Ca' Foscari University

Department of Computer Science

MASTER THESIS

Multiple Target Tracking As a Graph Transduction Game

Venice, October, 2013

By:

Tewodros Mulugeta Dagnew

838218

Supervisor:

Prof. Marcello Pelillo

ACKNOWLEDGMENTS

ABSTRACT

This thesis narrates about a semi-supervised learning approach applied on on-line/off-line multiple people/object tracking in a video surveillance scenarios as graph transduction based on the notion of game theoretic approach where very few (may be only one) frames be labeled (indicating the targets) and the approach tracks and even re-identifies the targets.

The off-line feature of the application will be helpful in forensic applications for investigating already recorded video footage where the on-line feature can be applied where real time knowledge is necessary like for the purpose of security or identifying people by detecting and tracking multiple people simultaneously .

Graph transduction is a semi supervised learning technique that tries to do classification over a graph of labeled and unlabeled data points (i.e. the labeled nodes with zero entropy, and the unlabeled ones with maximum entropy); here the data points are the detected persons in each frame. As we know, Videos are composed of frames and in each frame there are peoples. And using people detectors (like HOG -histogram oriented gradient), we can detect people. Then each picture of detected patches will be treated as a graph node.

And there will be a similarity based comparison between the nodes (the node or the game player is represented by a model) so that similar patterns would be recognized. In the beginning targets to be tracked will be labeled, and then the provided labels propagate to the unlabeled ones consistently which means the appearance of the targets will be tracked/identified in each frame of the video.

The frame work is based on game theoretic notion. The transduction or information propagation is formulated in terms of a non-cooperative multi player game, where equilibrium is in a sense of consistent labeling of the data or assigning targets id (which includes non-target)to each detected patches of the frames, which the video is composed of. And multiple targets can be tracked simultaneously.

It can be seen as a learning approach that considers the tracking problem as a semi-supervised learning problem, where given few target samples,we look forward for searching target occurrences in the video stream. The people's appearances are modeled by using covariance matrices on color and gradient information which lie on Riemannian manifolds.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTIONS

The accessibility of videos supplied by surveillance cameras focusing on human activity, in urban outdoor and indoor areas, has recently got enhanced. As a consequence, beyond simply upgrading the accessibility of these videos and just having collection of video information, the attention is growing on automatically understanding the knowledge contained in the videos and providing semantics and classification of their contents.

People and their reciprocal interactions are the most important element in surveillance videos, therefore the most immediate and demanding task is tracking people in the sequence of frames of videos. The intricacy of this task deals with some difficulties like the articulated human shape, variable movement patterns and littered surroundings, appearance change etc. yet which always remain as a natural phenomenon in this field of study. Betterments have been made by recent scientific works.

In this thesis we present our work where we are able to manage the problem of multiple target tracking utilizing graph transduction based on game theoretic approach in off-line and on-line manner. When we say 'off-line' it is intended to refer to a scenario involved in processing of already recorded footages. On the other hand when we say 'on-line' it is intended to refer to a scenario involved in processing of incoming video streams which involves nearly a real time operation.

And we address the problem of following people in different surroundings even when time/special coherence is not fully guaranteed i.e. when the target is not observable in every frame, when the movement of the target is not predictable or when its aspect changes unpredictably. One of the good things of our approach is that it avoids com-

plexity related with motion prediction and motion computation, utilizing the paradigm of people tracking by semi supervised learning instead of trying to predict tracking, because of the fact that in many complex and real life situations it is difficult to guess or predict motion. That is why we don't take motion related things into consideration, but as a future work we thought of exploiting relative position of targets in comparison with the current frame and the previous frame so that we could get probabilistic prediction which gives a little bit more information to enhance the classification task.

In our approach, following individuals for long periods and in various types of areas (e.g. under the coverage area of different types of cameras) couldn't be thought of just only as following or tracking problem, however generally a problem of pattern recognition i.e. Pattern (of people) identification by means of visual or appearance look.

Thus during the work we will relate this issue of people following or people identification with the context of pattern recognition sense, we will also use the term tracking with this broader meaning. The primary assumption is having initial detected patches of people from video frames as input may be manually like in many forensic application domains where a video investigation is taking place or extracted automatically using some state of art object /people detectors then following similar patterns in the next sequence of frames, in other words this leads to tracking or following the target in the whole video, even though the target's appearances is in constant change, we will discuss how we handled this problem known as template update problem, in detail in the next chapters. For example When someone enters a controlled area and walks across a gate, the standard tracking systems are very effectual and can be exploited to create the first step of short-term memory. We propose a game theoretic framework based on semi-supervised learning that utilizes sequences of some marked visual patterns neglecting any other temporal or spatial information aiming at learning how to follow the target by labeled and unlabeled frames of a video. Among semi-supervised methods we adopted a graph-based transductive learning. Transduction is different from Induction, it doesn't

intend to find a discriminative function for arranging all new instances of the people; it just defines the procedure to classify the supplied unlabeled data(a detected person from a frame which we have no label of the person ,meaning at a time 't' we don't know who is person and our intention goes to know who is the person at a time 't+1' by considering similarities of this unknown person with the known targets from the previous frames) as one of the targets or not. Here what makes our system powerful is that it is not for only single target tracking instead we became able to handle or track and follow multiple targets at the same time. A fresh side of this work is that we approached the problem of people tracking in video as a graph-based transductive learning problem mapping it with the notion of game theory.

In order to utilize the discerning power of the Semi supervised transductive learner we used the information from the labeled data by modeling it and then comparing it with unlabeled ones with the similarities of the information among the models . Initially given very few labeled datas and after computing models of the multiple targets which we are in favor of identifying them, we tried to discern the coming unlabeled data comparing relative similarities, this is very important because sometimes people tend to have similar appearances because of their dresses or poses e.t.c and the system should be able to learn and recognize the similarities and dissimilarities so as to identify and reach more or less a correct judgment as we human beigs do in every day life subconsciously.

We adopted a different approach for recognizing patterns of people in a sense that different from the visual feature generally used for people recognition. In our case, a covariance matrix descriptor was chosen, integrating information about textures, colors and shape, in just one formulation. This combined discriminative and compact information of covariance matrix descriptor which acts as a model for the detected people in the frame sequences of the videos lies in Riemannian space, here we have to keep in mind that, operations which are valid in Euclidean space might not work in this space.

It is fact that the appearances of targets can change through time, so one question is

that how we dealt with this change of appearances of target people, the answer would be utilizing the power of graph based transduction. Even though some erroneous models appear which are in the form of nodes of a graph ,one node(let's say an unlabeled node to be given label assignment) have a weighted similarity to every nodes, out of which the initially manually labeled are found and also correct results(label assignment) produced by our algorithm to the other unlabeled nodes, so what happens here is most of the edges from this unlabeled node to others would be a correct weighted edge so, the classification wouldn't be affected that much because of an injection of a wrong target models, because most of the edge weights reflect the correct similarity. In other words when we talk still with the mid set up of graph approach, the weighted edge between the person in the current frame and the same person in the previous frames will tend to be similar even though their appearance has changes because the similarity is still maintained in the successive frames, it is like associating a similar model to the same cluster. For example the appearance of targets in frame one will be similar to the next frame even though there might be a slight difference , our system is aware of this thing, it is like a flow the system will keep on learning similar models the appearance variability though the successive frames. As a result a person in frame 1 which faces to the front side of the camera will be recognized as the same person in frame 100 even though he/she changes their poses, because as it is mentioned the system learns the constant changes thought out the frames in the middle, and also the models used to describe a patches of a detected people not only have a color information instead it is a combined information of color, shape, edges, and textures thus by preventing and dealing with the common problem of drifting.

Our system has a purpose of tracking multiple targets a the same time which can be used in the surveillance of people in complex, cluttered and crowded environments. Forensics or multimedia retrieval can benefit from it, like for example when searching for targets in previously recorded video footage where the users of our system can

manipulate time in backward and forward manner. Our method has been compared with some method which have the same goal as ours and tests carried out on different video data sets which comprises real life situations, have shown that our method is efficient and powerful too, even outperforms some others methods presently in use. As we mentioned, an initial data association of a few labeled samples is needed in most systems aiming to track targets. One powerful feature of our system is that it gives good classification result even though when it is fed with only one labeled frame as an input which is a rare case in other methods. It propagates the information contained in this only one manually labeled frame to the many unlabeled frames existing with good accuracy to give a result of good classification/tracking of targets/ to give consistent labeling of the unlabeled data.

In our approach we tried to tackle the multitarget tracking problem in a way that it can flexible for the potential application domain (i.e for forensics video investigation or security by means of survaillance) by separating online and offline operation. when we say offline, it is in a sense that there will be already recored video, on the other hand when we say online it is in a sense that there will be a video stream coming as an input to our system, so that on this context we don't have a lready recorded video footage rather a video stream so the learning process will be iterative. even in this online tracking context, we have online tracking with out any memory, short term memory and long term memory. the top view of the system is as presented in the figure below.

OK.It is natural to ask what is the benefits of this kind of applications to in our world, where they can be applied , what purpose to serve, and many many other questions.....

The advantages of people tracking technology are particularly compelling in areas like the following:

. Retail Management :- By analyzing traffic scenarios , counting patrons, stores and restaurants can ameliorate customer service, respond more effectively to rushes, and define which displays and products are most efficacious in producing sales. Competition

among retailers is ferocious. Traffic analysis provides retailers a competitive benefit that can make a crucial difference in margins and market share, also in employee monitoring

. Interactive Entertainment :- Museums are testing different ways to create interactive spaces, where images, sound, and lights respond to the positions and actions of passers-by. For instance, in a bookstore, words may swarm around people feet, while in a stairwell, colored lights may follow people as they go upstairs. The computer systems sustaining these displays can also furnish traffic pattern data and other helpful marketing data for the venue.

. Security :- People tracking presents some important advantages over traditional video surveillance. People tracking systems can highlight the position of specific individuals in a wide space, such as an airport or a parking. They can also warn security staff to suspect activity, such as a person wandering around a locked door or visiting a location repeatedly. In a broader sense , object tracking technology can be used to find abandoned objects, such as a suitcase left in a train station.

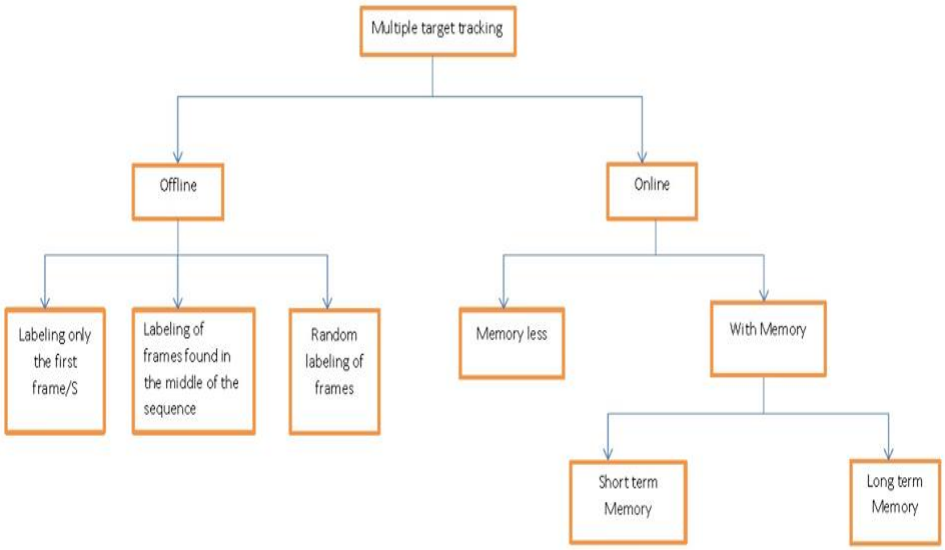


Fig. 1.1: top view of our approach

2. BACKGROUND CONCEPT

2.1 Semi Supervised Learning

In the field of Machine learning, semi-supervised learning now a days is getting popular and popular because it is in the mid-way of supervised and unsupervised learning, as a result it has a combined power of both Supervised Learning(i.e all the available training observations have label/class information) and Unsupervised Learning (i.e all the available data are with out labels), so it is getting attention in the research communities because of the matter of fact that it got the robustness to tackle problems that can't be approached by only Supervised Learning or Unsupervised Learning algorithms. In real world, unlabeled data are found in plenty rather than labeled data, such as images, text , bioinformatics e.t.c. so if we are able to study and be able to label only few of these plenty of raw data, we will be able to get a good classification result of the unlabeled ones using state of art semi-supervised learning algorithms. one may ask how this can be achieved. And the answer will be the following; Usually Semi-supervised algorithms use a graph based approach. data will be mapped as nodes, where some of them are labeled and some of them with out labels. first of all a data have features(a vector of D dimensions, where $D \in N$, where N is the set of Natural numbers) that characterize themselves. After getting the features of each data, it is needed to get the edge weight between nodes by calculating how far apart the data lie(i.e computing the distance between them). The distance might be euclidean distance, cosine distance e.t.c. it depends on the behavior of the features of the data and the application domain be-

ing treated. Then by using techniques like relaxation labeling, graph transduction e.t.c ,which will be explained in the next section, it will be able to propagate information from the labeled ones to the unlabeled ones, and as a result we will be able to know the labels/class of the previously unlabeled data finally. And this sounds good.

2.2 Introduction to Game Theory

The introduction of Game theory backed up with mathematical formulations was achieved by Von Neumann and Oskar Morgenstern in the year around 1940, but their formulation had a restriction and it works only for zero-sum games where the loses and gains are equal.

Then John F. Nash came and modified the formulation and showed and explained difference between cooperative and non cooperative games which helps to tackle problems in real world scenarios.

When the sum of the payoffs are not equal to zero or when the sum is no longer constant,maximizing and minimizing of the player's payoffs are no longer equal.

When the total summation of payoffs are no longer equal to zero or a fixed constant value,maximizing and minimizing of the player's payoffs will also be no longer the same.

A general sum game of two player could be represented as a matrix of entities where each entity have a pair of values that represents the payoff of player 'A' and player 'B' respectively, or it can be represented as two matrices, one for each player's payoff. For example if you see the matrix below, we can assume that player 'A' is the row player with 2 choices at his/her hand while player 'B' is the column player and has 3 choices. the payoffs can be represented as two sets, let's name them I and J so that the cross product ,IXJ, represents the whole set(the whole elements of the matrix). If player 'A' decides to select $i \in I$ and player 'B' decide to select $j \in J$ then player 'A' will get the payoff $u_1(x, y)$ and player 'B' will get the payoff $u_2(x, y)$. Where $u_1(x, y) and u_2(x, y)$ are

the utility/payoff functions. 3×3 matrix

$$\begin{pmatrix} (1,2) & (1,-2) & (3,4) \\ (-1,-2) & (-2,1) & (0,-1) \end{pmatrix}$$

Here if player 'A' decides a strategy which is i=1 and player 'B' chose j=2, player 'A' will get the payoff of $u_1(1,2) = 1$ and player 'B' get the payoff $u_2(1,2) = -2$.

An important thing here is that two kinds of strategies exist: **Pure and Mixed Strategies**. When we see pure strategies a game is described by *the set of players 'A', the set of strategies S and the payoff functions U*.let's see a well known game known as,**Prisoners Dilemma.** The payoff matrices is given as follows

Strategies	$telllie$	$telltruth$
tell lie	$(-5,-5)$	$(0,-9)$
tell truth	$(-9,0)$	$(-1,-1)$

Also can be represented as two matrices given below:

$$PlayerI'spayoff = \begin{pmatrix} -5 & 0 \\ -9 & -1 \end{pmatrix}, PlayerII'spayoff = \begin{pmatrix} -5 & -9 \\ 0 & -1 \end{pmatrix}$$

As we can see the information from the above matrix,if both the players will be advantageous if they lie taking the risk of telling the truth into consideration.This is because both the first row and the first column dominates the second row and column respectively.So a player choose their dominant strategy irrespective of the strategy chosen by the other player.That means a player choose his best response.Choosing the dominant strategy both players will get a penalty of 5 where as if both play the dominated strategies they will get a penalty of 1. Since the players will be better off when they both

choose the dominant strategy, the game is called prisoners dilemma.It is called Prisoners dilemma because of the following scenario, two criminals one with big and one with small crime have captured and the police take them in to two different rooms. The police don't know which one is the bigger criminal, so he tells them If one confesses and the other deny then the one who deny will be sent to jail for 9 years while the other will be free.If both confess,they will be in jail for 5 year otherwise If both deny both will be in jail for 1 year.

To solve the above problem,let's mark on those payoff responses when the players select their strategies and those which are circled pairs are the solution of the game. let's solve the above game.

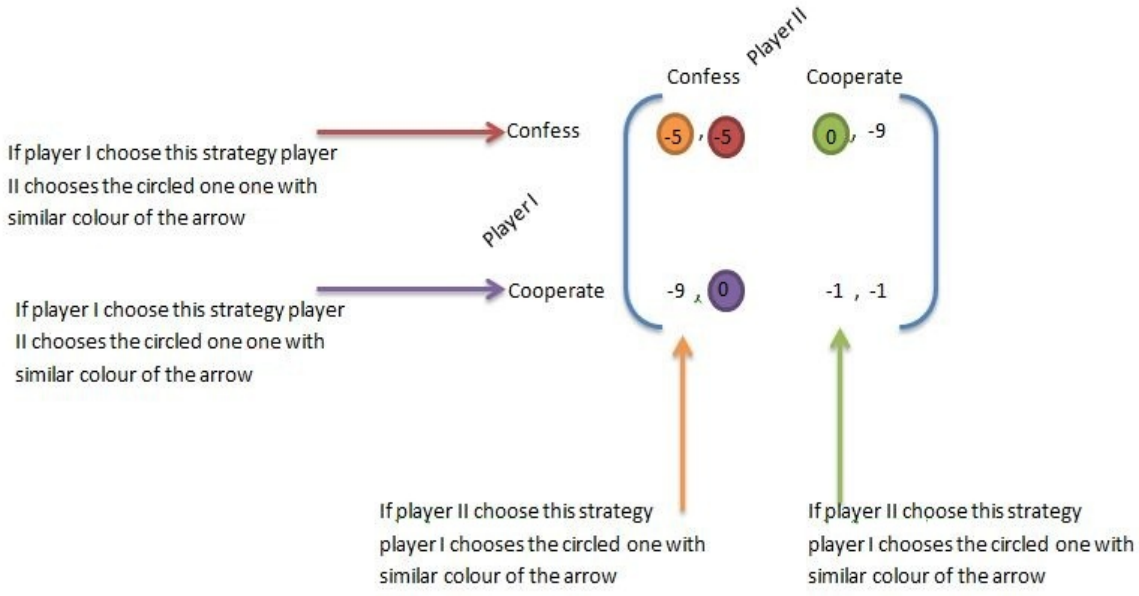


Fig. 2.1: Prisoners Dilemma

The pairs which are encircled from the figures above is the equilibrium point (Nash equilibrium) that the two players reached. Can we get this equilibrium point for all games? In pure strategies we can't where as in mixed strategies we can have the equilibrium point for all games.Let's see what mixed strategies means and see an example.

Mixed Strategies: The idea of mixed strategy is simply taking the probability

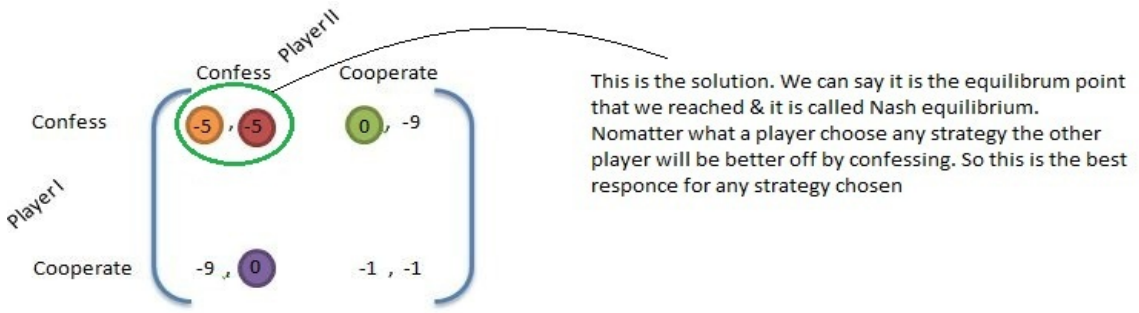


Fig. 2.2: The equilibrium point

distribution over the pure strategies of a players.If x_i is the mixed strategy for player i it is represented as a vector in R^m (m dimensional vector space). We know the probabilities are positive and that summed up to one, so it is in the standard simplex i.e $x_i \in \Delta_i = \{x_i \in R^m : x_{ih} >= 0, e^T x_i = 1\}$ and $\forall h \in \{1....m\}$ since $x_{ih} > 0$ it is called the support of the mixed strategy which is $\sigma(x_i)$. For each players i in a set of players I,We have a mixed strategy x_i in the standard simplex Δ_i, a set that contains the whole set of this mixed strategies for all the players $x = \{x_1....x_{|I|}\}$ is called **mixed strategy profile**. The Cartesian product of the simplexes, $\Theta = \Delta_1 \times \Delta_2....\Delta_{|I|}$ gives us the **mixed strategy space**.If we have a unit vector $e_i^h = (0, 0, 0, 1, 0, ..., 0)$ in the m-space, it represents the h^{th}vertex of the simplex Δ_i which is the h^{th} pure strategy.If we have payoff matrices of A_i for all players $i \in I$, the payoff that player i get when he played against player j is $U(x_i, x_j) = x_i^T A_i x_j$.If x_i is a mixed strategy which gives a higher payoff ,than any other mixed strategies give, for player i against the strategy x_j then we say x_i is the best response for player i against the mixed strategies x_j.We can define it simply as follows.

Definition (Best Reply): *A strategy profile x_i^* is a best response to the strategy x_{-i} of the other players if*

$$U_i(x_i^*, x_{-i}) \geq U_i(x_i, x_{-i}) \ \forall x_i \in \Delta_i$$

It solves the following maximization problem $\max_{x_i} U_i(x_i, x_{-i})$

The best reply in the extreme case is, in pure strategy, unique. But usually the best replies are always infinite and also if the best reply includes two or more pure strategies, any mixture of these strategies must also be a best reply. Or we can say If a mixed strategy is a best response then each of the pure strategies involved in the mix must itself be a best response. In particular, each must yield the same expected payoff.

Nash Equilibrium

Definition: *A strategy profile* $(x_1, x_2, x_3, ..., x_{|I|})$ *is a Nash equilibrium if for all i* $x_i \in$ *best response of* x_{-i} *(for each player, his choice* x_i^* *is the best response to the other player's choice* x_i^**). That means it is the best response to it self. We can write it in this form* $x^T Ax \geq y^T Ax$ *for all mixed strategies y.*

$U_i(x_i, x_{-i}) \geq U_i(x_i^*, x_{-i}) \ \forall x_i^* \in \Delta_i$ *, and we get* **Strict Nash Equilibrium** *with strict inequality* $\forall x_i^* \neq x_i$

Let's see the well known example which is Rock, Scissor,Paper game:

$$|x| = \begin{cases} x & \text{if } x \geq 0; \\ -x & \text{if } x < 0. \end{cases}$$

Player I

	Game	Rock	Scissor	Paper
	`Rock`	(0,0)	(1,-1)	(-1,1)
player II	`Scissor`	(-1,1)	(0,0)	(1,-1)
	`Paper`	(1,-1)	(-1,1)	(0,0)

Let's try to solve the problem in pure strategy i.e the case in which only one action is played at a time

As we can see from the selected points, we have no any indices that are with pair

Player I

Player II	Rock	Scissor	Paper
Rock	(0 , 0)	(1 , -1)	(-1 , 1)
Scissor	(-1 , 1)	(0 , 0)	(1 , -1)
Paper	(1 , - 1)	(-1 , 1)	(0 , 0)

Player I

Player II	Rock	Scissor	Paper
Rock	(0 , 0)	(1, -1)	(-1 ,1)
Scissor	(-1 ,1)	(0 , 0)	(1, -1)
Paper	(1, - 1)	(-1 ,1)	(0 , 0)

circled (chosen) elements. So we can say we have no any Nash equilibrium in this pure strategy. But we know from Nash theorem every finite game has a mixed strategy Nash equilibrium. Let's try to solve the mixed strategy Nash equilibrium.

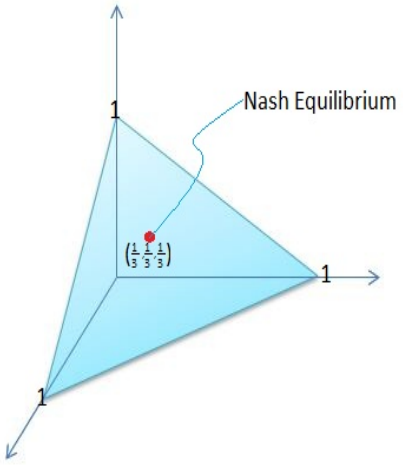
To solve the problem in this case, let's assume player I assigns the probabilities $p_r, p_s, and (1 - p_r - p_s)$ for the rock, scissor and paper respectively. The for player II,

Row, Rock payoff aganist $(p_r, p_s, and(1-p_r-p_s)) = p_r*0+p_s*1+(1-p_r-p_s)*-1$

Row, Scissor payoff aganist $(p_r, p_s, and(1-p_r-p_s)) = p_r*-1+p_s*0+(1-p_r-p_s)*1$

Row, Paper payoff aganist $(p_r, p_s, and(1-p_r-p_s)) = p_r*1+p_s*-1+(1-p_r-p_s)*0$

If we solve all these payoffs to be equal we will get $\frac{1}{3}$ for every probabilities assigned. So our Nash equilibrium will be $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$



Why study games*?*

Most of the real world situations in science and engineering can be modelled using games. Many peoples used the notion of game theory to solve real world problem.We have seen most of its application in the fields of economics, biology,networking in telecommunication[EARJTAEL06],in network routing, in resource allocation in distributed systems,resource allocation in information and service transaction in Internet[KP97].

Now a days, in the machine learning community peoples use the algorithmic aspect of game theory to design an algorithm for computational purpose. The notion of the algorithmic game theory was started by Nisan and Ronen [NR99]. In 2002, Ortiz and Kearns uses game theory for computational purpose in the study of machine learning. Here in our graph transduction case we used game theory for a consistent labeling problem, viewing the decision of the players (i.e nodes of the graph) for choosing class membership as their strategy and in our context this globally consistent labeling means the classification of all the detected peoples/patches from each frame of the video as

targets (multi) or not targets.

2.3 Graph Transduction

One of the immediate thoughts that comes up to machine learning community people's mind while speaking of semi supervised learning is graph transduction. it is a method that tries to propagate class membership information from labeled nodes to unlabeled nodes considering the similarity between the nodes on the environment where only labeled nodes and unlabeled nodes exist. usually the output of a graph transduction algorithm is the class assignment computed for the unlabeled nodes. when we see it from information theoretic point of view, the labeled nodes are the ones with zero entropy, i.e initially their class is known, as a result, the information they hold is with out any uncertainty, but on the other hand the unlabeled nodes are the ones with maximum entropy because initially there is much uncertainty on to which class they belong to. Classical graph transduction algorithms initially assume an unlabeled node's class might be any of the classes, that exist in the current frame work, with a uniform probability distribution. For example, if we are dealing with classification where there exist three classes , the initial prior probability for the unlabeled node to be any one of the class's will be 1/3.

In a little bit more formal way, assume there is a graph expressed as G= (V,E) where: V : the whole nodes (i.e the labeled and the unlabeled nodes together) E : pair-wise edges between nodes weighted by the similarity between the corresponding pairs of points and the data points grouped into

. Labeled data : $\{(X_1, Y_1), \ldots, (X_L, y_L)\}$. Unlabeled data : $\{X_{L+1}, \ldots, X_n\}$

Here in many cases the number of labeled nodes(L) is much lesser than the total nodes existing (n) i.e $L < n$ and if the edge between two nodes is high magnitude it means they have high degree of similarity and as a consequence they tend to belong to same class or we can say they belong to the same cluster too. this concept is similar to the homophily analogy in social network analysis.

And finally the goal will be to propagate the information available at the few labeled nodes to a greater number of the unlabeled nodes in a consistent fashion.

2.4 Nonlinear Relaxation labeling

The classical non-contextual pattern recognition algorithms related to classification problems are strongly affected by uncertainty and noise, that often lower the accuracy of the obtained results. However the relaxation labeling process, a class of parallel iterative procedure introduced by Rosenfeld, Hummel and Zucker [AR76] tries to give consistent solutions in the area of classification. relaxation labeling algorithms play a vital role in pattern recognition and computer vision areas, and their application lies in different problem domains. For a detailed concepts it is advised to refer [AR76], and also there some researchers who tried to prove the consistency of the relaxation labeling with different approach for example Pelillo M. in 1997 [Pel97] proved and showed how the non linear relaxation algorithms are related with Hummel's and Zucker's consistency theory. Now let's review the basic concept of nonlinear relaxation algorithm defined by Rosenfeld et al. [AR76]. later we will see how this setup can be associated and can be seen from a game theoretic point of view or approach.

The problem starts with a set of objects OB=$\{ob_1,....,ob_n\}$ and a set of 'm 'labels which the objects can get assigned L = {1,.....,m}. And our goal is to get each objects assigned to one of the available labels. In order to accomplish this, local measurement and contextual knowledge (i.e the interaction among the object's label's) should be

exploited. The contextual information is expressed as four dimensional matrix R={ $r_{ij}(\lambda,\mu)$ } which measures the compatibility between two hypotheses, and high magnitude indicates strong compatibility and vice versa. For example the notation 'r_{12} (L_1,L_3)' signifies the compatibility strength when object 1 is assigned to label 1 and object 2 assigned to label 3. At time zero the local measurement of an object is a probability distribution vector with m-dimension (i.e m is the number of available labels), so for the whole objects this information (i.e $p^{(0)}$) will be a matrix with size of $n \times m$ (i.e n is the number of objects and m is the number of labels existing). The local information for each object $ob_i \in OB$, is $p_i{}^{(0)} = (\ p_i{}^{(0)}\ (1), \ldots, p_i{}^{(0)}(m))^T$, where $p_i{}^{(0)}(\lambda) \geq 0$, $i \in OB$, $\lambda \in L$ and $\sum_\lambda p_i{}^{(0)}(\lambda) = 1$, $i \in OB$, $\lambda \in L$. Once this is clearly stated , the weighted labeling assignment for the whole objects that are found in OB ($p^{(0)}$) will be a concatenation of $p_1{}^{(0)}$, $p_2{}^{(0)}$, . . . , $p_n{}^{(0)}$ and it is the input for the relaxation labeling process. After taking the weighted labeling assignment($p^{(0)}$) as an input the relaxation labeling process iteratively update it by considering the compatibility model R. The space for the weighted labeling assignments is given by : Space= $\Big\{$ $P \in \mathbb{R}^{nm}$ | $p_i\ (\lambda) \geq 0$ and $\sum_{\lambda=1}^{m} p_i\ (\lambda) = 1$ where $i \in OB$, $\lambda \in L$ $\Big\}$. In this label space there two things : unambiguous assignments and strictly ambiguous. Geometrically thinking, the first ones are found in the vertices's of the space which signifies the assignment of a fixed labels to the objects while the latter are found in the interior of the space which signifies the probability of an assignment of a label to an object (i.e $0 < p_i\ (\lambda) < 1$). Now let's review an important component of the process called support function. It quantifies the agreement for a hypothesis that say an object $ob_i \in OB$ is labeled with some assignment $\lambda \in L$ in each iteration . And it is the function of the local measurement and the contextual knowledge / compatibility coefficient which is given by: $q_i^{(t)}(\lambda) = \sum_{j=1}^{n} \sum_{\mu=1}^{m} r_{ij}(\lambda,\mu) P_j^{(t)}(\mu)$.

By concatenating the $q_i(\lambda)$'s like we have seen for the $P_i(\lambda)$'s above, we will have a support vector with dimension of n x m, which is the same as P. $q_i(\lambda)$ will be high

when the labeling assignment λ for the ob$i$ is compatible with the other neighboring label assignments, and will be low when it is incompatible.$P_i(\lambda)$ is proportional to $q_i(\lambda)$. Finally the update formula of the process in a discrete manner will be like this :

$$P_i^{(t+1)}(\lambda) = \frac{{P_i}^{(t)}(\lambda).{q_i}^{(t)}(\lambda)}{\sum_{\mu=1}^{m} {P_i}^{(t)}(\mu).{q_i}^{(t)}(\mu)}$$

The denominator is just a normalizing factor so that the output result always lie on the Space defined above.

3. THE GRAPH TRANSDUCTION GAME FRAME WORK

In this section let's review the frame work for the Graph Transduction Game proposed by Erdem & pelillo [EP12] , that we used for our object/people tracking application in a video surveillance scenarios which is one of the domains of computer vision problems.

The goal of this framework is to achieve a multi-class node classification on a setup of a weighted graph G= {V,E,W}, where $V = (v_1, v_2, ...v_n)$ indicates the available number (n) of nodes, E as the available edges, between the nodes, with weights W . When the problem is faced with game theoretic formulation, the nodes act as non cooperative game players.So from now on the terms 'nodes of the graph' and 'players' will be used interchangeably and will have an equivalent meaning/ significance. One of the important assumption in the set up of the framework is, adjacent nodes often have same labels which can be inferred as the principle of *homophily* in the context of social network analysis analogy. And finally the labeling assignment for the whole players/nodes is expressed as a Nash Equilibrium.

The robustness of the frame work is that it can handle multi-class classification and can deal with different kind of similarities that may exist between nodes of the graph namely symmetric , asymmetric and negative similarity. And it is a simple framework which is not so much complicated.

Initially, before the algorithms starts (or before the beginning of the game) ,only some of the nodes/players are assigned labels, which have a full information of their class membership with full confidence (i.e the players with a pure strategy) and they are referred as the labeled players. On the other hand majority of the other players/

nodes who don't know their class membership referred as unlabeled players.

The frame work assists the players to choose a strategy from the available set of strategies that they can choose(i.e the existing classes). Si= { 1, . . , c } where c is the total number of classes; in other words it means a node will be assigned to one class or a player choose it's strategy (i.e a player chooses it's class membership). The mixed strategy for each player lies in a simplex with c dimensions which is defined above in 'game theory section'.

The group of the labeled players $P_L = \{P_{L|1}, ..., P_{L|c}\}$ differentiate themselves from the other unlabled players based on the pure strategies they always play. $P_{L|k}$ stands for the group of players playing their K^{th} pure strategy at any time 't' (i.e they know their class-membership at any point of time). For a player p playing it's K^{th} pure strategy equivalently means that it is playing it's extreme mixed strategy $\{{e_p}^k\}$ which lies in the vertex of the standard simplex defined above. This situation signifies that the labeled players don't care about achieving a higher pay off from their participation on the game because it is assumed that they already selected their strategies, and their role will be in guiding the unlabeled players $\{P_u\}$ to choose a strategy which is best for them so that they(the unlabeled players) will maximize their payoffs as a consequence the underlying real game is taking place between the unlabeled players.

According to Nash [**?**, **?**], with fixed number of players and fixed available strategies there might not exist an equilibrium if players only choose pure strategies but at the end , the game will be concluded with a Nash equilibrium in a mixed strategies where players mix their choices over the available decisions , so does this transduction game also. This Nash equilibrium of the game reached means each player now became in a stable state that they already chose their strategies that can get them with the highest payoffs, as a result of this, a consistent labeling is achieved globally. At this point the label / class-membership of a node/player i can be extracted from it's final mixed strategy which is in equilibrium state by picking the strategy with the highest probability. And finally the

the end of the game is reached, a complete classification result will be available which it's use will depend on the domain of the application that someone is working. Formally it can be put like the following :

$$ClassLabel\ _i = \underset{h=1...c}{argmax}\ P_{ih}$$

It is important to notice that the game taking place is the so called sub class of multi-player games known as polymatrix game. Polymatrix games are games that the whole interaction of the game is understood as the combination of pairwise interaction in which the edge between two nodes / players is seen as a single game, and the payoffs for this two players will be a function of the edge weight. Even though it is a multi player game, the polymatrix game rule states that at a specific moment one player can only interact with one of it's neighbor's only not with all it's neighbors at the same time. So the final payoff for a player P_i will be the summation of the payoffs that it got from the individual pairwise games that it played with it's neighbors. If we assume we have a pure strategy profile $s = (s_1, s_2, .., s_n) \in S$, the payoff for a player P_i is given by

$$\pi_i(s) = \sum_{j=1}^{n} A_{ij}(s_i, s_j)$$

And if we assume we have a mixed strategy profile $x = (x_1, x_2, .., x_n)$ the payoff is given by

$$u_i(x) = \sum_{p=1}^{n} {x_i}^T A_{ij} x_j$$

$A_{ij} \in R^{c \times c}$ is the partial payoff matrix between player i and player j and it comes from the weight of the edge between them such that $A_{ij} = I_c \times w_{ij}$, where I_c is an identity matrix with size $c \times c$ and c is the number of classes available which means the number of labels, in other words it also means the number of available decisions for the players.

Now let's see how the Nash Equilibrium is computed in the current framework. The

approach used to face this problem is motivated by using the principle of Evolutionary Stable Strategies [J.W95] which is famous and important in the context of game theory. The underlying concept is similar to the analogy of species evolution in biology , *survival of the fittest*. The game is seen as it is being played repeatedly, and as a result there will be a different generations of strategies and eventually the fittest strategy survives. To achieve this a discrete time version of multi population replicator dynamics used, but keep in mind that we can also use any other dynamics with less space and time complexity to boost performance efficiency. This will be mentioned in the section for 'future works'.

$$P_{ih}(t+1) = P_{ih}(t)\frac{u_i(e_i^h)}{u_i(P(t))}$$

The task of the equation is like a relaxation labeling process, at each iteration (or we can say generation) for each player, the probability for it's all available decisions will be updated individually. And the solution brings a Nash equilibrium . To see it's dynamical properties in detail it is advised to refer [Pel97] . The original formula for the multi-population replicator dynamic is

$$\dot{x}_{ih} = x_{ih}(u_i(e_i^h, x_{-i}) - u_i(x))$$

The cost complexity to compute Nash equilibrium of the described graph transduction game using the above discrete version of replicator dynamic is $O(IcP^2)$ where I is the number of iteration for the algorithm to finish , c indicates the number of classes (i.e the available pure strategies for the players to choose) and P corresponds to the number of the participant players (i.e the nodes of the graph)

4. MOTIVATION FOR OUR WORK

The motivation for our work vitally comes up from the idea of applying the above described frame work of graph transduction game, for a famous problem in computer vision known as multiple object/people tracking in a video surveillance. As it is mentioned in the section that describes about the Graph Transduction game frame work, it can powerfully deal with multi-class node classification based on game theoretic approach, so we thought about applying this frame work for multi target object tracking by the idea that representing detected persons as a node of a graph. while some other object tracking mechanisms proposed by some people can handle only single target object tracking.

5. RELATED WORKS ON OBJECT TRACKING

Object tracking specifically people tracking in this days is becoming an essential problem and many different kinds of approaches and methods have been implemented by many researchers. A brief review of literatures on this area is presented in [AYS06]. As we know these days a number of different kinds of video surveillance cameras appearing every where in the world for surveillance purpose, for security and for many other different reasons, as a consequence the availability of these videos in plenty to be meaning full and serve their purpose it is needed to do various types of analysis on the videos . Motivated from the idea that videos are composed of sequences of frames, almost all approaches on video analysis(i.e object tracking, video annotation, classification, e.t.c.) focuses on the analysis of the individual sequences of frames. It is logical to think that the most important part of surveillance videos are peoples and their actions, because of these reason people tracking on surveillance scenarios is getting so much attention in the community of computer vision. This issue won't come easily but comes up with its own challenges like varying motion of objects in the video, noisy environment, similar human structures, occlusions, e.t.c. Several works have been made and now it is possible to track single or multiple targets while keeping some occlusion constraints. Some research works in this field shows that tracking by learning is yielding good out put results in terms of accuracy than tracking by prediction when the target's motion is unpredictable or when the target is not existing in many of the video frames at the same time neglecting the complexity of motion prediction computation. One of the approach that is being used in many of the state of the art solutions, is Particle filtering tracking[BRL+09, DJ11,

LHT10, PP09]. The idea is that the tacking information is associated with probability distribution over the state of the target. Even though the method is robust, representing the features of the models under different kinds of lighting condition in a cluttered and noisy environment, results in wrong tracking. Another different approach is formulating the tracking problem as partitioning a graph. The goal of these methods is to find connected edge paths that connects nodes of the graphs which represent the same target to be tracked. Recently in video forensics this method is playing a vital role in tracking people by their appearance utilizing the weight/similarity matrix between the nodes of the graphs. but it's application is limited to off-line video processing [MW10]. Another attempt which have been tried is approaching the problem as a graph partitioning problem using semi-supervised techniques where the appearance similarities (weights) are used to facilitate the classification of the unidentified appearance by trying to match with the training data, the drawback of this approach is that ,the final tracking (classification) is strongly dependent on the training data's quality. if the training data for the target model is noisy and with many errors, the tracker's accuracy will be in question.

Now let's briefly review a recent work done on Transductive People Following in Unconstrained Surveillance done by Coppi D. , Calderara S. and Cucchiara R. [CCC11, appearance tracking]

They tried to face the one target tracking problem using a semi-supervised transductive learning on a graph based approach specifically by applying spectral graph theory. Initially few labeled samples of the target will be given to the algorithm then the goal will be searching for the appearance of the target in the next frame sequences of the video . Since the method is working iteratively on each frame of the video sequence , in the middle of the process there is a target model update because naturally (target)persons in videos are constantly changing their appearance way so learning a new model for the target is a critical thing to achieve accuracy . To avoid the huge amount of the

target models learned in each iteration, there will be done a clustering of the target models as a result similar models belong in the same cluster and only one target model will be selected from each clusters as a representative, so this evolutionary clustering method saves a huge amount of storage space. When we come to modeling of the people appearances, After detection of people from each frame using HOG (Histograms of Oriented gradients) based people detector ,they used covariance matrices on gradient and color information for each detected patch. The top view of the system looks like as it is shown in the following figure.

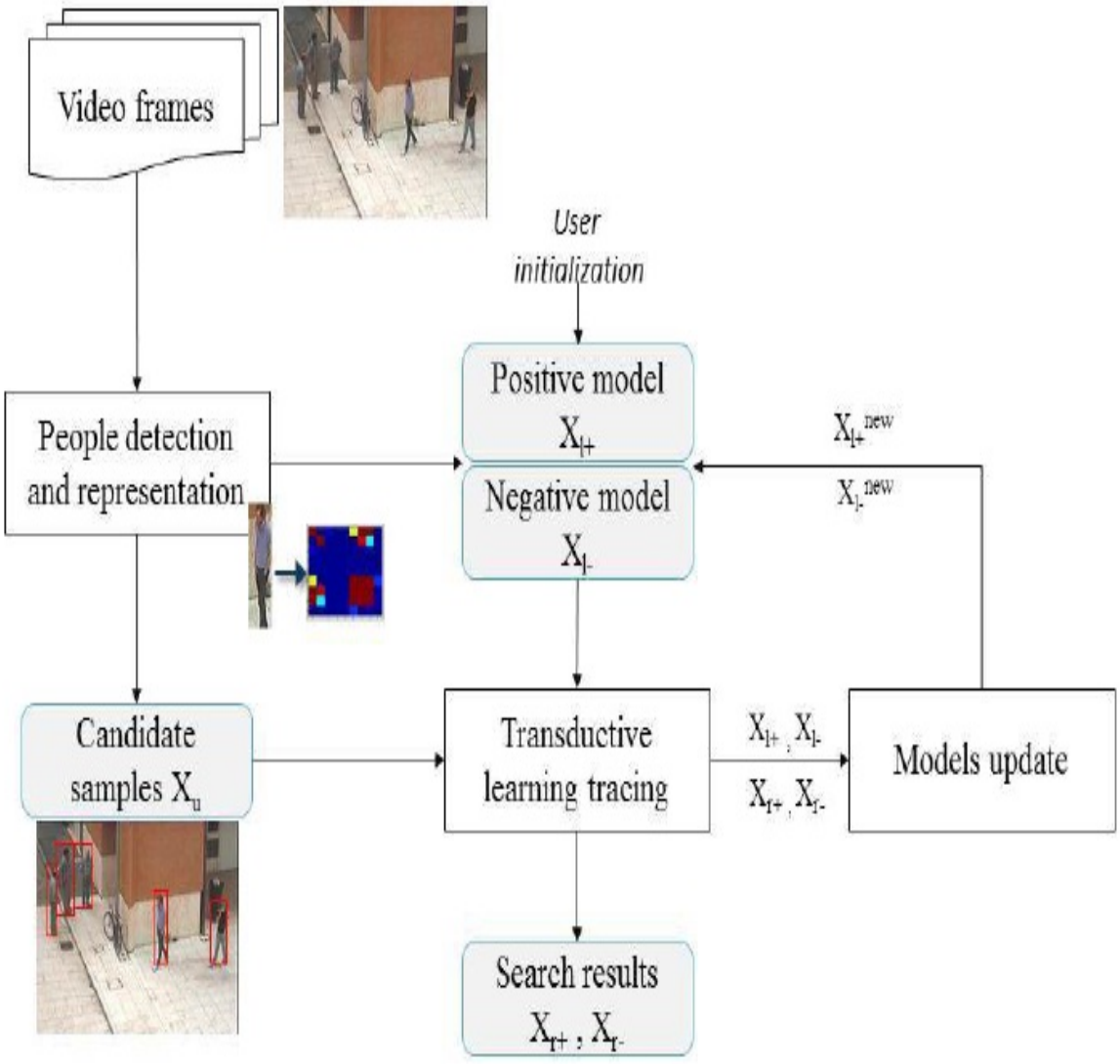


Fig. 5.1: Top View of the approach. Adopted from [CCC11]

6. OUR CONTRIBUTION

Our main contribution in this work is we applied the *Graph Transduction as non-cooperative Game* (GTG), which is a general frame work for multi node classification based on a game theoretic approach, proposed by Erdem A. & Pelillo M. [EP12] for a well known computer vision problem specifically for multiple object/people tracking in videos. We studied this frame work's application on tracking multiple people on video surveillance scenarios and get promising result in terms of the accuracy of target tracking.

let's review our work section by section.In the first section we start with the preliminaries ; a toy example of graph transduction utilizing relaxation labeling using the help of replicator dynamics and also we will try to see it relating with the notion of game theory in which the combination of the mentioned concepts constitutes the back bone of the frame work . And then on the second section, we show how the analogies in the first section of this chapter can be related to the problem of computer vision (i.e tracking multiple people on video surveillance scenarios). Finally on the third section we show the experimental results on real world video datasets.

6.1 Graph Transduction for a Classification Problem

As we have seen in the previous chapters Graph transduction is an instance of a graph based semi-supervised learning technique. Let's see one simple toy example to visualize and make things clear about our concept of building the application. Assume we have the following weighted graph $G = (V, E)$ and let's deal with a two class problem ,

formally the class environment/space domain can be given by $C = \{classA, classB\}$. (note that the problem formulation can handle multiple class problems, but two class problem is used here for the simplicity of explanation). The nodes are divided in two disjoint sets namely labeled nodes (V_L) and unlabeled nodes (V_U) respectively, where $V_L = (v_1, v_3)$ & $V_U = (v_2, v_4)$

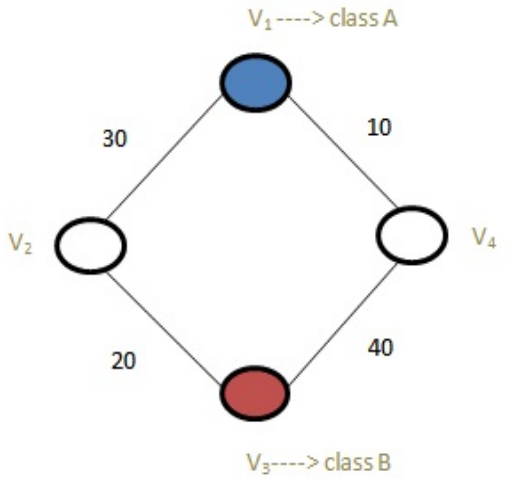


Fig. 6.1: Weighted graph with labeled (v_1&v_3) and unlabeled nodes (v_2&v_4).

V_1 belongs to class-A (i.e. $V_1 \in class-A$), V_3 belongs to class-B (i.e. $V_3 \in class-B$), inother words they know their class membership/ labels already, while V_2 and V_4 don't know their class membership yet. At this time we assume we already got the weights for the edges of the graph, but later we will discover how these edge weights are computed. The weights will be put as a similarity matrix and This weight/ similarity matrix between the nodes is the most important input to our algorithm because it holds critical information about the compatibility of the final labeling assignments of the nodes. it is a vital input resource.

Now let's shift our perception to game theoretic problem formulation and consider the nodes of the graph as game players (i.e the four nodes of the graph now became the four players participating in a non-cooperative transduction game).

let's get back to our previous simple weighted graph, and summarize it's property.

- The Adjacency/ Similarity/ Weight Matrix

	V1	V2	V3	V4
V1	0	30	0	10
V2	30	0	20	0
V3	0	20	0	40
V4	10	0	40	0

Tab. 6.1: Similarity Matrix

- Node 1 and Node 3 are labeled, belonging to 'class A' and 'class B' respectively (i.e they know their class membership), in other words player 1 and player 3 already know their decisions as a result they are playing their pure strategy. Here notice that, at this moment of our journey the terms '*Node*' and '*Player*' are equivalent and can be used interchangeably.

- Node 2 and Node 4 are unlabeled (i.e player 2 and player 4 didn't decided their strategy which can yield them a higher pay off yet, out of their strategy profile which is either selecting 'Class-A'/'Class-B' or a mixed strategy). In the beginning(i.e. at time t=0) they don't know their class member ship.

- Initial probability distribution matrix ($P^{(0)}$) i.e. at time t = zero (which is mentioned on 'Nonlinear Relaxation labeling' section of this thesis) will be like the figure below . $P^{(0)} \in \mathbb{R}^{N \times M}$. where N is the number of players and M is the number of labels (class) existing in the current configuration. In this specific toy example, because there are four nodes playing to maximize their payoff/reward by choosing a decision or selecting a best strategy from the available two strategies which is either be a member of '*class-A*' or '*class-B*' , N will be four and M will be two. The relaxation labeling process takes this $P^{(0)}$ as an input.

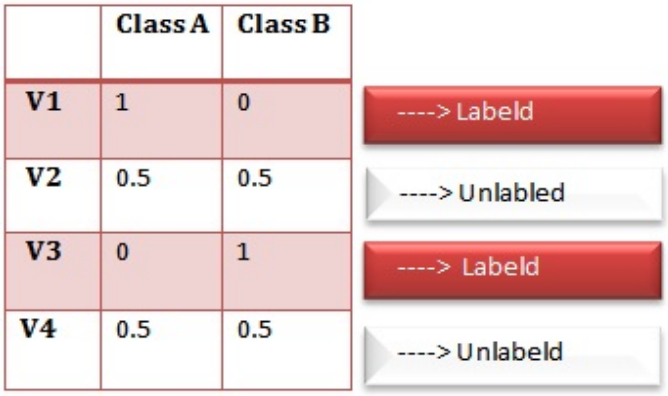


Fig. 6.2: The initial probability distribution of the decisions of the players i.e $P^{(0)}$

At this step the main goal is to propagate the information from the labeled nodes to the unlabeled nodes consistently by relaxation labeling using the discrete-time replicator dynamics (i.e the equation given below). In the standard way of graph transduction, the mixed strategies of each unlabeled player are initialized to uniform probabilities, but this is not necessarily a must case,the mixed strategies of the unlabeled players can be initialized to non-uniform probabilities, we will address this issue later .

$P_{ih}(t+1) = P_{ih}(t)\frac{u_i(e_i^h)}{u_i(P(t))}$

Once equilibrium is reached, the label of a data point (player) i is simply given by the strategy with the highest probability in the equilibrium mixed strategy of player i as

$$ClassLabel\ _i = \underset{h=1...c}{argmax}\ P_{ih}$$

The edge weights/similarity in the above graph gives us information on compatibility coefficient or Payoffs that the players receive as a result of their decision in combination with the other players decision. If we see the nodes as the game players and their choice of class membership as their strategies, the labeled nodes (players) play pure strategies (because they already know their class membership)i.e. they already decided a strategy

before the game begins, while the unlabeled players(nodes) initially don't know their decision as the best strategy, so later with the information guidance of the edge weight which corresponds to compatibility coefficient or Payoffs they will decide their choice. They might play mixed strategies, as mentioned above to extract the final assignment labels for the unlabeled players/nodes, we select the pure strategy with the highest probability among the mixed strategies selected as a best decision which is in equilibrium state. After some iterations, the algorithm converges and equilibrium will be reached taking a big account of the pay offs. This equilibrium corresponds to Nash's equilibrium , if we see it from the computational aspect, this Nash equilibrium corresponds to the fixed points of the multi-population version of the replicator dynamics.

And the final output of the player's class membership (or final probability distribution of the decisions of the players) in other words the final consistent labeling assignment will be the following.

Fig. 6.3: The final probability distribution of the decisions of the players, i.e. the final consistent labeling assignment $P^{(t)}$

As we can see from the output of the algorithm, for the two players which are already labeled initially, the information we got is zero entropy, because since the beginning , their labeling (i.e the strategy they are going to play for the whole time) is known in advance. But the most important new information which at first every body was not

sure is obtained from the output of the final labeling assignment for the unlabeled nodes. This is crucially significant in our case and in most of real world scenarios because the population of unlabeled entities are by far higher in number than the labeled ones.

6.2 Applying the above analogy to a computer vision problem

This section explains about how the above frame work of graph transduction as a game can be applied to tracking multiple people on video surveillances. We will see how we constructed the bridge between the two concepts.

As we all know, Videos are composed of multiple frames and in each frame there might exist many objects like people, animals , cars e.t.c. Since we are interested in people tracking from now on we will focus on people from the video data. First we need to transform the videos to be processed to a graph form model where the nodes indicates a detected person / patch from each of the whole sequence of the frames of the video. The same person doing any type of trajectory or in a stationary state, which his/her geographical position is in the coverage area of any video surveillance cameras like CCTV (closed circuit television) will appear in the consecutive frames, as a result when our transformation of the video to a graph model happens, this same person will correspond to more than one node in the newly formed graph. Let's see this transformation concept with example. assume we have the following two hypothetical frames of a video, and let's transform it to a graph model and see how does it look.

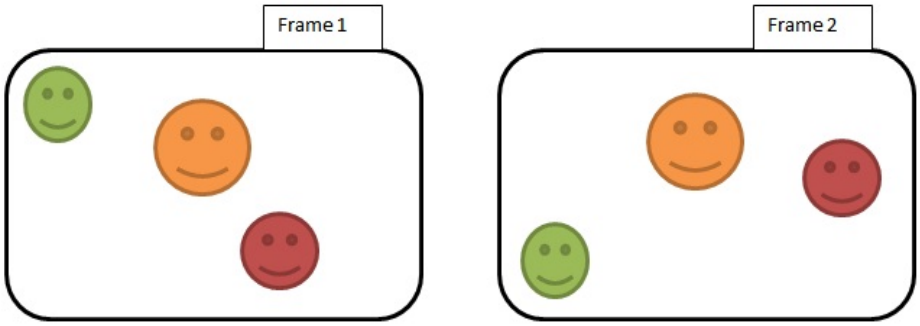


Fig. 6.4: two hypothetical frames of a video where 3 persons in each frame exist

In the above two hypothetical frames of a video, in the first frame the three faces with different color indicates, three different people. Also in the second frame these same people exist but their geographical positions changed. The first thing to do the transformation is, detection of the existing people from each frame. Here Any kind of people detectors can be used as per one's desire. In our case we used a people detector based on HOG (Histogram of Oriented Gradient)[DT05]. At this moment we have three patches/people extracted from each frames so in total we have six patches.

Fig. 6.5: Hypothetically detected patches/people from the above frames

The transformed Graph model of the above two frame video will look like the following.

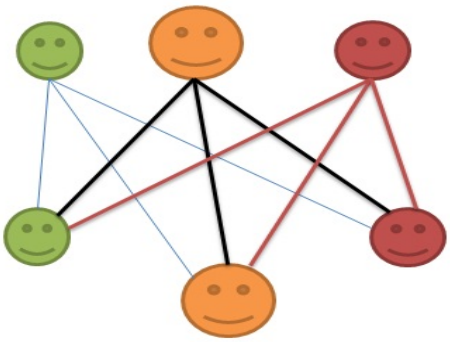


Fig. 6.6: A hypothetical 2-frame video represented as a Graph

The real valued weighted edges in the above figure indicates the similarity between the patches. higher valued weight corresponds to high degree of similarity and lower ones correspond to low degree of similarity. Zero valued edge weight indicates there is no similarity between the corresponding nodes taking into account what similarity measure is used. For example if we give labels to the people in the first frame of the video as target 1, target 2 and target 3 , and try to track or propagate information to the unlabeled nodes of the graph, the graph will be having the following property : there will be a high degree of similarity between node 1 of the first frame which is labeled as target, and the node of this same person in the second frame but which is unlabeled as a consequence after the convergence or consistent labeling has been done, we can see that this unlabeled node which is found in the next frames will be classified correctly indicating that he/she is the target which is labeled in the first frame.

Here an important question will come up; how we evaluate the similarities? the answer will be, the detected patches should be represented by some kind of models with features. And we adopted to represent these detected patches from each sequence of frames of the video by covariance matrices which lie in the reimaninan space, so it means the descriptor of the nodes/players is covariance matrix. By applying Gausian kernel on their Eigen difference we manged to get the edge weights of the graph, and by game theoretic approach this weights are like the payoff informations. Let's see it in a detail.

$$W_{ij} \quad = exp^{\frac{-(\mathbf{CovarianceDistance})^2}{2\sigma^2}}$$

How to compute the weight (similarity/ payoff) matrix

In the first section of this chapter, we directly jumped and started with already weight assigned graph. But the question is , how we compute these weights? these weights are so much important factors because they hold payoff informations.what we are trying here is to map the problem based on the notion of Game theoretic approach and to use these normalized weight/similarity matrix as a payoff information . To build similarity matrix between the nodes(or detected patches or the players), each of them should have a descriptor. Once we have achieved a descriptor for each nodes, we can build a similarity matrix between them by comparing the differences in their descriptors.

How to compute the descriptor for each player (i.e detected patches)

So far as per our pre-assumption, one of the main inputs to our framework are the patches of the people from the frame sequences of the video data, therefore logically the primary step should be people extraction from each frame of the video sequence. So we shall use patches detected by using people detector based on Histogram of Oriented Gradient (HOG), when we test our algorithm. Also many other people detectors might be used.

- Representing the players (i.e nodes of the graph) by Covariance matrix

We used the same procedure for covariance representation of the detected patches from the frames of the video sequence as it is used in [CCC11]. After the patches are extracted as a snapshot from all the frames, a descriptor is computed and this descriptor is used to match the varying appearance of the same person and also with other persons among from the multiple frames.

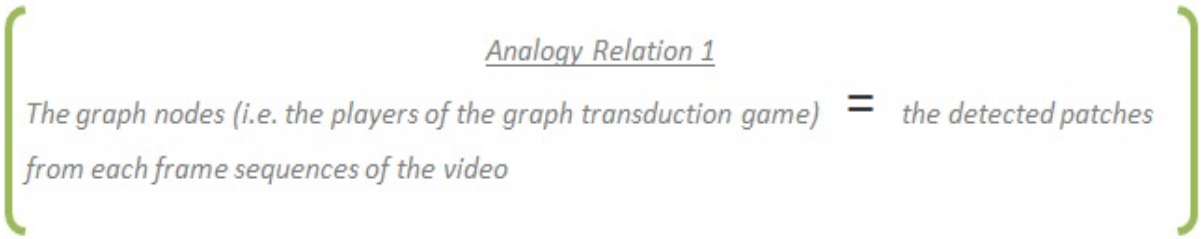


Fig. 6.7:

The simplest descriptor to represent the patches of the extracted people (i.e the players of the game) is a color histogram. However this method is not able to distinguish between two persons wearing same colors but not in same position , the reason is that it won't take into consideration shape and location information. To overcome this limitation, covariance matrix descriptor is adopted since this descriptor considers and takes into account shape,position cues and color informations too [PTM06]. Covariance matrices have got important characteristics that they are rotation and scale invariant also not affected by average pixel intensity change.

The covariance matrix is a square symmetric matrix $d \times d$, where d is the number of selected features independently from the size of the image window, carrying the advantage of being a low dimensional data representation. Given the covariance matrix C its diagonal entries represent the variance of each feature and the non-diagonal entries represent the correlations.

Considering 'Img' as color image with 3 dimensions and B as the $W \times H \times d$ dimensional feature image extracted from 'Img'.

$B(x, y) = \varphi(Img, x, y)$

where φ represents any mapping(i.e intensity, color, gradients, filter responses, etc.). Let $\{Z_i\}_{i=1...K}$ be the d-dimensional feature points inside B, while $K = W \times H$. The image 'Img' can be represented by a covariance matrix of $d \times d$ dimension.

$C_R = \frac{1}{K-1} \sum_{i=1}^{n} (z_i - \mu)(z_i - \mu)^T$

Where μ is the mean vector for the features. In our case Z_i is the feature vector composed for each pixel by its spatial, color and edge information. We use x and y pixel location in the image grid, HSV color values, Gx and Gy first order derivatives of the intensities calculated through Sobel operator w.r.t. x and y, and the magnitude $mag(x, y) = \sqrt{G_x^2 + G_y^2}$ and the angle o(x,y) = $arctan(\frac{G_y}{G_x})$ of the first order derivatives. Therefore each pixel of the image is mapped to a nine dimensional feature vector

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

Based on this features vector the covariance of a region is a 9 x 9 matrix, as represented in the figure below .

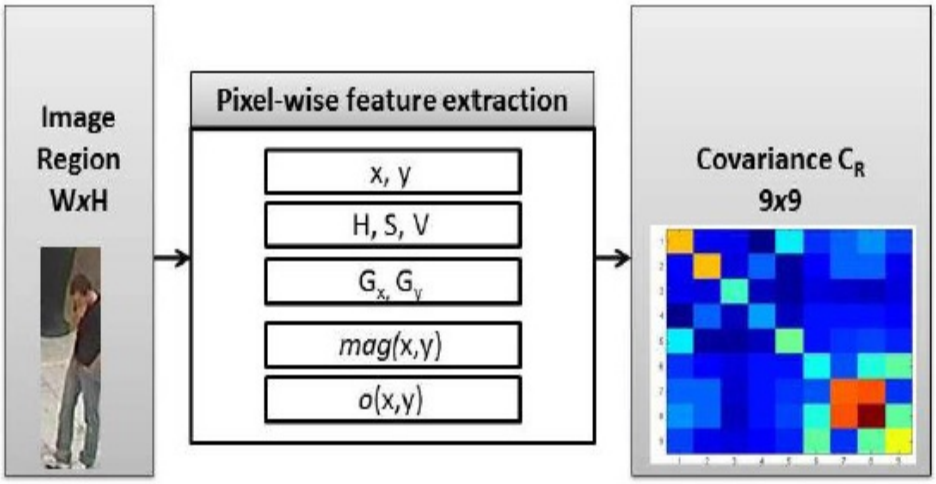


Fig. 6.8: Representation of a patch (i.e player) by Covariance matrix. Adopted from [CCC11]

It should be noted that HSV color space is used instead of the basic RGB color space because a group of researchers from Modena university experimented and saw a higher invariance to scale and light changes of the HSV components when compared to RGB.

Beyond the adopted feature vector, an adequate distance between covariance matrices must be defined to assess the appearance similarity between candidates regions and the target. However, the covariance matrices do not lie on the Euclidean space and arithmetic subtractions or simple operations between matrices are not correct. A robust distance metric between the covariance matrices is used as the sum of the squared logarithms of the generalized eigenvalues as proposed by [FM99]:

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

where $\lambda_k(C_i, C_j)_{k=1...d}$ are the generalized eigenvalues of C_i and C_j computed as: $\lambda_k C_1 x_k - C_j x_k = 0 \quad where \quad k = 0...d \quad and \quad x_k =$ the generalized eigenvectors.

The distance ρ gives us the similarity measure (i.e the weight) between the patches (i.e the players) which holds the compatibility information or payoff of their choice of decision strategy.

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 graph transduction.

After this we label only a few of the patches i.e. indicating to which target the patches belong to. Then we run our algorithm giving the weight matrix and the label information as input and get the final classification for each unlabeled patch as an output.

In our experiment we saw that even labeling only the patches detected in only one of the frames of the video sequence, can be enough to get a good result on tracking or identifying the persons.

6.3 Experiments Carried On

In this section we assess the performance our people tracker which is able to track multiple targets at a time. We evaluated the degree of reliability of our approach by testing on videos datasets THIS[thi], 3DPes[3DP] and CAVIAR[cav] .

In our setting we provided the patches (i.e. the detected people on the scene i.e on each video frames) that is obtained using the HOG based people detector and after we transformed the video sequences to graph models where the nodes representing the detected patches. we labeled few frames followed by computing 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/players then we normalized it. We used this normalized similarity matrix as payoff matrix for our set up of game theoretic approach . Next step we run the 'graph transduction as a game' algorithm after that we called 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 out put. To evaluate the accuracy we

further feed the ground truth which holds the actual target information for each patches detected and compared it with the output classification result(i.e the final consistent labeling assignments VS. the ground truth) Videos from CAVIAR and THIS datasets, precisely "Clips from shopping center in Portugal - Corridor view" and "Train Station" respectively, are recorded along the hallway of a shopping center and along the platforms and underpasses of a train station, they include people walking alone, meeting with other and entering or exiting gates.

Our approach of the experiment is based on two set ups.

A. Off-line mode

The first one is off-line detection and tracking of multiple people/object. In this set up we will process already recorded video footage from surveillance sensing which we know the beginning and the ending frames of the video. And 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 what happened previously like forensics and other similar domains. for example if we have 'N' frames of a particular video. the set up will look like the following.

1 . . . . LF	LF+1 . . . . N

LF is to denote 'Labeled Frame'. Only the first frames(from 1 to LF) will be labeled and the rest (from LF+1 to N) will be unlabeled. the goal will be giving a consistent labeling(tracking) for the unlabeled one in graph transduction manner as discussed above. And the weight/similarity graph (matrix) which serves as a payoff matrix for our game setup will be with the dimension $Np \times Np$, where Np is the number of patches detected from the N frames .

Or the desired number of labeled frames might be in the middle of the video, then

these informations will propagate back and forth to the rest of the unlabeled frames.

. . . . UF	. . LF . .	. . . . UF . . . .

'LF' is to denote 'labeled frames' and 'UF' to denote 'unlabeled frames'

or frames can be selected randomly from the whole sequence and the detected patches belonging to these frames be given target labels. a user might use some annotation to label the specified frames.

. . . . UF	LF	. . . . UF	LF	. . . . UF	LF	. . . UF . . . .

In the whole set up, the number of labeled frames(LF) are so much less in number when compared to the unlabeled frames (UF). To give specific examples a labeled frame might be one(1) out of 140 frames, which means 139 frames unlabeled and only one frames is labeled. And this information from the only labeled one propagate to the rest of 139 unlabeled frames. it can be seen as a similarity based pattern recognition. In our experiment we never used more than 8 labeled frames to minimize human intervention.

B. On-line mode

The second mode is on-line detection and tracking of multiple targets. In this set up, unlike the off-line mode we will not process already recorded video footage from surveillance sensing which we know the begging and the ending frame of the video, instead we will process some fixed amount (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 off-line mode. it is approximately near real time and it's application will be quite meaningful in a scenarios where real time knowledge is necessary . it's usage can be applied on the areas where surveillance security is desired to have a knowledge of which persons are existing , moving or doing

something in a particular place (the place may need a high security or might be a sensitive place) and other similar domains.

6.3.1 Evaluation Measures

Our evaluation measure is based on **Precision** , **Recall** and **Accuracy** of the classification of each patches , in other words ,the class for each patch (assigned target number as an output by the system for each patch) will be compared with the ground truth (the actual class / target of the patch).

. **Number of Total patches (NP)** : the number of total patches/detected people from the video sequence including the labeled and unlabeled ones together with their ground truth.

. **Number of Total Frames (NP)** : The number total frames to be proccesed in the offline/online tracking set up.

. **Number labeled Frames (NP)** : the number of frames that are labeled manually. labeling in our context, it means assigning a probability distribution which lies on the vertex of the standard simplex. for example ,let see what labeling means from technical point of view. Assume we want to label one frame and in this frame the people detector (in our case we used HOG-histogram oriented gradient based people detector) detected 4 people, so we provide this matrix which is hard labeled as an input.

0 1 0 0

0 0 0 1

1 0 0 0

0 0 1 0

There are 4 rows because there are 4 people (i.e the people detector detected 4 patches of people). The columns indexes indicate the target . this matrix represents patches with their associated target number. The semantic of the above matrix is as

follows: patch 1 is target 2, patch 2 is target 4, patch 3 is target 1 and patch 4 is target 3.

. **Number of target(NT)** : how many targets to be tracked

. **Labeled patches(LP)** : the number of labeled patches. the number of unlabeled patches will be NP-LP

. **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.

6.3.2 Testing on Ideal case

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 come because,we took the first 49 frames of the 'CAVIAR' video data set where the HOG-based people detector detected a good and full detection the persons with out partial detection and error(some times the detector detects plain backgrounds as a person, this is what it leads to classification error, which makes our algorithm confuses the real target and the plain patches detected as a person but which are not persons.)

Fig. 6.9: perfect sample out put of our algorithm on 'CAVIAR' VIDEO, where unambiguous detection of persons in the frames happened.

	Number of frames	Precision(%)	Recall(%)	Accuracy(%)
our method	49	100	100	100

Tab. 6.2: Performance on selected frames from CAVIAR video dataset, where the HOG-based people detector returned unambiguous detections of patches

Fig. 6.10: perfect sample out put of our algorithm on selected frames from 'THIS' VIDEO data set. With 100 percent PRECISION, RECALL AND ACCURACY

6.3.3 Experiments for off-line approach

In this experiment set up we performed an off-line tracking approach, where following the assumption that we have already a recorded video footage. and we tested this on 140 frames of the caviar video dataset. initially we randomly labeled few frames which is an input for our algorithm. and we tired to show the effect of the number of labeled frames on the precision and recall of the result. we wanted to show the result that can be obtained by randomly labeling frames. we randomly selected and labeled 1, 3, 5 , 8 and 20 frames from the 'Caviar' video frames which is obtained from surveillance camera. we did this test 20 times for each and got the result as shown in the figure below.

As it can be seen from the graph by labeling 5 random frames , we became able to get a good precision/recall values meaning we achieved almost perfect multi-target tracking.

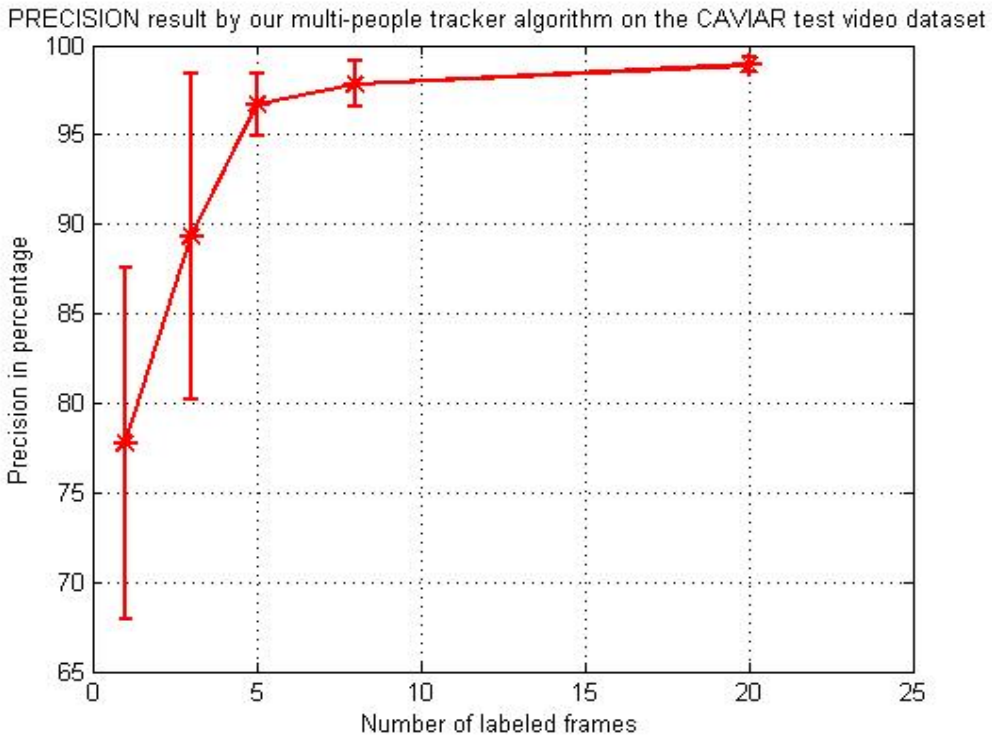


Fig. 6.11: precision graph for 'CAVIAR' VIDEO

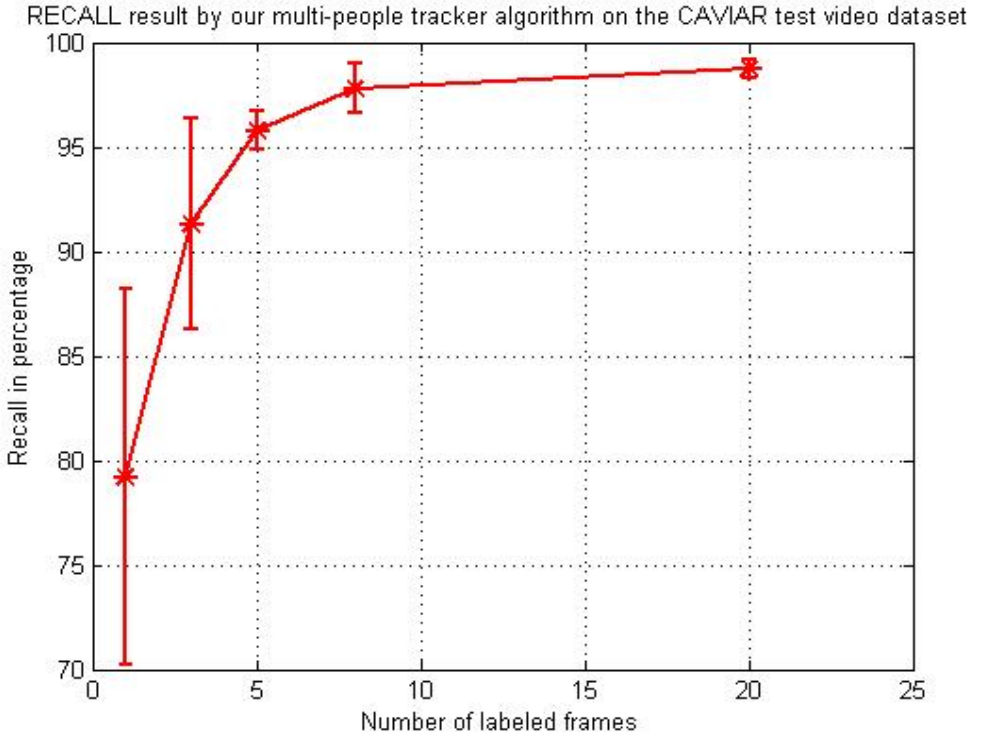


Fig. 6.12: Recall result on 'CAVIAR' VIDEO.

6.3.4 Experiments for on-line approach

Our method can handle multi target tracking on-line where the input videos are streaming by building temporary weight for the specified window size. window size means the number of unlabeled frames to process at one iteration.

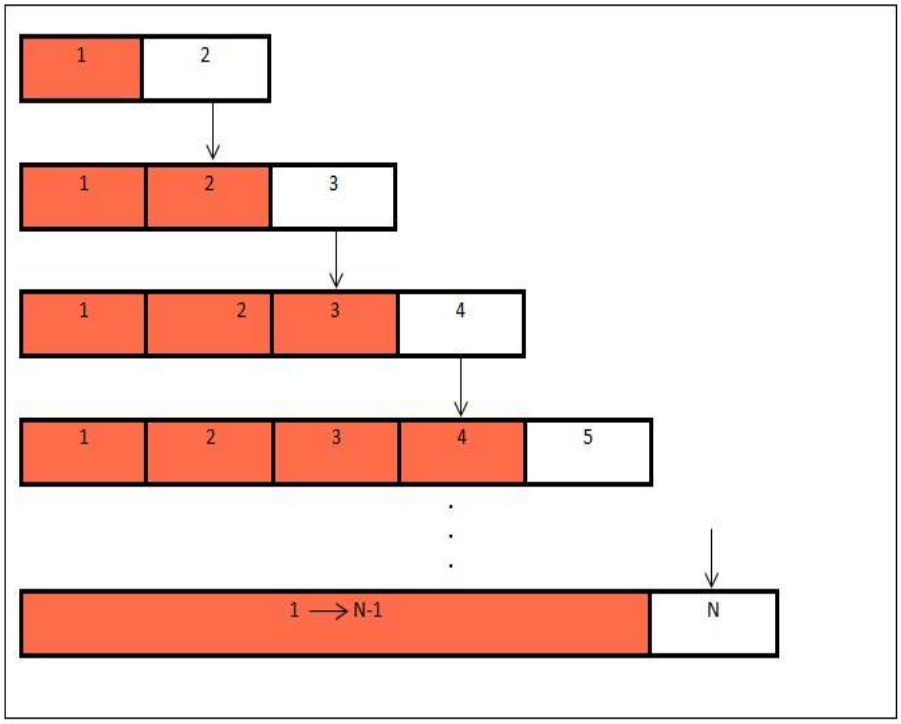


Fig. 6.13: frame structures for window size 1

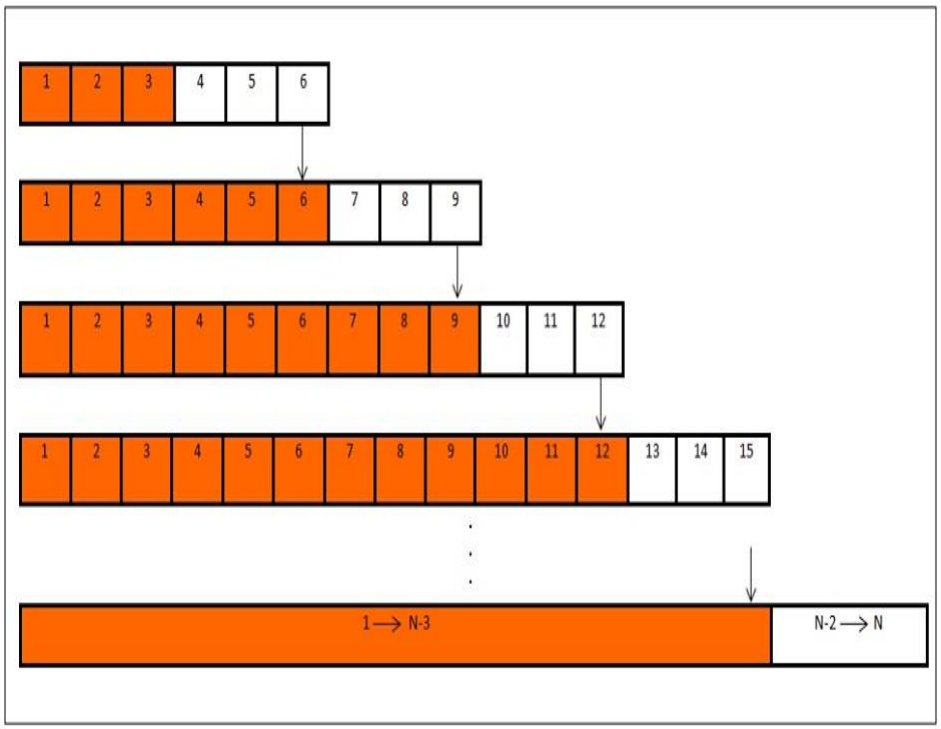


Fig. 6.14: frame structures for window size 3

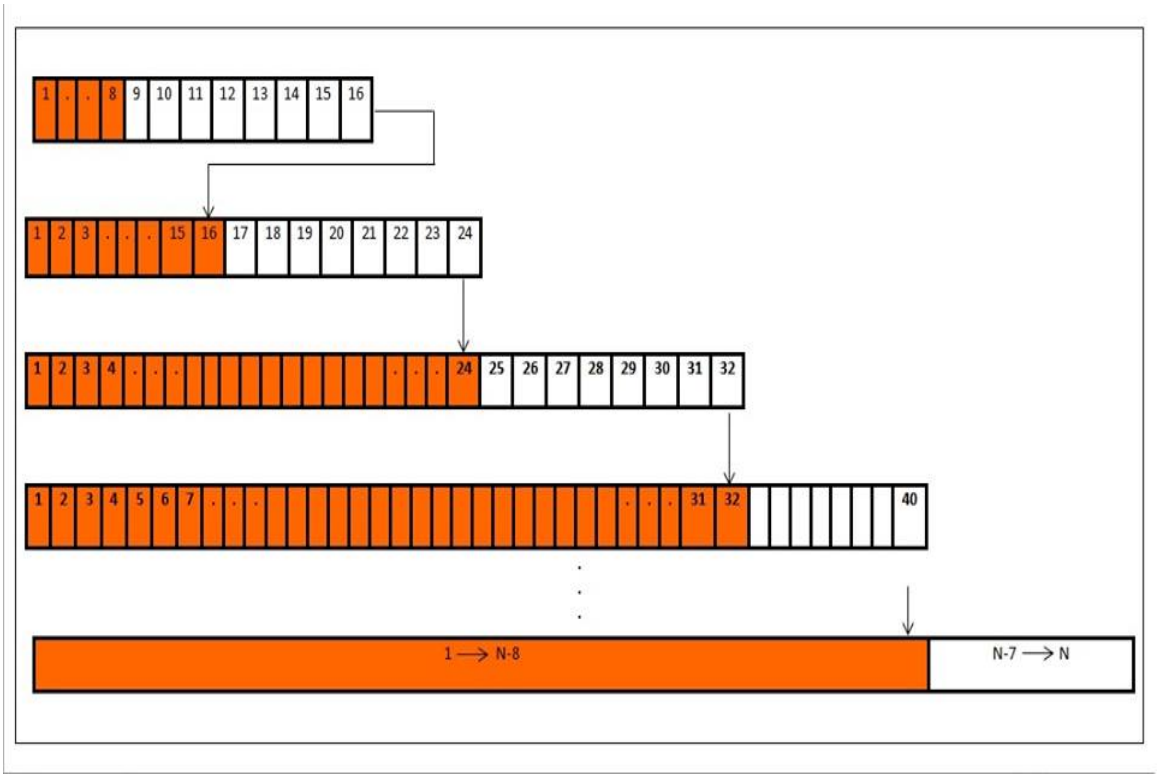


Fig. 6.15: frame structures for window size 8

The colored parts indicate labeled frames and the white parts indicate the unlabeled frames. There will be a linearly growing number of frames though out the tracking process. label assignment provided by the algorithm will be stored for learning the algorithm with the update of the model of the targets.

we prepared this frame data structure for memorizing the label assignment for the unlabeled nodes from the previous iteration so that this result will act as input labeled data at current iteration because of the reason that, at a current iteration the previous frames which were unlabeled in the specified window size are now labeled and will be used to give label information for the current unlabeled frames which are found in the specified window size.

Window Size	Number of frames	Precision(%)	Recall(%)	Accuracy(%)
1	93	84.33	70.42	
3	93	100	100	100
8	93	100	100	100
10	93	100	100	100

Tab. 6.3: Performance on 'THIS' video dataset,varying the window sizes

Window Size	Number of frames	Precision(%)	Recall(%)	Accuracy(%)
1	140	75.58	76.9	87.5
3	140	92.21	98.43	97.88
8	140	92.21	98.43	97.88
10	140	86.9	91.95	92

Tab. 6.4: Performance on 'Caviar' video dataset,varying the window sizes

as we see can see, we got almost perfect result on 'THIS' data set by adjusting the window size and of course the patches detected from the frames are unambiguous, i.e each persons are detected more than 75 percent their appearance on it. While in the 'CAVIAR' video there are ambiguous detection of patches of the persons in most frames i.e in one box there are some parts of two persons and in this case the window size matters, because using window size of 1, we will be dealing with many labeled labeled

frames and only one unlabeled frame as a result there can't be enough learning from the unlabeled datas, so as a result some classifications with errors might happen and this information will be put in the memory where the labeled data are collected, as a consequence in the next frames the errors will propagate. On the other hand using large size of window size also affect the classification result in the case where there are ambigious detection of people, because if there are errors in the previous iteration that will be used as a labeled dat in the current iteration, the error will propage to this large amount of window size, and this error full result again used as a labeled data for the next iteration for labeling the coming unlabeled nodes which will be put in this large size of window as consequence again the errors will propagate. so maintaining the right size of window size is neccesary for better performance

6.3.5 Accuracy testing

In this specific testing We used 100 frames from CAVIAR dataset and 109 from THIS dataset for our analysis. We gave labels only for the first frame of the videos .

THIS Dataset

	NP	NT	LP	Accuracy
PTGTG	270	3	3	0.8090

CAVIAR Dataset

	NP	NT	LP	Accuracy
PTGTG	407	4	4	0.9603

Fig. 6.16: Accuracy result. PTGTG:- people Tracking as a Graph Transduction Game which is our approach

(a) CAVIAR

(b) THIS

Fig. 6.17: Sample frames from 'CAVIAR' and 'THIS' video datasets.

Fig. 6.18: Sample multiple person detection/tracking result from 'CAVIAR' video

6.3.6 Comparison of our Method with Others

Here we compared our work's Precision and Recall with others testing it on '*CAVIAR*' Video dataset.

	Number of frames	Precision(%)	Recall(%)
[WN07]	140		75.2
[ZLN08]	140		76.4
[XAL]	140		81.8
[HWN08]	140		86.3
[LHN09]	140		89.0
[KHN10]	140		89.4
[CCC11]	140	94.0	95.0
Our Method	140	92.21	98.43

Tab. 6.5: Performance Comparison on CAVIAR video dataset

6.3.7 Experiments on tracking multiple targets with hard circumstances

The following video datasets 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 and wearing similar cloths. And from the result we can see that our algorithm can handle this kind of scenarios. Qualitative results are presented below.

Tracking targets with changing appearance

This experiment was carried on to show that our algorithm can handle targets with changing appearance (changing dresses) and poses .

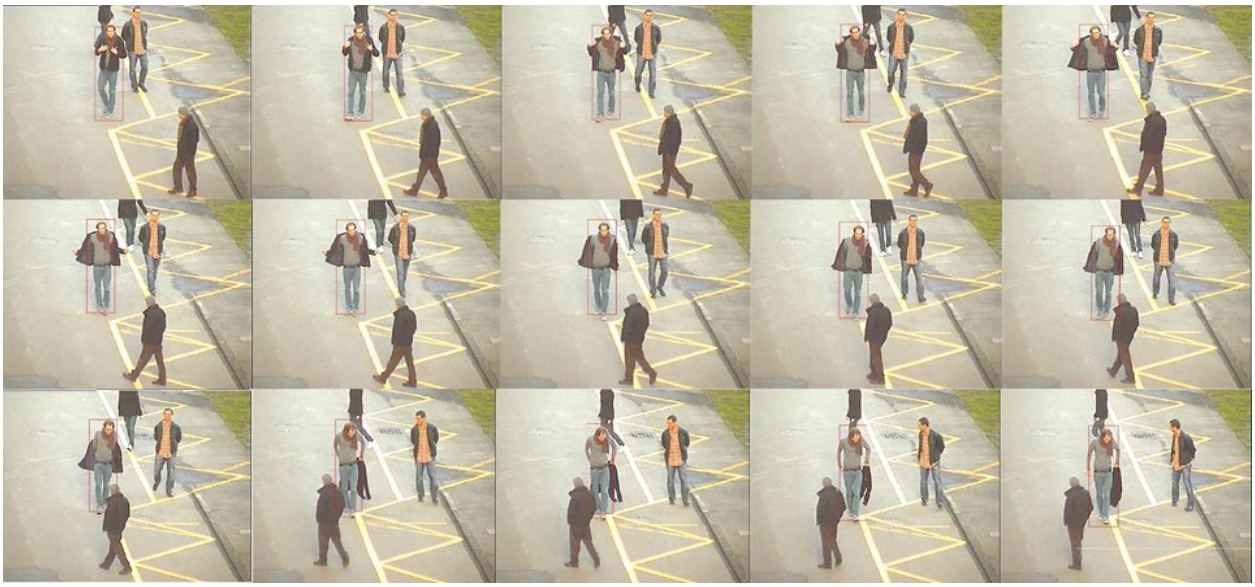


Fig. 6.19: Tracking of a target with changing dress

Experiments on small sized detected patches together with occlusions occurring

Fig. 6.20: sample output video frames of our algorithm where small sized detection of patches taken as input.

As it can be seen from the figure above our Algorithm shows good performance even when there is small sized detection of patches .

6.3.8 Experiments on Re-identification of a target

Re-identification of a person is the most common difficult problem on the area of object/people tracking. Re-identification means as we can perceive from it's name it is a process involved in identifying a target again, after the disappearance of the target for some period of time. This disappearance of the target for temporary time usually happens because of people occluding each other (i.e if the target is blocked by another object/people for some time from the view of the camera, e.t.c) or the target might be out of sight of the coverage area of the camera for some time and come back again. Or some times the HOG detector might not detect the target person for some frames in the middle.

we have done the following experiment to check that our approach can deal with the mentioned problems. and as it can be seen from the figures below it can handle very well this type of problems.

The scenario : one of the target person(the target initially tracked by the blue box) is covered by the other target person (the target tracked by yellow bounding box) for some period of time, then after when they separate and both of them become in a clear sight for the camera, our algorithm re-identifies the target who was previously covered by the other target.

Fig. 6.21: The original video scene.

Fig. 6.22: The tacking output result of our algorithm. Re-identifying one of the targets (i.e the target bounded with blue rectangle)

7. LIMITATIONS OF THE CURRENT SYSTEM AND FUTURE WORKS

Our current system is good on performance of recall measurement meaning targets will be recognized or will be classified correctly in most cases. In another words there are minimized number of false negative predictions. the lesser false negatives, the better performance in terms of recall. On the other hand,if there comes a new person and detected in the scene/frame which we are not interested to track him/she, eventually it will be classified as one of the targets but in reality this person is different. This implies there might be a higher number of false positives affecting the precision. The higher false positives the lesser precision result. This happens only for multiple target tracking because in our system in the initial configuration we need to specify how many targets to track. But for single target tracking this is not a problem because we have either a target or not(i.e we are dealing with only a binary classification problem). As a future work to tackle this problem we thought of applying a clustering technique(may be dominant set clustering, multi-objective similarity algorithms) on the patches detected as a consequence when a new person is detected ,the compatibility of this patch to belong to one of the clusters of the targets will be checked thus by avoiding false positives and increasing the precision of our system.

The other future work we thought is replacing the replicator dynamics which we used in our current system for computing the 'Nash equilibria' of the scenarios(i.e consistent labeling of the unlabeled patches/nodes/players resulting the tracking) by another faster dynamics for better performance in terms of time.

And at last, like the classical methods in graph transduction, our current system configuration uses initial uniform probability distribution for the unlabeled nodes. To introduce novelty We thought , instead of using uniform probability distribution, we planed to use the position information of the targets in the current frame by comparing it with the positions these targets got in the previous frames as the initial probability distribution. And we thought this also might bring a better performance because this prior information might help backed up by the payoff matrix which is the similarity measurement and the tracking problem might get better.

8. CONCLUSION

We presented a system that do on-line and off-line multi-target tracking utilizing graph transduction based on a game theoretic approach on the context of video surveillance . The off-line mode might have a potential application on the area of video forensics, where investigation of scenes on already recorded video footage takes place, on the other hand the on-line approach might have a potential application on the areas where real time knowledge is important(like security by surveillance, air port security, museum protection, e.t.c.) by processing in coming video streams instead of already recorded footages. And we showed qualitative and quantitative output results of our algorithm by making tests on some video data sets and got promising good results. The novelty of this work is on the point that it uses graph transduction learning technique utilizing the notion of game theoretic approach for multi target tracking.

we saw that our system is heavily dependent on the input detections of persons, the more unambiguous patches are detected, the more the classification result would get better even though there will be persons with similar appearance(i.e similar pose , similar dress colors, e.t.c).

We also saw the manipulation of the unlabeled and labeled frame data structures in the case of on-line tracking and choosing the right size of the window would result better performances benefiting from both labeled and unlabeled learning among the data.

BIBLIOGRAPHY

[3DP] 40

[AR76] Steven W. Zucker Azriel Rosenfeld, Robert A. Hummel. Scene labeling by relaxation operations systems. 1976. 16

[AYS06] O. Javed A. Yilmaz and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 2006. 24

[BRL+09] Michael D. Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *IEEE International Conference on Computer Vision*, October 2009. 25

[cav] 40

[CCC11] D. Coppi, S. Calderara, and R. Cucchiara. People appearance tracing in video by spectral graph transduction. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 920–927, 2011. vii, 25, 27, 36, 39, 55

[DJ11] Arnaud Doucet and Adam M. Johansen. A tutorial on particle filtering and smoothing: fifteen years later, 2011. 25

[DT05] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01, CVPR '05, pages 886–893, Washington, DC, USA, 2005. IEEE Computer Society. 34

[EARJTAEL06] Boulogne T. El-Azouzi R. Jimenez T. Altman E. and Wynter L. *survey on networking games in telecommunications.Computers and Operations Research*. 2006. 14

[EP12] Aykut Erdem and Marcello Pelillo. Graph transduction as a noncooperative game. *Neural Computation*, 24(3):700–723, 2012. 19, 28

[FM99] Wolfgang FÃ¶rstner and Boudewijn Moonen. A metric for covariance matrices, 1999. 39

[HWN08] Chang Huang, Bo Wu, and Ramakant Nevatia. Robust object tracking by hierarchical association of detection responses. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, pages 788–801, Berlin, Heidelberg, 2008. Springer-Verlag. 55

[J.W95] J.W.Weibull. Evolutionary game theory. mit press, cambridge ma. 1995. 22

[KHN10] Cheng-Hao Kuo, Chang Huang, and Ram Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *CVPR*, pages 685–692. IEEE, 2010. 55

[KP97] Daphne Koller and Avi Pfeffer. *Representation and solutions for game theoretic problems. Artificial Intelligence*. 1997. 14

[LHN09] Yuan Li, Chang Huang, and R. Nevatia. Learning to associate: HybridBoosted multi-target tracker for crowded scene. pages 2953–2960, June 2009. 55

[LHT10] Min Li, Wei Chen 0012, Kaiqi Huang, and Tieniu Tan. Visual tracking via incremental self-tuning particle filtering on the affine group. In *CVPR*, pages 1315–1322. IEEE, 2010. 25

[MW10] Michael J. Metternich and Marcel Worring. Semi-interactive tracing of persons in real-life surveillance data. In *Proceedings of the 2nd ACM workshop on Multimedia in forensics, security and intelligence*, pages 43–48, New York, NY, USA, 2010. ACM. 25

[NR99] Nisan and Ronen. *Algorithmic Mechanism Design*. 1999. 14

[Pel97] Marcello Pelillo. The dynamics of nonlinear relaxation labeling processes, 1997. 16, 22

[PP09] Fatih Porikli and Pan Pan. Regressed importance sampling on manifolds for efficient object tracking. In *AVSS*, pages 406–411. IEEE Computer Society, 2009. 25

[PTM06] Fatih Porikli, Oncel Tuzel, and Peter Meer. Covariance tracking using model update based on lie algebra. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 1:728–735, 2006. 37

[thi] 40

[WN07] Bo Wu and Ram Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *Int. J. Comput. Vision*, 75(2):247–266, November 2007. 55

[XAL] Junliang Xing, Haizhou Ai, and Shihong Lao. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 0. 55

[ZLN08] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. 55