Ca' Foscari University

Department of Computer Science

MASTER THESIS

Multi-target Tracking Using Dominant Sets

Venice, October, 2014

By:

Yonatan Tariku Tesfaye

Supervisor:

Prof. Marcello Pelillo

Acknowledgments

First and above all, I thank JEHOVAH God, for everything that he has done for me, for providing me this opportunity and granting me the capability to proceed successfully.

This thesis appears in its current form due to the assistance and guidance of several people. Therefore, I would like to offer my sincere gratitude to all of them. Professor Marcello Pelillo, my esteemed teacher and supervisor, my cordial thanks for the opportunity to work this thesis with him, your guidance, comments, correction of the thesis and encouragement.

I would like to thank all my friends for their excellent friendship, joyful gatherings, advises, friendly assistance with various problems and all their supports during the whole study time and preparation of this thesis. Surafel Melaku, Eyasu Zemene and Tewodros Mulgeta thanks for your excellent assistance and kindly answers for my questions. Thanks also to all the members of Computer Science department for providing a wonderful teaching learning atmosphere in our department. I also need to appreciate the (financial) support of the University during my study.

I cannot finish without thanking my family. I want to express my deep thanks and love to my parents; my beloved father Tariku Tesfaye, my beloved mother Senait Arage, my brother Bereket Fisseha and my aunty Tizita Arage. For the trust, discussion and valuable advices, for your support during the whole study period. This is the fruit of your love, support, patience, guidance and prayers.

ABSTRACT

Tracking of people in a video is important for many applications. We present an approach to track multiple target/people in a standing pose or walking from a single camera. We imposed the notion of Dominant sets to track multiple target/people in both online and offline mode. Since it's unsupervised, given unlabeled patches of the people to follow, we then propose a formulation of the multi-target tracking problem as identifying dominant sets.

Our approach effectively tracks multiple targets in a sequence of frames from the video, it also manages to re-identify a target after a few absence from the video, moreover, we have got a very promising result in tracking a target with changing appearance.

Offline mode of our approach will be quit helpful in video forensics applications for investigating prerecorded video footage while the online mode could be useful in cases were real time knowledge is necessary that is when there is a need off processing stream of videos rather than prerecorded video, this mode has numerous applications like security, identifying people and so on.

Dominant set describe very compact structures, which ideally suites to represent the appearance of a given person in any number of frames as a one cluster. A dominant set is a form of maximal clique that can be applied to edge weighted graphs so that the affinity between all the nodes that are in the set is higher than those which are external to it. We used peeling off strategy to our work, which help us identify all dominant sets in a graph.

Here the data points are the detected persons (patches) in each frame. As we all know,

Videos are composed of frames and in each frame there are peoples to be tracked. And we used HOG (histogram of oriented gradient) people detectors to extract the patches. Then each of the detected patches will be treated as a graph node and there will be a similarity comparison between the nodes. In order to capture the individual similarities in people (patches) of similar target and differentiate between different targets, it is compulsory that the graph is made using meaningful and robust similarity measure. We tend to describe people patches with covariance matrix feature descriptors and we build the similarity matrix using distance among covariance in Riemannian manifolds. We finally performed an experiment on different video datasets and got promising good results.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

To ensure security, increasing number of surveillance cameras are placed on public areas over the recent years. As a result, the augmented quantity of information has resulted in the adoption of automatic systems of video analysis both for real time and for the a-posterior mining and reasoning on the extracted security information. In this way, huge variety of video data provided by installed cameras are automatically analyzed for event detection, object and people tracking, these actions offer a valid support to investigations and crime detection.

Following people in general is a challenging task: variations in the type of camera, in the lighting conditions, in the scene settings (e.g. crowd or occlusions), noise in images and the point of view must be accounted. Referring to the video forensics and surveillance fields and considering the tracking problem in its common definition, "problem to estimate the location of target objects in a sequence of images starting from an initial detection"[CCC11b], many different approaches have been proposed. In [YJS06] a large experimental survey of various tracking approaches is presented evaluating the suitability of each approach in different situations and with different constraints (e.g assumptions on the background, on the motion model, on the occlusions, etc.). The authors in [CCC11b] used a tracing system where only the people appearance is used this allows them to deal with different types of camera (e.g. fixed or PTZ) and various background conditions, since no motion information is used and no background foreground segmentation is performed. The bounding boxes of people in the scene are extracted by

a people detector and provided to the system as a set of labeled and unlabeled elements reinterpreting tracing as a Semi Supervised Graph based problem. In [MP13] Tewodros M. and M. Pelillo propose to exploit the graph transduction to track multiple people in videos, Given few labeled patches of the people to follow they propose a formulation of the multitarget tracking as a graph transduction problem.

In this thesis we tried to exploit Dominant set to track multiple people in videos. Since it's unsupervised, given unlabeled patches of the people to follow, we then propose a formulation of the multitarget tracking problem as identifying dominant sets. A dominant set is a form of maximal clique that can be applied to edge weighted graphs so that the affinity between all the nodes that are in the set is higher than those which are external to it[PP07]. we propose an application which is able to work in both pre-recorded video streams(off-line) and live streaming video (online). we tend to describe people patches with covariance matrices and we build the similarity graph using distance among covariance in Riemannian manifolds. This approach differs from the recent work done on people tracking [MP13] which is semi-supervised graph based approach which takes people in the scene as extracted by a people detector and provided to the system as a set of labeled and unlabeled elements reinterpreting tracing as a Semi Supervised Graph based problem. This approach faces few limitations, from the fact that graph transduction has a tendency to assign all the people detected to one of the labeled data (target) in other words, if a new person appear(never seen before) on the video, the system will try to incorporate it to one of the sampled labels but in reality they are two different persons.

Finally, I suggest Pavan and Pelillo's peeling off strategy [PP07] to our work, which help us identify all the dominant sets in a graph and i will try to show the reliability of the proposed solution by solving a real world problem.

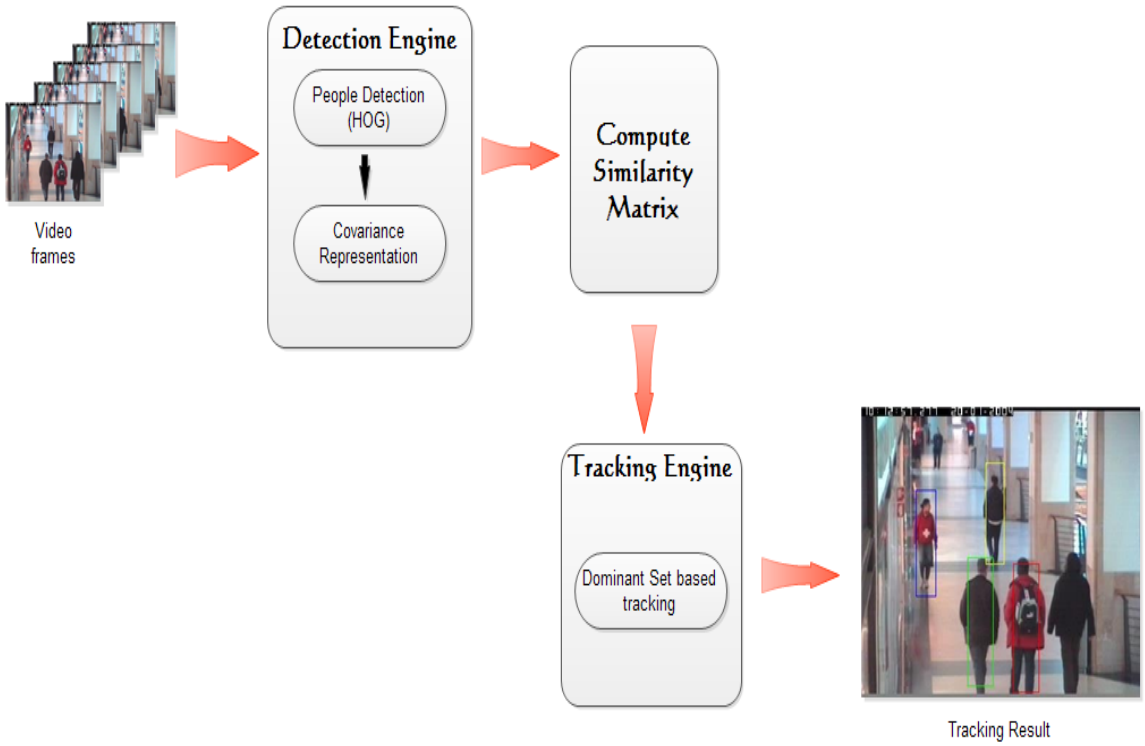


Figure. 1.1: Overview of the whole tracking algorithm

1.1 Motivation and our goal

The motivation for our work vitally comes up from the idea of applying the frame work of Dominant set for a famous problem in computer vision community known as multiple target tracking in the context of video surveillance.

As it is mentioned in the section that describes about the Dominant set frame work, it can powerfully deal with multi-class node clustering, so we thought about applying this frame work for multi target tracking by the idea that representing detected persons (patches) from the frames of the videos as a node of a graph and tend to describe people patches with covariance matrices and we build the similarity graph using distance among covariance in Riemannian manifolds.

1.2 Related Works On Object Tracking

This days object tracking mainly people tracking is becoming critical problem and many researchers around the world tried to implement varies kinds of approaches and methods. On this area [YJS06]presents a brief review of literature. This days everywhere we go, we can find different kinds of video cameras appearing for security, surveillance and for many more other reasons, as a result of the appearance of this cameras increases the availability of videos in plenty, to be meaningful and serve their purpose it is required to do several types of analysis on the videos. From the fact that all videos are composed of sequences of frames, the majority of techniques on video analysis that includes video annotation, object tracking classification and so on, all focused on the analysis of individual frame sequences.

Logically it is correct to think that people and their actions are the most important part of surveillance video, due to this on surveillance scenarios people tracking is getting more and more attention among computer vision community. People tracking, in general, is a challenging task. Difficulties in tracking people can arise due to abrupt people motion, changing appearance patterns of both the scene and the people, people-to-people and people-to-scene occlusions and camera motion. Numerous researches have been made and now it is possible to track single or multiple targets while some constraints like occlusion are kept. Some research works show that, in terms of accuracy tracking by learning has a prevailing result than tracking by prediction. When the target has unpredictable motion or when the target does not appear in most of the video frames at the same time neglecting the computational complexity of motion prediction.Despite the wide range of existing methods, most of the state-of-the art proposals are centered on Particle Filtering tracking [PP09],[BRL+09]. The underlining idea is that to represent the tracks association of the posterior density function by a set of random samples with

associated weights to compute estimates based on these samples and weights. Even though, the method has been proven to be robust, the model faces some difficulties in model representing objects under varying different kinds of lighting conditions, and noisy environment, leads to imprecise tracking results in complex scenarios.

Instead let us see one completely different perspective that is formulating the tracking problem as a graph partitioning problem, the idea is that to find path which is connected and link nodes that belongs to the same target. Recently in video surveillance this method is playing a critical role. In [MWS10a] they concentrate in investigations on post-incidence that is crime reconstruction using video data. For a complete construction of crime, the positions of all persons of interest should be known prior and also during the event. Here the tracking people by their appearance using the similarity matrix computed from the graph vertexes. However its application is constrained to only offline processing of videos. In another approach [CCC11a] they propose to exploit the transductive learning algorithm in combination with Riemannian similarity measure learning iteratively, frame-by frame, the target appearances are represented by Riemannian manifold. Here similarities between appearances are used to Expedite the classification of the unknown appearance by comparing it with training data set, the main downside of this technique is that, the tracking is strongly dependent on the quality of the training data set we have, that is the tracking accuracy will be very low if we use a noisy training set with a lots of errors for the target modeling.

Now, let's see few of the most recent works done on transductive people tracking in unconstrained surveillance, Coppi D. , Calderara S. and Cucchiara R. [CCC11b] they tried to solve the one target tracking problem using graph based approach that is semi-supervised transductive learning specifically by applying spectral graph theory. So here at the beginning limited labeled samples of the target will be given to the algorithm then the aim will be to search for the existence of the target in the upcoming frame

sequences of the video.

Since the technique iterates on each frame of the video sequence, target model update will be done in the middle of the process for the simple reason that targets in the video are continuously altering their appearances so it is critical to learn a new model for the target to achieve enhanced accuracy. In order to minimize the large amount of target model learning that occurs in each iteration, they performed clustering of the target models so that similar targets will be in the same cluster and only one representative target model will be chosen from every of the clusters in doing so they manages to save very large amount of storage space.

Another resent work done around people tracking is that masters thesis done by Tewodros Mulugeta and Marcello Pellilo [MP13] they followed a semi-supervised learning approach applied on multiple target tracking on both online and offline surveillance of videos as graph transduction based on the notion of game theoretic approach. Here the detected people from the video frames will be taken as data points, they used HOG(histogram of Oriented Gradient)based people detector to extract patches from each video frames and they will be treated as a vertices of the graph and they made similarity based comparison among vertices, as a result, similar patterns would be recognized from the given graph. Since they applied game theoretic notion to their framework, they formulated information propagation(transduction) in terms of a non-cooperative multi player game in which Nash equilibrium in a sense of consistence labeling of the data points. However the framework has got some shortcomings like; whenever a new target appearance in the middle of the video which is never seen before, their system will automatically consider it as one of exiting target and will assign it accordingly but in reality this person is different.

Chapter 2

Dominant Set Clustering

2.1 Introduction to clustering

clustering is effective in identifying and extracting different groups or compact sets which are different from other groups but similar among (within) each other in other terms clustering groups exhibit highest similarities with in the cluster than the elements from the outside of cluster.

Clustering problem appears in many areas like image processing, computer vision, bioinformatics, signal processing, medical imaging and others. The aim of clustering problem is to partition the given input, the set of n objects that can be arranged as an n by n matrix, in to different similar groups that satisfy some conditions.

We have two types of clustering problems in which they depend on the format of the input objects namely Central (feature based) and Pairwise (graph based). In the case of feature based clustering the input objects are given to the algorithm are represented in terms of feature vectors, so each object is represented as a point in an n dimensional space so that we can calculate some similarities between the points e.g. using Euclidean distance. Some of the algorithms of this type are Mean Shift, K-Means and others.

However there exist situations that applications where the feature based representation is not easily determined, in some cases the objects to be clustered are represented in Graphs. In these cases similarities between graphs have to be found, and there are

different ways to compute similarities between graphs.

Even though computing the similarity between arbitrary graphs is NP-hard, some instances of graphs can be computed in polynomial time. So even if extraction of feature based representation is not manageable, it is possible to get the similarity of objects. Pairwise clustering algorithm is applied in this case. A similarity matrix is given as an input and it tries to partition the data based on certain coherency criteria. As explained above this algorithm accepts similarity matrix as an input, regardless of how the data is represented. It could be vector,graph or any other kind of representations.So the second algorithm(pairwise) is more general than the first one(feature based) since it accepts the above all representations.

Let's formalize what a cluster is: The notion of a cluster is defined on the bases of two criteria[PP03]: **Internal criterion** which represents the inter cluster similarity of objects and the **External criterion** that deals with the dissimilarity of the objects outside the cluster and the inside one. So, the above two criteria should satisfy by a given cluster.

Let's see the following example: From the following figure 2.1 we can see that we have 2 clusters and the objects that are bounded using the curve can't form a single cluster as the group despite the fact that it satisfies the internal criteria it does not satisfy the external criteria since a larger coherent group contains it. In simpler terms we can define clustering in this way.

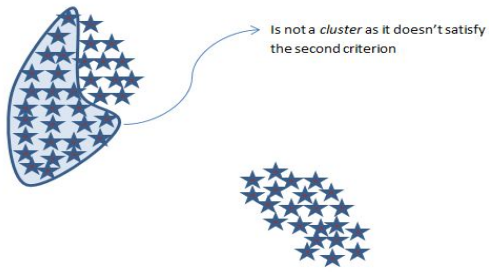


Figure. 2.1: Cluster example

2.2 Dominant set clustering

Pelillo and Pavan [PP03] first introduced the notion of dominant set in 2003. It is actually a combinatorial concept in graph theory that generalizes the notion of the maximal clique to an edge-weighted graph.

Generally when we use the pairwise clustering mechanism, we denote the objects to be clustered as an undirected edge weighted graph where the n given points are represented by the vertices, and weight between the adjacent nodes (edges) represent similarities between neighbor (edges). finally the graph will be presented as an n $\times$ n similarity matrix since we got n number of nodes so the values in the matrix represent the similarity between the corresponding nodes in the row and column. For instance if we represent our matrix with W and the similarity between vertex i and j will be represented in $w_{i,j}$ and also since in our original graph we don't have self loops means a link connected to itself so our similarity matrix will have zero diagonals.

Know let's start from the simple case which is binary case means our matrix will have only (0,1)values in other terms the value in between will not be considered, either they are similar or not. Here the graph is undirected un weighed graph. The kind of structure in this graph that qualifies both internal and external criteria is from a very classic notion of graph theory which is the notion of a `Maximal Clique`. Here let as say a few things

about clique and see its meaning in a graph.

`clique` is formed when there is mutually adjacent vertices exist. `Maximal Clique` is when a clique is not part of a larger clique or not contained in any larger clique. Furthermore there are two more notion `Maximum Clique` and a `Strictly Maximal Clique` that comes from the need of the stability of the set.

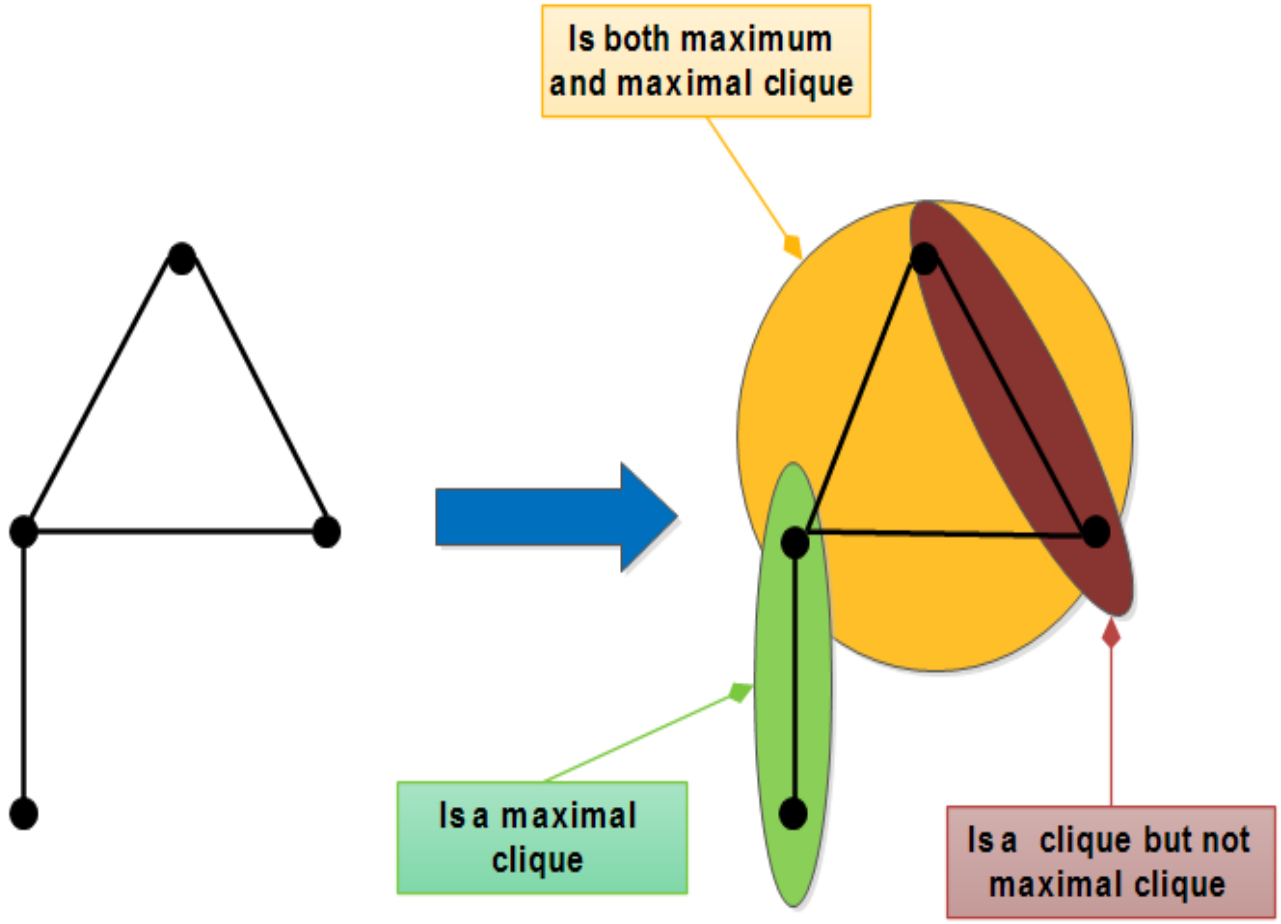


Figure. 2.2: Maximum and Maximal clique example

For a clique to be called `maximum clique` it has to have largest cardinality. A maximal clique is a clique that cannot be extended by including one more adjacent vertex, meaning it is not a subset of a larger clique. A maximum clique (i.e., clique of largest size in a given graph) is therefore always maximal, but the converse does not hold. More over

for a `clique` to be called `strictly maximal clique` which is stable set it must satisfy the following criteria if a graph is maximal clique, all the vertices outside it can't have a number of edges, incident on its vertices, which is more than one less the cardinality of itself.sometimes maximal cliques are unstable, when one of the vertex is dropped from the maximal clique and if one vertex is added from outside, a new maximal clique will be formed. lets see this instability in the next figure 2.3:

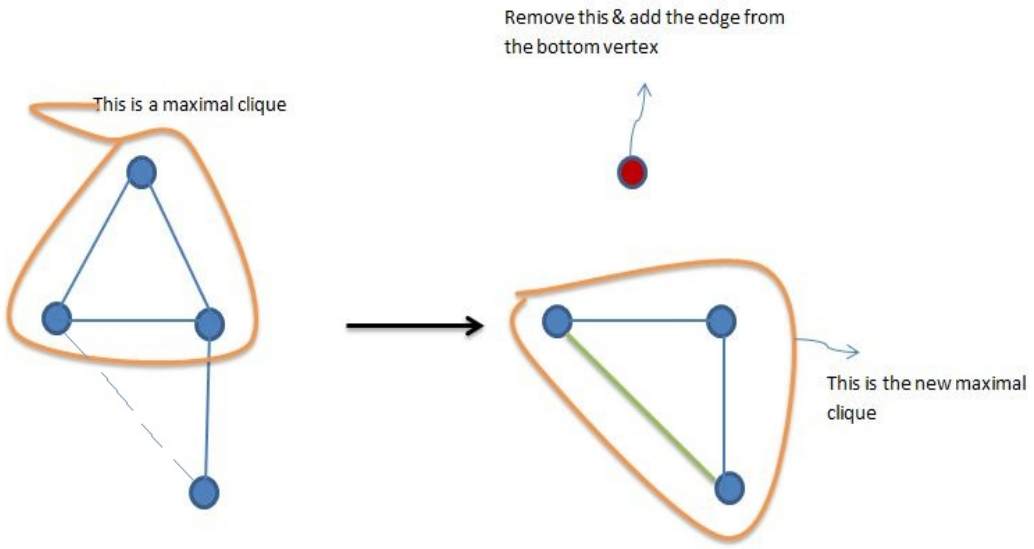


Figure. 2.3: Unstable Cluster example

The notion of maximal clique coincides with the notion of a cluster in the case of binary similarity this implies it satisfies both criteria. Nevertheless, this does not work in the case of weighted graph. So here the question is how can we generalize the maximal clique notion in to an edge weighted graph. Now it is time to bring the notion of dominant set in to the picture. When we talk about dominant set which is combinatorial concept in graph theory, the primary goal is to generalize the notion of maximal clique to an edge weighted graph. let's discus some ideas and definitions of the dominant set prior to the main definition of it given by Pelillo and Pavan [PP03],

Definition The sum of the weights (the similarities) of the edges that connects the point

to all other points in the set divided by the cardinality of the set gives us the average weighted degree of a point(AWDeg). Mathematically the average weighted degree of a point i with respect to a set of vertices S that is not an empty set is expressed as

$$AWDeg_S(i) = \frac{1}{|S|} \sum_{j \in S} w_{i,j}$$

Where $w_{i,j}$ represent the weight between the two vertex i and j

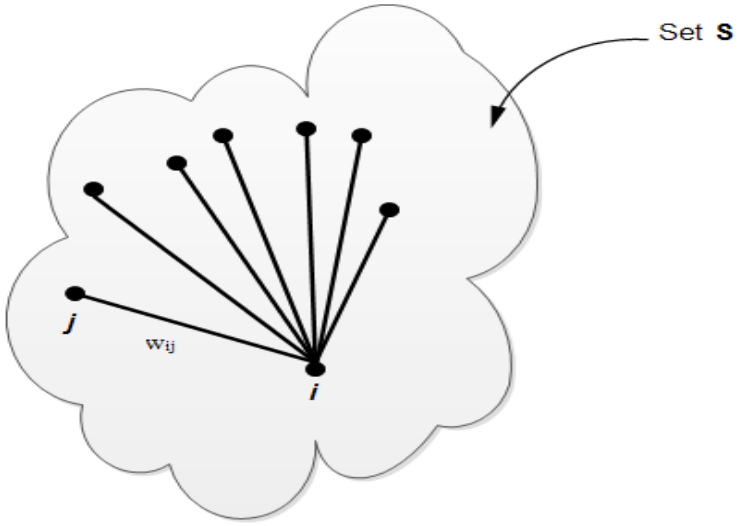


Figure. 2.4: Average weighted degree of point i

Here, it represent the average weighted similarity between point i and a point in a set S.

The relative similarity, $\phi_S(i, j)$, between two vertex(points),i and j(where j is not an element of the set s)is measures the similarity between node i and j with respect to the average similarity between nodes i and its neighbors in set S.

$$\phi_S(i, j) = w_{i,j} - AWDeg_S(i)$$

Here the value of the $\phi_S(i, j)$ could varies between negative value and positive depending on the value of the absolute similarity. If the absolute similarity is less than the average weighted similarity, the value will become negative and vice versa.

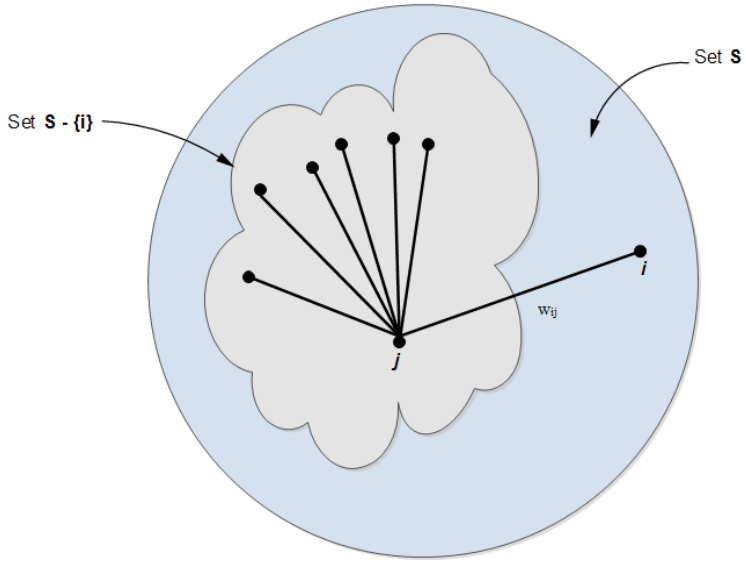


Figure. 2.5: Relative similarity between two objects

If the cardinality of the set S is 1 then by definition $W_S(i) = 1$(the weight of i with respect to S). Otherwise we have to add all the relative similarities between i and the rest of nodes in the set S, and with this we can see how closely related is point i with respect to the rest of the points in S.

$$W_S(i) = \sum_{j \in S \setminus \{i\}} \phi_{S \setminus \{i\}}(j, i) W_{S \setminus \{i\}}(j)$$

So we can compute the weight of the set S by summing up each weights $W_S(i)$ at this point we know that $W_S(i)$ evaluates how closely the vertex is linked with other sets of nodes. Before we jump in to the main definition of dominant set lets see some examples:

Example: The following graph has three vertices and lets try to calculate the weight.

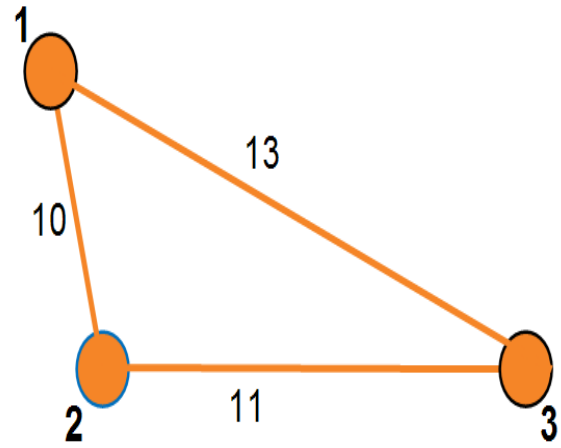


Figure. 2.6: Example on weighted graphs

Solution: Let's first simply see the weights of the vertices and see who got biggest weight and the smallest to do so we can add the edge weights of the links (directly connected) to that node like: for node 1 the edges weights directing to it are 10+13=23 similarly, we calculate for the rest of the verteces i.e 11+13 > 10+13 > 10+11. This shows that $W_{\{1,2,3\}}(3) > W_{\{1,2,3\}}(1) > W_{\{1,2,3\}}(2)$ Now lets use the formal formulas and see the results.

Bare in mind these points:

$$W_{\{i,j\}}(i) = W_{\{i,j\}}(j) = \omega_{i,j}$$

$$\phi_i(i,j) = \omega_{i,j}$$

$W_{\{i\}}(i) = 1$ is true by definition when the cardinality of the set S is one

$$\begin{aligned} W_{\{1,2,3\}}(1) &= \phi_{\{2,3\}}(2,1)W_{\{2,3\}}(2) + \phi_{\{2,3\}}(3,1)W_{\{2,3\}}(3) \\ &= (\omega_{2,1} - AWDeg_{\{2,3\}}(2))\omega_{2,3} + (\omega_{3,1} - AWDeg_{\{2,3\}}(3))\omega_{2,3} \\ &= (10 - 11/2)11 + (13 - 11/2)11 \\ &= 132 \end{aligned}$$

following the same procedure we compute $W_{\{1,2,3\}}(2) = 104$ and $W_{\{1,2,3\}}(3) = 140$. To

calculate the overall total weight of S, W(S), we add up the above 3 results we got which are $132 + 104 + 140 = 376$.

Now let us make some experiment and see the effect of adding a new vertex to the existing graph. Finally, check the sign of the weight of the vertices (including the newly added vertex) which exhibits the overall similarity of this new vertex with respect to the previous vertices. So, lets see the effects of adding vertices 4 and 5 to our original graph as seen in the figure below.

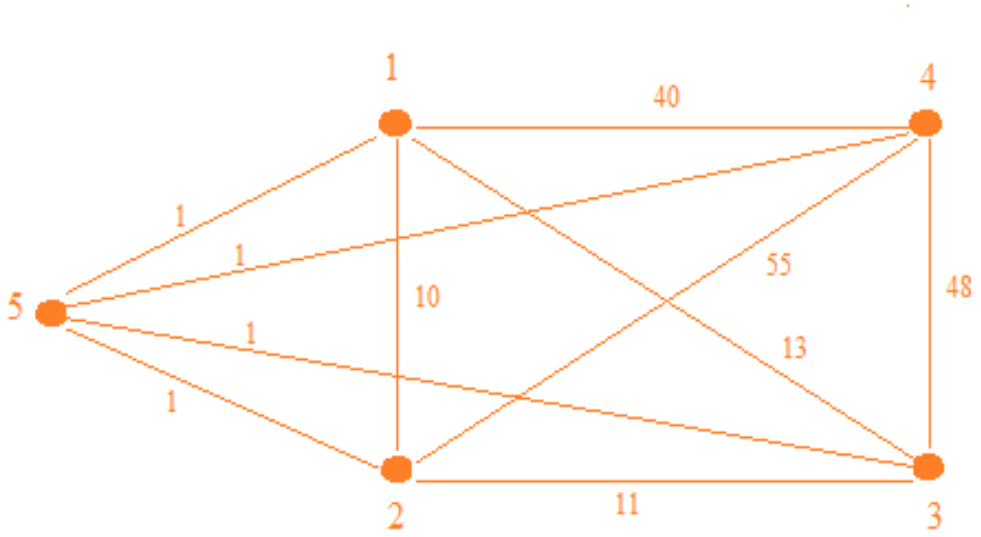


Figure. 2.7: Node 4 and 5 added to the previous Graph

Here we can easily see that vertex 4 is highly similar to the nodes 1,2 and 3 since the weight between them is higher so $W_{\{1,2,3,4\}}(4) > 0$ But on the contrary the other node added that is node 5 is loosely bonded with the existing graph so $W_{\{1,2,3,5\}}(5) < 0$ so from all this we can observe that adding vertex 5 to the graph will weaken the overall weight of our graph. Now it is time to see the formal and the main definition of dominant set given by Pellilo and Pavan.

Definition, Pelillo and Pavan [PP03]: A non-empty subset of vertices $S \subseteq V$ such that $W(T) > 0$ for any non-empty $T \subseteq S$, is said to be dominant if:

1. $W_S(i) > 0$ for all $i \in S$
2. $W_{S\cup\{i\}}(i) < 0$ for all $i \notin S$

The above two criterias are quite similar with the conditions of the clustering criteria. So now we are in a good position to say that both notions of clustering and dominant set coincides having said this it is time to question how to compute dominant set? Or in other words how can we partition a given data in to dominant set? To find dominant set instead of using standard algorithms Pavan and Pelillo they transform the purely combinatorial problem of finding a dominant set in a graph in to (continuous) quadratic optimization problem and this allows the use of straightforward dynamics from evolutionary game theory to determine them.

2.3 Dominant Set and the Quadratic Optimization

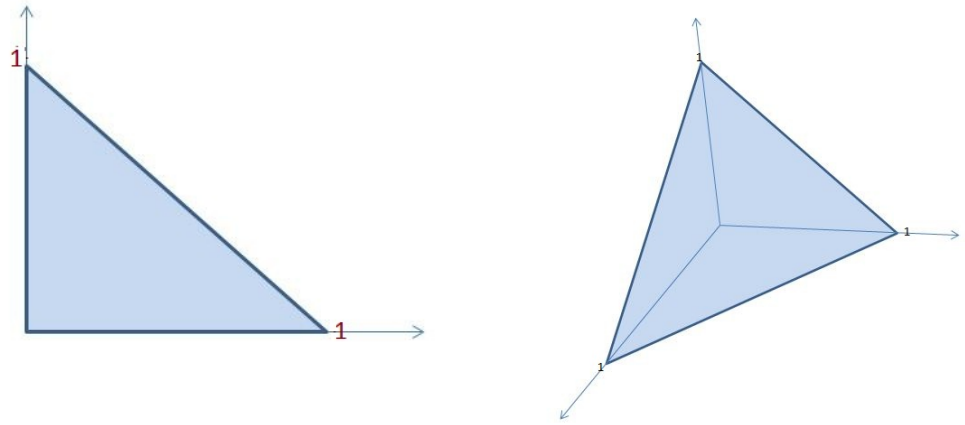
Now, since we are clear with the notion of dominant set, lets do some calculations so that given a set we will make partition and find clusters. As we mentioned above Pavan and Pelillo they transform the purely combinatorial problem of finding a dominant set in a graph in to a pure quadratic optimization problem and use evolutionary game theory dynamical system to solve the optimization problem.This problem is the general form of the well known problem from graph theory, Motzkin-Straus problem[SM65]. Now lets try to exploit what this theory states and then after we will see generalization of the problem.

Motzkin-Straus theorem *says given an undirected un-weighted graph G=(V,E) and W is the adjacency matrix of the graph. There is a one-to-one correspondence between the clique number of the graph $\omega(G)$ and the maximal optimizer of the problem:*

$$\begin{aligned} &maximize && \mathbf{f(x) = x^T W x} \\ &\text{subject to } && x \in \Delta \end{aligned}$$

If $\mathbf{x}^*$ *is the maximizer, then* $\omega(G) = \frac{1}{1-f(x^*)}$

The standard simplex(Δ) is a simple geometrical structure which satisfies these criterias; all the values of x_i should be either grater than or equal to 0 and if we sum up all x_i we should get 1. Lets see the pictorial representation of standard simplex with n equals to two and three



(a) (b)

Figure. 2.8: (a)The standard simplex when n=2 (b)The standard simplex when n=3

Few scholars followed different approach to solve clustering problem for instance S. Sarkar and K. L. Boyer [SB98],uses different quadratic program even though it is quit similar with the above mentioned one. To solve spectral clustering problem, they tried to find the biggest eigenvalue and the associated eigenvector of the given similarity matrix W. In order to do so they maximized this quadratic problem.

$$\textit{maximize} \quad \mathbf{x^T W x}, \textit{ subject to } x \in S(\textit{theSphere}) \textit{ (that is } x^T x = 1\textit{)}$$

We can easily see that the main objective function is similar but the different lice on the domain. Using this technique has few disadvantages like since it only considers the positive eigenvalues in addition to this finding the maximum value is NP hard since it is simulated to the maximum clique.

Here, we are going to see one definition prior to the theorem which creates 1-to-1 correspondence between dominant set and strictly local maximizer of the quadratic program $x^T W x$ over standard simplex. The following theorem generalizes Motzkin-Straus problem.

Definition: *A weighted characteristic vector is a vector in the standard simplex that can be defined as follows:*

$$x^S = \begin{cases} \frac{W_S(i)}{W(S)} & if\ i \in S; \\ 0 & otherwise. \end{cases}$$

Theorem,Pelillo and Pavan: *If S is a dominant subset of vertices, then its weighted characteristic vector x^S is a strict local solution of objective function in the standard simplex.*

Conversely if x^* is a strict local maximizer of the objective function in the standard simplex then its support

$$\sigma = \sigma(x^*) = \{i \in V : x_i^* \neq 0\}$$

is a dominant set provided that $w_{\sigma \cup \{i\}} \neq 0 \quad \forall i \neq \sigma$

The characteristic vector has elements equal to the number of vertices that exist in a graph in which the i^{th} value will equal to zero whenever $i \notin S$, otherwise as mentioned above in the theorem it takes the ratio. To end up at a point in the standard simplex we can add up all its elements. Since they are the ratio of non-negative numbers it is easy to see that all the components of the vector are also non-negative. So in the cases where S happens to be dominant set, this vector is the strict local maximizer of $x^T W x$ in the standard simplex. If we are told and given that x^* is the strict local maximizer of the above objective function in standard simplex, by considering only the support of the vector which correspond to the subset of nodes in a given graph which correspond to dominant set. Here dominant sets are putted as one of continuous form of optimization problem. Now it is time to analyze the mechanism to solve the optimization problem in order to find the dominant set. From the handful of methods that use to solve the problem here we decided to use the dynamic system equation, replicator dynamics, which evolves from the evolutionary game theory. Replicator dynamics help us find local solutions of the program, a class of continuous and discrete-time dynamical systems arising in evolutionary game theory. In our work, we used the following model:

$$x_i(t+1) = x_i(t) \frac{(AX)_i}{X(t)^T A X(t)}$$

for *i = 1 ... n*, which corresponds to the discrete-time version of first-order replicator equations. All trajectories which are starting with in the standard simplex $\triangle$ will remain in the simplex $\triangle$ for all future times.

Moreover, it can be proven that, since we consider symmetric matrix A , the our objective function $f(X) = X^T AX$ will increase strictly along any non constant trajectory of the above formula (formula for replicator dynamics for discrete time) its asymptotically stable points are in one-to-one correspondence to strict local solutions, in turn, correspond to dominant sets for the similarity matrix A.

Definition: *A Nash equilibrium is an Evolutionary Stable Strategy(ESS) if for all strategies y*

$$y^T Ax = x^T Ax \text{ implies } x^T Ay > y^T Ay.$$

Let us put it in other terms, assuming that both x and y are Nash equilibrium, here if we end up with result(score) in cases like y is playing against its opponent and when x plays with him self, by changing the role of y and x we end up $x^T Ay > y^T Ay$. In cases when strategy y is equally good strategy as x and if we change the role of them and if my opponent choose to play y, it is advantageous if i play x against y rather than playing strategy y. The above mentioned condition is that allow to make Nash equilibrium stable and resistance to any small perturbation in the component of the vector. So the notion of Evolutionary Stable Strategy is the notion that we are searching for our purpose of clustering.

Both the internal and the external criteria's of clustering are satisfied by the evolutionary stable strategy. In a doubly symmetric game that is $A = A^T$ in our optimization problem, we see that both notions of Nash equilibrium and Evolutionary Stable Strategies coincides in the standard simplex Δ. As Nash Equilibrium is local maximizer of $x^T Ax$, the Evolutionary Stable Strategies is the strict local maximizer of $x^T Ax$. Now at this point we can give identical definition for both symmetric and non-symmetric cases of dominant set, so we can generally say both notion of Evolutionary Stable Strategy and dominant set coincide.

Theorm, Pelillo & Pavan: If I take an Evolutionary strategy, then I take the support of the set and this is a directed dominant set. Or we can say if I take the dominant set and if I set a characteristic vector this is an Evolutionary Stable Strategy.

Replicator dynamics accepts and performs well both on non symmetric and non positive matrix. For this reason we can not say that at each step weather the algorithm is maximizing or minimizing something but rather one can state 2 points: (1) as it was stated in the theorem below that Nash equilibrium has 1-to-1 correspondence with limit point of the trajectory that begins from any where from the interior. This means from a point located inside if we start replicator dynamics, by the time it converges it will give as Nash equilibrium.(2) If we take a point in standard simplex which is ESS then it is asymptotically stable. So when algorithm converges in general we can say it is an Evolutionary stable state. Lets see one theorem which generalizes it.

Theorem,Pelillo & Pavan: *A point $x \in \Delta$ is the limit of a trajectory of the replicator dynamics starting from the interior of the standard simplex iff x is a Nash equilibrium. Further, if point $x \in \Delta$ is an ESS then it is asymptotically stable.*

Chapter 3

Our Contribution

Our main contribution in this work is we applied the notion of Dominant set which is a generalization of a maximal clique to edge weighted graphs Proposed by Pavan and Pelillo [PP07] for a well-known computer vision problem that is multiple target tracking in videos. We have studied the applications of this framework for tracking multiple people on video surveillance scenario and got some promising results.

let's review our work section by section. In the first section we will try to take one toy example and show how we first represent people (patches) from video frames in a graph. And then on the second section we will see how to represent patches/people from the video with covariance matrix, then after in the third part we will see how to compute the similarities between the covariance matrices (which represent vertices/patches) in the graph and formulate similarity matrix. Finally, we will discuss how the notion of dominant set is used to track multiple people in a video.

3.1 Graph Theoretic Definition of people tracking

Here we will see how the notion of dominant set can be used to track multiple targets on a given surveillance videos. But first we need to construct a graph from the videos that needed to be processed and also we need to compute the similarity between nodes (people).

Since videos are consists of multiple frames each frame is consists of many objects in

it like: it could be cars, houses, people, animals and so on but we are interested only on the people. This detected persons (patches) from all sequences of frames present in the video will be represented as a node on the graph. If the same person is doing any type of trajectory or it could be in a stationary state, in which his or her geographical position is in the coverage area of any video surveillance will appear in the consecutive frames, due to this when we perform the transformation of the video to a graph model, the same person could be represented to multiple node in the newly formed graph.

Let us take one toy example and try to construct the graph from a given video frames: now let's say we have the following two hypothetical frames in our video.

Figure. 3.1: Frames

As we can see in the above two frames of the video we got five different persons appearing in both of the frames but the geographical positions of the people in both frame is different than that of the first one. The very first thing that we have to do is we have to extract the persons from each frame using any people detector out there since our framework is independent of which detector we used, In our case we used HOG

(Histogram of Oriented Gradient) [DT05]. In this case HOG detected four of the five persons appearing on each frames so, at this point we have a total of eight patches (persons) detected from both frames, four patches each as shown in the diagram below:

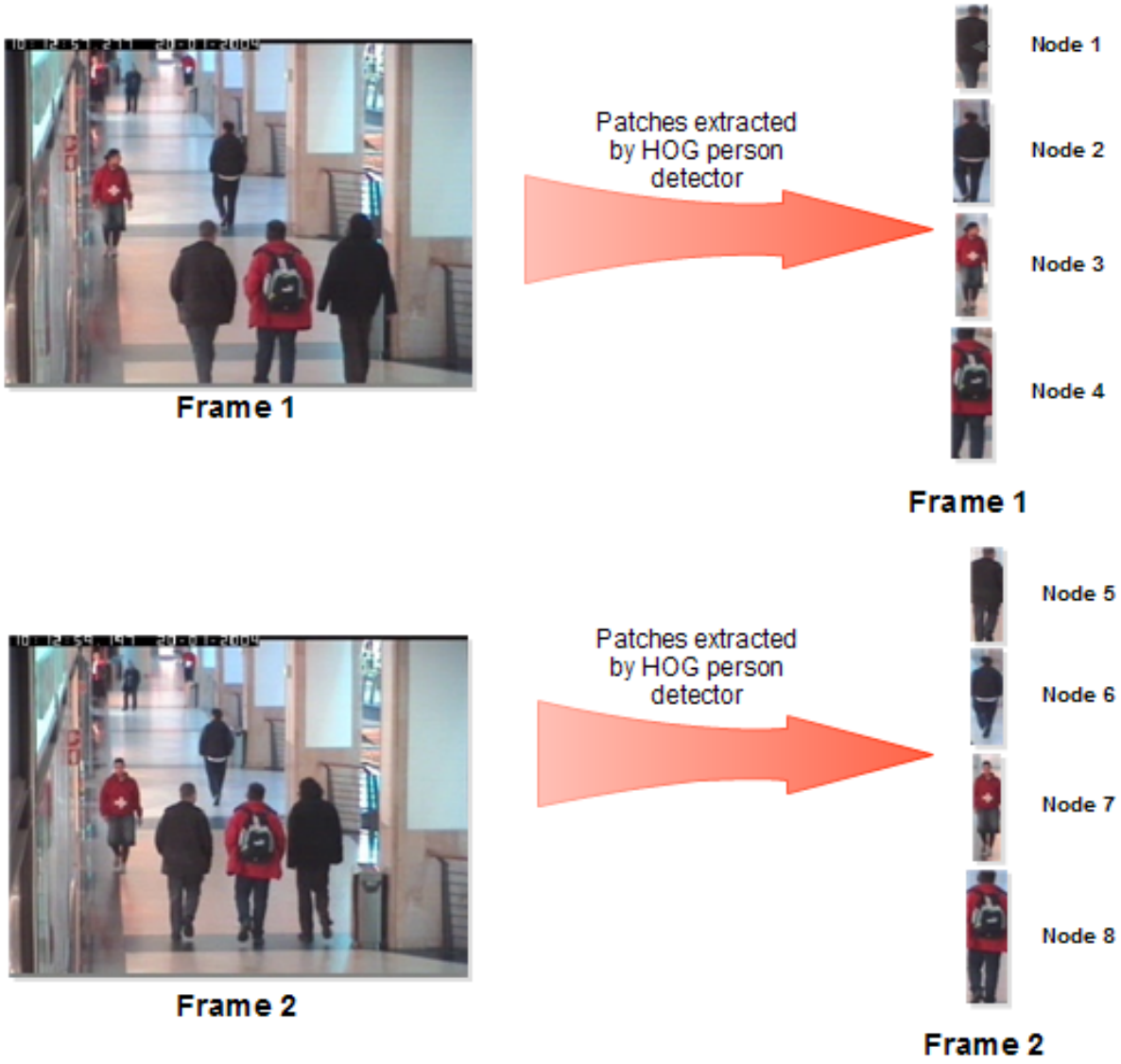


Figure. 3.2: Extracted Patches from each frame

The graph representation of the above two frames, each with four patches that are extracted by the HOG people detector will look something like in the figure below. Each patch can be represented as a graph $G = (V, E, w)$ with a set of vertices V,

edges E, and a positive edge weight function w. In this case, the vertices or nodes are patched extracted from all the images composing the video, E corresponds to the set of connections between the peoples (patches) and w represents the affinity (similarity) measured using some extracted features between each pair of patches in the video. We can express the relationship between all the nodes (patches) of each frames from videos by a weighted affinity matrix $\mathbf{A}$ such that each of its elements $a_{ij} = w(i, j)$.

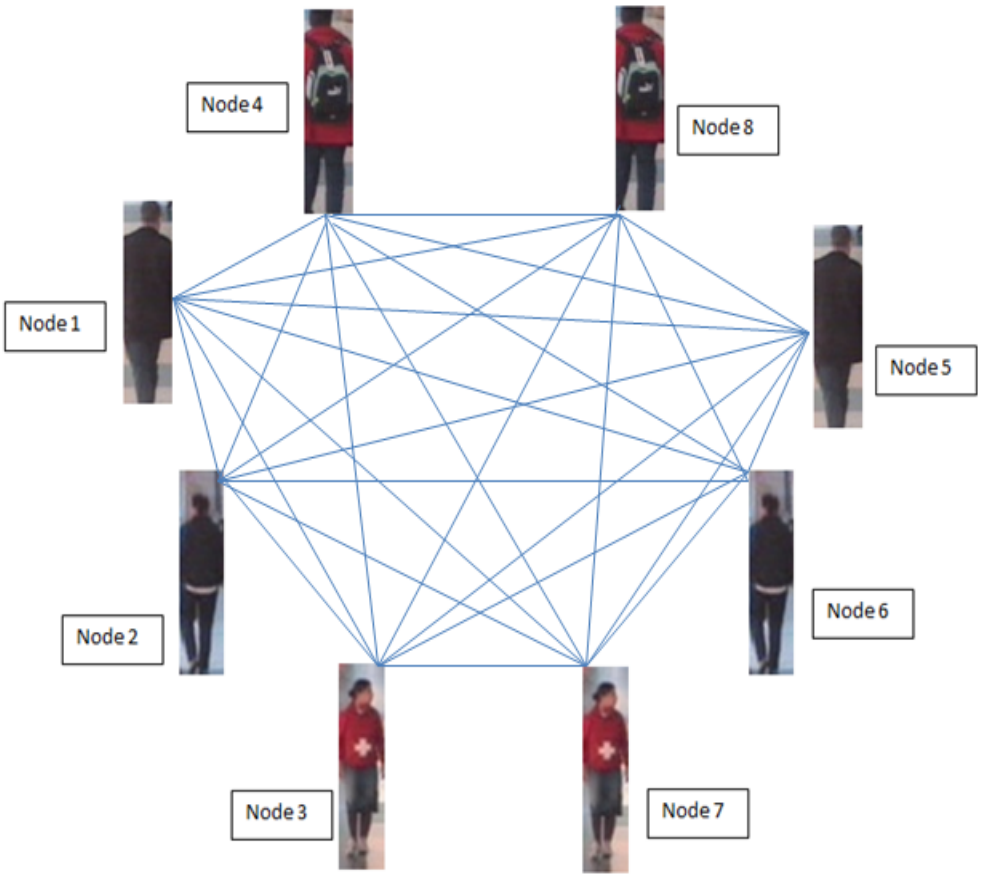


Figure. 3.3: The Constructed Graph using patches

The similarity between each patches can be seen from the weighted edges. A higher degree of similarity corresponds to higher edge weights between two patches. Similarly, lower edge weight implies low degree of similarity. If the edge weight is zero, that means there exist no similarity between the two patches at all.

So, when we see the figure above the similarity between node 1 and 5 is higher for the obvious reasons that this two nodes represent the same person in different frames, similarly the affinity between node 2 and 6 as well as node 3 and 7 and also node 4 and 8 is higher since they also represent the same person. So, since dominant set represents a very compact structure, ideally suites to represent the appearance of a given person in any number of frames as a one cluster. So here comes the main questions, how do we represent a node and compute the similarity between them? These questions will be answered in the next sections.

3.2 Covariance Representation of Nodes

It is vital that we represent each detected patches by some kind of model with features so that by taking the differences in their descriptor we can compute similarity matrix between them. In order to capture the individual similarities in people (patches) of similar target and differentiate between different targets, it is compulsory that the graph is made using meaningful and robust similarity measure. Color histogram is the simplest descriptor to represent patches. Nevertheless, due to the fact that information like shape and location won't be taken in to consideration by the method, it faces some shortcomings like it will not differentiate if two peoples wear the same color but not in the same position.

So, we decided to model people appearance using covariance matrix feature descriptors,[PTM06]. This representation has been adopted in multiple approaches [CCC11b], [MWS10b], [LLS10],[MP13] because of its robustness in capturing information like color, shape and location and also it is scale and rotation invariant.

The another reason of covariance matrix usage in object representation relies on the very fact that generally a matrix extracted from one region is sufficient to match the

region in numerous views and poses, because the noise corrupting individual samples is essentially filtered out with the average filter throughout covariance computation. In addition to that, covariance matrices have scale and rotation invariance property and are independent to the mean changes such as identical shifting of color values.

Considering d different selected pixel features extracted independently from the image window size, benefiting from being a low dimensional representation, the resulting covariance matrix C is a square symmetric matrix $d \times d$ where the diagonal entries represent the variance of each feature and the non-diagonal entries represent the correlations.

Let's considering Im as a color image with three dimensions and Y be the $W \times H \times d$ dimensional feature image extracted from Im.

$$Y(x, y) = \rho(Im, x, y)$$

Where the function ρ could be any mapping such as gradients, color, filter responses, intensity, etc. Let $\{t_i\}_{i=1...M}$ be the d-dimensional feature points inside Y, with $M = W \times H$. The image Im is represented with the $d \times d$ covariance matrix of the feature points:

$$C_R = \frac{1}{m-1} \sum_{i=1}^{m} (t_i - \mu)(t_i - \mu)^T$$

where each point in a region R is represented by a vector of the means of the corresponding features by μ.

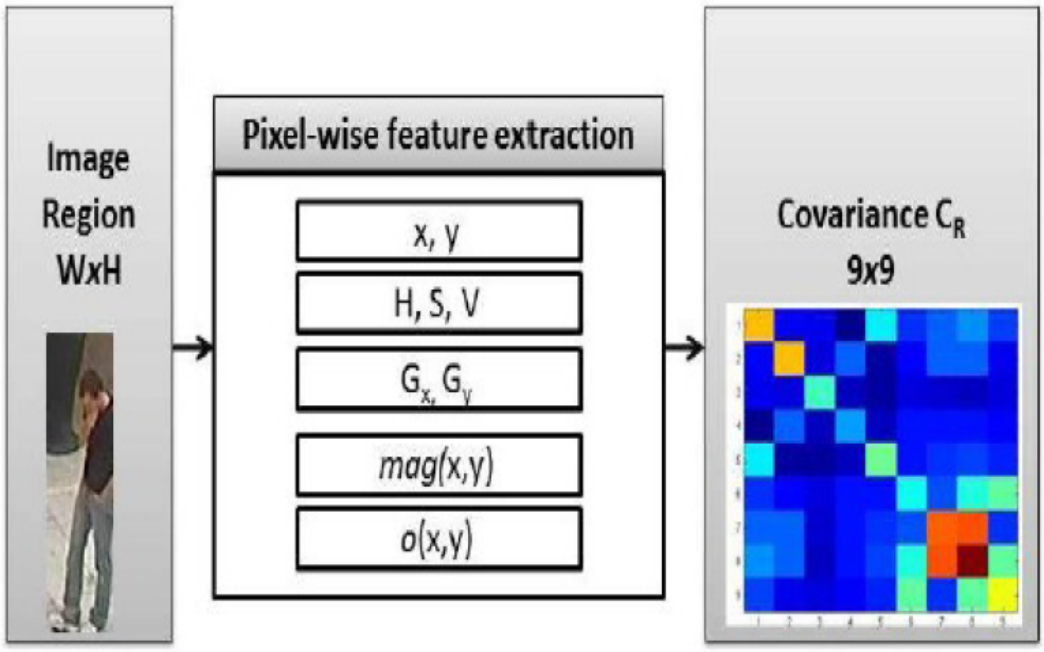


Figure. 3.4: Representation of a patch by Covariance matrix. Adopted from [PTM06]

In our case, we decided to model each pixel within a people patch with its HSV intensity values, its position (x, y), G_x and G_y are the first order derivatives of the intensities calculated through Sobel operator with respect to x and y, the magnitude $mag(x, y) = \sqrt{G_x^2 + G_y^2}$ and the angle of the first derivative $0(x, y) = arctan\left(\frac{G_y}{G_x}\right)$. So finally each pixel of the people(patch) is mapped to a 9-dimensional feature vector:

$t_i = [\ x\ y$ H S V $G_x\ G_y$ mag(x,y) $o(x, y)]^T$

Based on this 9-dimensional feature vector representation the covariance of a patch is 9×9 matrix.

It ought to be noted that we have chosen to used HSV color space rather than using the basic RGB space as a result of the experiment done by a group of researchers from Modena university, HSV components has higher invariance to light and scale changes than RGB [CCC11b].

3.3 Building The affinity matrix

It is vital that the graph is made using a meaningful and robust similarity measure capable of capturing individuals similarity in patches of identical target and differentiate between different targets.

In order to build the affinity matrix A, the distances between covariance matrices are necessary and since they do not lie on the Euclidean space. We used a technique proposed in [FM99] which is equal to the sum of the squared logarithms of the generalized eigenvalues. Formally the distance between two matrices C_i and C_j is expressed as:

$$\gamma(C_i, C_j) = \sqrt{\sum_{i=1}^{d} ln^2 \lambda_k(C_i, C_j)}$$

Where $\lambda_k(C_i, C_j)_{k=1....d}$ are the generalized eigenvalues of C_i and C_j which is computed as follows $\lambda_k C_1 x_x - C_j x_k = 0$ where $k = 0....d$ where x_k are the generalized eigenvector. It is proven that ρ satisfies the metric axioms, triangle inequality, symmetry, positivity, for positive definite symmetric matrices.

At this step, we have the similarity matrix between each detected patches. So now we have a graph with all information needed to carry on with our style of Dominant set detection.

3.4 People Tracking as Dominant set

As already mentioned, people tracking by its definition is "problem to estimate the location of the target object person in a sequence of images starting from the initial detection"[CCC11b]. Therefore, the mutual affinity between all the detections (patches) of the same target/person appearing on different frames should be higher than the affinity between any other persons (patches) detected. As already discussed in the above section Pavan and Pelillo [PP07] proposed a different way of thinking about a cluster as a dominant set, which is a generalization of maximal cliques to edge-weighted graphs. If we consider a subset $\mathbf{S}$ of the set of nodes $\mathbf{V}$ in graph $\mathbf{G}$, the average weighted degree of a vertex $P_{1i} \in \mathbf{S}$ with respect to set $\mathbf{S}$ is

$$AWD_s(P_{1i}) = \frac{1}{|s|} \sum_{p \in s} a_{p_{1i}p}$$

Note that in this definition of the degree of p_{1i}, the value is strictly related to only a subset of the graph $\mathbf{V}$. Ideally, $\mathbf{S}$ defines a semantically meaningful local context such as the tracking of a person P_1 at frames j, k, n (i.e P_{1j} = means the appearance of person one at the j^{th} frame and so on) in the figure below

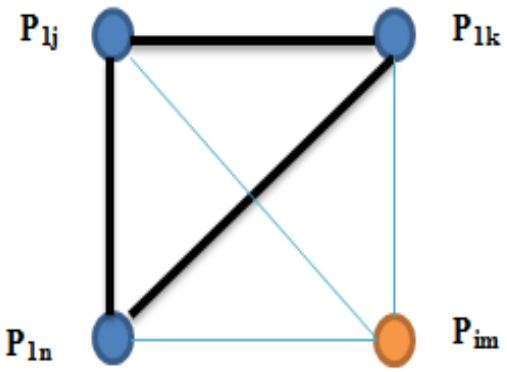


Figure. 3.5: Example of how the patches will be represented in a graph. The bold line represent high affinity while the thin line represent low affinity

So the relative affinity between node $p_{im} \notin s$ and P_{1j} where p_{im} means the appearance of person i at the m^{th} frame

$$\phi_S(p_{1j}, p_{im}) = a_{p_{1i},p_{im}} - AWD_S(p_{1j})$$

and the weight of each P_{im} (person i at frame m) with respect to a set S is defined recursively as:

$$W_S(p_{im}) = \begin{cases} 0 & if\ |s| = 1; \\ \sum_{p \in s \setminus \{p_{im}\}} \Phi_{s \setminus \{p_{im}\}}(p_{im}, p) w_{s \setminus \{p_{im}\}}(p) & otherwise. \end{cases}$$

Intuitively, $W_S(p_{im})$ gives us a measure of the overall similarity between vertex p_{im} and the rest of the vertexes in S, weighted by the overall affinity of the vertices in $s \setminus \{p_{im}\}$ with positive values indicating high internal coherency see figure 3.5.

Therefore, $w_{p_{1j},p_{1k},p_{1n}}(p_{1j})$,$w_{p_{1j},p_{1k},p_{1n}}(p_{1k})$,$w_{p_{1j},p_{1k},p_{1n}}(p_{1n})$ are all would be very high since the affinity between this nodes is higher from the fact they (nodes) all represent the appearance of the same target/person in different frames where as $w_{p_{1j},p_{1k},p_{1n},p_{im}}(p_{im})$ would be very low since the similarity between node p_{im} and all the others is very low.This relationship between internal and external nodes of a dominant set S is defined formally using the conditions $W_S(i) > 0$ for all $i \in S$ and $W_{S\cup\{i\}}(i) < 0$ for all $i \notin S$

Therefor, Dominant set describe very compact structures, which is ideally suites to represent the appearance of the same target/person in any number of frames as a one cluster.

Chapter 4

Experiments and Experimental Results

In this section, we evaluate the performance of our people tracker, which is able to track multiple targets at a time. We evaluated the degree of reliability of our approach by testing it on video datasets CAVIAR (clips of this dataset are collected in the hallway of a shopping center in Portugal) and 3DPes (clips taken from monitoring a section of the campus of the University of Modena and Reggio Emilia). In our setting we are provided the patches (i.e. the detected people on the scene from each video frames) that is obtained using the HOG based people detector then after we transformed the video sequences to graph models where the nodes representing the detected patches. We computed the similarity matrix by applying Gaussian kernel on the distance (i.e the distance between the covariance matrices which lie in the Reimanian space) that exist between the nodes. We used this matrix as affinity matrix for our set up of finding Dominant set. Next step we run dominant set and identify all the dominant sets (which represents appearance of the same target/person on different frames) from the graph using the pill off strategy after that we call a routine which marks colored rectangle boxes on the original frames of the video to indicate the tracked targets with different color so that we get the annotated frames as an output.

4.1 Evaluation Measures

Here in this work, we used evaluation measures like precision, recall and Accuracy of the classification of each patches. The number (class) assigned to each patch by the system will be compared with the actual class.

True Positive : is when a patch is classified correctly with its label.

False Positive : is when a patch is miss-classified (i.e the i^{th} target label is assigned to a different person)

False negative : is when missing estimation happens (i.e the i^{th} target label is non-assigned in the frame even if that target is present)

Accuracy : the number of correct detection/classification of the unlabeled patches divided by the number of unlabeled patches (i.e. when the ground truth and the output by the algorithm matches it is considered as a correct detection). It is the sum of true positives and true negatives divided by the sum of true positives, false positives, false negatives and true negatives.

Precision : is the number of true positives divided by the sum of the true positives and false positives.

Recall : is the number of true positives divided by the sum of the true positives and false negatives.

4.2 Offline Mode

Here, I considered the offline tracking of multiple people/object. In this set up we will process prerecorded video footage from surveillance which we know the beginning and the ending frames of the video. We will build the weight/similarity graph for the whole patches detected. It is not a real time process. This approach can't be applied to a situation where real time knowledge is mandatory. It's application will be on the areas of investigation of the things occurred previously like video forensics and other similar domains of applications.

4.2.1 Experiments for offline mode

In this experiment setup we performed an offline tracking approach, where following the assumption that we have prerecorded video footage and we tested this on different frames sizes of the CAVIAR video dataset. And we tried to show the effect of the number of frame size on the precision and recall of the result. As the number of frames increase the probability that person detector makes false estimations will also increase as a result the overall performance will be highly affected.

In figure 4.1 we can see some sample frames taken from the output results using Caviar datasets. Colored bounding boxes show the obtained tracking results. we measured the performance in terms of Precision and Recall, where we consider a True Positive as a patch classified correctly with its label, a False Positive a miss-classified patch (e.g the i^{th} target label is assigned to a different person) and False negative a missing estimation (e.g. the i^{th} target label is non-assigned in the frame even if that target is present)In order to abstract the overall performance we also evaluate the F-measure.

Figure. 4.1: Sample output of our framework on 'CAVIAR' video dataset

DataSet	Number of frames	Precision	Recall	F-measure
Caviar	49	1.0	1.0	1.0
	100	0.91	0.89	0.9

Table 4.1: Experimental result in Offline mode using CAVIAR dateset

4.3 Online Mode

The second mode is on-line detection and tracking of multiple targets. In this set up, unlike the offline mode we are not dealing with pre-recorded video footage from surveillance in which we know the beginning and the ending frame of the video, so

instead we will process some fixed number of frames from the stream input iteratively. And we will build a temporary and smaller dimension weight/similarity graph for the patches detected when compared to the offline mode.

It is approximately near real time and its application will be quite meaningful in a scenarios where real time knowledge is necessary. It can be applied on the areas where surveillance security is desired to have a knowledge of which persons are sitting , moving or doing something in a particular place, it might also be the case, certain places are in need of high security since they might be sensitive place like the bank and of similar domains.

4.3.1 Experiments for Online mode

In this experiment setup we performed an online tracking approach, with the assumption that the video frames are live streaming in a sense, we do not have pre-recorded video set up in which we know in advance the beginning and the end of the video. So in order to handle such scenarios we decided to iteratively run our frame work within certain frame size (number of unlabeled frames to be process at one iteration). Hence, our system works as follows: First we decide the window size then after we accumulate frames until we reached the maximum window side that we decided to use, then after collecting all the frames we will run our algorithm on those frames and label them accordingly and so on.

Here, varying window size has a significant effect on the precision and recall of our framework, the bigger widow size we select the more we are exposed to the errors caused by the person detector so we need to deal with them. On the other hand if we consider very small window size the it will be computationally costly since we iteratively run the algorithm on the corresponding frames. So, finding the right window size is crucial for

the overall performance of our framework.

I have done the experiment by considering different window sizes so that we can see the quantitative results and make comparison between them.

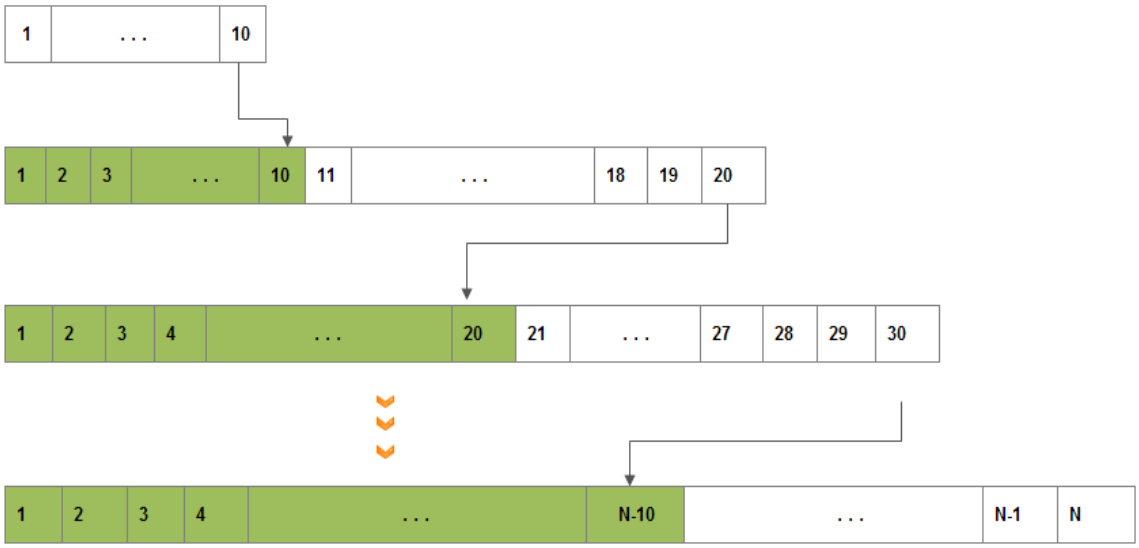


Figure. 4.2: When we consider 10 window size, the colored part indicates labeled frames at each iteration

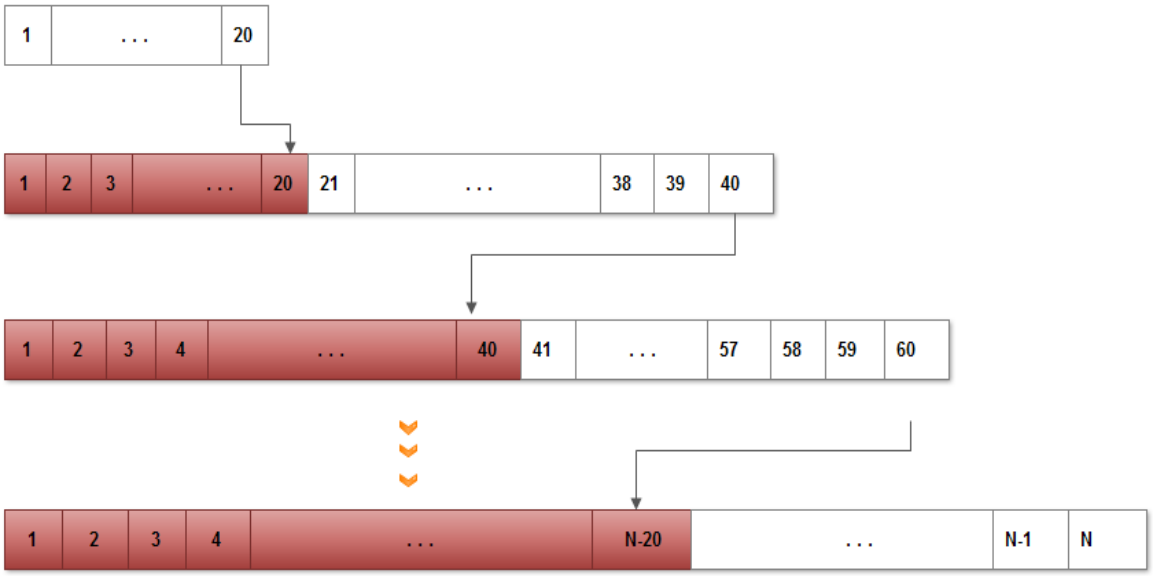


Figure. 4.3: When we consider 20 window size, the colored part indicates labeled frames at each iteration

Window Size	Precision	Recall	F-measure
20	0.99	0.983	0.986
40	0.955	0.951	0.95
80	0.899	0.8667	0.88

Table 4.2: Experimental result in Online mode with different window size on 140 frames of CAVIAR dataset

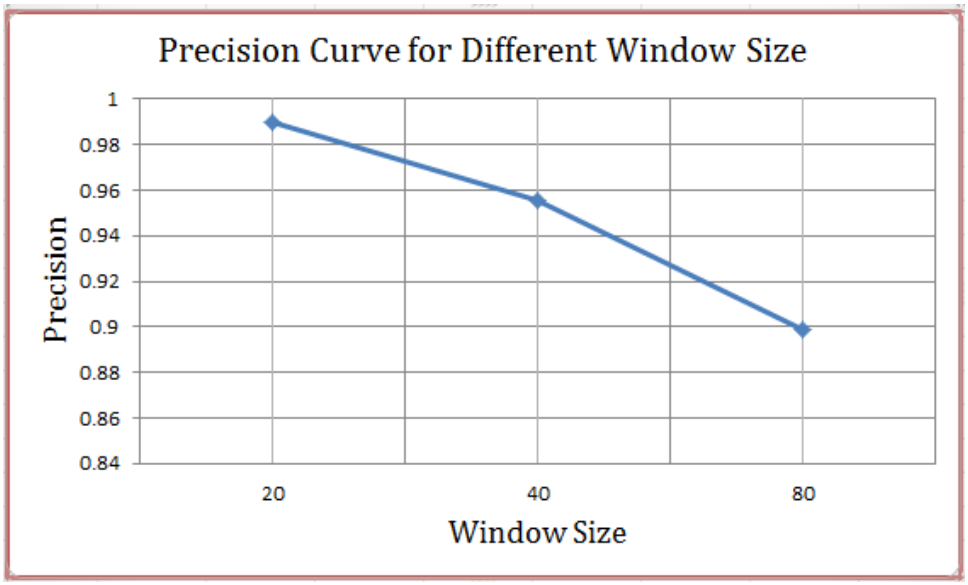


Figure. 4.4: Precision Curve for Different Window Size

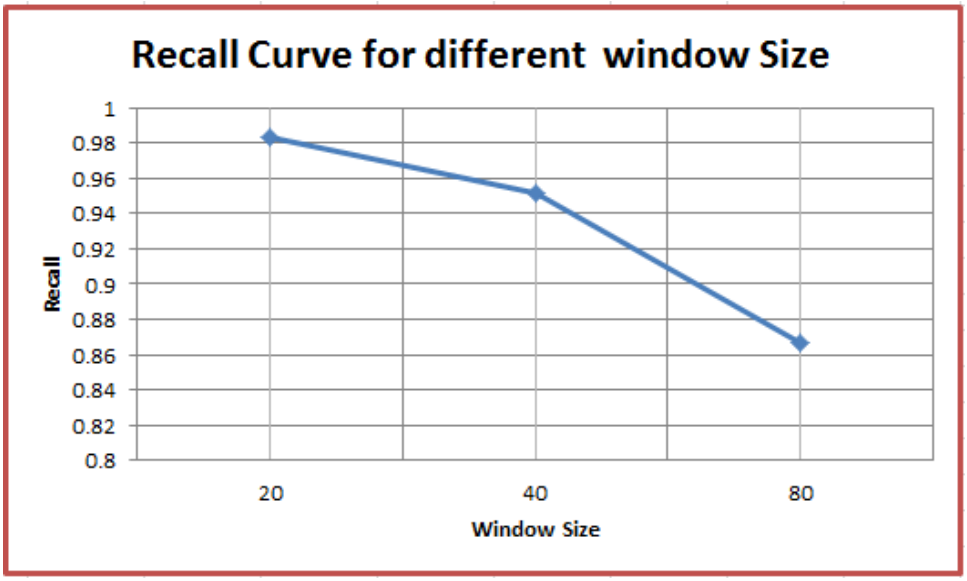


Figure. 4.5: Recall Curve for Different Window Size

4.4 Experiments done on different scenarios

The following video datasets (3DPes and CAVIAR) are specifically selected to test the robustness of our algorithm. Some hard situations are involved with this videos like Occlusions, Small sized detection of patches where the people are less distinguishable, people with changing appearances, wearing similar cloths and so on. And from the result we can see that our algorithm can handle well this kind of scenarios. Qualitative results are presented below.

Here are some sample frames taken from both video datasets (CAVIAR and 3DPes):

Figure. 4.6: Sample frames from CAVIAR Dataset

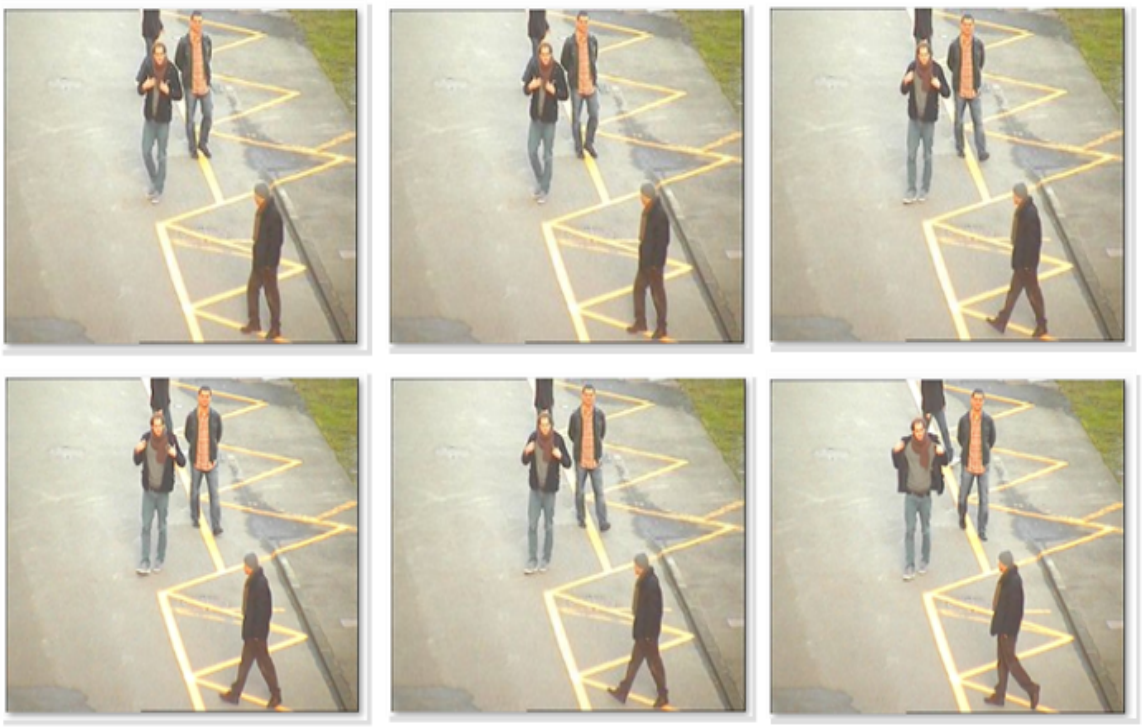


Figure. 4.7: Sample frames from 3DPes Dataset

4.4.1 Experiments On Perfect Cases

In this experiment set up we wanted to show that our algorithm will give a perfect and best output tracking result if it receives correct detection of people/patches as an input, as the table shows the precision, recall, accuracy are 100 percent. This happens from the fact that we considered the first 49 frames of the 'CAVIAR' video data set where the HOG-based people detector detected a good and full detection that is the persons without partial detection and error.

We have witnessed that in some cases, People detector detects plain backgrounds as a person or it detects half parts from two different target, this are some of the reasons that lead to an error in classification. The occurrences of such cases will make the framework fragile since their is no mechanism that the system can identify that given patch is a person or not as a result it will be confused between real patch/target and

fake patches/target detected as a person but which are not persons in reality. So such cases will affect highly the over all performance of the framework.

In figure 4.8 we can see few sample results using CAVIAR dataset.

Figure. 4.8: sample of perfect results on Caviar dataset

DataSet	Number of frames	Precision	Recall	F-measure
CAVIAR	49	1.0	1.0	1.0

Table 4.3: Performance of our framework on selected frames from CAVIAR dataset, where the HOG-based people detector generates clean detections of patches

4.4.2 Experiment On Tracking Target With Changing Appearance

Tracking target with changing appearance is one of the most trickiest scenario in people tracking but yet the most important exception that should be handled well. In a practical world people might change their appearances for many reasons among them

one might want to escape from security after involving in some criminal activities by changing his/her closes. In such cases it will become more difficult and challenging to track a target.

Hence, to test the performance of our frame work we considered a dataset(3Dpes), in which, in this dataset we can find one person trying to change his appearance by taking his jacket off.

This experiment was carried on to show that our algorithm can handle targets with changing appearance (changing dresses) and poses. As we can see from the output in figure 4.9 , we got a very promising result.

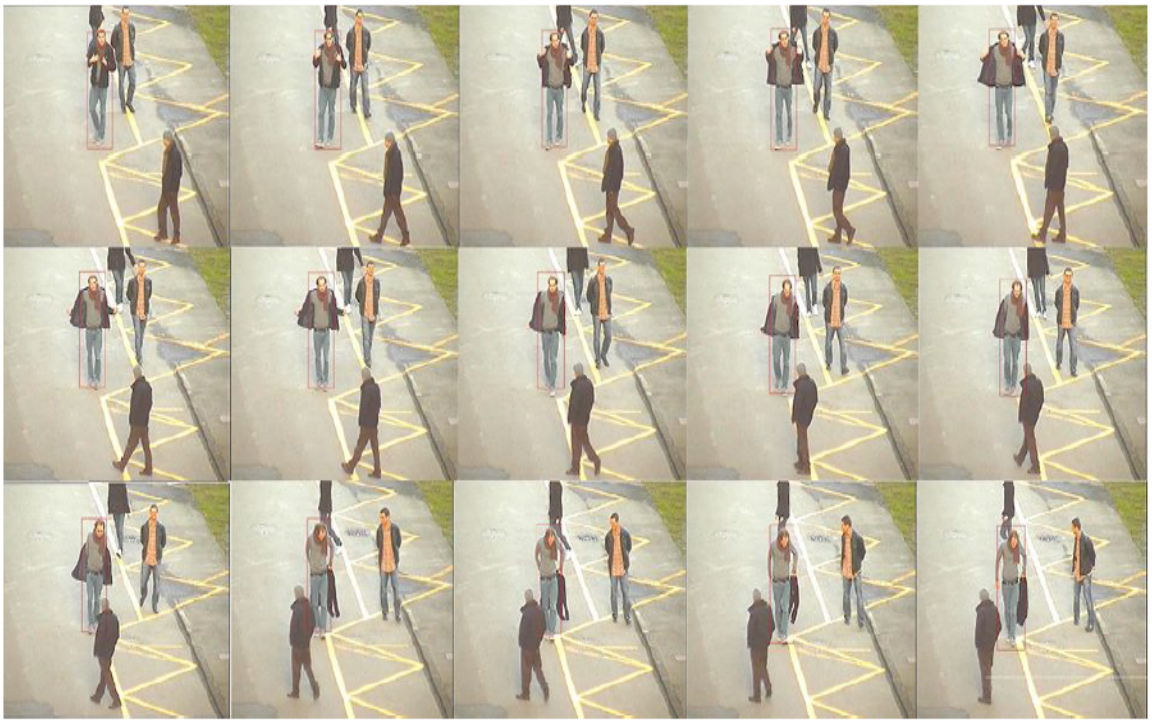


Figure. 4.9: Sample Frames from Tracking result of experiment done on a person with changing close

4.4.3 Experiment On Re-identifying Target

Re-identification of a person is the most common problem on the area of people tracking. Re-identification means as we can perceive from its name it is a process involved in identifying a target again, after the disappearance of the target for some period of time. This absence of a target for temporary time usually could happens for different reasons one is because of people occluding each other in other terms if the target is blocked by another person/target for some time from the line of site of the camera. The second reason is when a target/person left or went away and be out-of-sight of the camera for a certain time and comes back again. And also sometimes it might be the case that the person detector in this case HOG (Histogram of Oriented gradient) made some mistakes like: it identifies a person for certain period of time and then it fails to detect target for a couple of frames while he still exist in the video after a while it re-identifies target again. These are some of the exceptions that our framework should handle.

we have done the following experiment on CAVIAR dataset in which we can find the occlusion between two targets (i.e. two target cross by each other and one of the target is overshadowed by the other one and staid hidden for some time and then reappear) so it will be perfect video to test the robustness of our framework.

As it can be seen from the figures 4.10, our system can handle very well this type of problems. We can notice that one of the target initially tracked by the blue box is overshadowed by the other person, which is tracked by yellow bounding box for a while then after when they separate and both of them become in a clear sight for the camera, our framework re-identifies our target/person, that we were tracking with blue bounding box in which it were totally absent from our camera's line of site due to occlusion occurred between two targets.

Figure. 4.10: Tracking result Re-identifying on videos with occlusion between two targets/patches

4.4.4 Experiment On Identifying Newly Appearing Target

Identifying a newly appearing target is considered to be one of the most difficult scenario that we will face in people tracking problems, since in a given video numerous amount of people will appear in one part of the video and disappear from the line of site of the cameras similarly new person will appear in the middle of the video which is never seen before.

In one of the most recent works done on people tracking by Tewodros M. and M. Pelillo [MP13] in this paper they used game theoretic graph transduction framework to solve the people tracing problem but their framework suffers mainly with the above mentioned problem, that is, identifying a newly appearing person in a video, this happens from the simple fact that their system only tries to assign patches/people from the video to the closest similar initial labels. so, whenever their system encounters such cases it will

recognize him as one of the existing initial labels.

When we come to our framework since our system takes no initial labels from the user it will simply recognize and identify him as a new person rather than treating him as one of the existing persons from the previous videos.

Here, we can see how robust our system is whenever the system encounters such scenarios. For that purpose we have tested our framework on selected frames from CAVIAR dataset in which such cases are happening more frequently so that we can clearly simulate the desired conditions.

Figure. 4.11: Tracking result on Identifying new appearance of a target

As we can witness from the figure 4.11, at first, the target bounded in a yellow box were not detected by the person detector that means for our system that person is absent from the frames(false negative) but by the time it appear our system manages to detect it and identify him as a new person, rather than incorporating it to one of the existing targets.

4.5 Comparison Of Our Method With Other State-of-the-art Approaches

In order to measure the quality of tracking results, we use evaluation indexes like: Precision, recall and accuracy.

The table contains the average indexes' values obtained on 140 frames of CAVIAR datasets using our framework and different state-of-the-art approaches. Except our results, the rest that are mentioned in the table are adapted from their original papers.

	Precision %	Recall %	F-measure %
Graph Transduction Game	92.21	98.43	95.21
Transductive Learning Tracker	94.0	95.0	94.49
Our method	99.0	98.3	98.64

Table 4.4: Comparison between our framework and others (graph transduction game and transductive learning tracker) on 140 frames of CAVIAR dataset.

Referring to Table 4.4 , the first and the third row offers the contrast with the graph theoretic game based tracking in [MP13] and our framework. Our tracker exhibits higher values in precision and f-measure and have nearly similar result in recall. In a similar fashion, the second and the third row exhibit the comparison between our framework (dominant set based tracking) and transduction based tracking [CCC11b], here also our tracker prevails in all indexes precision, recall and f-measure.

4.6 Limitations Of Current Framework And Future Works

Our method performs well on most of the datasets that we have selected to perform the experiment on. We also noted that our system is highly dependent on the quality of the patches detected by the person detector, in a sense, as we have witnessed the person detector makes occasional false negative (i.e. when detector fails to extract a person even-thought the person exist in a frame) and false positive (i.e. when the detector detects an object which is not a person or when a detector detects only part of a person). As a consequence, the more false negative patches detected, the worst the recall became. In general terms, as more unambiguous patches are detected, the more miss-classification errors will occur and influence the over all result. In addition to that, our method is sensitive in cases like when a person appearance changes completely trough out the video (if he changes closes or when he appear in different poses).

In this work we haven't considered any noise analysis techniques in order to smoothen the error caused by the person detector. We just took all the out puts of the person detector and try to represent the people (patches) in covariance matrix and then compute the distance between them in order to formulate similarity matrix.

Noise analysis has been studied extensively in signal and systems and helps us in improving results of many frameworks. *Moving average* is one of the methods used to analyze noise. So as a future work to tackle the above problem, we can apply this technique to our framework as follows: First we fix a frame size to a certain number and by taking the previous frames as a back history we compute the average of the covariance matrices of the respective frames and apply this technique iteratively for all frames.

In doing so, we not only smoothen the error caused by the person detector but also it will help us reduce the confusion caused by appearance changes that is if one of the detected persons in a video constantly changes his/her appearances by changing closes or by the way they stand and so on.

Chapter 5

Conclusion

We presented a system that perform in both online and off-line multi-target tracking by utilizing Dominant set approach in the context of video surveillance. The off-line mode might have a potential application on the area of video forensics, where post-incident investigation is held on prerecorded video footage. On the contrary, the on-line approach might have a potential application on the areas where real time knowledge is important for instance in cases like security by surveillance, airport security, museum protection and so on. This is done by processing incoming video streams instead of waiting for prerecorded footage. we showed qualitative and quantitative output results of our algorithm by making tests on varies video data sets and got promising good result. The novelty of this work is on the point that it uses Dominant set approach for multi-target tracking.

we saw that our system is dependent on the quality of the input from the person detector, the more unambiguous patches are detected (false positive and false negative), the more miss-classification errors occur and influence the overall result.

Bibliography

[BRL+09] Michael D. Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc J. Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *ICCV*, pages 1515–1522, 2009. 4

[CCC11a] Dalia Coppi, Simone Calderara, and Rita Cucchiara. Appearance tracking by transduction in surveillance scenarios. In *AVSS*, pages 142–147, 2011. 5

[CCC11b] Dalia Coppi, Simone Calderara, and Rita Cucchiara. People appearance tracing in video by spectral graph transduction. In *ICCV Workshops*, pages 920–927, 2011. 1, 5, 26, 29, 30, 47

[DT05] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 886–893, 2005. 24

[FM99] Wolfgang Förstner and Boudewijn Moonen. A metric for covariance matrices. *Technical report, Dept.Geodesy Geoinform., Stuttgart Univ., Stuttgart, Germany,*, 1999. 29

[JTEARAET06] Wynter L. Jimenez T. El-Azouzi R. Altman E. and Boulogne T. *survey on networking games in telecommunications.Computers and Operations Research.* 2006.

[LLS10] Yunpeng Liu, Guangwei Li, and Zelin Shi. Covariance tracking via geometric particle filtering. *EURASIP J. Adv.Signal Process*, pages 1–22, 2010. 26

[MP13] Tewodros Mulugeta and Marcello Pelillo. Multiple target tracking as a graph transduction game. *masters thesis at ca foscari university*, 2013. 2, 6, 26, 45, 47

[MWS10a] Michael J. Metternich, Marcel Worring, and Arnold W. M. Smeulders. Color based tracing in real-life surveillance data. *T. Data Hiding and Multimedia Security*, 5:18–33, 2010. 5

[MWS10b] Michael J. Metternich, Marcel Worring, and Arnold W. M. Smeulders. Color based tracing in real-life surveillance data. *T. Data Hiding and Multimedia Security*, 5:18–33, 2010. 26

[NR99] Noam Nisan and Amir Ronen. Algorithmic mechanism design (extended abstract). In *STOC*, pages 129–140, 1999.

[PK97] Avi Pfeffer and Daphne Koller. *Representation and solutions for game theoretic problems. Artificial Intelligence.* 1997.

[PP03] Massimiliano Pavan and Marcello Pelillo. Graph-theoretic approach to clustering and segmentation. In *CVPR (1)*, pages 145–152, 2003. 8, 9, 11, 16

[PP07] Massimiliano Pavan and Marcello Pelillo. Dominant sets and pairwise clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):167–172, 2007. 2, 22, 30

[PP09] Fatih Porikli and Pan Pan. Regressed importance sampling on manifolds for efficient object tracking. In *AVSS*, pages 406–411, 2009. 4

[PTM06] Fatih Porikli, Oncel Tuzel, and Peter Meer. Covariance tracking using model update based on lie algebra. In *CVPR (1)*, pages 728–735, 2006. viii, 26, 28

[SB98] S. Sarkar and K. L. Boyer. *Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors.*, volume 71. 1998. 18

[SM65] E.G. Straus and T.S. Motzkin. *Maxima for Graphs and a New Proof of a Theorem of Turan.* 1965. 16

[YJS06] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), 2006. 1, 4