2014 American Control Conference (ACC)
June 4-6, 2014. Portland, Oregon, USA

Multiple Target Tracking using Recursive RANSAC

Peter C. Niedfeldt and Randal W. Beard
Department of Electrical and Computer Engineering
Brigham Young University, Provo, UT 84602

Abstract— **Estimating the states of multiple dynamic targets is difficult due to noisy and spurious measurements, missed detections, and the interaction between multiple maneuvering targets. In this paper a novel algorithm, which we call the recursive random sample consensus (R-RANSAC) algorithm, is presented to robustly estimate the states of an unknown number of dynamic targets. R-RANSAC was previously developed to estimate the parameters of multiple static signals when measurements are received sequentially in time. The R-RANSAC algorithm proposed in this paper reformulates our previous work to track dynamic targets using a Kalman filter. Simulation results using synthetic data are included to compare R-RANSAC to the GM-PHD filter.**

I. Introduction

Multiple target tracking (MTT) continues to be an active field of research due to inherent difficulties caused by noisy measurements, false detections, missed detections, and the interactions between multiple maneuvering targets. There exists a wide range of target tracking applications in both civilian and military applications. Some of these applications include tracking pedestrians [1], air and ground vehicles [2], bacteria [3], air traffic control [4], and many others. In this paper we present a new algorithm that efficiently and robustly tracks multiple targets in clutter.

Classical MTT techniques include nearest neighbor [5], multiple hypothesis tracking (MHT) [6], and the joint probabilistic data association (JPDA) [7] algorithms. Numerous variations and applications of these algorithms can be found in the literature. Unfortunately, there are limitations with each of these algorithms: nearest neighbor is not robust, MHT is computationally expensive and difficult to implement, and JPDA requires a priori information on the number of targets to be tracked and makes a hard data association each time step. Recently, the probabilistic hypothesis density (PHD) filter has successfully applied random finite set theory to efficiently track an unknown number of targets in highly-cluttered environments [8]. A significant advantage of the PHD filter is that the computations are linear in the number of measurements. However, a disadvantage is that additional steps are required to maintain track continuity. Also, the estimated number of targets has a large variance in clutter [9], and Erdnic et al. note that the PHD filter is sensitive to track loss after very few missed detections [10].

This research was made with Government support through the DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a.

KAMLANSAC is another algorithm that robustly tracks a dynamic target using causal measurements [11]. Both KALMANSAC and the recursive-RANSAC algorithm developed in this paper rely on the principles of the random sample consensus (RANSAC) algorithm. RANSAC was developed as a new paradigm to regression analysis given a batch of data [12]. Traditional regression techniques form an estimate using all or most of the available data set, and then remove observations inconsistent with the hypothesis before producing a final estimate. Alternatively, RANSAC forms numerous hypothesis estimates using the minimum number of required points and computes the support of each hypothesis, where the hypothesis support refers to the number of measurements, or *inliers*, whose residual is less than a specified threshold. The hypothesis with the greatest support is selected and smoothed using the inlier set, allowing RANSAC to quickly and accurately estimate the parameters of an underlying signal in clutter. For a recent survey of the many variations of RANSAC, see [13].

The KALMANSAC algorithm utilizes the RANSAC paradigm when tracking a dynamic target. KALMANSAC first selects a random minimum subset of measurements to estimate the most likely state estimate, which is subsequently used to estimate the most likely inlier/outlier measurement labeling over the measurement set. These steps are iterated until a stopping criterion is satisfied. The resulting measurement labeling is used to seed the iteration of subsequent time-steps. Since KALMANSAC computes the maximum a posterior estimate, the underlying measurement distribution, excluding the outliers, must be unimodal. However, this assumption does not hold when there exist multiple targets since there is no mechanism for data association between targets.

In this paper, we extend RANSAC to recursively estimate multiple dynamic targets using the recursive-RANSAC (R-RANSAC) framework, previously developed for estimating the parameters of multiple static signals [14]. While KALMANSAC uses iteration to compute the MAP estimate, R-RANSAC stores multiple hypothesis tracks in memory to allow subsequent inlier measurement to refine the current estimate. By tracking multiple hypotheses across several time steps, R-RANSAC can naturally track multiple targets without prior knowledge of the number of existing targets [15].

The R-RANSAC algorithm is summarized as follows. At each time step, existing tracks are propagated using the Kalman filter. Current measurements that are inliers to existing tracks are used to update those tracks. The remaining

current measurements are outliers and are used in conjunction with a random minimum set of previous measurements to hypothesize the initial states of a new track that characterizes the outlier. This new track is propagated forward and updated using measurements that are within the inlier threshold to obtain a current state estimate of the target. Tracks with high support are identified as valid target tracks.

The rest of the paper is organized as follows. Section II describes the problem and notation. In Section III the initial states of a single target using RANSAC using maximum likelihood. Section IV presents the R-RANSAC algorithm to recursively estimate the states of multiple targets. Simulations are presented in Section V using synthetic data.

II. Problem Description

We desire to estimate the states of multiple dynamic targets given a set of measurements and a dynamic model. Let the surveillance region be defined as a subset of the measurement space, $\mathcal{R}_{\mathbf{y}} \subset \mathbb{R}^m$. Let $\mathbf{x}_i \in \mathbb{R}^n$ be the n states of the i^{th} target whose dynamics are given by

$$\mathbf{x}_i[k] = A\,\mathbf{x}_i[k-1] + \mathbf{w}_i[k], \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^n$ is wide-sense stationary (WSS), zero-mean Gaussian process noise with covariance Q. Let $\mathbf{y}_i \in \mathcal{R}_{\mathbf{y}}$ be the measurement of the i^{th} target, given by

$$\mathbf{y}_i[k] = C\,\mathbf{x}_i[k] + \mathbf{v}_i[k], \tag{2}$$

where $\mathbf{v} \in \mathbb{R}^m$ is a WSS, zero-mean Gaussian process with covariance R.

Let $M[k]$ be the unknown number of true targets at time k, where at each time step the probability that the i^{th} target is observed is modeled by the probability of detection $p_i \in (0, 1]$. Define a sensor *scan* as the set of ψ_k observations received after processing the sensor data at time k. The observation $\mathbf{y}_j, j = 1, \ldots, \psi_k$, can be a measurement of an underlying true target or instead a realization of some random distribution defined over the surveillance region.

A false measurement, or clutter, is received according to a Poisson distribution with parameter λ_{GE}, where λ_{GE} is the expected amount of clutter per scan. Unless prior knowledge of the clutter distribution is known, we assume it is independent and uniformly distributed over the surveillance region as in [16]. We assume that target evolution, target measurements, and clutter measurements are all independent from each other, and that the system (A, C) is observable. Finally, the probability of detection is state independent. These assumptions are common in the literature [8], [5].

III. Computing the Initial States from Batch Estimation

Least-squares estimation finds the optimal parameters to fit a model to a batch of data by minimizing the mean-squared error. Alternatively, the random sample consensus (RANSAC) algorithm generates hypotheses using the fewest necessary measurements from a batch of data; the hypothesis with the most support, or inliers, is selected as the best estimate. Inliers are measurements whose residual is less than a threshold τ_{R}. In the following, we estimate the current states of a targets using maximum likelihood and the RANSAC framework.

A. Single Target with No Clutter

Consider the scenario when there is a single target that is measured with probability $p_1 = 1$ and that the probability of clutter is $\lambda_{\text{GE}} = 0$. Under these assumptions, there is exactly one measurement per time step that corresponds to the target of interest. For simplicity, we define $k_N \triangleq k - N + 1$ as the initial time index of the measurement window of length N. Consider a sequence of N measurements characterized by

$$\begin{bmatrix} \mathbf{y}[k_N] \\ \mathbf{y}[k_N+1] \\ \vdots \\ \mathbf{y}[k] \end{bmatrix} = \begin{bmatrix} C\mathbf{x}[k_N] \\ C\mathbf{x}[k_N+1] \\ \vdots \\ C\mathbf{x}[k] \end{bmatrix} + \begin{bmatrix} \mathbf{v}[k_N] \\ \mathbf{v}[k_N+1] \\ \vdots \\ \mathbf{v}[k] \end{bmatrix}. \tag{3}$$

Using (1), we write (3) in terms of $\mathbf{x}[k_N]$, as shown in (10). Define $Y_k = \big[\mathbf{y}[k_N], \ldots, \mathbf{y}[k]\big]^\top$, $W_k = \big[\mathbf{w}[k_N], \ldots, \mathbf{w}[k]\big]^\top$, $V_k = \big[\mathbf{v}[k_N], \ldots, \mathbf{v}[k]\big]^\top$, and

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N-1} \end{bmatrix},\; G = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 \\ 0 & C & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & CA^{N-2} & CA^{N-3} & \ldots & C \end{bmatrix}.$$

Matrix $\mathcal{O}$ is $mN \times n$ and describes the expected measurements evolving from the initial state $\mathbf{x}[k_N]$ to subsequent time steps; it is the observability matrix if $N = n$. Matrix G has dimensions $mN \times nN$ and propagates the process noise.

We can write (10) as

$$Y_k = \mathcal{O}\mathbf{x}[k_N] + \xi_k, \tag{4}$$

where $\xi_k = GW_k + V_k$ is the combined propagated process noise and the measurement noise. Since $\mathbf{w}$ and $\mathbf{v}$ are i.i.d. and zero-mean Gaussian, then ξ is also zero-mean Gaussian with covariance Ξ given by

$$\Xi = E\big[\xi_k \xi_k^\top\big] = GE\big[W_k W_k^\top\big]G^\top + E\big[V_k V_k^\top\big]. \tag{5}$$

Define $\mathbf{Q} = E\big[W_k W_k^\top\big]$ and $\mathbf{R} = E\big[V_k V_k^\top\big]$, where, due to the assumed independence of the process and measurement noise, $\mathbf{Q}$ and $\mathbf{R}$ are both block diagonal matrices. Using this notation, (5) can be written as

$$\Xi = G\mathbf{Q}G^\top + \mathbf{R}. \tag{6}$$

Briefly consider the scenario where there is no process or measurement noise, i.e., $\mathbf{w} = \mathbf{v} = 0$. In this case, (4) reduces to $Y_k = \mathcal{O}\mathbf{x}[k_N]$ and assuming $\mathcal{O}$ is full rank, the initial state $\mathbf{x}[k_N]$ is found using least-squares, as

$$\mathbf{x}[k_N] = (\mathcal{O}^\top\mathcal{O})^{-1}\mathcal{O}^\top Y_k. \tag{7}$$

By observability, this is true if $N \geq n$. When including the process and measurement noise, the maximum likelihood estimate (MLE) of $\mathbf{x}[k_N]$ and the covariance are given by

$$\hat{\mathbf{x}}[k_N] = \left(\mathcal{O}^\top\Xi^{-1}\mathcal{O}\right)^{-1}\mathcal{O}^\top\Xi^{-1}Y_k \tag{8}$$

$$P[k_N] = \left(\mathcal{O}^\top\Xi^{-1}\mathcal{O}\right)^{-1}. \tag{9}$$

$$\begin{bmatrix} \mathbf{y}[k_N] \\ \mathbf{y}[k_N+1] \\ \vdots \\ \mathbf{y}[k] \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N-1} \end{bmatrix} \mathbf{x}[k_N] + \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & C & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & CA^{N-2} & CA^{N-3} & \dots & C \end{bmatrix} \begin{bmatrix} \mathbf{w}[k_N] \\ \mathbf{w}[k_N+1] \\ \vdots \\ \mathbf{w}[k] \end{bmatrix} + \begin{bmatrix} \mathbf{v}[k_N] \\ \mathbf{v}[k_N+1] \\ \vdots \\ \mathbf{v}[k] \end{bmatrix} \tag{10}$$

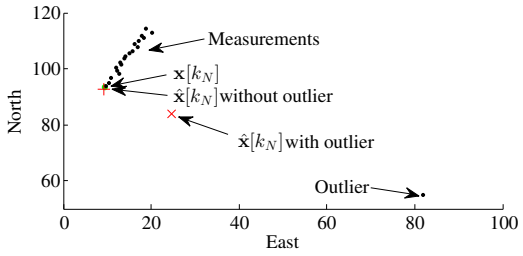

Fig. 1. MLE diverges when even one false measurement is introduced.

Note that (8) is similar to the psuedoinverse in (7), but weighted by the noise covariance. After computing the initial states $x[k_N]$, the Kalman filter is used to estimate the current states. The initial state is updated using the measurement $y[k_N]$, then the states are propagated forward and updated for each subsequent time step.

Unfortunately, even a single false measurement may cause the MLE to diverge. A simple example is shown in Fig. 1, and is analogous to Fischler and Bolles' demonstration that one false measurement can corrupt the least-squares solution [12]. In the following, we use a minimum subset of data to find the initial states of a dynamic target. In Section III-C we then use RANSAC to robustly estimate the target track.

B. Multiple Targets in Clutter

Let the number of true targets be $M[k] \geq 1$ and the clutter rate $\lambda_{\text{GE}} \geq 0$. Under these conditions, multiple measurements are received at each time step. As in the previous section, the objective is to estimate the initial states $\mathbf{x}_i[k_N], i = 1, \dots, M[k]$. We do so by applying the RANSAC paradigm and using only a minimal number of randomly selected measurements b to estimate the initial states, where b is defined as the observability index of the system. The observability index is determined by finding the smallest natural number b such that $\operatorname{rank}(\mathcal{O}_b) = \operatorname{rank}(\mathcal{O}_{b+1})$, where

$$\mathcal{O}_b = \begin{bmatrix} C & CA & \dots & CA^{b-1} \end{bmatrix}^\top.$$

Recall that ψ_k describes the number of measurements received during the k^{th} measurement scan. Let $\Psi_k = \sum_{\kappa=k_N}^{k} \psi_\kappa$ be the total number of measurements in the measurement window at time step k. For convenience, define $\mathbb{S} = \binom{Y_k}{b}$ as the set of $\binom{\Psi_k}{b} = \frac{\Psi_k!}{b!(\Psi_k-b)!}$ possible combinations of b observations, where each $S \in \mathbb{S}$ contains the minimum subset of points necessary to estimate the initial states $\mathbf{x}[k_N]$ according to the observability index b.

Since we receive multiple measurements per scan, we define an $m\Psi_k \times mN$ matrix Φ_k to associate the correct time indices to all the measurements received during that time step. Let $\mathbf{1}_{\psi_k}$ be defined as a column vector of ψ_k ones, and let I_m be defined as a $m \times m$ identity matrix. Also, let the $\otimes$ operator be the standard Kronecker product. Define

$$\Phi_k = \begin{bmatrix} \mathbf{1}_{\psi_{k_N}} \otimes I_m & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{\psi_{k_N+1}} \otimes I_m & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{1}_{\psi_k} \otimes I_m \end{bmatrix}. \tag{11}$$

With Φ_k, the windowed measurements are expressed as

$$\begin{bmatrix} \mathbf{y}_1[k_N] \\ \vdots \\ \mathbf{y}_{\psi_{k_N}}[k_N] \\ \mathbf{y}_1[k-N+2] \\ \vdots \\ \mathbf{y}_{\psi_k}[k] \end{bmatrix} = \Phi_k \mathcal{O} \mathbf{x}_i[k_N] + \Phi_k G_k W_k + \Phi_k V_k$$

$$Y_k = \Phi_k \mathcal{O} \mathbf{x}_i[k_N] + \Phi_k \xi_k. \tag{12}$$

The estimated initial state $\hat{\mathbf{x}}_i[k_N]$ can be found by modifying Equation (8). Let $\mathbf{q}$ be a random vector with b elements uniformly distributed over the set $\{1, 2, \dots, \Psi_k\}$. Define $S_{\mathbf{q}} \in \mathbb{S}$ as a minimum subset of b measurements randomly selected from the set of $\binom{\Psi_k}{b}$ measurement combinations, where $\mathbf{q}$ defines the indices of the selected measurements. To improve performance, we restrict $S_{\mathbf{q}}$ to contain no more than one measurement per scan. We define a binary indicator matrix that allows us to select specific measurements of interest from the measurement vector Y_k. In general, define $\mathbf{d} = [\mathbf{d}_1, \dots, \mathbf{d}_d]^\top$ to be a vector where each element is an index to a specific measurement in Y_k. Let $B(\mathbf{d})$ be defined as a binary indicator matrix of dimensions $md \times m\Psi_k$, where

$$B(\mathbf{d}) = \begin{bmatrix} \mathbf{0}_{m \times m(\mathbf{d}_1 - 1)} & I_m & \mathbf{0}_{m \times m(\Psi_k - \mathbf{d}_1)} \\ \vdots & \vdots & \vdots \\ \mathbf{0}_{m \times m(\mathbf{d}_d - 1)} & I_m & \mathbf{0}_{m \times m(\Psi_k - \mathbf{d}_d)} \end{bmatrix}. \tag{13}$$

Starting with Equation (12), the binary indicator matrix $B(\mathbf{q})$ is pre-multiplied to both sides. Letting $\overline{Y} = B(\mathbf{q})\, Y_k$, $\overline{\mathcal{O}} = B(\mathbf{q})\, \Phi_k\, \mathcal{O}$, and $\overline{\xi}_k = B(\mathbf{q})\, \Phi_k\, \xi_k$, we have

$$\overline{Y} = \overline{\mathcal{O}} \mathbf{x}[k_N] + \overline{\xi}_k, \tag{14}$$

where the noise term $\overline{\xi}_k$ has covariance $\overline{\Xi}$ given by

$$\begin{aligned} \overline{\Xi} &= E\left[\overline{\xi}_k\, \overline{\xi}_k^\top\right] \\ &= B(\mathbf{q})\, \Phi_k\, E\left[\xi_k\, \xi_k^\top\right] \Phi_k^\top\, B(\mathbf{q})^\top \\ &= B(\mathbf{q})\, \Phi_k\, \Xi\, \Phi_k^\top\, B(\mathbf{q})^\top. \end{aligned} \tag{15}$$

The MLE of the initial states and covariance is given by

$$\hat{\mathbf{x}}[k_N] = \left(\overline{\mathcal{O}}^\top \overline{\Xi}^{-1} \overline{\mathcal{O}}\right)^{-1} \overline{\mathcal{O}}^\top \overline{\Xi}^{-1} \overline{Y} \tag{16}$$

$$P[k_N] = \left(\overline{\mathcal{O}}^\top \overline{\Xi}^{-1} \overline{\mathcal{O}}\right)^{-1}. \tag{17}$$

Inliers to the track described by (16) can be found by finding the measurements whose residual is less than a threshold τ_{R},

$$\chi[k] = \left\{\kappa \in \{1, \ldots, \Psi_k\} : \left\| B(\kappa) Y_k - B(\kappa) \Phi_k \mathcal{O} \hat{\mathbf{x}}[k_N] \right\|_\infty < \tau_{\mathrm{R}}\right\}. \tag{18}$$

C. RANSAC for Dynamic Targets

Since the ML estimate in (16) is computed using a random minimum subset of measurements, the estimate will only accurately estimate the i^{th} target when the b measurements all measure the i^{th} target. For this reason, we use RANSAC to generate and test multiple hypotheses. The RANSAC algorithm consists of two repeated steps: A hypothesis generation step and a hypothesis validation step. During hypothesis generation, random minimum subsets of measurements are used to form a set of hypothesis tracks $\hat{\mathbf{x}}'$. During hypothesis validation, the support for each hypothesis is determined by counting the number of measurements whose residual to the hypothesis estimate is within a specified threshold τ_{R}. After ℓ hypotheses are tested, the hypothesis with the most support is selected and smoothed over the associated set of inliers [12]. An optional early termination threshold $\gamma \in (0, 1]$ can be used to stop generating hypotheses when $|\chi[k]| > \gamma N$.

The RANSAC implementation is summarized in Algorithm 1. Line 3 uses (16) to estimate a hypothesis of the initial conditions. Line 4 uses the hypothesized initial condition and (18) to find the subset of measurements are within a threshold τ_{R} of the predicted states for that time step, where $|\cdot|$ denotes the size of a set. These steps are then repeated up to ℓ times to find the hypothesis with the most inliers. Lines 11-20 are the Kalman filter prediction and update step.

Figure 2 gives a simple example. Suppose we want to fit the current measurement $y[k]$ to previous data assuming a target whose dynamics are described by a constant acceleration model. In this case, three measurements are needed to estimate the initial states at time k_N. For each hypothesis, two additional measurements are randomly selected from previous measurements to form a minimum subset, which is used to estimate the initial states $\hat{\mathbf{x}}'[k_N]$. Two hypotheses are considered, where the hypothesis with the most inliers is selected to be the most accurate. The final estimate $\hat{\mathbf{x}}[k]$ is found by propagating the initial states, updating as necessary using the associated inlier measurements with a Kalman filter.

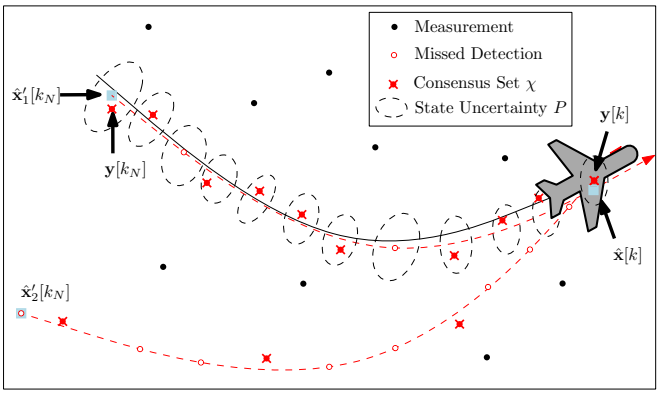

Fig. 2. RANSAC hypothesis tracks $\hat{\mathbf{x}}'_1$ and $\hat{\mathbf{x}}'_2$ are generated to fit the current measurement $\mathbf{y}[k]$. The hypothesis described by $\hat{\mathbf{x}}'_1$ is selected as the best track since it has ten inliers, as opposed to four for the other hypothesis. The final estimate is smoothed by propagating the initial states $\hat{\mathbf{x}}'_1[k_N]$ using a Kalman filter and updating as necessary using the associated inliers.

Algorithm 1 RANSAC for Dynamic Targets

1: **for** ℓ iterations, or until $|\chi'[k]| > \gamma N$ **do**
2: Select $S_{\mathbf{q}} \in \mathbb{S}$
3: $\hat{\mathbf{x}}'[k_N] = \left(\overline{\mathcal{O}}^\top \overline{\Xi}^{-1} \overline{\mathcal{O}}\right)^{-1} \overline{\mathcal{O}}^\top \overline{\Xi}^{-1} \overline{Y}$
4: $P'[k_N] = \left(\overline{\mathcal{O}}^\top \overline{\Xi}^{-1} \overline{\mathcal{O}}\right)^{-1}$
5: $\chi'[k] = \left\{\kappa \in \{1, \ldots, \Psi_k\} : \left\| B(\kappa) Y_k - B(\kappa) \Phi_k \mathcal{O} \hat{\mathbf{x}}[k_N] \right\|_\infty < \tau_{\mathrm{R}}\right\}$
6: Store track if $|\chi'[k]|$ is larger than all previous tracks.
7: **end for**
8: **for** $\kappa = k_N : k$ **do**
9: **if** $\kappa \neq k_N$ **then**
10: $\hat{\mathbf{x}}[\kappa] = A\hat{\mathbf{x}}[\kappa - 1]$, $P[\kappa] = AP[\kappa - 1]A^\top + Q$
11: **end if**
12: **for** $\{j \in \{1, \ldots, \psi_\kappa\} : y_j[\kappa] \in \chi'[k]\}$ **do**
13: $L = P[\kappa] C^\top \left(R + CP[\kappa]C^\top\right)^{-1}$
14: $P[\kappa] = (I_n - LC) P[\kappa]$
15: $\hat{\mathbf{x}}[\kappa] = \hat{\mathbf{x}}[\kappa] + L\left(y_j[\kappa] - C\hat{\mathbf{x}}[\kappa]\right)$
16: **end for**
17: **end for**

IV. Recursive RANSAC for Dynamic Targets

In the previous section, we designed a RANSAC-based technique to estimate the target states from a window of N measurements using a measurement at time k. The recursive RANSAC (R-RANSAC) algorithm can be used when subsequent measurements are received to update the set of stored tracks. The R-RANSAC algorithm consists of three main steps and is summarized in Algorithm 2. The first step in Line 2 is to predict existing R-RANSAC tracks forward using the Kalman prediction step. The second step in Line 4 classifies each measurement as either an inlier or an outlier. Line 6 uses each outlier to seed the RANSAC algorithm described in Algorithm 1 to generate a new track, while Line 8 uses inliers to the i^{th} track to update that track using the Kalman update step.

The remaining steps are for R-RANSAC track management. In Line 11, the consensus set for the i^{th} track is updated and the *inlier ratio* $\rho[k] = \frac{|\chi_i[k]|}{N}$ is recomputed. Lines 12 and 13 merges similar tracks and prunes the tracks to keep the best $\mathcal{M}$ tracks. Finally, Line 14 identifies as good tracks those whose inlier ratio and lifetime are above a threshold.

The R-RANSAC algorithm is related to the multiple hypothesis algorithm (MHT) [6] in that a set of hypotheses

Algorithm 2 Recursive RANSAC

for each k **do**
  $\hat{\mathbf{x}}_i[k] = A\hat{\mathbf{x}}_i[k-1], \forall i = \{1, \ldots, \mathcal{M}\}$.
  for each $\mathbf{y}_j[k], \forall j = \{1, \ldots, \psi_k\}$ **do**
    $\mathcal{I} = \{i : |y_j[k] - A\hat{x}_i[k]| < \tau_{\mathrm{R}}\}, \forall i = \{1, \ldots, \mathcal{M}\}$.
    if $|\mathcal{I}| = 0$ **then**
      Create new track found using Algorithm 1, where each $S_{\mathbf{q}}$ is chosen such that $\mathbf{y}_j[k] \in S$
    else
      Update i^{th} track using Kalman update $\forall i \in \mathcal{I}$.
    end if
  end for
  Update $\chi_i[k]$ and $\rho_i[k]$ $\forall i = \{1, \ldots, \mathcal{M}\}$.
  Eliminate similar tracks.
  Prune to keep $\mathcal{M}$ best tracks
  Determine good track, $\rho_i[k] \geq \tau_\rho$ and $\kappa_i \geq \tau_\kappa$.
end for

are stored in memory between time steps. Similar to MHT, the R-RANSAC framework naturally identifies target births and deaths. However, the complexity of the MHT algorithm is proportional to $O(c^{E[\psi_k]})$ to account for all data associations, whereas the worst-case R-RANSAC performance is proportional $O(E[\psi_k]^2)$ resulting from performing RANSAC on each measurement. Where MHT builds a hypothesis/track tree to describe the data associations, R-RANSAC maintains a fixed number of tracks that are most probable given the current measurements. In short, R-RANSAC is much easier to code and computationally more efficient than MHT, but sacrifices the potential of optimal estimates.

Three additional parameters must be specified for the R-RANSAC algorithm: the number of stored tracks $\mathcal{M}$, the track similarity threshold $\tau_{\mathbf{x}}$, and the good track threshold τ_ρ. Each parameter can be tuned to account for various trade-offs depending on the simulation. The number of stored tracks $\mathcal{M}$ should be greater than the expected number of true targets. Note that storing up to $\mathcal{M}$ tracks in memory inherently allows up to $\mathcal{M}$ targets at a time within the surveillance region, regardless of their birth or death time during the simulation. Similar tracks are pruned if the Mahalanobis distance is less then a threshold $\tau_{\mathbf{x}}$, leaving the track with the most inliers. The good track threshold τ_ρ can be increased to reduce the presence of false tracks, and decreased to reduce missed tracks. Increasing the measurement window length N leads to greater stability in the estimated number of good tracks. However, large N requires an increased delay before recognizing good tracks with sufficient inlier ratio and hinders detection of short-lived targets. We define an optional, but frequently used parameter to accept as good tracks those that persist for more than τ_κ time steps [17].

V. Results

During the last decade, the probability hypothesis density (PHD) filter has shown promise when applied to multiple target tracking (MTT). The theoretical and mathematical framework behind the PHD filter was introduced in [18]. Vo and Ma present a closed-form solution to the PHD recursion in [8], where they assume Gaussian dynamics and model the underlying random finite sets as Gaussian mixtures, referred to as the GM-PHD filter. A survey of PHD filter implementations is found in [19]. It has been shown that in some scenarios the PHD filter outperforms classical algorithms such as the multiple hypothesis tracker and the joint probabilistic data association filter [9]. In this section, we compare the GM-PHD filter as presented in [8] to the R-RANSAC algorithm.

Twelve targets are initialized as summarized Table I. Ten of the targets persist throughout the 200 second simulation, while two targets are born after the simulation starts and die before it ends. The simulation step size is $\delta k = \frac{1}{3}$ second. The state dynamics are defined by a 2D, constant-velocity model, and the targets are measured with probability of detection $p_{\mathrm{D}} = 0.95$. The target states are defined by $\mathbf{x} = [n\ e\ \dot{n}\ \dot{e}]$, and modeled as described by Example 1 in [8]. The PHD filter parameters are also the same in [8]. The parameters for R-RANSAC are: number of RANSAC iterations $\ell = 10$, inlier threshold $\tau_{\mathrm{R}} = 3\sigma_\epsilon$, where σ_ϵ is the measurement noise. The good track threshold is $\tau_\rho = 0.75$. The number of stored measurement scans is $N = 25$, the number of stored tracks is $\mathcal{M} = 25$, and the early termination threshold is $\gamma = \tau_\rho$. As with the PHD filter, the similar track threshold is when the Mahalanobis distance between two tracks is less than $\tau_{\mathbf{x}} = 4$. Finally, tracks must persist longer than $\tau_\kappa = 10$ time steps before being labeled a good track.

Both algorithms are capable of real-time performance. The algorithms require almost the same amount of computation, with R-RANSAC requiring 0.0739 sec per iteration and the PHD filter requiring 0.0754 sec per iteration in MATLAB on a 3 GHz Core$^{\mathrm{TM}}$2 Duo processor. However, the true computational complexity is dependent on the ratio of targets to clutter. As the number of average clutter returns increases, the PHD filter becomes notably faster than R-RANSAC since RANSAC must be used more frequently in the inner loop of the filter. Conversely, RRANSAC becomes notably faster as the ratio of targets to clutter increases since R-RANSAC only performs a Kalman update. This trade-off will be explored further in future work.

In this example, the ability of the PHD filter and R-RANSAC to detect the targets is approximately equal, at 95.7% and 96.0% respectively. The mean RMS error is comparable, with the R-RANSAC algorithm slightly more accurate with an RMS error of 5.6 m, compared to 7.6 m for the PHD filter. Most importantly, the false track rate is significantly lower for R-RANSAC at 0.023 false tracks per time step compared to 0.062 for the PHD filter. The number of targets over time and the inlier ratio are shown in Figure 4.

We note that track management can be incorporated by assigning a label to each good track which persists between time steps. Figure 4 shows the results when targets 1 and 3 and targets 9 and 10 cross at times 45 and 100, respectively. In the first case, the targets are moving so slow that their tracks were merged; in the second case, track uniqueness persisted during the crossing targets.

Target Number	1	2	3	4	5	6	7	8	9	10	11	12
Birth (sec)	0	0	0	0	0	0	0	0	0	0	25	25
Death (sec)	200	200	200	200	200	200	200	200	200	200	175	150
North Pos (m)	200	-200	200	600	600	-600	800	800	-900	-900	400	-800
East Pos (m)	0	-200	200	600	-600	-600	-800	800	-900	900	-400	-800
North Vel (m/s)	0	0	0	0	0	3	0	-5	0	0	0	9
East Vel (m/s)	2	0	-3	0	5	0	4	0	9	-9	1	0

TABLE I

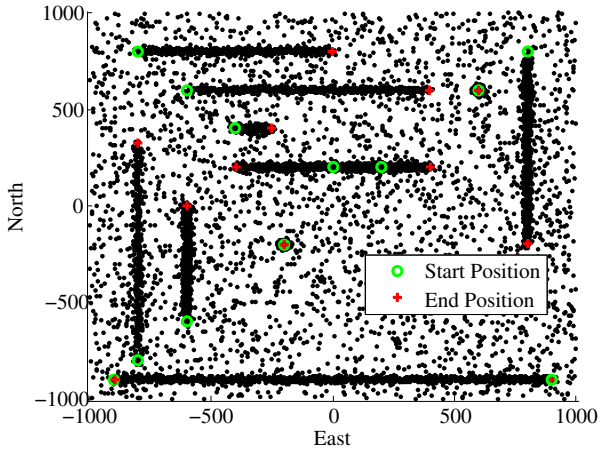

Fig. 3. Simulation measurement history.

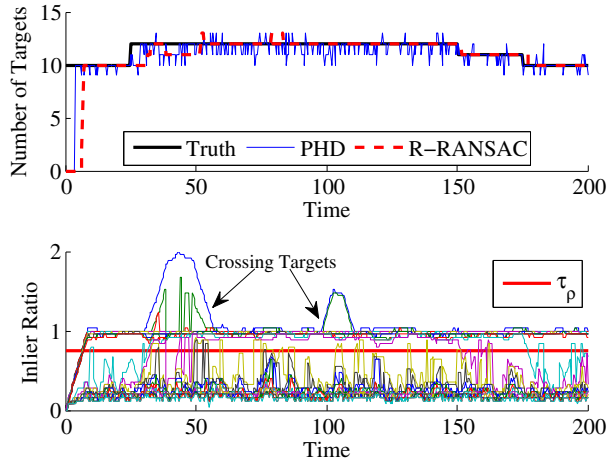

Fig. 4. Simulation results showing the true and estimated number of targets and the inlier ratio of the R-RANSAC filter during the simulation.

VI. Conclusion

In this work, we have applied the recursive RANSAC (R-RANSAC) framework developed in [14] to track dynamic targets. The new algorithm is capable of both estimating the true, dynamic number of targets and their states in clutter. R-RANSAC is compared to the state-of-the-art GM-PHD and is shown to be comparable in terms of accuracy and computational complexity.

Several avenues of possible research exist to extend and apply R-RANSAC. Similar to the PHD filter, we are considering methods of incorporating prior knowledge of target birth and spawning. This may improve the ability of R-RANSAC to more rapidly detect targets, although good tracks would still need to satisfy the good track threshold. Finally, direct comparisons with classical tracking algorithms, such as JPDA and MHT, should be explored.

References

[1] N. A. Tsokas and K. J. Kyriakopoulos, "Multi-Robot Multiple Hypothesis Tracking for Pedestrian Tracking", *Autonomous Robots*, vol. 32, no. 1, pp. 63–79, Nov. 2011.

[2] G. Thomaidis, L. Spinoulas, P. Lytrivis, M. Ahrholdt, G. Grubb, and A. Amditis, "Multiple Hypothesis Tracking for Automated Vehicle Perception", in *IEEE Intelligent Vehicles Symposium*, 2010, pp. 1122–1127.

[3] T. M. Wood, C. A. Yates, D. A. Wilkinson, and G. Rosser, "Simplified Multitarget Tracking Using the PHD Filter for Microscopic Video Data", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 5, pp. 702–713, 2012.

[4] J. K. Kuchar and L. C. Yang, "A Review of Conflict Detection and Resolution Modeling Methods", *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 179–189, 2000.

[5] Y. Bar-Shalom and T.E. Fortmann, *Tracking and Data Association*, Academic Press, 1988.

[6] D. Reid, "An Algorithm for Tracking Multiple Targets", *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[7] T.E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Multi-Target Tracking Using Joint Probabilistic Data Association", in *9th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, 1980, vol. 19, pp. 807–812.

[8] B.-N. Vo and W.-K. Ma, "The Gaussian Mixture Probability Hypothesis Density Filter", *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, 2006.

[9] R.P.S. Mahler, *Statistical Multisource-Multitarget Information Fusion*, Artech House, Norwood, 2007.

[10] O. Erdinc, P. Willett, and Y. Bar-Shalom, "Probability Hypothesis Density Filter for Multitarget Multisensor Tracking", in *7th International Conference on Information Fusion*, 2005, pp. 146–153.

[11] A. Vedaldi, H. Jin, P. Favaro, and S. Soatto, "KALMANSAC: Robust Filtering by Consensus", in *Tenth IEEE International Conference on Computer Vision*, 2005, pp. 633–640.

[12] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[13] S. Choi, T. Kim, and W. Yu, "Performance Evaluation of RANSAC Family", *Procedings of the British Machine Vision Conference*, 2009.

[14] P.C. Niedfeldt and R.W. Beard, "Recursive RANSAC: Multiple Signal Estimation with Outliers", in *9th IFAC Symposium on Nonlinear Control Systems*, 2013.

[15] P.C. Niedfeldt and R.W. Beard, "Convergence Analysis of the Recursive RANSAC Multiple Target Tracking Algorithm", *Under Review in IEEE Transactions on Automatric Control*.

[16] P.H.S. Torr and A. Zisserman, "MLESAC: A New Robust Estimator with Application to Estimating Image Geometry", *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[17] S.S. Blackman, "Multiple Hypothesis Tracking for Multiple Target Tracking", *Aerospace and Electronic Systems Magazine, IEEE*, vol. 19, no. 1, pp. 5–18, 2004.

[18] R.P.S. Mahler, "A Theoretical Foundation for the Stein-Winter" Probability Hypothesis Density (PHD)" Multitarget Tracking Approach", Tech. Rep., Lockheed Martin, 2000.

[19] R.P.S. Mahler, "A Survey of PHD filter and CPHD filter Implementations", in *Defense and Security Symposium*, 2007.