LANG HONG
WEI-CHANG WANG
Dept. of Electrical Engineering
Wright State University
329 Russ Center
Dayton, OH 45435

MICHAEL LOGAN
THOMAS DONOHUE
Fusion Development Branch
Wright Laboratory
Wright-Patterson AFB, OH 45433

REFERENCES

[1] Alouani, A. T., Rice, T. R., and Helmick, R. E. (1994)
On sensor track fusion.
In *Proceedings of the American Control Conference*, June 1994, 1042–1046.

[2] Bar-Shalom, Y., and Formann, T. E. (1988)
Tracking and Data Association.
San Diego, CA: Academic Press, 1988.

[3] Blackman, S. S. (1986)
Multi-Target Tracking With Radar Application.
Norwood, MA: Artech House, 1986.

[4] Blom, H. A. P., and Bar-Shalom, Y. (1988)
The interacting multiple model algorithm for system with Markovian switching coefficients.
IEEE Transactions on Automatic Control, **33** (1988), 780–783.

[5] Chong, C. Y., Mori, S., and Chang, K. C. (1990)
Distributed multitarget multisensor tracking.
In Y. Bar-Shalom (Ed.), *Multitarget-Multisensor Tracking: Advanced Applications*.
Norwood, MA: Artech House, 1990, 247–295.

[6] Daubechies, I. (1988)
Orthonormal bases of compact supported wavelets.
Communications on Pure and Applied Mathematics, **XLI** (1988), 909–996.

[7] Hashemipour, H. R., Roy, S., and Laub, A. J. (1988)
Decentralized structure for parallel Kalman filtering.
IEEE Transactions on Automatic Control, **33**, 1 (1988), 88–94.

[8] Hong, L. (1991)
Centralized and distributed multi-sensor integration with uncertainties in communication networks.
IEEE Transactions on Aerospace and Electronic Systems, **27**, 2 (Mar. 1991), 370–379.

[9] Hong, L. (1994)
Multiresolutional distributed filtering.
IEEE Transactions on Automatic Control, **39**, 4 (1994), 853–856.

[10] Hong, L., and Scaggs, T. (1993)
Real-time optimal filtering for stochastic systems with multiresolutional measurements.
System and Control Letters, **21** (1993), 381–387.

[11] Hong, L. (1994)
Multiresolutional multiple-model target tracking.
IEEE Transactions on Aerospace and Electronic Systems, **30**, 2 (1994), 518–524.

[12] Lacoss, R. T. (1987)
Distributed mixed sensor aircraft tracking.
In *Proceedings of American Control Conference*, June 1987, 1827–1830.

[13] Roecker, J. A., and McGillem, C. D. (1988)
Comparison of two-sensor tracking methods based on state vector fusion and measurement fusion.
IEEE Transactions on Aerospace and Electronic Systems, **24**, 4 (1988), 447–449.

[14] Tenney, R. R., and Delaney, J. R. (1984)
A distributed aeroacoustic tracking algorithm.
Proceedings of American Control Conference, June 1984, 1440–1450.

[15] Waltz, E., and Llinas, J. (1990)
Multisensor Data Fusion.
Boston, MA: Artech House, 1990.

Parallelization of a Multiple Model Multitarget Tracking Algorithm with Superlinear Speedups

The interacting multiple model (IMM) estimator has been shown to be very effective when applied to air traffic surveillance problems. However, because of the additional filter modules necessary to cover the possible target maneuvers, the IMM estimator also imposes an increasing computational burden. Hence, in an effort to design a real-time multiple model multitarget tracking algorithm that is *independent* of the number of modules used in the state estimator, we propose a "coarse-grained" (dynamic) parallelization that is superior, in terms of computational performance, to a "fine-grained" (static) parallelization of the state estimator, while not sacrificing tracking accuracy. In addition to having the potential of realizing superlinear speedups, the proposed parallelization scales to larger multiprocessor systems and is robust, i.e., it adapts to diverse multitarget scenarios maintaining the same level of efficiency given any one of numerous factors influencing the problem size. We develop and demonstrate the dynamic parallelization on a shared-memory MIMD multiprocessor for a civilian air traffic surveillance problem using a measurement database based on two FAA air traffic control radars.

I. INTRODUCTION

A. Motivation

Within the aerospace community, there is increasing interest in applying parallel processing techniques to computationally intensive problems associated with ground-based, airborne, and space-based surveillance [1–3, 6, 13, 15, 16]. In the context of military and civilian air traffic surveillance, real-time multisensor-multitarget tracking of airborne targets is one such application. One of the primary

Manuscript received July 8, 1995; revised April 24, 1996.

IEEE Log No. T-AES/33/1/01096.

This research is sponsored, in part, by grants AFOSR F409620-93-1-0399, ONR/BMDO N00014-91-J-1950, and NASA NGT-303104.

0018-9251/97/$10.00

advantages offered by a multiprocessing system is the ability to exploit the processing capabilities of multiple processors in parallel. Indeed, for dense multitarget problems, parallel processing is often necessary because of the limited computational performance of tracking algorithms on conventional uniprocessor systems [13–16].

In general, the objective of a tracking algorithm is to detect and estimate, i.e., track, the states of an unknown number of targets, in the presence of spurious observations and occasional missed detections, using sensor measurements of possibly unknown origin and contaminated by noise and clutter. The *tracking* performance of a tracking algorithm is mainly governed by the accuracy of the state estimator used. The traditional and most widely used state estimator is the Kalman filter. However, because of the single motion model assumption, when its design parameters are tuned to track maneuvering targets via increased process noise covariance, the Kalman filter has been found to provide marginal tracking performance [5, 10]. Alternatively, as demonstrated in a companion paper [20], an interacting multiple model (IMM) estimator [5] can yield a 30% reduction in the RMS prediction error with RMS errors in altitude rate estimates reduced by a factor of 3 over the Kalman filter.

For many realistic multitarget air traffic surveillance problems, a systematic estimator that automatically adapts to each (maneuvering and/or nonmaneuvering) target's individual motion mode is necessary. The IMM estimator has proven to be very effective for such problems by allowing the motion of targets to be described by different state equations in different time intervals [10]. However, the IMM estimator, because of the additional filters, also imposes a higher *computational* burden than a Kalman filter. In general air traffic surveillance problems, a multiple model tracking algorithm should have a reasonable number of filter modules to cover the possible target maneuvers.[1] However, to be practical in a real-time environment, computation time needs to be considered, since increasing the number of filter modules increases the computational load of the estimator considerably. To date, there has been a lack of efficient parallelizations of multiple model tracking algorithms reported in the literature that are *independent* of the number of filter modules used in the state estimator. Hence, filling this gap is one of the primary focuses of the present work.

[1]Theoretically, in terms of tracking accuracy, it is as bad to use too many filter modules as it is to use too few [11]. However, for higher dimensional systems, where a large number of filter modules may be necessary, a *variable-structure* estimator, where only a relatively small *fixed* number of filter modules are active at any particular time, proves to be an excellent alternative [11].

B. Related Research

A literature survey [2, 3, 8, 18] of parallel algorithms developed for multitarget tracking problems reveals a plethora of state estimator parallelizations proposed over the years. However, in each of these studies, "fine-grained" parallelizations were primarily explored, wherein many numerically intensive computations—coordinate transformations, state and covariance estimates, linear algebraic operations such as matrix multiplication, Cholesky (square root) factorization, and inner products—were parallelized. In particular, Atherton, et al. [2] and Averbuch, et al. [3] each developed fine-grained parallelizations of the IMM estimator, the former on a 4-processor distributed-memory transputer, and the latter, on a 4-processor shared-memory MIMD multiprocessor. When using a fine-grained approach, however, the parallelization is static (fixed) *within* the state estimator. As we demonstrate in this work, for an air traffic surveillance problem, fine-grained parallelizations of a multiple model state estimator prove to be inadequate, in terms of computational performance, unless the number of filter modules used is unrealistically high. The alternative that we propose is a "coarse-grained" (dynamic) parallelization *across* the numerous track states found in a multitarget problem.

Current work documented in the literature [1, 13, 14, 17] tends to view the data association problem as one of statistical estimation, wherein a set of measurements received is associated with a set of tracks and false alarms by an unknown random permutation. In particular, Pattipati, et al. [13] describe efficient mappings of multiple hypothesis tracking (MHT) algorithms onto MIMD multiprocessors, while Atherton, et al. [1] describe a parallelization of a track-splitting algorithm for a distributed-memory transputer. However, these enumerative approaches require a search of a large number of hypothesized permutations (feasible associations) to determine the correctness of such hypotheses, with the complexity increasing exponentially with the number of feasible tracks. An alternative is to use an optimization-based approach, wherein a state estimator, such as a Kalman filter or the IMM, is embedded into an assignment-based framework [15, 16, 19, 20]. In such a formulation, the problem is that of finding an optimal assignment of measurements from the latest scan to the most likely tracks from the previous scans using a global cost function, i.e., a *maximum likelihood* (ML) criterion as opposed to a *maximum a posteriori* (MAP) criterion as employed in MHT algorithms. In such a formulation, any well-known polynomial-time assignment algorithm [4, 7] can be used in finding the optimal (minimum cost) assignment.

C. Scope

We begin by describing our multiple model multitarget tracking algorithm in terms of an IMM estimator embedded into the two-dimensional (2D) assignment framework. In an effort to design a real-time tracking algorithm that is *independent* of the number of modules used in the state estimator, we next describe our coarse-grained (dynamic) parallelization for shared-memory MIMD multiprocessor systems. In particular, we perform the parallelization *across* the IMMs, enabling the rather numerous gating, state estimates, covariance calculations, and likelihood function evaluations (used as costs in a global assignment) to be computed in parallel. Moreover, the dynamic scheduling of these computations also takes advantage of the fact that they are not of equal size, and, thus, can schedule them efficiently. In addition to being computationally superior to a static parallelization of the state estimator, the proposed parallelization realizes superlinear speedups,[2] scales to larger multitarget problems, and is robust. We develop and demonstrate the dynamic parallelization for a civilian air traffic surveillance problem using a measurement database based on two FAA air traffic control radars.

II. MULTITARGET TRACKING ALGORITHM

Since it is not the primary focus of this work, we omit detailed exposition of the multitarget scenario here. Instead, a brief description of the scenario is included in Appendix A. However, we do provide a brief explanation of our multitarget tracking algorithm in terms of the state estimation problem (via the IMM) embedded into the data association problem (via a 2D assignment), which serve as the basis of our tracking approach.

A. State Estimation

State estimation provides 1) a measure of how likely a particular measurement from a given scan originated from a particular target track, i.e., provides a *measurement-to-track* association cost that we utilize in the data association problem, and 2) an estimate of the target state. In the state estimation problem, we assume that the target state evolves according to a known linear dynamic model, corrupted by process noise, and driven by a known input, i.e.,

$$x(t_{m_k}) \stackrel{\Delta}{=} F(\delta)x(t_{m_{k-1}}) + G(\delta)v(t_{m_{k-1}}) \tag{1}$$

[2]When nonalgorithmic issues such as context switches, effective memory size, memory access costs, and scheduling order are considered, superlinear speedups in practice may indeed occur [9, 12].

where $\delta = t_{m_k} - t_{m_{k-1}}$ is the time interval, $F(\cdot)$ is the state transition matrix, $G(\cdot)$ is the disturbance matrix, and $v(\cdot)$ is zero-mean, white Gaussian *process* noise with (known) covariance matrix $Q(\cdot)$. Moreover, the measurements are linear functions of the target state corrupted by measurement noise, i.e.,

$$z(t_{m_k}, m_k) \stackrel{\Delta}{=} Hx(t_{m_k}) + w(m_k) \tag{2}$$

where $m_k = 1,\ldots,M(k)$ denotes the m_kth measurement from the kth scan, $H = [I\ 0\ 0]$ is the measurement matrix, and $w(\cdot)$ is zero-mean, white Gaussian *measurement* noise with (known) covariance matrix $R(\cdot)$.

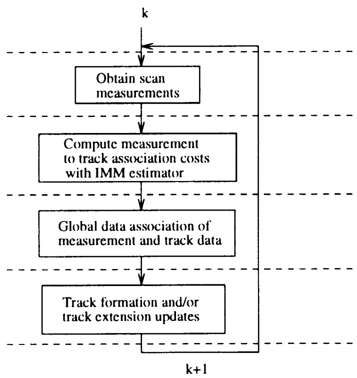
Recall, inherent to a multitarget air traffic surveillance problem is uncertain target dynamics; thus, a set of state equations modeling the motions of such targets is necessary [5, 6]. Hence, in our multitarget tracking algorithm, termed IMM-2D, we use an IMM state estimator. Since it is not the focus of the present work, we omit detailed exposition of the IMM algorithm here and refer interested readers to Bar-Shalom an Li [5, 6] for a more descriptive presentation of the material. However, what is important to know here is that, in IMM-2D, the IMM estimator yields an overall likelihood score, $\Lambda(\cdot)$, which serves as the basis of a candidate measurement-to-track association cost used in the data association (assignment) problem [20].

B. Data Association

Data association is the decision process of *linking* measurements (from successive scans) of a common origin (i.e., a target or false alarms) such that each measurement is associated with at most one origin. In the IMM-2D tracking algorithm, we formulate the data association problem as a 2D assignment problem. Specifically, $M(k)$ measurements from the latest scan k are to be assigned to the $N(k-1)$ most likely existing tracks from the previous scans using a global cost minimization function [6] (based on the likelihood functions $\Lambda(\cdot)$ from the IMM estimators). Specifically, let $n = 0,\ldots,N(k-1)$ denote a particular track from the "set" of existing tracks (including a *dummy* track $n = 0$), and $m_k = 0,\ldots,M(k)$ denote a particular measurement from the latest "set" (scan) of measurements (including a *dummy* measurement $m_k = 0$). Define the binary "assignment" variable

$$x_{nm_k} = \begin{cases} 1 & \text{if} \quad m_k \text{ is assigned to } n \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

Note that $x_{n0} = 1$ implies that track n is unassigned and has missed a detection at scan k. Furthermore, $x_{0m_k} = 1$ implies that measurement m_k is unassigned, that is, not assigned to any of the $N(k-1)$ previously established tracks; in IMM-2D, the set of new tracks to initialize at scan k is $\{m_k : x_{0m_k} = 1\ \forall m_k\}$. Since



Workload Distribution

# filters in IMM: (Kalman filter) 1	2	3	5
.8%	.6%	.5%	.3%
94.3%	94.7%	95.2%	96%
1.1%	0.8%	0.6%	0.2%
3.8%	3.9%	3.7%	3.5%

Fig. 1. IMM-2D block diagram and workload distribution.

measurement errors within a scan are independent of each other, maximizing the likelihood function, consisting of the joint probability density function (pdf)-probability [6] of measurements given their origins and the corresponding detection events, over the set of feasible assignments can be cast into the following 2D assignment problem:

$$\begin{aligned} \min \quad & \sum_{n=0}^{N(k-1)} \sum_{m_k=0}^{M(k)} x_{nm_k} c_{nm_k} \\ \text{s.t.} \quad & \sum_{m_k=0}^{M(k)} x_{nm_k} = 1 \qquad \forall \quad n \\ & \sum_{n=0}^{N(k-1)} x_{nm_k} = 1 \qquad \forall \quad m_k \end{aligned} \tag{4}$$

where the cost of assigning measurement m_k to track n is

$$c_{nm_k} = \begin{cases} 0 & \text{if} \quad n = 0 \text{ or } m_k = 0 \\ -\log\left(\dfrac{\Lambda(n,m_k)}{\Lambda(0,m_k)}\right) & \text{if} \quad -\log(\cdot) < 0 \\ \infty & \text{otherwise} \end{cases}. \tag{5}$$

The numerator in $-\log(\cdot)$ from (5), obtained from the IMM state estimator [20], is the likelihood that the m_kth measurement at scan k originated from the nth track, and the denominator is the likelihood that the m_kth measurement corresponds to none of the existing tracks (i.e., a false alarm). The occurrence of false alarms is assumed uniformly probable over the sensor field of view.

III. IMM-2D PARALLELIZATION

A. Workload

As shown in the control-flow block diagram of Fig. 1, besides *data association* and *state estimation*, two ancillary tasks of the IMM-2D tracking algorithm are *obtaining scan measurements* and *track formation/extension*. In terms of workload distribution, Fig. 1 clearly shows that the vast majority of the workload per scan involves processing the "set" of candidate associations in setting up the data association (assignment) problem, i.e., this constitutes 94.3%, 94.7%, 95.2%, and 96% of the workload for an IMM having 1, 2, 3, and 5 filter modules, respectively. As illustrated in Fig. 1, because this portion of the IMM-2D tracking algorithm is the significant computational bottleneck, we focus on parallelizing only this block.

In order to understand the components that comprise the processing time, we now describe the various operations that occur in this block. Define the set of *candidate associations* at the kth scan by

$$\mathcal{C}(k) \stackrel{\Delta}{=} \{(n,m_k) : (n,m_k) \in N(k-1) \times M(k)\} \tag{6}$$

where $N(k-1) \times M(k)$ denotes the cross product of the track and measurement sets, with $|\mathcal{C}(k)| = N(k-1)M(k)$. For each $(n,m_k) \in \mathcal{C}(k)$, a *coarse* gating test is first applied, consisting of both a "maximum velocity" gate and a high process noise Kalman filter "elliptical" gate,[3] denoted by

$$\mathcal{G}_c : \mathcal{C}(k) \rightarrow \{0,1\} \tag{7}$$

where $\mathcal{G}_c(n,m_k) = 1$ denotes measurement m_k fell within both of the nth target's maximum velocity and elliptical gates, while $\mathcal{G}_c(n,m_k) = 0$ denotes measurement m_k fell outside either of track n's gates. Define the set of candidate associations passing the coarse gating test $\mathcal{G}_c(\cdot)$ by

$$\mathcal{L}(k) \stackrel{\Delta}{=} \{(n,m_k) : (n,m_k) \in \mathcal{C}(k), \mathcal{G}_c(n,m_k) = 1\}. \tag{8}$$

This set $\mathcal{L}(k)$ forms the set of candidate measurement-to-track associations requiring the

[3]The elliptical gate is computed only if the measurement falls within the maximum velocity gate of the target.

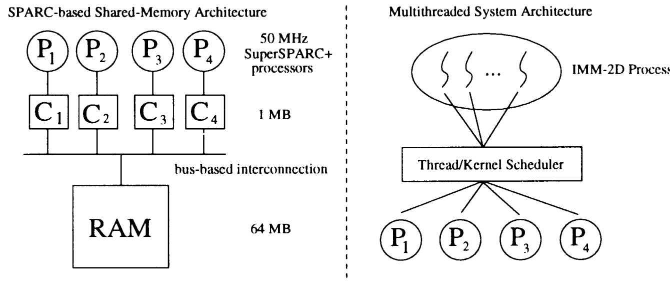


Fig. 2. Shared-memory multiprocessor environment.

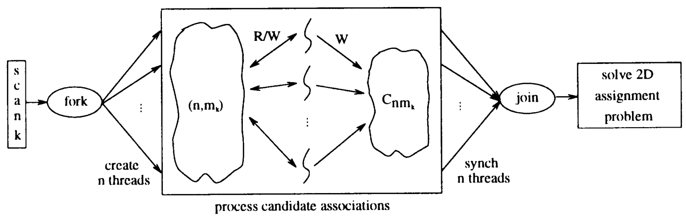


Fig. 3. Task graph of parallel computation.

computation of an association cost based on the likelihood function $\Lambda(\cdot)$, as specified in (5), from the IMM estimator. Following evaluation of the likelihood function, which, when performed, is the single most costly operation in terms of processing time, a *fine* gating test is applied, i.e.,

$$\mathcal{G}_f : \mathcal{L}(k) \rightarrow \{0,1\} \tag{9}$$

where $\mathcal{G}_f(n,m_k) = 0$ denotes that candidate association (n,m_k) is to be filtered out of the 2D assignment problem because it is more likely that measurement m_k corresponds to a false alarm than to track n, i.e., $c_{nm_k} \geq 0$. Conversely, $\mathcal{G}_f(n,m_k) = 1$ denotes that candidate association (n,m_k) is to participate in the 2D assignment problem, with the cost of assigning measurement m_k to track n is $c_{nm_k} < 0$ as defined in (5). With respect to the implementation of the IMM-2D algorithm, the filtering out of unnecessary candidate associations, via coarse gating $\mathcal{G}_c(\cdot)$ and fine gating $\mathcal{G}_f(\cdot)$ of $\mathcal{C}(k)$ and $\mathcal{L}(k)$, respectively, serve as a sparsification technique applied to the 2D assignment problem.

B. Implementation Environment

The computing environment used in this work consists of several shared-memory MIMD multiprocessor systems, i.e., a 2-processor and 4-processor SPARCstation 20, and a 12-processor SPARCcenter 2000. A simple model of the 4-processor SPARCstation 20 architecture with corresponding hardware specifications is illustrated in Fig. 2. The other architectures (not illustrated) have comparable hardware specifications. The software utilized consists of Sun Microsystem's Solaris 2.4.2 environment, which includes the SunOS 5.4.2 Unix operating system (kernel), SPARCworks 3.0.1 C compiler, and the parallel processing interface via the multithreaded system architecture.

While a traditional Unix process has always contained a single *thread* of control, multithreading separates a process into many independent, lightweight threads, each of which executes (possibly concurrently) a sequence of the process's instructions. Two levels of thread scheduling occurs in SunOS: application-level threads are dispatched across a set of kernel supported threads via a library-supported threads scheduler, and kernel-level threads (i.e., lightweight processes) are, in turn, dispatched across the processor set of the multiprocessor via a kernel scheduler. Numerous synchronization mechanisms are supported in SunOS allowing threads to cooperate in accessing shared data.

C. Shared-Memory Parallelization

The coarse-grained parallelization of the IMM-2D tracking algorithm is based on both the supervisor/worker model and the use of a homogeneous data partitioning strategy. Specifically, per scan, multiple identical worker threads, asynchronously and dynamically, perform the same task—process measurement-to-track associations $(n,m_k) \in \mathcal{C}(k)$—across different track and measurement data (mutually exclusive n and m_k). Upon processing all candidate associations by workers, the supervisor thread solves the global data association (2D assignment) problem. In contrast, a "strictly" fine-grained parallelization of IMM-2D would sequentially iterate over the set of candidate associations, computing their corresponding association costs in parallel. In Fig. 3, we provide a task graph of the parallel computation illustrating the parallelization.

When the routine corresponding to setting up the data association problem is called, a supervisor thread creates (forks) some number of worker threads (based on a design parameter) to process $\mathcal{C}(k)$ and waits for all newly created threads to terminate prior to solving the 2D assignment problem. Each worker thread will process some number of candidate measurement-to-track associations per "serialized" critical section access via the mutex lock *mutex*. For each measurement-to-track association (n, m_k), the thread applies the coarse gating test $\mathcal{G}_c(\cdot)$, and, if $\mathcal{G}_c(n, m_k) = 1$, the thread computes an association cost, based on (5), and applies the fine gating test $\mathcal{G}_f(\cdot)$, and, if $\mathcal{G}_f(n, m_k) = 1$, stores c_{nm_k} as input into the 2D assignment problem. Conceptually, each worker thread executes the pseudocode in IMM-2D( ) as described below:

IMM-2D( )
  for (; ;) /* forever */
    lock(*mutex*)
    task = get_job( ) /* get tasksize (n, m_k)'s */
    unlock(*mutex*)
    if *task* = Ø done( )
    for each $(n, m_k) \in task$ **do**
      if $\mathcal{G}_c(n, m_k) = 1$ **and** $\mathcal{G}_f(n, m_k) = 1$
        $C[n, m_k] = c_{nm_k}$ /* Eq. (5) */

IV. RESULTS

A. Preliminaries

In this section, based on the 2-processor and 4-processor SPARCstation 20, we demonstrate the computational performance of IMM-2D tracking algorithm. We present performance results for IMM-2D via the proposed coarse-grained parallelization and using a fine-grained parallelization of the IMM estimator, wherein the multiple filter modules are evenly distributed across the processor set in the multiprocessor. The results are based on a (relatively sparse) multitarget air traffic surveillance problem, as described in Appendix A, using data from two FAA air traffic control radars, courtesy of Rome Laboratory. To get some sense of the data, in Fig. 4, we plot as a function of the scan number, post mortem, the number of tracks, number of measurements in the scan, the size of the candidate association set $|\mathcal{C}(k)|$, and, based on various coarse gating policies,[4] the size of the set $|\mathcal{L}(k)|$ of candidate measurement-to-track associations requiring the computation of an association cost based on the likelihood function $\Lambda(\cdot)$, as specified in (5), from the IMM estimator.

Note that, even for this relatively sparse problem, hundreds of candidate associations, per scan, require a likelihood function evaluation. The reason for this is because we chose to keep a track for up to 100 s with no measurements before discarding it, i.e., a track is dropped if unassigned in the 2D assignment problem across 20 consecutive scans. Consequently, such "mature" tracks will have "large" gates [15, 16] with many candidate measurements to associate with, which, however, might go to other tracks. Indeed, many of the 300+ tracks that exist from scan 20 onwards are tracks initiated based on false alarms, which, in turn, get discarded in future scans; roughly 75 actual targets are tracked in the measurement database.

[4]Even though unrealistic for the civilian air traffic surveillance problem at hand, a Mach 5 gate was used only to illustrate other computational aspects of the IMM-2D parallelization in the sequel.

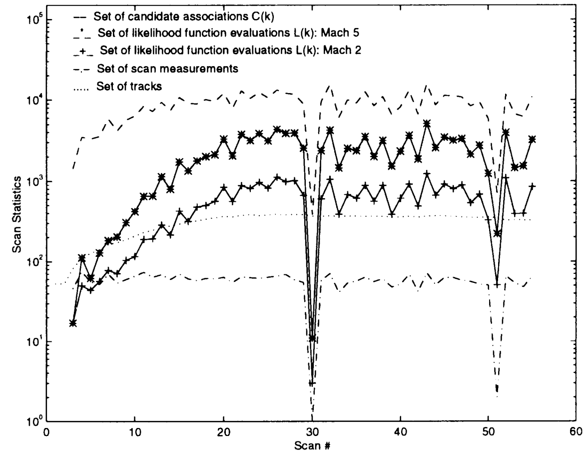


Fig. 4. Comparison of various scan statistics.

B. Execution Time and Parallel Efficiency

Since the multiprocessors used in this work are time-shared systems, the execution times of IMM-2D depended, in part, on random system events such as the system load and thread schedule order. Consequently, Monte Carlo simulations were performed, and all results presented represent the means of those simulations with standard errors less than 3%. In Fig. 5, we plot the execution times of the sequential, the proposed coarse-grained, and the fine-grained parallelizations of IMM-2D based on the 2-processor and 4-processor SPARCstation 20 using various coarse gating policies. Clearly, the coarse-grained approach demonstrates superior execution time performance, independent of the number of filter modules used in the IMM estimator, over the fine-grained approach. Furthermore, even though the fine-grained parallelization does show improvement in execution time over sequential time, it does so only for an IMM estimator having a large number of filter modules. In fact, for an IMM estimator having a small to moderate number of filter modules (3 or less), the fine-grained parallelization has *greater execution time than sequential time.* The primary overhead in using this approach is a substantial number of fork-join costs relevant to the computation performed (i.e., an association

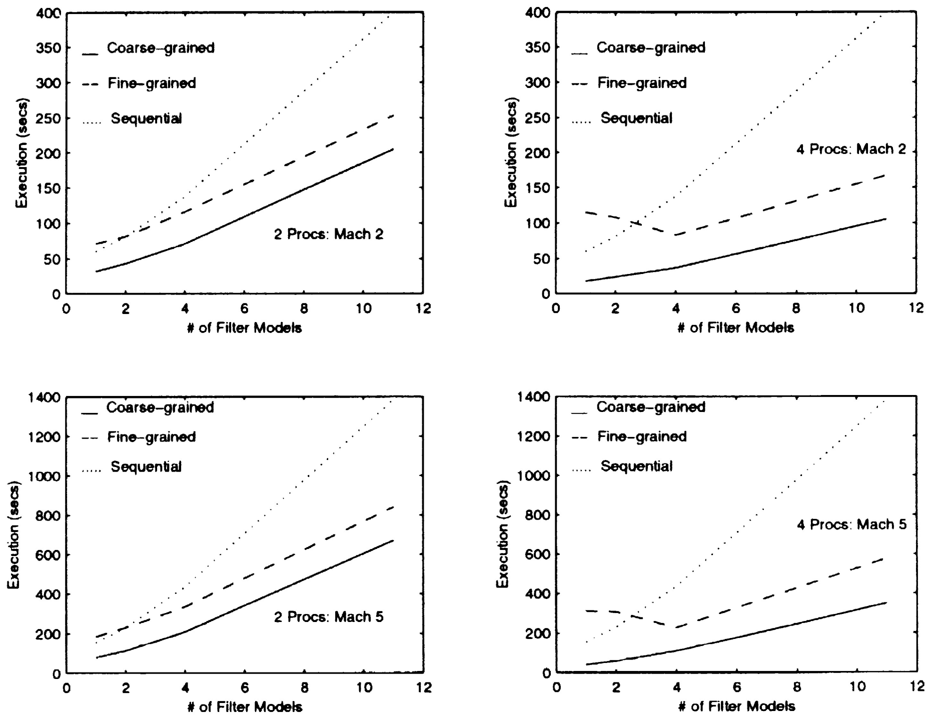


Fig. 5. Execution time versus number of motion models in IMM for various number of processors and coarse grating policies.

cost), whereas the coarse-grained approach suffers from numerous less intensive serialized accesses to the shared task queue when processing candidate associations in setting up the data association problem.

In Fig. 6 we plot the *parallel efficiency*, based on a conventional formulation, for both the coarse-grained and fine-grained parallelizations of IMM-2D, i.e.,

$$\mathcal{E}_{ff} = \frac{\tau_1}{p\tau_p} \qquad (10)$$

where τ_1 denotes the sequential execution time of IMM-2D utilizing 1 processor, and τ_p denotes the parallel execution time utilizing p processors. Clearly, in terms of efficiency, the coarse-grained parallelization is superior to the fine-grained parallelization for any number of filter modules used in the IMM estimator. Moreover, as Fig. 6 illustrates, the computational performance of the coarse-grained parallelization is independent of the number of filter modules used in the IMM estimator, whereas the fine-grained parallelization performs rather inefficiently unless the number of filter modules used is unrealistically high. Furthermore, near-unity efficiency, and, given a large enough problem size (e.g., 2 processors using Mach 5 coarse gating policy), greater than unity efficiency, which implies *superlinear speedup*, is possible via the coarse-grained parallelization.

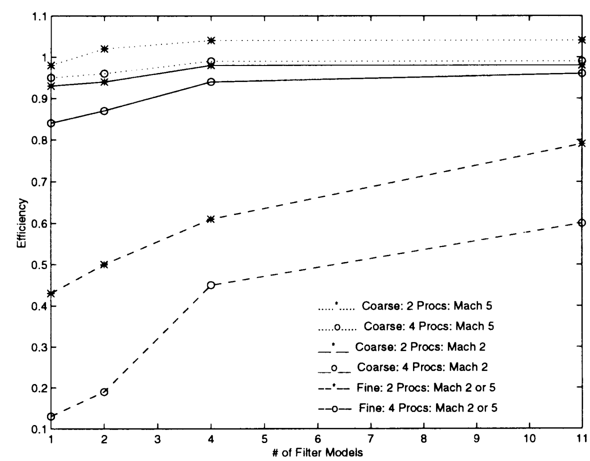


Fig. 6. Parallelization efficiency versus number of motion models in IMM for various coarse grating policies.

C. Robustness and Scaling

Many factors determine the performance of a parallel algorithm, in particular, the multiprocessor architecture and the problem size are directly related. In the context of a multitarget air traffic surveillance problem, the problem size is a function of multiple factors: whether or not the track corresponds to an actual target or a false alarm, noise intensity, degree of clutter, target density in the neighborhood of a tracked target, and the computation size of the association cost itself (i.e., increases in the number of filter modules). When considering the *scalability* of a multitarget tracking algorithm for larger multiprocessor systems, a highly desirable feature is when the algorithm can maintain the same "high" level of performance (e.g., speedup) on the larger multiprocessor system as it did on the smaller one, and do so in terms of *any* of the factors that can influence the problem size. Certainly,

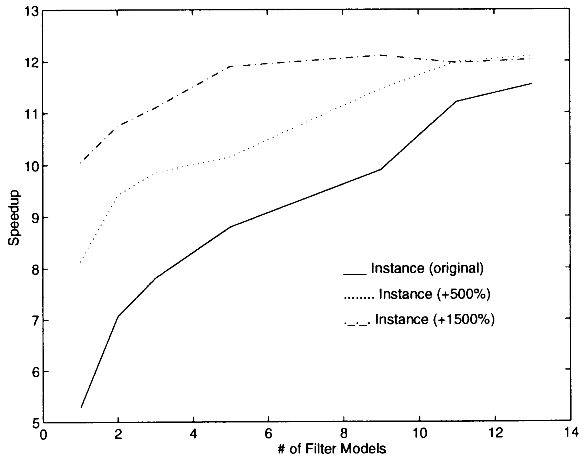


Fig. 7. Speedup versus number of motion models in IMM using Mach 5 coarse grating policy.

a multitarget tracking algorithm having this attribute would be considered *robust*, and could easily adapt, without modification, to diverse multitarget problems.

The coarse-grained parallelization of IMM-2D scales when any of the factors influencing problem size increase, i.e., we are able to maintain "roughly" the same level of performance across a larger multiprocessor system as we are able to on a smaller one. In Fig. 7, we plot the speedup for the coarse-grained parallelization on a 12-processor SPARCcenter 2000 for several problem instances. In (*Instance* (*original*)), all factors influencing the problem size are unchanged. From the plot we can see marginal speedup results when the number of filter modules in the IMM estimator is small. This indicates that the problem size is too small for the particular multiprocessor architecture. However, as the number of filter modules in the IMM estimator increases, implying that the problem size increases, clearly the speedup increases to near linear. In (*Instance* (+500%)), we simulated a more dense multitarget scenario by increasing the problem size (relative to (*Instance* (*original*))) in terms of track set size by 500%. Clearly, we see the robustness of the coarse-grained parallelization where, again, the plot is approaching linear speedup when fewer filter modules are in the IMM estimator, and linear when many filter modules are used. And lastly, (*Instance* (+1500%)) increases the problem size 1500% by increasing the track set, per scan, by the same amount, and clearly we can obtain even better speedup performance than in the previous two cases.

D. When to Parallelize?

In the case of single target tracking, the parallelization of a multiple model state estimator would be, by necessity, fine grained (static), and, as demonstrated in this work and [15], extremely inefficient unless the number of filter modules used is unrealistically high. Indeed, for an IMM estimator having a small to moderate number of filter modules, the fine-grained parallelization has worse execution time performance than sequential time (recall Fig. 5). When using an IMM configured with 12 filter modules, Averbuch, et al. [3] demonstrated that efficiency with p processors is roughly $1 - p^{-1}$ for small p, and, as shown here, comparable with our results. However, to expect an improvement for larger p would by necessity require using *many* filters and larger state and measurement vectors [3]. Hence, one should not waste time trying to parallelize a single target "small" multiple model state estimator on a general-purpose shared-memory multiprocessor; however, a fine-grained parallelization on an ASIC chip could be made efficient.

In the multitarget situation, based on our results, we suggest that fine-grained state estimator parallelizations should be used with caution. However, a coarse-grained parallelization across IMMs is a good strategy when any of the factors influencing problem size is large, i.e., many models in the IMM, large track/scan set sizes, or many candidate associations requiring a likelihood function evaluation because of clutter, dense scenarios, and/or coarse gating policies. For example, note that in Fig. 6, even for our relatively sparse air traffic surveillance problem based on an average of 300 tracks and 50 measurements per scan, when using a Kalman filter (e.g., IMM configured with 1 filter module) with a fairly tight coarse gating policy (e.g., Mach 2), roughly 85% efficiency was obtainable in a coarse-grained parallelization where only 500 likelihood function evaluations were computed, on average.

Hence, a rule of thumb for when a coarse-grained parallelization *across IMMs* is efficient ($\geq 80\%$) in a multitarget case is when

$$\frac{(\#\text{ fine gate associations}) \times (\#\text{ modules in IMM})}{\#\text{ processors}} \geq 250.$$

Similarly, the fine-grained parallelization *within IMMs*, which does not depend on the number of fine gate associations, becomes efficient ($\geq 80\%$) if

$$\frac{\#\text{ modules in IMM}}{\#\text{ processors}} \geq 4.$$

V. CONCLUSIONS

We have proposed a robust, scalable, coarse-grained (dynamic) parallelization of a multiple model multitarget tracking algorithm that realizes superlinear speedups on shared-memory MIMD multiprocessors. The advantage of the dynamic parallelization presented here shows up in a reasonably large multitarget problem where hundreds of candidate measurement-to-track associations must be processed;

TABLE I
Radar Specifications

	f_p	β_θ	β_ϕ	τ	$R_{\max} = c/2f_p$	$\Delta r = c\tau/2$
Rem	340 Hz	5.4°	1.3°	6 μs	441.2 km	0.9 km
Dan	350 Hz	3.75°	1.2°	1.8 μs	428.6 km	0.27 km

indeed, the parallelization cannot be efficiently utilized in the case of a single target tracking problem which is inherently static. In this case, however, the fine-grained parallelization is extremely inefficient, unless the number of filter modules used in the estimator is unrealistically high. We demonstrated the computational superiority of the proposed parallelization over the static parallelization via several performance measures utilizing a measurement database based on two FAA air traffic control radars.

APPENDIX A. AIR TRAFFIC SURVEILLANCE PROBLEM

Important aspects of the raw scan measurement database and the sensor model are as follows.

1) The data consist of scans from two L-band FAA radars located at Remsen and Dansville, NY. The databases used consist of 55 scans (26 Remsen, 29 Dansville) and 210 scans (98 Remsen, 112 Dansville).

2) The data from the two FAA radars consists of scans at approximately every 10 s. Each of these scans contains a number of primary radar or *skin* returns. Each of these skin returns consists of a time stamp, a slant range, and azimuth angle measurements. For *cooperative* targets, a secondary or beacon return is also obtained, which provides, in addition to the above, a target identification number (ID or squawk) and a target altitude measurement.

3) The observability of the target state requires a full measurement of its position. Only beacon returns provide a measurement of the full target position.

4) Skin returns provide only a partial measurement of the target state. In an effort to treat skin returns similarly to beacon returns, we augment the former with an "estimated" altitude measurement for *noncooperative* targets. These are used only for coordinate transformations from sensor polar to target local coordinates. See [20] for details relating to the incorporation of skin returns into our algorithmic framework.

The sensor parameters for the radars are given in Table I, where f_p is the pulse repetition frequency, β_θ is the vertical beamwidth, β_ϕ is the horizontal beamwidth, τ is the pulse width, $R_{\max}$ is the maximum range, Δr is the range resolution cell, and c is the speed of light. From the above sensor parameters, the range and azimuth standard deviations (i.e., σ_r and σ_ϕ, respectively) are determined using the assumption that the range and azimuth measurements are uniformly distributed in the corresponding resolution cells. Hence, $\sigma_r = \Delta r/2\sqrt{3}$ and $\sigma_\phi = \beta_\phi/2\sqrt{3}$. The altitude standard deviation $\sigma_h = 17.6$ m and the probability of detection $P_D = 0.95$ are chosen based on FAA standards [19, 20]. We assume that the pdf of the location of extraneous measurements (false alarms) is uniform in the validation gate of the track under consideration [6]. Hence, the pdf of an extraneous measurement is the inverse of the gate volume, denoted by Ψ. We assume that Ψ is approximately constant and proportional to the resolution cell volume, i.e., $\Psi = K(\Delta r)(R_{\max}\beta_\theta)(R_{\max}\beta_\phi)$. The factor K (e.g., $3^3 = 27$) is needed since the gate is, in general, "spread" over several adjoining resolution cells.

ACKNOWLEDGMENTS

We would like to thank Rome Laboratory, in particular, Rick Gassner, Vince Vannicola, and Mark Linderman for providing the data sets and parallel processing resources that enabled much of this work to continue.

ROBERT L. POPP
Dept. of Computer Science and Engineering
University of Connecticut
Storrs, CT 06269-2155

KRISHNA R. PATTIPATI
YAAKOV BAR-SHALOM
MURALI YEDDANAPUDI
Dept. of Electrical and Systems Engineering
University of Connecticut
U157
Storrs, CT 06269-2157

REFERENCES

[1] Atherton, D. P., Gul, E., Kountzeris, A., and Kharbouch, M. M. (1990)
Tracking multiple targets using parallel processing.
IEE Proceedings of Control, Theory and Applications, **137**, 6 (July 1990), 225–234.

[2] Atherton, D. P., and Lin, H.-J. (1994)
Parallel implementation of IMM tracking algorithm using transputers.
IEE Proceedings of Radar, Sonar Navigation, **141**, 6 (Dec. 1994), 325–332.

[3] Averbuch, A., Itzikowitz, S., and Kapon, T. (1991)
Parallel implementation of multiple model tracking algorithms.
IEEE Transactions on Parallel and Distributed Systems, **2**, 2 (Apr. 1991), 242–252.

[4] Balas, E., Miller, D., Pekny, J., and Toth, P. (1991)
A parallel shortest augmenting path algorithm for the assignment problem.
Journal of the Association of Computing Machinery, **38**, 4 (Oct. 1991), 985–1004.

[5] Bar-Shalom, Y., and Li, X. R. (1993)
Estimation and Tracking: Principles, Techniques and Software.
Boston, MA: Artech House, 1993.

[6] Bar-Shalom, Y., and Li, X. R. (1995)
Multitarget-Multisensor Tracking: Principles and Techniques. Storrs, CT: YBS Publishing, 1995.
[7] Bertsekas, D. P., and Castanon, D. A. (1991)
Parallel synchronous and asynchronous implementations of the auction algorithm.
Parallel Computing, **17** (1991), 707–732.
[8] Desai, U. B., and Das, B. (1985)
Parallel algorithms for Kalman filtering.
In *Proceedings of the American Control Conference*, Boston, MA, June 1985.
[9] Janben, R. (1987)
A note on superlinear speedup.
Parallel Computing, **4** (1987), 211–213.
[10] Li, X. R., and Bar-Shalom, Y. (1993)
Design of an interacting multiple model algorithm for air traffic control tracking.
IEEE Transactions on Control Systems Technology, **1**, 3 (Sept. 1993), 186–194; special issue on air traffic control.
[11] Li, X. R., and Bar-Shalom, Y. (1992)
Mode-set adaptation in multiple model approach to hybrid state estimation.
In *Proceedings of the American Control Conference*, Chicago, IL, June 1992, 1794–1799; also to appear in *IEEE Transactions on Automatic Control*, 1996.
[12] Parkinson, D. (1986)
Parallel efficiency can be greater than unity.
Parallel Computing, **3** (1986), 261–262.
[13] Pattipati, K. R., Kurien, T., Lee, R.-T., and Luh, P. B. (1990)
On mapping a tracking algorithm onto parallel processors.
IEEE Transactions on Aerospace and Electronic Systems, **26**, 5 (Sept. 1990), 774–791.
[14] Pattipati, K. R., Deb, S., Bar-Shalom, Y., and Washburn, R. (1992)
A new relaxation algorithm and passive sensor data association.
IEEE Transactions on Automatic Control, **37**, 2 (Feb. 1992), 197–213.
[15] Popp, R. L., Pattipati, K. R., Bar-Shalom, Y., and Yeddanapudi, M. (1995)
The parallelization of a large-scale IMM-based multitarget tracking algorithm.
In *Proceedings of the SPIE Conference on Signal and Data Processing of Small Targets*, San Diego, CA, July 1995; paper 2561.
[16] Popp, R. L., Pattipati, K. R., and Gassner, R. R. (1996)
Multitarget tracking parallelization for high-performance computing architectures.
In *Proceedings of the 4th Annual SCS Symposium on High-Performance Computing*, New Orleans, LA, 1996.
[17] Reid, D. B. (1979)
An algorithm for tracking multiple targets.
IEEE Transactions on Automatic Control, **24**, 6 (Dec. 1979), 423–432.
[18] Rhodes, I. B. (1990)
A parallel decomposition for Kalman filters.
IEEE Transactions on Automatic Control, **35**, 3 (Mar. 1990), 322–324.
[19] Yeddanapudi, M., Bar-Shalom, Y., Pattipati, K. R., and Gassner, R. R. (1995)
MATSurv: Multisensor air traffic surveillance system.
Proceedings of SPIE Conference on Signal & Data Processing of Small Targets, San Diego CA, July 1995; paper 2561.
[20] Yeddanapudi, M., Bar-Shalom, Y., and Pattipati, K. R. (1996)
IMM estimation for multitarget-multisensor air traffic surveillance.
In *Proceedings of 34th IEEE CDC*, New Orleans, LA, Dec. 1995; also to appear in *IEEE Proceedings*, 1996.

Autocorrelation Sidelobe Considerations in the Characterization of Multipath Errors

The effects of autocorrelation sidelobes on multipath errors in pseudorandom noise (PRN) ranging systems are investigated. It is shown that both medium-delay (i.e., on the order of 1 PRN chip) as well as long-delay multipath errors are affected. Results are applied to the case of the Global Positioning System (GPS).

I. INTRODUCTION

Satellite-based pseudorandom noise (PRN) ranging systems such as the Navigation Satellite Timing And Ranging Global Positioning System (NAVSTAR GPS) are subject to a variety of error sources [1]. Augmentations such as differential operation help to reduce or eliminate many of these errors [1]. However, multipath remains the dominant error source in most high precision applications [2].

Multipath effects in PRN ranging systems have been under study for over two decades [2–14]. In virtually every study, however, the effects of autocorrelation function sidelobes are neglected. Van Nee [14] touched briefly on the subject but stopped short of performing a full characterization.

This article derives the bounds on multipath error in terms of a more representative autocorrelation function. The results show a distinct change in the error envelope from that which has been derived previously using a simplified model for the autocorrelation function. In Section II, a brief overview of the characteristics of the PRN code autocorrelation function (including sidelobes) is given and the impact on the delay-lock loop (DLL) discriminator function is derived. In Section III, a general description of multipath error is derived in terms of the new parameterization. This quantifies the influence of sidelobe levels on receiver tracking error due to multipath. Section IV applies the results of Section III to the case of the Global Positioning System (GPS).

II. PRN CODE AUTOCORRELATION SIDELOBES AND THE DELAY-LOCK LOOP DISCRIMINATOR FUNCTION

In order to lay the groundwork for the multipath development, a brief description of the autocorrelation

Manuscript received August 31, 1995; revised April 2, 1996.

IEEE Log No. T-AES/33/1/01097.

This work was funded by the Federal Aviation Administration Satellite Program Office, AGS-100, under Grant 92-G-0231.