Performance of Dot-product preprocessing for Track-Before-Detect tracking of noise objects

Przemysław Mazurek
West Pomeranian University of Technology
71-126 Szczecin, ul. 26. Kwietnia 10, e-mail: przemyslaw.mazurek@zut.edu.pl

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 preprocessing technique based on the window approach and dot-product calculations emphasis the differences between global and local empirical distributions. The Monte Carlo tests are applied for the analysis of performance for two smoothing coefficients, different width of the window of analysis and different size of the object.

KEYWORDS: Tracking, Estimation, Track-Before-Detect, Dot-Product

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.

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, 13]. The main difference is the order of the operations (Fig. 1).

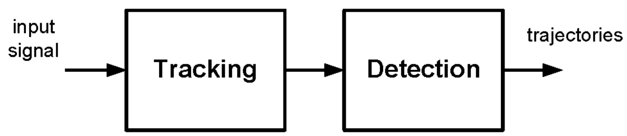

Fig. 1. TBD system schematic

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 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.

The signal could be noise itself, what is very interesting case. 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.

2. Noise objects

Preprocessing of the measurement allows the processing of the noise object signal in the proper way. The preprocessing of the signal to the positive signal values is necessary.

The first method [12] applied for such object is the technique based on the sliding window approach, where $2L+1$ is the length of the window and computation of standard deviation:

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

In [6] 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.

In [5] is proposed approach based on dot-product comparison [7] of two empirical distributions. 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 angle between vectors.

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

$$G_i = \sum_{m=1}^{length(M)} \begin{Bmatrix} 1 & : & R_i < M(m) \le R_{i+1} \\ 0 & : & otherwise \end{Bmatrix} \tag{3}$$

where i denotes particular region. The local vector is dependent on the position s:

$$O(s)_i = \sum_{m=s-L}^{s+L} \begin{Bmatrix} 1 & : & R_i < M(m) \le R_{i+1} \\ 0 & : & otherwise \end{Bmatrix} \tag{4}$$

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:

$$X(s) = \frac{G \cdot O(s)}{|G| \cdot |O(s)|} \tag{5a}$$

$$|G| = \sqrt{\sum_{i=1}^{N} G_i^2} \tag{5b}$$

$$|O(s)| = \sqrt{\sum_{i=1}^{N} O(s)_i^2} \tag{5c}$$

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 (5a):

$$X(s) = \frac{|G \cdot O(s)|}{|G| \cdot |O(s)|} \tag{6}$$

Obtained result of the preprocessing is delivered to Track-Before-Detect algorithm, like ST TBD presented briefly in next section.

3. Spatio-temporal Track-Before-Detect algorithm

The Spatio-Temporal TBD algorithm is a kind of the multidimensional IIR filter [8-11]. The information update formula (7c) is responsible for new data input. The motion update formula (7b) 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 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 (7a)

For $k \geq 1$

$P^-(k,s) = \int_S q_k(s \mid s_{k-1}) P(k-1, s_{k-1}) ds_{k-1}$ // motion update (7b)

$P(k,s) = \alpha P^-(k,s) + (1-\alpha) X(k,s)$ // information update (7c)

EndFor

Stop

where: 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 discrete distribution from measurement M is computed. The numbers of regions and regions boundaries are fixed and equal to the N.

4. Performance evaluation of Dot-product preprocessing

The following test uses Monte Carlo approach for evaluation of the detection and tracking properties of algorithm. The background noise is Gaussian with variable value of standard deviation. The noise object has fixed standard deviation (1.0). An additive case is considered. The length of the moving window size is another parameter. The tests evaluated windows length (size) that could be smaller, adequate or larger in comparison to the length (size) of the object. It is necessary, because the length of the object is usually unknown parameter. In specific cases the length could be known a priori and applied for the better estimation. The set of window sizes is: 3, 5, 7, 9, 11, 13.

Two smoothing coefficients are tested: 0.95 and 0.98. The higher value of the smoothing coefficient allows better filtering of noise but the response time is longer. Switching between trajectories is highly delayed for higher smoothing coefficient.

The additive case for two Gaussian signals is interesting because the shape of the distribution is identical for the background and the object. The detected difference is related to the boundary area of the object. The position of the moving window contains two distributions. The first distribution is the background noise distribution and the second distribution is the additive distribution (background + object). The Gaussian mixture is obtained for such position. Two cases are possible for two boundaries of the object (Fig. 2).

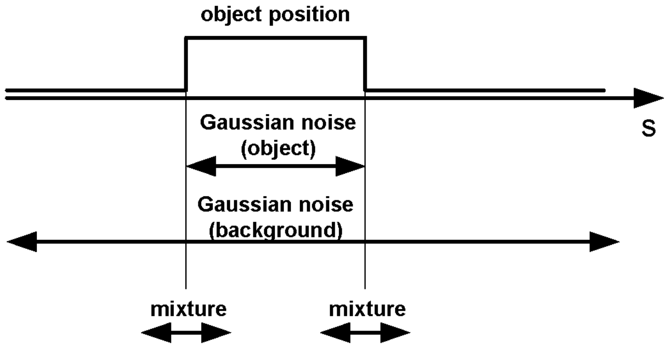

Fig. 2. Gaussian mixtures emphasized by the dot-product preprocessing

Regions are adaptively selected using the computation of the standard deviation of measurements. The following boundaries are specified:

$$\{-Inf,-2,-1,0,+1,+2,+Inf\}std(M) \tag{8}$$

Monte Carlo test is applied for the performance analysis (Fig. 3-8). Assumed object has width 3, 5 or 7 samples. There are 1200 samples of signal in every 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). Single curve for specified smoothing coefficient and window size is obtained. The increased number of test cases influences the smoothness of the curve and 1000 or more cases are desired.

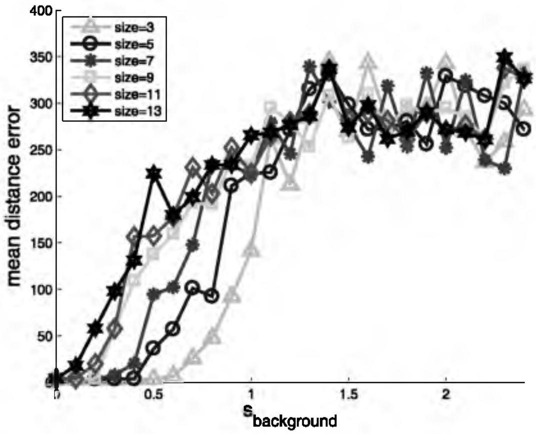

Fig. 3. Results for object size = 3 (smoothing coefficient 0.95)

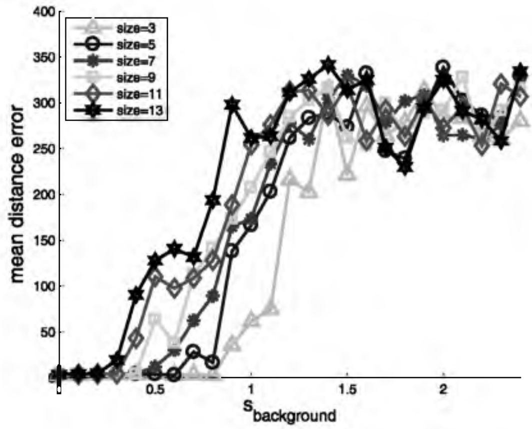

Fig. 4. Results for object size = 3 (smoothing coefficient 0.98)

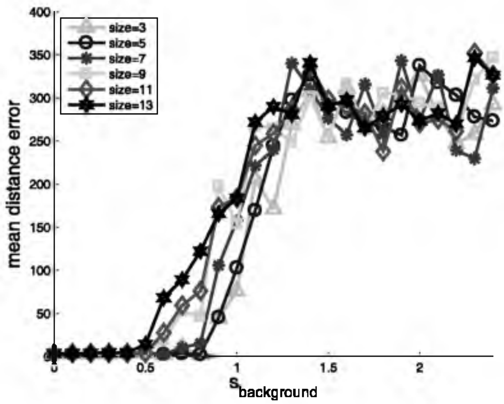

Fig. 5. Results for object size = 5 (smoothing coefficient 0.95)

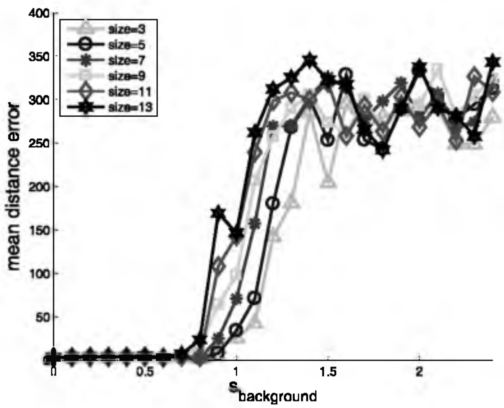

Fig. 6. Results for object size = 5 (smoothing coefficient 0.98)

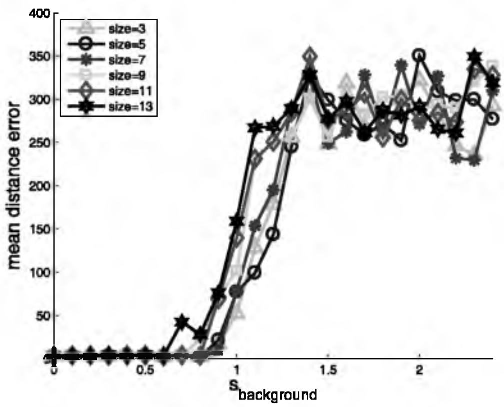

Fig. 7. Results for object size = 7 (smoothing coefficient 0.95)

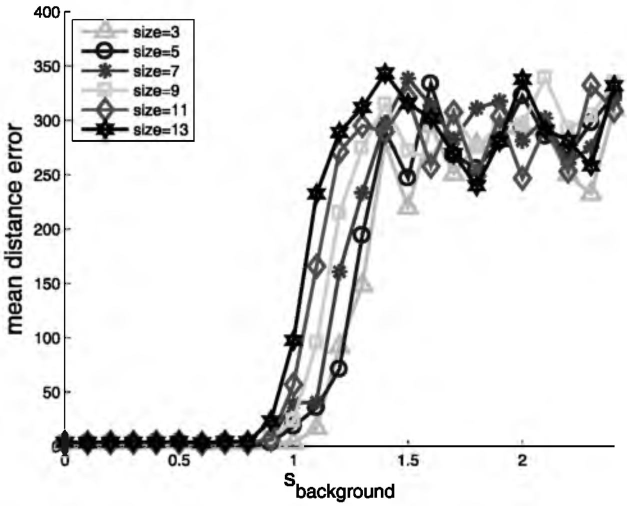

Fig. 8. Results for object size = 7 (smoothing coefficient 0.98)

5. Conclusions

Higher value of the smoothing coefficient gives better performance of detection, what is expected.

The detection of the Gaussian signal of object is possible by two ways. The first way is related to the difference between empirical distributions. The second way is related to the difference in object boundary regions due to Gaussian mixtures.

The best detection results are obtained for smaller window of analysis. The empirical distribution is very poor but the tracking abilities of TBD reduce the weakness of estimation. Large window size may overlap object or mixture area in many ways so position is not well estimated. The influence of the window size is smaller for large objects.

Small objects are more difficult to detect what is expected behavior. The proposed preprocessing is especially recommended for larger objects.

The influence of the selection of the distribution boundaries will be considered in further work.

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., Application of dot product for Track-Before-Detect tracking of noise objects, Poznań University of Technology Academic Journals – Electrical Engineering, no. 76, 101-107, 2013.
[6] 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.
[7] 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.
[8] 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.
[9] Mazurek P., Optimization of bayesian Track-Before-Detect algorithms for GPGPUs implementations, Electrical Review, R. 86 no. 7/2010, 187-189, 2010.
[10] Mazurek P., Optimization of Track-Before-Detect systems for GPGPU, Measurement Automation and Monitoring, vol. 56 no. 7, 665-667, 2010.
[11] Mazurek P., Optimization of Track-Before-Detect Systems with Decimation for GPGPU, Measurement Automation and Monitoring, vol. 56 no. 12, 1523-1525, 2010.
[12] Mazurek P., Track-Before-Detect Algorithm for Noise Objects, Measurement Automation and Monitoring, vol. 56 no. 10, 1183-1185, 2010.
[13] Stone L. D., Barlow C. A., Corwin T. L. Bayesian Multiple Target Tracking. Artech House, 1999.