POZNAN UNIVERSITY OF TECHNOLOGY ACADEMIC JOURNALS
No 76 Electrical Engineering 2012

Przemysław MAZUREK*

APPLICATION OF DOT PRODUCT FOR TRACK-BEFORE-DETECT TRACKING OF NOISE OBJECTS

The Track-Before-Detect (TBD) algorithms are applied for the tracking of signals below the noise floor. The noise object is the signal that has noise samples only. The processing of such signal using Spatio-Temporal TBD is not possible directly. The proposed preprocessing technique allows analysis of the signal using moving window approach and dot product calculations. Two vectors, related to the distributions, are compared: the overall signal and the local, related to the window position. The Monte Carlo tests are applied for the analysis of performance.

1. INTRODUCTION

There are numerous tracking algorithms [1-3]. Typical tracking algorithms are used for the tracking of well-separated objects. The difference between background noise and the object signal should be high. It is necessary for simplest tracking algorithms (SNR >> 1). The detection of such objects is simple using threshold based algorithms.

There are many real applications, where SNR is low, so many false detections occurs, and many detections of true object are missing. This is common for the space, air, water surface and underwater surveillances. Stealth techniques are applied for the reduction of SNR.

Tracking algorithms improve tracking by the application of the motion model. Missing measurements could be simulated using prediction abilities of the tracking filters. It means that low performance of the detection algorithm is suppressed by the tracking abilities.

The detection is not possible if signal is lost in noise floor (SNR<1). Threshold level can not be applied because giant amount of other false measurement occurs.

Conventional tracking systems are based on the detection and tracking approach. Alternative approach based on the track-before-detect (TBD), could be applied for the processing of such signals [4,12]. The main difference is the order of the operations (Fig. 1). The tracking is applied for all possible trajectories, even if no one object is in the range. Every trajectory is processed so signal values are filtered. Large values correspond to the object or objects signal. Low values

* West-Pomeranian University of Technology, Szczecin.

correspond to the trajectories that are not related to any object. Low values are around zero level, assuming zero mean value Gaussian noise. Positive values of the signal are accumulated and the detection is possible using threshold. Well fitted hypothetical trajectory to the real trajectory of the object gives noise suppression and the object detection.

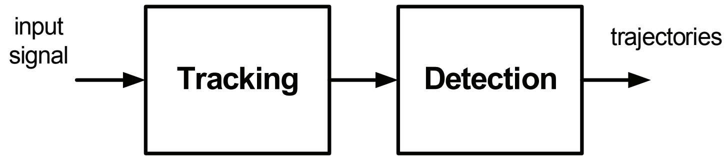

Fig. 1. TBD system schematic

The main problem of TBD is the computation cost, because computations do not depend on the number of tracked objects. The computation cost is fixed and huge, what is important for the real-time applications.

Almost all considered tracing scenarios considered by the other researchers are related to the additive signal and the Gaussian noise. Other types of the noise could be also processed.

2. NOISE OBJECTS

Very interesting case if the signal is the situation where the tracked signal is the noise itself. The difference between this signal noise and the background noise is related to the statistical parameters only. In such case the zero mean value could be measured for the background and the object also. Accumulation techniques, applied directly, cannot give proper results.

The technique for the preprocessing of signal using sliding window approach is proposed in [11]. The local values of the standard deviation are calculated, so TBD processed signal is non-negative. The following formula is applied for the signal preprocessing:

$$X(k,s) = std(M(k,s-L)...M(k,s+L)) \tag{1}$$

The detection of such objects that occupy a few sample points (pixels) is possible after tacking. The window size is *2L+1* for 1D data.

In [5] is proposed another approach based on the chi-square statistic. Two empirical distributions are compared using the following formula:

$$X(k,s) = \chi^2(k,s) = \sum_{i=1}^{N} \frac{(O(k,s)_i - G(k)_i)^2}{G(k)_i} \tag{2}$$

The reference distribution is obtained from all available samples *G(k)*. The observed distribution is a local one. The sliding window is applied for the processing of the current observations $O(k,s)$. Both distributions are discrete and defined by the set of subareas. The number of subareas and boundaries of subareas are applied a priori (N) to both distributions identically.

3. DOT PRODUCT BASED PREPROCESSING

The empirical and observed distributions are compared in chi-square formula. The nominator corresponds to the square error, and the application of the denominator value allows the computation of the relative square errors. All errors are summed together, and the chi-square value is obtained.

The application of the square error is arbitrary. It means that there are possible available other methods of comparison of two distributions. The square error is reduced to zero if both values are identical.

In [6] is proposed application of the different measurement space for improving the detection of signal. The computation of the dot product is an example technique. Two vectors could be compared using angle between them. The length of the vector corresponds to the magnitude of vector only. The relation between two vectors (x,y), described by the angle, is related to the orthogonality of them. Similar signals as shapes have lows angle value. The cosine of angle θ is:

$$\cos\theta = \frac{x \cdot y}{|x| \cdot |y|} \tag{3}$$

where the dot product is:

$$x \cdot y = \sum_{i=1}^{K} x_i y_i \tag{4}$$

and K is the length of vectors. The computation of the dot product and lengths of both vectors gives the cosine of angle. Both formulas are applied using sliding window for input measurements. One of the vectors is the part of the signal and the second is the reference shape known a priori.

The measurement of noise object needs modified idea, because computations of stochastic signals are observed only.

4. SPATIO-TEMPORAL TRACK-BEFORE-DETECT ALGORITHM AND DOT PRODUCT BASED PREPROCESSING

The Spatio-Temporal TBD algorithm is a kind of the multidimensional IIR filter [7-10]. There are two main formulas of this algorithm. The information update is responsible for new data input. The motion update is the predictor and the predictions are mixed with new data using information update. The output is the results of one of the formulas, depending on the choice. The smoothing coefficient it responsible for the balance between influence of the new data and predicted state.

Start

$P(k=0,s)=0$ // initialization (5a)

For $k \geq 1$

$P^{-}(k,s)=\int\limits_{S} q_k(s\,|\,s_{k-1})P(k-1,s_{k-1})ds_{k-1}$ // motion update (5b)

$P(k,s)=\alpha P^{-}(k,s)+(1-\alpha)X(k,s)$ // information update (5c)

EndFor

Stop

k – iteration number,
s – particular space,
q_k – Markov matrix,
X – preprocessed data,
P^{-} – predicted TBD output,
P – TBD output,
α – weight (smoothing coefficient), range: 0-1.

The input signal *M* is preprocessed like in the chi-square statistic. The discrete distribution from measurement is computed. The numbers of regions and regions boundaries are fixed and equal to the *N*.

There are two vectors: the global *G* and the local *O*, related to the position of the sliding window. Values of the vectors are the number of cases within particular region. Two discrete distributions are stored as vectors and compared using the formula (3).

The global vector is computed using the counting of measurement values for the specified region R_i:

$$G_i=\sum_{m=1}^{length(M)}\begin{Bmatrix}1 & : & R_i<M(m)\leq R_{i+1}\\ 0 & : & otherwise\end{Bmatrix} \tag{6}$$

where *i* denotes particular region.

The local vector is dependent on the position *s*:

$$O(s)_i=\sum_{m=s-L}^{s+L}\begin{Bmatrix}1 & : & R_i<M(m)\leq R_{i+1}\\ 0 & : & otherwise\end{Bmatrix} \tag{7}$$

where *2L+1* is the window length, but in this formula boundaries effects are omitted for simplification.

The input measurement *X* is computed using the following formulas:

$$X(s) = \frac{G \cdot O(s)}{|G| \cdot |O(s)|} \tag{8a}$$

$$|G| = \sqrt{\sum_{i=1}^{N} G_i^2} \tag{8b}$$

$$|O(s)| = \sqrt{\sum_{i=1}^{N} O(s)_i^2} \tag{8c}$$

The time, denoted by k in formulas (5) is omitted in formulas (6,7,8) for simplification.

The values of X are from <*-1, 1*> range. Assuming recognition of negative and positive signal as the same case, the range <*0, 1*> is obtained for modified formula (8a):

$$X(s) = \frac{|G \cdot O(s)|}{|G| \cdot |O(s)|} \tag{9}$$

Overall TBD system is depicted in Fig.2.

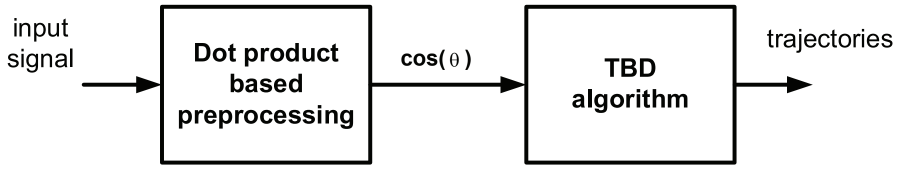

Fig. 2. Proposed TBD system schematic with dot product based preprocessing

The TBD algorithm needs specification of the default boundary value for state space computations. Zero value is assumed, so input X should be transformed to another range. The absolute value of X is sufficient for the signal processing. The following formula should be used:

$$X(s) = 1 - \frac{|G \cdot O(s)|}{|G| \cdot |O(s)|} \tag{10}$$

Zero value is obtained for the orthogonal distributions. Maximal value (1.0) is obtained for the equal distributions (zero angle between vectors).

5. RESULTS

This example is related to the Gaussian noise signals. The background noise is Gaussian and the signal of object is Gaussian also (standard deviation 1.0). An additive case is considered and the different statistical parameters are obtained in the place of the object. Regions are adaptively selected using the computation of the standard deviation of measurements. The following boundaries are specified:

$$\{-Inf,-2,-1,0,+1,+2,+Inf\}std(M) \tag{11}$$

The Monte Carlo test is applied for the performance analysis. Assumed object has width 7 samples. There are 1200 samples of signal in iteration. There are 11 of motion vectors (0-10) and the motion of the object is randomly selected. The mean distance error is computed between true and computed positions after 80 iterations, for 1000 cases (every case has randomly selected standard deviation of the background noise). The results are shown in Fig.3 for α=0.95.

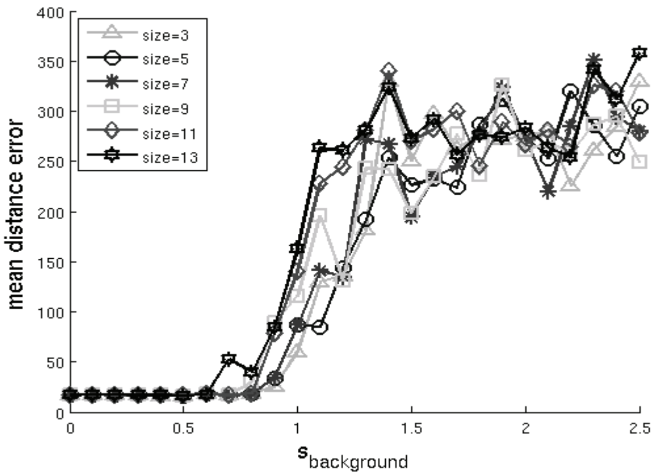

Fig. 3. Mean distance error – performance analysis of the proposed preprocessing

6. CONCLUSIONS

The proposed preprocessing of the measurements, based on the computation of the dot product, allows correct detection of the object. The maximal standard deviation of the background noise is about 0.8-0.9. The detection of the noise signal is important for the tracking applications. The Spatio-Temporal TBD allows tracking of the preprocessed signals, so zero mean value noises of the background and object are allowed. The correction of the results is possible by the application of the gate and analysis of more then single iteration.

ACKNOWLEDGEMENTS

This work is supported by the UE EFRR ZPORR project Z/2.32/I/1.3.1/267/05 "Szczecin University of Technology - Research and Education Center of Modern Multimedia Technologies" (Poland).

REFERENCES

[1] Bar-Shalom Y., Multitarget-Multisensor Tracking: Applications and Advances, vol. II, Artech House, 1992.

[2] Blackman S., Multiple-Target Tracking with Radar Applications. Artech House, 1986.

[3] Blackman S., Popoli R., Design and Analysis of Modern Tracking Systems, Artech House, 1999.

[4] Boers Y., Ehlers F., Koch W., Luginbuhl T., Stone L.D., Streit R.L., Track Before Detect Algorithm, EURASIP Journal on Advances in Signal Processing, 2008.

[5] Mazurek P., Chi-square statistic for noise objects tracking in Track-Before-Detect systems, Poznań University of Technology Academic Journals – Electrical Engineering, no. 71, 177-184, 2012.

[6] Mazurek P., Comparison of Different Measurement Spaces for Spatio–Temporal Recurrent Track–Before–Detect Algorithm, Advances in Intelligent and Soft Computing, vol. 102 - Image Processing and Communications Challenges 3, Springer Verlag, 157-164, 2011.

[7] Mazurek P., Hierarchical Track–Before–Detect Algorithm for Tracking of Amplitude Modulated Signals, Advances in Intelligent and Soft Computing, vol. 102 - Image Processing and Communications Challenges 3, Springer Verlag, 511-518, 2011.

[8] Mazurek P., Optimization of bayesian Track-Before-Detect algorithms for GPGPUs implementations, Electrical Review, R. 86 no. 7/2010, 187-189, 2010.

[9] Mazurek P., Optimization of Track-Before-Detect systems for GPGPU, Measurement Automation and Monitoring, vol. 56 no. 7, 665-667, 2010.

[10] Mazurek P., Optimization of Track-Before-Detect Systems with Decimation for GPGPU, Measurement Automation and Monitoring, vol. 56 no. 12, 1523-1525, 2010.

[11] Mazurek P., Track-Before-Detect Algorithm for Noise Objects, Measurement Automation and Monitoring, vol. 56 no. 10, 1183-1185, 2010.

[12] Stone L. D., Barlow C. A., Corwin T. L. Bayesian Multiple Target Tracking. Artech House, 1999.