

Resolving Motion Correspondence for Densely Moving Points

Cor J. Veenman, Marcel J.T. Reinders, and Eric Backer, *Member, IEEE*

Abstract—This paper studies the motion correspondence problem for which a diversity of qualitative and statistical solutions exist. We concentrate on qualitative modeling, especially in situations where assignment conflicts arise either because multiple features compete for one detected point or because multiple detected points fit a single feature point. We leave out the possibility of point track initiation and termination because that principally conflicts with allowing for temporary point occlusion. We introduce individual, combined, and global motion models and fit existing qualitative solutions in this framework. Additionally, we present a new efficient tracking algorithm that satisfies these—possibly constrained—models in a greedy matching sense, including an effective way to handle detection errors and occlusion. The performance evaluation shows that the proposed algorithm outperforms existing greedy matching algorithms. Finally, we describe an extension to the tracker that enables automatic initialization of the point tracks. Several experiments show that the extended algorithm is efficient, hardly sensitive its few parameters, and qualitatively better than other algorithms, including the presumed optimal statistical multiple hypothesis tracker.

Index Terms—Motion correspondence, feature point tracking, target tracking, algorithms.

1 INTRODUCTION

MOTION correspondence has a number of applications in computer vision, ranging from motion analysis, object tracking and surveillance, to optical flow and structure from motion [11], [24], [25], [27]. Motion correspondence must be solved when features are to be tracked that appear identical or that are retrieved with a simple feature detection scheme which loses essential information about their appearance. Hence, the motion correspondence problem deals with finding corresponding points from one frame to the next in the absence of significant appearance identification (see Fig. 1a). The goal is to determine a path or track of the moving feature points from entry to exit from the scene or from the start to the end of the sequence. During presence in the scene, a point may be temporarily occluded by some object. Additionally, a point may be missed and other points may be falsely detected because of a failing detection scheme, as in Fig. 1b and Fig. 1c.¹

A candidate solution to the correspondence problem is a set of tracks that describes the motion of each point from scene entry to exit. We adopt a uniqueness constraint stating that one detected point uniquely matches one feature point. When 2D projections from a 3D scene are analyzed, this is not trivial because one feature point may obscure another. If we further assume that all M points are

detected in all n frames, the number of possible track sets is $(M!)^{n-1}$. Among these solutions, there is a unique track set that describes the true motion of the M points. In order to identify the true motion track set, we need prior knowledge about the point motion because, otherwise, all track sets are equally plausible. This knowledge can range from general physical properties, like inertia and rigidity, to explicit knowledge about the observed objects, like the possible movements of a robot arm in the case that points on a robot arm are to be tracked. Clearly, generic motion correspondence algorithms cannot incorporate scene information. Moreover, they do not differentiate between the points in the scene, i.e., all points are considered to have similar motion characteristics.

When many similar points are moving through a scene, ambiguities may arise because a detected point may well fit correctly to the motion model of multiple features points. Additional ambiguities are caused by multiple detected points that fit correctly to the model of a single feature. These correspondence ambiguities can be resolved if combined motion characteristics are modeled, like least average deviation from all individual motion models. Besides resolving these ambiguities, we also have to incorporate *track continuation* in order to cope with point occlusion and missing detections. Other events that we may need to model are *track initiation* and *track termination* so that features can enter and leave the scene, respectively.

The available motion knowledge is usually accumulated in an appropriate model. Then, a specific strategy is needed to find the optimal solution among the huge amount of candidate solutions defined by the model. When the nearest-neighbor motion criterion is used (see also Section 3) and there are neither point occlusions nor detection errors, track set optimality only depends on the point distances between any two consecutive frames. Then, it is legitimate to restrict the scope of the correspondence decision to one frame ahead, which we call a *greedy*

1. In the remainder of this paper, we display the measurements from different time instances in one box and use t_k labels to indicate the time the point was detected.

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

Manuscript received 6 Dec. 1999; revised 20 June 2000; accepted 18 Sept. 2000.

Recommended for acceptance by I. Sethi.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 111039.

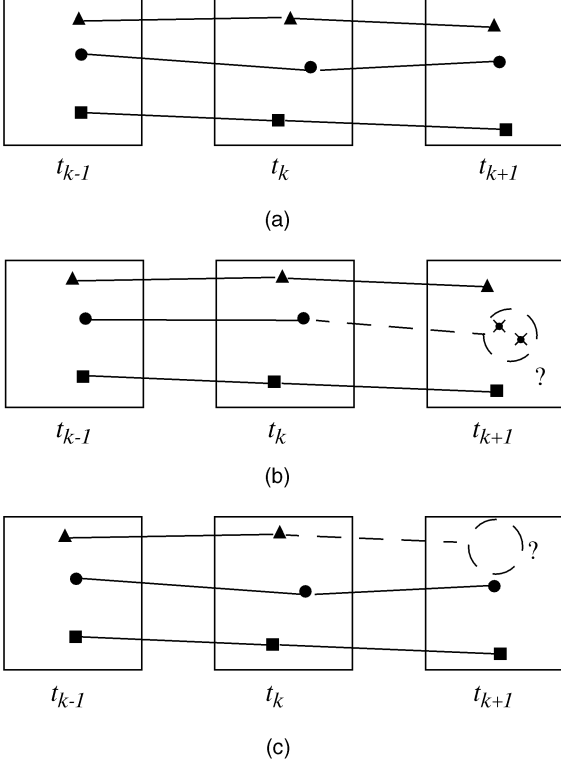


Fig. 1. Three moving points are measured at three time instances. The lines represent the point correspondences in time. In (a), all points are measured at every time instance. In (b), there is an extra or false measurement at t_{k+1} , and, in (c), there is a missing measurement at t_{k+1} .

matching solution to the correspondence problem. In other cases in which velocity state information is involved, correspondence decisions for one frame influence the optimal correspondence for the next frames and the problem becomes increasingly more complex. In such cases, only a *global matching* over all frames can give the optimal result. In this paper, we consider the more difficult cases, i.e., dense and fast moving points, which makes the use of velocity state information essential. Because there are no efficient algorithms to find the optimal track set by global matching, only approximation techniques apply. Several statistical [2] and qualitative approximation techniques have been developed both in the field of target tracking and computer vision.

1.1 Statistical Methods

The two best known statistical approaches are the Joint Probabilistic Data-Association Filter (JPDAF) [9] and the Multiple Hypothesis Tracker (MHT) [20]. 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 [3], [4], [5], [6], [15], [16]. 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 [7], [8], [17], [18], [23].

A number of reasons make the statistical approaches less suitable as a solution to the motion correspondence problem. First, the assumptions that the points move independently and, more strongly, that the measurements are distributed normally around their predicted position may not hold. Second, since statistical techniques model all events as probabilities, these techniques typically have quite a number of parameters, such as the Kalman filter parameters, and a priori probabilities for false measurements and missed detections. In general, it is certainly not trivial to determine optimal settings for these parameters. In Section 6.3, we show that the best-known statistical method (MHT) is indeed quite sensitive to its parameter setting. Moreover, the a priori knowledge used in the statistical models is not differentiated between the different points. As a consequence, the initialization may be severely hampered if the initial point speeds are widely divergent because the state of the motion models only gradually adapts to the measurements. Finally, the statistical methods that optimize over several frames are, despite their approximations, computationally demanding since the complexity grows exponentially with the number of points.

1.2 Heuristic Methods

Alternatively, a number of attempts has been made to solve the motion correspondence problem with deterministic algorithms [1], [12], [14], [19], [22]. These algorithms are usually conceptually simpler and have fewer parameters. Instead of probability density functions, *qualitative motion heuristics* are used to constrain possible tracks and to identify the optimal track set. By converting qualitative descriptions like smoothness of motion and rigidity into quantitative measures, a distance from the optimal motion can be expressed (where a zero distance makes a correspondence optimal). The most commonly known algorithm is the conceptually simple greedy exchange algorithm [22], which iteratively optimizes a local smoothness of motion criterion averaged over all points in a sequence of frames. The advantage of such deterministic algorithms is that it is quite easy to incorporate additional constraints, like (adaptive) maximum speed and a maximum deviation from smooth motion, while this a priori knowledge can restrain the computational cost and improve the qualitative performance, e.g., [1], [10].

The main contributions of this paper are the presentation of a 1) *qualitative motion modeling framework* for the motion correspondence problem. We introduce the notion of individual motion models, combined motion models, and a global motion model and we differentiate between strategies to satisfy these models. Further, we propose 2) a *new efficient algorithm* that brings together the motion models, an optimal strategy, and an effective way to handle detection errors and occlusion. Finally, we present 3) an extensive *comparative performance evaluation* of a number of different qualitative methods.

The outline of the paper is as follows: We start by giving a formulation of the motion correspondence problem in the

next section. Then, in Section 4, we present our qualitative motion model and show how existing deterministic motion correspondence algorithms can be fit into it. Additionally, we present a new algorithm that effectively resolves motion correspondence using the presented model in Section 5. In Section 6, we compare the qualitative performance, the efficiency, and the parameter sensitivity of the described algorithms. Further, we show how the proposed algorithm can be extended with self-initialization and evaluate it with synthetic data experiments in Section 7. We broaden this evaluation in Section 8 with real-data experiments. We finish the paper with a discussion on possible extensions and some conclusions.

2 PROBLEM STATEMENT

In this section, we describe the motion correspondence problem as treated in this paper. In motion correspondence, the goal is tracking points that are moving in a 2D space that is essentially a projection of a 3D world. The positions of the points are measured at regular times, resulting in a number of point locations for a sequence of frames. For the moment, we assume that we have initial motion information of all points, which is given by point correspondences between the first two frames. From Section 7 onward this restriction is lifted. Since the measured points are projections, points may become occluded and, thus, missing. Moreover, the point detection may be imperfect, resulting in missing and false point measurements. Because long occlusion, on the one hand, and scene entrance and exit, on the other hand, are conflicting requirements, we leave out the possibility of track initiation and track termination so the number of features to be tracked is constant. Applications using this problem definition range from object tracking in general, like animal tracking to perform behavior analysis, particle tracking, and cloud system tracking, to feature tracking for motion analysis. In the remainder of this paper, we abbreviate the moving points to “points” and their measured 2D projections to “measurements.”

More formally, there are M points, p_i , moving around in a 3D world. Given is a sequence of n time instances for which, at each time instance t_k , there is a set X^k of m_k measurements x_j^k , with $1 \leq j \leq m_k$, and $1 \leq k \leq n$, of points p_i . The measurements x_j^k are vectors representing 2D coordinates in a 2D space, with dimensions S_w (width) and S_h (height). The number of measurements, m_k , at t_k , can be either smaller (occlusion) or larger (false measurements) than M . At t_1 , the M points ($M \leq m_1$) are identified among the m_1 measurements. Moreover, the corresponding M measurements at t_2 are given. The task is to return a set of M tracks that represent the (projected) motion of the M points through the 2D space from t_1 to t_n using the movements between t_1 and t_2 as initial motion characteristics. A track T_i , with $1 \leq i \leq M$, is an ordered n -tuple of corresponding measurements: $\langle x_{j_1}^1, x_{j_2}^2, \dots, x_{j_n}^n \rangle$, with $1 \leq j_k \leq m_k$. It is assumed that points do not enter or leave the scene and that the movement can be modeled independently. A track that has been formed up to t_k is called a track head and is denoted as T_i^k , where $1 \leq i \leq M$.

3 QUALITATIVE MOTION MODELING

The assumption underlying the qualitative model that we advocate is that points move smoothly from time instance to time instance. That is, besides that individual points move smoothly, also the total set of points moves smoothly between time instances as well as over the whole sequence. We define a qualitative model in which these qualitative statements are explicitly represented by a composition of motion models, that we have called the *global motion model*, the *combined motion model* and the *individual motion model*. The individual motion model represents the motion of individual points. To embed the motion smoothness constraints, we can make use of well-known general physical properties like rigidity and inertia. Without loss of generality, we only consider first-order motions and, thus, leave out acceleration-state information. Consequently, the motion vector of a feature point can be estimated from only two consecutive measurements. On the basis of the motion vector and the adopted individual motion model, the position of the point at the next time instance can be predicted. The measurement that is closest to this prediction can then be selected as corresponding measurement. In reality, however, the points do not move exactly according to their predictions because of shortcomings of the adopted individual motion model. These are, among others, caused by the limited order of the motion model, the fact that measurements are 2D projections of 3D movements, and by noise in the system.

To express the misfit between a measurement and the predicted position, the candidate motion vector between the candidate measurement and the last measurement in the track is calculated. Using the inertia argument, the cost representing the misfit is expressed in terms of the candidate motion and the previous true motion vector. This cost can be used to select the appropriate candidate measurement to make a correspondence. When points are moving far apart from each other or when they move reasonably according to their models, their measurements can easily be assigned to the corresponding feature point. With densely moving point sets, however, assignment conflicts can easily occur. That is, one measurement fits correctly multiple individual motion models or multiple measurements correctly fit one motion model. To resolve these ambiguities, the motion smoothness constraint is also imposed on the complete set of points. To this end, we introduce the *combined motion model* that expresses the deviation from this motion constraint. As an example, we could enforce that the average deviation from the individual motion models is minimal.

Even with the use of the combined motion model, it is not always possible to decide on point correspondences. For that reason, the motion smoothness constraint is additionally extended over the whole sequence in the *global motion model*.

In the remainder of this section, we present some individual motion models, combined motion models, and a global motion model and we give quantitative expressions for each of them. To simplify expressing the criteria that lead to the point tracks T_i , we introduce the assignment matrix $A_k = [a_{ij}^k]$, where the entries a_{ij}^k have the following meaning: $a_{ij}^k = 1$ if and only if measurement x_j^{k+1} is assigned to track head T_i^k and, otherwise, zero. Because

some measurements are false and others are missing, there can be some measurements that are not assigned to a track head (all zeros in a column in A^k) and some track heads that have no measurement assigned to them (all zeros in a row in A^k). Or, more formally:

$$\begin{aligned} \sum_{i=1}^M a_{ij}^k &\leq 1, \quad 1 \leq j \leq m_{k+1}; \\ \sum_{j=1}^{m_{k+1}} a_{ij}^k &\leq 1, \quad 1 \leq i \leq M; \quad a_{ij}^k \in \{0, 1\}. \end{aligned} \quad (1)$$

We use two alternative notations for a correspondence between a measurement and a track head. First, we define α_j^k as:

$$\alpha_j^k = i \Leftrightarrow a_{ij}^k = 1. \quad (2)$$

Second, we use ordered pairs (i, j) to indicate that measurement \mathbf{x}_j^{k+1} has been assigned to track head T_i^k . Z^k then contains all assignment pairs from t_k to t_{k+1} according to:

$$Z^k = \{(i, j) \mid a_{ij}^k = 1\}. \quad (3)$$

Tracks T_i can now be derived from A , which is the concatenation of the assignment matrices A^k . We introduce a deviation matrix $D^k = [c_{ij}^k]$ to denote all individual assignment costs c_{ij}^k between track heads T_i^k and measurements \mathbf{x}_j^{k+1} .

The assignment matrix identifies all correspondences from frame to frame, while the deviation matrix quantifies the deviation from the individual motion track per correspondence. The matrices A^k and D^k both have M rows and m_{k+1} columns. The rows represent the M track heads, T_i^k , and the columns represent the m_{k+1} measurements, \mathbf{x}_j^{k+1} , that have been detected at t_{k+1} .

3.1 Individual Motion Models

We now formulate three individual motion models, together with an expression to compute a *deviation* from the optimal track. The first model uses only one previous measurement to predict the new position. We have indicated the dependence of only one previous measurement by the order of the individual model: $O_{im} = 1$. The other two individual models depend on two measurements and consequently have order $O_{im} = 2$. The following motion criteria coefficients c_{ij}^k are all defined from track head T_i^k to a measurement \mathbf{x}_j^{k+1} .

- **im1.** 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} .

$$c_{ij}^k = \|\mathbf{x}_j^{k+1} - \mathbf{x}_i^k\|, \quad \text{where } 0 \leq c_{ij}^k \leq \sqrt{S_w^2 + S_h^2}. \quad (4)$$

- **im2.** The *smooth motion* model as introduced by Sethi and Jain [22] assumes that the velocity magnitude and

direction change gradually. The smooth motion is formulated quantitatively in the following criterion:

$$\begin{aligned} c_{ij}^k = & 0.1 \left[1 - \frac{(\mathbf{x}_i^k - \mathbf{x}_{\alpha_i^k}^{k-1}) \cdot (\mathbf{x}_j^{k+1} - \mathbf{x}_i^k)}{\|\mathbf{x}_i^k - \mathbf{x}_{\alpha_i^k}^{k-1}\| \|\mathbf{x}_j^{k+1} - \mathbf{x}_i^k\|} \right] \\ & + 0.9 \left[1 - 2 \frac{\sqrt{\|\mathbf{x}_i^k - \mathbf{x}_{\alpha_i^k}^{k-1}\| \|\mathbf{x}_j^{k+1} - \mathbf{x}_i^k\|}}{\|\mathbf{x}_i^k - \mathbf{x}_{\alpha_i^k}^{k-1}\| + \|\mathbf{x}_j^{k+1} - \mathbf{x}_i^k\|} \right], \end{aligned} \quad (5)$$

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

- **im3.** The *proximal uniformity* model by Rangarajan and Shah [19] assumes little motion in addition to constant speed. The deviation is quantified in the following criterion:

$$\begin{aligned} c_{ij}^k = & \frac{\|(\mathbf{x}_i^k - \mathbf{x}_{\alpha_i^k}^{k-1}) - (\mathbf{x}_j^{k+1} - \mathbf{x}_i^k)\|}{\sum_{p=1}^M \sum_{q=1}^{m_{k+1}} \|(\mathbf{x}_p^k - \mathbf{x}_{\alpha_p^k}^{k-1}) - (\mathbf{x}_q^{k+1} - \mathbf{x}_p^k)\|} \\ & + \frac{\|\mathbf{x}_j^{k+1} - \mathbf{x}_i^k\|}{\sum_{p=1}^M \sum_{q=1}^{m_{k+1}} \|\mathbf{x}_q^{k+1} - \mathbf{x}_p^k\|}, \quad \text{where } 0 \leq c_{ij}^k \leq 1. \end{aligned} \quad (6)$$

3.2 Combined Motion Models

Combined motion models serve to resolve correspondence conflicts between two successive frames in case of dense moving point sets, making the individual model errors dependent on each other. Next, we give two combined model criteria C^k as a function of A^k and D^k that are defined at t_k over all established track heads.

- **cm1.** The *average deviation* model. This is a typical combined model which usually is realistic. It accounts for the average deviation from the optimal track according to the individual model [21], [22], [27]. Quantitatively, we use the generalized mean, which has a z parameter, to differentiate between emphasis on large and small deviations from the optimal individual track (see Fig. 2).

$$C^k(A^k, D^k) = \left[\frac{1}{M} \sum_{i=1}^M \sum_{j=1}^{m_{k+1}} a_{ij}^k (c_{ij}^k)^z \right]^{\frac{1}{z}}. \quad (7)$$

- **cm2.** The *average deviation conditioned by competition and alternatives* model is derived from [1], [19]. In this combined model, measurements are assigned to that track head that gives low deviation from the optimal track, while both the other tracks are less attractive for this measurement and the other measurements are less attractive for this track.

$$C^k(A^k, D^k) = \frac{1}{M} \sum_{i=1}^M \sum_{j=1}^{m_{k+1}} (a_{ij}^k c_{ij}^k - w_1 R_a(i) - w_2 R_c(j)), \quad (8)$$

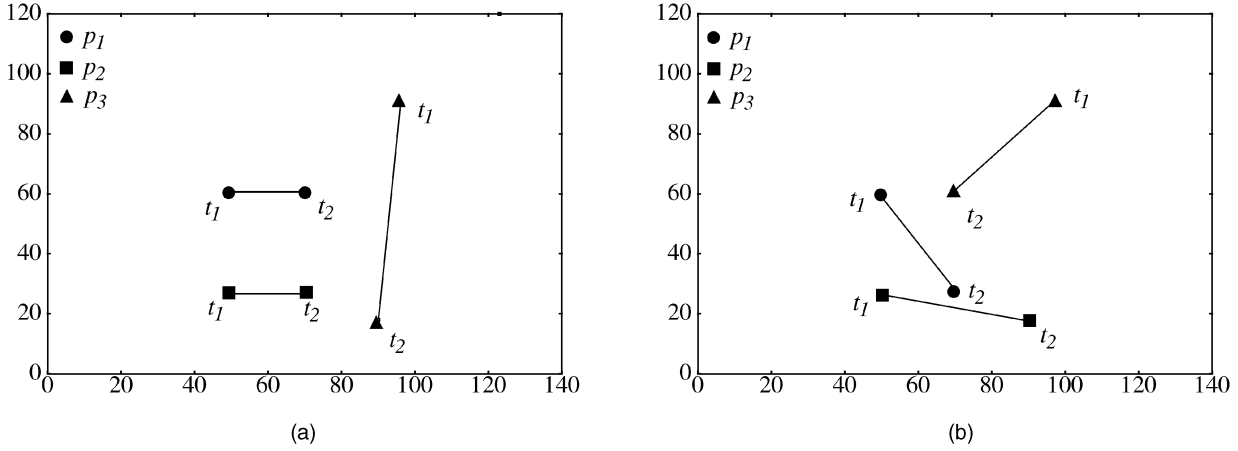


Fig. 2. Three moving points that are matched with **im1** and **cm1** using either (a) $z = 1$ or (b) $z = 2$. As a consequence, larger deviations are penalized more in (b).

where

$$R_a(i) = \frac{1}{m_{k+1} - 1} \sum_{q=1}^{m_{k+1}} (1 - a_{iq}^k) c_{iq}^k; \quad (9)$$

$$R_c(i) = \frac{1}{M - 1} \sum_{p=1}^M (1 - a_{pj}^k) c_{pj}^k$$

$R_a(i)$ represents the average cost of alternatives for T_i^k and $R_c(j)$ the average cost for competitors of x_j^{k+1} .

3.3 Global Motion Model

To find the optimal track set (over all frames) according to a certain combined model, we need to compute the accumulated global motion deviation $S(D)$ as in the following expression:

$$S(D) = \min_{A \in U} \sum_{k=O_{im}}^{n-1} C^k(A^k, D^k), \quad (10)$$

where U is the set of matrices A that satisfy (1).

That is, the overall minimum of averaged combined criterions defines the optimal track set. Because finding this

minimum is computationally expensive, a *greedy matching* is considered in this paper. This means that, instead of finding correspondences over all frames, we establish optimal correspondences between two successive frames, given the state of the individual motion models and the combined model up to that moment. After these suboptimal correspondences have been established, the states of the individual models are adjusted and the next frame is considered. In other words, (10) is approximated by minimizing $C^k(A^k, D^k)$ separately, i.e.:

$$\hat{S}(D) = \sum_{k=O_{im}}^{n-1} C_{min}^k(D^k), \quad \text{where } C_{min}^k(D^k) = \min_{A^k \in U^k} C^k(A^k, D^k). \quad (11)$$

This approximation approach reduces the complexity of the problem considerably, although at the cost of greedy, possibly less plausible, correspondence decisions (see, for example, Fig. 3). In the remainder, we leave out the D and D^k parameter for S , \hat{S} , and C_{min}^k , respectively.

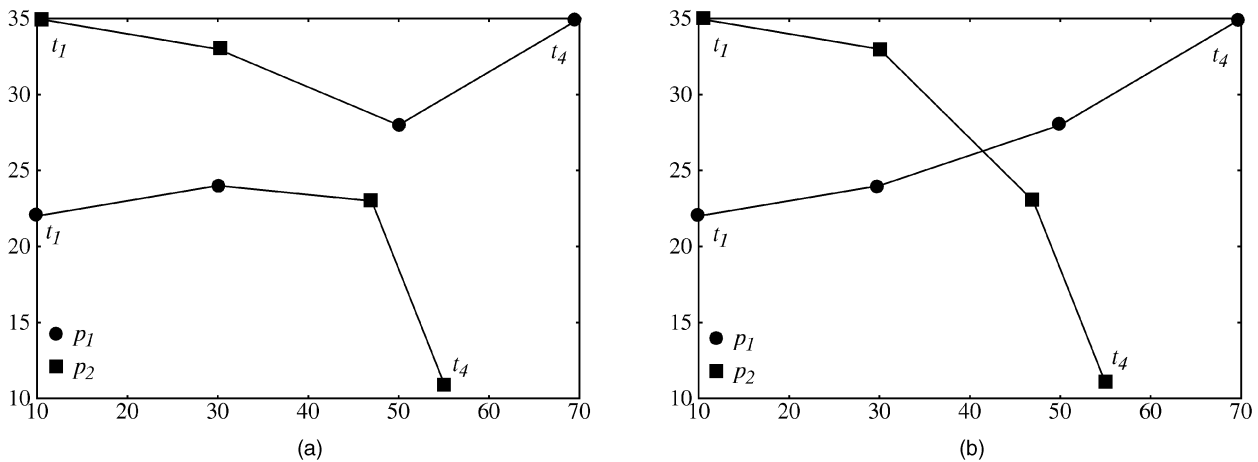


Fig. 3. Two moving points at four time instances. When the smooth motion model (**im2**) with the average deviation model (**cm1**) are assumed, (b) gives a two times lower deviation from the optimal path than (a). However, (a) is decided for when greedy matching is used.

3.4 Model Constraints

The motion models we have described so far allow for any point speed and for any deviation from smoothness. The models only state that those assignments are preferred that have little deviation from the individual model. There are, however, situations in which there is more knowledge available about the point motions, like the minimum speed (d_{min}) and the maximum speed (d_{max}) [1], [12], [14], [21], maximum violation of smoothness (ϕ_{max}) [1], [14], [21], and spatial or temporal adaptive speed and smoothness violation constraints [10]. When imposed on the individual motion models, these constraints enable the recognition of impossible assignments, which can be both qualitatively and computationally beneficial. These constraints can for instance, be implemented by setting the individual criterion to a very high value when some constraint is violated. The strategy (see Section 3.5) that satisfies the models can exploit these constraints more adequately by leaving out of consideration those correspondences that violate the motion constraints.

3.5 Strategy

To find the optimal track set, we compute the global motion deviation. However, we are not interested in the actual value of S , but in the assignment matrix A that results in the minimal global motion deviation. In the next section, we first show how existing algorithms approximate the minimization of C^k and consequently deliver a suboptimal solution A^k . In Section 5, we present an optimal as well as efficient algorithm to find that A^k that minimizes C^k .

4 ALGORITHMS

Having modeled the feature point motion and having described quantitative expressions that can be used to identify the optimal track set, we now review a number of existing algorithms and fit them in our motion framework using our concept of individual and combined motion models. Further, we describe the strategy they use to find the optimal correspondences. Because all algorithms perform greedy matching, their task is to find C_{min}^k .

4.1 S&S Algorithm (im2/cm1/z = 1)

The first algorithm we looked at was originally developed by Sethi and Jain [22]. The original algorithm assumes a fixed number of feature points to be tracked and does not allow for occlusion and detection errors. Here, we describe the adjusted algorithm by Salari and Sethi [21] that partially fixes these shortcomings. The algorithm adopts a smooth motion model for individual motion (**im2**). The combined motion model is an average deviation model (**cm1/z = 1**). To find an optimum of the global motion, the algorithm iteratively exchanges measurements between tracks to minimize the criterion on average.

Initially, the tracks are led through the nearest-neighbor measurements in the sequence. In this stage, conflicts are “resolved” on a first-come-first-served basis. That is, at t_{k+1} , measurements are assigned to the closest track parts T_i^k that have been formed up to t_k to which no point was assigned yet. Consequently, the initialization procedure is a greedy **im1/cm1** approximation.

Then, each iteration step modifies at most two assignment pairs somewhere in the sequence by exchanging the second entry of the pair. The algorithm considers all possible exchanges within the d_{max} range of two track heads in the whole sequence and the exchange that gives the highest gain by decreasing the average criterion deviation is executed. The iteration phase stops when gain can no longer be obtained. The exchange gain between assignment pairs (i, p) and (j, q) (see (3)) is defined in the following way:

$$g_{ij}^k = c_{ip}^k + c_{jq}^k - (c_{iq}^k + c_{jp}^k). \quad (12)$$

To achieve even better tracking results, the algorithm first optimizes correspondences over all frames in forward direction and then (after this iteration phase stops) it optimizes correspondences in backward direction. Only when the optimization process did not change anything in either direction the algorithm stops. This bidirectional optimization process can indeed increase the tracking quality, but, unfortunately, this process is not guaranteed to converge, especially with densely moving points [19].

In contrast with what we said before, this algorithm seems to optimize over the whole sequence. However, when we look carefully at the optimization process within one iteration phase, we see that this is only partially true. As long as the tracks are wrong at the start, exchanges in the remainder of the track will mostly be useless. This is due to the fact that the tracks were initialized using another criterion than the one that is considered in the iteration phase. Consequently, the optimization is only effective at the initial measurements of the tracks. This problem is most severe when the sequence is long and when the difference between the initialization criterion (nearest-neighbor) and the optimization criterion (smooth motion) is large, i.e., with high speeds and high densities. We tested this statement by feeding to the S&S algorithm the example shown in Fig. 3. If we do not optimize in both directions, the S&S algorithm indeed makes greedy correspondences as in Fig. 3a, which supports the statement that S&S is a greedy matching algorithm.

The Salari and Sethi version of this so-called greedy exchange algorithm additionally proposes a way to resolve track continuation, initiation, and termination. They introduce a number of *phantom points* to the set of measurements in each frame. These phantom points serve as replacements of missing measurements, while satisfying local constraints. By imposing the maximum allowed local smoothness criterion and a maximum speed, missed measurements are recognized and filled in with phantom points. Moreover, the constraints also allow the detection of false measurements. Effectively, false measurements are replaced by phantom points if the introduction of a phantom point results in a lower criterion value.

This approach generally works fine except that missing measurements (represented by a phantom point) always have the maximum criterion and displacement. For instance, if point p_i has not been measured at t_k , the algorithm can easily associate a measurement of p_i at t_{k+1} to another point which is within the criterion range ϕ_{max} . It is important to remark that the phantom points only enforce

that the local movement constraints are satisfied, but when a phantom point is put in a track, the track is in fact divided up into two tracks. In other words, this maximum criterion approach solves the correspondence problem *up to* the maximum criterion. Choosing a low maximum criterion leads to many undecided track parts and a higher maximum criterion leads to possibly wrong correspondences. This is where the track initiation/termination and occlusion events become conflicting requirements, as already mentioned in Section 2.

4.2 R&S Algorithm (im3/cm2)

A different approach to the correspondence problem is chosen by Rangarajan and Shah [19]. They have a different combined motion model and do not use an iterative optimization procedure. The R&S algorithm assumes a fixed number of feature points and it allows for temporary occlusion or missing point detections, but not for false detections. It uses the proximal uniformity model (**im3**) as the individual motion model and **cm2** as the combined motion model. This algorithm does not constrain the individual point motion, i.e., it does not have a d_{max} or ϕ_{max} parameter.

To find the minimum of the combined model (8), the authors use a greedy noniterative algorithm. In each step of the algorithm, that particular point x_j^{k+1} is assigned to track head T_i^k that has a low deviation from the optimal motion (low individual deviation) while, on average, all alternative track heads have a larger deviation with respect to x_j^{k+1} and, on average, all other measurements have a worse criterion with respect to T_i^k .

We continue the description of the algorithm in terms that fit the proposed motion framework as established in Section 3. The algorithm selects that assignment pair (i, j) that maximizes $R'_a(i) + R'_c(j)$ among all minimal track head extensions, where $R'_a(i)$ and $R'_c(j)$ are derived from (9) according to:

$$R'_a(i) = \frac{1}{m_{k+1} - 1} \sum_{q=1, q \neq j}^{m_{k+1}} c_{iq}^k, \quad R'_c(j) = \frac{1}{M - 1} \sum_{p=1, p \neq i}^M c_{pj}^k. \quad (13)$$

Then, an optimal assignment pair $g(X_t, X_m)$ is repeatedly selected in the following way:

$$g(X_t, X_m) = \left((i, j) \mid i = \arg \max_{p \in X_t} (R'_a(p) + R'_c(j)), j = \arg \min_{q \in X_m} c_{pq}^k \right), \quad (14)$$

where X_t is the set of track head indices that have not yet been assigned a measurement and X_m is the set of measurement indices that have not yet been assigned to a track head. After an assignment has been found, the track head and measurement are removed from the respective index sets X_t and X_m . The algorithm accumulates the assignment costs and eventually stops when X_t is empty. The criterion computation can be summarized in the recurrence relation as follows:

$$C'(X_t, X_m) = \begin{cases} 0, & \text{if } X_t = \emptyset \\ (c_{ij}^k + C'(X_t - \{i\}, X_m - \{j\})) \mid (i, j) = g(X_t, X_m), & \text{otherwise.} \end{cases} \quad (15)$$

The matching assignment pairs are collected similarly:

$$Z'(X_t, X_m) = \begin{cases} \emptyset, & \text{if } X_t = \emptyset \\ ((i, j) \cup Z'(X_t - \{i\}, X_m - \{j\})) \mid (i, j) = g(X_t, X_m), & \text{otherwise.} \end{cases} \quad (16)$$

Consequently, this strategy results in the following approximation of C_{min}^k :

$$\hat{C}_{min}^k = C'(\{i \mid 1 \leq i \leq M\}, \{j \mid 1 \leq j \leq m_{k+1}\}) \quad (17)$$

and the set of assignment pairs as defined in (3):

$$Z^k = Z'(\{i \mid 1 \leq i \leq M\}, \{j \mid 1 \leq j \leq m_{k+1}\}). \quad (18)$$

Additionally, the algorithm differentiates between two cases: 1) all measurements are present and 2) some measurements are missing, by occlusion or otherwise. In the first case, the algorithm works as described above. Otherwise, because there is a lack of measurements at t_{k+1} , the problem is not which measurement should be assigned to which track head, but which track head should be assigned to which measurement. Then, the assignment strategy is similar to the above. When all track head assignments T_i^k to measurements x_j^{k+1} are found, it is clear for which tracks a measurement is missing. The R&S algorithm directly fills in these points with extrapolated points. The disadvantage of this track continuation scheme becomes apparent when the point occlusion lasts for a number of frames. Direct extrapolation results in a straight extension of the last recognized motion vector, which, in the long term, can deviate much from the true motion track so that recovering becomes increasingly difficult (see the experiments in Section 6.2.4).

4.3 C&V Algorithm (im2/cm2)

The third and last scheme we describe, has been developed by Chetverikov and Verestoy [1]. Their method allows for track initiation, track termination, and occlusion only during two time instances. C&V assume the smooth motion model (**im2**) and **cm2** as the combined motion model. The algorithm extends track heads T_i^k by first collecting all candidate measurements x_j^{k+1} in the circle with radius d_{max} around x_i^k whose criterion does not exceed ϕ_{max} . The candidate measurements are considered in optimal criterion order with respect to the track head. Then, for each measurement, all competing track heads are collected. The candidate measurement will be rejected if it is the best alternative for any of the competing track heads. When there are no candidates left, the track head will not be connected. Remaining unconnected track parts, caused by occlusion or otherwise, are handled in a postprocessing step, which we leave out of the discussion.

This scheme does not maximize the cost of the alternatives (i.e., $w_1 = 0$ in (8)) and track heads are only then considered as competitors if they are within the d_{max} as well as ϕ_{max} range. Moreover, their cost is not averaged as in (8):

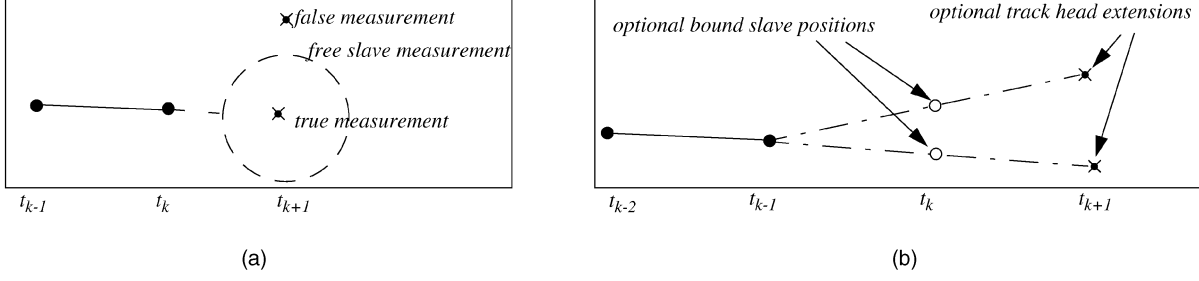


Fig. 4. (a) Shows a true measurement, a false measurement, and a free slave measurement at t_{k+1} . The slave measurement is on the border of the dotted circle. (b) Shows possible bound slave measurement positions related to possible track head extensions.

Any competitor that fits a measurement *best* prevents that the measurement is assigned to T_i^k .

The basics of this algorithm can be summarized as follows:² Let $X_a(i)$ be the set of alternative track head extensions for track head T_i^k as defined below:

$$X_a(i) = \left\{ j \in X_m \mid i \in X_c(j), \forall p \in X_c(j) \left(j = \arg \min_{q \in X_m} c_{pq}^k \implies p = i \right) \right\}, \quad (19)$$

where each measurement \mathbf{x}_j^{k+1} has a set of competing track heads $X_c(j)$ according to:

$$X_c(j) = \left\{ i \in X_t \mid \|\mathbf{x}_j^{k+1} - \mathbf{x}_i^k\| < d_{max}, c_{ij}^k < \phi_{max} \right\}. \quad (20)$$

The algorithm selects a measurement from $X_a(i)$ for a track head from X_t according to:

$$g(X_t, X_m) = \left((i, j) \mid i \in X_t, j = \arg \min_{q \in X_a(i)} c_{iq}^k \right). \quad (21)$$

Substituted into (15), (16), (17), and (18), this leads to the minimal combined criterion approximation and corresponding set of assignment pairs Z^k .

The advantage of this scheme is that the d_{max} parameter is exploited very efficiently. With low point densities, there is usually just one candidate point and there are no competing track heads for that point. However, higher point densities or large d_{max} values can reveal the inadequacy of the strategy to find the minimal combined motion model deviation. Because the deviation is not averaged over competitors and alternatives, greedy assignment decisions are the result.

5 OPTIMAL ALGORITHM TO MINIMIZE C^k

In the previous section, we saw that known algorithms adopt a suboptimal search strategy to minimize C^k . In this section, we propose an algorithm that finds the minimum of the combined motion models efficiently. To this end, we use the Hungarian algorithm, which efficiently finds the solution of the classical assignment problem [13]. Danchick and Newman [6] first used it in a similar context, to find hypotheses for the Multiple Hypothesis Tracker. In general, the algorithm minimizes the following expression:

2. We describe the algorithm in our own terms, assuming a fixed number of points and verification depth = 2, see [1] for details.

$$C = \sum_{i=1}^m \sum_{j=1}^m a_{ij} w_{ij} \quad (22)$$

subject to:

$$\begin{aligned} \sum_{i=1}^m a_{ij} &= 1, \quad 1 \leq j \leq m; \\ \sum_{j=1}^m a_{ij} &= 1, \quad 1 \leq i \leq m; \quad a_{ij} \in \{0, 1\}. \end{aligned}$$

It typically finds the minimal cost assignment, which can be represented in a weighted bipartite graph consisting of two sets of vertices, X and Y . The m vertices from X are connected to all m vertices of Y with weighted edges w_{ij} . The algorithm then assigns every vertex from X to a separate vertex in Y in such a way that the overall cost is minimized.

In order to be able to apply the Hungarian algorithm and to handle detection errors and occlusion, we prepare the measurement data such that the problem becomes squared. We propose handling the false detection problem by introducing *false tracks*, as proposed earlier in [27]. False tracks do not have to adhere to any motion criterion so that measurements that do not fit the motion model of any true track will be moved to these false tracks. By associating a maximum cost deviation (ϕ_{max}) to assignments to false tracks, we even recognize false measurements if other measurements are missing.

We propose implementing track continuation by introducing the concept of *slave measurements* (Fig. 4a), similar to the interpolation scheme in [27]. Slave measurements have two states: free and bound. A *free* slave is not willing to be assigned to a track. Consequently, it has a maximum deviation cost from the optimal motion track. Free slave measurements serve similar goals as the phantom points in [21]. A slave measurement is *bound* when it has been assigned to a track, despite its high deviation. Bound slaves imitate the movements of their neighboring measurements. Their position is calculated by interpolating the positions of preceding and succeeding measurements in the track established so far (Fig. 4b). The interpolated positions enable more accurate calculation of the motion criterion. In this way, we retain as much motion information as possible and we are therefore able to make plausible correspondences. Additionally, we assign high cost ($> \phi_{max}$) to correspondences that have d_{max} exceeded. This ensures that, in such cases, a slave measurement is preferred over a measurement that does not fit the model constraints.

5.1 Greedy Optimal Assignment (GOA) Tracker: Formal Description

To properly handle missing and false measurements, we extend the assignment matrices A^k . That is, we want to be able to assign false measurements to false tracks and slave measurements to true track heads that have no measurement at t_{k+1} . Since all measurements can be false and all track heads may miss their measurement, we add m_{k+1} rows to allow for m_{k+1} false tracks and we add M columns to allow for M slave measurements, resulting in the definition of the square matrix A_λ^k (resembling the dummy rows and columns in the validation matrix as proposed in [9]).

The size of the individual criterion matrix is adjusted similarly. The entries in the m_{k+1} extra rows and in the M extra columns all equal the maximum cost resulting in cost matrix D_λ^k .

Having defined these square matrices, the linear assignment problem can be solved for one frame after the other, assuming that the correspondences between the first two frames are given (in case $O_{im} > 1$) to be able to compute the initial velocity vector.

In order to calculate the motion criterion, the individual motion models with $O_{im} = 2$ need the vector $(x_i^k - x_{\alpha_i^k}^{k-1})$ and all need $(x_j^{k+1} - x_i^k)$. If either of $x_{\alpha_i^k}^{k-1}$ or x_i^k is a slave measurement, we estimate these vectors by scanning back in T_i^k to collect two true measurements in the nearest past being $x_{\alpha_i^{k-p}}^p$ and $x_{\alpha_i^{k-q}}^q$, respectively, with $1 \leq p < q \leq k$ and α_i^{k-q} means $k-q$ times recursive application of α_i^k . Consequently, the vector estimates are defined as follows:

$$\begin{aligned} x_i^k - x_{\alpha_i^k}^{k-1} &= \frac{x_{\alpha_i^{k-q}}^q - x_{\alpha_i^{k-p}}^p}{q - p}; \\ x_j^{k+1} - x_i^k &= \frac{x_i^{k+1} - x_{\alpha_i^{k-q}}^q}{k + 1 - q}. \end{aligned} \quad (23)$$

Having obtained these velocity vector estimates, we can now compute the individual motion criteria c_{ij}^k . We transform the criterion matrix to a bipartite graph and prune all edges with weights that exceed ϕ_{max} . Then, to satisfy the combined motion model, we adjust the edge weights w_{ij} as defined below.

- **cm1** average deviation model:

$$w_{ij} = \left(c_{ij}^k \right)^z. \quad (24)$$

- **cm2** average deviation conditioned by competition and alternatives, using (13);

$$w_{ij} = c_{ij}^k - w_1 R'_a(i) - w_2 R'_c(j). \quad (25)$$

As mentioned before, the actual value of the minimized C^k is not important. Therefore, in **cm1**, the $1/z$ power can be ignored because the $1/z$ power function is monotonic increasing.

Algorithm

1. Starting with $k = O_{im}$ compute all costs c_{ij}^k in the cost matrix D_λ^k as follows:
 - a. true tracks to true measurements, i.e., $1 \leq i \leq M$, $1 \leq j \leq m_{k+1}$:
If the maximum speed (d_{max}) constraint is violated, then $c_{ij}^k = \phi_{max} + \epsilon$.
Otherwise, c_{ij}^k is according to the individual motion model.
 - b. all other entries: $c_{ij}^k = \phi_{max}$
2. Construct a bipartite graph based on the criterion matrix D_λ^k .
3. Prune all edges that have weights exceeding ϕ_{max} .
4. Adjust the edge weights according to the combined motion model in (24) and (25).
5. Apply the Hungarian algorithm to this graph which results in the minimal cost assignment. The resulting edges (assignment pairs) correspond to an output A_λ^k , from which the first M rows and m_{k+1} columns represent the assignment matrix A^k .
6. Increase k ; if $k < n$, go to 1, otherwise, done.

6 PERFORMANCE EVALUATION

To evaluate the performance of the different algorithms, we compared them qualitatively and quantitatively. In Section 6.1, we start by looking at their correspondence quality by using a specially constructed example that (also) tests the algorithm's track continuation capabilities. Then, in Section 6.2, we explore the sensitivity of the algorithms to some problem parameters like point density and the total number of points, and algorithm parameters like d_{max} . In all experiments in this section, the correspondences between the first two frames are known and passed to all algorithms (even to those that are capable of self-initialization, avoiding favoring one of the methods).

6.1 Constructed Example

The carefully constructed example shows two crossing feature points with a missing measurement at t_4 for the first and at t_5 for the second point (see Fig. 5a). The difficulty of this data set is that, in two consecutive frames, a measurement is missing, but for different points. With all algorithms, we used the smooth motion model (**im2**). For algorithms that have a ϕ_{max} parameter, we varied its value from 0.05 to 1 (lower values do not allow the initial motion of p_2). Further, we fixed the d_{max} value to 20.

6.1.1 S&S Results

The S&S algorithm either leads to wrong correspondences or to disconnected track parts. We used two different settings of ϕ_{max} to show the shortcomings of S&S. First, with a high ϕ_{max} ($0.1 \leq \phi_{max} \leq 1$), the algorithm makes wrong correspondences (Fig. 5b). When assigning measurements to track heads T_i^k , the algorithm prefers track heads that have a true measurement at t_k over track heads that have a phantom point at t_k . Of course, the motion criterion for that true measurement assignment may not exceed the maximum criterion. On the other hand, if ϕ_{max} is lower (e.g., 0.05), the algorithm separates four track parts, while

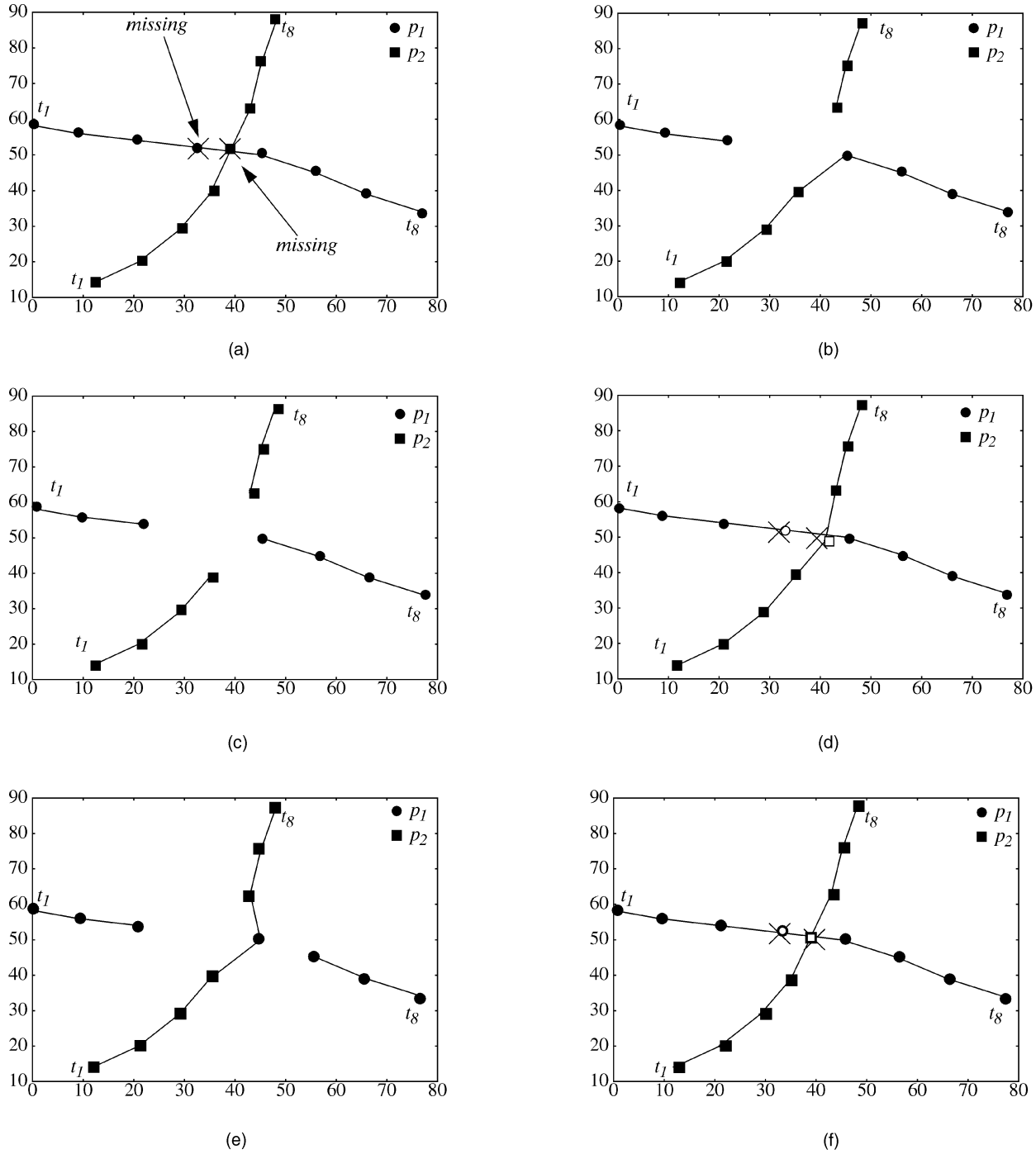


Fig. 5. (a) Two input measurements at eight time instances. At t_4 , a measurement for point p_1 is missing and, at t_5 , a measurement of point p_2 is missing. The figures show the results of (b) S&S using $0.1 \leq \phi_{max} \leq 1$, (c) S&S using $\phi_{max} = 0.05$, (d) R&S, (e) C&V, and (f) the GOA tracker, respectively. In the figures, the estimated points are shown as nonfilled boxes and crosses indicate the true positions of the missing points.

correspondences between the track parts have to be made afterward (see Fig. 5c).

6.1.2 R&S Results

The R&S algorithm, which has no parameters, chooses the right correspondence when one measurement lacks at t_4 . Then, it estimates the missing measurement by extrapolation and continues with the next frame. With point extrapolation for one frame only, the deviation is limited. In the next frame (t_5), the situation is similar to the previous

frame. The algorithm connects the single present measurement to the right track head and extrapolates the missing measurement. The processing of the last three frames is straightforward (see Fig. 5d).

6.1.3 C&V Results

At t_4 , C&V assigns the single measurement to the right track head (T_2^3). Then, at t_5 , only one track head T_2^4 remains to which the measurement can be assigned. If it wouldn't fit because the distance was too great, this measurement could

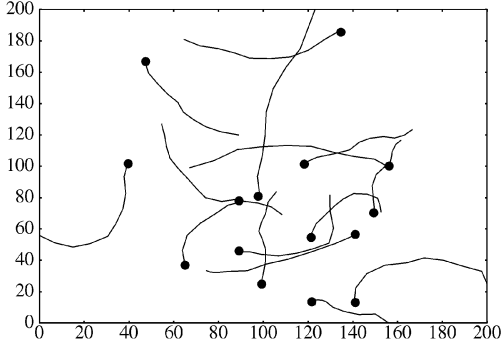


Fig. 6. Example PSMG data set with 15 points during eight time steps.

start a new track. Since it is not too far away, the only point at t_5 is also assigned to T_2 . The two track parts that belong to p_1 are not connected in the postprocessing step (Fig. 5e).

6.1.4 GOA Tracker Results

When the algorithm proposed in this paper is applied to this data set with the smooth motion and average deviation model, all correspondences are made correctly. Moreover, the algorithm interpolates the missing measurements better than R&S and, hence, forms the most plausible tracks (see Fig. 5f).

6.2 Performance with Generated Data

In this section, we describe the tests we did to evaluate various aspects of the described algorithms. To this end, we used a data set generator that is able of creating data sets of uncorrelated random point tracks of various densities and speeds. Among the described algorithms, only the R&S algorithm does not exploit the d_{max} parameter to improve quality and efficiency. For the experiments, we added the d_{max} parameter to R&S (now called R&S*)³ similar as in the GOA tracker. Then, we tuned all algorithms to find the optimal d_{max} setting for each of them and used that setting in all experiments. For C&V, R&S*, and the GOA tracker the true maximum is optimal and for S&S a very high value, $d_{max} = 50$, is optimal. In Section 6.2.6, we consider the sensitivity of the algorithms for the d_{max} parameter setting. We did not test the ϕ_{max} sensitivity, because it constrains the motion similarly. Other experiments evaluate the performance for increasingly difficult data sets, an increasing number of missing point detections, and the efficiency of the algorithms.

For the generation of the uncorrelated tracks, we used the data set generator called Point Set Motion Generator (PSMG) according to [26] (see example in Fig. 6). Because this data generator model allows feature points to enter and leave the 2D scene, which we do not consider in this paper, we modified the model to prevent this by replacing invalid tracks until all tracks are valid. The PSMG has the following parameters (defaults in brackets):

1. number of feature point tracks ($M = 50$),
2. number of frames per point track ($n = 8$),
3. size ($S_w = S_h$) of the square space ($S = 100$).

3. When d_{max} is very high, then R&S* behaves like the original R&S, i.e., unconstrained speed.

4. uniform distributions for both dimensions of initial point positions between 0 and S ,
5. normal distribution for the magnitude of the initial point velocity vector: $v_i^0 = N(\mu_{v_0} = 5, \sigma_{v_0} = 0.5)$,
6. uniform distribution for the angle of the initial velocity vector, between 0 and 2π ,
7. normal distribution for the *update* of the velocity vector magnitude v_i^k , from t_k to t_{k+1} : $v_i^{k+1} = N(v_i^k, \sigma_{v_u} = 0.2)$,
8. normal distribution for the *update* of the velocity vector angle β_i^k from t_k to t_{k+1} : $\beta_i^{k+1} = N(\beta_i^k, \sigma_{\beta_u} = 0.2)$,
9. probability of occlusion ($p_o = 0$, i.e., no occlusion).

A number of different measures have been proposed to quantify the quality of performance, like the *distortion measure* [19] and the *link-based error* and *track-based error* [26]. We use the track-based error as in [26], which is defined as follows:

$$E_{track} = 1 - \frac{T_{correct}}{T_{total}}, \quad (26)$$

where T_{total} is the total number of true tracks and $T_{correct}$ is the number of *completely* correct tracks.

Some remarks about the experiments: First, in all cases, the shown results are an average of 100 runs. We did not incorporate significance levels because the minimal possible track error depends on the actual presented data, hence, the appropriateness of the individual motion model. Nevertheless, the ranking and relative quality were, for each experiment, the same as illustrated in the figures. Second, in this section, we ran the S&S algorithm only with a forward optimization loop because, otherwise, the algorithm would not converge (see also Section 4).

6.2.1 Tuning Individual and Combined Models

To find an optimal combination of individual and combined motion models, we assume that the individual models and combined models are independent. In order to find the best individual model for the PSMG generated data, we ran experiments with the individual models **im1**, **im2**, and **im3**, together with the combined model **cm1**/ $z = 1$ implemented in the GOA tracker. In Fig. 7a, we show the results of this experiment. Clearly, the model **im2** fits this generated data set best.

In order to identify the best combined model for this data set, we ran tests with **cm1**/ $z = 1$, **cm1**/ $z = 2$, **cm2**/ $w_1 = w_2 = 0.3$, and **cm2**/ $w_1 = w_2 = 0.2$, as shown in Fig. 7b. We chose w_1 equal to w_2 because we want to express that the lack of alternatives is equally important as the absence of competing track heads. With even lower w_1 and w_2 values, **cm2** becomes better until it finally equals **cm1**/ $z = 1$ when $w_1 = w_2 = 0$. From these tests, we conclude that the smooth motion model ($\phi_{max} = 0.2$) with average deviation model ($z = 1$) is the best combined modeling for PSMG data. Hence, we used these models in the remaining experiments, if possible. That is, only the GOA tracker allows for *combined model* settings and can be adjusted in that sense.

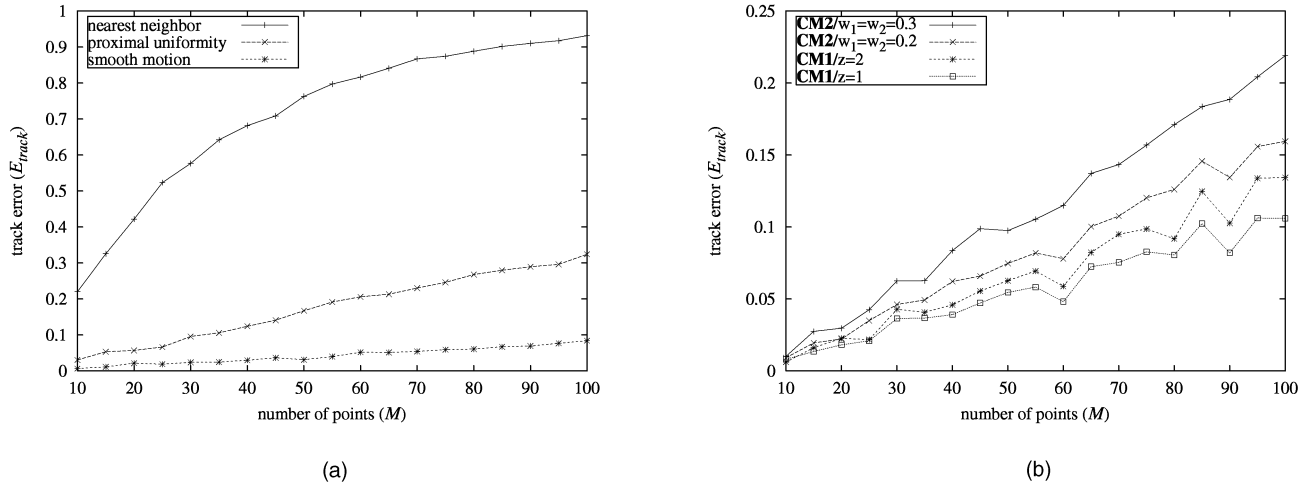


Fig. 7. (a) Track error of the GOA tracker with the average deviation model (cm1), in combination with the nearest-neighbor, smooth motion, or proximal uniformity model. (b) Track error of the GOA tracker with the smooth motion model in combination with cm1 and cm2.

6.2.2 Variable Density Performance

To show how the algorithms perform with an increasing number of conflicts, we applied them to several data sets with an increasing point density. To this end, we generated the data in a fixed sized 2D space and varied the number of point tracks. In Fig. 8a, we display the results of all algorithms. The figure clearly shows that the GOA tracker performs best.

6.2.3 Variable Velocity Performance

Another experiment to test the tracking performance of the algorithms is varying the mean velocity and keeping the number of points constant. In order to have reasonable speed variances with all mean velocities, we scaled both σ_{v_0} and σ_{v_u} with the mean values according to $\sigma_{v_0} = 0.1\mu_{v_0}$ and $\sigma_{v_u} = 0.04\mu_{v_0}$. In addition, we enlarged the space in which the point tracks are generated to $S = 200$ to prevent having mainly diagonal tracks allowed. The ranking of the algorithms is similar to the variable density experiment and, again, the GOA tracker performs better than all other schemes (see Fig. 8b).

6.2.4 Track Continuation Performance

In this experiment, we compared the track continuation performance of the R&S extrapolation scheme and the slave measurements interpolation, as proposed in this paper. We left out the other two algorithms because S&S does not really handle track continuation and C&V only allows very limited occlusion. In order to properly compare the track extrapolation and the slave measurements interpolation, we implemented them both in the GOA tracker. We tested the track continuation performance in a variable occlusion experiment, with $0 \leq p_o \leq 1$. In Fig. 9a, we display the track error results of the GOA tracker with both track continuation schemes with either 50 or 100 points.

As illustrated in this figure, the slave measurements approach proposed in this paper clearly achieves better track continuation results than the track extrapolation scheme as proposed by Rangarajan and Shah [19]. The difference is larger with higher probability of occlusion (p_o), because then there will be occlusion more often during a number of consecutive frames, in which case the difference between interpolation and extrapolation becomes apparent.

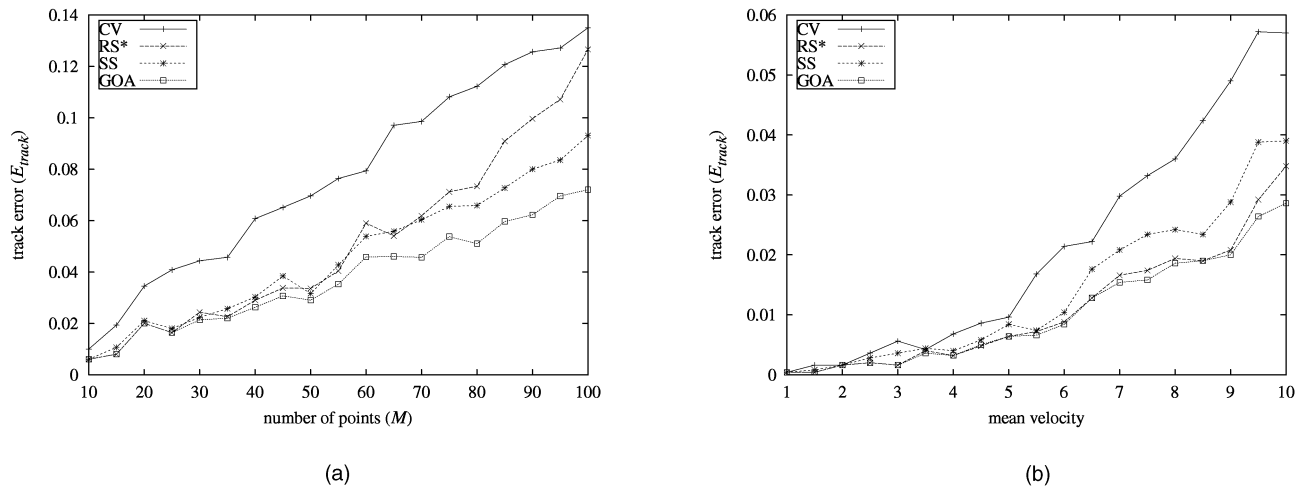


Fig. 8. (a) Results of the algorithms applied to increasingly dense point sets. (b) Track error as a function of the mean velocity.

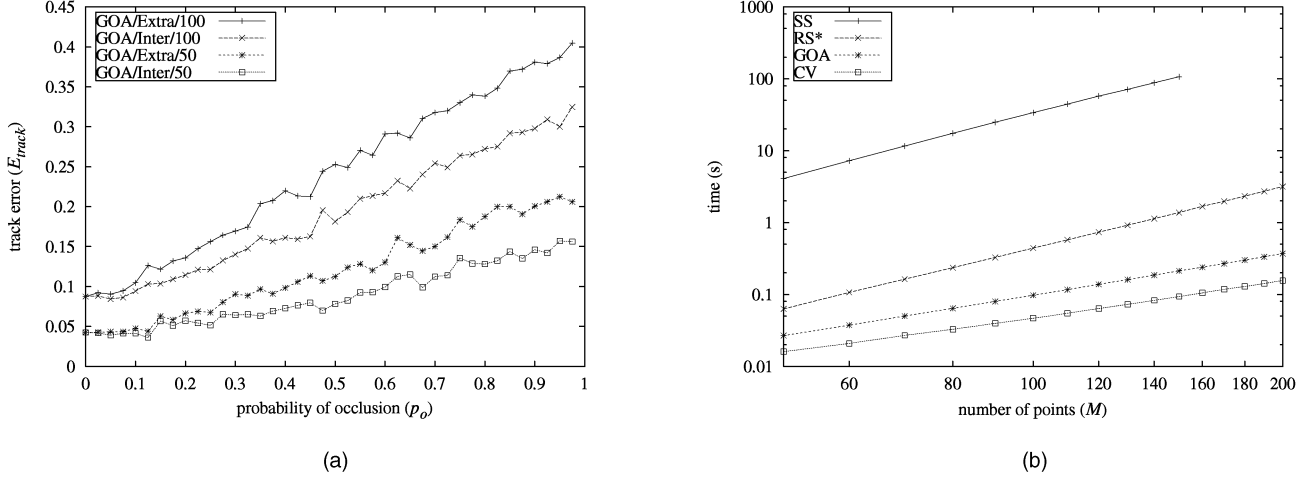


Fig. 9. (a) Track error of the GOA tracker with either slave interpolation (Inter) or the R&S extrapolation scheme (Extra) in a variable occlusion experiment with 50 or 100 points. (b) Illustration of the efficiency of the algorithms in a variable volume experiment.

6.2.5 Variable Volume Performance

This test is directed toward measuring the computational efficiency of the different algorithms. We keep the point density constant while increasing the number of point tracks (and, thus, enlarging the size of the 2D space proportionally). Consequently, the correspondence problem remains equally difficult, but the problem size grows. In Fig. 9b, we show the results with logarithmically scaled axes. The figure shows that, with optimal d_{max} , C&V is the fastest. Further, the computation time of the algorithms is widely divergent, but all algorithms have polynomial complexity. We list the polynomial orders in the summary of the experiments in Section 6.3.

6.2.6 Sensitivity for d_{max} Parameter Setting

As mentioned, so far all algorithms used the tuned and optimal settings of the d_{max} parameter. In this sensitivity experiment, we show the importance of a priori knowledge about a reasonable value for this parameter. To this end, we varied the d_{max} parameter from the known true value up to a high upper limit, $d_{max} = 50$ (lower values than the true

maximum speed are clearly not sensible). Fig. 10a clearly shows that both S&S and R&S* are most sensitive to variations in this parameter. Remarkably, S&S performs better when d_{max} is set far too high. We expect that the poor initialization, together with the exchange optimization, causes this effect because every point exchange must obey the d_{max} constraint. Both C&V and the GOA tracker are hardly sensitive to d_{max} variations (which implies that they do not take advantage of it either). Computationally, especially the C&V algorithm is hampered by an incorrect or ignorant d_{max} value, as Fig. 10b illustrates. Consequently, the GOA tracker is the fastest when d_{max} is over five times the true maximum speed.

6.3 Summary of Experiments

In conclusion, for tracking a fixed number of points, the GOA tracker is qualitatively the best algorithm among the algorithms we presented, according to its track continuation handling in the first test and its performance in all PSMG experiments. Moreover, it is hardly sensitive to the d_{max} parameter setting. S&S performs only slightly worse

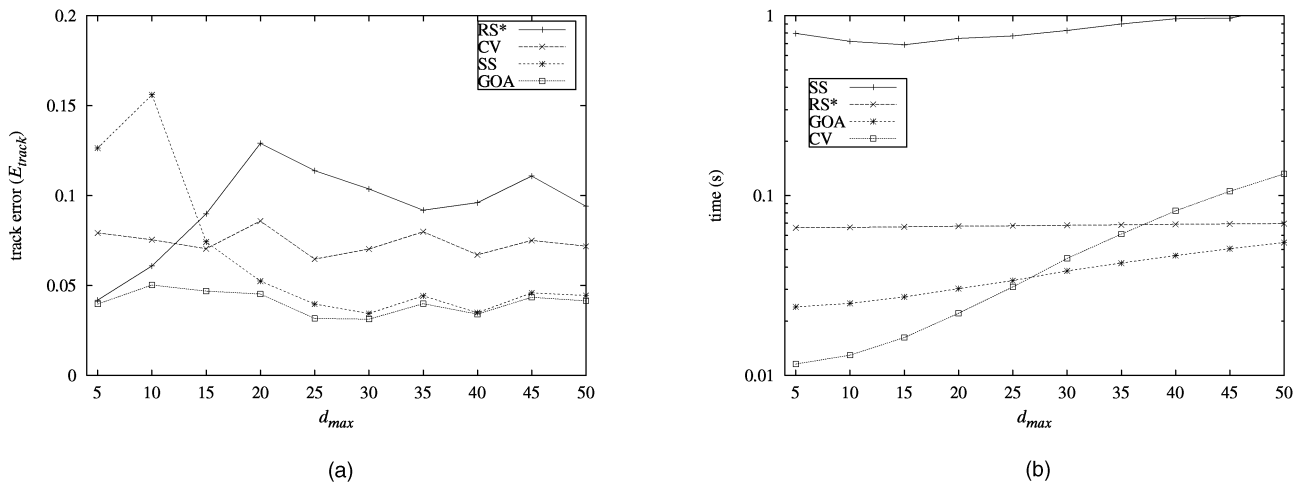


Fig. 10. Illustrates the sensitivity of the algorithms to d_{max} variations. (a) Shows the track error performance and (b) shows the computational performance.

TABLE 1
Summary of the PSMG Experiments

Algorithm	variable density ($M=100$) E_{track}	variable velocity ($\mu_{v_0}=10$) E_{track}	variable volume ($M=100$) time (s)	polynomial order a in $O(M^a)$
GOA	0.07	0.029	0.097	1.9
S&S	0.09	0.039	34	3.0
C&V	0.14	0.057	0.047	1.6
R&S	0.13	0.035	0.44	2.8

when we used the optimal d_{\max} setting ($d_{\max} = 50$), but it is an order of magnitude slower than the GOA tracker. Moreover S&S did not perform well on the specially constructed example nor does it give interpolated positions of the missed points. The version of R&S*, with added d_{\max} parameter and modified individual model, is efficient and qualitatively good as long as it has an accurate estimate of d_{\max} . The sensitivity experiment shows that R&S* performs worst of all if this value is not known (or not used as in the original R&S implementation). With (near) optimal maximum velocity setting, C&V is the fastest. If this optimal value is not known (which is usually the case), then the efficiency of C&V degrades rapidly. We should also note that, in our experiments, S&S performed consistently better than C&V, which does not agree with the results reported in [26]. This is probably because, in [26], a different d_{\max} setting for S&S is used for which we showed in Section 6.2.6 that the S&S algorithm is quite sensitive. This implies that S&S cannot exploit the d_{\max} parameter effectively to handle missing and spurious measurements. Finally, the variable occlusion experiment clearly showed that the slave measurements implement track continuation better than the point extrapolation scheme [19]. In Table 1, we summarize the PSMG experiments. The last column shows the polynomial order of complexity of the algorithms as derived from the variable volume experiment.

7 ALGORITHM EXTENSION WITH SELF-INITIALIZATION

In the problem statement in Section 2, the correspondences between the first two frames were assumed to be known. In this section, we generalize the problem by lifting this restriction and elaborating on how self-initialization is incorporated in the GOA tracker.

Two algorithms we discussed have an *integrated* way of automatically initializing the point tracks. That is, both S&S and C&V only use the measurement positions for the initialization. R&S, on the other hand, use additional information, i.e., the optical flow field, which is computed between the first two frames. We advocate the integrated approach because it is more generally applicable and it allows for optimizing the initial correspondences using a number of frames, as we proposed in the *global motion model* in Section 3. Here, we propose to extend the GOA tracker with features of the S&S algorithm. After that, we demonstrate the appropriateness of this extension and

again analyze the parameter sensitivity of the algorithms that support self-initialization.

7.1 Up-Down Greedy Optimal Assignment Tracker (GOA/up-down)

The S&S algorithm has a number of shortcomings, of which its computational performance has been shown to be the most apparent. Also, as mentioned, we deliberately left out the bidirectional optimization which quite often does not converge. However, for self-initialization, the bidirectional optimization is essential.

We propose to modify the GOA tracker in the spirit of [21] and [22] by initializing the correspondences between the first two frames using the described optimal algorithm to minimize C^k with **im1/cm1**. After these correspondences are made, we continue the optimization of the remaining frames (*up*) in the normal way and additionally optimize the same frames backward (*down*). Further forward and backward optimization proved to be useless because the optimization process already converged. The reason for this fast convergence is that both the initial correspondences and the optimization scheme have been improved considerably compared to S&S.

7.2 Self-Initialization Experiments

To test the performance of the algorithms that are capable of self-initializing the tracks, together with the just-described extended GOA/up-down tracker, we did another variable density experiment and a sensitivity experiment using the PSMG track generator. The individual models need not be tuned again because the parameter settings of the PSMG are the same as in Section 6.2. This time, we left out S&S because of serious convergence problems with their bidirectional optimization scheme, which is essential for self-initialization. R&S does not implement self-initialization using only point measurements, so it cannot be applied within these experiments.

Although we did not discuss statistical motion correspondence techniques in detail in this paper, we included the multiple hypothesis tracker (MHT) as described and implemented by Cox and Hingorani [3] in this experiment in order to see how it relates to nonstatistical greedy matching algorithms. We should note that this MHT implementation is not the most efficient (for improvements, see, e.g., [15]) though qualitatively equivalent to the state of the art of the statistical motion correspondence algorithms.

7.2.1 Variable Density Experiment

For this experiment, we tuned the algorithms optimally for the given data sets. That is, both C&V and GOA/up-down use the true d_{\max} . The (eight) parameters of the MHT (like the Kalman filter and Mahalanobis distances) were tuned with a genetic algorithm for which we used the track error as fitness function.

Actually, the only difference with the variable density experiment in Section 6.2.2 is that, here, the initial correspondences are not given. Fig. 11a shows the performance of the algorithms. Clearly, GOA/up-down performs best and, remarkably, almost as well as when the initial correspondences were given. The performance of the MHT

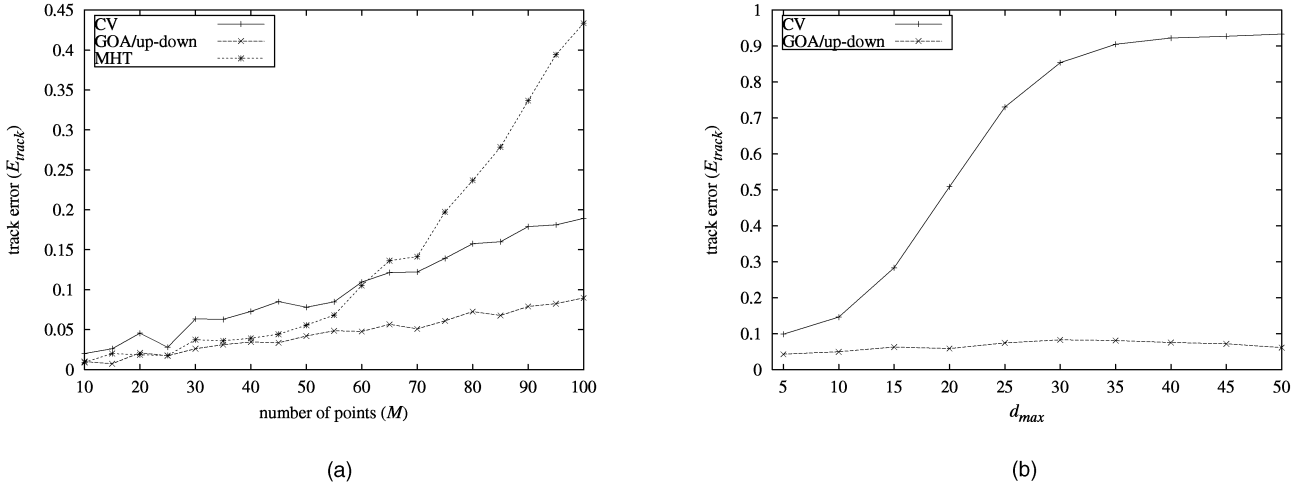


Fig. 11. (a) Shows the track error as a function of the number of points in a variable density experiment with self-initialization and (b) shows the track error as a function of the d_{max} setting.

is similar to the GOA tracker until it seriously degrades, when the number of points exceeds 50, see Fig. 11a. This can be explained from the fact that the parameters for the MHT were trained for (only) 50 points. We did not include more points because the training was already very time consuming (> 2 days on a Silicon Graphics Onyx II). It is, however, striking to see that the GOA tracker also performs consistently better than the MHT even with less than 50 points, although the latter optimizes over several frames. Among others, this may be caused by the effective self-initialization scheme of the GOA tracker. The up-down scheme can be said to optimize the initial correspondences over the whole sequence when optimizing up. Then, the remaining correspondences are established in the down phase.

7.2.2 Sensitivity Experiment

When the correspondences for the initial frames are not given, we expect the algorithms to be more sensitive to the d_{max} setting. Namely, when the initial velocity is unconstrained, the greedy matching algorithms easily make implausible initial choices from which they cannot recover. To study this behavior, we did another sensitivity experiment for C&V and the GOA tracker and, additionally, an experiment to test the sensitivity of the MHT. We studied the MHT separately because it has different parameters (and no d_{max}). First, Fig. 11b indeed shows that for, C&V, a good estimate of d_{max} is essential. GOA/up-down, however, hardly suffers from lack of a priori knowledge concerning d_{max} , which is partially because the *global cost* was optimized for the initial frames. Moreover, the up-down optimization scheme can no longer be considered purely greedy because correspondences are reconsidered in backward direction. Since both algorithms were computationally influenced similarly as when the initial correspondences were given, we did not include the figure here. We have to mention, however, that the computation time of C&V increased even 10 times faster (11 sec. when $d_{max} = 50$) because, in this experiment, the number of alternatives becomes much higher in the first frame. As a consequence,

the GOA tracker was already the fastest when d_{max} was set over three times the true maximum speed.

In order to fairly test the sensitivity of the MHT and to show the results for all parameters in the same figure, we tested the performance in the range from 1/5 of the optimal setting to 10 times the optimal setting of all essential parameters (10 runs per setting). Consequently, the results in Fig. 12a can easily be compared in relation to Fig. 11a in which 5 (d_{max}) is also optimal. The figure clearly shows that there is a small parameter range in which the performance is (sub)optimal. Especially, increasing or decreasing the Mahalanobis distance or the initial state variance parameter with 1/5th results in a performance penalty of roughly a factor two. Also, the computation time increases dramatically if the parameters are not properly set, as Fig. 12b shows. We plotted the names of the essential parameters in the figures, but refer to [3], [20] for a complete description.

8 REAL DATA EXPERIMENT: TRACKING SEEDS ON A ROTATING DISH

Our final experiment is based on real image data. In this experiment, we put 80 black seeds on a white dish and rotated the dish with more or less constant angular velocity, which implies the use of the smooth motion model ($\mathbf{im2}/\phi_{max} = 0.1$). The scene was recorded with a 25 Hz progressive scan camera using 4 ms shutter speed, resulting in a 10 image video sequence with very little motion blur.⁴ The segmentation of the images was, consequently, rather straightforward, i.e., in all 10 images all 80 seeds were detected and there were no false measurements. There was a large difference in speed of the seeds ranging from 1 pixel/s in the center to 42 pixel/s at the outer dish positions. As in Section 7, we tested only those algorithms that have self-initialization capability and, again, we included the MHT. Clearly, in contrast with Section 7, in this experiment, the point motion is strongly dependent. Since all algorithms are hampered equally, this experiment

4. The rotating dish sequence is available for download at <http://www-ict.its.tudelft.nl/tracking/datasets/sequences/rotdish80.tgz>.

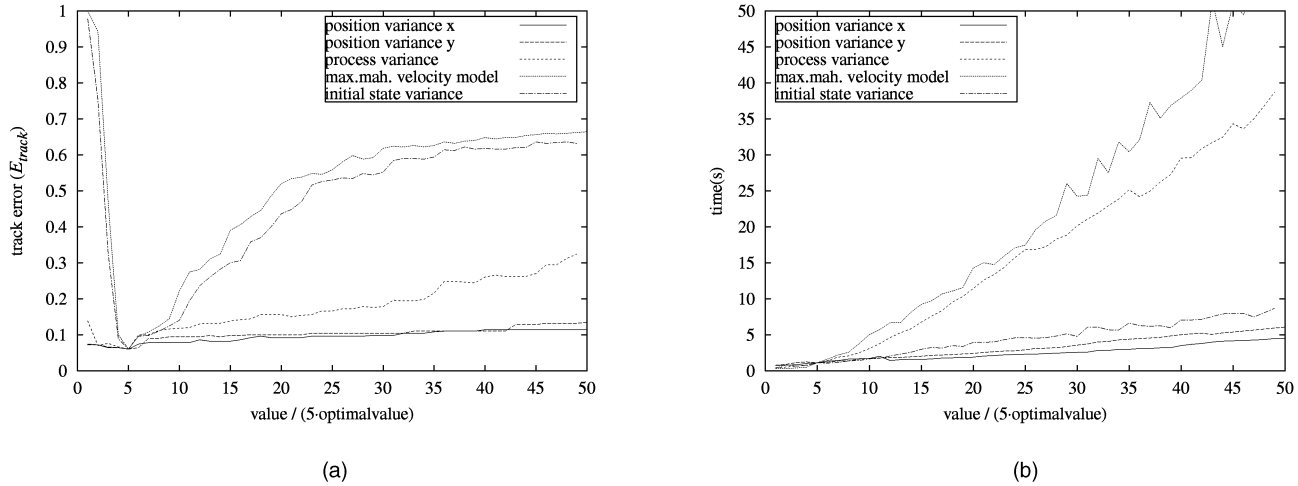


Fig. 12. Parameter sensitivity experiment for the MHT. (a) Shows the track error as function of parameter variations and (b) shows the computation time.

actually tests the general applicability of the algorithms.⁵ To be able to run the MHT properly, we tuned its main parameters by applying a genetic algorithm (the ground truth was established by manual inspection). For this experiment, we also added the S&S algorithm, because this time it converged consistently, that is, with different d_{max} settings.

Fig. 13 shows the resulting tracks overlaid on the first image of the sequence. Only the GOA/up-down tracker was able to find all the true seed tracks, while the d_{max} setting did not influence the results. Even GOA/up (not down!) was able to track the 80 seeds correctly over all 10 frames, regardless of the d_{max} value. Not surprisingly, the C&V algorithm that already proved to be sensitive to d_{max} suffers severely from the divergent seed speeds. The S&S algorithm, which is also sensitive to d_{max} , again makes fewer errors when d_{max} is relaxed. In general, the behavior of S&S turned out to depend greatly on the d_{max} and ϕ_{max} settings. Although the MHT is extensively tuned and it optimizes over several frames simultaneously, the MHT still makes a few errors. Besides, the MHT is substantially slower than the other algorithms.

9 DISCUSSION

Throughout this paper, we introduced a framework for motion modeling and we presented the greedy optimal assignment (GOA) tracker that we extended with self-initialization. In this section, we discuss some potential other extensions and improvements.

Although the tracking of a variable number of points conflicts with occlusion handling, it is certainly a feature that should be considered as an extension to the GOA tracker. Among the described algorithms we have seen two ways to approach this conflict of requirements, either by actually not implementing track continuation (S&S) or by only allowing occlusion during a very limited number of frames (C&V). First, the GOA tracker can support track

initiation and termination by replacing the slave measurements with the phantom points as in S&S. Alternatively, the GOA scheme can be incorporated in the C&V algorithm. The idea is that, at each time instance, the GOA scheme is applied first to find corresponding measurements for all point tracks that have been established so far. Then, the original C&V scheme links the remaining measurements if possible. As a result, the tracking features of C&V still apply and its performance increases.⁶

Further, to deal more effectively with the underlying physical motion, the order of the individual motion models could be increased, e.g., by modeling point acceleration. Clearly, extending the scope of the individual models implies difficulties for the model initialization and the track continuation capabilities.

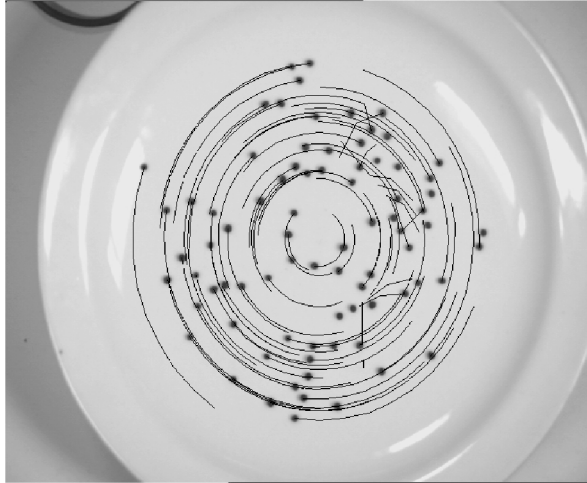
Finally, the scope over which the global matching is approximated can be extended. In this paper, we approximated $S(D)$ in a greedy sense, i.e., we only minimized the combined model over two successive frames. We already illustrated in Fig. 3 that extending the scope for this minimization would yield more plausible tracking results. Extending the scope, however, implies that we need to cope with an increasingly complex problem, to which the efficient Hungarian algorithm as such cannot be applied anymore.

10 CONCLUSION

In this paper, we showed an adequate way to model the motion correspondence problem of tracking a fixed number of feature points in a nonstatistical way. By fitting existing algorithms in this motion framework, we showed which approximations these algorithms make. An approximation that all described algorithms have in common is that they greedily match measurements to tracks. For this approximation, we proposed an *optimal* algorithm, the Greedy Optimal Assignment (GOA) tracker, which obviously

5. One could, of course, argue that for this data set a rotational individual model or polar coordinates for the measurement positions would fit better.

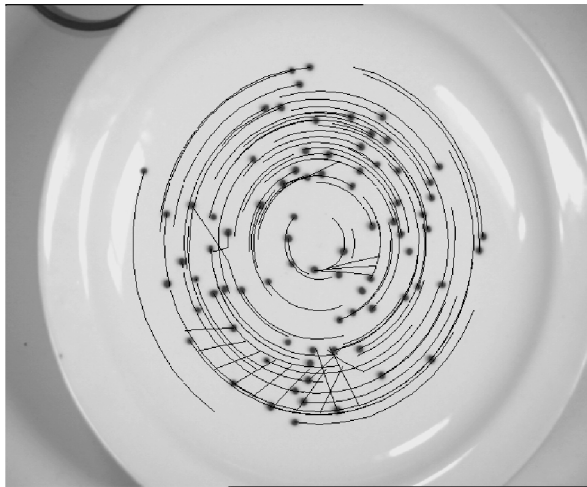
6. We have already implemented this idea, but we did not include it in the experiments for the sake of clarity. With a fixed number of points, its performance indeed rated in between GOA/up-down and the original C&V algorithm.



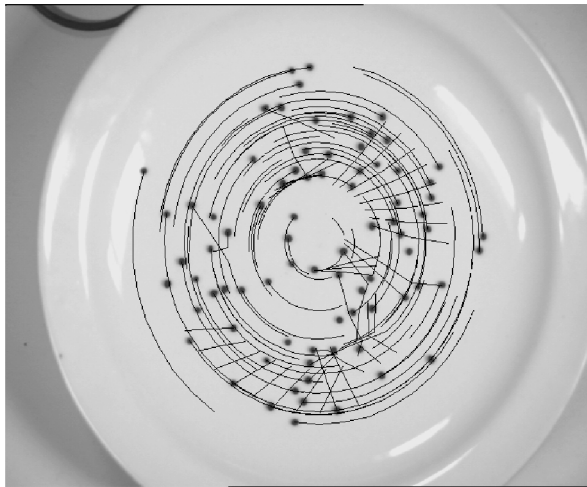
(a)



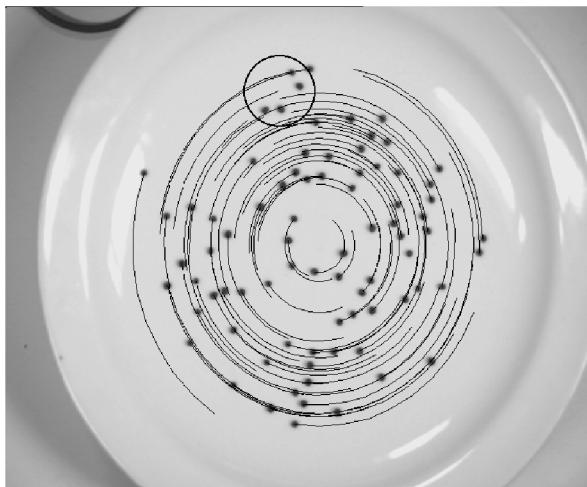
(b)



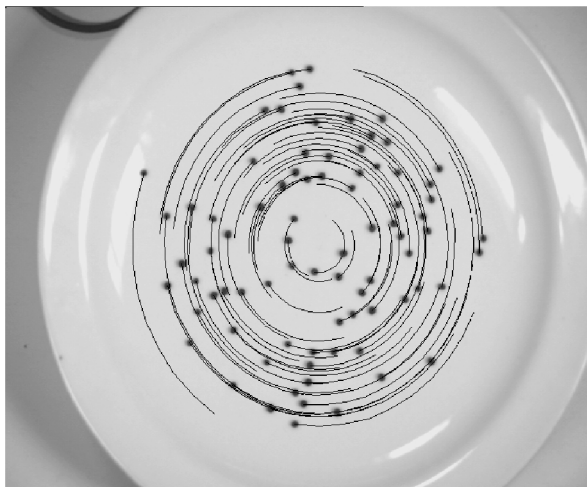
(c)



(d)



(e)



(f)

Fig. 13. Results of applying the self-initializing algorithms to the rotating dish sequence, consisting of 10 frames with each 80 seeds; true $d_{max} = 42$ pixels/sec. (a) S&S: $d_{max} = 42$ p/s : 25 errors, 7.4 sec. (b) S&S: $d_{max} = 50$ p/s : 18 errors, 13 sec. (c) C&V: $d_{max} = 42$ p/s : 18 errors, 44 msec. (d) C&V: $d_{max} = 50$ p/s : 31 errors, 58 msec. (e) MHT: 3 errors, 86 msec. (f) GOA/up-down: 0 errors, 90(160) msec.

qualitatively outperforms all other algorithms. The way the proposed algorithm handles detection errors and occlusion turned out to be effective and more accurate than the other described algorithms. Moreover, the experiments show clearly that its computational performance is among the fastest. Also, the self-initializing version of the GOA tracker turned out to be adequate and hardly sensitive to the maximum speed constraint (d_{max}) setting. Briefly, for the tracking of a fixed number of feature points, the proposed tracker has proven to be efficient and qualitatively best.

Among the described algorithms, the R&S algorithm is completely surpassed because it operates under the same conditions, while the GOA tracker outperforms R&S both qualitatively and computationally. The S&S algorithm, which does not support track continuation, is computationally very demanding. The major drawbacks of the C&V algorithm are its relatively poor performance, especially with respect to the initialization, its restricted track continuation capability, and its sensitivity to the d_{max} setting. Still, S&S and C&V may be considered because both support the tracking of a variable number of points and C&V can be very fast. In the previous section, we indicated how their performance can be improved by incorporating GOA features in these algorithms. In a number of experiments, we included the statistical multiple hypothesis tracker. Even though the MHT optimizes over several frames, which makes it computationally demanding, it turned out that it does not perform better than the GOA tracker. Possible causes are the effective initialization of the GOA tracker and the fact that the MHT models the tracking of a varying number of points, although we set the respective probabilities as to inform that the number of points is fixed. Most importantly, the MHT has quite a few parameters for which the tuning proved to be far from trivial.

In conclusion, the proposed qualitative motion framework has proven to be an adequate modeling of the motion correspondence problem. As such, it reveals a number of possibilities to achieve qualitative improvements, ranging from more specialized individual models to $S(D)$ approximations with an extended temporal scope.

ACKNOWLEDGMENTS

This work was supported by the foundation for Applied Sciences (STW). The authors would like to thank Dr. Dmitry Chetverikov for the discussions on the details of his tracking algorithm and the anonymous reviewers for their comments and suggestions.

REFERENCES

- [1] D. Chetverikov and J. Verestoy, "Feature Point Tracking for Incomplete Trajectories," *Computing*, vol. 62, pp. 321-338, 1999.
- [2] I.J. Cox, "A Review of Statistical Data Association Techniques for Motion Correspondence," *Int'l J. Computer Vision*, vol. 10, no. 1, pp. 53-66, 1993.
- [3] 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 Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 138-150, Feb. 1996.
- [4] I.J. Cox and M.L. Miller, "On Finding Ranked Assignments with Applications to Multitarget Tracking and Motion Correspondence," *IEEE Trans. Aerospace and Electronic Systems*, vol. 32, no. 1, pp. 486-489, Jan. 1995.
- [5] I.J. Cox, M.L. Miller, R. Danchick, and G.E. Newnam, "A Comparison of Two Algorithms for Determining Ranked Assignments with Application to Multitarget Tracking and Motion Correspondence," *IEEE Trans. Aerospace and Electronic Systems*, vol. 33, no. 1, pp. 295-300, Jan. 1997.
- [6] R. Danchick and G.E. Newnam, "A Fast Method for Finding the Exact n-Best Hypotheses for Multitarget Tracking," *IEEE Trans. Aerospace and Electronic Systems*, vol. 29, no. 2, pp. 555-560, Apr. 1993.
- [7] S. Deb, K.R. Pattipati, and Y. Bar-Shalom, "A New Algorithm for the Generalized Multidimensional Assignment Problem," *Proc. IEEE Int'l Conf. Systems, Man and Cybernetics; Emergent Innovations in Information Transfer Processing and Decision Making*, pp. 249-254, 1992.
- [8] S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom, "A Generalized S-D Assignment algorithm for Multisensor-Multitarget State Estimation," *IEEE Trans. Aerospace and Electronic Systems*, vol. 33, no. 2, pp. 523-538, 1997.
- [9] T.E. Fortmann, Y. Bar-Shalom, and M. Sheffe, "Sonar Tracking of Multiple Targets Using Joint Probabilistic Data Association," *IEEE J. Oceanic Eng.*, vol. 8, no. 3, pp. 173-184, July 1983.
- [10] K.I. Hodges, "Adaptive Constraints for Feature Tracking," *Monthly Weather Rev.*, vol. 127, pp. 1362-1373, 1998.
- [11] B.K.P. Horn and B.G. Schunck, "Determining Optical Flow," *Artificial Intelligence*, vol. 17, pp. 185-203, 1981.
- [12] V.S.S. Hwang, "Tracking Feature Points in Time-Varying Images Using an Opportunistic Selection Approach," *Pattern Recognition*, vol. 22, no. 3, pp. 247-256, 1989.
- [13] H.W. Kuhn, "The Hungarian Method for Solving the Assignment Problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83-97, 1955.
- [14] R. Mehrotra, "Establishing Motion-Based Feature Point Correspondence," *Pattern Recognition*, vol. 31, no. 2, pp. 23-30, 1998.
- [15] M.L. Miller, H.S. Stone, and I.J. Cox, "Optimizing Murty's Ranked Assignment Method," *IEEE Trans. Aerospace and Electronic Systems*, vol. 33, no. 3, pp. 851-861, 1997.
- [16] V. Nagarajan, M.R. Chidambara, and R.N. Sharma, "Combinatorial Problems in Multitarget Tracking—A Comprehensive Solution," *IEE Proc.*, vol. 134, no. 1, pp. 113-118, Feb. 1987.
- [17] A. B. Poore, "Multidimensional Assignments and Multitarget Tracking," *Proc. Partitioning Data Sets, DIMACS Workshop*, pp. 169-196, 1995.
- [18] A.B. Poore and X. Yan, "Data Association in Multiframe Processing," *Proc. Second Int'l Conf. Information Fusion*, vol. II, pp. 1037-1044, 1999.
- [19] K. Rangarajan and M. Sha, "Establishing Motion Correspondence," *CVGIP: Image Understanding*, vol. 24, no. 6, pp. 56-73, July 1991.
- [20] D.B. Reid, "An Algorithm for Tracking Multiple Targets," *IEEE Trans. Automatic Control*, vol. 24, no. 6, pp. 843-854, Dec. 1979.
- [21] V. Salari and I.K. Sethi, "Feature Point Correspondence in the Presence of Occlusion," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 97-91, Jan. 1990.
- [22] I.K. Sethi and R. Jain, "Finding Trajectories of Feature Points in a Monocular Image Sequence," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 1, pp. 56-73, Jan. 1987.
- [23] P.J. Shea and A.B. Poore, "Computational Experiences with Hot Starts for a Moving Window Implementation of Track Maintenance," *Proc. SPIE: Int'l Soc. Optical Eng.*, vol. 3373, pp. 428-439, 1998.
- [24] R.Y. Tsai and T.S. Huang, "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surface," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 1, pp. 13-26, Jan. 1984.
- [25] S. Ullman, *The Interpretation of Visual Motion*. Cambridge, Mass.: MIT Press, 1979.
- [26] J. Verestoy and D. Chetverikov, "Experimental Comparative Evaluation of Feature Point Tracking Algorithms," *Proc. Workshop Evaluation and Validation of Computer Vision Algorithms*, pp. 183-194, 2000.
- [27] C.J. Veenman, E.H. Hendriks, and M.J.T. Reinders, "A Fast and Robust Point Tracking Algorithm," *Proc. IEEE Int'l Conf. on Image Processing*, vol. 3, pp. 653-657, Oct. 1998.



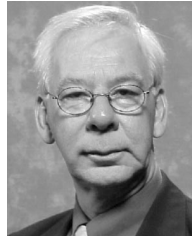
Cor J. Veenman received the BEng degree in electrical engineering from the HTS, Hilversum, The Netherlands, in 1987, and the MSc degree in computer science from the Free University, Amsterdam, The Netherlands, in 1996. From 1991 to 1997, he worked as a software developer in multimedia applications and taught programming courses. In 1997, he started as a research assistant at the Delft University of Technology, Delft, The Netherlands. His re-

search interests are in optimization and artificial intelligence and in applying related techniques in pattern recognition, computer vision, and image segmentation.



Marcel J.T. Reinders received the MSc degree in applied physics and the PhD degree in electrical engineering from Delft University of Technology, The Netherlands, in 1990 and 1995, respectively. Currently, he is an associate professor in the Information and Communication Theory Group, Mediamatics Department of the Faculty of Information Technology and Systems at Delft University of Technology. His research interests are in computer vision, supervised and

unsupervised pattern recognition, and machine learning. Recently, he became interested in the rapidly growing field of bioinformatics that uses machine learning techniques to study biological phenomena, such as the behavior of gene expressions or protein mapping.



Eric Backer received the MSc and the PhD degrees from the Delft University of Technology, The Netherlands, in 1969 and 1978, respectively. He was appointed a full professor at Delft University of Technology in 1982 and was a visiting professor at Michigan State University in 1978 and 1982 and at the University of South Florida in 1993. His current research interests include pattern recognition, image processing, and machine intelligence. Dr. Backer is the

managing editor of *Pattern Recognition Letters* and was the general cochairman of the 11th International Conference on Pattern Recognition. He is a member of the IEEE.