A Rank Constrained Continuous Formulation of Multi-frame Multi-target Tracking Problem

Khurram Shafique, Mun Wai Lee, and Niels Haering
Center for Video Understanding Excellence
ObjectVideo, Reston, VA 20191
kshafique, mlee, nhaering@objectvideo.com

Abstract

This paper presents a multi-frame data association algorithm for tracking multiple targets in video sequences. Multi-frame data association involves finding the most probable correspondences between target tracks and measurements (collected over multiple time instances) as well as handling the common tracking problems such as, track initiations and terminations, occlusions, and noisy detections. The problem is known to be NP-Hard for more than two frames. A rank constrained continuous formulation of the problem is presented that can be efficiently solved using nonlinear optimization methods. It is shown that the global and local extrema of the continuous problem respectively coincide with the maximum and the maximal solutions of the discrete counterpart. A scanning window based tracking algorithm is developed using the formulation that performs well under noisy conditions with frequent occlusions and multiple track initiations and terminations. The above claims are supported by experiments and quantitative evaluations using both synthetic and real data under different operating conditions.

1. Introduction

Tracking large number of targets in a video is a challenging task due to high ambiguity in data association caused by frequent occlusions among targets, arrival and departure of targets from the scene, and the presence of detection errors and noise. The task is further complicated when there are not many distinguishing features among the targets (either because of sensor limitations or because the targets are themselves alike) or if the tracking is performed solely on the spatial locations of the targets. These scenarios commonly occur in applications like surveillance, optical flow, feature tracking, and structure from motion. The task of a tracker is to uniquely identify each target in the scene and to specify its position in every frame from its arrival to departure from the scene. That is, at each time step, a tracker must be able to estimate target states and perform data association between measurements and tracks.

Formally, let $\mathbf{Z}(i) = \{z_1(i), z_2(i), \ldots, z_{k_i}(i)\}$ be the set of target measurements at time instant t_i, $1 \leq i \leq T$. A track is defined as a non-empty set of points $\tau = \{z_a(i_1), z_b(i_2), \ldots, z_m(i_k)\}$ such that for all i, $|\tau \cap \mathbf{Z}(i)| \leq 1$. Data association is the problem of finding a partition, $\omega = \tau_1 \cup \tau_2 \cup \ldots \tau_m$ of $\bigcup_{i=1}^{T} \mathbf{Z}(i)$, such that each track is either a set of all the measurements of a single target (element integrity principle) or it is a set of clutter measurements.

In this paper, we present a multi-frame scanning window based algorithm to solve the above data association problem. The proposed algorithm observes subsequent measurements from more than one scan (frame) before assigning a label to earlier measurements. The importance of deferred inference provided by the multi-frame algorithm to resolve ambiguities is well documented in both psychology and computer vision literature [10, 8, 23]. The idea of delayed inference was introduced in computer vision as early as 1976 by David Marr [10] who introduced the principle of least commitment. Multi-frame algorithms for data association are not new either and date back to Reid's multiple hypothesis tracker[19]. The comparison of our work and these methods is provided in the next section. The major contribution of this paper is the presentation of a continuous formulation of the multi-frame data association problem (DAP). The formulation has some interesting properties, for example, the global solutions of both the continuous and discrete problems coincide. In addition, the local extrema of the continuous problem and the maximal solutions of the discrete counterpart are also tightly related. This is practically advantageous because it allows to exploit the full range of well established continuous and efficient optimization techniques to solve a difficult discrete problem. We show that the resulting tracking algorithm performs well under noisy conditions with frequent occlusions and multiple track initiations and terminations.

The organization of the paper is as follows. In Section 2,

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

we present a survey of the related work. In Section 3, we provide a formulation of the problem and detail the proposed solution in Section 4. In Section 5, we present quantitative and qualitative results to validate our claims. The paper is concluded in Section 6.

2. Related Work

Earlier work in motion correspondence focused on the 2-frame problem, where the track labels for the measurements of each scan are fixed right after the scan. Ullman [24] used a linear programming approach to the problem. Salari and Sethi [20] used an iterative greedy exchange algorithm to optimize an objective function based on nearest neighbor and smoothness constraints. Rangarajan and Shah [18] proposed a non-iterative greedy algorithm that used 3 frames to find correspondences using proximal uniformity constraint. Veenman *et al*. [25] proposed a Hungarian search algorithm and showed that their GOA algorithm generalizes many of the previous 2-frame algorithms.

One of the best known examples of true look-ahead algorithms is the multiple hypothesis tracker (MHT) [19]. MHT maintains a set of hypotheses for each track assignment until a clear winner (the hypothesis with the highest posterior) can be determined. However, the algorithm suffers from combinatorial explosion as the number of hypotheses to be maintained grows exponential over time. Efficient implementations and approximations of MHT have been suggested (see [15] for a survey). A large class of these algorithms attempts to solve the multidimensional (S-D) assignment problem in a sliding window of fixed temporal scope [7, 11, 13, 17, 22]. This problem is known to be NP-Hard (i.e., the existence of an efficient algorithm is highly unlikely) when $S > 2$, i.e., if the window contains more than two frames. Further approximations, such as, Lagrangian and linear relaxations are employed for practical applications. Poore and Yan[17] used Lagrangian relaxation to iteratively reduce the S-D assignment problem to the 2D assignment problem. Murphey *et al*. [13] used a greedy randomized adaptive search procedure (GRASP) to solve this problem. Morefield [11] modelled the problem as an integer programming problem and solved its linear programming (LP) relaxation. Linear relaxations of integer programming problems for DAP were more recently used in [9] and [22], where a greedy rounding procedure was used to obtain an integer solution from LP solution. One disadvantage of using relaxations of a general integer program is that the solution does not utilize the combinatorial structure and the associated polyhedra of the problem in hand, which has been shown to be a valuable tool for both the theoretical insight into the problem structure as well as its approximability.

Two alternative approaches to the problem were recently proposed in [14] and [21]. In [14], a Markov chain Monte Carlo method was used for random sampling of the solution space. In [21], the problem was modelled as that of finding a maximum weighted path cover of a directed graph and a polynomial time optimal algorithm was presented. However, the limiting factor of the model was that the weights of each hypothesized link was assumed to be independent of past history. This is generally not true especially when the only useful cue for correspondence is the motion of the target. A greedy heuristic was used to relax this condition.

The work presented in this paper is more closely related to the multi-dimensional assignment based approaches, *e.g.*, [11, 22] for the data association problem (DAP). However, instead of relaxing a general integer programming problem, we attempt to solve an equivalent combinatorial problem of finding the maximum weighted stable set in a graph. Modelling the problem as a well known graph theoretical problem allows us to properly identify its solution space. A rank-constrained continuous relaxation of a semi-definite program of the maximum stable set problem is then applied to efficiently solve the problem. We show empirically that the quality of solution obtained using the proposed relaxation is better than that of the LP relaxation.

3. Problem Formulation

Recall from Section 1 that $\mathbf{Z}(i) = \{z_1(i), z_2(i), \ldots, z_{k_i}(i)\}$ is the set of target measurements at time instant t_i, $1 \leq i \leq T$. For an S-D assignment problem (sliding window of size S), we construct a hyper-graph $G = (V, E)$, such that $\{V_1, V_2, \ldots, V_S\}$ partitions V. Each set $V_i \subset V$ consists of $k_i + 1$ vertices $v_0(i), v_1(i), \ldots, v_{k_i}(i)$ such that $\forall a > 0$, the vertex $v_a(i) \in V_i$ corresponds to a true measurement $z_a(i) \in \mathbf{Z}(i)$. The vertex $v_0(i)$, $\forall i$, represents a dummy measurement to deal with occlusions, missed detections, false alarms, track initiations, and track terminations. Furthermore, $E \subseteq V_1 \times V_2 \times \ldots V_S - \{(v_0(1), v_0(2), \ldots, v_0(S))\}$, where each $e \in E$ is an S-dimensional vector and represents a track hypothesis τ. From here onwards, we will be using τ to represent edges of graph G for the sake of simplicity. Similarly, vertices of G, $v_a(i)$, and corresponding measurements, $z_a(i)$ will be used interchangeably. The edge weights $w(\tau)$ are defined as the log likelihood $L(\tau)$ of the track hypothesis. A sample hyper-graph constructed this way is shown in Figure 1 (Please note that except for the first frame where the tracks are initiated, the vertices of the first set V_1 correspond to the (established) tracks. However, for simplicity, the same notation will be used in the rest of the paper). The three edges shown in Figure 1 represent three distinct types of tracks: The edge τ_1, represents a track where the target is not observed in the second frame, τ_2 shows a continuation of an existing track, whereas τ_3 is a hypothesis of new target arrival in the frame S. A hypothesis with only one true measurement (*e.g.*, τ_3) may also indicate false alarms.

From Section 1, we know that a feasible solution of the

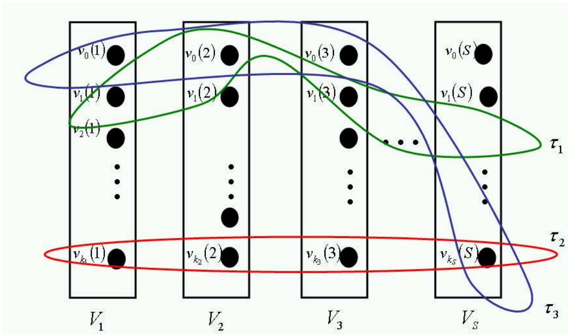


Figure 1. A sample hyper-graph G for DAP. Each edge represents a distinct track hypothesis, for example, τ_1 is a hypothesis that target corresponding to the measurement (track) $z_1(1)$ is not detected in the second scan whereas τ_2 is a hypothesis that the measurement $z_{k_S}(S)$ in the scan S initiates a new track.

multi-frame DAP is a set of tracks $\omega = \{\tau_1, \tau_2, \ldots, \tau_m\}$ that partitions the set of measurements $\mathbf{Z}$, i.e., $\bigcup_{i=1}^{m} \tau_i = \mathbf{Z}$ and for each $i \neq j$, $\tau_i \cap \tau_j = \emptyset$. In terms of the constructed hyper-graph, a feasible solution is a set of edges $\omega = \{\tau_1, \tau_2, \ldots, \tau_l\}$ such that for each i, $1 \leq i \leq S$ and j, $1 \leq j \leq k_i$, there exists a unique edge $\tau \in \omega$ for which $\tau(j) = z_j(i)$, i.e., each true measurement is contained in exactly one track. An optimal solution of the problem is a feasible solution ω^* such that for all feasible sets ω, $\sum_{\tau\in\omega^*} w(\tau) \geq \sum_{\tau\in\omega} w(\tau)$, i.e., the objective function that we seek to maximize is the sum of log likelihoods of all the tracks in a feasible solution. The problem of finding ω^* can easily be formulated as a 0-1 integer programming problem [22, 11] by assigning a binary variable to each track τ and defining constraints to ensure element integrity principle.

Let us construct another graph $G' = (V', E')$, where each vertex of the graph corresponds to a track hypothesis (or an edge of graph G) and there is an edge between two vertices if and only if the corresponding tracks have a common true measurement. Formally, $V' = E = \{\tau_1, \tau_2, \ldots, \tau_{|E|}\}$ and $E' = \{(\tau_a, \tau_b) | \exists S \geq i \geq 1, j \geq 1, \text{ for which}, \tau_a(i) = \tau_b(i) = z_j(i)\}$. Also, we define the weight of each vertex in G' to be equal to the weight of the corresponding track. It is easy to see that the feasible solutions ω of DAP correspond to the maximal stable sets of graph G'. (A stable set is a set of vertices in a graph, no two of which have a common edge. A maximum weighted stable set is a stable set for which the sum of the weights of its vertices is largest among all stable sets of the graph.) Now, let $A = \{\tau_1, \tau_2, \ldots, \tau_q\}$ be a maximal stable set of graph G' such that $\{\tau_1, \tau_2, \ldots, \tau_q\}$ is not a feasible solution of DAP. By definition of stable sets and construction of graph G, we know that the element integrity principle is not violated. Hence, there must exist a true measurement $z_j(i)$ that is not contained in any track in A. By construction of graph G', this implies that the track $\tau = (z_0(1), z_0(2), \ldots, z_0(i-1), z_j(i), z_0(i+1), \ldots, z_0(S))$ does not have any edge in common with any track $\tau_p \in A$. But then $A \cup \tau$ is a stable set, which contradicts that A is a maximal stable set. Thus, DAP is equivalent to the problem of finding maximum weighted stable set in graph G'. An immediate consequence of this equivalence is that no polynomial algorithm can approximate the optimal solution of DAP within a factor of n^ϵ, for fixed but arbitrary $\epsilon > 0$ (since the above is true for maximum stable set problem [1]). It also allows us to use a wide array of techniques and heuristics developed for maximum stable set problem (or maximum clique problem, since both are equivalent), such as, tabu search, simulated annealing, replicator dynamics, and continuous-based heuristics based on Motzkin-Strauss[12] like formulations of maximum clique (see [3] for a detailed survey of approximate algorithms for maximum clique problem).

4. Continuous Formulation

The use of continuous formulations for discrete problems has recently gained significant interest in the literature [3]. These formulations allow the use of efficient and well known continuous optimization techniques to the solution of hard discrete problems. The formulation presented here is an extension of the continuous formulation of maximum stable set, due to [6], to the weighted graphs. The maximum weighted stable set problem for a graph $G = (V, E)$ is to find a maximal weighted subset $A \subseteq V$ of vertices with weights $w \in \mathbb{R}_+^{|V|}$, such that no two vertices are connected by an edge in E, equivalently, we may write the problem as $\max\{w(A) | A \subseteq V,\ ij \notin E\ \forall i, j \in A\}$.

A well known semi-definite relaxation of this problem (due to Lovász and Schrijver) is given as:

$$\begin{array}{ll} \text{Maximize} & \mathbf{W} \bullet \mathbf{Y} \\ \text{subject to} & y_{ij} = 0, \forall ij \in E, \\ & \text{trace}(\mathbf{Y}) = 0, \\ & \mathbf{Y} \succeq 0 \end{array} \tag{1}$$

where $\mathbf{W} = \sqrt{w}\sqrt{w}^T$, $\mathbf{Y}$ is a symmetric matrix of size $|V| \times |V|$, $\mathbf{W} \bullet \mathbf{Y}$ is the matrix inner product, i.e., $\mathbf{W} \bullet \mathbf{Y} = \text{trace}(\mathbf{W}^T\mathbf{Y})$, and the constraint $\mathbf{Y} \succeq 0$ requires that Y be positive semi-definite.

Many approximation algorithms for maximum stable set are based on the above semi-definite program (SDP) as it (in the un-weighted graphs) provides an upper bound (Lovász theta number $\vartheta(G)$) on the stability number of the graph. Most of these algorithms require explicit solution of some semi-definite program. However, the existing methods for solving SDP, such as, interior point methods and spectral bundle methods for SDP are not suitable for large scale problems that are common in DAP instances.

The formulation presented in this paper is obtained by restricting the above SDP program to the matrices ($\mathbf{Y}$) of rank at most 1 or 2. For rank 1, we can write $\mathbf{Y} = yy^T$ for some $y \in \mathbb{R}^n$. This results in the following non-linear program (call it NLP1):

$$\begin{array}{ll} \text{Maximize} & (\sqrt{w}^T y)^2 \\ \text{subject to} & y \in \mathbb{R}^n,\ \|y\|^2 = 1, \\ & y_i y_j = 0, \forall ij \in E \end{array} \tag{2}$$

For any $y \in \mathbb{R}^n$, let S_y be a set of vertices $i \in V$ for which $y_i \neq 0$, i.e., $S_y = \{i \in V | y_i \neq 0\}$. In other words, y is a continuous incidence vector of S, or y induces set S_y in the graph G. The following theorem describes a relationship between the local extrema y' of NLP1 and the sets induced by y' in G.

Theorem 1 *For $y' \in \mathbb{R}^n$, such that $\|y'\|^2 = 1$, y' is a local maximizer of NLP1 if and only if $S_{y'}$ is a maximal stable set of G.*

Hence there is a tight correspondence between the local solutions of NLP1 and the local (maximal) solutions of maximum stable set problem. As a matter of fact, the same is true for the global maxima of both problems, i.e., it can be shown that the global maxima of NLP1 correspond to the maximum weighted stable sets of graph G. This results is formally stated as:

Theorem 2 *The optimal value of NLP1 is equal to the weight of a maximum stable set of graph G. Also, y^* is a global maximizer of NLP1 if and only if S_{y^*} is a maximum weighted stable set of G.*

The proof of both theorems follow similar line of reasoning as that of the un-weighted case [6] and though simple, are rather long and out of scope of this paper. Similar results also hold when matrix ($\mathbf{Y}$) is restricted to be of rank at most 2, i.e., $\mathbf{Y} = xx^T + yy^T$ for some $x, y \in \mathbb{R}^n$. Due to this tight correspondence between the discrete and continuous problems, we can find a stable set by first solving NLP1 with an appropriate continuous optimization technique and then extracting the set induced by the solution found. In other words NLP1 allows us to perform combinatorial optimization using continuous optimization techniques.

We used an augmented Lagrangian algorithm based heuristic [6, 27] to solve the above non-linear program. Augmented Lagrangian relaxation is a standard technique that allows to place the difficult constraints of the program in the objective function itself. In this case, the difficult constraints are the edge constraints, i.e., $y_i y_j = 0$, $\forall ij \in E$. Thus, the augmented Lagrangian relaxation of the NLP1 is given as follows (call it NLP2):

$$\begin{array}{ll} \text{Maximize} & (\sqrt{w}^T y)^2 + \left(\lambda - \frac{\sigma}{2} c(y)\right) c(y) \\ \text{subject to} & y \in \mathbb{R}^n,\ \|y\|^2 = 1, \end{array} \tag{3}$$

where, $\lambda = (\lambda_{ij})_{ij \in E}$, $\sigma > 0$ is a fixed penalty parameter, and $c(y) = (y_i y_j)_{ij \in E}$, are the edge constraints.

The augmented Lagrangian algorithm iteratively performs successive minimization of NLP2 with respect to y while keeping the other two parameters (i.e., λ and σ) fixed, which are updated between iterations. For minimization of NLP2 w.r.t. y in each iteration, we use a strong Wolfe-Powell line search and a gradient based limited memory BFGS technique for generating the search directions [5].

4.1. Motion Model

In this section, we describe how the likelihood value for each track τ is computed. We use a linear system model for the target dynamics. Let $x_j(i)$ be the state of the target j at time t_i. The state transition model is described by: $x_j(i+1) = F_j x_j(i) + w_j(i)$ whereas the measurement model is given as: $z_j(i) = H_j x_j(i) + v_j(i)$. Here, $w_j(i)$ and $v_j(i)$ are zero mean white Gaussian noise variables with covariances $Q_j(i)$ and $R_j(i)$ respectively. Each observation of the state of the target is measured with a detection probability p_d. In addition to track observations, false alarms and new targets are observed from Poisson arrival distributions with parameters λ_f and λ_t respectively. The spatial locations of false alarms and track initiations are assumed to be uniformly distributed over the sensor's field of view with probability densities d_f and d_t respectively.

For each track hypothesis τ, we estimate the state of the corresponding target $\bar{x}(i)$ (in τ) and its covariance by applying the Kalman Filter at each time step i. The residual covariance at each time step is given as $U(i) = H(i)P(i)H(i)^T + R$. The log likelihood, $L(\tau)$ of the track $\tau = (z_{j_1}(1), z_{j_2}(2), \ldots, z_{j_S}(S))$ can then be written as [16]:

$$L(\tau) = \sum_{i=1}^{S} \ln g(z_{j_i}(i), \tau)$$

where,

$$g(z_{j_i}(i), \tau) = \begin{cases} 1 - p_d & \text{if } j_i = 0 \\ \frac{\lambda_t d_t}{\lambda_f d_f} & \text{if } z_{j_i}(i) \text{ initiates the track} \\ \frac{p_d p(z_{j_i}(i)|\tau)}{\lambda_f d_f} & \text{otherwise} \end{cases}$$

and $p(z_{j_i}(i)|\tau) = \frac{e^{0.5[z_{j_i}(i) - \bar{x}(i)]^T U(i)^{-1} [z_{j_i}(i) - \bar{x}(i)]}}{\sqrt{(2\pi)^n |U(i)|}}$

The above model is used to independently assign a weight (log likelihood) to each track in the solution space. Since, the number of tracks in the solution space $E \subseteq V_1 \times V_2 \times \ldots V_S - \{(v_0(1), v_0(2), \ldots, v_0(S))\}$ is very large, it is therefore imperative to efficiently prune the less likely tracks. Next we present the strategies to reduce the solution space.

4.2. Solution Space Reduction & Problem Decomposition

To prune the unlikely tracks, filtering techniques commonly known as "gating" are generally utilized. A survey of gating techniques can be found in [2]. We use three different gating tests to reduce the size of our problem.

The first gating test only allows a new hypothesis to be created if the measurement fits the target's predicted state with a certain degree of confidence. The gating test is based on the Mahalanobis distance of the measurement and the target's predicted state and is given as $[z_{j_i}(i) - \bar{x}(i)]^T U(i)^{-1} [z_{j_i}(i) - \bar{x}(i)] \leq \beta$, where the value of β is decided based on the choice of validation region.

The second gating test uses a parameter for maximum absence of a target measurement. If a target is not observed for more than the duration of this parameter, it is considered to have left the scene and the track for that target is no longer used in the data association.

The third gating test prunes all the track hypothesis that have negative weights. It is easy to show that the tracks with negative weights cannot be part of a maximal solution. Suppose otherwise and let τ be such a track in a maximal solution A, then a better feasible solution can be obtained by replacing τ in A with the set of tracks each of which has exactly one true measurement corresponding to the true measurements in τ.

In addition to pruning the less likely tracks from the solution space, we also decompose the solution space into independent components, each of which can be independently solved without affecting the quality of the overall solution. Note that, if a graph G' is not connected then a maximum weighted stable set of graph G' is the union of maximum weighted stable sets of each connected component of G'. Hence, the solution space of the DAP can be decomposed by finding the connected components of the graph. This procedure generally results in multiple problems of much reduced size (than the original) that can be independently solved and thus significantly improve the efficiency of the algorithm.

5. Results

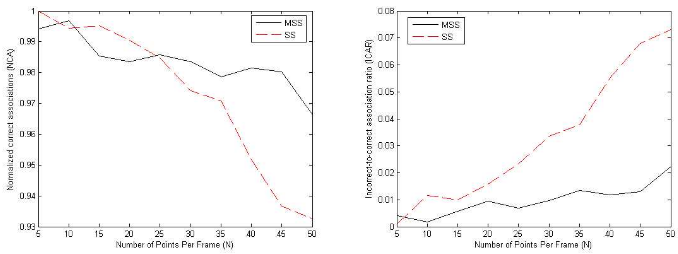
In this section, we present quantitative and qualitative evaluation of the performance of our algorithm using both synthetic and real sequences.

5.1. Quantitative Evaluation

We used synthetic data to evaluate the performance of the proposed algorithm under different operating conditions, such as, traffic densities, detection probabilities and false alarms. The synthetic target sequences were generated by a modified *Point Set Motion Generator (PSMG)* [26]. PSMG provides controls over the size of image space, number of points, number of frames, mean and variance of initial velocity, mean and variance of the change in velocity, probability of occlusion, and probability of false alarms. Initially, a given number, N, of points are initialized in the given image space with uniform probability. The random points move independent of each other with uniformly distributed initial motion directions and normally distributed speeds. In every frame, Gaussian perturbations are introduced in both motion directions and speeds. The state of each random point (i.e., whether the point is detected or not) is independently determined with a uniform probability of detection, P_d. The points may remain undetected for a given maximum duration after which they are forced to be observed at least once. The points leave the scene when they cross the scene boundaries. In each frame, new points are introduced in the scene with a Poisson distribution, λ_N, however, the maximum number of points in a scene at a given time is bounded. Finally, false alarms are introduced in each frame with uniform spatial distribution, where the number of false alarms has a Poisson distribution, λ_F.

The results of the proposed tracker (denoted by MSS, for maximum stable set) are compared with the multi-frame tracker (SS) proposed by Shafique and Shah [21] that has been shown to perform well under a variety of scenarios. To measure the effectiveness of each tracker and to compare the results, we use two metrics proposed in [14], i) normalized correct associations (NCA) and ii) incorrect-to-correct association ratio (ICAR). Let ω^* be the true solution (ground-truth) of the DAP. For any ω in the solution space, the set of all associations (links) in ω is defined as $\mathrm{SA}(\omega) = \bigcup_{\tau \in \omega} \{(\tau(i), \tau(j)) | 1 \leq i < j \leq S \wedge \tau(i) \neq z_0(i) \wedge \tau(j) \neq z_0(j) \wedge \forall k, i < k < j,\ \tau(k) = z_0(k)\}$. The set of correct associations $\mathrm{CA}(\omega)$ in ω with respect to ω^* is defined as $\mathrm{CA}(\omega) = \mathrm{SA}(\omega) \cap \mathrm{SA}(\omega^*)$. The normalized correct associations (NCA) are defined as the ratio between the number of correct associations and the number of true associations, i.e., $\mathrm{NCA}(\omega) = \frac{|\mathrm{CA}(\omega)|}{|\mathrm{SA}(\omega^*)|}$. The incorrect-to-correct association ratio (ICAR) is defined as $\mathrm{ICAR}(\omega) = \frac{|\mathrm{SA}(\omega)| - |\mathrm{CA}(\omega)|}{|\mathrm{CA}(\omega)|}$. For each scenario, we report the average of both metrics computed on the output of 20 random sequences generated by using the same parameters. The parameters of both the algorithms were kept constant in all scenarios. Particularly, the sliding window size of 5 was used in all the experiments.

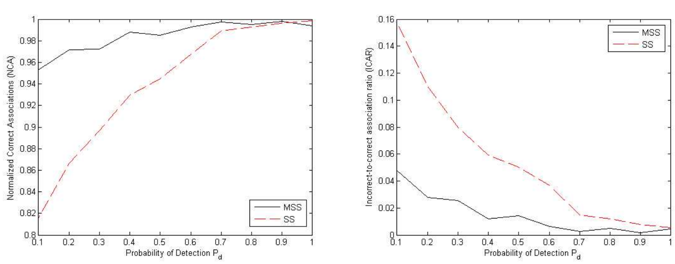
In our first experiment, we evaluated the performance of the proposed algorithm with respect to the track density in the scene. The number of initial tracks, N, was varied from 5 to 50 in a fixed image space of size 300×300. All the other parameters of the point set motion generator were kept constant. Specifically, we chose the length of each sequence to be 50 frames, probability of detection, $P_d = 0.9$, the expected number of false alarms per frame, $\lambda_F = 1$, and the expected number of arrivals, $\lambda_N = 1$. However, the maximum number of active targets (not including false alarms and terminated tracks) was bounded by N. The maximum duration of continued absence of a point was chosen to be 4. The average values of the performance metrics (over 20 random sequences) are shown in Figure 2.



(a) (b)

Figure 2. Performance of the proposed tracker (MSS) and the SS tracker with respect to the density of points in the scene. The plots for MSS and SS are shown with solid black and dashed red plots respectively. (a) Normalized correct associations (NCA), (b) Incorrect-to-correct association ratio (ICAR).

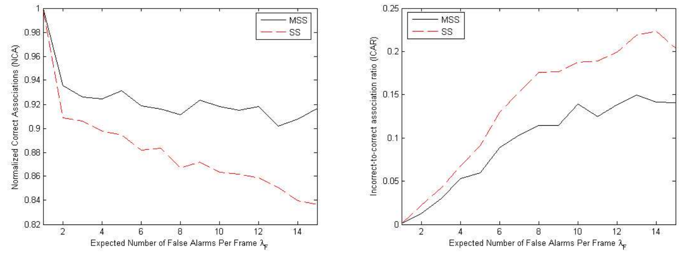
In the next two experiments, we evaluated the robustness of the algorithm with respect to the probability of detection and the presence of false alarms in the scene. The number of initial tracks was fixed at $N = 10$, while the other parameters were kept the same as the previous experiment. First we varied the probability of detection P_d from 1 to 0.1 with a decrement of 0.1 at each step. Again, the performance metrics NCA and ICAR were computed at each step on 20 different sequences. The average values of both metrics are plotted in Figure 3. Next, we fixed P_d back to 0.9 and varied the expected number of false alarms per frame, λ_F, from 1 to 15. The results of this experiment are shown in Figure 4.



(a) (b)

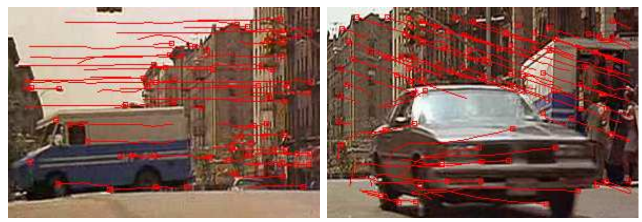
Figure 3. Tracking performance with respect to detection probability P_d: (a) NCA, (b) ICAR.

The results in plots of Figures 2, 3, and 4 show that the proposed algorithm is quite robust to the target densities, occlusions, and false alarms. The algorithm achieved better than 90% correct links in all the cases, whereas the ratio of



(a) (b)

Figure 4. Tracking performance with respect to false alarm density (a) NCA, (b) ICAR.



(a) (b)

Figure 5. Tracking results on feature detection output.

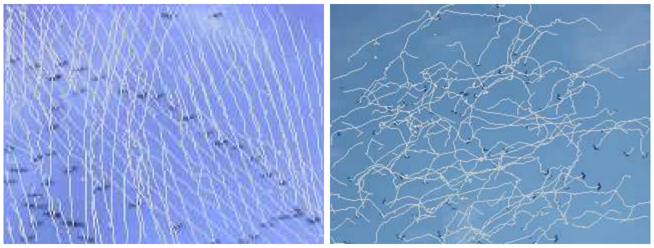
incorrect to correct associations was also significantly low for most cases. In addition, the proposed algorithm clearly performed better than the greedy algorithm of [21] on all performance metrics especially under more difficult scenarios (high number of false alarms or low probability of detection). With gating and problem decomposition, the proposed algorithm performs in real-time (on average 7 frames per second) on most moderate sized problems (around 25 points per frame).

5.2. Qualitative Results

Next, we show the results of our algorithm on real data. The detection results on different scenarios were used as the input to the algorithm. In tradition with the evaluation of data association algorithms, we restricted the input to our algorithm to the spatial measurements of the targets in each frame. In other words, no shape or appearance information was used in the generation of the results of this section.

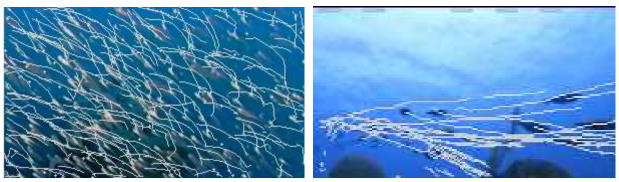
We first tried our algorithm on feature point tracking problem, where feature points were first detected using Harris feature detector on image sequences captured from a moving camera. The spatial locations of the detected features were used for tracking. The generated tracks on two such sequences are shown in Figure 5.

Next we tested the proposed algorithm on real sequences with high traffic density and frequent occlusions. Moving objects were detected by background subtraction and their centroids were used as the feature points for tracking. In Figure 6, we show the tracks of individual birds in flocks. In first sequence, the birds are exhibiting group



(a) (b)

Figure 6. Tracks generated for birds in motion



(a) (b)

Figure 7. Tracks generated for schools of fish

motion whereas in the second sequence, each bird is exhibiting unique independent motion. Visual inspection of the tracks show that the tracker successfully tracked individual birds in both instances. We also tested the algorithm to track schools of fish and achieved similar results. The tracks for these experiments are shown in Figure 7.

We also tested our algorithm on human tracking in a sports scenario (hockey players). Evenly sampled frames from the sequence with tracking results are shown in Figure 8. All these results show good tracking performance on challenging data (especially since no appearance information was used for correspondence) and validate the quantitative analysis of the previous subsection.

6. Discussion and Conclusion

In this section, we discuss objections and questions raised by the anonymous reviewers.

Comment 1: *Target merging and splitting is a significant problem in real surveillance applications. It is not clear how the proposed framework can accommodate such one-to-many and many-to-one scenarios.*

Response: The framework deals with many-to-one scenarios (occlusions) as well as missed detections by employing the dummy measurements (See Figure 1) that indicate that the target is either missing or occluded in the given frame. Traditional data association methods assume point models for targets and do not deal with one-to-many scenarios (single target generating multiple measurements). Only recently, attempts have been made to tackle these scenarios in data association framework [4, 28]. While we have not considered target splitting in this paper, theoretically it can be handled by allowing hyper-edges to contain more than one measurement in each frame. This is equivalent of saying that the measurements belong to the same target. However, doing so may significantly increase the problem size and aggressive gating or other optimizations may be required for real-time performance. One way to reduce the complexity of such a system is to use tracklets from a high precision tracker as measurements (vertices of hyper-graph) and use the proposed algorithm for linking the tracklets.

Comment 2: *In tracking real objects, shape and appearance information can be very powerful to reduce the complexity. How (or weather) these characteristics could be incorporated into the system?*

Response: The framework does not impose any restriction on the features used, motion model, or objective function for edge-weight computation. One way to incorporate appearance features is by assuming independence with the spatial features and modifying likelihood computation accordingly.

Comment 3: *How the sliding window is handled?*

Response: The sliding window is overlapping and is incremented one step after each frame. The tracks are only committed once the measurements fall out of the window.

Comment 4: *The gating parameters can be selected to make the algorithm arbitrarily fast at the expense of accuracy. How are the gating parameters selected?*

Response: The gating parameter β is based on chi-square distribution and was chosen to be have the 99% of the validation region, i.e., $\beta = 9.2$. The maximum absence parameter was assumed to be 30 frames in all of our experiments.

Comment 5: *How is track initiation handled?*

Response: For each measurement, a track initiation hypothesis is modelled by a hyper-edge that contains only dummy measurements in the previous frames. The velocity for such measurement is assumed to be zero and the motion model reduces to the nearest neighbor model.

Conclusion: In conclusion, we have presented a continuous formulation of the multi-frame data association problem that is based on a semi-definite program of the maximum weighted stable set problem. The formulation readily allows the use of many continuous optimization techniques as well as competing algorithms and heuristics, such as tabu search, simulated annealing, and replicator dynamics to the data association problem. A sliding window based multi-frame tracker was developed using the formulation and was shown to perform well on both real and synthetic data.

Acknowledgment: This material is based upon work supported by, or in part by, the U.S. Army Research Laboratory and the U.S. Army Research Office under contract number W911NF-07-C-0065.

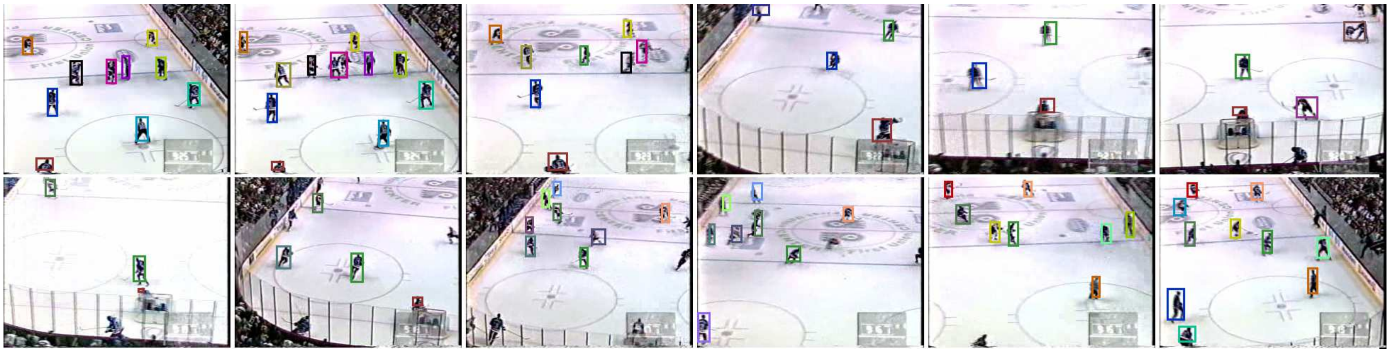


Figure 8. People tracking performance in a sports scenario.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. "Proof verification and hardness of approximation problems". In *Proceedings 33rd Annual Symposium on Foundations of Computer Science*, pages 14–23, 1992.

[2] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, MA.

[3] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo. The maximum clique problem. In *Handbook of Combinatorial Optimization, Supplement Volume A*.

[4] B. Bose, X. Wang, and E. Grimson. Multi-class object tracking algorithm that handles fragmentation and grouping. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007.

[5] S. Burer and R. Monteiro. "A projected gradient algorithm for solving the maxcut sdp relaxation". *Optimization Methods and Software*, 15:175–200, 2001.

[6] S. Burer, R. Monteiro, and Y. Zhang. "Maximum stable set formulations and heuristics based on continuous optimization". *Mathematical Programming*, 94, 2002.

[7] S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom. "A generalized S-D assignment algorithm for multisensor-multitarget state estimation". *IEEE Transactions on Aerospace and Electronic Systems*, 33(2):523–538, 1997.

[8] J. J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflen, Boston, MA, 1979.

[9] H. Jiang, S. Fels, and J. Little. A linear programming approach for multiple object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[10] D. Marr. Early Processing of Visual Information. *Royal Soc. of London Phil. Trans. Ser. B*, 275, Oct. 1976.

[11] L. Morefield. "Application of 0-1 integer programming to multitarget tracking problems". *IEEE Transactions on Automatic Control*, 22:302–312, 1977.

[12] T. S. Motzkin and E. G. Strauss. "Maxima for graphs and a new proof of a theorem of turin". *Canadian Journal of Mathematics*, 17:533–540, 1965.

[13] R. A. Murphey, P. M. Pardalos, and L. Pitsoulis. "A grasp for the multitarget multisensor tracking problem". *DIMACS Ser. in Discr. Math. and Theoretical Comp. Sci.*, 40, 1998.

[14] S. Oh, S. Russell, and S. Sastry. Markov chain monte carlo data association for general multiple-target tracking problems. In *IEEE Conf. on Decision and Contr.*, 2004.

[15] K. Pattipati, R. Popp, and T. Kirubarajan. Survey of assignment techniques for multitarget tracking. In Y. Bar-Shalom and W. Blair, editors, *Multitarget-Multisensor Tracking: Advanced Applications, vol. III*. Artech House, 2000.

[16] A. B. Poore. Multidimensional assignments and multitarget tracking. In *Partitioning Data Sets; DIMACS Workshop*, pages 169–196. American Mathematical Society, 1995.

[17] A. B. Poore and X. Yan. Data association in multi-frame processing. In *Proceedings of the Second International Conference on Information Fusion*, pages 1037–1044, 1999.

[18] K. Rangarajan and M. Shah. "Establishing motion correspondence". *CVGIP*, 54(1):56–73, july 1991.

[19] D. B. Reid. "An algorithm for tracking multiple targets". *IEEE Trans. on Auto. Contl.*, 24(6), Dec 1979.

[20] V. Salari and I. K. Sethi. "Feature point correspondence in the presence of occlusion". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(1), Jan 1990.

[21] K. Shafique and M. Shah. "A non-iterative greedy algorithm for multi-frame point correspondence". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1), January 2005.

[22] P. P. A. Storms and F. C. R. Spieksma. "An lp-based algorithm for the data association problem in multitarget tracking". *Computers and Operations Research*, 30(7):1067–1085, June 2003.

[23] J. T. Todd. "Visual information about rigid and nonrigid motion: A geometric analysis". *Journal of Experimental Psychology: Human Perception and Performance*, 8, 1982.

[24] S. Ullman. *The Interpretation of Visual Motion*. M.I.T. Press, Cambridge, MA, 1979.

[25] C. J. Veenman, M. J. T. Reinders, and E. Backer. "Resolving motion correspondence for densely moving points". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1):54–72, January 2001.

[26] J. Verestóy and D. Chetverikov. "Experimental comparative evaluation of feature point tracking algorithms". In *Proceedings Workshop on Evaluation and Validation of Computer Vision Algorithms*, pages 183–194. Kluwer series in Computational Imaging and Vision, 2000.

[27] N. J. Wright. *Numerical Optimization*. Springer-Verlag.

[28] Q. Yu, G. Medioni, and I. Cohen. Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007.