Bayesian Multiple Hypothesis Tracking of Merging and Splitting Targets

Alexandros Makris, Clémentine Prieur

Bayesian Multiple Hypothesis Tracking of Merging and Splitting Targets

Alexandros Makris, Clémentine Prieur

Abstract

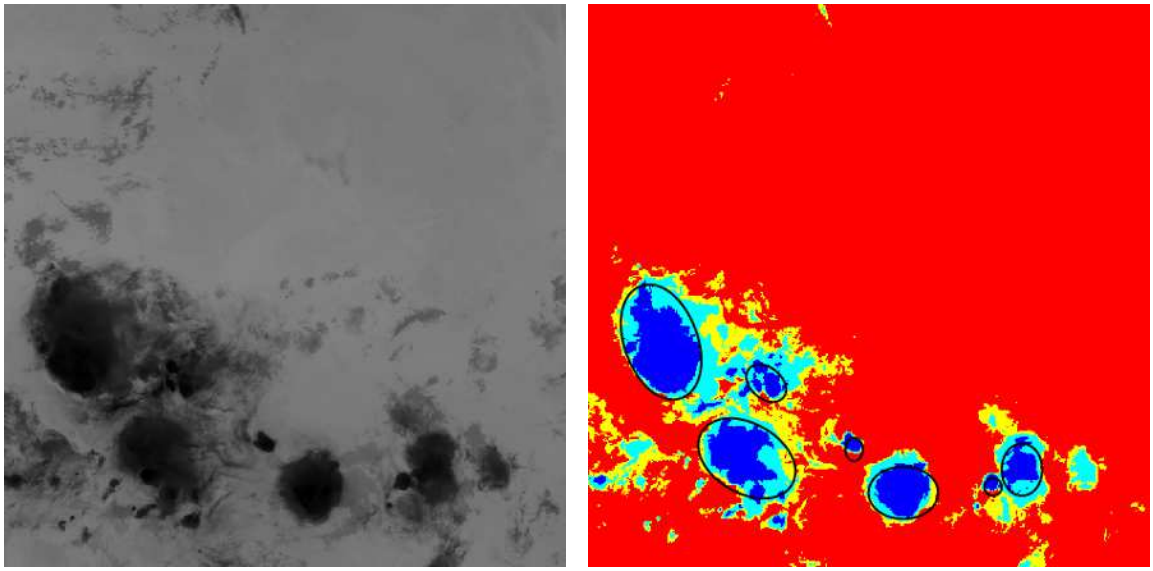
This paper presents a Bayesian model for the multiple target tracking problem that handles a varying number of splitting and merging targets applied to convective cloud tracking. The model decomposes the tracking solution into events and targets state. The events include target births, deaths, splits, and merges. The target state contains both the target positions and attributes. By updating the target attributes and conditioning the events on their updated values we can include high level domain knowledge into the system. This strategy improves the tracking accuracy and the computational efficiency since we focus only on likely events for each situation. A two-step multiple hypothesis tracking algorithm has been developed to estimate the model state. The proposed approach is tested by both simulation and real data for mesoscale convective systems tracking.

I. Introduction

Multi-target tracking (MTT) is a required component in many applications such as surveillance, aerospace, intelligent vehicles, and monitoring of geophysical processes. Tracking systems employ one or more sensors of various types (e.g. cameras, radar, laser). The aim of a tracking algorithm is to estimate the positions of a varying number of targets. This is performed using a series of typically noisy measurements with missed detections, inaccurate target positions, and false alarms. The most commonly studied cases are applications where the number of targets is constant and known or it varies due to target birth and death events. Less studied are the cases where additionally the events of splitting and merging of targets occur which is the main focus of this study. In geophysical applications, that provided the motivation for our approach, these types

MOISE Team, INRIA Rhônes-Alpes, Campus de Saint Martin d'Hères, 38041 Grenoble, FRANCE. Email: first-name.lastname@inria.fr

of events are quite common. Over the last decades, the study of geophysical phenomena benefits from a constantly increasing amount of data mainly due to the multiplication of remote sensing missions. Novel methods are required to exploit these raw data and extract useful high level information. In this study we are particularly interested in the behavior of Mesoscale Convective Systems (MCS) that have a considerable influence on the global climate. These systems are typically detected by applying a temperature threshold in infrared (IR) images as shown in Fig.1. Treating these systems as objects that are detected and tracked simplifies the subsequent analysis by providing a high level compact representation. In the following paragraphs we provide a literature review that covers the domains of general multi-target tracking, tracking of merging and splitting targets, and MCS detection and tracking.



(a) Meteosat Infrared Image (b) Ellipse Representation

Fig. 1. The input infrared images are thresholded and ellipses are fitted to the detected MCS.

Many approaches have been proposed in the literature to address the multi-target tracking problem [1], [2]. The Multiple Hypothesis Tracking (MHT) algorithm is one of the most successful [3], [4]. It maintains a set of possible association hypotheses between targets and measurements. MHT algorithms can be grouped into hypothesis oriented [3], and track oriented [5] depending on the strategy they use to form the hypotheses. Hypothesis oriented MHT maintains a set of consistent hypotheses whereas track oriented maintains a track tree which should be checked for consistency. The main problem of all the MHT algorithms is the combinatorial explosion of

hypotheses when the targets are close to each other or due to dense false alarms.

For the problem of tracking a known and fixed number of targets the Joint Probabilistic Data Association Filter (JPDAF) [6], [7] provides an efficient alternative to the MHT. Under the JPDAF framework the state of each target is updated individually, however several possible associations between target and measurements are considered. The algorithm updates the weights of each possible association. However, if the targets are not separated there is a combinatorial increase of components and pruning is required to ensure that their number remains fixed.

The probability hypothesis density (PHD) filter [8]–[11] overcomes the combinatorial problems of the MHT methods by only propagating the first-order moment of the multi-target state for a variable number of targets. The original PHD algorithm does not solve the target association problem, so individual target tracks cannot be computed. However, recent developments have overcome this drawback [12]–[14]. A comparison between the MHT and PHD algorithms can be found in [15]. The PHD filter however requires independent target evolution which is violated in the presence of merging events.

Particle filtering methods have been used extensively for multi-target tracking. The methods can be viewed as a generalization of MHT where the particles play the role of the hypotheses. These methods, in contrast with MHT, do not rely on a set of thresholded detections but they can directly exploit the raw sensor output. In [16], [17] efficient particle filtering schemes to directly estimate the Joint Multitarget Probability Density [18] are proposed. By estimating the join density the methods solve simultaneously the estimation and the target to measurements association problems. A combination of a Rao-Blackwellized particle filter with the Unscented Kalman Filter is used in [19] to track a variable number of vehicles using radar data. In [20] MTT is treated using multiple single target particle filters. A particle guiding technique is used to improve the predicted position of an object using the information from the neighboring objects. The method is tested in airborne vehicle tracking in urban areas.

The MTT algorithms face additional challenges when splitting and merging of targets have to be considered. There are not many methods in the literature that explicitly model these events. In [21], groups of targets that are allowed to merge or split are tracked using a probabilistic data association algorithm. The algorithm is applied in tracking groups of people. In [7], a MCMC algorithm is used to track targets that potentially share measurements. In [22], the authors use the track linking approach in order to track objects that potentially occlude each other for long

periods while maintaining their initial identities. The approach is tested in highway monitoring scenarios. All of the aforementioned approaches however concern cases where the targets are discrete entities and merging is a result of close interactions between them. In MCS tracking however, targets can actually merge or split.

For the problem of MCS tracking the previous approaches can be broadly grouped into deterministic [23]–[28] and stochastic ones [29], [30]. In [23], [24], deterministic MCS tracking methods based on target overlap between frames are presented. In [27], a level-set based tracking algorithm is proposed to cope with the shape variations of the convective clouds. Another deterministic tracking approach based on the curvelet transform is presented in [28]. An image segmentation method to detect salient MCS features that can be used for tracking is presented in [31]. Similarly, in [25], the MCS are detected and tracked in a single step using 3D segmentation in the spatio-temporal domain of a sequence of IR images. Initially the cores of the systems are detected by applying a low temperature threshold and then a region growing procedure is used to detect the warmer regions around each core. In [30], PHD filtering is used to exploit the temporal correlations to improve cloud detection. An approach that uses MHT and explicitly models merging and splitting events of MCS is presented in [29]. The authors make two main assumptions. Firstly, hypothesis generation is considered to be independent of the previous target positions and attributes (such as shape, size etc). Additionally target attributes are considered constant and are not updated during tracking in order to simplify the likelihood decomposition. Both these assumptions however have a negative impact on tracking accuracy. In [32] for instance, the authors show that by estimating jointly the target positions and attributes the tracking accuracy of a probabilistic MHT algorithm is improved.

In this work we introduce a novel probabilistic model and variable decomposition for the MTT problem. The proposed model can handle a variable number of targets and detect the target birth, death, split, and merge events. The model is related to the work in [29] but has several structural differences, most notably here we use a Bayesian approach whereas in [29] a maximum likelihood was used. The proposed probabilistic model decomposes the state into two sub-states: (i) the *events* and (ii) the *targets positions/attributes*. At each time step, the events sub-state is updated first conditioned to the previous targets sub-state. Then given the new events the target sub-state is updated. An efficient MHT algorithm is developed to implement the aforementioned procedure. The algorithm has two steps that follow the state decomposition of the probabilistic

model: at the first step it generates the event hypotheses and at the second step the association hypotheses. Since the events are conditioned on the targets sub-state of the previous time instant unlikely hypotheses are pruned in the first step thus we avoid examining many low likelihood asociations. Another advantage over [29], is that we include and estimate the target attributes in the state. This is beneficial for most applications since typically the attributes change over time. The proposed approach is tested experimentally with both simulation and real MCS data. We compare it to [29] and demonstrate the advantages of our approach.

The rest of the article is structured as follows. Section II provides the theoretical background on the Bayesian approach to MTT and the MHT algorithm. The proposed tracking model is described in Section III followed by the description of the implementation for MCS tracking in Section IV. In Section V we provide the experiments.

II. Background

A. Bayesian Formulation of the Multiple Target Tracking problem

Let $\boldsymbol{X}_t \equiv \{\boldsymbol{X}_{t,i}\}_{i=1}^{N_t}$ be the state at time t including the positions and attributes of the N_t targets and $\boldsymbol{Z}_t$ the corresponding set of measurements. Using the Bayesian approach the recursive formula to estimate the state for time 0 to t, $\boldsymbol{X}_{0:t}$, is:

$$p(\boldsymbol{X}_{0:t}|\boldsymbol{Z}_{1:t}) \propto p(\boldsymbol{X}_{0:t-1}|\boldsymbol{Z}_{1:t-1})p(\boldsymbol{Z}_t|\boldsymbol{X}_t)p(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}) \tag{1}$$

where we make the assumptions of markovian dynamics $\boldsymbol{X}_t \perp \boldsymbol{X}_{0:t-2}, \boldsymbol{Z}_{0:t-1}|\boldsymbol{X}_{t-1}$ and independent measurements given the state $\boldsymbol{Z}_t \perp \boldsymbol{X}_{0:t-1}, \boldsymbol{Z}_{1:t-1}, |\boldsymbol{X}_t$.

This equation gives the general solution of the MTT problem. However, exact calculation of the probability density functions (pdf) involved is difficult in most practical situations since it requires the calculation of the likelihood function and the propagation of the state over a high-dimensional space.

B. Standard Multiple Hypothesis Tracking Algorithm

The MHT algorithm is a popular approach to solve the MTT problem. It assumes that the sensor model provides a finite number, M_t, of distinct measurements $\boldsymbol{Z}_t \equiv \{\boldsymbol{Z}_{t,j}\}_{j=1}^{M_t}$. The algorithm associates these positions with the tracked targets. Many variations of this algorithm exist for constant or variable number of targets and for perfect or imperfect sensor models. An

imperfect sensor model requires the ability to cope with one or more of the following events: false alarms, missed detections, and inaccurate target positions.

To estimate the state, the MHT algorithm maintains at each time step t a set of hypotheses $\boldsymbol{H}_t$ that associate the targets with the measurements. Each hypothesis associates each target with at most one measurement and treats the possibly remaining measurements as false alarms. The algorithm defines a dynamic model over the hypotheses, this way it can model target associations, additions, and deletions. The joint pdf of object hypotheses and positions given the measurements is given by:

$$\begin{aligned} p(\boldsymbol{X}_{0:t}, \boldsymbol{H}_{0:t} | \boldsymbol{Z}_{1:t}) \propto & p(\boldsymbol{Z}_t | \boldsymbol{X}_t, \boldsymbol{H}_t) p(\boldsymbol{X}_t | \boldsymbol{H}_t, \boldsymbol{X}_{t-1}) \\ & p(\boldsymbol{H}_t | \boldsymbol{H}_{t-1}) p(\boldsymbol{X}_{0:t-1}, \boldsymbol{H}_{0:t-1} | \boldsymbol{Z}_{1:t-1}) \end{aligned} \tag{2}$$

where the dynamic model over the positions $p(\boldsymbol{X}_t | \boldsymbol{H}_t, \boldsymbol{X}_{t-1})$ updates the state of the targets that exist given the hypothesis $\boldsymbol{H}_t$. The combined measurement likelihood for each hypothesis is given as the product of the likelihood of each measurement:

$$p(\boldsymbol{Z}_t | \boldsymbol{X}_t, \boldsymbol{H}_t) = \prod_{j=1}^{M_t} p(\boldsymbol{Z}_{t,j} | \boldsymbol{X}_t, \boldsymbol{H}_t) \tag{3}$$

The likelihood of a single measurement $\boldsymbol{Z}_{t,j}$ is:

$$\begin{aligned} & p(\boldsymbol{Z}_{t,j} | \boldsymbol{X}_t, \boldsymbol{H}_t) = \\ & = \begin{cases} N(\boldsymbol{Z}_{t,j}; \boldsymbol{X}_{P_{t,j},t}, \Sigma_M) & \text{, if assigned to a target.} \\ U(\boldsymbol{Z}_{t,j}; V) & \text{, otherwise.} \end{cases} \end{aligned} \tag{4}$$

where V is the area of the region covered by the sensor, by $U(\boldsymbol{z}; V)$ we denote the pdf of the uniform distribution over that area evaluated at $\boldsymbol{z}$, and by $N(\boldsymbol{z}; \boldsymbol{\mu}, \Sigma)$ we denote the pdf of the multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix Σ, evaluated at $\boldsymbol{z}$.

Each hypothesis $\boldsymbol{H}_t$, includes the number of target births, the number of previous targets that continue, and the number of false alarms denoted as N_t^B, N_t^T, N_t^F respectively. Additionally it includes the variable $\mathbf{C}_t$ that represents a particular configuration of the M_t measurements into new targets, previous tracks, and clutter. Finally, it includes the random variable $\mathbf{P}_t$ which represents the assignment of the N_t^T previous tracks to the measurements. With the assumptions of Poisson false alarms and new targets and p_{ND} the probability of not detecting an existing target we have:

$$
\begin{aligned}
p(\boldsymbol{H}_t|\boldsymbol{H}_{0:t-1}) =& B(N_t^T; N_t, 1-p_{ND})F(N_t^F;\lambda^F)F(N_t^B;\lambda^B) \\
& p(\boldsymbol{C}_t|N_t^B, N_t^T, N_t^F)p(\boldsymbol{P}_t|\boldsymbol{C}_t) \\
=& \frac{N_t^B!N_t^F!}{M_t!}(1-p_{ND})^{N_t^T}p_{ND}^{N_t-N_t^T} \\
& F(N_t^F;\lambda^F)F(N_t^B;\lambda^B)
\end{aligned} \tag{5}
$$

where by $F(n,\lambda)$ we denote the probability mass function (pmf) of Poisson distribution with parameter λ evaluated at n and by $B(n;N,p)$ we denote the pmf of the binomial distribution with parameters N, p. Considering that all the configurations and associations are equally probable, and after the simplifications we have the final result of (5) [3].

Even by maintaining a constant number of hypotheses after each time step, the number of hypotheses that have to be evaluated can be prohibitively large. Various optimization techniques have been proposed in order to maintain an acceptable computational cost. The most common ones are gating and clustering the targets into independent groups [33]. Gating uses the kinematic model in order to form regions around the targets where the measurements are expected to be. Measurements outside each target's region are not considered for that target. Clustering decomposes the problem into several smaller ones by splitting the targets into independent groups that don't share common measurements. These heuristics however can work only when there is a restrictive kinematic model and the density of the measurements is low.

III. Proposed MTT Model

The proposed system is an MHT based approach for the problem of multiple target tracking. We consider that the number of targets is variable due to the following possible events: target births, deaths, splits, and merges. We split the hypothesis variable $\boldsymbol{H}$ of the standard MHT algorithm into the events variable $\boldsymbol{E}$ and the associations $\mathbf{P}$. $\boldsymbol{E}_t$ contains the information about the events that occur at the targets of the previous time step. $\boldsymbol{P}_t$ associates the new set of targets to the observations. The notation is regrouped for clarity in Table I.

To derive a recursive formula for the posterior we use the Dynamic Bayesian Network (DBN) formulation [34], [35]. The DBN which encodes the conditional independence assumptions is shown in Fig. 2. The main assumptions that we make are:

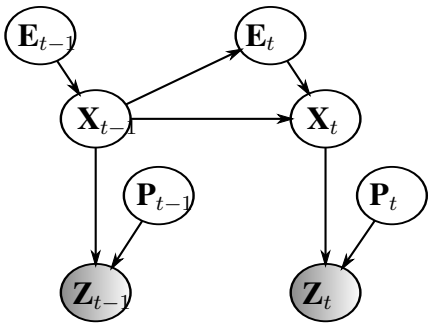


Fig. 2. Dynamic Bayesian network of the proposed model depicting slices $t-1$ and t of the temporal dimension.

- $\boldsymbol{E}_t \perp \boldsymbol{Z}_{1:t-1}, \boldsymbol{E}_{1:t-1} | \boldsymbol{X}_{t-1}$ We consider that at each time step the events are independent of the previous measurements and events given the previous state. This independence captures the intuition that the events only depend on the positions and attributes of the targets at the previous time step that are included in $\boldsymbol{X}_{t-1}$.
- $\boldsymbol{X}_t \perp \boldsymbol{Z}_{1:t-1}, \boldsymbol{E}_{1:t-1} | \boldsymbol{X}_{t-1}$ The current state is independent of the previous measurements and events given the previous state.

The recursive formula for the posterior is given by:

$$\begin{aligned} p(\boldsymbol{X}_{0:t}, \boldsymbol{E}_{0:t} | \boldsymbol{Z}_{1:t}) \propto & p(\boldsymbol{X}_{0:t-1}, \boldsymbol{E}_{0:t-1} | \boldsymbol{Z}_{1:t-1}) \\ & p(\boldsymbol{Z}_t | \boldsymbol{X}_t, \boldsymbol{P}_t) p(\boldsymbol{X}_t, \boldsymbol{E}_t | \boldsymbol{X}_{t-1}) \end{aligned} \tag{6}$$

A. State evolution model

The dynamic model is decomposed using the event random variable $\boldsymbol{E}$. This variable fully describes the events that take place at each time step and includes birth, death, split, merge, and trajectory continuation $(e_b, e_d, e_s, e_m, e_t)$. The dynamic model for the state is:

$$p(\boldsymbol{X}_t, \boldsymbol{E}_t | \boldsymbol{X}_{t-1}) = p(\boldsymbol{X}_t | \boldsymbol{E}_t, \boldsymbol{X}_{t-1}) p(\boldsymbol{E}_t | \boldsymbol{X}_{t-1}) \tag{7}$$

The presence of the merging events makes the evolution model of each target dependent on the other targets. For simplicity we authorize merging between two targets, however multiple pairs can merge on the same time step. Therefore, in order to calculate the probability of a set of events $\boldsymbol{E}_t$ we have to consider all the possible merging pair combinations between the targets.

Variable	Description
$\boldsymbol{X}_t$	State vector for all targets including their positions/attributes.
$\boldsymbol{H}_t$	Hypothesis variable including the events $\boldsymbol{E}_t$ and the associations $\boldsymbol{P}_t$.
$\boldsymbol{E}_t$	Event state variable.
$\boldsymbol{P}_t$	Target to measurements associations variable.
M_t	Total number of measurements.
N_t	Total number of targets.
N_t^B, N_t^D, N_t^T N_t^S, N_t^M	Number of targets that originate from birth, death, track continuation split, and merge events.
Q_t	Number of events.
N_h	Number of hypotheses of the MHT algorithm.
C	Number of possible event configurations.
K	Number of possible target to measurements associations.
$\boldsymbol{W}$	Weight of each MHT algorithm hypothesis.
Σ_D, Σ_M	Dynamic and measurement model covariance matrices.
$p_{M_0}, p_{S_0}, p_{D_0}$	Prior merge, split, and death event likelihood.
d_M, d_S, d_D	Merge, split, and death event likelihood distance metric.
$\sigma_M, \sigma_S, \sigma_D$	Merge, split, and death event likelihood variance.

TABLE I

PROPOSED TRACKING MODEL AND MHT ALGORITHM NOTATION

The number of pair combinations given that N_t^M targets merge is:

$$C_m(N_t^M) = \prod_{n=0}^{(N_t^M/2)-1} \binom{N_t^M - 2n}{2} \tag{8}$$

For the $N_t - N_t^M$ targets that don't merge the possible configurations are:

$$C_r(N_t - N_t^M) = 3^{N_t - N_t^M} \tag{9}$$

The total number of possible event configurations for N_t targets is:

$$C(N_t) = \sum_{n=0}^{\lfloor N_t/2 \rfloor} C_r(N_t - 2n) C_m(2n) \tag{10}$$

The prior of each event configuration $\boldsymbol{e}$ for N_t targets is given by:

$$p(\boldsymbol{E} = \boldsymbol{e}|\boldsymbol{X}) = \frac{L(\boldsymbol{e}|\boldsymbol{X})}{\sum_{c=1}^{C(N_t)} L(\boldsymbol{e}^c|\boldsymbol{X})} \tag{11}$$

where $L(\boldsymbol{e}|\boldsymbol{X})$ is the likelihood of the configuration $\boldsymbol{e}$ defined as:

$$L(\boldsymbol{e}|\boldsymbol{X}) = \prod_{n=1}^{Q_t} L(\boldsymbol{e}_n|\boldsymbol{X}) \tag{12}$$

where by $\boldsymbol{e}_n$ we denote the n-th event and Q_t is the total number of events.

B. Likelihood Model

The data likelihood for a state $\boldsymbol{X}_t$ is given by the product of individual target likelihoods considering the target to data associations $\boldsymbol{P}_t$:

$$p(\boldsymbol{Z}_t|\boldsymbol{X}_t, \boldsymbol{P}_t) = \prod_{i=1}^{N_t} p(\boldsymbol{Z}_{t,\boldsymbol{P}_{t,i}}|\boldsymbol{X}_{t,i}) \tag{13}$$

where $\boldsymbol{P}_{t,i}$ is the measurement index associated to the i-th target.

C. Tracking Algorithm Description

The posterior of (6) is approximated using a two-step MHT algorithm. The algorithm maintains at each time a number of N_h hypotheses. The set of hypotheses is comprised of C event configurations, with K^c association configurations each. The hypotheses are weighted by their posterior $\boldsymbol{W}_t^{c,p}$.The proposed algorithm exploits the variable decomposition into events and associations to reduce the computational cost. Firstly, starting from the previous targets the prior for each possible event combination is evaluated. The unlikely event combinations are then pruned by comparing the event prior with a threshold k_{pe}, before any data likelihood evaluations. Examples of events that are pruned at that stage are the merging between distant targets. For each event combination the likelihood is then calculated for every possible target to measurement association as in the standard MHT algorithm. Gating is used in order to avoid unnecessary likelihood evaluations. Additionally the targets are clustered in independent groups that don't compete for the same measurements. The steps of the algorithm for each cluster are summarized in Algorithm 1. To further reduce the computational cost after the update at time step t we find the most likely hypothesis at $t - \Delta t$ and we delete the rest and their respective hypothesis trees down to t. This strategy reduces the computational cost but not the performance of the algorithm since after a sufficiently large Δt we do not expect that new observations will influence the decision at the step $t - \Delta t$. The initialization at $t = 0$ is performed be forming a single hypothesis that considers all the measurements as births.

Algorithm 1 Proposed MHT Algorithm steps

Input: Weighted hypotheses set at t:
$\left\{\left\{\boldsymbol{E}^c_{0:t-1}, \boldsymbol{X}^{c,p}_{0:t-1}, \boldsymbol{W}^{c,p}_{t-1}\right\}_{p=1}^{K^c_{t-1}}\right\}_{c=1}^{C_{t-1}}$.

Input: Observations set at t: $\mathbf{Z}_t$.

Create all possible event configurations to estimate $p(\boldsymbol{E}_t|\boldsymbol{X}_{t-1})$.

Prune the low prior configurations: $p(\boldsymbol{E}^c_t|\boldsymbol{X}_{t-1}) < k_{pe}$.

for each configuration $c = 1$ to C_t **do**

Calculate the prior: $p(\boldsymbol{X}^c_t|\boldsymbol{X}_{t-1}, \boldsymbol{E}^c_t)$.

for each association $p = 1$ to K^c_t **do**

Calculate the data likelihood: $p(\boldsymbol{Z}_t|\boldsymbol{X}_t, \boldsymbol{P}^{c,p}_t)$.

Update the state using the calculated prior/likelihood.

end for

end for

Output: Weighted hypotheses set at t:
$\left\{\{\boldsymbol{E}^c_{0:t}, \boldsymbol{X}^{c,p}_{0:t}, \boldsymbol{W}^{c,p}_t\}_{p=1}^{K^c_t}\right\}_{c=1}^{C_t}$.

IV. System Implementation for MCS Tracking

In this section we apply the proposed MTT model into the tracking of mesoscale convective systems. The measurements are provided by infrared images. A pre-processing step applies thresholding to isolate the convection areas followed by ellipse fitting [23]. The resulting set of ellipses are then used by the MTT algorithm as input measurements. The object model for each target is also an ellipse. For each target the state vector is: $\boldsymbol{X}_i = [x, y, v_x, v_y, s, r, \phi]^T$. The elements of the vector represent the x-y position of the center, the x-y velocity of the center, the area, the ratio of the axes and the orientation of the major axis w.r.t. the horizontal respectively. In the following we describe the state evolution probabilities and the likelihood for this model.

The state evolution model for the events given in (11) requires the definition of the likelihood of each event type given the previous target states. For the MCS tracking problem we want to ensure that the merging only occurs when the two targets overlap, that splitting is more probable for bigger targets, and that death is more probable for smaller ones. More precisely, for the

merging events we have:

$$L(\boldsymbol{E}_n = e_m|\boldsymbol{X}_{i1}, \boldsymbol{X}_{i2}) = p_{M_0} exp\left\{-\frac{d_M}{2\sigma_M^2}\right\}$$
$$d_M = 1 - \frac{s_\cap}{min(s_1, s_2)} \tag{14}$$

where s_1, s_2 are the areas of the two ellipses and $s_\cap$ is the area of their intersection and p_{M_0} is the probability of a merge for two ellipses with a complete overlap.

The likelihood for the splits is given by:

$$L(\boldsymbol{E}_n = e_s|\boldsymbol{X}_i) = p_{S_0} exp\left\{-\frac{d_S}{2\sigma_S^2}\right\}$$
$$d_S = \frac{s_{max} - s_i}{s_{max} - s_{min}} \tag{15}$$

where s_i is the area of the ellipse and p_{S_0} is the probability of split of an ellipse with the maximum area. The min and max area of an ellipse is denoted by s_{min} and s_{max} respectively.

The death event likelihood is:

$$L(\boldsymbol{E}_n = e_d|\boldsymbol{X}_i) = p_{D_0} exp\left\{-\frac{d_D}{2\sigma_D^2}\right\}$$
$$d_D = \frac{s_i - s_{min}}{s_{max} - s_{min}} \tag{16}$$

where p_{D_0} is the probability of death of an ellipse with the minimum area.

Given the event configuration the dynamic model is decomposed as:

$$p(\boldsymbol{X}_t|\boldsymbol{E}_t, \boldsymbol{X}_{t-1}) = \prod_{i=1}^{N_t} p(\boldsymbol{X}_{t,i}|pa(\boldsymbol{X}_{t,i})) \tag{17}$$

where by $pa(\boldsymbol{X}_{t,i})$ we denote the parents of the i-th target. Depending on the event, a target might have zero parents in the case of a birth, one in the case of track continuation or split, and two in a case of a merge. We represent the dynamics of each target using a constant velocity model with additive noise:

$$p(\boldsymbol{X}_{t,i}|pa(\boldsymbol{X}_{t,i})) = \mathcal{N}(\boldsymbol{X}_{t,i}; \boldsymbol{X}_{t,i}^-, \Sigma_D) \tag{18}$$

where $\boldsymbol{X}_{t,i}^-$ is the predicted target position using the dynamic model and Σ_D is the diagonal covariance matrix. For the case of merging events the predicted velocity and position are given as the weighted average of the parents' velocities and positions where the contribution of each

parent is weighted by its size. The predicted area is calculated as the sum of the area of the parents:

$$\begin{aligned} s_m^- &= s_{pa1} + s_{pa2} \\ x_m^- &= \frac{x_{pa1}s_{pa1} + x_{pa2}s_{pa2}}{s_{pa1} + s_{pa2}} \end{aligned} \tag{19}$$

Where s_{pa1}, s_{pa2} are the areas of the parent ellipses. Similarly to x_m^- we calculate y_m^-, v_{xm}^-, and v_{ym}^-.

For the splitting events the new ellipses are given the predicted position and velocity of the parent ellipse using the standard dynamic model. For a given split ratio β_s, the predicted sizes of the new ellipses are:

$$\begin{aligned} s_{s1}^- &= \beta_s s_{pa} \\ s_{s2}^- &= (1 - \beta_s) s_{pa} \end{aligned} \tag{20}$$

The likelihood model given the target positions and associations is the same as the one of the standard MHT algorithm defined in (3). It calculates the discrepancy between the predicted and the measurement ellipses.

V. Experiments

In this section we present the experiments that we performed to test the proposed tracking method. Experiments on both, a simulated data-set, and on real MCS data were performed. We demonstrate that by applying the proposed MHT algorithm we can accurately and efficiently track a variable number of merging and splitting targets. The algorithm is tested in the presence of false alarms, missed detections, and inaccurate target positions. We compare our approach with the one presented in [29] which also takes into account the same types of events.

A. Simulation Experiments

For the first series of experiments we use simulated data to quantitatively evaluate our method. The data-set contains a series of simulations with 1000 frames each. We vary the complexity of the simulations by varying the false alarms to targets ratio from 0 to 2 (which correspond to values for the λ^F from 0 to 3), the missed detections probability p_{ND} from 0 to 0.5, and the observation noise variance ratio R_{wn} from 0 to 3. The variance of the observation noise is

expressed as a ratio w.r.t. the variance of the dynamic model Σ_D. The parameters of the dataset are summarized in Table II. The details of the simulation are given in Appendix A.

We compare the proposed approach with the approach of [29]. In our model both position and attributes are included in the object model state and are estimated. The proposed model architecture considers the dependence of the target events at time t on the target state at time $t-1$. We show experimentally that this strategy has clear benefits over the independent event prior that is used in [29]. Firstly, we can take into account the dependence of the events on the target attributes that is commonly required in the applications. For example death events might be more probable for targets with a small size whereas split events could occur more often to bigger targets. Not taking these dependencies into account negatively affects the tracking accuracy. Furthermore, we show that by considering these dependencies on the event generation stage decreases the computational cost of the approach compared to considering them during the likelihood calculation stage. This happens because we prune the unlikely event combinations in the event generation stage so we avoid many unnecessary likelihood evaluations when the new measurements become available. We compare four methods in order to measure the performance impact for each of the aforementioned method differences:

1) **MHT-S** This is the baseline method of [29].
2) **MHT-E** The events are conditioned on the previous target states.
3) **MHT-A** The target attributes are updated.
4) **MHT-EA** The full proposed method with both conditioned events and target attributes update.

For the initial tests we set the model parameters to match the simulation parameters of Table II. The parameters that concern the event priors are only used by **MHT-E**, **MHT-EA**, the rest of the methods assume uniform event priors.

We use two comparison strategies to evaluate the methods. For an overall comparison of the general tracking accuracy we calculate the Optimal Subpattern Assignment (OSPA) measure [36] which is designed for the MTT problem. This metric calculates the discrepancy between the real target positions and the tracking estimate at each time step:

$$d_p^c(X,Y) = \left(\frac{1}{n}\left(\min_{\pi\in\Pi_n}\sum_{i=1}^{m} d^{(c)}(x_i, y_{\pi(i)})^p + c^p(n-m)\right)\right)^{1/p} \tag{21}$$

Where by $d^{(c)}(x,y) = min(c, d(x,y))$ we denote the distance between x, y cut off at c, and by

Π_n the set of permutations on $\{1, 2, ..., n\}$, $X = \{x_1..., x_m\}$,$Y = \{y_1..., y_n\}$, we consider that $m < n$. For our experiments we used $c = 100$ and $p = 2$.

In order to particularly focus on the capability to correctly identify the events, which is crucial for our application, we also calculate the precision, recall, and F1 score [37] for the event occurrences. This way we can assess the capability of the system to correctly identify the event occurrences and if required focus on particular event types. The metrics for each event type e are defined as:

$$\begin{aligned} P_e &= \frac{TP_e}{TP_e + FP_e}, \\ R_e &= \frac{TP_e}{TP_e + FN_e}, \\ F_e &= \frac{2P_eR_e}{P_e + R_e}, \end{aligned} \tag{22}$$

where by P_e, R_e, F_e we denote the precision, recall, and F1 score for the event type e respectively and by TP_e, FP_e, FN_e we denote the number of true-positive, false-positive, and false-negative events respectively. The assessment is performed on a per-frame basis. An event occurrence at time t is classified as true-positive if in both groundtruth and tracking output:

- Has the same event label.
- The target(s) that produced the event were associated with the same measurements at times $t - 1$ and t.

The results are displayed in Fig. 3. The performance of the proposed method, **MHT-EA** is generally higher compared to **MHT-S**. In the first row of Fig. 3 we examine the performance of the methods for different false alarms rates. As is expected the performance of all the methods drops with the increase of false alarm rate but the drop is lower for the proposed method. This is explained because the use of event priors allows our method to better discriminate between different event hypotheses in the presence of dense clutter. In the second row of Fig. 3 the influence of noise of the observed positions is studied. As we see again our methods performance is steadily higher for the different levels of observation noise. Finally, in the third row of Fig. 3, we plot the metrics w.r.t. the M/D rate. In this case we see that the performance of our approach is clearly superior till the rate reaches about 0.25. After that all methods have similar performance which drops very rapidly. This behavior is expected since when more than 25% of the targets are not observed the difficulty of the problem rises significantly.

Parameter	T	N_{max}	R_{wn}	p_{ND}	λ^F	λ^B	σ_D	p_{D_0}	σ_S	p_{S_0}	σ_M	p_{M_0}
Value	1000	6	$[0-2.0]$	$[0-0.50]$	$[0-3.00]$	0.30	0.05	0.50	0.25	0.85	0.25	0.85

TABLE II

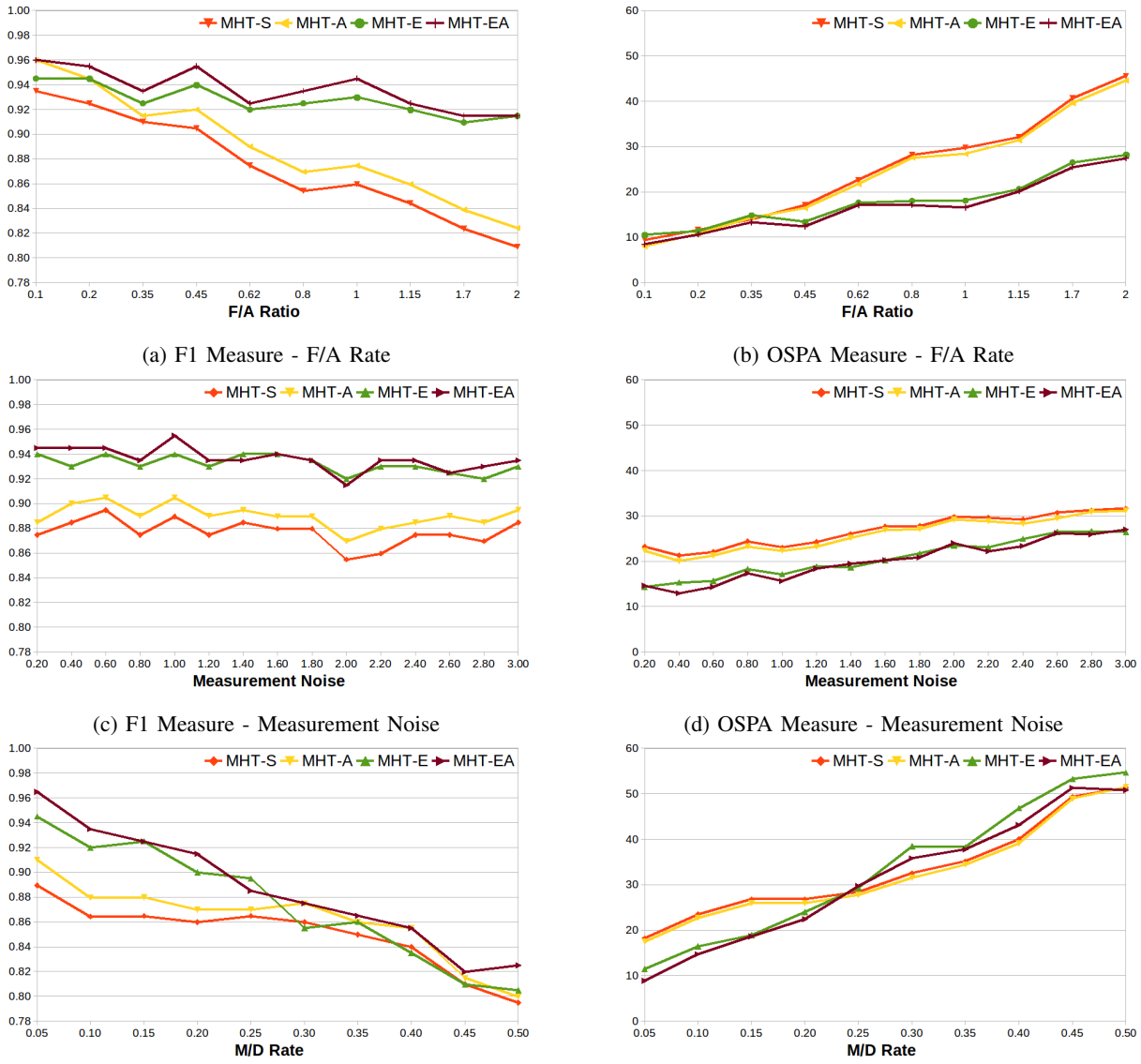
DATASET PARAMETERS

We observe that the most significant boost in performance results from the event conditioning on the previous states. The performance is less influenced by the update of the attributes. This is due to the fact that even though the attributes are not updated during normal tracking, after each event their values are re-initialized. Between events, even though their real values change, the likelihood of the correct associations is still higher and thus does not affect too much the tracking performance.

The conditioning of the events on the previous target state assumes that there is a dependence that can be modeled by an appropriate distribution. This way we can inject high level knowledge for the target behavior of the specific problem. For the target deaths for example this distribution might represent the fact that smaller targets are more likely to disappear. In the previous results we used the same distribution parameters for simulation and for tracking. However, in a real application we have an estimate of the distribution parameters but we don't know the exact values. Therefore we need to test the sensitivity of our model to these parameters. More particularly we examine how the variance parameters for death, merge, and split events (σ_D, σ_M, and σ_S) affects the performance. To do so we vary the values for these parameters and we rerun the experiments with these new values. Figure 4 contains the results of this experiment. As a means of comparison we also included the results when no event conditioning takes place. We observe that even with big under or over-estimation of the parameters the performance of our approach is still higher compared to the baseline. This shows that the method is expected to work in real world applications even when we only have rough estimates of these parameters.

B. MCS Tracking Experiments

In this section we evaluate our method using real data of Sahelian Mesoscale Convective Systems. The data are taken from the infrared Meteosat satellite images and concern the sahelian geographic zone with longitude range from -20 to 30 degrees and latitude range from 0 to 25 degrees. They concern the active season of the year 2005 with one image every $30min$, 5225



(a) F1 Measure - F/A Rate

(b) OSPA Measure - F/A Rate

(c) F1 Measure - Measurement Noise

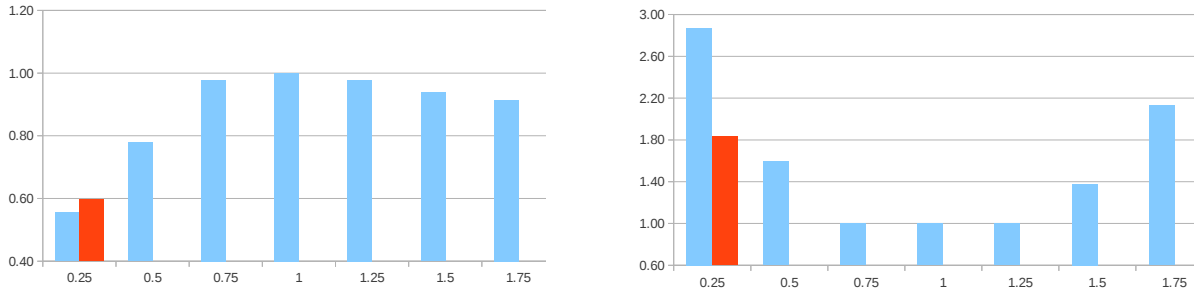
(d) OSPA Measure - Measurement Noise

(e) F1 Measure - M/D Rate

(f) OSPA Measure - M/D Rate

Fig. 3. Simulation Results. Performance of the tested methods in terms of F1 and OSPA measures on a series of simulations. Top Row: Simulations with varying F/A rate. Middle Row: Simulations with varying measurement noise. Bottom Row: Simulations with varying M/D rate.

images in total. The MCS are isolated using binarization and ellipse fitting is performed in order to get a compact representation [23]. An example image of the region is shown in Fig. 5. The tracking parameters are estimated using the deterministic tracking output of [23]. The variances of the dynamic and measurement model for each dimension (diagonal values of the covariance matrices Σ_D, Σ_M) are equal and set to:



(a) F1 Measure

(b) OSPA Measure

Fig. 4. Sensitivity to the variance of the events prior. In the x-axis we have the ratio R_{var} by which we change all the three variance parameters σ_D^{tr}, σ_M^{tr}, and σ_S^{tr} compared to their real values that we used during simulation σ_D^{sim}, σ_M^{sim}, and σ_S^{sim}, so we have $\sigma_D^{tr} = R_{var}\sigma_D^{sim}$, $\sigma_M^{tr} = R_{var}\sigma_M^{sim}$, $\sigma_S^{tr} = R_{var}\sigma_S^{sim}$. For each ratio we run the experiments and we compare the performance of the method as the ratio of F1 score and OSPA measure compared with the run with the true values **MHT-E** (blue bars). For comparison we include the performance ratios of the method without conditioning, **MHT-S** (red bars). Parameters: $N_h = 5$, $N_{max} = 9$.

$[x : 0.1,\ y : 0.1,\ v_x : 0.15,\ v_y : 0.15,\ r : (0.7(r_{max} - r_{min}))^2,\ s : (0.7(s_{max} - s_{min}))^2,\ \phi : \pi^2/30]$. The variance units are in $degrees^2$ for the positions, $(degrees/min)^2$ for the velocities, $degrees^4$ for the area, and $rads^2$ for the angle of the major axis. By $r_{max}, r_{min}, s_{max}, s_{min}$ we denote the minimum and maximum values of the axis ratios and sizes within the dataset. The event prior parameters are the same ones used for the simulation experiments (Table II).

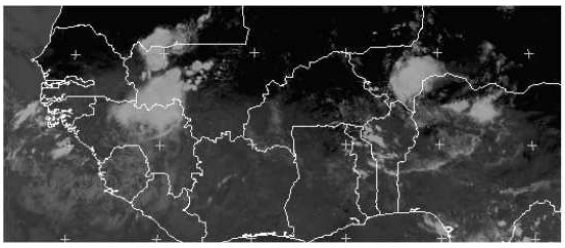


Fig. 5. Infrared image of the region that we used for the experiments.

Contrary to the simulated data case, with real data there is no ground-truth in order to quantitatively measure the performance of the tracking method. This is because there is no consistent way of labeling the events that take place. By visually inspecting the tracking results however we observe that the method is capable of interpreting the events in a similar way as a field expert. Fig. 6 shows a series of snapshots of a tracking sequence. We observe the

challenging nature of the data with many targets, large size variations and many merging and splitting events. The trajectories and the events are marked.

VI. Conclusions and Future Work

In this work we presented a new Bayesian MTT model and a two-step MHT algorithm capable of handling a varying number of targets that undergo births, deaths, merges, and splits. The motivation for the method comes from the problem of tracking MCS using infrared satellite images. However, the provided methodology is generic and can be applied in other types of data that contain the same types of events.

As future work we consider to extend the proposed method to raw sensor data. For the MCS tracking for instance, we plan to investigate the possibility of tracking using the infrared data directly without any pre-processing. This way we could avoid the artifacts that are caused by the thresholding and the ellipse fitting procedures. To do so the same probabilistic model is applicable but a particle filter based approximation should replace the MHT algorithm. We believe that, the proposed model that splits the state into events and target positions/attributes will help to reduce the number of particles that are required to approximate the underlying distributions.

Appendix A
Simulation Model Details

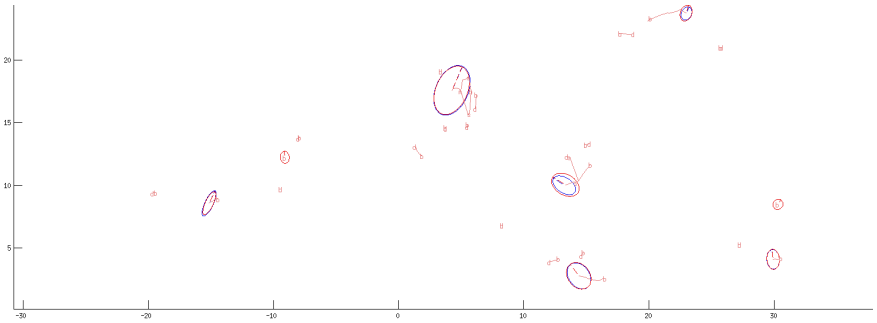
In this section we describe in detail the simulation procedure. It follows the state evolution model described in Sec. III-A, using the decomposition into events and positions. The steps of the simulation algorithm are the following:

1) Calculate the likelihood of the events using (14),(15), (16).
2) Sample from the pmf of (11).
3) Calculate the target states given the dynamic model.

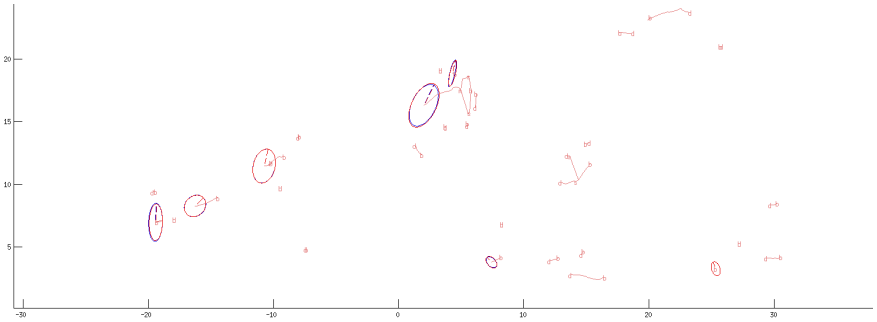
The first two steps of the approach are fully described in Sec. III-A. Here we are giving the details of the calculation of the targets' states. The simulation takes place within a predetermined window. Targets that exit this window are automatically deleted.

A. Target Birth

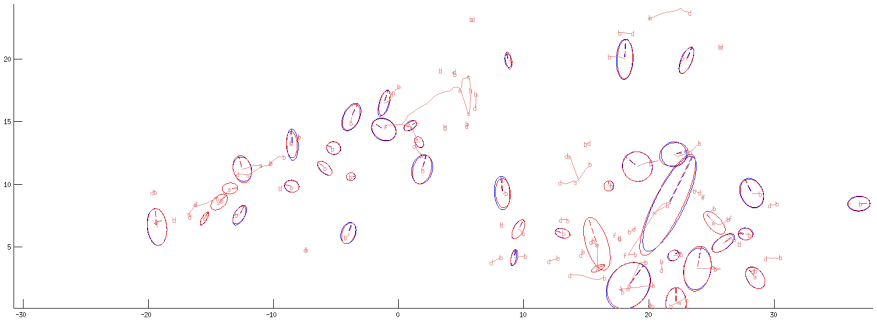
The position distribution for the new targets is uniformly distributed within the bounded simulation window. Their velocities, size, axis ratio, and orientation follow normal distribution



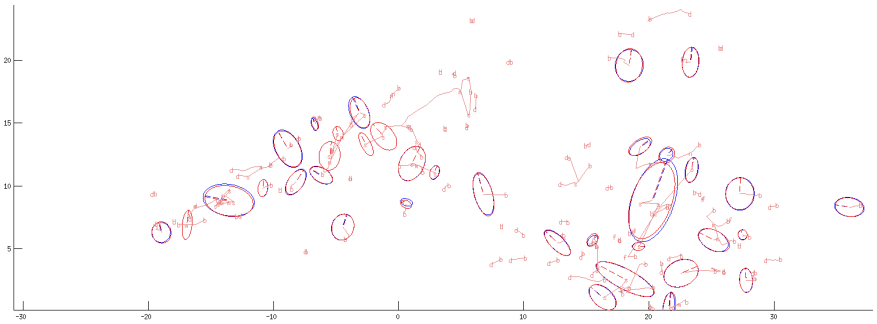
(a) t=1



(b) t=22



(c) t=32



(d) t=37

Fig. 6. MCS 2005 Tracking Sequence. For each time step the active systems and the past trajectories and events are plotted. The tracking estimates are plotted in red while the measurements in blue. The events are marked by their initial letters: (b)irths, (d)eaths, (m)erges, and (s)plits.

using mean and variance priors that simulate the behavior of the real MCS systems birth process (small systems with westward velocities and vertical orientation of the main axis).

B. Track Continuation

For the targets that continue their track we sample from the following pdf:

$$p(\boldsymbol{X}_{t,i}|pa(\boldsymbol{X}_{t,i})) \propto \mathcal{N}(\boldsymbol{X}_{t,i}; \boldsymbol{X}^{-}_{t,i}, \Sigma_{pa})\mathcal{N}(\boldsymbol{X}_{t,i}; \boldsymbol{X}^{pr}_{t,i}, \Sigma_{pr}) \tag{23}$$

The above pdf takes into account the previous state of the target and a prior general direction of motion with mean $\boldsymbol{X}^{pr}_{t,i}$ and a generally large variance. This way the target maintain a smooth trajectory that is forced toward a prior direction.

C. Split

For the case of a splitting target we use a two step procedure. Initially we simulate the new position as for normal track continuation. Then we perform the splitting on that new position. The split has two parameters, the split angle θ_s which is the angle of the split axis w.r.t. the axis of the parent ellipse and the area ratio of the two resulting ellipses β_s. Both parameters are selected from uniform distributions over their respective value ranges. Given the above parameters the new ellipses are created using the following two criteria: (i) Preservation of the total area, (ii) maximum overlap with the parent ellipse. In the following we describe the calculation of the state vectors $\boldsymbol{X}_{s1}$, $\boldsymbol{X}_{s2}$ of the ellipses that result from the split of $\boldsymbol{X}_{pa}$. The statte vector of an ellipse is: $\boldsymbol{X}_i = [x, y, v_x, v_y, s, r, \phi]^T$.

The areas of the new ellipses s_{s1} and s_{s2} is given by:

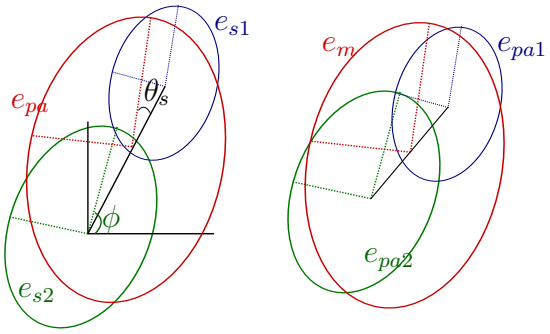
$$\begin{aligned} s_{s1} &= \beta_s s_{pa} \\ s_{s2} &= (1 - \beta_s) s_{pa} \end{aligned} \tag{24}$$

where s_{pa} is the area of the parent ellipse.

To enforce the second criterion we calculate the axes ratio of the new ellipses as:

$$r_s = r_{pa}(1.5sin(\theta_s) + 0.5) \tag{25}$$

where r_{pa} is the axes ratio of the parent ellipse. The above function goes from 0.5 to 2 as the split angle goes from 0 to $\pi/2$.



(a) Split (b) Merge

Fig. 7. Represantation of parent and resulting ellipses for splitting and merging events.

The radius of the ellipse on the split direction is:

$$\rho_s = \frac{a_{pa} b_{pa}}{\sqrt{(b_{pa} cos(\theta_s))^2 + (a_{pa} sin(\theta_s))^2}} \tag{26}$$

where a_{pa}, b_{pa} are the major and the minor axes of the parent ellipse.

Using this radius we calculate the coordinate of the centers of the new ellipses as:

$$\begin{aligned} x_{s1} &= x_{pa} + 0.5\rho_s cos(\theta_s + \phi_{pa}) \\ y_{s1} &= y_{pa} + 0.5\rho_s sin(\theta_s + \phi_{pa}) \\ x_{s2} &= x_{pa} - 0.5\rho_s cos(\theta_s + \phi_{pa}) \\ y_{s2} &= y_{pa} - 0.5\rho_s sin(\theta_s + \phi_{pa}) \end{aligned} \tag{27}$$

where x_{pa}, y_{pa} are the coordinates of the center of the parent ellipse. Finally, the angle and speed of the new ellipses are the same to the parent's angle: $\phi_{s1} = \phi_{s2} = \phi_{pa}$, $v_{x,s1} = v_{x,s2} = v_{x,pa}$, $v_{y,s1} = v_{y,s2} = v_{y,pa}$.

D. Merge

The same criteria that we have for the split hold for the merge. We ensure that the total area is preserved and we attempt to maximize the overlap between the new and the parent ellipses. The state of the new ellipse is given by:

$$
\begin{aligned}
x_m &= w_{pa1}x_{pa1} + w_{pa2}x_{pa2} \\
y_m &= w_{pa1}y_{pa1} + w_{pa2}y_{pa2} \\
\phi_m &= w_{pa1}\phi_{pa1} + w_{pa2}\phi_{pa2} \\
v_{x,m} &= w_{pa1}v_{x,pa1} + w_{pa2}v_{x,pa2} \\
v_{y,m} &= w_{pa1}v_{y,pa1} + w_{pa2}v_{y,pa2} \\
a_m &= \gamma_m * (w_{pa1}a_{pa1} + w_{pa2}a_{pa2}) \\
b_m &= \gamma_m * (w_{pa1}b_{pa1} + w_{pa2}b_{pa2}) \\
s_m &= \pi a_m b_m \\
r_m &= a_m / b_m
\end{aligned}
\tag{28}
$$

where γ_m is a normalizing parameter that ensures the preservation of the total area and w_{pa1}, w_{pa2} are the weights of the two parent ellipses that are proportional to their area.

REFERENCES

[1] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and data fusion: A Handbook of Algorithms.* Storrs, 2011.

[2] G. Pulford, "Taxonomy of multiple target tracking methods," *Radar, Sonar and Navigation, IEE Proceedings* -, vol. 152, no. 5, pp. 291 – 304, october 2005.

[3] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, pp. 843–854, 1979.

[4] S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 19, no. 1, pp. 5 –18, jan. 2004.

[5] T. Kurien, "Framework for integrated tracking and identification of multiple targets," in *Digital Avionics Systems Conference, 1991. Proceedings., IEEE/AIAA 10th*, oct 1991, pp. 362 –366.

[6] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *Oceanic Engineering, IEEE Journal of*, vol. 8, no. 3, pp. 173 – 184, jul 1983.

[7] Z. Khan, T. Balch, and F. Dellaert, "Multitarget tracking with split and merged measurements," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, june 2005, pp. 605 – 610 vol. 1.

[8] R. Mahler, "Multitarget bayes filtering via first-order multitarget moments," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 39, no. 4, pp. 1152 – 1178, oct. 2003.

[9] B.-N. Vo, S. Singh, and A. Doucet, "Sequential monte carlo methods for multitarget filtering with random finite sets," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 41, no. 4, pp. 1224 – 1245, oct. 2005.

[10] B.-N. Vo and W.-K. Ma, "The gaussian mixture probability hypothesis density filter," *Signal Processing, IEEE Transactions on*, vol. 54, no. 11, pp. 4091 –4104, nov. 2006.

[11] B.-T. Vo, B.-N. Vo, and A. Cantoni, "The cardinalized probability hypothesis density filter for linear gaussian multi-target models," in *Information Sciences and Systems, 2006 40th Annual Conference on*, 2006, pp. 681–686.

[12] K. Panta, B.-N. Vo, and S. Singh, "Novel data association schemes for the probability hypothesis density filter," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 43, no. 2, pp. 556–570, 2007.

[13] M. Ulmke, O. Erdinc, and P. Willett, "Gaussian mixture cardinalized phd filter for ground moving target tracking," in *Information Fusion, 2007 10th International Conference on*, 2007, pp. 1–8.

[14] K. Panta, D. Clark, and B.-N. Vo, "Data association and track management for the gaussian mixture probability hypothesis density filter," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 45, no. 3, pp. 1003–1016, 2009.

[15] K. Panta, B.-N. Vo, S. Singh, and A. Doucet, "Probability hypothesis density filter versus multiple hypothesis tracking," pp. 284–295, 2004. [Online]. Available: +http://dx.doi.org/10.1117/12.543357

[16] C. Kreucher, K. Kastella, and A. Hero, "Multitarget tracking using the joint multitarget probability density," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 41, no. 4, pp. 1396–1414, 2005.

[17] M. Morelande, C. Kreucher, and K. Kastella, "A bayesian approach to multiple target detection and tracking," *Signal Processing, IEEE Transactions on*, vol. 55, no. 5, pp. 1589–1604, 2007.

[18] K. Kastella, "Event-averaged maximum likelihood estimation and mean-field theory in multitarget tracking," *Automatic Control, IEEE Transactions on*, vol. 40, no. 6, pp. 1070–1074, 1995.

[19] D. Henke, C. Magnard, M. Frioud, D. Small, E. Meier, and M. Schaepman, "Moving-target tracking in single-channel wide-beam sar," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 50, no. 11, pp. 4735–4747, 2012.

[20] I. Szottka and M. Butenuth, "Advanced particle filtering for airborne vehicle tracking in urban areas," *Geoscience and Remote Sensing Letters, IEEE*, vol. PP, no. 99, pp. 1–5, 2013.

[21] G. Gennari and G. Hager, "Probabilistic data association methods in visual tracking of groups," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, 2004, pp. II–876–II–881 Vol.2.

[22] A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu, "Multi-object tracking through simultaneous long occlusions and split-merge conditions," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, june 2006, pp. 666 – 673.

[23] V. Mathon and H. Laurent, "Life cycle of sahelian mesoscale convective cloud systems," *Quarterly Journal of the Royal Meteorological Society*, vol. 127, no. 572, pp. 377–406, 2001. [Online]. Available: http://dx.doi.org/10.1002/qj.49712757208

[24] D. A. Vila, L. A. T. Machado, H. Laurent, and I. Velasco, "Forecast and tracking the evolution of cloud clusters (fortracc) using satellite infrared imagery: Methodology and validation," *Weather and Forecasting*, vol. 23, no. 2, pp. 233–245, Apr 2008. [Online]. Available: http://dx.doi.org/10.1175/2007WAF2006121.1

[25] T. Fiolleau and R. Roca, "An algorithm for the detection and tracking of tropical mesoscale convective systems using infrared images from geostationary satellite," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 51, no. 7, pp. 4302–4315, 2013.

[26] D. Mukherjee and S. Acton, "Cloud tracking by scale space classification," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 40, no. 2, pp. 405–415, 2002.

[27] C. Papin, P. Bouthemy, E. Mémin, and G. Rochard, "Tracking and characterization of highly deformable cloud structures," in *Computer Vision ECCV 2000*, ser. Lecture Notes in Computer Science, D. Vernon, Ed. Springer Berlin Heidelberg, 2000, vol. 1843, pp. 428–442. [Online]. Available: http://dx.doi.org/10.1007/3-540-45053-X_28

[28] J. Ma, A. Antoniadis, and F.-X. Le Dimet, "Curvelet-based snake for multiscale detection and tracking of geophysical fluids," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 44, no. 12, pp. 3626–3638, 2006.

[29] C. Storlie, T. Lee, J. Hannig, and D. Nychka, "Tracking of multiple merging and splitting targets: A statistical perspective," *Statistica Sinica*, vol. 19, no. 1, pp. 1–52, 2009. [Online]. Available: http://nldr.library.ucar.edu/repository/collections/OSGC-000-000-000-072

[30] P. Addesso, R. Conte, M. Longo, R. Restaino, and G. Vivone, "Map-mrf cloud detection based on phd filtering," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. 5, no. 3, pp. 919–929, 2012.

[31] B. Root, T.-Y. Yu, and M. Yeary, "Consistent clustering of radar reflectivities using strong point analysis: A prelude to storm tracking," *Geoscience and Remote Sensing Letters, IEEE*, vol. 8, no. 2, pp. 273–277, 2011.

[32] T. Long, L. Zheng, X. Chen, Y. Li, and T. Zeng, "Improved probabilistic multi-hypothesis tracker for multiple target tracking with switching attribute states," *Signal Processing, IEEE Transactions on*, vol. 59, no. 12, pp. 5721–5733, 2011.

[33] I. Cox and S. Hingorani, "An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 2, pp. 138 –150, feb 1996.

[34] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[35] R. E. Neapolitan, *Learning Bayesian Networks*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2003.

[36] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *Signal Processing, IEEE Transactions on*, vol. 56, no. 8, pp. 3447–3457, Aug.

[37] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," in *Advances in Information Retrieval*. Springer, 2005, pp. 345–359.