An Efficient Implementation of Reid's Multiple Hypothesis Tracking Algorithm and Its Evaluation for the Purpose of Visual Tracking

Ingemar J. Cox and Sunita L. Hingorani

Abstract—An efficient implementation of Reid's multiple hypothesis tracking (MHT) algorithm is presented in which the k-best hypotheses are determined in polynomial time using an algorithm due to Murty [24]. The MHT algorithm is then applied to several motion sequences. The MHT capabilities of track initiation, termination, and continuation are demonstrated together with the latter's capability to provide low level support of temporary occlusion of tracks. Between 50 and 150 corner features are simultaneously tracked in the image plane over a sequence of up to 51 frames. Each corner is tracked using a simple linear Kalman filter and any data association uncertainty is resolved by the MHT. Kalman filter parameter estimation is discussed, and experimental results show that the algorithm is robust to errors in the motion model. An investigation of the performance of the algorithm as a function of look-ahead (tree depth) indicates that high accuracy can be obtained for tree depths as shallow as three. Experimental results suggest that a real-time MHT solution to the motion correspondence problem is possible for certain classes of scenes.

Index Terms—Multiple hypothesis tracking, motion correspondence, data association, tracking, visual tracking, ranked bipartite graph matching.

1 Introduction[1]

The analysis of image sequences for purposes of estimating camera motion and/or 3-D scene geometry often requires the tracking of geometric features over long image sequences. Typically, predictions are first made as to the expected locations of the current set of features of interest. These predictions are then matched to actual measurements. At this stage, ambiguities may arise. Predictions 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? These ambiguities must be resolved in order to solve the motion correspondence problem.

Visual tracking has been extensively studied in recent years. However, almost all such work has assumed that the motion correspondence problem has been solved or is trivial so that a nearest neighbor strategy is effective. In some cases, a nearest neighbor strategy is indeed adequate. For example, Tomasi and Kanade [30]. track corner features over very many frames using such an approach. A nearest neighbor strategy usually relies on the frame-to-frame image motion being extremely small. Much more data must then be processed than if a sparser sampling were used. However, if significant frame to frame motions are present, then ambiguities can quickly arise. Zheng and Chellappa [34] minimize these ambiguities by using a weighted correlation window to detected tracked features in the next frame. While correlation techniques can significantly reduce the motion correspondence ambiguity, our experiments suggest that partial occlusion and significant changes in the background can be problematic for such methods. Moreover, such techniques are only appropriate to the detection of measurements from *existing* tracking, not for the detection of *new* tracks. Many researchers have used the Kalman filter to track geometric features such as lines [1], [17] and corners [5], [4] in a scene, under the assumption that motion correspondence is straightforward. The motivation and significance of this work was in designing stable and reliable algorithms to infer the 3-D structure and motion from 2-D image plane measurements. Shapiro et al. [28] describe tracking corners in the image plane. Their system has several similarities to the one described herein, specifically, the use of Kalman filtering and a cross correlation measure to compare corners. However, the motion correspondence problem is not rigorously addressed: correspondences are determined between two consecutive frames based on a similarity measure between corners. Correspondences are determined without looking at subsequent frames and there is no mechanism for dealing with ambiguous motion correspondences.

1. Portions of Sections 1 and 2 are taken from [8] and are reprinted courtesy of Kluwer Academic Publishers.

• *I.J. Cox is with NEC Research Institute, 4 Independence Way, Princeton, NJ 08540. E-mail: ingemar@research.nj.nec.com.*
• *S.L. Hingorani is with AT&T Bell Laboratories, 184 Liberty Corner Road, Warren, NJ 07059. E-mail: sunita@cartoon.lc.att.com.*

Manuscript received Aug. 25, 1994; revised Aug. 18, 1995.
Recommended for acceptance by: A. Singh.
For information on obtaining reprints of this article, please send e-mail to: transactions@computer.org, and reference IEEECS Log Number P95149.

The target tracking and surveillance community has extensively studied the motion correspondence problem [2] and a number of statistical data association techniques have been developed. These algorithms are now receiving wider attention, especially within the computer vision community [8]. For example, Chang and Aggarwal [6] have applied the joint probabilistic data association (JPDA) filter [18] to the problem of 3-D structure reconstruction from an ego motion sequence. However, the JPDA is only appropriate if the number of tracks is known a priori and remains fixed throughout the motion sequence. Zhang and Faugeras [32] have used the track splitting filter of Smith and Buechler [29] for dynamic motion analysis. The track splitting filter is similar to multiple hypothesis tracking in its use of track trees to delay correspondence decisions until more evidence is available. However, the track splitting filter allows measurements to be shared between tracks. This is physically unrealistic. More reasonable, is that a measurement originates from only a single source feature, e.g., 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 [22] and in motion correspondence it is called the element integrity principle [16]. 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 it is necessary to use an MHT approach. Moreover, one is unable to develop and efficient implementation, as discussed in Section 2.3, without the disjointness constraint.

This paper describes an efficient implementation of the multiple hypothesis tracking (MHT) algorithm originally proposed by Reid [27] and evaluates its usefulness in the context of visual tracking and motion correspondence. Our interest in the MHT is motivated by the fact that the MHT is the only statistical data association algorithm that integrates all the capabilities of

1) *Track Initiation.* The automatic creation of new tracks as new geometric features enter the field of view.
2) *Track Termination.* The automatic termination of a track when the geometric feature is no longer visible for an extended period of time
3) *Track Continuation.* The continuation of a track over several frames in the absence of measurements. Thus, the algorithm is capable of providing a level of support for temporary occlusion.
4) *Explicit Modeling of Spurious Measurements.*
5) *Explicit Modeling of Uniqueness Constraints.* A measurement may only be assigned to a single track and a track may only be the source of a single measurement per frame.

The multiple hypothesis tracking (MHT) algorithm is outlined in Section 2. Unfortunately, the MHT algorithm is computationally exponential both in time and memory. An approximation to the algorithm must therefore be implemented. Section 2.3 describes an efficient approximation to the MHT algorithm, the key contribution being the use of an algorithm due to Murty [24] to generate directly the k-best hypotheses in polynomial time [12] without explicitly enumerating all possible hypotheses. This is a significant contribution to the practical application of the MHT methodology which has recently been shown to be approximately three orders of magnitude faster than previous hypothesis generation strategies [13].

Section 3 then describes experimental results on three motion sequences. In each motion sequence, corner features are automatically detected using a variant of the Lucas and Kanade corner detector [21]. The MHT then tracks these corners over the sequence of frames. Each corner is tracked in the image plane using a simple linear Kalman filter. Section 3.4 demonstrates that the algorithm is robust to errors in the motion model. The most significant experimental problem encountered was that of track initiation during the first two or three frames of the sequence. Section 3.2 describes the approach used to reduce this problem. Section 3.4.1 investigates how the performance of the MHT varies as a function of the depth of the hypothesis tree. Finally, Section 4 summarizes the experimental results and suggests several promising lines of future work.

2 Multiple Hypothesis Algorithm

The multiple hypothesis tracking algorithm was originally developed by Reid [27] in the context of multi-target tracking. Recently, Cox and Leonard [9], [10][2] demonstrated its utility in the context of building and maintaining a map of a mobile robot's environment using acoustic sensors. Fig. 1 outlines the basic operation of the MHT algorithm. An iteration begins with the set of current hypotheses from iteration $(k-1)$. Each hypothesis represents a different set of assignments of measurements to features, i.e., it is a collection of *disjoint tracks*. A track is defined to be a sequence of measurements that are assumed to originate from the same geometric feature. A dummy track in each global hypothesis denotes spurious measurements.

Different sets of assignments expect to see different sets of measurements. Thus, each hypothesis predicts the location (in the image plane) of a set of expected geometric features (specifically corners) and these are compared with actual measurements detected in the next camera frame on the basis of their Mahalanobis distance.[3] These comparisons are represented in the form of an *ambiguity matrix*,[4] defined

2. The interested reader is also directed to Cox et al. [14] who applied the MHT to the problem of contour grouping and segmentation.

3. For normally distributed measurements, 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 γ can, therefore, be obtained from χ^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, i.e., $P(\mathbf{z}(k+1) \in \tilde{V}(\gamma)) = 0.95$, then γ is set to $\gamma = 5.99$. Conversely, if a measurement fails the inequality test then there is a 5% or less chance that it is associated with the geometric feature.

4. The ambiguity matrix is more often referred to as a hypothesis matrix. However, we feel that this is somewhat confusing since many hypotheses can be generated from a single (ambiguity) matrix.

in Section 2.1, which concisely models the ambiguities present in assigning measurements to features.

Each measurement may either 1) belong to a previously known geometric feature, 2) be the start of a new geometric feature, e.g., a previously unseen corner that has entered the field of view of the camera, 3) be a spurious measurement (also called 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 say a corner feature leaves the field of view of the camera. Alternatively, 5) there is the possibility of continuation of a geometric feature, the missed measurement perhaps being due to either noise or a temporary occlusion caused by the motions of the camera and objects in the scene.

After matching, each global hypothesis (from iteration $(k-1)$), has an associated ambiguity matrix from which it is necessary to generate a set of legal assignments (see Section 2.1). Each subsequent child hypothesis represents one possible interpretation of the new set of measurements and, together with its parent hypothesis, represents one possible interpretation of *all* past measurements.

Finally, in order to contain the growth of the tree, it is necessary to prune unlikely branches (see Section 2.3). In order to do this intelligently, we need to evaluate the likelihood of each hypothesis. Section 2.2 provides the mathematical framework for estimating the probability of each leaf in the tree.

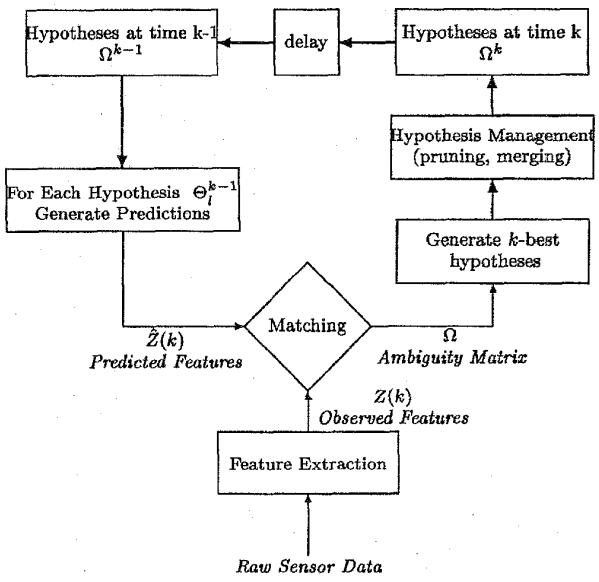

Fig. 1. Outline of the multiple hypothesis algorithm.

2.1 Hypothesis Generation

A particular global hypothesis at time k is defined by Θ_l^k. Let $\Theta_{m(l)}^{k-1}$ denote the parent hypothesis from which Θ_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 Θ_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 $\theta_i(k)$ based on the current measurements is defined to consist of τ measurements from known geometric features, ν measurements from new geometric features, ϕ spurious measurements (false alarms), and χ deleted (or obsolete) geometric features from the parent hypothesis.

A set of current assignments or events $\theta_i(k)$ can be generated by first creating an ambiguity 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 Fig. 2 is represented by the hypothesis matrix shown in Fig. 3.

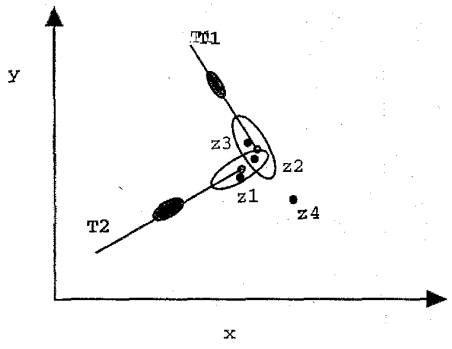

Fig. 2. Predicted target locations and elliptical validation regions for a situation with two known geometric features (T_1 and T_2) and four new measurements ($\mathbf{z}_1(k)$, $\mathbf{z}_2(k)$, $\mathbf{z}_3(k)$ $\mathbf{z}_4(k)$).

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

Fig. 3. Hypothesis matrix for the situation depicted in Fig. 2.

It is desired to constrain the legal set of assignments to be disjoint so that 1) a measurement originates from only one source feature and that 2) a geometric feature has at most one associated measurement per iteration. This is equivalent to restricting an ambiguity 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. If the first and last columns of the ambiguity matrix are replicated m_k times for each of the m_k measurements, then there is only a single nonzero in any row or column and the ambiguity matrix can be thought of as a cost matrix in a linear assignment problem (or weighted bipartite graph matching).

Enumeration of all legal sets of assignments, $\theta_l(k)$, is straightforward [35], but impractical for anything other than a trivial example. Section 2.3.3 describes in more detail how the ambiguity matrix can be modified to represent a classical assignment matrix from which the k-best assignments (hypotheses) can be generated using an algorithm due to Murty [24].

2.2. Probability Calculations

The new hypothesis at time k, Θ_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 $k-1$, i.e.,

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

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

$$\begin{aligned} P\left\{\Theta_l^k \middle| Z^k\right\} &= P\left\{\theta_l(k), \Theta_{m(l)}^{k-1} \middle| Z(k), Z^{k-1}\right\} \\ &= \frac{1}{c} p\left[Z(k) \middle| \theta_l(k), \Theta_{m(l)}^{k-1}, Z^{k-1}\right] \\ &\quad P\left\{\theta_l(k) \middle| \Theta_{m(l)}^{k-1}, Z^{k-1}\right\} P\left\{\Theta_{m(l)}^{k-1} \middle| Z^{k-1}\right\} \end{aligned} \tag{2}$$

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 (2) is obtained by combining results from [2] and [20] to yield

$$\begin{aligned} P\left\{\theta_l(k) \middle| \Theta_{m(l)}^{k-1}, Z^{k-1}\right\} = \frac{\phi! \nu!}{m_k!} \mu_F(\phi) \mu_N \prod_t \\ \left(P_D^t\right)^{\delta_t} \left(1 - P_D^t\right)^{1-\delta_t} \left(P_\chi^t\right)^{\chi_t} \left(1 - P_\chi^t\right)^{1-\chi_t} \end{aligned} \tag{3}$$

where $\mu_F(\phi)$ and $\mu_N(\nu)$ are the prior probability mass functions (PMFs) of the number of spurious measurements and new geometric features, P_D^t and P_χ^t are the probabilities of detection and termination (deletion) of track t and δ_t and χ_t are indicator variables defined by

$$\delta_t \triangleq = \begin{cases} 1 & \text{if geometric feature } t \text{ (in } \Theta_{m(l)}^{k-1}\text{) is detected} \\ & \text{at time } k \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$\chi_t \triangleq = \begin{cases} 1 & \text{if geometric feature } t \text{ (in } \Theta_{m(l)}^{k-1}\text{) is deleted} \\ & \text{at time } k \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

To determine the first term on the right hand side of (2) it is assumed that a measurement $\mathbf{z}_i(k)$ has a Gaussian probability density function (pdf)

$$\begin{aligned} N_{t_i} &= N\left[\mathbf{z}_i(k)\right] \triangleq N\left[\mathbf{z}_i(k); \hat{\mathbf{z}}_i(k|k-1), \mathbf{S}^{t_i}(k)\right] \\ &= \left|2\pi \mathbf{S}^{t_i}(k)\right|^{-\frac{1}{2}} e^{-\frac{1}{2}\{(\mathbf{z}(k)-\hat{\mathbf{z}}(k|k-1))^T \{\mathbf{S}^{t_i}(k)\}^{-1}(\mathbf{z}(k)-\hat{\mathbf{z}}(k|k-1))\}} \end{aligned} \tag{6}$$

if it is associated with geometric feature t_i, where $\hat{z}_i(k|k-1)$ denotes the predicted measurement for geometric feature t_i and $S^{t_i}(k)$ is the associated innovation covariance. If the measurement is spurious (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[5] with pdf V^{-1}. Under these assumptions, we have that

$$\begin{aligned} p[Z(k)|\theta_l(k), \Theta_{m(l)}^{k-1}, Z^{k-1}] &= \prod_{i=1}^{m_k} [N_{t_i}[\mathbf{z}_i(k)]]^{\tau_i} V^{-(1-\tau_i)} \\ &= V^{-\phi-\nu} \prod_{i=1}^{m_k} [N_{t_i}[\mathbf{z}_i(k)]]^{\tau_i} \end{aligned} \tag{7}$$

where τ_i is an indicator variable defined as

$$\tau_i \triangleq = \begin{cases} 1 & \mathbf{z}_i(k) \text{ came from a known geometric feature} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

and ν and ϕ are the total number of new geometric features and false alarms, respectively.

Substituting (7) and (3) into (2) yields the final expression for the conditional probability of an association hypothesis

$$\begin{aligned} P\left\{\Theta_l^k \middle| Z^k\right\} = \frac{1}{c} \frac{\phi! \upsilon!}{m_k} \mu_F(\phi) \mu_N(\upsilon) V^{-\phi-\upsilon} \prod_{i=1}^{m_k} \left[N_{t_i}[\mathbf{z}_i(k)]\right]^{\tau_i} \\ \left\{\prod_t (P_D^t)^{\delta_t} (1 - P_D^t)^{1-\delta_t} (P_\chi^t)^{\chi_t} (1 - P_\chi^t)^{1-\chi_t}\right\} \\ P\left\{\Theta_{l(m)}^{k-1} \middle| Z^{k-1}\right\} \end{aligned} \tag{9}$$

If the number of false alarms and new features are assumed to be Poisson distributed[6] with densities λ_F and λ_N, respectively, then (9) reduces to

$$\begin{aligned} P\left\{\Theta_m^k \middle| Z^k\right\} = \frac{1}{c'} \lambda_N^\upsilon \lambda_F^\phi \prod_{i=1}^{m_k} \left[N_{t_i}[\mathbf{z}_i(k)]\right]^{\tau_i} \\ \left\{\prod_t (P_D^t)^{\delta_t} (1 - P_D^t)^{1-\delta_t} (P_\chi^t)^{\chi_t} (1 - P_\chi^t)^{1-\chi_t}\right\} \\ P\left\{\Theta_{l(m)}^{k-1} \middle| Z^{k-1}\right\} \end{aligned} \tag{10}$$

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

2.3. Implementation[7]

Because of the exponential complexity of the multiple hypothesis approach only an approximation to the MHT algorithm can be implemented. In particular, it is simply not feasible to search the entire space of hypotheses in order to determine the most likely set of assignments. Several implementation strategies were employed in order to contain the growth of the hypothesis tree and reduce the number of hypotheses that must be considered.

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. Portions of this section are taken from [12].

2.3.1 Track Trees

The same track may appear in more than one global hypothesis. Rather than duplicate a track for each hypothesis containing it, thereby incurring additional computational and storage overheads, track trees are formed [20]. Each branch of a track tree represents the assignment of a different measurement to the track. Each global hypothesis then contains pointers to leaves of the track trees. Each set of pointers, i.e., global hypothesis, represents a different permutation of track leaf nodes from different track trees, and enforce the constraints of disjoint partitions. The track tree provides considerable savings and is discussed in detail by Kurien [20]. Track trees also eliminate the need for an explicit hypothesis tree; only the leaf nodes of a hypothesis tree need to be kept—parent hypotheses can be reconstructed by following the set of track tree pointers.

2.3.2 Spatially Disjoint Hypothesis Trees

A considerable reduction in the combinatorics can be achieved by realizing that it is not necessary to form a single global hypothesis tree if there are tracks that do not compete for common measurements. Instead, tracks can be partitioned into separate *clusters* as proposed by Reid [27]. Tracks within each cluster compete for common measurements, whereas tracks in different clusters do not. A separate hypothesis tree is grown for each spatially disjoint region 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.

2.3.3 Generating the k-Best Hypotheses

A brute force implementation of the MHT would, at each iteration, enumerate *all* possible global hypotheses, calculate the probability of each hypotheses and then prune so as to keep only the k-best. This enumeration is impractical. Recently, several researchers have recognized the importance of generating the k-best directly without recourse to a costly enumeration. Nagarajan et al. [25] present an algorithm in which the k-best hypotheses are generated by an "easy search process instead of going through an extensive enumeration." The authors do not provide a theoretical analysis of the computational complexity of their branch and bound scheme. However, while evidence is presented to demonstrate that in some cases a very significant reduction in computation is achieved, in the worst case the cost may still be exponential. Brogan [3] provides an algorithm for determining a ranked set of p assignments. However, this set is not guaranteed to be the p-best, i.e., it is possible to miss certain good combinations. A sufficient condition is provided to determine $q \leq p$ such that the first q assignments are optimal. Once again, no formal analysis of the computational complexity is provided.

Danchick and Newnam [15] recognize that finding the best hypothesis can be formulated as a classical linear assignment problem, and then show how modifications to the cost matrix followed by repeated solutions to these new assignment problems allow the k-best assignments to be computed. Danchick and Newnams' algorithm has two disadvantages. First, in the worst case, Danchick and Newnams' algorithm requires the solution of $k!$ linear assignment problems. Though the average case is expected to be considerably better, it is highly desirable to reduce the order of this dependency. Second, at the end of each iteration (or "sweep") Danchick and Newnams' algorithm must identify and eliminate duplicate assignments. A comparison of an optimized version of the implementation described next with that of Danchick and Newnam revealed that our approach was approximately three orders of magnitude faster [13].[8]

In order to generate the k-best hypotheses, we used an algorithm due to Murty [24] to optimally determine the k-best assignments in polynomial time. The number of linear assignment problems that must then be solved is *linear* in k. In fact, "the computations required at each stage are the solving of at most $(n-1)$ assignment problems, each of sizes 2, 3, ..., n" [24]. The algorithm avoids solving duplicate assignment problems, thereby eliminating the need to compare and delete duplicate hypotheses. Finally, in the average case, the dimension of the assignment problems that must be examined decreases with k.

Consider first the problem of finding the single most probable hypothesis. This can be cast as a weighted bipartite matching problem by constructing a bipartite graph in which each node on one side represents one of the measurements, each node on the other represents one of the targets, and each arc, $< z_i, t_j, l >$, gives the log likelihood, l, that measurement z_i should be assigned to target t_j. The log of the likelihood of a given assignment can be found by summing the log likelihoods of all the arcs that it specifies. These log likelihoods can be calculated from (10).

Finding the best hypothesis, then, is a matter of finding the assignment that maximizes this sum. This is an instance of the classical assignment problem from combinatorial optimization, and can be solved very efficiently in polynomial time [26]. Murty's algorithm is also guaranteed to find the k-best assignments in polynomial time. A brief description of Murty's algorithm follows:

Given a solution, S, to an assignment problem, P, we can partition the assignment problem into a list of new problems with the following properties:

1) The set of valid solutions for any one of the problems in the list doesn't intersect with the set of solutions for any other problem in the list. That is, there are no duplicate problems.
2) The union of the sets of valid solutions for all the problems in the list is exactly the set of solutions for problem P, minus solution S.

Murty gives a method for computing this partitioning in $O(N^2)$ time.

8. To generate the 20 best hypotheses for 20 problems of dimension 20×20 in which the matrix weights are randomly generated.

For the k-best algorithm, a list of problem/solution pairs is kept. Each pair consists of an assignment problem and its best solution. The list is initialized with the initial problem to be solved. In each iteration, the best solution is found, then removed from the list, and replaced with its partitioning. So, in the first iteration, the single best solution, S_0, is found to the problem, and the list is altered so the set of possible solutions no-longer contains S_0. The next iteration gives the next-best solution, S_1, and changes the list so that possible subsequent solutions no-longer include S_1 or S_0; and so on. Fig. 4 outlines the algorithm. The partitioning is performed by the loop in step 4.4.

The reader is directed to [24], [12], [13] for more detail.

1) Find the best solution, S_0, to P_0 (this can be done using a standard algorithm like the Hungarian method)
2) Initialize the list of problem/solution pairs with $< P_0, S_0 >$
3) Clear the list of solutions to be returned
4) For $i = 1$ to k, or until the list of problem/solution pairs is empty
   4.1 Search through the list of problem/solution pairs, and find the pair, $< P, S >$ that has the best solution value
   4.2 Remove $< P, S >$ from the list of problem/solution pairs
   4.3 Add S to the list of solutions to be returned
   4.4 For each triple, $< t, z, l >$, found in S
      4.4.1 Let $P' = P$
      4.4.2 Remove the triple $< t,z, l >$ from P'
      4.4.3 Look for the best solution, S', to P'
      4.4.4 If S' exists
         4.4.4.1 Add $< P', S' >$ to the set of problem/ solution pairs.
      4.4.5 From P, remove all triples that include t, and all triples that include z, except $< t, z, l >$ itself. (This reduces the dimension of the problem by one)

Fig. 4. Murty's algorithm for finding the k-best solutions to an assignment problem, P_0.

2.3.4 Pruning

Pruning is essential to any practical implementation of this algorithm. Pruning is based on a combination of an "N-scan-back" algorithm [20] and ratio pruning, i.e., a simple lower limit on the ratio of the probabilities of the current and best hypotheses.

The "N-scan-back" algorithm assumes that any ambiguity at time k is resolved by time $k + N$, i.e., it defines the number of frames to look ahead in order to resolve an ambiguity. Then, if hypothesis Θ_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. It is clearly computationally advantageous to set N as small as possible. Section 3.4.1 investigates how the performance of the MHT varies as a function of N. Results suggest that very good performance can be obtained for $N = 3$ and even $N = 2$ and this conclusion is supported by other work [9], [10], [20].

Generating the k-best hypotheses obviously restricts the maximum number of new hypotheses to k. However, in many situations there may be little need to consider all k hypotheses particularly if there is little or no ambiguity. Moreover, hypotheses only have a finite number of iterations to increase their probabilities, before N-scan back pruning deletes them. In these circumstances there is little point in generating all k hypotheses if they are such that their probabilities are especially low. A threshold can therefore be set that prevents hypotheses from being considered if the ratio of their probability to that of the best hypothesis becomes too small.

3 Experimental Results

The MHT algorithm was tested on three image sequences: the PUMA sequence and the Toycar sequence, both from the University of Massachusetts, and the the J7 outdoor sequence from IRISA, France. For each sequence, corners were automatically extracted from each image frame using a variant of the Lucas and Kanade [21] corner detector. This method looks for square regions in the image that can be found most accurately through correlation by examining the eigenvalues of a 2×2 matrix representing the intensities inside the window. If both eigenvalues are high, the intensity profile changes rapidly when the window shifts in any direction.[9] The threshold was set so that approximately 200 corners were located in each frame of the PUMA sequence and 55 corners were located in each frame of the Toycar sequence. For the J7 sequence, we used the corners extracted by Zheng and Chellappa [34]. Table 1 lists the MHT parameter values used for all three sequences.

TABLE 1
PARAMETER VALUES FOR THE MHT ALGORITHM

Probability of detection	P_D	0.999
Termination likelihood	λ	20
Mean rate of false alarms	λ_F	0.00002
Mean rate of new tracks	λ_N	0.004
Depth of tree	N-scan	3
Maximum number of hypotheses	H_{max}	300
Ratio pruning		0.001

3.1 Kalman Filter

Each corner feature was tracked in the image plane using a simple linear Kalman filter with state vector $\mathbf{x} = [x\ \dot{x}\ y\ \dot{y}]'$, where x and y are the pixel coordinates of a feature. The state transition matrix, $\mathbf{F}$, is given by

$$\mathbf{F}(k) = \begin{pmatrix} 1 & \delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \delta t \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

9. Our implementation used source code kindly provided by J. Barron.

The measurement vector $\mathbf{z}(k) = [x\ y]'$ and the observation matrix $\mathbf{H}$ is given by

$$\mathbf{H}(k) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

In order to estimate the process and measurement noises, we manually tracked a few corners over approximately 10 frames. We then adjusted the process and measurement noises so that all the measurements passed the Mahalanobis matching test. If this is not done, then of course, a measurements will not validate to the correct track and will therefore be incorrectly assigned. The process noise, $\mathbf{S}(k)$ and measurement noise $\mathbf{R}(k)$ were set to

$$\mathbf{S}(k) = \begin{pmatrix} dt^3/3 & dt^2/2 & 0 & 0 \\ dt^2/2 & dt & 0 & 0 \\ 0 & 0 & dt^3/3 & dt^2/2 \\ 0 & 0 & dt^2/2 & dt \end{pmatrix} q$$

where $q = 9$ for the Puma and J7 sequences and $q = 0.5$ for the Toycar sequence and

$$\mathbf{R}(k) = \begin{pmatrix} 1.0 & 0 \\ 0 & 1.0 \end{pmatrix}$$

3.2 Track Initiation

Although track initiation is handled automatically within the MHT framework, there is still the problem that the Kalman filter associated with each track cannot be initiated from a *single* measurement since a single measurement does not provide velocity information. There are two solutions to this problem. The first is to delay track initiation until two consecutive measurements are available to give a reliable estimate of a feature's velocity. The second solution is to initiate the velocity estimates of the state vector to zero while simultaneously initializing the corresponding elements of the state covariance matrix to a large value in order to represent the uncertainty in the velocity estimates. We chose to follow the second approach and initialized the state covariance, $\mathbf{P}(k)$, to

$$\mathbf{P}(k) = \begin{pmatrix} 1.0 & 0 & 0 & 0 \\ 0 & 200 & 0 & 0 \\ 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 200 \end{pmatrix}$$

for the puma and toy car sequences and

$$\mathbf{P}(k) = \begin{pmatrix} 1.0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 10 \end{pmatrix}$$

for the J7 sequence

The magnitude of the velocity variances was established by manually examining two consecutive image frames to determine the maximum displacement between two corresponding points. For the PUMA sequence, this displacement was typically 31 pixels for features closest to the camera. This is a considerable displacement which necessitated examining a very large initial window.

3.3 Validation/Matching

As noted earlier, measurements were matched to predictions based on the Mahalanobis test. However, because of the very large initial search window, ±31 pixels, there were a very large number of possible matches between a prediction and current measurements. This resulted in very large ambiguity matrices and very few disjoint clusters which caused significant computational problems.

In order to reduce the total number of possible initial matches (and also increase the number of disjoint clusters) we supplemented the Mahalanobis test with a cross correlation test[10] in order to prevent nonsense matches, such as matching a black corner with a white corner. The 3×3 neighborhood of intensities centered at a corner in frame $(k-1)$, $I_{k-1}(i, j)$, were compared with the 5×5 neighborhood of intensities in frame k, $I_k(i, j)$ such that

$$\lambda = \max_{p,q=-1,0,1} \frac{\sum_{i,j \in \mathcal{N}} (I_{k-1}(i,j) - \bar{I}_{k-1})(I_k(i-p, j-q) - \bar{I}_k)}{\sqrt{\sum_{i,j \in \mathcal{N}} (I_{k-1}(i,j) - \bar{I}_{k-1})^2} \sqrt{\sum_{i,j \in \mathcal{N}} (I_{k-1}(i,j) - \bar{I}_k)^2}}$$

where $\mathcal{N}$ is the 3×3 neighborhood and $\bar{I}$ is the mean of I. A threshold was then set on the maximum cross correlation coefficient, i.e., if two corners passed the original Mahalanobis test but failed the cross correlation test then the two corners did not validate/match. A threshold of 0.9 was used for the PUMA and Toycar sequences and 0.05 for the J7 sequence.

This cross correlation technique significantly improved the performance of the algorithm on the PUMA sequence, eliminating many (erroneous) matches from consideration. However, the same test caused a few problems with the "Toycar" sequence due to occlusion, see Section 3.5. Zheng and Chellappa [34] use a weighted correlation technique which they claim is robust to "feature mutation." Such an approach could also be used within the framework described here. However, it is unclear whether the weighted correlation technique would be more robust to rapid changes in background as objects partially occlude one another.

3.4 The PUMA Sequence

Fig. 5 shows the 1st, 10th, 20th, and 29th frames of the PUMA sequence. The extracted corners are overlayed on each frame. Fig. 6 shows those trajectories that were tracked from frame 1 and additional trajectories that began in frame 2. Only trajectories of length greater than 6 are displayed. The square and circle symbols denote the start and end of a track, respectively. The results are qualitatively very good. In particular, it should be noted that these circular trajectories were tracked despite the the underlying constant velocity motion model. Note, however, the two erroneous trajectories in the top right of Fig. 6a. Detailed examination of this area revealed that the constant velocity model was responsible for the erroneous classification. When a constant

10. Shapiro et al., [28] call this the *product moment coefficient*. They point out that such a measure is invariant to linear changes in intensity and therefore compares the structure of the patches rather than their absolute intensities.

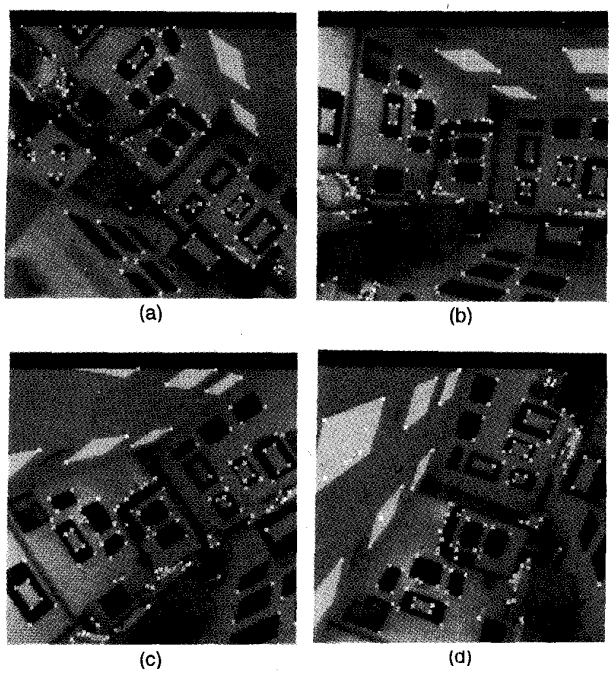

Fig. 5. The (a) 1st, (b) 10th, (c) 20th, and (d) 29th frames of the PUMA sequence; courtesy of the University of Massachusetts.

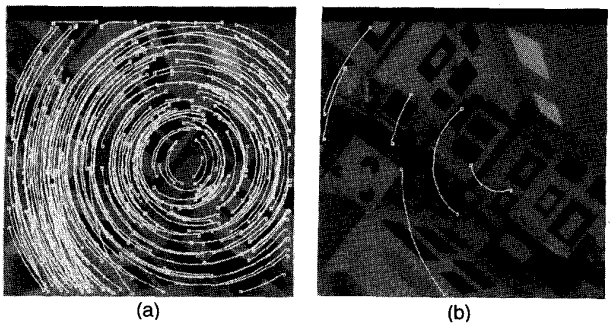

Fig. 6. Corner trajectories tracked from (a) frame 1 and (b) from frame 2 of the PUMA sequence. Only trajectories of length greater than 6 are displayed.

acceleration model was used, the measurements were tracked correctly.

Fig. 6b shows tracks that were started in frame 2. While many of these corners were visible in the first frame, they were either 1) not located by the corner detector until the second frame or 2) the corresponding corners in the 1st frame did not validate because intensity variations between frames caused the cross correlation test to fail. Fig. 7 shows the computation time and number of measurements for each frame of the Puma sequence. Although there are significant variations in the number of measurements per iteration, the computational time per frame is approximately constant at 1.5 sec per frame on a MIPS R4400 150 MHz processor.

3.4.1 Performance as a Function of Tree Depth

The classifications of Fig. 6 were obtained with an "N-scan" of 3. In order to investigate how the performance of the MHT varies with the depth of the tree, the experiment was repeated for N-scan depths of 0, 1, and 2. Note that an N-scan of 0 provides no look ahead capability and is similar to a nearest neighbor solution to the assignment problem.

The resulting tracks are shown in Fig. 8, including that of Fig. 6a for reference. Notice that for an N-scan of 0, 1, and 2 there are several erroneous straight line trajectories in the lower left quandrant of the image. Experimental results showed no perceptable improvement for N-scans of greater than 3, supporting earlier claims [27], [20], [11] that near optimum performance can be obtained from quite shallow tree depths.

3.5 The "Toycar" Sequence

Fig. 9 shows the 1st, 3rd, 5th, and 7th frames of a nine frame sequence in which two vehicles are moving from left to right, a vehicle is moving from right to left and a fourth vehicle is stationary for seven frames and then moves to the bottom left quadrant in the eighth frame. This latter motion is not tracked.

Fig. 10 shows the trajectories of the tracked corners that are started in frames 1 and frame 2. While most of the tracks started in frame 2 are visible in frame 1, once again they were either not located by the corner detector until the second frame or the corresponding corners in the first frame did not validate because intensity variations between frames caused the cross correlation test to fail. Several of the tracks associated with the van were temporarily occluded due to the motion of the "jeep." These occlusions were successfully handled by the MHT algorithm, which continued the tracks (despite missed measurements) until the tracks were visible again. Of course, had a track been occluded for longer, then the MHT might well have terminated it and started a new track for the feature when it became visible again.

Failing to validate a correct measurement based on the cross correlation coefficient occurred in this sequence as well. Significantly, the cross correlation measure provided a very poor method of matching when objects partially occlude one another, as in the case of the "jeep" passing in front of the van, because of the very significant changes in the 3×3 intensity neighborhood.

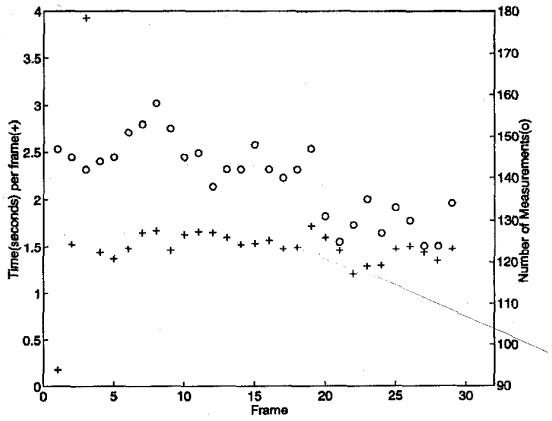

Fig. 7. Time and number of measurements per frame for the MHT algorithm applied to the Puma sequence.

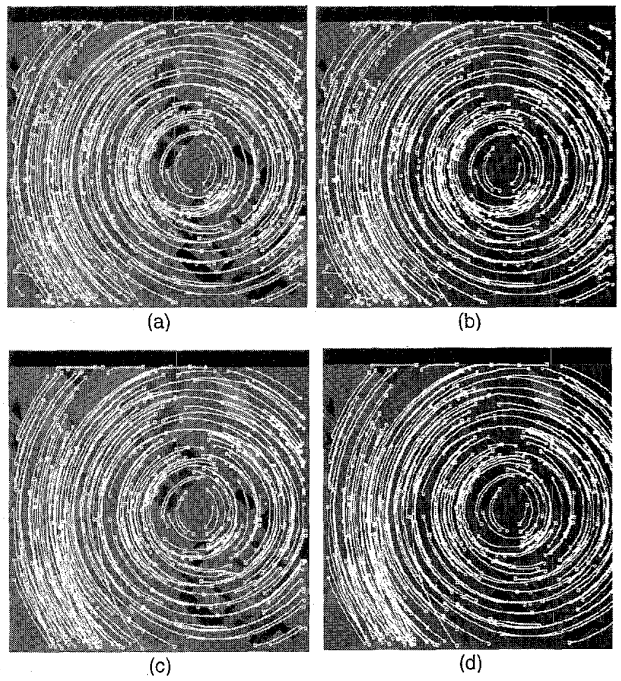

Fig. 8. The corner trajectories tracked through the PUMA sequence for N-scan of (a) 0, (b) 1, (c) 2, and (d) 3.

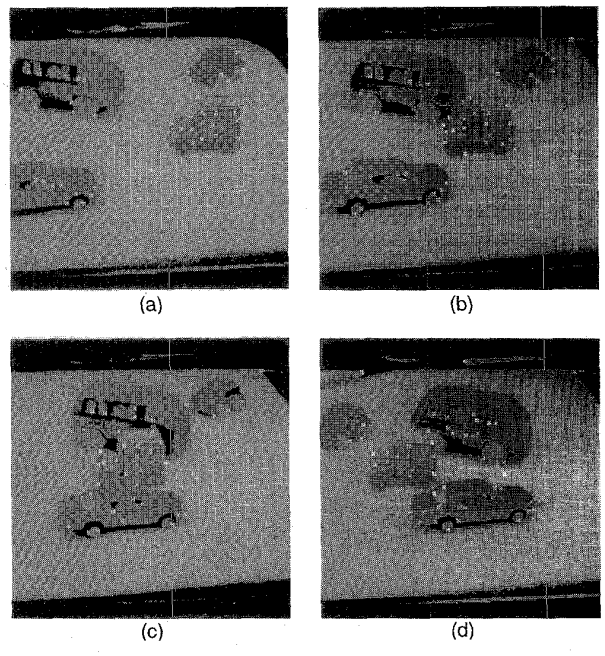

Fig. 9. The (a) 1st, (b) 3rd, (c) 5th, and (d) 7th frames of the Toycar sequence; courtesy of the University of Massachusetts.

The average time-per-iteration of the MHT was 3.06 seconds on the Toycar sequence. However, it should be noted that almost half the total computational time is spent processing frames 5 and 6 during which the vehicles are passing in front of one another, see Fig. 11. The increase in computation time for frames 5 and 6 is not due to an increase in the number of measurements but rather to the motion correspondence ambiguity that arises during the partial occlusion of objects.

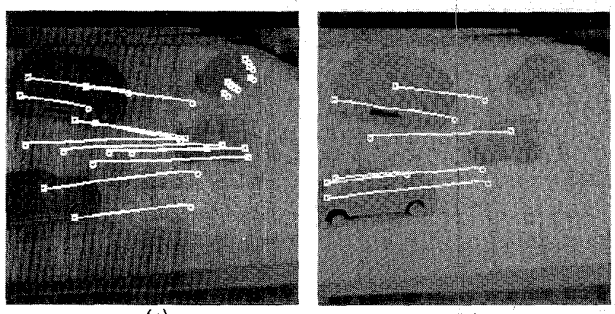

Fig. 10. Corner trajectories tracked from (a) frame 1 and (b) from frame 2 of the Toycar sequence. Only trajectories of length greater than 6 are displayed.

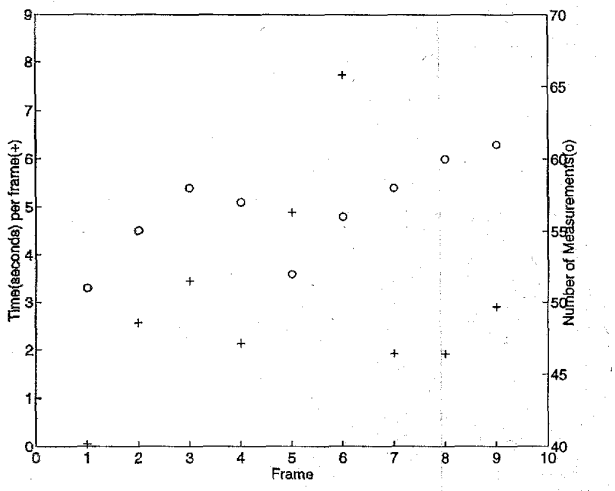

Fig. 11. Time and number of measurements per frame for the MHT algorithm applied to the Toycar sequence.

3.6 The "J7" Outdoor Sequence

Fig. 12 shows the 5th, 22nd, 39th and 55th frames of a 60 frame sequence in which the camera is mounted in a moving vehicle and is following behind a van. For approximately half the sequence, the camera is approaching the van and for the remainder, the van is receding. This results in tracks whose direction reverse. Fig. 13 shows the trajectories of the tracked corners that started in frame 1 and were tracked to frames 22 and 55, respectively. The relatively large process noise allows the Kalman filter to cope with the change in direction of the tracks. Note that the cross correlation threshold was set to 0.05, effectively switching off this gating mechanism and relying almost exclusively on the Mahalanobis test.

Comparison with the results of Zheng and Chellappa [34] show very few differences. Of the 100 tracks, 74 tracks were identical, 18 tracks that contained only one measurement were classified as false alarms by the MHT, five tracks with only three measurements in each differed somewhat but again, these tracks are not significant. Of the tracks con-

taining greater than three measurements there were only three tracks that differed; a single measurement in each track was assigned as a false alarm. These three tracks were following: 1) the shadow of a car, 2) leaves of a tree, and 3) a corner created by the occlusion of a pole by a van.

The time to process the 51 frame sequence was approximately 7 sec. Fig. 14 shows the computation time and number of measurements per frame for the J7 sequence. The rapid increase in the computation time during the first three frames reflects the large uncertainty at startup due to track initiation. At the end of the third iteration the N-scan back pruning ($N = 3$) removes many of the hypotheses as decisions are made as to the assignments of measurements to tracks. The gradual reduction in the time-per-iteration particularly between time $t = 3$ and $t = 15$ is because the number of measurements per iteration is monotonically decreasing with time (see Fig. 14).

To compare these results with an (approximately) nearest neighbor strategy, we reduced the N-scan lookahead depth to zero. In this case, 108 tracks were found with no false alarms, compared with 80 tracks and 28 false alarms for N-scan of three. An examination of the structure of the tracks revealed 45 tracks with fewer than four measurements that were subsequently extrapolated (continued) for 15 iterations, as shown in Fig. 15. Most of these tracks contained only one or two actual measurements. The circles visible in Fig. 15 are indicative of tracks with zero velocity. The small duration tracks visible on the right side of Fig. 15 and in the mid left were typically initialized with only measurements from three consecutive frames yet persist for significantly longer based on extrapolation over 15 frames. These tracks do not exist in practice and are evidence of the inferior data associations obtained using a nearest neighbor strategy.

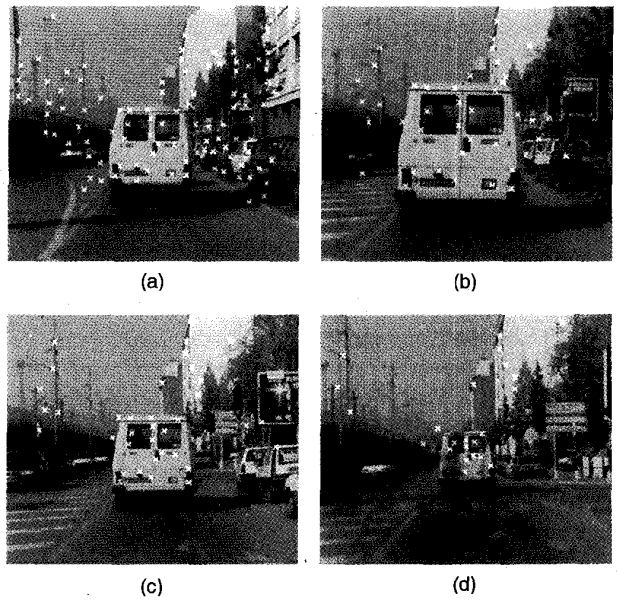

Fig. 12. The (a) 1st, (b) 22nd, (c) 39th, and (d) 55th frames of the "J7" sequence, courtesy of F. Meyer, IRISA, France and Thomson LER, Cesson-Sevigne, France.

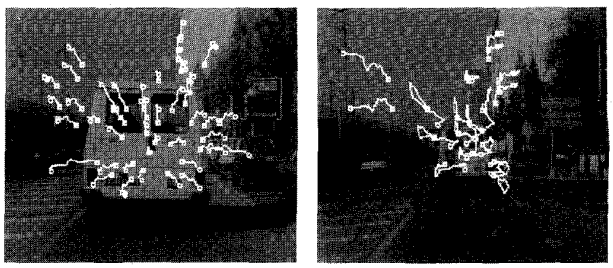

Fig. 13. The corner trajectories tracked through the "J7" sequence. (a) is tracks up to Frame 22, (b) is tracks up to Frame 55.

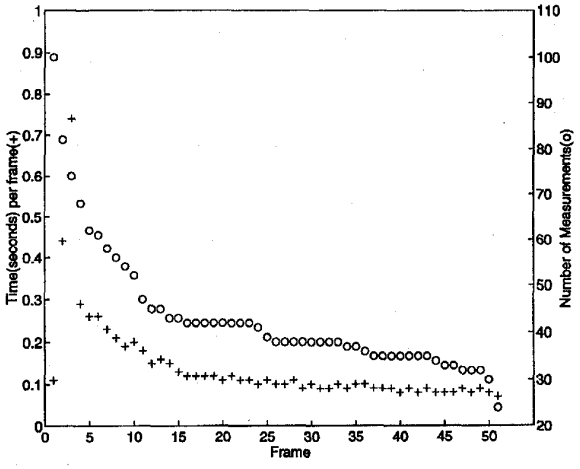

Fig. 14. Time and number of measurements per frame for the MHT algorithm applied to the J7 sequence.

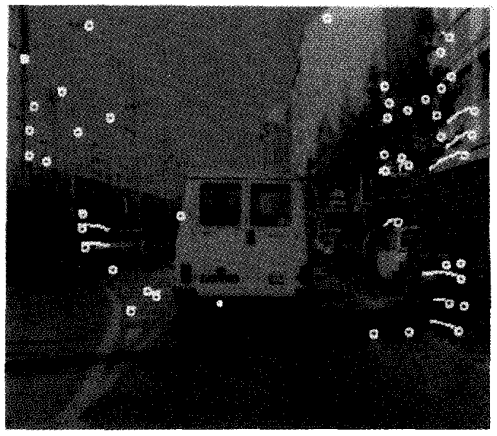

Fig. 15. Corner trajectories with 4 or less actual measurements tracked through the "J7" sequence up to Frame 55 with N-scan lookahead set to zero.

3.7. Discussion of Results

Table 2 tabulates certain MHT run-time statistics for each of the three motion sequences. This data indicates that the run-time of the MHT algorithm is less affected by the total number of measurements and/or tracks, c.f. the Puma and Toycar sequences, and more by the degree of ambiguity present in the measurements. Note too, that for the PUMA

and J7 sequences, 66% and 47%, respectively, of the total comutation time is spent in the validation phase (tentative matching) of the algorithm. This could be significantly sped up by using metric [7] or VP trees [31].

Tables 3, 4, and 5 tabulate for each motion sequence, the number of disjoint clusters, the total number of hypotheses and the maximum number of hypotheses in a cluster for each frame of a sequence. The number of disjoint clusters is approximately equal to the number of measurements, for all three sequences. However, Table 5, for the J7 sequence, reveals that both the total number of global hypotheses as well as the maximum number of hypotheses in a cluster, are very small, indicating that the image sequence contains few motion correspondence ambiguities. The Puma sequence, Table 3, has significantly more global hypotheses. The fluctuations in the maximum number of hypotheses in a cluster indicates various degrees of motion correspondence ambiguity. Finally, Table 4 indicates via the total number of hypotheses and the large maximum number of hypotheses within a cluster, that the Toycar sequence contains significant ambiguity. This ambiguity and the resultant large number of hypotheses that must be considered, cause the processing time to increase for the Toycar sequence.

Tables 3 and 4 suggests that an MHT framework could run in real-time for certain classes of video sequences, though identifying such a sequence a priori is not entirely straightforward. In both the PUMA and J7 sequences a very significant amount of time is spent in the validation (tentative matching) phase of the algorithm, see Table 2. As noted earlier, algorithms exist to significantly speedup this phase [7], [31]. Moreover, the implementation of Murty's algorithm for hypothesis generation could be improved significantly and the entire MHT algorithm is amenable to parallelization [20].

4 Conclusion

We have demonstrated how the multiple hypothesis tracking algorithm of Reid may be applied to visual tracking. The MHT algorithm provides a Bayesian framework for motion analysis. In particular, it is the only statistical data association algorithm to explicitly model track initiation and termination, spurious measurements and track continuation. The latter characteristic provides a low level mechanism for dealing with temporary occlusions. Moreover, the algorithm enforces disjoint constraints so that a measurement can only be associated with one feature and a feature can only be the source of a single measurement each iteration.

The principal disadvantage of the MHT is its computational complexity. This paper describes a significant contribution to the design of an efficient implementation of the MHT—the use of Murty's algorithm to generate the k-best hypotheses (in order $O(N^4)$ time, worst case,) thereby avoiding enumerating many unnecessary hypotheses. We expect Murty's algorithm to become the hypothesis generation strategy of choice for many MHT applications. An optimized version of Murty's algorithm has been demonstrated to be three orders of magnitude faster than the best alternative algorithm. Moreover, experimental results indicate that a real-time implementation of the MHT to motion correspondence is feasible for certain classes of scene.

TABLE 2
Comparison of MHT Run-Time Statistics for the Three Sequences

	PUMA	Toycar	J7
Average time-per-iteration	1.54	3.06	0.14
Average number of tracks	390	165	77
Average number of measurements per frame	140	57	44
Total number of frames	30	9	51
Percentage of time in hypothesis generation and pruning	12.8	70	9.4
Percentage of time in validation phase	66%	11%	47%

TABLE 3
Run-Time Statistics for the Puma Sequence

Frame	Time (sec)	No. of Measurements	No. of Groups	No. of Hypotheses	No. of Hypos in a Group (max)
1	0.18	147	147	294	2
2	1.53	145	124	681	96
3	3.93	142	97	183	19
4	1.44	144	157	241	20
5	1.37	145	178	263	5
6	1.48	151	183	289	13
7	1.65	153	170	263	8
8	1.67	158	170	237	9
9	1.46	152	179	266	10
10	1.63	145	170	277	13
11	1.66	146	181	286	5
12	1.65	138	172	283	10
13	1.60	142	162	262	9
14	1.53	142	159	237	15
15	1.54	148	161	244	21
16	1.57	142	158	253	19
17	1.48	140	159	235	8
18	1.50	142	160	279	34
19	1.72	147	167	274	14
20	1.60	131	158	296	33
21	1.46	125	155	236	7
22	1.21	129	151	226	6
23	1.29	135	161	263	26
24	1.30	127	150	247	11
25	1.48	133	146	256	32
26	1.51	130	143	265	50
27	1.44	124	132	217	17
28	1.35	124	138	225	17
29	1.48	134	132	227	26

TABLE 4
Run-Time Statistics for the Toycar Sequence

Frame	Time (sec)	No. of Measurements	No. of Groups	No. of Hypotheses	No. of Hypos in a Group (max)
1	0.05	51	51	102	2
2	2.56	55	33	720	300
3	3.45	58	33	126	27
4	2.13	57	55	374	254
5	4.89	52	65	197	67
6	7.74	56	62	247	94
7	1.92	58	74	325	95
8	1.92	60	83	536	298
9	2.90	61	82	505	292

TABLE 5
RUN-TIME STATISTICS FOR THE J7 SEQUENCE

Frame	Time (sec)	No. of Measurements	No. of Groups	No. of Hypotheses	No. of Hypos in a Group (max)
1	0.11	100	100	200	2
2	0.44	82	96	209	5
3	0.74	74	76	82	3
4	0.29	68	80	90	4
5	0.26	62	79	98	4
6	0.26	61	72	88	3
7	0.23	58	67	80	3
8	0.21	56	59	65	3
9	0.19	54	59	70	4
10	0.20	52	55	67	5
11	0.18	47	56	64	3
12	0.15	45	51	65	4
13	0.16	45	50	64	3
14	0.15	43	46	52	3
15	0.13	43	44	47	2
16	0.12	42	44	49	3
17	0.12	42	42	44	2
18	0.12	42	42	45	3
19	0.12	42	41	41	1
20	0.11	42	42	42	1
21	0.12	42	42	42	1
22	0.11	42	42	42	1
23	0.11	42	42	42	1
24	0.10	41	42	42	1
25	0.11	39	42	43	2
26	0.10	38	42	46	3
27	0.10	38	41	46	3
28	0.11	38	39	41	3
29	0.09	38	38	38	1
30	0.10	38	38	38	1
31	0.09	38	38	38	1
32	0.09	38	38	38	1
33	0.10	38	39	40	2
34	0.09	37	38	41	4
35	0.10	37	36	37	2
36	0.10	36	38	40	3
37	0.09	35	36	37	2
38	0.09	35	34	37	3
39	0.09	35	34	36	3
40	0.08	35	34	34	1
41	0.09	35	33	33	1
42	0.08	35	34	34	1
43	0.09	35	33	33	1
44	0.08	34	32	32	1
45	0.08	33	31	33	2
46	0.08	33	31	34	3
47	0.09	32	30	32	2
48	0.08	32	30	36	3
49	0.09	32	26	31	3
50	0.08	30	26	29	2
51	0.07	24	27	33	3

Experimental results support the belief that motion correspondence accuracy can be improved by examining more than just the current frame. At the same time, there appears to be little or no further improvement in looking beyond three consecutive frames. This is encouraging, since the depth of the hypothesis tree is quite shallow.

Examination of the k-best hypotheses is predicated on the assumption that the correct hypotheses is contained therein. Typically, one would want and expect the probability of a hypothesis to fall off quickly as a function of k. This is usually the case, but not at the beginning of a motion sequence. Since there are no predictions for measurements in the first frame, all such measurements must be considered either new tracks or spurious measurements. The probabilities associated with each hypothesis are quite flat and there is a high risk of pruning (or not examining) the correct hypothesis. In the second frame, this problem is compounded by the fact that the validation volume is much larger than in the steady state situation, in order to compensate for the lack of velocity information. This increases the motion correspondence (data association) uncertainty. Thus, k may need to be quite large in order to be confident that the correct correspondence was not pruned.

In order to reduce the data association uncertainty (and thereby keep k manageable), we introduced a second gating mechanism based on the cross correlation coefficient of a 3×3 neighborhood centered on the corner position. This significantly reduced the number of possible matches and was beneficial for the most part. However, several correct correspondences failed the cross correlation test, particularly when one object moves in front of another, e.g., when the "van" passed behind the "jeep," since a significant change within the 3×3 intensity neighborhood can occur. This suggests that other methods are needed for tentatively matching features. One such possibility would be some form of 2D matching that incorporated geometric constraints and perhaps included a low level perceptual grouping strategy that (attempted to) identify and group features originating from a common rigid object in a fashion similar to Jacobs [19] or Meyer and Bouthemy [23].

Reliably detecting corners was surprisingly difficult. Corner detection was applied independent of the tracking algorithm, but a coupled feature detection and tracking mechanism, perhaps along the lines of Zheng and Chellapa [33], [34], should be investigated.

The corners were tracked using simple linear Kalman filters. Tuning the various parameters, e.g., process and measurement noise, was straightforward once a few tracks had been manually tracked for several frames. The PUMA and J7 sequences demonstrated that (with sufficient process noise) the algorithm is robust to errors in the motion model. Nevertheless, an accurate motion model is desirable to minimize incorrect motion correspondences. The MHT framework allows several motion models to run in parallel, i.e., instead of a single new track, there can be n possible new tracks, one for every motion model. This has been used in the past to deal with manouvering aircraft and in robot map making [11]. Of course, this increased flexibility comes at the expense on increased combinatorial complexity.

Finally, the MHT framework integrates both a top-down expectation level process, based on predictions from multiple Kalman filters, together with a bottom-up explanation driven process in the form of a Bayesian hypothesis tree. Such a framework appears to be well suited for active vision applications in which sensing is directed to resolve ambiguities in the hypothesis tree.

ACKNOWLEDGMENTS

It is a pleasure to thank Yaakov Bar-Shalom of the University of Connecticutt, Takeo Kanade of CMU, and John J.

Leonard of MIT for many fruitful discussions. Thanks to Rama Chellappa and Qinfen Zheng of the University of Maryland for providing the motion sequence of Figs. 5 and 9 and the trajectory data of the "J7" sequence. Thanks to F. Meyer, IRISA, France and Thomson LER, Cesson-Sevigne, France, for the "J7" motion sequence. Special thanks to David W. Jacobs and Chakra Chennubhotla of NECI for their help and suggestions. An abridged version of this paper appeared in the International Conference on Pattern Recognition, pp. 437-443, 1994.

REFERENCES

[1] N. Ayache and O. Faugeras, "Maintaining representations of the environment of a mobile robot," *IEEE Trans. Robotics and Automation*, vol. 5, no. 6, pp. 804-819, 1989.
[2] Y. Bar-Shalom and T.E. Fortmann, *Tracking and Data Association*. Academic Press, 1988.
[3] W.L. Brogan, "Algorithm for ranked assignments with applications to multiobject tracking," *IEEE J. of Guidance*, vol. 12, no. 3, pp. 357-364, 1989.
[4] T.J. Broida, S. Chandrashekhar, and R. Chellappa, "Recursive 3-d motion estimation from a monocular image sequence," *IEEE Trans. Aerospace and Electronic Systems*, vol. 26, no. 4, pp. 639-656, 1990.
[5] T.J. Broida and R. Chellappa, "Kinematics and structure of a rigid object from a sequence of noisy images," *Proc. Workshop on Motion: Representation and Analysis*, pp. 95-100, 1986.
[6] Y.L. Chang and J.K. Aggarwal, "3d structure reconstruction from an ego motion sequence using statistical estimation and detection theory," *Proc. IEEE Workshop on Visual Motion*, pp. 268-273, 1991.
[7] J.B. Collins and J.K. Uhlmann, "Efficient gating in data associaton with multivariate distributed states," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 28, no. 3, 1992.
[8] 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.
[9] I.J. Cox and J.J. Leonard, "Probabilistic data association for dynamic world modeling: A multiple hypothesis approach," *Proc. Int'l Conf. Advanced Robotics*, Pisa, Italy, 1991.
[10] I.J. Cox and J.J. Leonard, "Unsupervised learning for mobile robot navigation using probabilistic data association," *Proc. Workshop on Computer Learning and Natural Learning*, Berkeley, Calif. 1991.
[11] I.J. Cox and J.J. Leonard, "Modeling a dynamic environment using a multiple hypothesis approach," *J. of A.I.*, vol. 66, no. 1, pp. 311-344, 1994.
[12] I.J. Cox and M.L. Miller, "On finding ranked assignments with application to multi-target tracking and motion correspondence," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 32, no. 1, pp. 486-489, 1995.
[13] I.J. Cox, M.L. Miller, R. Danchick, and G.E. Newnam, "A comparison of two algorithms for determining ranked assignments with application to multi-target tracking and motion correspondence," Tech. Report, NEC Research Inst., 1995.
[14] I.J. Cox, J.M. Rehg, and S. Hingorani, "A Bayesian multiple hypothesis approach to contour grouping and segmentation," *Int'l J. of Computer Vision*, vol. 11, no. 1, pp. 5-24, 1993.
[15] 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, 1993.
[16] M.R.W. Dawson, "The how and why of what went where in apparent motion: Modeling solutions to the motion correspondence problem," *Psychological Review*, vol. 98, no. 4, pp. 569-603, 1991.
[17] R.Deriche and O. Faugeras, *Tracking Line Segments*. O. Faugeras, ed., *Proc. European Conf. on Computer Vision*, pp. 259-268, Springer-Verlag, 1990.
[18] T.E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE J. of Oceanic Engineering*, vol. 8, no. 3, pp. 173-184, 1983.
[19] D.W. Jacobs, "Grouping for recognition," AI memo 1177, MIT, 1989.
[20] T. Kurien, "Issues in the design of practical multitrget tracking algorithms," Y. Bar-Shalom, ed., *Multitarget-Multisensor Tracking: Advanced Applications*. pp. 43-83, Artech House, 1990.
[21] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proc. Seventh Int'l Joint Conf. on Artificial Intelligence*, pp. 674-679, 1981.
[22] J.E.W. Mayhew and J.P. Frisby, "Psychophysical and computational studies towards a theory of human stereopsis," *Artifical Intelligence*, vol. 17, 1981.
[23] F. Meyer and P. Bouthemy, "Region-based tracking in an image sequence," *European Conf. on Computer Vision*, pp. 476-484, 1992.
[24] K.G. Murty, "An algorithm for ranking all the assignments in order of increasing cost," *Operations Research*, vol. 16, pp. 682-687, 1968.
[25] V. Nagarajan, M.R. Chideambara, and R.N. Sharma, "Combinatorial problems in multitarget tracking—a comprehensive survey," *IEE Proc., Pt F*, vol. 134, no. 1, pp. 113-118, 1987.
[26] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization*. Prentice Hall, 1982.
[27] D.B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. on Automatic Control*, vol. 24, no. 6, pp. 843-854, Dec. 1979.
[28] L.S. Shapiro, H. Wang, and J.M. Brady, "A matching and tracking strategy for independently moving objects," *Proc. British Machine Vision Conf.*, pp. 306-315, 1992.
[29] P. Smith and G. Buechler, "A branching algorithm for discriminating and tracking multiple objects," *IEEE Trans. on Automatic Control*, vol. 20, pp. 101-104, 1975.
[30] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *Int'l J. of Computer Vision*, vol. 9, no. 2, pp. 137-154, 1992.
[31] P.N. Yianilos, "Data structures and algorithms for nearest neighbor search in general metric spaces," *Proc. Fourth ACM-SIAM Symp. on Discrete Algorithms*, 1993.
[32] Z. Zhang and O.D. Faugeras, "Three-dimensional motion computation and object segmentation in a long sequence of stereo frames," *Int'l J. Computer Vision*, vol. 7, no. 3, pp. 211-241, 1992.
[33] Q. Zheng and R. Chellappa, "Automatic feature point extraction and tracking in image sequences from unknown camera motion," *Proc. Fourth Int'l Conf. on Computer Vision*, pp. 335-339, 1993.
[34] Q. Zheng and R. Chellappa, "Automatic feature point extraction and tracking in image sequences for arbitrary camera motion," *Int'l J. of Computer Vision*. (to be published)
[35] B. Zhou, "Multitarget tracking in clutter: algorithms for data association and state estimation," PhD thesis, Pennsylvania State Univ., 1992.

Ingemar J. Cox received his PhD degree from Oxford University, England, in 1983. From 1984–1989, he was a principal investigator in the Robotics Principles Department at AT&T Bell Laboratories, Murray Hill, New Jersey, where his research interests focused on issues of autonomous mobile robots. Dr. Cox joined NEC Research Institute as a senior research scientist in 1989. His principal research interests are now in computer vision and robotics.

Sunita Hingorani received the BSc degree in mathematics from St. Xavier's College, Bombay, India, in 1981 and the MSc degree in mathematics from IIT, Bombay, in 1983. In 1988, she received her MS degree in computer science from New Jersey Institute of Technology. From 1990 to 1995, she was a research associate at NEC Research Institute in the Computer Vision group. While there she was involved with research activities in the areas of contour segmentation, stereo vision, and motion tracking. Currently she is a member of the technical staff at AT&T Bell Laboratories, Warren, New Jersey.