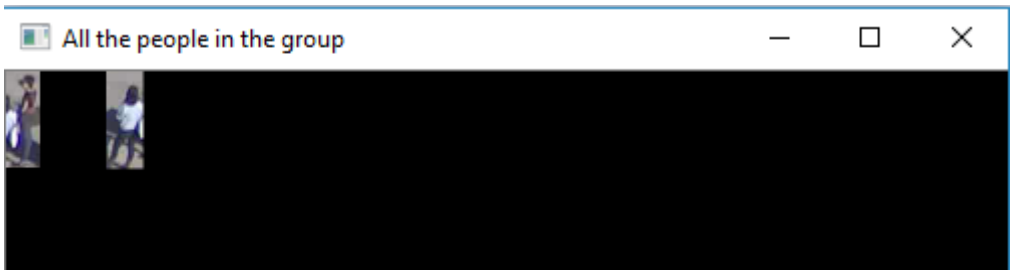


Figure 3-9 (B). – An image showing all the friends of the selected person.

3.2.5 Ground Truth and Results

Experiments were run on different videos and the results are as follows: The white boxes show the ground truth and the colored number indicate groupings according to the algorithm. In every image, two black “X”’s are seen on two people. They indicate the person in question and the friend of the person who the system is trying to track. These people have to be selected interactively using user input in the beginning of the execution. This implementation was created for demonstration purposes.

The screenshots of the video frames of different videos can be seen in Figures 3-10,3-11, 3-12, 3-13, and 3-14. The results of the corresponding videos can be seen in the Tables 3-2, 3-3, 3-4, 3-5, and 3-6 seen on the following pages.

Video 1 was used as a training video to find the parameter values mentioned in Section 3.2.3. The other videos were used to test the performance of the algorithm using those parameter values.

Figure 3-10. An image showing groups for UCLA courtyard dataset video 1.

Table 3-2. Result table for UCLA courtyard dataset video 1

Number of Groups in the Ground Truth	16
Number of Groups According to the algorithm	17
Number of hits	15
Number of misses	0
Number of false positives	1
Precision	0.94
Recall	1

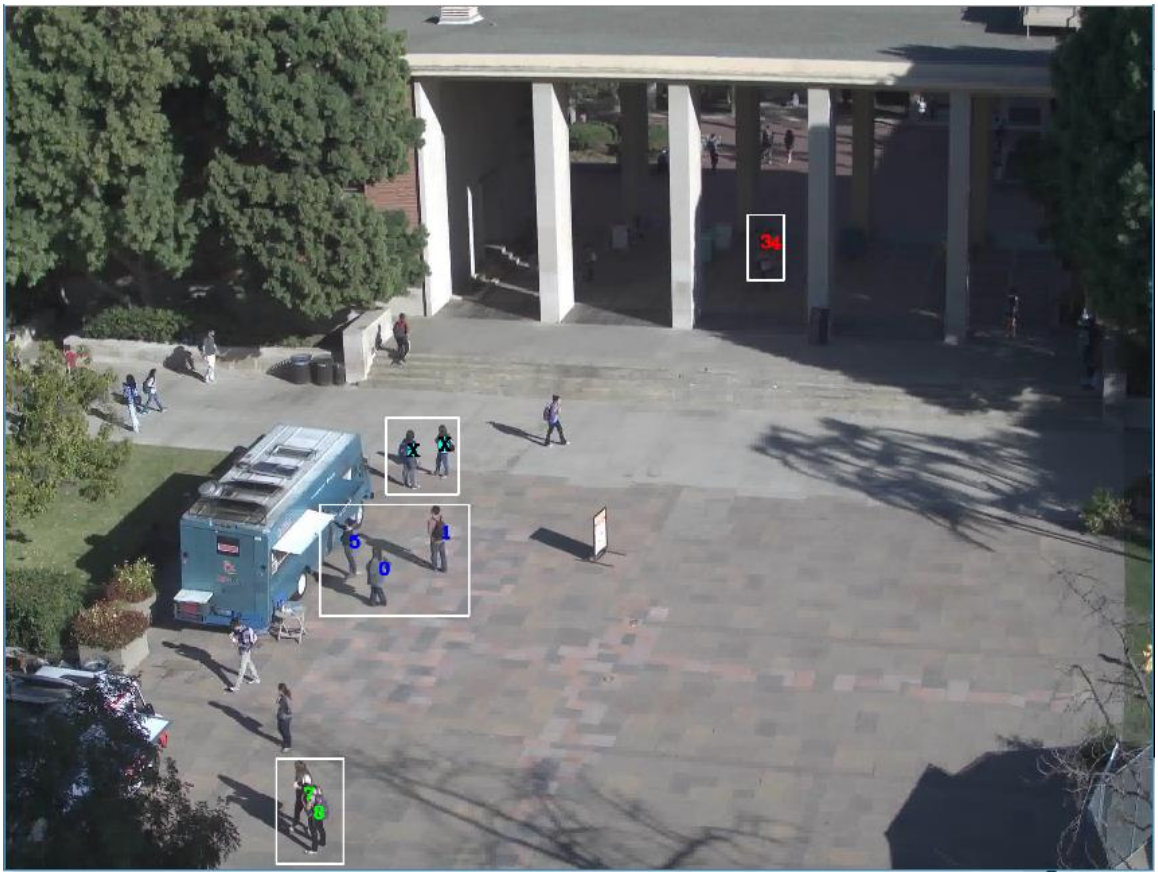


Figure 3-11. An image showing groups for UCLA courtyard dataset video 2.

Table 3-3. Result table for UCLA courtyard dataset video 2

Number of Groups in the Ground Truth	4
Number of Groups According to the algorithm	4
Number of hits	4
Number of misses	0
Number of false positives	0
Precision	1
Recall	1

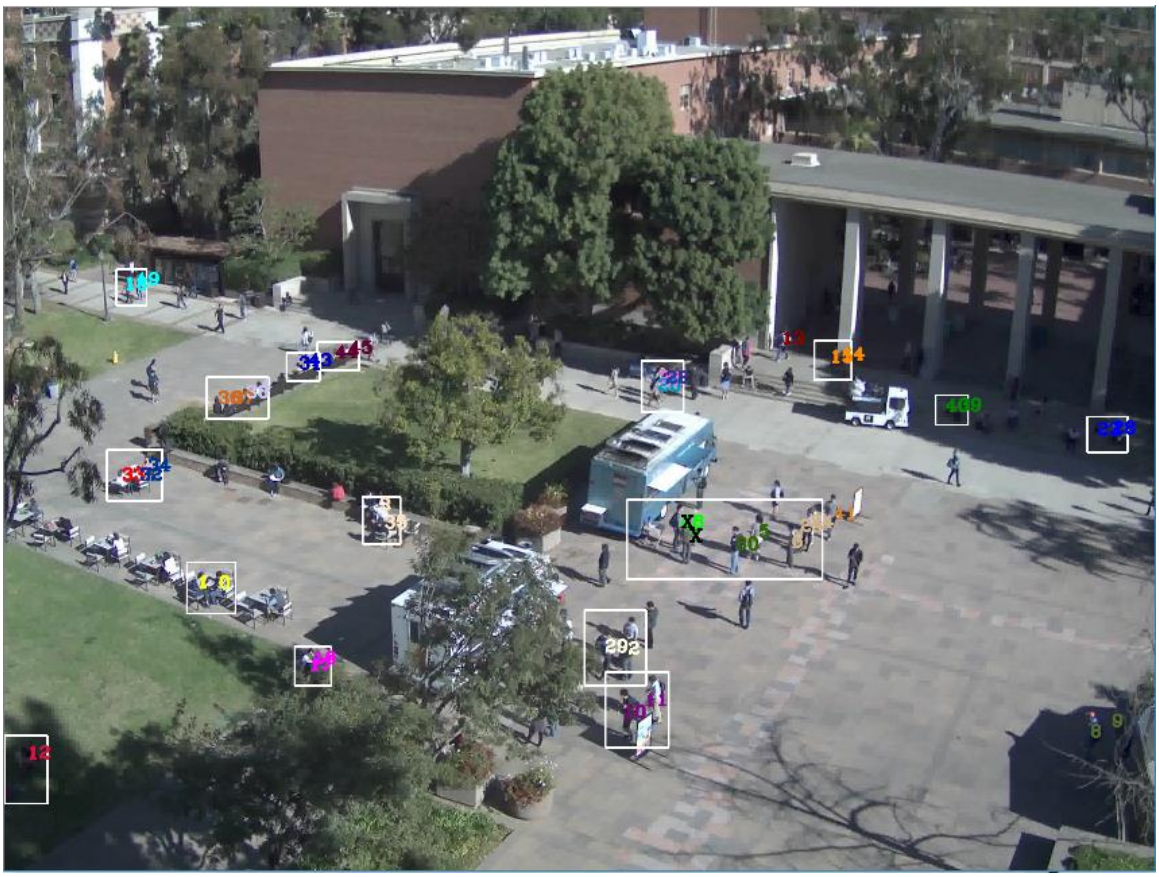


Figure 3-12. An image showing groups for UCLA courtyard dataset video 3.

Table 3-4. Result table for UCLA courtyard dataset video 3

Number of Groups in the Ground Truth	18
Number of Groups According to the algorithm	28
Number of hits	12
Number of misses	0
Number of false positives	7
Precision	0.63
Recall	1

In above video, precision is low because, people waiting in a queue is considered as one large group by the ground truth. However, due the chosen value of sigma, the group was divided into four different groups. The algorithm results can be considered correct because, people standing closer to each other become grouped, while those standing comparatively apart get separated. Also, there a few genuine misclassifications in this video. For example, the group near the top of the blue bus gets identified as three different groups instead one which created two false positives. This video showed comparatively higher false positives which caused the precision value to decrease.

Figure 3-13. An image showing groups for UCLA courtyard dataset video 4.

Table 3-5. Result table for UCLA courtyard dataset video 4

Number of Groups in the Ground Truth	12
Number of Groups According to the algorithm	14
Number of hits	10
Number of misses	2
Number of false positives	3
Precision	0.77
Recall	0.83

The results of the above video show similar detection problems are the previous one. The larger group of people waiting in a queue, which is classified as one group according to the ground truth because they are doing the same activity, is being divided into multiple groups because of the distances between the people. In this video as we can see, there are two misses as well because the groups in the left part of the image frame are recognized as one group instead of being two separate groups.

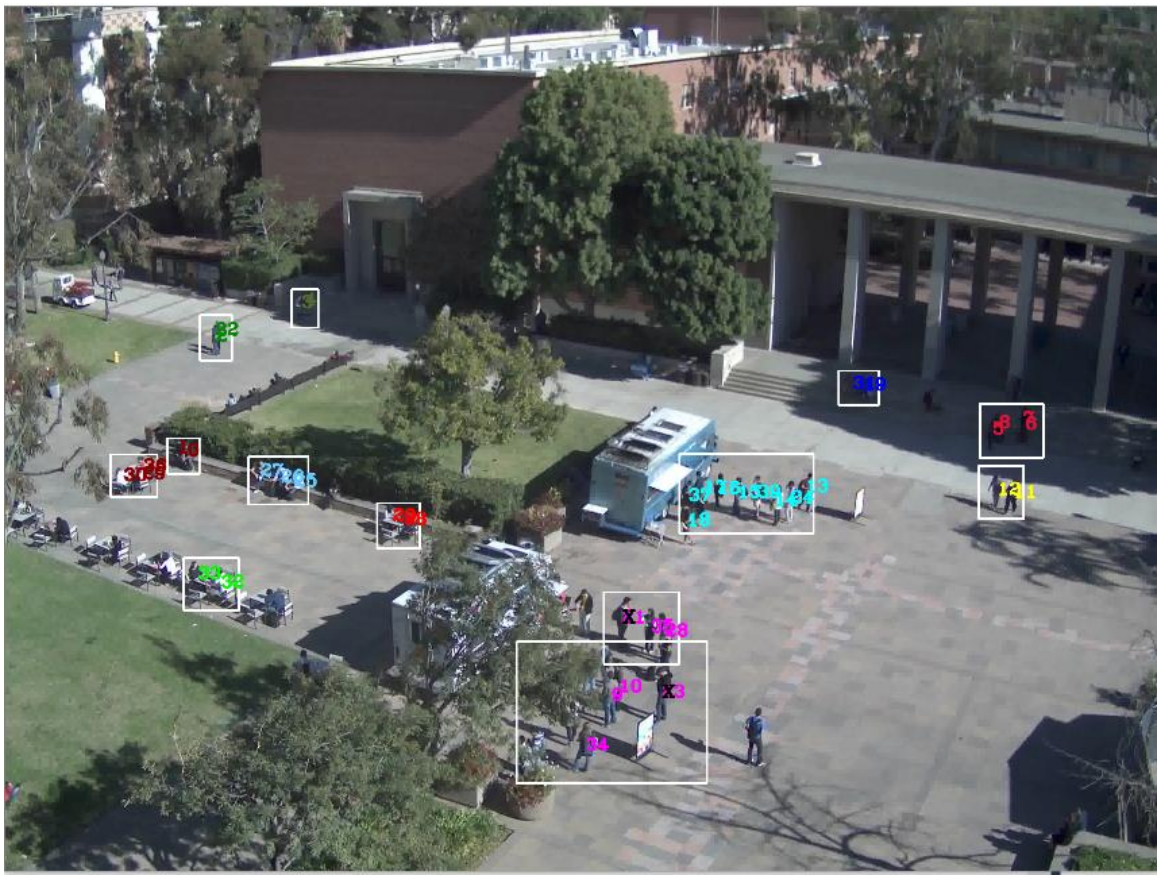


Figure 3-14. An image showing groups for UCLA courtyard dataset video 5.

Table 3-6. Result table for UCLA courtyard dataset video 5

Number of Groups in the Ground Truth	13
Number of Groups According to the algorithm	11
Number of hits	10
Number of misses	2
Number of false positives	0
Precision	1
Recall	0.83

All the results can be summarized in a single table as follows:

Table 3-7. Table showing precision and recall values for UCLA courtyard dataset videos

Video	1	2	3	4	5
Precision	0.94	1	0.63	0.77	1
Recall	1	1	1	0.83	0.83

CHAPTER 4: DISCUSSION

4.1 CONCLUSIONS

This thesis addressed the emerging new problem of discovering social network groups from videos. A Computer Vision solution has been proposed which could be used in solving various problems faced by the security and surveillance industries. Specially, this thesis presents a system that analyzes interactions between different individuals and tries to extract useful information by using an elegant graph-cut technique. The results are promising and can be improved by future work. As mentioned earlier, the results depend on the reliable working of the detection and tracking systems.

4.2 CHALLENGES AND LIMITATIONS

There are a few challenges that the proposed system faces. The system performs well for the datasets and simulations. This is because a considerable amount of domain-specific knowledge is available.

Also, the clustering method inherently works because of the assumption and presence of similar shaped clusters. All the clusters are assumed to be geometrically similar in shape, usually spherical or elongated in some direction. In real life scenarios, there might exist groups which do not agree to the inherent assumption of the cluster shape and would lead to inaccurate results of clustering.

Another limitation is while deciding the ideal number of clusters, it is assumed, there exists only one elbow point. Even though this is true for the majority of cases, there might exist scenarios with multiple elbow points. In such scenarios, there would be a need for a more-sophisticated method to find the ideal number of clusters. The assumption of the shape of

the curve of the graph that decides the ideal number of clusters makes the algorithm highly domain specific and a more generalized form of the application may not produce satisfactory results.

All these limitations could be worked on and improved to make the proposed system more robust, scalable, and efficient.

4.3 FUTURE WORK

This thesis presents a method to extract social network groups from video data. The same task can be performed in numerous other ways. The three sections discussed can have other methods that work, e.g., selecting a different metric to weigh the edges of the graph or selecting a different method to cut the graph into smaller graphs or clusters. Changing even one of the sections might produce significantly different results. Various permutations of these methods can be evaluated to determine the one that produce the optimum results.

Parameter selection can be improved by finding out a way that would automatically select the best parameters according to the input data. In this thesis as discussed earlier, the parameters were selected based on various experiments conducted on the dataset. These parameters might not work well with another datasets. Hence, it is important to find a way to generalize the parameters for all the datasets or to automatically adjust the parameters according to the changing datasets.

Videos used in this thesis were taken from a surveillance camera. All the videos were taken from a considerable height but did not give actual ground-plane values. Ground plane coordinates give information of the actual depth of the video which are important in calculating the associations between people. Converting the video to the ground plane

from a state plane could be a task that would considerably improve the results of the system.

This algorithm performs post analysis of the videos and requires the entire video to extract results. This system could be extended to make it real time. This allows us to dynamically update the social network graphs and perform clustering in real time. This could make the system more useful and efficient.

REFERENCES

[1] L. Dong, V. Parameswaran, V. Ramesh and I. Zoghlami, "Fast Crowd Segmentation Using Shape Indexing," 2007 IEEE 11th International Conference on Computer Vision, Rio de Janeiro, 2007, pp. 1-8.

[2] Tu P., Sebastian T., Doretto G., Krahnstoever N., Rittscher J., Yu T., “Unified Crowd Segmentation." In: Forsyth D., Torr P., Zisserman A. (eds) Computer Vision – ECCV 2008. Lecture Notes in Computer Science, vol 5305. Springer, Berlin, Heidelberg, 2008.

[3] M. Andriluka, S. Roth and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, 2008, pp. 1-8.

[4] Joshila Grace. L. K and K. Reshmi, "Face recognition in surveillance system," 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, 2015, pp. 1-5.

[5] R. Hu, J. Zhang, H. Yu and Y. Liu, "Design and implementation of a surveillance camera system with face recognition functionality," 2014 IEEE International Conference on Electron Devices and Solid-State Circuits, Chengdu, 2014, pp. 1-2.

[6] Amir Sjarif N.N., Shamsuddin S.M., Mohd Hashim S.Z., Yuhaniz S.S., “Crowd Analysis and Its Applications.” In: Mohamad Zain J., Wan Mohd W.M.., El-Qawasmeh E. (eds) Software Engineering and Computer Systems - ICSECS 2011. Communications in Computer and Information Science, vol 179. Springer, Berlin, Heidelberg, 2011.

[7] S. A. A. Shukor, S. Amiruddin and B. Ilias, "Analysis and evaluation of human tracking methods from video," 2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Batu Ferringhi, 2016, pp. 310-315.

[8] T. Yu, S. N. Lim, K. Patwardhan and N. Krahnstoever, "Monitoring, recognizing and discovering social networks," 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, 2009, pp. 1462-1469.

[9] Y. Zhang, L. Dong, S. Li and J. Li, "Abnormal crowd behavior detection using interest points," 2014 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, Beijing, 2014, pp. 1-4.

[10] Yeung, S., Russakovsky, O., Mori, G., & Fei-Fei, L., “End-to-End Learning of Action Detection from Frame Glimpses in Videos,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2678-2687.

[11] S. V. Tathe and S. P. Narote, "Real-time human detection and tracking," 2013 Annual IEEE India Conference (INDICON), Mumbai, 2013, pp. 1-5.

[12] Y. Wu, Y. Ye, C. Zhao and Z. Shi, "Collective Density Clustering for Coherent Motion Detection," in IEEE Transactions on Multimedia, Early Access, doi: 10.1109/TMM.2017.2771477

[13] B. Zhou, X. Tang, H. Zhang and X. Wang, "Measuring Crowd Collectiveness," in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, pp. 1586-1599.

[14] Yu-Chi Lai, Stephen Chenney, and ShaoHua Fan., “Group motion graphs,” In Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation, New York, NY, 2005, pp. 281-290.

[15] T. Wang et al., "Understanding Graph Sampling Algorithms for Social Network Analysis," 2011 31st International Conference on Distributed Computing Systems Workshops, Minneapolis, MN, 2011, pp. 123-128.