International Journal of Computer Vision, 10:1, 53–66 (1993)


# A Review of Statistical Data Association Techniques for Motion Correspondence

INGEMAR J. COX
*NEC Research Institute, 4 Independence Way, Princeton, NJ 08540*

Received November 9, 1992.

**Abstract**
Motion correspondence is a fundamental problem in computer vision and many other disciplines. This article describes statistical data association techniques originally developed in the context of target tracking and surveillance and now beginning to be used in dynamic motion analysis by the computer vision community. The Mahalanobis distance measure is first introduced before discussing the limitations of nearest neighbor algorithms. Then, the track-splitting, joint likelihood, multiple hypothesis algorithms are described, each method solving an increasingly more complicated optimization. Real-time constraints may prohibit the application of these optimal methods. The suboptimal joint probabilistic data association algorithm is therefore described. The advantages, limitations, and relationships between the approaches are discussed.

## 1 Introduction

Recently there has been significant interest in the analysis of image sequences for purposes of estimating camera motion, 3-D scene geometry, and optical flow, for example. Such analysis usually requires knowledge of frame-to-frame correspondences, that is, do two measurements taken at different times originate from the same geometric feature? This is the motion correspondence problem of computer vision.

There are a number of reasons why motion correspondence is hard. Usually predictions are first made as to the expected locations of the current set of objects of interest. These predictions are then matched to actual measurements. At this stage, ambiguities may arise. Predicitons may not be supported by measurements—have these objects ceased to exist or were they simply occluded? There may be unexpected measurements—do these measurements originate from newly visible objects or are they spurious readings from noisy sensors? More than one measurement may match a predicted feature—which measurement is the correct one and what is the origin of the other measurements? Or a single measurement may match to more than one feature—which feature should the measurement be assigned to? Resolving these ambiguities is the essence of motion correspondence.

Correspondence problems occur in a variety of diverse domains, for example, in psychology, where it is called perceptual grouping [25], in biological [13] and computer vision, robotics, particle physics [18], molecular dynamics [18] and target tracking where it is referred to as the data association problem.[1] The target tracking and surveillance community has extensively studied the motion correspondence problem [5] and a number of statistical data association techniques have been developed. These algorithms are now receiving wider attention, especially within the computer vision community; we review here several of these techniques.

A first step toward the assignment of measurements to features is to estimate the likelihood of a measurement originating from a specific geometric feature. Section 2 briefly summarizes one such measure, the Mahalanobis distance, and discusses how measurements can be matched to predicted geometric features using a statistical validation test. The distance metric immediately suggests a solution to the correspondence problem based on assigning measurements to their closest geometric features. This nearest-neighbor approach is discussed in section 3.

Nearest-neighbor algorithms usually perform badly since there is always a finite chance that the closest measurement is incorrect. However, this possibility is ignored since assignment decisions are based solely on the current image frame. More accurate decisions can be made by postponing decisions and examining the information from several frames. A conceptually straightfoward approach is the track-splitting filter of section 4 in which a *track tree* is formed, each branch denoting a different assignment of a measurement to a feature.

The track-splitting filter may assign the same measurement to more than one geometric feature. This is not physically realistic. More reasonable, is that a measurement originates from only a single source feature. This disjointedness constraint is satisfied by the joint-likelihood algorithm described in section 5.

As the image sequence progresses, some existing geometric features will exit the field of view and new geometric features will become visible. And all the while, spurious measurements will occasionally be reported by noisy sensors. Since the number of perceptually relevant geometric features can change with time as objects enter and/or leave the camera's field of view, an algorithm is needed that is capable of initiating and terminating geometric features. These capabilities together with the disjoint constraints are satisfied by the recursive multiple-hypothesis algorithm of section 6.

Unfortunately, the track-splitting, joint-likelihood, and multiple-hypothesis algorithms all have exponential complexity; and, while heuristics can be used to constrain the search space, large memory and computational resource may still be required. For some applications, these requirements cannot be met, because of real-time and/or cost constraints. Suboptimal approximations can be developed that have the advantage of fixed finite memory and computations. One such algorithm, the joint-probabilistic data-association filter, is described in section 7. Finally, the advantages and disadvantages of all the algorithms are discussed in section 8.

## 2 The Mahalanobis Distance

Consider the case in which $n$ geometric features are being tracked and $n$ measurements are found in the next image frame. In principle, any measurement vector might have originated from any geometric feature and there are $n^2$ possible combinations of assignments. In practice, some measurements are more likely to originate from one track than another. A distance measure is therefore needed that quantifies this likelihood; the smaller the distance between a measurement and its predicted value, the more likely it is to have originated from it.

If measurement $\mathbf{z}_i(k)$ at time $k$ is normally distributed about its predicted value $\hat{\mathbf{z}}(k \mid k-1)$, then a common distance measure is the Mahalanobis distance [15, 28].[2] For Gaussian or normally distributed random variables, the probability density function is given by

$$\begin{aligned} N[\mathbf{z}_i] &\triangleq N[\mathbf{z}_i;\ \hat{\mathbf{z}},\ \mathbf{S}] \\ &= (2\pi)^{-d/2}|\mathbf{S}(k)|^{-1/2} \\ &\quad \times \exp\left(-\frac{1}{2}\{[\mathbf{z}_i(k) - \hat{\mathbf{z}}(k|k-1)]' \right. \\ &\quad \left. \times \mathbf{S}^{-1}(k)[\mathbf{z}_i(k) - \hat{\mathbf{z}}(k|k-1)]\}\right) \end{aligned} \tag{1}$$

where $\mathbf{S}(k)$ represents the covariance of the error estimate $[\mathbf{z}_i(k) - \hat{\mathbf{z}}(k|k-1)]$ and $d$ is the dimension of the measurement vector $\mathbf{z}(k)$. Contours of constant probability density are defined by

$$\begin{aligned} N[\mathbf{z}_i] &= (2\pi)^{-d/2}|\mathbf{S}(k)|^{-1/2} \\ &\quad \times \exp\left(-\frac{1}{2}\{[\mathbf{z}_i(k) - \hat{\mathbf{z}}(k|k-1)]' \right. \\ &\quad \left. \times \mathbf{S}^{-1}(k)[\mathbf{z}_i(k) - \hat{\mathbf{z}}(k|k-1)]\}\right) = c \end{aligned} \tag{2}$$

or equivalently by

$$\begin{aligned} &[\mathbf{z}_i(k) - \hat{\mathbf{z}}(k|k-1)]^T\mathbf{S}(k)^{-1}[\mathbf{z}_i(k) - \hat{\mathbf{z}}(k|k-1)] \\ &= \mathbf{v}_i^T(k)\mathbf{S}(k)^{-1}\mathbf{v}_i(k) \\ &= \gamma \end{aligned} \tag{3}$$

where $\nu_i(k)$ is termed the innovation or error. The left side of equation (3) is called the Mahalanobis distance and can be considered to be a generalization of the Euclidean distance which accounts for the relative uncertainties in the innovation, $\nu_i(k)$.[3] The locus of points of given Mahalanobis distance, is a $d$-dimensional ellipse, where $d$ is the dimension of the measurement vector $\mathbf{z}_i(k)$ and is illustrated in figure 1 for two dimensions.

Repeated measurements of the same geometric feature cluster about the predicted mean value. The distribution of measurements is densest about the predicted value and monotonically decreases with increasing distance from the mean value, as depicted in

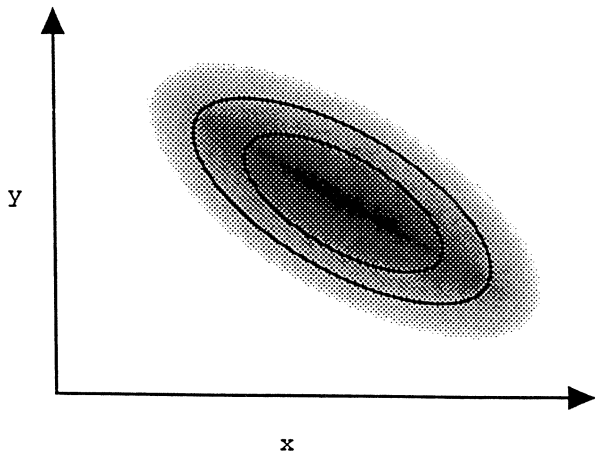


*Fig. 1.* Contours of equal Mahalanobis distance in two dimensions. The shading represents the probability density function of measurements about their predicted value. The darker the region, the more likely a measurement is to be found there.

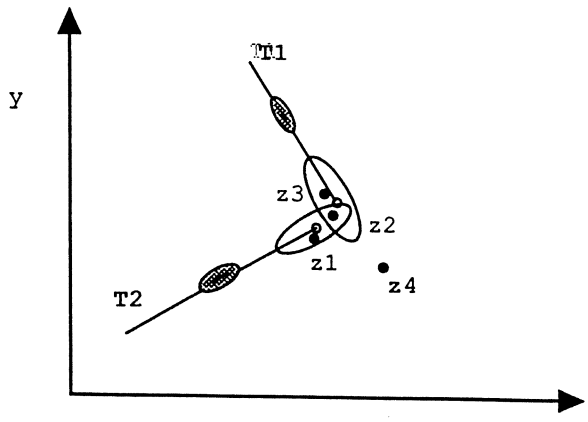


*Fig. 2.* Predicted target locations and elliptical validation regions for a situation with two known geometric features ($T_1$ and $T_2$) and four new measurements $[\mathbf{z}_1(k), \mathbf{z}_2(k), \mathbf{z}_3(k), \mathbf{z}_4(k)]$.

figure 1. What is the probability that the next measurement will have a Mahalanobis distance less than or equal to $\gamma$? That is, that the measurement will fall within the ellipsoidal validation volume $V$ defined by

$$\tilde{V}_k(\gamma) \triangleq \{\mathbf{z} : [z(k) - \hat{\mathbf{z}}(k \mid k-1)]' \mathbf{S}^{-1}(k)[\mathbf{z}(k) - \hat{\mathbf{z}}(k \mid k-1)] \leq \gamma\}. \quad (4)$$

It can be shown that the Mahalanobis distance is chi-squared distributed with number of degrees of freedom equal to the dimension $n_z$ of the measurement vector. The probability that the distance is less than the parameter $\gamma$ can therefore be obtained from $\chi^2$ distribution tables. For example, if the measurement vector is two dimensional, $n_z = 2$, and a validation or search volume is to be established in which there is a 95% probability of finding the measurement, that is, $P(z(k+1) \in \tilde{V}(\gamma)) = 0.95$, then $\gamma$ is set to $\gamma = 5.99$. Conversely, if a measurement fails the inequality test of equation (4) then there is a 5% or less chance that it is associated with the geometric feature.

The validation volume $V$ excludes measurements with low probabilities of assignment, thereby reducing the combinatorics of the correspondence problem. For example, figure 2 depicts a situation in which we have two known geometric features ($T_1$ and $T_2$). The ellipses depict the validation volume around each feature. Four new measurements $(\mathbf{z}_1(k), \ldots, \mathbf{z}_4(k))$ are obtained at time $k$. Ambiguity arises here because the measurement $\mathbf{z}_2(k)$ falls inside the validation volume of both features. However, measurement $\mathbf{z}_4(k)$ can be ignored for the purposes of tracking features $T_1$ and $T_2$, since it falls outside the validation regions of both features. The validation volume also provides an attention focusing capability; only the volume $V$ associated with each track need be examined for measurements.

The validation of $m$ measurements to $n$ geometric features can take $O(mn)$ time. Surprisingly, this gating procedure can become the principal computational bottleneck in tracking large numbers of targets (geometric features) [32] because even if only one measurement validates to each geometric feature, each measurement must still be checked against all other features. For scalar-valued measurements, one can imagine first sorting the predictions and then performing a binary search to locate geometric features that validate to a measurement, thereby reducing the time to $O(m \log n)$. The problem is somewhat more difficult for $n_z$ dimensional measurements, but Uhlmann et al. [29, 30, 12] have recently shown how to efficiently implement the validation gate procedure using multidimensional binary trees in $O(mn^{(1-1/n_z)})$. An alternative technique, described by Zhang and Faugeras [32] is to partition the measurement space into *buckets* [23]. A measurement then need only be validated with those geometric features that fall inside buckets in the vicinity of the measurement. The key ideas are that, on average, the number of geometric features intersecting a bucket is much smaller than the total number of features and the computation of the buckets is linear in the number of geometric features. Finally Orr et al. [22] suggest calculating an approximation to the Mahalanobis distance, actually a lower bound, that avoids the computationally expensive matrix inversion. Only if this lower bound exceeds the threshold $\gamma$ does the actually Mahalanobis distance need to be computed.

In an ideal environment, only a single measurement would validate to each of the geometric features and the data association problem would be solved. This simple solution has been applied to the data association problem, notably in the work of Ayache and Faugeras [1]. However, later work by Deriche and Faugeras [14] revealed that simple validation was not sufficient. In fact, multiple measurements will, from time to time, fall inside the validation volume of a geometric feature in all but the most benign environments. When multiple measurements are validated to a geometric feature, a decision must be taken as to which of the measurements to assign to the track. An obvious solution is assign each measurement to its closest track based on the Mahalanobis distance. This nearest neighbor solution is discussed next.

## 3 Nearest Neighbor

The simplest suboptimal data-association algorithm is the nearest-neighbor algorithm. This assumes that each measurement originates from the closest corresponding feature, where closest is usually defined using the Mahalanobis distance. The attraction of this algorithm is its simplicity, both conceptually and computationally. Examples of nearest neighbor use include Crowley et al. [11] and Deriche and Faugeras [14].

For nearest neighbor correspondences, there is always a finite chance that the association is incorrect and this can lead to serious effects as noted by Bar-Shalom and Fortmann [5], Zhang and Faugeras [32] and Uhlmann [31]. In particular, if a Kalman filter is used to track a geometric feature, then a misassigned measurement may cause the filter to converge slowly or even fail to converge. Consequently, while the nearest neighbor algorithm is simple to implement, users should beware!

The nearest-neighbor algorithm makes assignment decisions based solely on the current image frame. However, much more information is available by examining subsequent images. Significantly better correspondences can therefore be achieved by postponing the decision process in the hope that future measurements will clarify current ambiguities. The earliest such algorithm to attempt this was the track-splitting filter described next.

## 4 The Track-Splitting Filter

The track-splitting filter was originally proposed by Smith and Buechler [27] and more recently Zhang and Faugeras [32] have used the approach for dynamic motion analysis. Figure 3a, b illustrates the principle involved. A single geometric feature of interest is being tracked, when at time $k = 4$ two measurements are found inside its validation region. Rather than arbitrarily assign the closest measurement to the track, a tree is formed as shown in figure 3b. The two branches denote the alternative assignments of the two measurements to the track. No assignment decision is made at this stage. Instead, decisions are postponed until additional measurements have been gathered to either support or refute earlier assignments. The implicit assumption is that ambiguities at time $k$ are resolved by future measurements.

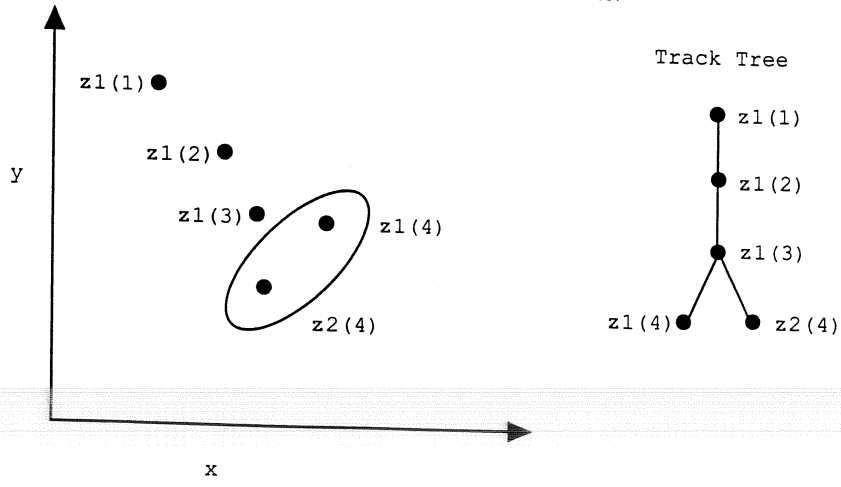


*Fig. 3.* At iteration 4, two measurements validate to the track. The corresponding track tree forms two branches, one representing the assignment of measurement $z_1(4)$ and the other representing the assignment of $z_2(4)$ to the track.

Track trees can very quickly become very large due to simple combinatorial explosion. It is therefore imperative that some measure of the likelihood of an assignment be made so that unlikely hypotheses, that is, branches, can be deleted from the track tree. This likelihood measure is derived next, and is then used to guide the pruning of the track tree which is critical to any practical implementation of the approach. Possible pruning strategies are discussed in section 4.2.

### *4.1 Track Likelihoods*

A track is defined to be a sequence of measurements that are all assumed to originate from the same geometric feature. The correspondence problem is then to assign the right measurements to the right tracks. In each track tree, there are as many tracks as there are paths from leaf nodes to root. Consider one such track, $l$, and denote by $\theta^{k,l}$ the event that the sequence of assignments

$$Z^{k,l} \triangleq \{z_{i_1,l}(1), \ldots, \mathbf{z}_{i_k,l}(k)\} \tag{5}$$

from time 1 to $k$ originate from the same geometric feature $l$.

The likelihood of the event $\theta^{k,l}$ is the joint probability density function[4]

$$\Gamma(\theta^{k,l}) = \prod_{j=1}^{k} p[z_{i_j,l}(j) \mid Z^{j-1}, \theta^{k,l}] \tag{6}$$

where $Z^j$ denotes all measurements up to and including time $j$. If the Gaussian assumptions of section 2 are made, equation (6) becomes

$$\Gamma(\theta^{k,l}) = \prod_{j=1}^{k} (2\pi)^{-d/2} |\mathbf{S}(j)|^{-1/2} \times \exp\left[-\frac{1}{2}\sum_{j=1}^{k} \nu'_{i_j,l}(j)\mathbf{S}^{-1}_{i_j,l}(j)\nu_{i_j,l}(j)\right] \tag{7}$$

The modified log-likelihood function, $\lambda^l$ for track $l$, is then defined as

$$\begin{aligned}\lambda^l(k) &\triangleq -2 \log\left[\frac{\Gamma(\theta^{k,l})}{\Pi_{j=1}^{k}|2\pi\mathbf{S}(j)|^{-1/2}}\right] \\ &= \sum_{j=1}^{k} \nu'_{i_j,l}(j)\mathbf{S}^{-1}_{i_j,l}(j)\nu_{i_j,l}(j) \\ &= \lambda(k-1) + \nu'_{i_k,l}(k)\mathbf{S}^{-1}_{i_k,l}(k)\nu_{i_k,l}(k)\end{aligned} \tag{8}$$

the latter formula allowing the modified log-likelihood to be recursively calculated. The modified log-likelihood is simply the sum of the Mahalanobis distances of all the measurements assigned to track $l$. Thus, rather than assign the measurement that is currently closest as in the nearest neighborhood algorithm, the *sequence* of measurements that minimize the *total* Mahalanobis distance over some interval is selected.

The modified log-likelihood function provides a figure of merit for a particular node in the track tree, some measurement sequences being more likely than others. The cost of calculating the modified log-likelihood is low, especially since the second term is calculated as part of the Mahalanobis test and any associated Kalman filtering. Nevertheless, the track tree may generate a very large number of possible tracks, each one of which needs associated memory and computational resources.

### *4.2 Pruning the Track Tree*

The only way to contain the computation and memory requirements is to delete unlikely nodes and branches from the track tree. There are many possible pruning strategies, including:

1. Deleting unlikely tracks. This may be accomplished by comparing the modified log-likelihood function $\lambda(k)$, which has a $\chi^2$ distribution with number of degrees of freedom $kn_z$ with a threshold $\alpha$ obtained from the $\chi^2$ tables in a manner similar to that for the validation gate procedure of section 2. Care must be taken in applying this test to tracks with long histories because the modified log-likelihood becomes dominated by old measurements and responds very slowly to new ones. This problem can be alleviated by using a sliding window or fading decay term.
2. Merging track nodes. If the state estimates of two or more track nodes are similar, it is likely that they represent the same track. This occurs, for example, when a track splits into two tracks and then both tracks validate identical subsequent measurements.
3. Only keeping the most likely $M$ tracks. For computational reasons it is often desirable to fix an upper limit on the size of a track tree. Consequently, if, after deleting unlikely tracks and merging similar track nodes, the number of leaf nodes exceeds the upper limit $M$, a further level of pruning may be applied.

The merging of track nodes is necessary because tracks may share measurements. This is physically unrealistic. More reasonable, is that a measurement originates from only a single source feature, for example, a single measurement might originate from either a wall or corner feature but not from both. The motion correspondence now becomes one of partitioning measurements into *disjoint* tracks (or sets). Disjointness is also a common constraint in human vision, where in stereo correspondence it is called uniqueness [20] and in motion correspondence it is called the element-integrity principle [13]. It may also be reasonable to assume that a geometric feature gives rise to only a single measurement vector within a time frame. The track-splitting algorithm cannot cope with these constraints and, instead, a joint-likelihood integer-programming algorithm must be used.

## 5 Joint-Likelihood Integer Programming

The joint-likelihood method developed by Morefield [21] produces *disjoint* measurement partitions so that a measurement is assigned to a single unique geometric feature. Morefield's algorithm is a batch process that is formulated as a well-known integer-programming problem [26]. The basic idea is to first group measurements into feasible tracks. This set of tracks is not necessarily disjoint, and a subset of disjoint tracks must therefore be selected. However, there are many such legal sets, and so a search is performed to find the best set of disjoint tracks. A joint-likelihood measure is used to quantify which set of tracks is best.

A set of disjoint tracks, also called an hypothesis or world model, is denoted $\Theta_l^k$ and defined by

$$\Theta_l^k = \{Z^{k,l_i}\}_{i=0}^{I} \tag{9}$$

where $Z^{k,l_i}$ represents the set of measurements assigned to track $l_i$ and the set $Z^{k,l}$ denotes spurious measurements or false alarms. That is, measurements are either categorized as originating from geometric features in the environment or else are considered spurious due to noise. The first stage of the algorithm is *track feasibility construction*, in which sequences of measurements are grouped into feasible tracks that are reasonable to incorporate into an hypothesis. Track feasibility can be tested in the same manner as the track-splitting algorithm. This is more efficient than first forming a hypothesis, $\Theta_l^k$, and then checking the feasibility of its tracks, $Z^{k,l_i}$, since the same track may appear in more than one hypothesis. The set, $\mathcal{T}$, of feasible tracks contain $r$ tracks but these tracks are not all disjoint.

Next, a hypothesis $\Theta_l^k$ must select a set of *disjoint* tracks from the feasible set $\mathcal{T}$, and the likelihood of the hypothesis must be calculated. Just as the track-splitting filter calculated the likelihood of a *track*, Morefield estimates the joint likelihood of a *partition*, $\Theta_l^k$, that is, a set of disjoint tracks. If $\Phi$ denotes the set of all feasible partitions, then the joint-likelihood method finds the most likely partition, that is,

$$\max_{\Theta_l^k \in \Phi} p\,[Z^k \mid \Theta_l^k] \tag{10}$$

The fundamental assumption in the derivation of the linear programming formulation is that the data in each track are independent, so that equation (10) becomes

$$\max_{\Theta_l^k \in \Phi} \prod_{i=0}^{I} p\,[Z^{k,l_i} \mid \Theta_l^k] \tag{11}$$

Any hypothesis $\Theta_l^k$ can be represented by a binary vector $\boldsymbol{\rho}$ of length $r$, the $i$th element of $\boldsymbol{\rho}$ being unity if track $i$ belongs to hypothesis $\Theta_l^k$. Each hypothesis, $\Theta$, can then be evaluated using a linear functional $\mathbf{c}'\boldsymbol{\rho}$ where the $i$th component of $\mathbf{c}$ is the contribution of track $Z^{k,i}$ to $p[Z^k \mid \Theta]$. The $r$-dimensional vector, $\mathbf{c}$, is defined as

$$\mathbf{c} \triangleq \{\tilde{\lambda}^{k,1}, \ldots, \tilde{\lambda}^{k,r}\}' \tag{12}$$

where $\tilde{\lambda}^{k,j}$ is the log-likelihood ratio of track $j$. It is defined as

$$\begin{aligned}\tilde{\lambda}^{k,j} &= -\log\left[\frac{p[Z^{k,j} \mid \Theta_l^k]}{p[Z^{k,j} \mid \text{all false}]}\right] \\ &= \log p[Z^{k,j} \mid \Theta_l^k] + N_j \log V^{-1}\end{aligned} \tag{13}$$

where $N_j$ is the number of measurements in the sequence $Z^{k,j}$ and $V$ is the surveillance volume or field of view of the camera.

Finally, the constraint that all tracks in a legal partition should be disjoint must be imposed. Different feasible partition sets, $\Theta_l^k$, may have a different number, $I$, of partitions/tracks, but for any feasible set of partitions, *all* measurements are accounted for, that is,

$$Z^k = \bigcup_{l_i=0}^{I} Z^{k,l_i} \tag{14}$$

and the partitions of a hypothesis are disjoint

$$Z^{k,i} \bigcap Z^{k,j} = \emptyset \qquad \forall i \neq j \tag{15}$$

so that a measurement originates from only one geometric feature.

In order to impose the disjointness constraint of equation (15), a binary vector $\Psi^l$ of dimension $N = \dim(Z^k)$ the total number of observed measurements is defined for each track $l$. A nonzero element in the $i$th position of $\Psi^l$ indicates that this measurement has been assigned to track $l$. Equation (15) can now be rewritten as

$$\Psi^i + \Psi^j \leq \mathbf{1} \qquad i \neq j, \quad \Psi_i, \Psi_j \in \Theta_l^k \tag{16}$$

where $\mathbf{1}$ is a vector of 1's and both tracks are included in the same hypothesis. To form a complete set of inequalities, the matrix $\mathbf{A}$,

$$\mathbf{A} \triangleq (\Psi_1, \ldots, \Psi_r) \tag{17}$$

is formed, so that equation (16) becomes

$$\mathbf{A}\rho \leq \mathbf{1} \tag{18}$$

The above results allow the joint-likelihood data-association problem to be formulated as

$$\min \mathbf{c}'\rho \quad \text{subject to} \quad \mathbf{A}\rho \leq \mathbf{1} \text{ and } \rho \text{ binary} \tag{19}$$

This is a very well-studied optimization problem. Known as the set-packing problem, it is of considerable importance in many diverse fields. Salkin [26] provides a good review of the topic. The set-packing problem is known to be NP-complete [17]. However, because of its importance, standard packages do exist to solve it. Morefield also notes that a significant reduction in computation can be achieved if the problem can be reduced to $L$ subproblems each of which have no data points in common. Such a situation arises in the tracking of two or more geometric features that are spatially separated from one another, that is, their validation volumes do not overlap.

The main disadvantage of the joint-likelihood approach is that it is a batch process. An on-line recursive formulation is possible, but the performance of on-line set-packing algorithms can be significantly worse than batch techniques. A second problem is that the initiation and termination of geometric features is not explicitly handled by the algorithm, but is performed instead by the track feasibility stage. If the number of perceptually relevant geometric features changes with time as objects enter and/or leave the camera's field of view, then an algorithm is needed that is capable of initiating and terminating geometric features. These problems are addressed by the multiple-hypothesis algorithm of Reid [24].

## 6 Multiple-Hypothesis Algorithm

Track initiation and termination are important capabilities for motion-correspondence algorithms. After all, it is seldom the case that all relevant geometric features are known a priori and are present from the start to finish of the motion sequence. More usually, as the camera moves, new areas of an object or the environment come into view and new geometric features must be initiated to model and track them. Similarly, track termination is important to handle features that leave the field of view of the sensor, for example. The multiple-hypothesis filter, originally developed by Reid [24] provides these capabilities. Most recently, Cox and Leonard [8] have demonstrated its utility in the context of building and maintaining a map of a mobile robot's environment. The interested reader is also directed to Cox and Leonard [9] for a discussion of unsupervised learning and the motion-correspondence problem.

Figure 4 outlines the basic operation of the algorithm. An iteration begins with the set of current hypotheses from iteration $(k - 1)$. Each hypothesis (leaf) contains a set of active tracks, and becomes a parent hypothesis node in the current iteration. Each hypothesis provides an interpretation of all past measurements consisting of a collection of disjoint tracks. Predictions are made as to the expected location of measurements and these predictions are matched to actual measurements using the Mahalanobis distance. Each measurement may either (1) belong to a previously known geometric feature or (2) be the start of a new geometric feature or (3) be a false alarm. In addition, for geometric features that are not assigned measurements, there is the possibility of (4) deletion of the geometric feature. This situation may arise when a learned feature such as a stationary desk, is moved to a new position. The resulting enumeration of associations, described in section 6.1, produces a set of children (events) for each parent node, extending the depth of the tree by another level. Associated with each new leaf is a probability whose computation is described in section 6.2. In the final step of the iteration, the tree is pruned to remove unlikely correspondences.

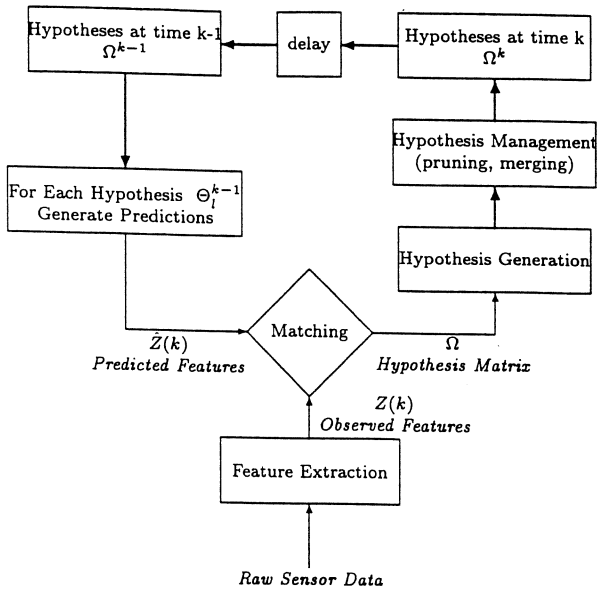


*Fig. 4.* Outline of the multiple-hypothesis algorithm.

### 6.1 Hypothesis Generation

A particular global hypothesis at time $k$ is defined by $\Theta_l^k$. Let $\Theta_{m(l)}^{k-1}$ denote the parent hypothesis from which $\Theta_l^k$ is derived, and $\theta_m(k)$ denote the *specific* set of assumed assignments that map $\{\Theta_{m(l)}^{k-1}, Z(k)\}$ to $\Theta_l^k$. That is, $\theta_m(k)$ is a set of assignments of the origins of all measurements received at time $k$ with all the geometric features postulated by the parent hypothesis, $\Theta_{m(l)}^{k-1}$, at time $k$. The event $\theta_l(k)$ based on the current measurements is defined to consist of $\tau$ measurements from known geometric features, $\nu$ measurements from new geometric features, $\phi$ false alarms and $\chi$ deleted (or obsolete) geometric features from the parent hypothesis.

A set of current assignments or events $\theta_l(k)$ can be generated by first creating a hypothesis matrix in which known geometric features are represented by the columns of the matrix and the current measurements by the rows. A nonzero element at matrix position $c_{i,j}$ denotes that measurement $\mathbf{z}_i(k)$ is contained in the validation region of geometric feature $t_j$. In addition to the total number, $T$, of known geometric features postulated by a hypothesis, the hypothesis matrix has appended to it a column 0 denoting false alarms and a column $T+1$ denoting new geometric features. The situation depicted in figure 2 is represented by the hypothesis matrix shown in figure 5.

The two constraints that (1) a measurement originates from only one source feature and (2) a geometric feature has at most one associated measurement per iteration can be imposed by restricting an assignment matrix to have only a single nonzero value in any row or column, except for the first and last columns since any number of measurements might be false alarms or new geometric features. Figure 5 therefore concisely represents the ambiguities that are present in figure 2.

$$\Omega = \begin{array}{c} \begin{matrix} T_F & T_1 & T_2 & T_N \end{matrix} \\ \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} \end{array} \begin{matrix} \mathbf{z}_1(k) \\ \mathbf{z}_2(k) \\ \mathbf{z}_3(k) \end{matrix}$$

*Fig. 5.* Hypothesis matrix for the situation depicted in figure 2.

Hypothesis generation is then performed by picking one unit per row and one unit per column except for columns $T_F$ and $T_N$ as illustrated in figure 5. Enumeration of all legal sets of assignments, $\theta_l(k)$, is straightforward [33] and is followed by the calculation of the probability of each new hypothesis.

### 6.2 Probability Calculations

The new hypothesis at time $k$, $\Theta_l^k$ is made up of the current set of assignments (also called an event), $\theta_l(k)$, and a previous hypothesis, $\Theta_{m(l)}^{k-1}$ based on measurements up to and including time $k-1$, that is,

$$\Theta_l^k = \{\Theta_{m(l)}^{k-1}, \theta_l(k)\} \tag{20}$$

The probability of an hypothesis, $P\{\Theta_l^k|Z^k\}$ can be calculated using Bayes' rule, so that

$$\begin{aligned} P\{\Theta_l^k|Z^k\} &= P\{\theta_l(k), \Theta_{m(l)}^{k-1}|Z(k), Z^{k-1}\} \\ &= \frac{1}{c} p[Z(k)|\theta_l(k), \Theta_{m(l)}^{k-1}, Z^{k-1}] \\ &\quad \times P\{\theta_l(k)|\Theta_{m(l)}^{k-1}, Z^{k-1}\} \\ &\quad \times P\{\Theta_{m(l)}^{k-1}|Z^{k-1}\} \end{aligned} \tag{21}$$

where $c$ is a normalization constant. The last term of this equation, $P\{\Theta_{m(l)}^{k-1}|Z^{k-1}\}$, represents the probability of the parent global hypothesis and is therefore available from the previous iteration. The remaining two terms may be evaluated as follows.

The second factor of equation (21) is obtained by combining results from [5] and [19] to yield

$$P\{\theta_l(k)|\Theta^{k-1}_{m(l)}, Z^{k-1}\} = \frac{\phi!\nu!}{m_k!}\mu_F(\phi)\mu_N(\nu) \prod_t (P^t_D)^{\delta_t}(1 - P^t_D)^{1-\delta_t}(P^t_\chi)^{\chi_t}(1 - P^t_\chi)^{1-\chi_t} \quad (22)$$

where $\mu_F(\phi)$ ans $\mu_N(\nu)$ are the prior probability mass functions (pmfs) of the number of false measurements and new geometric features, $P^t_D$ and $P^t_\chi$ are the probabilities of detection and termination (deletion) of track $t$ and $\delta_t$ and $\chi_t$ are indicator variables defined by

$$\delta_t \triangleq \begin{cases} 1 & \text{if geometric feature } t \\ & \text{(in } \Theta^{k-1}_{m(l)}\text{) is detected at time } k \\ 0 & \text{otherwise} \end{cases} \quad (23)$$

$$\chi_t \triangleq \begin{cases} 1 & \text{if geometric feature } t \\ & \text{(in } \Theta^{k-1}_{m(l)}\text{) is deleted at time } k \\ 0 & \text{otherwise} \end{cases} \quad (24)$$

To determine the first term on the right hand side of equation (21), it is assumed that a measurement $\mathbf{z}_i(k)$ has a Gaussian probability density function (pdf)

$$N_{t_i} = N[\mathbf{z}_i(k)] \triangleq N[\mathbf{z}_i(k); \hat{\mathbf{z}}_i(k|k-1), \mathbf{S}^{t_i}(k)] = |2\pi\mathbf{S}^{t_i}(k)|^{-1/2} \times \exp\frac{1}{2}\{[\mathbf{z}(k) - \hat{\mathbf{z}}(k|k-1)]'\{\mathbf{S}^{t_i}(k)\}^{-1}[\mathbf{z}(k) - \hat{\mathbf{z}}(k|k-1)]\} \quad (25)$$

if it is associated with geometric feature $t_i$, where $\hat{\mathbf{z}}_i(k|k-1)$ denotes the predicted measurement for geometric feature $t_i$ and $\mathbf{S}^{t_i}(k)$ is the associated innovation covariance. If the measurement is a false alarm, then its pdf is assumed uniform in the observation volume, $V$. The probability of a new geometric feature is also taken to be uniform with pdf $V^{-1}$.[5] Under these assumptions, we have that

$$p[Z(k)|\theta_l(k), \Theta^{k-1}_{m(l)}, Z^{k-1}] = \prod_{i=1}^{m_k}\{N_{t_i}[\mathbf{z}_i(k)]\}^{\tau_i}V^{-(1-\tau_i)} = V^{-\phi-\nu}\prod_{i=1}^{m_k}[N_{t_i}\{\mathbf{z}_i(k)]\}^{\tau_i} \quad (26)$$

where $\tau_i$ is an indicator variable defined as

$$\tau_i \triangleq \begin{cases} 1 & \mathbf{z}_i(k) \text{ came from a} \\ & \text{known geometric feature} \\ 0 & \text{otherwise} \end{cases} \quad (27)$$

and $\nu$ and $\phi$ are the total number of new geometric features and false alarms respectively.

Substituting equations (26) and (22) into equation (21) yields the final expression for the conditional probability of an association hypothesis

$$P\{\Theta^k_l|Z^k\} = \frac{1}{c}\frac{\phi!\nu!}{m_k}\mu_F(\phi)\mu_N(\nu)V^{-\phi-\nu}\prod_{i=1}^{m_k}\{N_{t_i}[\mathbf{z}_i(k)]\}^{\tau_i} \times \left\{\prod_t (P^t_D)^{\delta_t}(1 - P^t_D)^{1-\delta_t}(P^t_\chi)^{\chi_t}(1 - P^t_\chi)^{1-\chi_t}\right\} \times P\{\Theta^{k-1}_{m(l)}|Z^{k-1}\} \quad (28)$$

If the number of false alarms and new features are assumed to be Poisson distributed[6] with densities $\lambda_F$ and $\lambda_N$, respectively, then equation (28) reduces to

$$P\{\Theta^k_m|Z^k\} = \frac{1}{c'}\lambda^\nu_N\lambda^\phi_F\prod_{i=1}^{m_k}\{N_{t_i}[\mathbf{z}_i(k)]\}^{\tau_i} \times \left\{\prod_t (P^t_D)^{\delta_t}(1 - P^t_D)^{1-\delta_t}(P^t_\chi)^{\chi_t}(1 - P^t_\chi)^{1-\chi_t}\right\} \times P\{\Theta^{k-1}_{m(l)}|Z^{k-1}\} \quad (29)$$

The probability of each hypothesis can be used to guide a pruning strategy described next.

### 6.3 Implementation

The multiple-hypothesis approach has exponential complexity. Consequently, heuristic pruning strategies must be applied to contain the growth of the hypothesis tree. Several implementation issues are worthy of mention.

First, just as with the joint-likelihood approach, the same track may appear in more than one hypothesis. Rather than replicate common tracks over all such hypotheses and incurring the corresponding computational and storage overheads, a separate track tree is formed [19]. Each hypothesis then contains pointers to nodes in the track tree as illustrated in figure 6. Each set represents a different permutation of track leaf nodes from different track trees, i.e., the global hypotheses enforce the assumptions of disjoint partitions. The track tree provides considerable savings and is discussed in detail by Kurien [19].

***6.3.1 Spatially Disjoint Cluster.*** A considerable reduction in the combinatorics can be achieved by realizing that it is not necessary to form a single global hypothesis tree containing tracks that do not have any

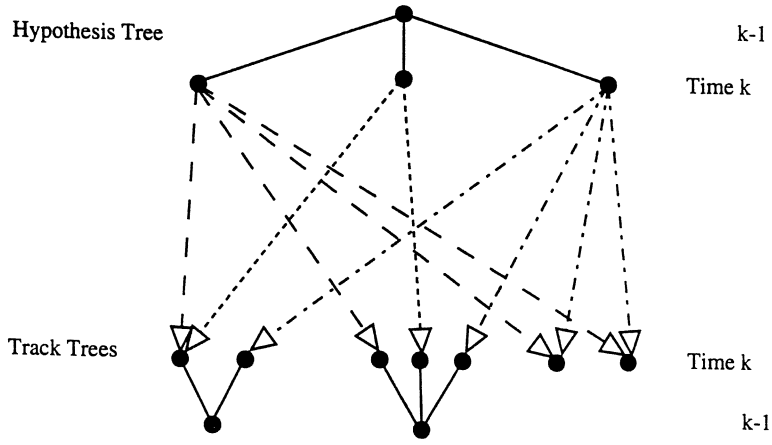


*Fig. 6.* The track and hypothesis trees.

common measurements. Instead, tracks can be partitioned into separate *clusters* as proposed by Reid [24].[7] Tracks within each cluster share common measurements whereas tracks in different clusters do not. A separate hypothesis tree is grown for each spatially disjoint cluster and consequently, the combinatorial problem associated with forming global hypotheses is significantly reduced.

Of course, each new set of measurements must be checked to determine whether a measurement is shared (falls in the validation region) between two or more clusters. If so, these clusters must be merged. Similarly, a cluster containing two or more geometric features that do not share common measurements may be split.

***6.3.2 Pruning.*** Pruning is essential to any practical implementation of this algorithm. Pruning is based on a combination of an "N-scan-back" algorithm [19] and a simple lower limit probability threshold. The $N$-scan-back algorithm assumes that any ambiguity at time $k$ is resolved by time $k + N$. Then, if hypothesis $\Theta_i^k$ at time $k$ has $q$ children, the sum of the probabilities of the leaf nodes is calculated for each of the $q$ branches. Whichever branch has the greatest probability is retained and all other branches are pruned. The result is an irrevocable decision regarding the assignment of measurements to tracks based on looking ahead $N$ time steps. Consequently, below the decision node there is a tree of depth $N$ while above the decision node the tree has degenerated into a simple list of assignments, as illustrated in figure (7). It is clearly computationally advantageous to set $N$ as small as possible. Research suggests that very good results can be obtained for $N = 3$ [8, 9] and even $N = 2$ [19] can provide near optimum solutions in some circumstances. Nevertheless, after $N$-scan-back pruning the number of leaf nodes can still be high. A second phase of pruning removes all nodes whose probability is less than a lower limit, say 0.01, so that at the end of each iteration there are at most 100 hypotheses.

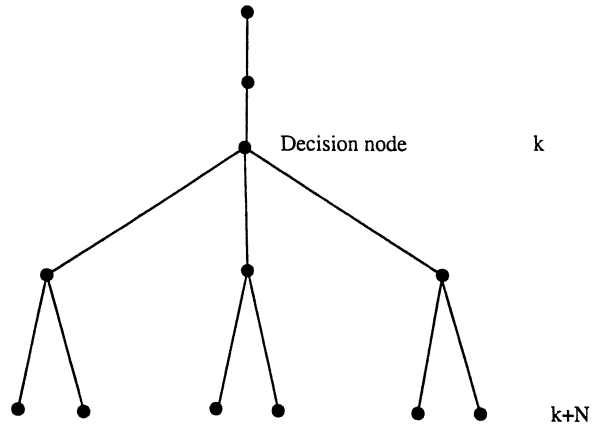


*Fig. 7.* An hypothesis tree after $N$-scan-back" pruning.

## 7 The Joint-Probabilistic Data-Association Algorithm

The main problem with the three previous approaches is their exponential complexity, which can demand large computation and memory resources even if significant pruning is applied. Whether such demands can be met

depends very much on the application, for example, the level of ambiguity present, the computational time involved in a single cycle from time $k$ to $k + 1$ can vary significantly, depending on the level of ambiguity and the corresponding sizes of the trees, can be very problematic in some real-time situations. There is therefore a need for simpler, albeit suboptimal, data-association algorithms.

A class of suboptimal algorithms exist that requires almost fixed computational resources per cycle. The joint probabilistic data association algorithm weights *all* measurements with *all* tracks. The weights represent the probability that measurement $\mathbf{z}_i(k)$ originated from geometric feature $l$. Hence the term *probabilistic data association*. The original probabilistic data-association filter (PDAF) assumed the existence of only a single target whose track has been initialized [6]. The joint-probabilistic data-association filter (JPDAF) [16] extended this to a fixed *known* number of targets and is described below. These are strong assumptions, but acceptable in some situations. Most recently, the JPDAF has been further enhanced to deal with the case of a track splitting into two component tracks [4].

Consider a cluster of $T$ tracks as defined in the cluster partitioning of the multiple-hypothesis filter, section (6.3.1). At time $k$, $Z(k) = \{\mathbf{z}_1(k), \ldots, \mathbf{z}_{m_k}\}$ measurements are detected in the combined validation regions of these tracks. Let $\hat{\mathbf{z}}^l$ denote the predicted measurement value from track $l$ and $\boldsymbol{\nu}_{i,l}$ the associated innovation due to measurement $\mathbf{z}_i$. The JPDA associates *all* measurements with each track to form a combined weighted innovation given by

$$\boldsymbol{\nu}^l = \sum_{i=1}^{m_k} \beta_i^l \boldsymbol{\nu}_{i,l} \tag{30}$$

where $\beta_i^l$ is the posterior probability that measurement $i$ originated from geometric feature $l$ and $\beta_0^l$ is the probability that none of the measurements originated from feature $l$. The weighted innovation is then applied in the standard Kalman filter update equations for each track $l$.

In order to determine $\beta_i^l$ it is necessary to determine the probability that measurement $\mathbf{z}_i$ originated from feature $l$. To do this, a hypothesis matrix is constructed for each track and sets of assignments, $\theta_l$, are generated, assuming as before that a measurement originates from only a single feature and that a feature generates only a single measurement. Then the innovation weightings $\beta_i^l$ are defined by

$$\beta_{i,l} \triangleq \sum_{\theta} P[\theta_l(k) \mid Z^k] \tau_{i,l}(\theta) \begin{cases} i & = 1, \ldots, m_k \\ l & = 0,1, \ldots, T \end{cases} \tag{31}$$

where the indicator variable $\tau_{i,l}(\theta)$ is unity if measurement $\mathbf{z}_i(k)$ is associated with track $l$ in the assignment $\theta_l(k)$ and zero otherwise.

The conditional probability of the joint association event $\theta_l(k)$ is

$$P(\theta_l(k) \mid Z^k) = P[\theta_l(k) \mid Z(k), Z^{k-1}] \tag{32}$$

$$= \frac{1}{c} p[Z(k) \mid \theta_l(k), Z^{k-1}] P[\theta_l(k)] \tag{33}$$

where $c$ is a normalization constant. The pdf is given by

$$p[Z(k) \mid \theta_l(k), Z^{k-1}] = \prod_{i=1}^{m_k} p[\mathbf{z}_i(k) \mid \theta_l(k), Z^{k-1}] \tag{34}$$

Assuming a normal distribution for measurements associated to tracks and a uniform distribution in the field of view (surveillance volume) $V$ for measurements not assigned to tracks, then

$$p[\mathbf{z}_i(k) \mid \theta_l(k), Z^{k-1}] = \begin{cases} N_l[\mathbf{z}_i(k)] & \text{if } \tau_{i,l}[\theta_l(k)] = 1 \\ V^{-1} & \text{if } \tau_{i,l}[\theta_l(k)] = 0 \end{cases} \tag{35}$$

Thus equation (34) becomes

$$p[Z(k) \mid \theta_l(k), Z^{k-1}] = V^{-\phi[\theta_l(k)]} \prod_{i=1}^{m_k} \{N_l[\mathbf{z}_i(k)]\}^{\tau[\theta_l(k)]} \tag{36}$$

where $\phi[\theta_l(k)]$ is the total number of false alarms present in the specific assignment event $\theta_l(k)$.

Fortmann et al. show that the probability of the assignments, $\theta_l(k)$, is given by

$$P[\theta_l(k)] = \frac{\phi!}{m_k!} \mu_F(\phi) \prod_{t=1}^{T} (P_D^l)^{\delta_l} (1 - P_D^l)^{1-\delta_l} \tag{37}$$

assuming each event equally likely. $P_D^l$ is the probability of detection of geometric feature $l$, $\mu_F(\phi)$ is the prior PMF of the number of false measurements and $\delta_l$ is an indicator variable that is unity if a measurement is assigned to a track $l$ in the event $\theta_l(k)$ and zero otherwise.

Combining equations (34) and (37) into equation (32) yields

$$P[\theta_l(k) \mid Z^k] = \frac{1}{c}\frac{\phi!}{m_k!}\mu_F(\phi)V^{-\phi} \prod_{j=1}^{m_k}\{N_l[\mathbf{z}_j(k)]\}^{\tau_j}\prod_{l=1}^{T}(P_D^l)^{\delta_l}(1 - P_D^l)^{1-\delta_l} \quad (38)$$

There are two versions of the JPDA depending on the assumed model for the false measurements, $\mu_F(\phi)$. The parametric JPDA assumes a Poisson distribution while the nonparametric JPDA assumes a uniform distribution.

Chang and Aggarwal [7] have applied the JPDA to the problem of 3-D structure reconstruction from an ego motion sequence. Good performance can be achieved with this filter and the filter is becoming increasingly popular within the air traffic control community [2]. The principal limitation of the JPDA is its inability to perform track initiation and deletion. Since the JPDA algorithm is designed for the restricted case in which a *known* number of geometric features are to be tracked, serious errors can arise if the number of geometric features should change. However, recently there has been progress toward supplementing the JPDA with these capabilities [3], including combining the JPDA with a version of the multiple-hypothesis filter for track formation [2] and a version of the JPDA that can track an object that "splits" into two targets [4].

## 8 Conclusion

Although the data-association problem can sometimes be significantly reduced if the sampling rate of the sensors can be increased and/or the sensor's motion between sampling intervals can be restricted, ultimately some ambiguity in motion correspondences is unavoidable, especially in dynamic environments. Resolving motion correspondences is a very difficult problem which has received significant attention within the surveillance and target-tracking community. Here, we have reviewed several statistical data association techniques developed by the target-tracking community but having applicability to computer vision.

An assumption common to all the reviewed algorithms is that measurements are normally distributed about their predicted values. In this case, the Mahalanobis distance can be used to quantify the likelihood that a measurement originated from a specific geometric feature and a validation region can be defined within which a measurement will be found with some specified probability. The validation region or gate serves two purposes. First, it provides an attention focusing capability. Second, it significantly reduces the assignment combinatorics by not considering measurements that fall outside of the validation volume of a track.

The Mahalanobis distance suggests an obvious solution to the motion correspondence problem, the nearest-neighbor algorithm. Unfortunately, there is always a finite chance that such an assignment is incorrect, which can have serious affects on the performance of the Kalman filter. The nearest-neighbor algorithm makes assignment decisions based solely on the current image frame, but much more information is available by postponing assignment decisions and examining subsequent images. The track-splitting, joint-likelihood and multiple-hypothesis algorithms all exploit some form of look-ahead capability, but all three algorithms have exponential complexity.

The track-splitting algorithm is the simplest of the three. However, it allows measurements to be shared between tracks, which does not represent the physics of the imaging process. The joint-likelihood algorithm improves on the track-splitting filter by enforcing disjoint partitions of the measurements, that is, a measurement can originate from only a single geometric feature. The principal criticisms of the joint-likelihood approach are that it is a batch rather than recursive procedure and, like the track-splitting algorithm before it, there is no explicit modeling of when to initiate a new track or terminate an existing track. These disadvantages are overcome in the multiple-hypothesis algorithm, a recursive algorithm that provides the capabilities of track initiation and termination. An efficient implementation of the multiple-hypothesis algorithm involves creating a list of track trees, as with the track-splitting algorithm, and a set of hypothesis trees, each hypothesis pointing to a set of disjoint track tree nodes. The multiple-hypothesis algorithm is therefore somewhat more complicated than track-splitting, but avoids many of the heuristics needed by the latter to deal with initiation, termination, and merging of tracks. Consequently, while the initial cost of implementing the multiple hypothesis may be high, this is more than offset by its significantly better performance.

Of course, not all applications require solution to the general-purpose motion correspondence problem. Further, suboptimal algorithms may be needed to meet

real-time constraints. The joint-probabilistic data-association filter is one such algorithm and is applicable to situations in which the number of targets to be tracked is known a priori. The JPDAF assigns all measurements to all tracks, weighting each measurement based on an estimate of the probability that the measurement belongs to the track. The original JPDA algorithm suffered from an inability to initiate or delete tracks, but there has been significant recent work to address this limitation.

The choice of which statistical data-association algorithm to use depends strongly on the particular application: for example, is the number of geometric features to be tracked known a priori or is automatic track initiation and deletion necessary and what real-time constraints must be met? Currently, the choice appears to be between some form of hybrid JPDA algorithm that offers better real-time capabilities and the multiple-hypothesis filter which has superseded the track-splitting and joint-likelihood methods and provides superior track initiation and deletion capabilities which may be critical in dynamic environments.

## Acknowledgments

It is a pleasure to thank Y. Bar-Shalom, H. Durrant-Whyte, T. Kanade, and J.J. Leonard for many fruitful discussions.

## Notes

1. Strictly, the term *temporal* data association should be used to distinguish motion correspondence from the static data association problem in which pixels in a single image must be partitioned into perceptually relevant groups. In the latter case, there is no explicit ordering of the measurements, except for the arbitrary ordering imposed by the sampling lattice. Temporal data-association techniques can sometimes be applied to the static case, as in the work of Cox et al. [10].
2. The term $\hat{\mathbf{x}}(i \mid j)$ should be read as "the estimate of the vector $\mathbf{x}$ at time step $i$ given all observations up to time step $j$."
3. The prediction $\hat{\mathbf{z}}_i(k+1 \mid k)$ and innovation covariance $S(k+1)$ are precisely what is calculated using the Kalman filter. It is therefore often convenient to associate a Kalman filter with every geometric feature, the state vector representing the position and velocity of the associated feature.
4. Equation (6) assumes that a measurement for a track/geometric feature is present at each iteration, i.e., that a measurement is always found inside the validation region of each track. However, the algorithm can be modified to deal with detection probabilities of less than unity.
5. Intuitively, the choice of uniform pdf's for false alarms and new features seems less justifiable for robotic applications than for traditional radar and underwater sonar tracking applications. The impact of these assumptions needs further investigation.
6. Uniform distributions can also be easily accommodated.
7. This spatial partitioning is also common to Morefield's joint likelihood integer programming scheme.

## References

1. N. Ayache and O. Faugeras, Maintaining representations of the environment of a mobile robot, *IEEE Trans. Robotics Autom.* 5(6): 804–819, 1989.
2. H.A.P. Blom, R.A. Hogendoorn, and B.A. van Doorn, Design of a multisensor tracking system for advanced air traffic control. In Y. Bar-Shalom, ed., *Multitarget-Multisensor Tracking: Advanced Applications*. Artech House: Norwood, MA, vol. 2, pp. 31–63, 1992.
3. Y. Bar-Shalom, K.C. Chang, and H.A.P. Blom, Automatic track formation in clutter with a recursive algorithm. In Y. Bar-Shalom, ed., *Multitarget-Multisensor Tracking: Advanced Applications*, Artech House: Norwood, MA, pp. 25–42, 1990.
4. Y. Bar-Shalom, K.C. Chang, and H.A.P. Blom, Tracking splitting targets in clutter by using an interacting multiple model joint probabilistic data association filter. In Y. Bar-Shalom, ed., *Multitarget-Multisensor Tracking: Advanced Applications*, Artech House: Norwood, MA, vol. 2, pp. 93–110, 1992.
5. Y. Bar-Shalom and T.E. Fortmann, *Tracking and Data Association*, Academic Press: San Diego, CA, 1988.
6. Y. Bar-Shalom and A.G. Jaffer, Adaptive nonlinear filtering for tracking with measurements of uncertain origin, *Proc. 11th IEEE Conf. on Decision and Control*, pp. 243–247, 1972.
7. Y.L. Chang and J.K. Aggarwal, 3d structure reconstruction from an ego motion sequence using statistical estimation and detection theory, *IEEE Workshop in Visual Motion*, pp. 268–273, 1991.
8. I.J. Cox and J.J. Leonard, Probabilistic data association for dynamic world modeling: A multiple hypothesis appraoch, *Proc. Intern. Conf. Advanced Robotics*, Pisa, Italy, 1991.
9. I.J. Cox and J.J. Leonard, Unsupervised learning for mobile robot navigation using probabilistic data association, *Workshop on Computer Learning and Natural Learning*, Berkeley, CA, 1991.
10. I.J. Cox, J.M. Rehg, and S. Hingorani, A Bayesian multiple hypothesis approach to contour segmentation, *Proc. 2nd Europ. Conf. Comput. Vis.*, pp. 72–77, Italy, 1992.
11. J.L. Crowley, P. Stelmaszyk, and C. Discours, Measuring image flow by tracking edge-lines, *Proc. Intern. Conf. Comput. Vis.*, pp. 658–664, Tampa, FL, 1988.
12. J.B. Collins and J.K. Uhlmann, Efficient gating in data association with multivariate Gaussian distributions, *NRL*, 1992.
13. M.R.W. Dawson, The how and why of what went where in apparent motion: Modeling solutions to the motion correspondence problem, *Psychological Review* 98(4): 569–603, 1991.
14. R. Deriche and O. Faugeras, Tracking line segments. In O. Faugeras, ed., *Proc. Europ. Conf. Comput. Vis.*, pp. 259–268. Springer-Verlag: New York, 1990.
15. R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. Wiley: New York, 1973.

16. T.E. Fortmann, Y. Bar-Shalom, and M. Scheffe, Sonar tracking of multiple targets using joint probabilistic data association, *IEEE J. Ocean. Engineer.* OE-8(3): 173–184, 1983.
17. M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman: New York, 1979.
18. R.W. Hockney and J.W. Eastwood, *Computer Simulation Using Particles*. Adam Hilger: Bristol, UK, 1988.
19. T. Kurien, Issues in the design of practical multitarget tracking algorithms. In Y. Bar-Shalom, ed., *Multitarget-Multisensor Tracking: Advanced Applications*, pp. 43–83, Artech House: Norwood, MA, 1990.
20. J.E.W. Mayhew and J.P. Frisby, Psychophysical and computational studies towards a theory of human stereopsis, *Artificial Intelligence*, 17, 1981.
21. C.L. Morefield, Application of 0–1 interger programming to multitarget tracking problems, *IEEE Trans. Autom. Contr.* AC-22(6), June 1977.
22. M.J.L. Orr, J. Hallam, and R.B. Fisher, Fusion through interpretation. In G. Sandini, ed., *Second European Conference on Computer Vision*, pp. 801–805. Springer-Verlag: New York, 1992.
23. F.P. Preparata and M.I. Shamos, *Computational Geometry: An Introduction*. Springer-Verlag: New York, 1985.
24. D.B. Reid, An algorithm for tracking multiple targets. *IEEE Trans. Autom. Contr.* AC-24(6): 843–854, December 1979.
25. I. Rock and S. Palmer, The legacy of gestalt psychology, *Scientific American* 263(6): 84–90, December 1990.
26. H.M. Salkin, *Integer Programming*. Addison-Wesley: Reading, MA, 1975.
27. P. Smith and G. Buechler, A branching algorithm for discriminating and tracking multiple objects, *IEEE Trans. Autom. Contr.* AC-20: 101–104, 1975.
28. C.W. Therrien, *Decision Estimation and Classification: An Introduction to Pattern Recognition and Related Topics*. Wiley: New York, 1989.
29. J.K. Uhlmann, Adaptive partitioning strategies for ternary tree structures, *Patt. Recog. Ltts.* 12: 537–541, 1991.
30. J.K. Uhlmann, Satisfying general proximity/similarity queries with metric trees, *Inform. Process. Ltts.* 40: 175–179, 1991.
31. J.K. Uhlmann, Algorithms for multiple-target tracking. *American Scientist* 80: 128–141, 1992.
32. Z. Zhang and O.D. Faugeras, Three-dimensional motion computation and object segmentation in a long sequence of stereo frames, *Intern. J. Comput. Vis.* 7(3): 211–241, 1992.
33. B. Zhou, *Multitarget Tracking in Clutter: Algorithms for Data Association and State Estimation*. Ph.D. thesis, Pennsylvania State University, 1992.