A TRACK-BEFORE-DETECT ALGORITHM FOR THE DETECTION OF A MARKOV TARGET IN THE PRESENCE OF MISSING OBSERVATIONS

Emanuele Grossi, Marco Lops, and Luca Venturino

e.grossi@unicas.it, lops@unicas.it, l.venturino@unicas.it
DIEI – Università degli Studi di Cassino e del Lazio Meridionale, via G. di Biasio 43, Italy

ABSTRACT

In this paper we address the problem of multi-frame detection (MFD) of a Markov target observed through noisy measurements. To limit the system complexity, we gate the detector with a pre-processing stage which discards unreliable observations in each frame. A novel dynamic programming algorithm is introduced, which applies a track-before-detect (TBD) logic to declare the presence of the target and jointly estimate its position. The closed-form complexity analysis and the numerical examples show that advantageous complexity-performance tradeoffs can be obtained with respect to the single-frame detector and with respect to other strategies already present in the literature.

***Index Terms*—** Multi-frame detection (MFD), track-before-detect (TBD), dynamic programming, Markov targets.

1. INTRODUCTION

Multi-frame detection (MFD) has been proposed to detect weak signals from noisy measurements by integrating their contribution over multiple consecutive frames. In the presence of target motion (or, more generally, of a dynamical system), MFD requires track-before-detect (TBD) techniques to correctly integrate the target signal along its unknown trajectory. In [1–3], such integration take place in the sensor measurement space, so as to obtain reliable detections to be possibly forwarded to a tracking stage. In [4–6], instead, the integration take place in the target state space, and detection and tracking are fully merged.[1]

MFD can be hardly implementable in the presence of agile targets and sensors with a *large* number of resolution elements (e.g., long-range, surveillance radars), even resorting to dynamic programming algorithms, such as the Viterbi algorithm [8]. To overcome such a complexity limitation we consider here the adoption of a pre-processing stage to limit the number of measurements on the basis of the their likelihood function. The main contribution is the derivation of a novel dynamic-programming TBD algorithm to efficiently perform MFD on the reduced-size set of observations, whose cardinality is now random and time-varying. A closed-form complexity analysis is given, and a performance assessment is provided, including comparisons with the single-frame detector, the MFD with un-processed data, and the two-stage detector analyzed in [9, 10].

The reminder of the paper is organized as follows. In Sec. 2, problem formulation is given, while in Sec. 3 the test statistics are derived. The proposed TBD algorithm is presented in Sec. 4, and its performance is analyzed in Sec. 5. Finally, concluding remarks are given in Sec. 6.

1.1. Relation to prior works

TBD was originally proposed in [4, 11–13] as a means to improve detection of weak moving objects, and has been successfully applied to both passive [5, 6, 14, 15] and active sensors [1, 16–23]. The two-stage decision approach considered here was envisioned in [18, 19, 24]; however, these previous works did not account for the fact that the number of observations surviving the first stage can be much smaller than the number of resolution elements. Finally, our work shares many aspects with [9, 10], but the TBD procedure described there does not take into account misses at the current frame, so that detection performance is impaired.

2. PROBLEM FORMULATION

Consider a sensor with $M \times N$ resolution elements (or pixels), where measurements are recorded at discrete time instants $n \in \mathbb{Z}$. At most one target is present in the scene, and only one pixel at each epoch n contain the signal from the prospective target. The signal from pixel (i, j) at time n is denoted $z_n(i, j)$, and it is a random variable with probability density f_1, if the target is present in (i, j), and f_0, otherwise. The process $z_n(i, j)$ is independent over the space (i.e., over i, j) and, conditioned on the absence or on the presence of a target in a specified pixel sequence, over the time (i.e., over n).

This works was carried out under a research contract sponsored by Selex Galileo, Nerviano (MI), Italy.

[1]As pointed out in [7], TBD should be more properly defined as track-before-declare in this case, since target is tracked before declaring it to be a valid target. The term "detect," instead, may generate confusion, in that it could be referred to target detections (i.e., measurements).

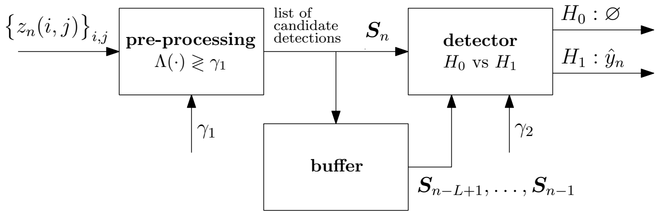

Fig. 1. The detector declares the presence of a target and jointly outputs its pixel location based on the lists of candidate detections $\{\boldsymbol{S}_\ell\}_{\ell=n-L+1}^{n}$.

The detection scheme for this measurement model is reported in Fig. 1. A pre-processing stage receives the measurements $\{z_n(i,j)\}_{i,j}$ and, at each epoch n, discards all the observations that, with high probability, do not come from a target. The merit function adopted at this step is the log-likelihood ratio (LLR) of f_1 to f_0, i.e., $\Lambda = \ln(f_1/f_0)$, and the pruning rule is

$$\begin{cases} \Lambda\big(z_n(i,j)\big) > \gamma_1 & \Rightarrow \text{ keep } z_n(i,j) \\ \Lambda\big(z_n(i,j)\big) \leq \gamma_1 & \Rightarrow \text{ discard } z_n(i,j) \end{cases}$$

The surviving measurements at epoch n, whose number is denoted $D_n \in \{0, 1, \ldots, MN\}$, are organized in the matrix

$$\boldsymbol{S}_n = \big(\boldsymbol{s}_{1,n}^T \cdots \boldsymbol{s}_{D_n,n}^T\big)^T$$

for $D_n \neq 0$, where $(\cdot)^T$ denotes transpose, and $\boldsymbol{s}_{k,n} = (x_{k,n} \;\; \zeta_{k,n})$ is a 3-dimensional, row-vector containing the pixel location $x_{k,n} \in \{1, \ldots, M\} \times \{1, \ldots, N\}$ and the corresponding measurement $\zeta_{k,n} = z_n(x_{k,n})$ at epoch n.

The detector jointly elaborates the current data-frame $\boldsymbol{S}_n$ and the past $L-1$ frames $\{\boldsymbol{S}_\ell\}_{\ell=n-L+1}^{n-1}$, and decides if the target is present (hypothesis H_1) or not (hypothesis H_0) at epoch n. If H_1 is declared, then an estimate of the target pixel location, say $\hat{y}_n$, is also given. Observe that this detection scheme is general enough to subsume both single-frame detection (when $L = 1$) and multi-frame detection with raw input data (when $\gamma_1 = -\infty$ and $L \geq 2$).

3. JOINT DETECTION AND ESTIMATION

Without loss of generality, assume that $n = L$, i.e., $\boldsymbol{S}_L$ is the current data frame. Let y_ℓ be the (unknown) pixel occupied by the target in the ℓ-th frame under hypothesis H_1. The sequence of pixels $\boldsymbol{y} = (y_1 \cdots y_L)$ defines the target trajectory over the L processed time intervals, and the detection problem can be solved by resorting to a generalized likelihood ratio test (GLRT): the LLR conditioned to $\boldsymbol{y}$ is maximized over the set of target trajectories, say $\mathcal{R}$, and is compared with a secondary threshold, say γ_2. It can be shown that the GLRT is

$$\max_{\boldsymbol{y}\in\mathcal{R}} \sum_{\ell=1}^{L} \Big(\Lambda\big(z_\ell(y_\ell)\big) - \kappa\Big) \mathbb{1}_{\{\Lambda(z_\ell(y_\ell)) > \gamma_1\}} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma_2 \qquad (1)$$

where $\mathbb{1}_A$ is the indicator function of the event A, and $\kappa = \ln \frac{\beta}{1-\alpha}$, with $\beta = \int_{\mathbb{R}} \mathbb{1}_{\{\Lambda(z) \leq \gamma_1\}} f_1(z)dz$ and $\alpha = \int_{\mathbb{R}} \mathbb{1}_{\{\Lambda(z) > \gamma_1\}} f_0(z)dz$ being the local (i.e., pixel-wise) probability of miss and false alarm, respectively. If a target is declared to be present, then the position estimate at epoch L can be obtained from the maximum-likelihood estimate of the trajectory

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}\in\mathcal{R}} \sum_{\ell=1}^{L} \Big(\Lambda\big(z_\ell(y_\ell)\big) - \kappa\Big) \mathbb{1}_{\{\Lambda(z_\ell(y_\ell)) > \gamma_1\}}.$$

For simplicity, here we take the last pixel index of $\hat{\boldsymbol{y}}$ that corresponds to a surviving measurement in $\{\boldsymbol{S}_\ell\}_{\ell=1}^{L}$.[2]

In this work, we consider a first-order Markov model for the target evolution with a constraint on the maximum target speed, say v_{max} pixels per frame, along each direction, whereby set of target trajectories takes the form

$$\mathcal{R} = \Big\{\boldsymbol{y} \in \big(\{1, \ldots, M\} \times \{1, \ldots, N\}\big)^L : \|y_\ell - y_{\ell-1}\| \leq v_{\text{max}}, \ell = 2, \ldots, L\Big\} \qquad (2)$$

$\|\cdot\|$ denoting a preassigned vector norm. Notice that, differently from [9, 10], trajectories with a miss at the last frame L are considered admissible.

4. PROPOSED TBD ALGORITHM

As shown in [3–6, 19], the maximization (1) can be carried out avoiding an exhaustive search through the Viterbi algorithm [8]. However, even the complexity of the Viterbi algorithm can be unaffordable in many applications. E.g., if $M = N = 10^3$, and that the target is allowed a transition of ± 20 pixels in each dimension, the Viterbi algorithm has to compute a maximum between 1681 elements (41×41 cells along the two dimensions), for each of the $MN = 10^6$ resolution elements of each frame. The algorithm we propose here takes advantage of the fact that the pre-processing stage reduces the number of non-zero data measurements in each frame, and efficiently implement the GLRT in (1). It is a modification of the procedure introduced in [9,10] (and in [25,26] in a radar framework), where trajectories with a miss at the last frame could not be handled.

Observe that the summation in the GLRT in (1) only depends on the observations surviving the pre-processing stage, and, therefore, the maximization can be carried out on a subset of $\mathcal{R}$. Every element of this subset can be specified by a L-dimensional vector, say $\boldsymbol{t} = (t_1 \cdots t_L)$, with $t_\ell \in \{0, 1, \ldots, D_\ell\}$ for $\ell = 1, \ldots, L$: specifically, $t_\ell = m$

[2] If an accurate dynamic model of the target evolution is available, a regression (or any other curve-fitting method) can be used to obtain a better estimate of the target trajectory and, therefore, of its final position (especially in the case of a missing observation in the last frame).

means that the target is observed at epoch ℓ, and the corresponding detection is $\boldsymbol{s}_{m,\ell}$, while $t_\ell = 0$ that there is a missing measurement at epoch ℓ. The sequence of positions indexed by $\boldsymbol{t}$ is $x_{t_1,1}, \ldots, x_{t_L,L}$, with $x_{t_\ell,\ell}$ not defined if $t_\ell = 0$ (i.e., in case of a missing observation). The corresponding sequence of statistics entering the GLRT in (1) is $\lambda_{t_1,1}, \ldots, \lambda_{t_L,L}$, where

$$\lambda_{k,\ell} = \big(\Lambda(\zeta_{k,\ell}) - \kappa\big)\mathbb{1}_{\{k\neq 0\}}.$$

With these definitions, the GLRT in (1) can be now recast as

$$\max_{\boldsymbol{t}\in\mathcal{R}'} \sum_{\ell=1}^{L} \lambda_{t_\ell,\ell} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma_2 \tag{3}$$

where

$$\mathcal{R}' = \Big\{\boldsymbol{t} : \|x_{t_p,p} - x_{t_q,q}\| \leq v_{\max}|p-q|, \\ \forall\, p,q \in \{1,\ldots,L\} \text{ such that } t_p, t_q \neq 0\Big\}. \tag{4}$$

It is not difficult to see that, upon defining

$$F_{k,\ell} = \max_{\substack{\boldsymbol{t}\in\mathcal{R}':\ t_\ell=k,\\ \text{and } t_p=0\ \forall p>\ell}} \sum_{p=1}^{L} \lambda_{t_p,p}$$

$$T_{k,\ell} = \arg\max_{\substack{\boldsymbol{t}\in\mathcal{R}':\ t_\ell=k,\\ \text{and } t_p=0\ \forall p>\ell}} \sum_{p=1}^{L} \lambda_{t_p,p}$$

the test in (3)—and therefore the GLRT in (1)—can be rewritten also as

$$\max_{\substack{(k,\ell):\ell\in\{1,\ldots,L\}\\ \text{and } k\in\{1,\ldots,D_\ell\}}} F_{k,\ell} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma_2.$$

As to position estimation,

$$(\hat{k},\hat{\ell}) = \arg\max_{\substack{(k,\ell):\ell\in\{1,\ldots,L\}\\ \text{and } k\in\{1,\ldots,D_\ell\}}} F_{k,\ell}$$

is an estimator of the pixel index of the last observation of the target, so that we take

$$\hat{y}_L = x_{(T_{\hat{k},\hat{\ell}})_{\hat{\ell}},\hat{\ell}}$$

where $(T_{\hat{k},\hat{\ell}})_{\hat{\ell}}$ is the $\hat{\ell}$-th entry of the vector $T_{\hat{k},\hat{\ell}}$.

A dynamic programming algorithm to compute the statistics $\{F_{k,\ell}, T_{k,\ell}\}$, for $\ell = 1,\ldots,L$, and $k = 1,\ldots,D_\ell$, is presented next.

4.1. Track formation algorithm

Let $\mathcal{M}_{k,\ell}$ be the set of indexes addressing all past alarms compatible with the current alarm $\mathcal{S}_{k,\ell}$, i.e.,

$$\mathcal{M}_{k,\ell} = \Big\{(j,p) : p \in \{1,\ldots,\ell-1\}, j \in \{1,\ldots,D_p\}, \\ \text{and } \|x_{k,\ell} - x_{j,p}\| \leq v_{\max}(\ell-p)\Big\} \tag{5}$$

and let $(\,\cdot\,)_{1:m}$ denote the sub-vector composed of the first m entries of the vector. Then, we give the following

Algorithm 1 (Computes $\{F_{k,\ell}, T_{k,\ell}\}$, for $\ell = 1,\ldots,L$ and $k = 1,\ldots,D_\ell\}$)**.**

1. Initialization: $\ell = 1$

For $k = 1,\ldots,D_1$

$F_{k,1} = \lambda_{k,1}$

$T_{k,1} = \big(k\ 0\ \cdots\ 0\big)$

2. Recursion: $\ell = 2,\ldots,L$

For $k = 1,\ldots,D_\ell$

$$F_{k,\ell} = \begin{cases} \lambda_{k,\ell} + \max\limits_{(j,p)\in\mathcal{M}_{k,\ell}} F_{j,p}, & \textit{if } \mathcal{M}_{k,\ell} \neq \varnothing \\ \lambda_{k,\ell}, & \textit{otherwise} \end{cases}$$

$$T_{k,\ell} = \begin{cases} \big((T_{h,m})_{1:m}\ 0\ \cdots\ 0\ \underset{\ell\text{-th entry}}{k}\ 0\ \cdots\ 0\big), \\ \qquad \textit{if } \mathcal{M}_{k,\ell} \neq \varnothing, \textit{ where} \\ \qquad (h,m) = \arg\max\limits_{(j,p)\in\mathcal{M}_{k,\ell}} F_{j,p} \\ \big(0\ \cdots\ 0\ \underset{\ell\text{-th entry}}{k}\ 0\ \cdots\ 0\big), \quad \textit{otherwise} \end{cases}$$

4.2. Complexity analysis

The complexity of Algorithm 1 is ruled by the innermost loop, which requires to check the constraints between $\boldsymbol{s}_{k,\ell}$ and $\boldsymbol{s}_{j,p}$, for $j = 1,\ldots,D_p$, $p = 1,\ldots,\ell-1$, to obtain the set $\mathcal{M}_{k,\ell}$. Since $\ell = 2,\ldots,L$ and $k = 1,\ldots,D_\ell$, the number of operation needed by Algorithm 1 is in the order of $\sum_{\ell=2}^{L} D_\ell \sum_{p=1}^{\ell-1} D_p$. Notice now that $\{D_\ell\}_{\ell=1}^{L}$ is a sequence of i.i.d. random variables. Specifically, each D_ℓ is the sum of two independent Binomial random variables, and its mean is $(MN-K)\alpha + K(1-\beta)$, where $K = 1$ under H_1, and $K = 0$ under H_0. Therefore, the average number of operations of Algorithm 1 is in the order of $L(L-1)[(MN-K)\alpha + K(1-\beta)]^2$, i.e., quadratic in the number of integrated frames and in the average number of candidate detections per frame, and the average computational complexity is $\mathcal{O}\big((LMN\alpha)^2\big)$.

Recall now that the complexity of the Viterbi-based routine is $\mathcal{O}(LMNv_{\max}^2)$.[3] Therefore, if $\alpha = o\big(v_{\max}/\sqrt{LMN}\big)$, then the average complexity of Algorithm 1 is smaller that that of the Viterbi algorithm. Observe, finally, that if γ_1 is chosen so that $\alpha = \mathcal{O}\big(1/(L\sqrt{MN})\big)$, then the number of operations required by Algorithm 1 is in the same order as that required by the first stage to evaluate the MN LLR's, so that the complexity of the proposed detection architecture is equal to that of the single-frame detector, i.e., $\mathcal{O}(MN)$.

5. NUMERICAL RESULTS

We consider a sensor with $N = M = 100$ and a target evolving according to a random walk with independent tran-

[3]In the recursive step of the Viterbi algorithm a maximization on the admissible afferent states is to be performed for each of the MN resolution elements. Since the cardinality of this set is in the order of $v_{\max}^2$ (which may be dependent of M and N), the computational complexity is $\mathcal{O}(LMNv_{\max}^2)$.

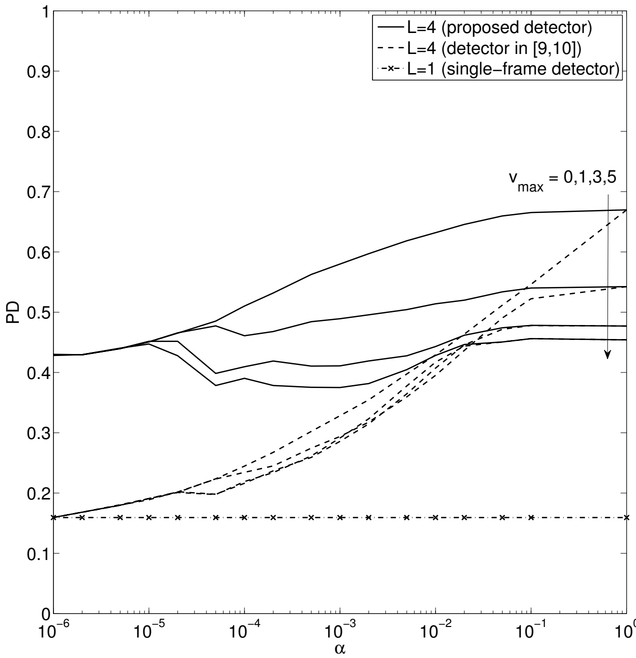

Fig. 2. PD vs α (in logarithmic scale) for different values $v_{\max}$ (in pixels per frame), PFA $= 10^{-2}$, and SNR $= 8$ dB.

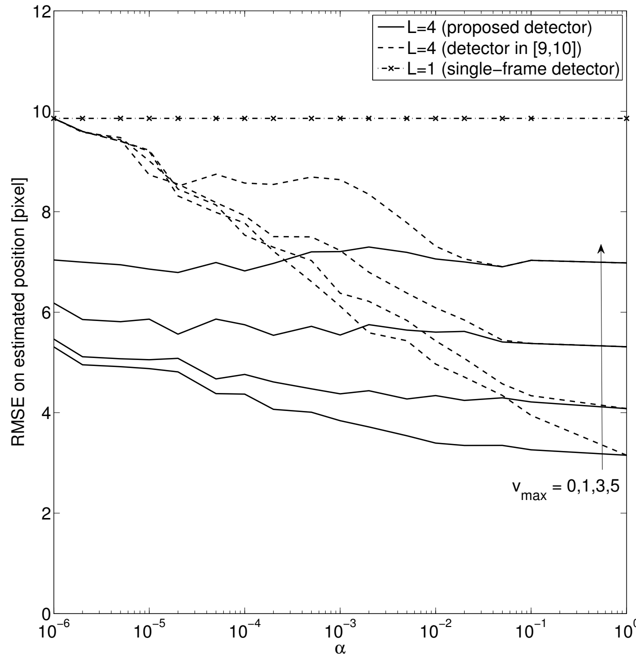

Fig. 3. RMSE on the estimated position vs α (in logarithmic scale) for different values $v_{\max}$ (in pixels per frame), PFA $= 10^{-2}$, and SNR $= 8$ dB.

sitions along the two directions, so that the infinity norm is used in (2), (4), and (5). The random variables $z_n(i, j)$ are exponentially distributed with $f_0(z) = e^{-z}\mathbb{1}_{[0,\infty)}(z)$ and $f_1(z) = (1 + \text{SNR})^{-1}e^{-z/(1+\text{SNR})}\mathbb{1}_{[0,\infty)}(z)$, where SNR is the signal-to-noise ratio. The memory of the detector is $L = 4$, and the performance of the test is assessed in terms of probability of false alarm (PFA), i.e., accept H_1 under H_0, probability of correct detection (PD), i.e., accept H_1 under H_1, and root mean square error (RMSE) on the estimated target position. For the sake of comparison, we report the cases of standard single-frame detection ($L = 1$), MFD with raw input-data ($L \geq 1$ and $\alpha = 1$), and the MFD procedure considered in [9, 10]. The plots have been obtained using the Monte Carlo method with $6 \cdot 10^5$ independent realizations of batches of L frames.

Fig. 2 shows PD versus α for PFA $= 10^{-2}$, $L = 4$, and different values of $v_{\max}$, while Fig. 3 reports the corresponding RMSE on the position estimate. A large gain with respect to the single-frame detector is observed for all the inspected velocities in terms of both PD and RMSE on the position estimate. Interestingly, the dependence of PD on α is not monotonic when $v_{\max} > 0$, since two conflicting effects are playing. On the one hand, large values of α allow to take more observations of the target during the observation window; on the other, small values of α guarantee that most of the false alarms in the data set have been removed, so that the cardinality of the search set $\mathcal{R}'$ is reduced. When $v_{\max} = 0$, instead, $\mathcal{R}'$ has the smallest cardinality, equal to MN irrespective of α: in this case MFD (which reduces to pixel-wise incoherent integration) achieves its best performance, which degrades as α is decreased. Finally, notice that the TBD strategies proposed here is equivalent to that in [9, 10] for raw input data ($\alpha = 1$), but it is superior for censored data in terms of both PD and RMSE, and the gain increases as α is decreased: this due to the fact that the procedure in [9, 10] does not handle a missed observation of the target at the end of the processing window, so that its performance converges to that of the single-frame detector when $\alpha \to 0$.

6. CONCLUSION

A novel procedure for MFD of a Markov target has been proposed here. Its core is a dynamic programming algorithm to perform TBD on a set of pre-processed observations. The complexity analysis showed that proper selection of the threshold in the pre-processing stage permits to have the same computational complexity as the single-frame detector, while simulations demonstrated that large gains with respect to the single-frame detector and to the MFD procedure of [9,10] can be obtained.

7. REFERENCES

[1] S. W. Shaw and J. F. Arnold, "Design and implementation of a fully automated OTH radar tracking system," in *IEEE Proc. Int. Radar Conf.*, Alexandria, VA, USA, May 1995, pp. 294–298.

[2] P. Wei, J. Zeidler, and W. Ku, "Analysis of multiframe target detection using pixel statistics," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 31, no. 1, pp. 238–247, Jan. 1995.

[3] H. Im and T. Kim, "Optimization of multiframe target detection schemes," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 35, no. 1, pp. 176–186, Jan. 1999.

[4] Y. Barniv, "Dynamic programming solution for detecting dim moving targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 21, no. 1, pp. 144–156, Jan. 1985.

[5] J. F. Arnold, S. W. Shaw, and H. Pasternack, "Efficient target tracking using dynamic programming," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 29, no. 1, pp. 44–56, Jan. 1993.

[6] S. M. Tonissen and R. J. Evans, "Peformance of dynamic programming techniques for track-before-detect," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 32, no. 4, pp. 1440–1451, Oct. 1996.

[7] W. R. Blanding, P. K. Willett, Y. Bar-Shalom, and R. S. Lynch, "Directed subspace search ML-PDA with application to active sonar tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 1, pp. 201–216, Jan. 2008.

[8] G. D. J. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 29, no. 3, pp. 268–277, Mar. 1973.

[9] E. Grossi, M. Lops, and L. Venturino, "Track-before-detect with censored observations," in *IEEE Int. Conf. Acoustic, Speech, Signal Process. (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 3941–3944.

[10] E. Grossi, M. Lops, and L. Venturino, "A track-before-detect procedure for sparse data," in *IEEE Stat. Signal Process. Workshop (SSP)*, Ann Arbor, MI, USA, Aug. 2012, pp. 772–775.

[11] N. C. Mohanty, "Computer tracking of moving point targets in space," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 3, no. 5, pp. 606–611, Sep. 1981.

[12] I. S. Reed, R. M. Gagliardi, and H. M. Shao, "Application of three-dimensional filtering to moving target detection," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 19, no. 9, pp. 898–905, Nov. 1983.

[13] Y. Barniv and O. Kella, "Dynamic programming solution for detecting dim moving targets. Part II: Analysis," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 23, no. 6, pp. 776–788, Nov. 1987.

[14] I. Reed, R. Gagliardi, and L. Stotts, "A recursive moving-target-indication algorithm for optical image sequences," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 26, no. 3, pp. 434–440, May 1990.

[15] T. Kirubarajan and Y. Bar-Shalom, "Low observable target motion analysis using amplitude information," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 32, no. 10, pp. 1367–1384, Oct. 1996.

[16] J. D. R. J. Kramer and W. S. Reid, "Track-before-detect processing for an airborne type radar," in *Proc. IEEE Int. Radar Conf.*, Arlington, VA, USA, May 1990, pp. 422–427.

[17] J. L. Harmon, "Track-before-detect performance for a high prf search mode," in *Rec. IEEE Nat. Radar Conf.*, Los Angeles, CA, USA, Mar. 1991, pp. 11–15.

[18] B. D. Carlson, E. D. Evans, and S. L. Wilson, "Search radar detection and track with the Hough transform. Part I: system concept," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 30, no. 1, pp. 102–108, Jan. 1994.

[19] S. Buzzi, M. Lops, and L. Venturino, "Track-before-detect procedures for early detection of moving target from airborne radars," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 3, pp. 937–954, Jul. 2005.

[20] D. Orlando, L. Venturino, M. Lops, and G. Ricci, "Track-before-detect strategies for STAP radars," *IEEE Trans. Signal Process.*, vol. 58, no. 2, pp. 933–938, Feb. 2010.

[21] E. Grossi and M. Lops, "Sequential along-track integration for early detection of moving targets," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3969–3982, Aug. 2008.

[22] E. Grossi and M. Lops, "Joint sequential detection and estimation of markov targets," in *IEEE Inform. Theory Workshop*, Porto, Portugal, May 2008, pp. 308–312.

[23] S. Buzzi, M. Lops, L. Venturino, and M. Ferri, "Track-before-detect procedures in a multi-target environment," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 3, pp. 1135–1150, Jul. 2008.

[24] P. Uruski and M. Sankowski, "On estimation of performance of track-before-detect algorithm for 3D stacked-beam radar," in *Proc. Int. Conf. Microwaves, Radar, Wireless Commun. (MIKON)*, Warsaw, Poland, May 2004, pp. 97–100.

[25] E. Grossi, M. Lops, and L. Venturino, "A two-step multi-frame detection procedure for radar systems," in *Proc. Int. Conf. Inform. Fusion (FUSION)*, Singapore, Jul. 2012, pp. 1196–1201.

[26] E. Grossi, M. Lops, and L. Venturino, "A novel dynamic programming algorithm for track-before-detect in radar systems," *IEEE Trans. Signal Process.*, to be published.