Establishing Motion Correspondence using Extended Temporal Scope

C.J. Veenman[*], M.J.T. Reinders, and E. Backer[†]

Abstract This paper addresses the motion correspondence problem: the problem of finding corresponding point measurements in an image sequence solely based on positional information. The motion correspondence problem is most difficult when the target points are densely moving. It becomes even harder when the point detection scheme is imperfect or when points are temporarily occluded. Available motion constraints should be exploited in order to rule out physically impossible assignments of measurements to point tracks. The performance can be further increased by deferring the correspondence decisions, that is, by examining whether the consequences of candidate correspondences lead to alternate and better solutions. In this paper, we concentrate on the latter by introducing a scheme that extends the temporal scope over which the correspondences are optimized. The consequent problem we are faced with is a multi-dimensional assignment problem, which is known to be NP-hard. To restrict the consequent increase in computation time, the candidate solutions are suitably ordered and then additional combined motion constraints are imposed. Experiments show the appropriateness of the proposed extension, both with respect to performance as well as computational aspects.

Keywords: Computer vision, feature point tracking, multi-target tracking, motion correspondence, multi-frame optimization, multi-dimensional assignment problem.

1 Introduction

Computer vision deals with the interpretation of image sequences. Because the problem of semantic labeling of an arbitrary scene is far from solved, any information that can help the scene interpretation should be exploited. The known temporal dependencies between frames in a sequence together with known physical properties like inertia and rigidity have been proven to be very helpful. More than that, temporal relations can be crucial in circumstances in which the objects in the recorded scene are difficult to distinguish, either because of poor recordings, poor recording conditions, restricted recording devices/media, or because the objects appear identical anyway. The research fields concerned with these issues are among others object tracking [25], feature or token tracking [3] [13] [27] [38], and optical flow or motion estimation [12] [19]. Applications range from surveillance [20] [30] [36], motion analysis, and structure from motion [31] [32] [35] [37] to (multi-)target tracking [11] [21] [24].

Here, we restrict ourselves to the case that for some reason the objects have indeed an identical appearance, which leaves us with the positional information as sole feature for identification. Therefore, the objects are simply referred to as points in the remainder of this paper. Clearly, without significant visual identification appearance-based methods like optical flow estimation do not apply. The consequent problem that has to be solved is called the *motion correspondence* problem, that is, finding corresponding measurements through an image sequence solely based on the measured positions and

[*] Corresponding author

[†] The authors are with the Department of Mediamatics, Faculty of Information Technology and Systems, Delft University of Technology, P.O.Box 5031, 2600 GA, Delft, The Netherlands. E-mail: {C.J.Veenman, M.J.T.Reinders, E.Backer}@its.tudelft.nl

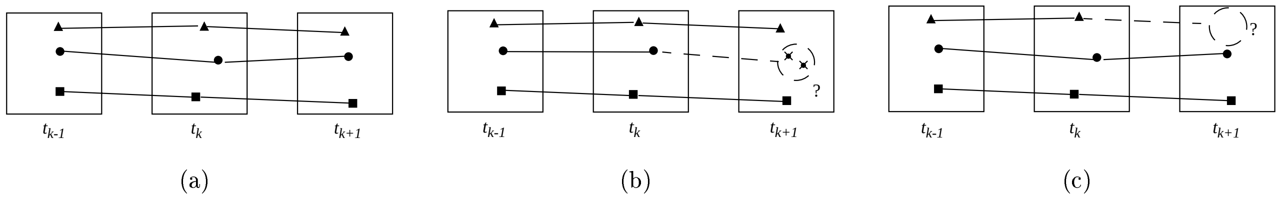


Figure 1: Three moving points are measured at three time instances. The lines represent the point correspondences over time. In (a) all points are measured at every time instance. In (b) there is an extra or spurious measurement at t_{k+1}, and in (c) there is a missing measurement at t_{k+1}.

derived motion characteristics (Fig. 1(a)). Additionally, like among others [3] [9] [24] [27], we adopt a uniqueness constraint which states that a measurement originates from (at most) one point and a point results in (at most) one measurement.

There is a number of conditions under which establishing motion correspondence is especially difficult: *1)* the points move densely together, *2)* the detection is imperfect, i.e. there are spurious (Fig. 1(b)) and missing (Fig. 1(c)) measurements, *3)* points are temporarily occluded, and *4)* the number of points varies. Here, we consider such difficult cases, except we assume that the number of points is fixed. Namely, without additional constraints, coping with both condition *3)* and *4)* gives rise to conflicting requirements for a tracking algorithm, as noted in [33].

Several statistical [4] and non-statistical methods have been developed to establish motion correspondence both in the field of target tracking and computer vision. The two best known statistical approaches are the Joint Probabilistic Data-Association Filter (JPDAF) [11] and the Multiple Hypothesis Tracker (MHT) [24]. The JPDAF matches a fixed number of features in a greedy way and is especially suitable for situations with clutter. It does not necessarily select point measurements as exact feature point locations, but, given the measurements and a number of corresponding probability density functions, it estimates these positions. The MHT attempts to match a variable number of feature points globally, while allowing for missing and false detections. Quite a few attempts have been made to restrain the consequent combinatorial explosion, such as [5], [6], [7], [8], [16], [18]. More recently, the equivalent sliding window algorithms have been developed, which match points using a limited temporal scope. Then, these solve a multidimensional assignment problem, which is again NP-hard, but real-time approximations using Lagrangian relaxation techniques are available [9], [10], [21], [22], [28].

Recently, in [33] we proposed a non-statistical motion framework together with the GOA tracker, which is a greedy matching method that efficiently finds optimal correspondences between two frames given a smoothness of motion criterion. We showed that the GOA tracker outperforms other non-statistical greedy trackers [3] [23] [26] and even the presumed optimal MHT [24] for the tracking of a fixed number of points. In [33], we have also suggested a global matching model that optimizes over the whole sequence, but did not report an algorithm that satisfies the model. However, the more difficult the problem, the more important it becomes to perform a global matching, i.e. to defer correspondence decisions.

In this paper, we propose an algorithm that enables us to defer correspondence decisions by introducing a temporal scope parameter s. With scope $s = 1$ this algorithm equals the previously introduced GOA tracker [33] and when $s = n - 1$ the algorithm performs a global matching, where n is the number of frames in the sequence. The extended temporal scope tracking resembles the beam-search principle in [38], trajectory aging in [13] and the N-scan-back principle in the statistical data association filters like the track-splitting filter [29], the MHT [24], and the sliding window algorithms [9] [10] [21] [22][1]. In contrast with our problem setting, [13] [29] [38] do not adopt the uniqueness constraint, hence, they

[1] A temporal scope $s = 1$ is similar to $N = 0$ in N-scan-back filters.

optimize the tracks independently. Clearly, that problem is less complex, though it may lead to unrealistic assignments. With respect to the multi-frame optimization, our approach is more similar to the MHT in [5] [8] [24] in the way that we also rank the best assignments per frame and finally decide for the 'optimal' assignment after a certain number of frames has been processed. However, our optimization strategy is quite different. We search the alternative solutions up to s levels in a depth-first way, whereas the track maintaining algorithms like the MHT can be said to search in a breadth-first way. The advantage of our method is that it needs less memory and allows for more effective pruning of unlikely alternatives. In the experiments section we give an indication of a suitable value for the temporal scope s.

In the next section we formulate the problem and give the notation we use. Then, we summarize and modify the motion models that we proposed in [33]. In Section 4 we introduce the new tracking algorithm that embodies the extended scope optimization scheme. In the experiments section, we show the appropriateness of the new algorithm both with synthetic and real-world data.

2 Problem Statement

Given is a sequence of n time instances for which at each time instance t_k there is a set of m_k measurements $\mathbf{x}_j^k$ of points p_i moving in a 3-D world, with $1 \leq j \leq m_k$, $1 \leq k \leq n$, and $1 \leq i \leq M$. The measurements are vectors in a two-dimensional space, with dimensions S_w (width) and S_h (height), representing 2-D coordinates. The number of measured points m_k can be either smaller (occlusion or missing detections) or larger (spurious measurements) than M.

The problem is to find a set of M tracks that represents the (projected) motion of the M points through the 2-D space from t_1 to t_n. A track T_i is an ordered n-tuple of corresponding measurements: $\langle \mathbf{x}_{j_1}^1, \mathbf{x}_{j_2}^2, ..., \mathbf{x}_{j_n}^n \rangle$, with $1 \leq j_k \leq m_k$. It is assumed that points do not enter or leave the scene (ignoring condition 4) mentioned above). A point track that has been formed up to t_k is called a track head and is denoted as T_i^k.

We use two additional ways to denote which measurement corresponds to which track head. First, we introduce the assignment matrix $A^k = [a_{ij}^k]$, where $a_{ij}^k = 1$ if and only if $\mathbf{x}_j^{k+1}$ corresponds to T_i^k and zero otherwise. Alternatively, we use $\alpha_j^k = i$ if $a_{ij}^k = 1$. Further, a concatenation of s assignment matrices from t_k to t_{k+s-1} is called a multi-assignment, denoted as an s-tuple: $A^{k:s} = \langle A^k, A^{k+1}, ..., A^{k+s-1} \rangle$, where $A^{k:s}[1] = A^k$, $A^{k:s}[2] = A^{k+1}$, etc.

3 Modeling

Here, we only give a brief description of our way to model the motion correspondence problem. For a more detailed description and analysis we refer to [33].

In order to select the corresponding measurement for a track head from the list of candidate measurements we need to have a model of the point motion: the *individual* motion model. In addition to prior motion models the parameters of such a model could be constructed on-line from the tracked measurements. Since it is impossible to rule out model errors, or, in other words, to predict the point positions perfectly, usually there will be correspondence ambiguities. Therefore, additional *combined* and *global* motion models have been proposed to make prediction errors dependent. Here, we summarize the individual, combined, and global models.

3.1 Individual Motion Model

The individual motion model expresses predictions about the position of a moving point based on historical track information. Further, it states the cost when deviating from these predictions. Here, we formulate two different individual motion models.

*im*1 The nearest-neighbor model does not incorporate velocity information. It only states that a point moves as little as possible from t_k to t_{k+1}. Consequently, the model uses only measurements of one previous time instance for the position prediction.

$$c_{ij}^k = \| \mathbf{x}_j^{k+1} - \mathbf{x}_i^k \|, \quad \text{where} \quad 0 \leq c_{ij}^k \leq \sqrt{S_w^2 + S_h^2}. \tag{1}$$

*im*2 The smooth-motion model as first introduced in [27] assumes that the velocity magnitude and direction both change gradually. This model uses measurements from two previous time instances. The smooth motion is formulated quantitatively with the following criterion:

$$c_{ij}^k \quad = \quad 0.1 \left[1 - \frac{\left(\mathbf{x}_i^k - \mathbf{x}_{\alpha_i^k}^{k-1}\right) \cdot \left(\mathbf{x}_j^{k+1} - \mathbf{x}_i^k\right)}{\left\|\mathbf{x}_i^k - \mathbf{x}_{\alpha_i^k}^{k-1}\right\| \left\|\mathbf{x}_j^{k+1} - \mathbf{x}_i^k\right\|} \right] + 0.9 \left[1 - 2 \frac{\sqrt{\left\|\mathbf{x}_i^k - \mathbf{x}_{\alpha_i^k}^{k-1}\right\| \left\|\mathbf{x}_j^{k+1} - \mathbf{x}_i^k\right\|}}{\left\|\mathbf{x}_i^k - \mathbf{x}_{\alpha_i^k}^{k-1}\right\| + \left\|\mathbf{x}_j^{k+1} - \mathbf{x}_i^k\right\|} \right], \tag{2}$$

where $0 \leq c_{ij}^k \leq 1$.

To enable the modeling of spurious and missing measurements, we first need to modify the assignment matrix format. To this end, we extend A^k such that it has $M + m_{k+1}$ rows and $M + m_{k+1}$ columns. The first M rows represent the track heads of the target points and the first m_k columns represent the true measurements. The remaining rows and columns represent *false tracks* (to assign spurious measurements to) and *slave measurements* (to replace missing measurements), respectively. Additionally, the matrix $D^k = [c_{ij}^k]$ contains the individual motion criterion coefficients, where c_{ij}^k expresses the deviation from the predicted position for measurement $\mathbf{x}_j^{k+1}$ to track head T_i^k. For true track heads to true measurements these coefficients are computed as defined above. All other entries in D^k equal ϕ_{max}, which is a known maximum of the individual motion criterion. For candidate correspondences that exceed a certain maximum speed (d_{max}) we set $c_{ij}^k = \phi_{max} + \epsilon$ to effectively disregard them[2].

Slave Interpolation

If any of the measurements in the vectors $(\mathbf{x}_i^k - \mathbf{x}_{\alpha_i^k}^{k-1})$ and $(\mathbf{x}_j^{k+1} - \mathbf{x}_i^k)$ are missing, the vectors are estimated by interpolation according to:

$$\mathbf{x}_i^k - \mathbf{x}_{\alpha_i^k}^{k-1} = \frac{\mathbf{x}_{\alpha_i^{k \to q}}^q - \mathbf{x}_{\alpha_i^{k \to p}}^p}{q - p}; \qquad \mathbf{x}_j^{k+1} - \mathbf{x}_i^k = \frac{\mathbf{x}_i^{k+1} - \mathbf{x}_{\alpha_i^{k \to q}}^q}{k + 1 - q}, \tag{3}$$

where $\mathbf{x}_{\alpha_i^{k \to p}}^p$ and $\mathbf{x}_{\alpha_i^{k \to q}}^q$ are true measurements in the nearest past in T_i^k, $1 \leq p < q \leq k$ and $\alpha_i^{k \to q}$ means $k - q$ times recursive application of α_i^k.

3.2 Combined Motion Model

The combined motion model serves to make individual model errors dependent between two successive frames. Here, we give only one such combined motion cost definition $C^k(A^k)$ (see [33] for alternative ones) that aims at spreading the errors as much as possible.

$$C^k(A^k) = \frac{1}{M} \sum_{i=1}^{M+m_{k+1}} \sum_{j=1}^{M+m_{k+1}} a_{ij}^k c_{ij}^k, \tag{4}$$

[2] Where ϵ is a arbitrary (small) positive number

3.3 Global Motion Model

The global motion model serves to model the overall motion from t_1 to t_n. It averages out the combined motion errors over time, and in this way it ensures that the combined motion errors depend on each other.

$$S(D) = \min_{A^{1:n-1}} \sum_{k=2}^{n-1} C^k(A^k) \tag{5}$$

This global model is, however, hard to optimize. Therefore, we redefine the global model with the temporal scope s as parameter.

$$S_s(D) = \sum_{k=2}^{n-1} C^k(A^{k:s}_{min}[1]), \tag{6}$$

where

$$A^{k:s}_{min} = \arg\min_{A^{k:s}} C^{k:s}(A^{k:s}) \quad \text{with} \quad C^{k:s}(A^{k:s}) = \sum_{p=1}^{s} C^{k+p-1}(A^{k:p}[p]) \tag{7}$$

When $s = 1$ Eq.6 equals $\hat{S}(D)$ in [33] and when $s = n - 1$ Eq.6 equals Eq.5.

4 Restrained Optimal Assignment Decision (ROAD) Tracker

As we already mentioned, the computation of the global motion model (Eq.5) is intractable in general. The complexity can be reduced by using a limited temporal scope s as in Eq.6. However, the problem to be solved is an $(s + 1)$-dimensional assignment problem with non-decomposable cost [1], which is known to be NP-hard for $s + 1 \geq 3$ [14]. Finding a greedy matching solution ($s = 1$) can be formulated as a classical (2-dimensional) assignment problem [33], for which a number of efficient solutions has been reported in the literature, among which the Hungarian method [15] is the best known. When the scope is larger, $s > 1$, the problem is still NP-hard. Therefore, it is important to limit s and to use a specific strategy to search the alternatives efficiently. Here, we propose the ROAD tracker, a recursive algorithm that searches the alternatives depth first up to s levels. Since it has been shown that the greedy solution is close to optimal [33], a best-first heuristic per recursion level is a suitable strategy.

Because the global motion criterion is additive and monotonic increasing, we also propose to use the *branch-and-bound* mechanism, where the initial bound is determined as the algorithm searches best first per recursion level in a depth-first way. Moreover, we adaptively lower the bound by introducing a combined motion constraint γ_{max}. For the computation of γ_{max} we introduce two assumptions. First, we assume that the cost of the solution $A^{k:s}_{min}$ is more or less uniformly spread over the recursion levels (scope). So after finding a temporary solution $A^{k:s}_{sol}$ with global cost (bound) C_b we stop testing alternative assignments if their cost exceeds $F^{\gamma}_g C_b/s$, where s is the (remaining) scope and F^{γ}_g ($F^{\gamma}_g \geq 1$) is a factor that expresses the maximum allowed deviation from C_b. Second, we assume that the optimal assignment $A^{k:s}_{min}[1]$ cannot have a much higher cost than the cost C^k_{min} of the local best assignment A^k_{min}. So we additionally stop the testing of alternatives if their cost exceeds $F^{\gamma}_l C^k_{min}$, where F^{γ}_l ($F^{\gamma}_l \geq 1$) expresses the maximum allowed deviation from C^k_{min}. This leads to the following combined motion constraint:

$$\gamma_{max} = \min(F^{\gamma}_l C^k_{min}, F^{\gamma}_g C_b/s) \tag{8}$$

In contrast with the constraints on the individual motion (d_{max} and ϕ_{max}), γ_{max} is not physically motivated. Consequently, when γ_{max} is used, the solution can no longer be guaranteed to be optimal with respect to the individual motion models.

Now we need a way to generate the assignments between t_k and t_{k+1} in best-first C^k-order. To this end we use Murty's algorithm [17], which is an efficient algorithm to rank assignments in order of increasing cost. This algorithm was used before to enumerate hypotheses for the statistical MHT [5]. In short, the Murty algorithm returns the minimum cost assignment for an assignment problem given a number of assignments Y is no longer allowed, where $Y \subseteq U^k$:

$$A^k_{min}(Y, D^k) = \arg \min_{A \in U^k - Y} C^k(A, D^k), \tag{9}$$

where U^k is the set of all possible assignment matrices at t_k.

4.1 Basic ROAD Tracker

After having introduced the main elements, we now describe the complete ROAD algorithm. The ROAD tracker has five parameters. The first parameter is A^{k-1} serves to initialize the individual motion models, hence to compute D^k (for $im2$). So far, we did not include this parameter in any of the criterion definitions (Eq.4-9). In the recursive calling of the ROAD algorithm, however, we include the A^{k-1} parameter, because the controlled permutation of A^{k-1} is the main ingredient of this recursive algorithm. Clearly, in the first frame the assignment for the previous frame A^{k-1} is not available, leading to an initialization problem. We return to this afterwards. The second parameter is the frame number to be processed k. The third parameter is the (remaining) scope to be optimized s. The fourth and fifth parameter are the cost bound C_b and the partial solution $A^{k:s}_{sol}$ that must be improved. If the tracker is not able to deliver a solution with lower cost than C_b then it returns the given $A^{k:s}_{sol}$.

The algorithm works as follows, see pseudo code in Tab. 1. First, it computes the criterion matrix D^k using A^{k-1} (line 1). If the scope $s = 1$, it determines the minimal cost assignment A^k_{min} with the Hungarian method, which is the same as the GOA tracker result (line 3). If the cost of A^k_{min} is below the cost bound C_b, then ROAD updates the solution $A^{k:s}_{sol}$ (line 5).

On the other hand if the scope is larger it starts enumerating the assignments according to increasing cost. To this end, it accumulates all processed assignments in Y which is initialized to the empty set (line 8). It computes the next best assignment A excluding Y (line 10), updates Y (line 11), and computes the cost C_0 of A (line 11). Then, it puts the tail from the current best solution $A^{k:s}_{sol}$ in T, so that it can be returned if it cannot be improved (line 12). Now ROAD calls itself recursively using A as A^{k-1} parameter and other updated parameters for the next recursion level. Further, it stores the resulting multi-assignment in R (line 14). Then at line 15 it composes a new solution which is tested against the current bound at line 16. If the cost is lower than the bound, the bound is updated (line 17) as well as the current best solution $A^{k:s}_{sol}$ (line 18).

Finally, the stopping criterion is tested. First, at line 20 the adaptive constraint γ_{max} is computed. Then, the algorithm halts if all solutions have been tried ($Y = U^k$), the cost of the next best assignment A exceeds the bound ($C_0 > C_b$) or the combined constraint γ_{max} ($C_0 > \gamma_{max}$), see line 21.

4.2 Self-Initializing ROAD Tracker

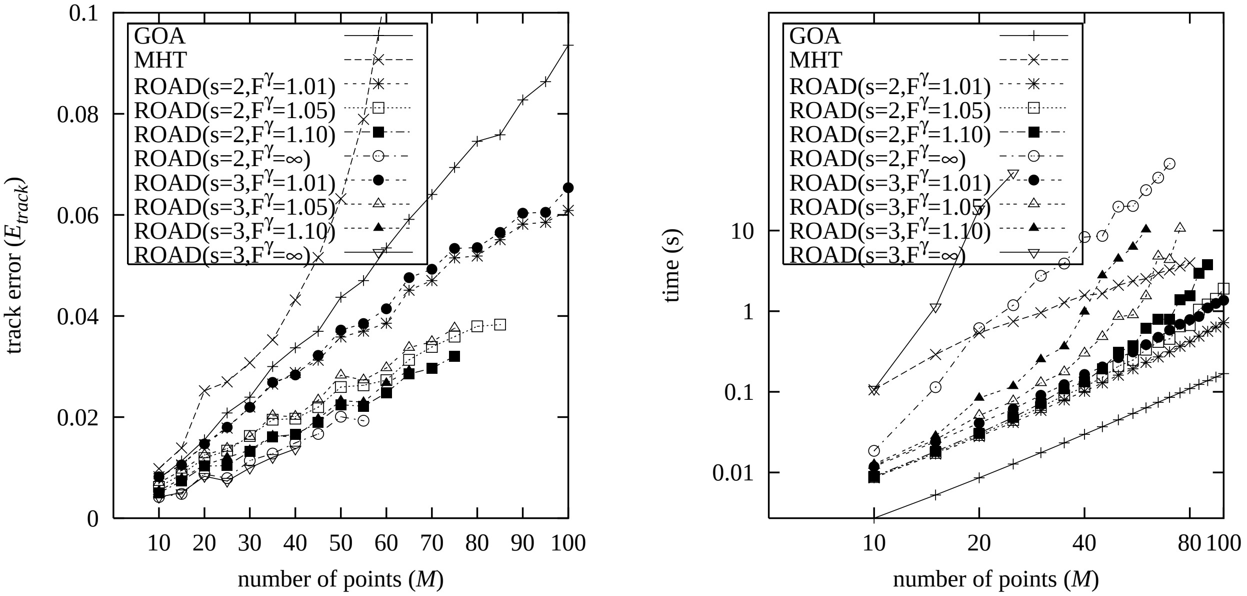
As mentioned, for the processing of the first frame the previous assignment $A^{k-1}(= A^1)$ is not available. We solve this initialization problem in the same way as in [33], that is, by initializing A^1 with the minimal cost assignment A^k_{min} using the individual model $im1$ and running the algorithm once *up* and once *down*. This results in the algorithm shown in Tab. 2.

```
ROAD($A^{k-1}, k, s, C_b, A^{k:s}_{sol}$)
$D^k = computeCostMatrix(A^{k-1}, k)$                        ; composee cost matrix $D^k$
if $s = 1$ then                                              ; at lowest recursion level?
    $A^k_{min} = minCostAssignment(D^k)$                     ; find minimum cost assignment
    if $C^k(A^k_{min}) < C_b$ then                           ; better than cost bound?
        $A^{k:s}_{sol} = \langle A^k_{min} \rangle$          ; update solution
    end
else
    $Y = \emptyset$                                          ; set of processed matrices
    do
        $A = getNextBestAssignment(Y, D^k)$                  ; get next best with Murty
        $Y = Y \cup \{A\}$                                   ; add to processed set
        $C_0 = C^k(A)$                                       ; compute cost
        $T = A^{k:s}_{sol}[2...s]$                           ; get default solution
        $R = \mathrm{ROAD}(A, k+1, s-1, C_b - C_0, T)$       ; call recursively to improve $T$
        $A^{k:s} = \langle A \rangle \circ R$                ; concatenate $A$ with new tail
        if $C^{k:s}(A^{k:s}) < C_b$ then                     ; better than global bound?
            $C_b = C^{k:s}(A^{k:s})$                         ; update global bound
            $A^{k:s}_{sol} = A^{k:s}$                        ; update solution
        end
        $\gamma_{max} = \min(F^{\gamma}_l C^k_{min}, F^{\gamma}_g C_b/s)$   ; compute combined constraint
    while $(Y \neq U^k \wedge C_0 < C_b \wedge C_0 < \gamma_{max})$       ; stop when global bound or
end                                                          ; combined constraint exceeded
                                                             ; or no alternatives are left
return $A^{k:s}_{sol}$                                       ; return solution
```

Table 1: The ROAD tracker for recursive multiple frame assignment optimization

	let $A^1 = A^1_{min}$	initialize first assignment with *im*1
	let $k = 2$	start at second frame
up:	$A^k = \mathrm{ROAD}(A^{k-1}, k, s, \infty, \langle\rangle)$	find optimal assignment with *im*2 and scope s
	increase k	
	if $k < n$ go to *up*	
	otherwise go to *down*	
down:	decrease k	
	$A^k = \mathrm{ROAD}(A^{k+1}, k, s, \infty, \langle\rangle)$	find optimal assignment with *im*2 and scope s
	if $k > 2$ go to *down*	
	otherwise done	

Table 2: The self-initializing ROAD tracker.



(a)

(b)

Figure 2: (a) Shows the track error as a function of the number of points and (b) shows the computation time as a function of the number of points.

5 Experiments

With the experiments, we intend to show the improved performance that can be achieved by restraining the assignment decisions using an increased temporal scope ($s > 1$). First, we did a number of synthetic data experiments for which we used the PSMG data generator [34]. Afterwards we also applied the algorithm to a recorded image sequence.

With the PSMG data generator we ran several tests with varying scope s and constraint factors F_l^γ and F_g^γ. For details on the used PSMG parameter settings, see [33]. We always set F_l^γ equal to F_g^γ (both denoted as F^γ), $\phi_{max} = 0.2$, and we fed the algorithm with the true (known) maximum speed in order to disregard physically impossible correspondences. In the synthetic experiments, we compared the results with those of the original GOA tracker, which is the same as the ROAD tracker with scope $s = 1$. As a reference tracking algorithm we added the well-known statistical MHT [24] as described and implemented by Cox and Hingorani [5]. The essential parameters of the MHT were trained with a genetic algorithm on labeled data sets of 50 points; there were no missing or spurious measurements. For the MHT the parameters referring to the probabilities of detection and false alarms were adjusted according to the PSMG settings in the respective experiment. Importantly, since the MHT is able to track a varying number of points, we set the probabilities of track initiation and termination to zero as to inform that the number of tracked points is fixed. Both for the ROAD tracker and the MHT we set (additional) pruning parameters to limit the solution space in addition to the model constraints. Further, the ROAD tracker evaluates no more than 300 candidates at each recursion level. The MHT has at maximum 300 global hypotheses[3] per group, a track tree depth of 3, while the minimum ratio between the likelihoods of the best and the worst hypothesis is 0.005. All displayed results are the average of 500 runs. When the average run time of an experiment exceeded 10 seconds, the experiment was stopped. As a consequence, some curves in the figures are incomplete.

[3] The number of candidates for the ROAD tracker and the number of global hypotheses per group for the MHT have different meanings.

5.1 Variable Density Experiment.

In the first experiment, we explored the performance of the ROAD tracker as a function of the point density. The performance is expressed as the average ratio of incorrect tracks and total number of tracks, which we call the track error (E_{track}) after [34]. Fig.2(a) clearly shows that the ROAD tracker outperforms both the GOA tracker and the MHT. Further, the less constrained the combined motion is, the better the performance. Remarkably, the track error with scope $s = 3$ is larger than with $s = 2$ when the same combined constraint setting is used. This is because with a larger scope $s = 3$ the global cost C_b can decrease faster, resulting in a stricter combined motion constraint at the highest search levels. However, the unconstrained experiments ($F_l^\gamma = F_g^\gamma = \infty$) show that with $s = 3$ the best results can be accomplished[4]. Although the difference between the unconstrained $s = 2$ and $s = 3$ performance can hardly be noticed in the figure, the significance study below supports that with $s = 3$ the best performance can be achieved. Nevertheless the computation time quickly becomes a bottleneck, as the next experiment will demonstrate.

In order to establish the significance of the relative ranking of the different trackers and/or tracker parameter settings, we computed the Wilcoxon matched-pairs signed ranks test for the directional hypothesis that one tracker is better than the other for the tracking of 50 points. The ranking of the trackers with siginificance level $\alpha \ll 0.001$ is as follows: ROAD($s = 3, F^\gamma = \infty$) $>$ ROAD($s = 2, F^\gamma = \infty$) $>$ ROAD($s = 2, F^\gamma = 1.10$) $>$ ROAD($s = 2, F^\gamma = 1.05$) $>$ ROAD($s = 2, F^\gamma = 1.01$) $>$ GOA $>$ MHT. So with the same F^γ setting there is no significant difference between the ROAD tracker performance with scope $s = 2$ and $s = 3$, except when $F^\gamma = \infty$.

5.2 Variable Volume Experiment.

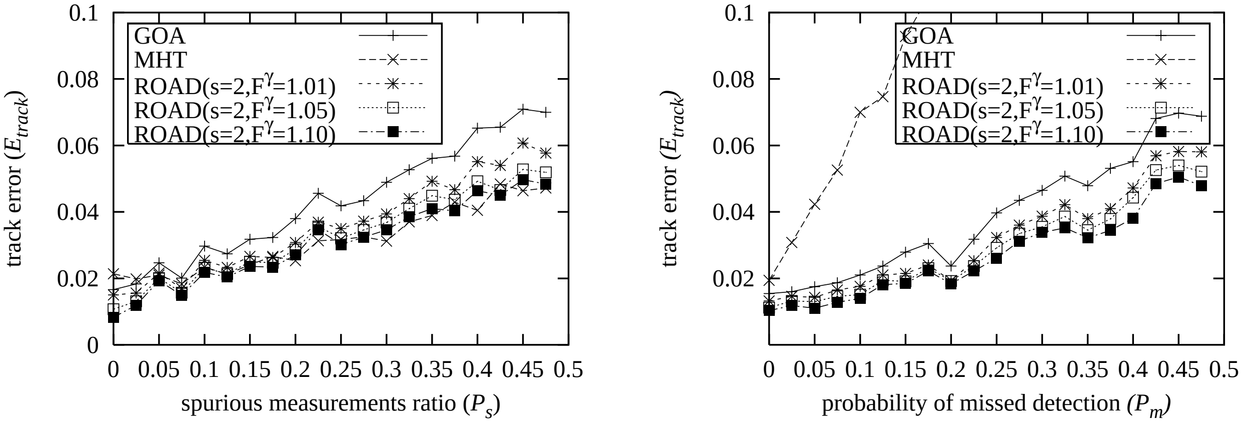
In the next experiment, we varied the number of points while the point density remained the same. Consequently, the problem remains equally difficult. Fig.2(b) shows that the GOA tracker is the fastest and has polynomial complexity. The ROAD tracker has exponential complexity, but when $s = 2$ and F_l^γ and F_g^γ are low ($1 \leq F_l^\gamma, F_g^\gamma \leq 1.05$), the exponential order is also quite low, so that near-polynomial behavior is achieved over a range from 10 to 100 points. The MHT is slow but, because of the pruning parameters, it has near-polynomial complexity. It has, however, to be mentioned that the track error increases considerably with the number of points[5].

5.3 Variable Number of Spurious Measurements.

In order to show the importance of deferring correspondence decisions in the presence of noise, we did an experiment where we gradually incremented the number of spurious measurements. The number of spurious measurements is normally distributed around the displayed mean ratio P_s of the number of points ($M = 20$). As an example: $P_s = 0.5$ implies an average of 10 spurious measurements per frame. The position of these measurements is uniformly spatially distributed. Fig.3(a) indeed shows that the average track error is lower when $s > 1$. Moreover, deferring the assignment decisions even has as a result that the difference between the ROAD tracker and the GOA tracker becomes larger as the amount of spurious measurements grows. With the specifically trained MHT the average track error hardly increases. When the noise ratio $P_s > 0.1$, the MHT performs best on the average. For $P_s = 0.25$ the difference between the ROAD tracker and the MHT is, however, not significant as follows from the Wilcoxon matched-pairs signed ranks test. Both the ROAD tracker and the MHT indeed perform significantly better than the GOA tracker ($\alpha \ll 0.001$) .

[4] In the unconstrained experiments we set the maximum average run time to 100 sec. in order to show that the performance indeed improves.

[5] In contrast to the MHT, the GOA tracker has a constant track error, and with the ROAD tracker the track error even decreases with the number of points.



(a) (b)

Figure 3: (a) Shows the track error as a function of the spurious measurements ratio P_s and (b) shows the track error as a function of the probability of missed detection P_m.

It followed from additional experiments that the GOA tracker and the ROAD tracker performed worse for high P_s values because of the noise sensitivity of their initialization scheme. That is, especially the spurious measurements in the second frame result in deviant initial motion vectors. Then, during the *up* optimization, some true measurements may be considered as noise. Because the *down* optimization scheme only operates on tracks that have measurements in the last frames, it is not always possible to undo the effects of such an ill initialization. However, if there are no spurious measurements in the second frame, the GOA tracker and the ROAD tracker always perform better than the MHT for all settings of P_s.

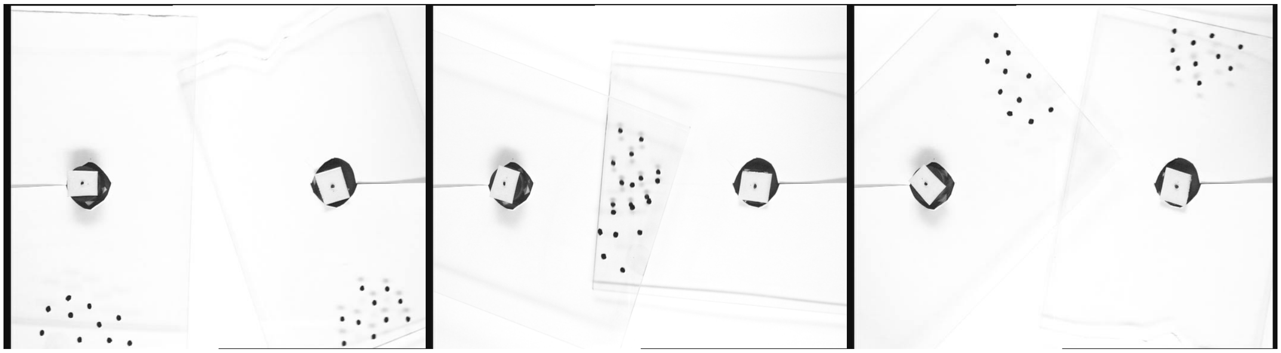
5.4 Variable Number of Missing Measurements.

In the last synthetic experiment, we evaluated the influence of varying the number missing detections. In Fig.3(b) we display the track error as a function of the probability that a point was not detected or missed P_m. According to the problem definition, i.e. no scene entrance and exit, all points are detected in the first and last two frames. Again the number of points is $M = 20$. The ROAD tracker with various settings performs better than the GOA tracker. Also in this experiment the difference between the ROAD tracker and the GOA tracker increases as the problem becomes more difficult. The MHT turns out to be extremely sensitive to occlusion. Part of the problem is that, although the probability of detection is set properly, the MHT easily divides tracks into separate parts. The ranking of the trackers for $P_m = 0.25$ with siginificance level $\alpha \ll 0.001$ is as follows: ROAD$(s = 2, F^{\gamma} = 1.05)$ > ROAD$(s = 2, F^{\gamma} = 1.01)$ > GOA > MHT, while ROAD$(s = 2, F^{\gamma} = 1.10)$ outperforms ROAD$(s = 2, F^{\gamma} = 1.05)$ with significance $\alpha < 0.02$.

5.5 Revolving Transparent Plates Experiment.

After the synthetic data experiments, we recorded an image sequence with two transparent plates revolving on top of each other in opposite direction. The plates that each contain 10 black spots have similar rotational speed. In Fig. 4 we show some images from this sequence. We applied the trackers to the spot positions found after segmentation of the images. The difficulty with this sequence is that when the spots of both plates meet (see Fig. 4(b)) some spots are occluded and the spot tracks cross, which leads to many assignment ambiguities.

For both the GOA tracker and the ROAD tracker we set $d_{max} = 100$ and $\phi_{max} = 0.1$. We ran the



(a) frame 1 (b) frame 8 (c) frame 14

Figure 4: The first, middle, and last image from the revolving plates sequence; recorded with a 25 Hz progressive scan camera using 2 ms shutter speed.

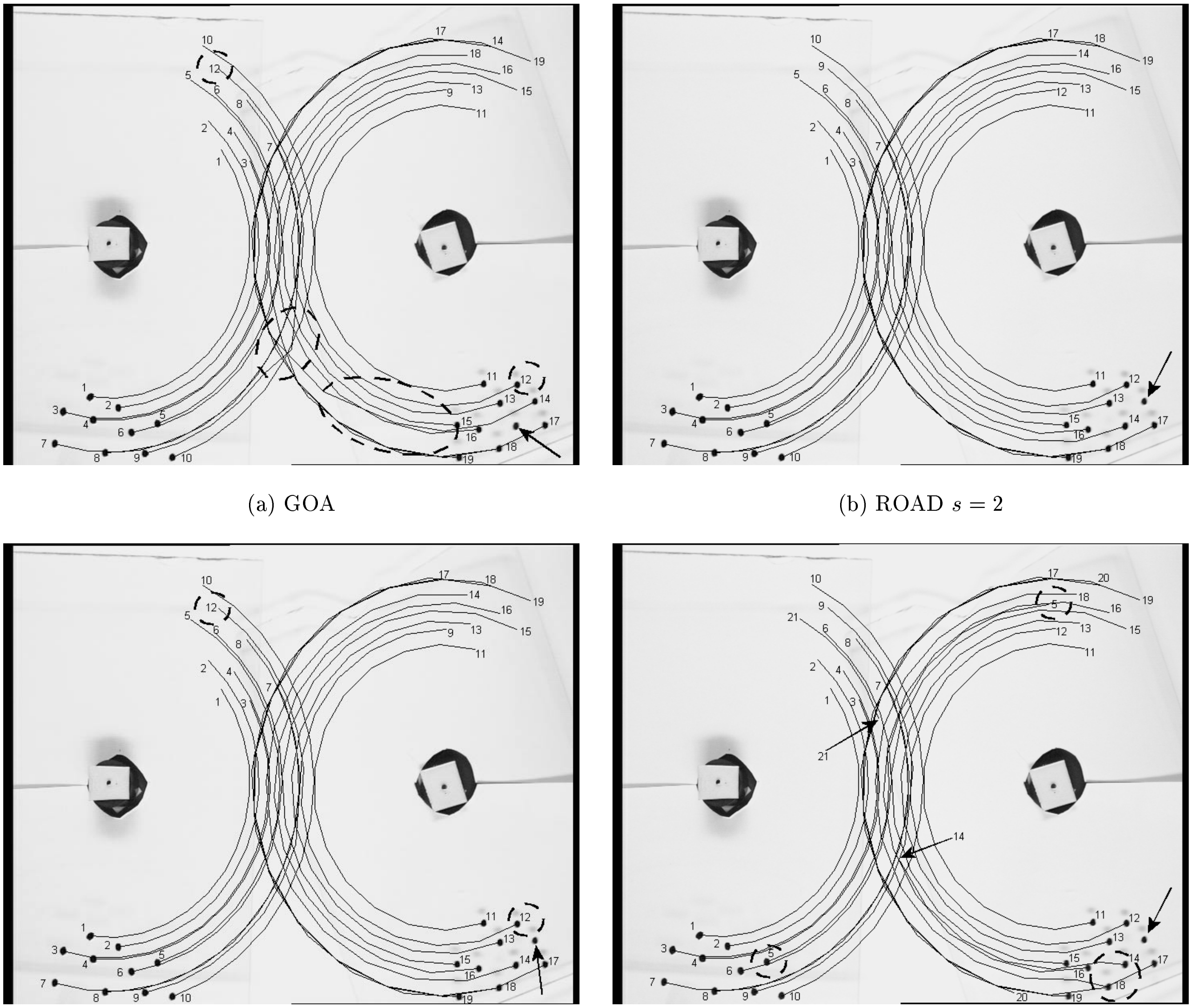
ROAD tracker once with $s = 2/F^\gamma = 1.05$ and once with $s = 3/F^\gamma = 1.05$. Again the MHT parameters were trained on manually labeled tracks using a genetic algorithm. It has to be noted that this type of training is actually undesirable. First, in practical situations the track labels are not available. More importantly, the derived performance measure is unreliable because the parameters are specifically fit to this set of tracks. Accordingly, the error, that is achieved after training, shows to what extent the tracker can be adjusted to a certain data set. Since the MHT is quite sensitive to its parameter settings, it is, however, extremely difficult to find the right setting. For a parameter sensitivity study we refer to [33].

As is indicated with an arrow in the respective subfigures in Fig.5, GOA and ROAD lost track of one spot. This is caused by an intialization error in the *up* optimization direction. Further, Fig.5(a) shows that the GOA tracker makes some additional errors as indicated with the dashed ellipses. The ROAD tracker with scope $s = 2$ perfectly tracks the remaining spots (visually inspected) which can be verified from the displayed track id's at the start and end of the tracks, see Fig. 5(b). With scope $s = 3$ the ROAD tracker makes one error that the GOA tracker also made. That is, it wrongly connects two track parts from spots from different plates (track 12) as is indicated with the dashed ellipse in Fig. 5(c). This error is caused by a misleading assignment that is considered at the deepest recursion level which turns out to be non-optimal afterwards. With scope $s = 4$ this error is not made (not shown in the figure) leading to the same result as with $s = 2$. Also the specifically trained MHT was able to track most points, though it made some errors as indicated with arrows and ellipses in Fig. 5(d). Among others it wrongly connects track parts from different plates (track 5) and it made some initializition errors (track 14 and 18). Further there are two partial tracks (track 14 and 21), from which track 14 connects some measurements from a point that was temporary occluded (the other trackers completely missed this track) and track 21 should be the end of the erroneous track that starts as track 5 in the lower left region, see Fig. 5(d).

6 Conclusion

In this paper, we described the ROAD tracker, a recursive algorithm that establishes motion correspondence by optimizing over several frames using a non-statistical motion framework. At each recursion level the tracker evaluates candidate assignments in best-first order using Murty's algorithm.

As an extension of the GOA tracker, the ROAD tracker uses the same individual, combined and global motion models. We introduced an approximation of the global motion model, which additionally has a temporal scope parameter. Further, we introduced an adaptive combined motion constraint γ_{max} on top of a branch-and-bound mechanism to reduce the exponential growth in computation time. The



(a) GOA

(b) ROAD $s = 2$

(c) ROAD $s = 3$

(d) MHT

Figure 5: Tracking results obtained with (a) the GOA tracker, (b) the ROAD tracker with $s = 2$ and $F^\gamma = 1.05$, (c) the ROAD tracker with $s = 3$ and $F^\gamma = 1.05$, and (d) the specifically trained MHT.

various synthetic and real-world experiments showed that the deferment of assignment decisions indeed improves the tracking performance significantly. Even with a very strict combined constraint setting ($1 \leq F_l^\gamma, F_g^\gamma \leq 1.05$), the ROAD tracker clearly outperforms the GOA tracker in all experiments. Relaxing this constraint further improves the performance, but care must be taken since unconstrained assignment optimization over several frames is intractable in general, as the experiments have shown. The experiments have also shown that setting the temporal scope to $s = 2$ gives the best compromise between qualitative and computational performance. In the experiments we also included the statistical MHT. Unlike the proposed algorithm the MHT is able to track a varying number of points. We set the respective parameters as to inform the MHT that the number of points was fixed. Like in [33], we noticed that setting the remaining MHT parameters is generally difficult. Having said this, the ROAD tracker also performs better than the specifically trained MHT, except when there were a lot of spurious measurements in the first frames. This issue needs further investigation.

Some additional remarks about the efficiency of the algorithms. The GOA tracker is the fastest and the only algorithm with polynomial complexity. The MHT is the slowest, though the MHT is difficult to judge in terms of computational complexity. That is, it was not possible to configure the MHT such that it had a constant track error in the variable-volume experiment. Accordingly, its reported efficiency is probably too optimistic.

As the unconstrained experiments show, the track error performance can hardly be improved given the applied composite motion model. The efficiency of the proposed algorithm can, however, be improved by implementing optimizations to Murty's algorithm, as reported in [2] and [16].

Finally, the next extension to the ROAD tracker must be to allow for the tracking of a variable number of points. When points or objects can enter and leave the scene, the algorithm can be applied in a broader domain.

References

[1] H-J. Bandelt, Y. Crama, and F.C.R. Spieksma. Approximation algorithms for multi-dimensional assignment problems with decomposable costs. *Discrete Applied Mathematics*, 49:25–50, 1994.

[2] M. Bellmore and J.C. Malone. Pathology of traveling-salesman subtour-elimination algorithms. *Operations Research*, 19:278–307, 1971.

[3] D. Chetverikov and J. Verestóy. Feature point tracking for incomplete trajectories. *Computing, Devoted Issue on Digital Image Processing*, 62:321–338, 1999.

[4] I.J. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, 10(1):53–66, 1993.

[5] I.J. Cox and S.L. Hingorani. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):138–150, February 1996.

[6] I.J. Cox and M.L. Miller. On finding ranked assignments with applications to multi-target tracking and motion correspondence. *IEEE Transactions on Aerospace and Electronic Systems*, 32(1):486–489, January 1995.

[7] I.J. Cox, M.L. Miller, R. Danchick, and G.E. Newnam. A comparison of two algorithms for determining ranked assignments with application to multi-target tracking and motion correspondence. *IEEE Transactions on Aerospace and Electronic Systems*, 33(1):295–300, January 1997.

[8] R. Danchick and G.E. Newnam. A fast method for finding the exact N-best hypotheses for multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 29(2):555–560, April 1993.

[9] S. Deb, K.R. Pattipati, and Y. Bar-Shalom. A new algorithm for the generalized multidimensional assignment problem. *IEEE International Conference on Systems, Man and Cybernetics; Emergent Innovations in Information Transfer Processing and Decision Making*, pages 249–254, 1992.

[10] 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.

[11] T.E. Fortmann, Y. Bar-Shalom, and M. Sheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3):173–184, July 1983.

[12] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[13] V.S.S. Hwang. Tracking feature points in time-varying images using an opportunistic selection approach. *Pattern Recognition*, 22(3):247–256, 1989.

[14] R.M. Karp. Reducibility among combinatorial problems. In R.E. Miller and J.W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.

[15] H.W. Kuhn. The hungarian method for solving the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[16] M.L. Miller, H.S. Stone, and I.J. Cox. Optimizing Murty's ranked assignment method. *IEEE Transactions on Aerospace and Electronic Systems*, 33(3):851–861, 1997.

[17] K.G. Murty. An algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, 16:682–687, 1968.

[18] V. Nagarajan, M.R. Chidambara, and R.N. Sharma. Combinatorial problems in multitarget tracking - a comprehensive solution. *IEE Proceedings*, 134(1):113–118, February 1987.

[19] H.H. Nagel. Displacement vectors derived from second order intensity variations in image sequences. *Computer Vision Graphics and Image Processing*, 21(1):85–117, 1983.

[20] R. Pless, T. Brodský, and Y. Aloimonos. Detecting independent motion: The statistics of temporal continuity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):768–773, August 2000.

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

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

[23] K. Rangarajan and M. Sha. Establishing motion correspondence. *CVGIP: Image Understanding*, 24(6):56–73, July 1991.

[24] D.B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, December 1979.

[25] J.W. Roach and J.K. Aggarwal. Determining the movement of objects from a sequence of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(6):554–562, 1980.

[26] V. Salari and I.K. Sethi. Feature point correspondence in the presence of occlusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):87–91, January 1990.

[27] I.K. Sethi and R. Jain. Finding trajectories of feature points in a monocular image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):56–73, January 1987.

[28] P.J. Shea and A.B. Poore. Computational experiences with hot starts for a moving window implementation of track maintanance. In *Proceedings of the SPIE: The International Society for Optical Engineering*, volume 3373, pages 428–439, 1998.

[29] P. Smith and G. Buechler. A branching algorithm for discriminating and tracking multiple objects. *IEEE Transactions on Automatic Control*, 20:101–104, February 1975.

[30] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, August 2000.

[31] R.Y. Tsai and T.S. Huang. Estimating three dimensional motion parameters of a rigid planar patch, III: Finite point correspondences and the three view problem. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32:213–220, 1984.

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

[33] 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.

[34] 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.

[35] J.A. Webb and J.K. Aggarwal. Structure from motion from rigid and jointed objects. *Artificial Intelligence*, 19:107–130, 1982.

[36] L. Wixson. Detecting salient motion by accumulating directionally-consistent flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):774–780, August 2000.

[37] Z. Zhang. Estimating motion and structure from correspondences of line segments between two perspective images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1129–1139, 1995.

[38] Z. Zhang and O. Faugeras. Three-dimensional motion computation and object segmentation in a long sequence of stereo frames. *International Journal of Computer Vision*, 7(3):211–241, 1992.