

# Adaptive Kernel Background Intensity Estimation Based on Local 2D Orientation

Johannes Wintenby  
Saab Electronic Defence Systems  
Göteborg, Sweden

Email: johannes.wintenby@saabgroup.com

Daniel Svensson  
Chalmers University of Technology  
Göteborg, Sweden

Email: daniel.svensson@chalmers.se

**Abstract**—Many target tracking algorithms for radar systems assume homogeneous backgrounds of clutter. However, real backgrounds are rarely homogeneous. By estimating background intensity, and using the estimate in the likelihood measure, the tracking algorithm is given the ability to adapt to the background. In this work, a method for estimating the clutter intensity is introduced. The method is based on locally adaptive Kernel Density Estimation (KDE), where local 2D structure of the background in terms of energy and orientation controls the smoothing properties of the filter kernels. In regions with low clutter intensity, the kernel adopts low-pass characteristics, and the intensity estimate is based on observations from a larger volume. In regions where there are ridges in the clutter intensity, kernels are selected such that smoothing is carried out along ridges instead of across them. Peaks in the clutter intensity are left unsmoothed. The proposed method is compared to other methods on synthetic data. Additionally, a demonstration is given on recorded radar data.

## I. INTRODUCTION

The radar clutter background consists of all radar observations from non-targets, i.e. targets whose observation pattern deviate from that of typical targets. Radar systems have dedicated filtering techniques, such as CFAR (constant false alarm rate) [1] and clutter maps, designed to suppress or eliminate observations from various clutter sources. In practice, residual clutter observations may have spatial intensity that varies greatly from one region to a neighboring region. Such non-homogeneous clutter intensity constitutes a problem for a tracker which assumes a known and constant clutter intensity. A significant mismatch between the two reduces performance. A tracker therefore benefits from a function that estimates a spatially varying background intensity.

The background is suitably modeled as a non-homogeneous Poisson point process. The process is parametrized with its intensity function, which should be estimated. A natural step is to try out a PHD (Probability Hypothesis Density) filter [10], and in particular a Gaussian Mixture (GM) PHD [13], since these filters maintain the intensity of a non-homogeneous Poisson point process. However, the standard model used in the derivation of these filters for target tracking assumes that observations are generated by point sources that persist over time. For extended clutter sources, such as a rain cloud, this model is typically incorrect.

With modifications, a PHD for target tracking can still be used. In [8], a PHD filter was derived with separate dynamic

models for the clutter, e.g., a random walk and blown up system noise. A rain cloud is then for example modeled as a set of point sources that jump around. Unfortunately, the complexity of the spawning and merging process in the GM-PHD implementation is quite high for extended clutter. In [9] the measurement model is altered instead. The measurement noise is derived from a single scan estimate of the intensity, such that the measurement noise is low in dense regions and high in sparse regions. A rain cloud is then modeled as a set of point sources with fuzzified locations. In [4], clutter is modeled by a mixture of sources with Gaussian extent. A rain cloud is then represented by a set of extended sources, each producing at most one observation per scan distributed according to the Gaussian extent. Since the covariances are estimated, the mixture components follow a normal-Wishart distribution.

Mahler [11] adopts a general Poisson mixture for the background. This model is appropriate for extended clutter. However, there is an underlying problem of associating clutter to mixture components. It is shown in [11] that a PHD filter for the model requires a weighted sum over all partitions of observations into mixture components. Practical methods for this problem approximate the sum over partitions with clustering, e.g., using soft clustering according to Expectation Maximization (EM) GM estimation.

Considering the possible complexity of real radar data, see the example in Figure 1, there is a risk that methods building on GM-PHD require excessive management, in particular merging, and that the real time performance suffers. As an alternative, [7] provides an inverse intensity estimate derived from the center of a background cell to the time averaged distance to the nearest clutter point. The estimator is unbiased for inverse intensity in homogeneous clutter. Given non-homogeneous clutter, the performance can be greatly reduced.

Yet another alternative is Kernel Density Estimation (KDE) [12]. In standard, non-adaptive KDE, the observed data points are filtered spatially with a low-pass kernel uniformly over the observation space. Again, this is suitable for homogeneous clutter. Non-homogeneous contributions in the intensity functions such as point sources and ridges are smoothed to the same degree. Locally adaptive KDE approaches this problem by adapting the kernel bandwidth to the local structure and intensity of data at hand. In [3], a scheme is presented for adap-

tively selecting kernel bandwidth. An optimization method for a Gaussian bandwidth matrix is given that minimizes the mean integrated square error (MISE) of the clutter density estimate. To separately optimize a large number of local kernels with the presented method might be prohibitively expensive, and the local bandwidth matrix used in the scheme is a combination of an overall MISE optimized kernel, and a local scaling derived from a pilot estimate of the clutter intensity. Further reduction in computational effort is achieved with merging, where kernels related to past observations are accumulated into a Gaussian mixture. The merging is similar to that in GM-PHD.

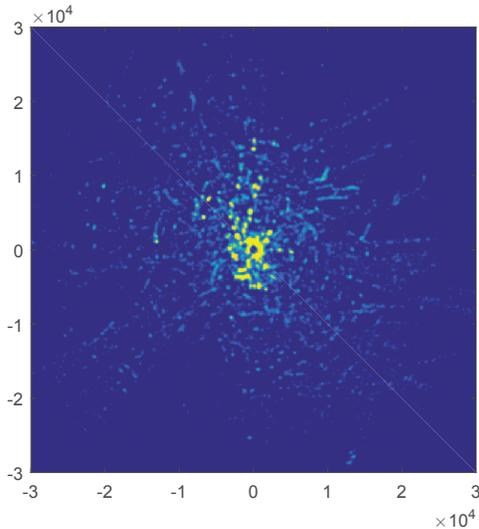


Figure 1. Histogram of observations from an experimental radar system. Most real targets are excluded. The color intensity represents number of samples per grid element.

In this paper, we present an alternative method to derive kernels which are locally adaptive in the horizontal 2D plane. The method is inspired from image processing, and in particular from estimation of local orientation. The aim is to select kernels based on the 2D spatial structure of the background such that

- 1) smoothing is made on similar kinds of background with slowly varying intensity (isotropic low-pass inside a region with similar background)
- 2) smoothing is made along ridges and edges instead of across them (low-pass along edges and all-pass across edges)
- 3) intensity of concentrated clutter sources should be left unsmoothed (isotropic all-pass).

Note that the kernels are fixed and uncoupled in other measurement dimensions, e.g., elevation and radial velocity. We believe this approximation is a reasonable trade-off between performance and complexity. The horizontal adaptation capture a lot of structure like ridges and peaks visible from above, and within a local neighborhood imposed by a horizontally

adapted kernel the correlations between the horizontal coordinates and other dimensions are typically low.

The paper has the following structure: Section II presents a short problem formation. Section III introduces kernel intensity estimation. In Section IV, an overview of the method for local orientation estimation is given. Section V provides a mapping of local orientation to kernel bandwidth. In Section VI, a comparison between some of the previous methods and our method is carried out on a synthetic test scenario. Additionally, a demonstration of results on real radar data is given.

## II. PROBLEM FORMULATION

We assume that the background is modeled with a non-homogeneous Poisson point process. In any region  $A$ , the number of observations,  $N(A)$ , is Poisson distributed with

$$P(N(A) = k) = \frac{(\mu(A))^k e^{-\mu(A)}}{k!},$$

where  $\mu(A) = E\{N(A)\}$  is the integrated intensity over  $A$ ,  $\mu(A) = \int_A I(x) dx$ . The problem is to estimate the intensity function  $I(x)$ . In target tracking, the intensity enters the standard data association likelihood measures, which include a Poisson model for the clutter process [2]. The intensity is often called  $\beta_{fa}$  in target tracking.

## III. KERNEL INTENSITY ESTIMATION

A locally adaptive kernel density estimator is given by [12]

$$\hat{f}(x) = \frac{1}{N} \sum_{i=1}^N K_{H(x_i)}(x - x_i) \quad (1)$$

where  $x_1, x_2, \dots$  is a set of spatial samples recorded over a time period,  $K$  is a kernel which integrates to one, and  $H(x_i)$  is a locally adaptive bandwidth matrix selected point-wise at each data sample, see Sections IV and V below. Alternatively, the bandwidth is selected at the evaluation point,  $H(x)$ , but the estimated density then loses the properties of smoothness and being a probability density function. In this work, the Gaussian kernel has been used primarily,

$$K_H(x) = \frac{1}{\sqrt{(2\pi)^n |H|}} e^{-x^T H^{-1} x / 2}.$$

To estimate intensity, the normalization with the number of samples in (1) is removed,

$$\hat{I}(x) = \sum_{i=1}^N K_{H(x_i)}(x - x_i). \quad (2)$$

Note that integrating  $\hat{I}(x)$  over a volume  $A$  gives the approximate number of recorded observations in the volume, which in a Poisson context is an estimate of  $\mu(A)$ , the expected number of observations in  $A$ . In this document, intensity estimation of the form (2) is denoted Kernel Intensity Estimation (KIE).

Due to the limited support of the kernels, only samples in the vicinity of  $x$  are required to evaluate (2). With samples

maintained in a horizontal position grid, and given a local kernel bandwidth we can automatically select relevant samples to the sum. Moreover, the more data in a region the smaller the bandwidths, and the number of samples in the sum is effectively limited.

*Temporal filtering:* To handle slowly changing backgrounds exponential forgetting is applied. The kernel for observation  $i$  in sum (2) is then extended to,

$$K_{H(x_i)}(x - x_i)e^{-(t-t_i)/t_f},$$

where  $t$  is usually time of the most recent scan  $k$ ,  $t_i$  is the time of observation  $i$ , and  $t_f$  is the forgetting time constant. Additionally, in target tracking we are interested in the number of observations per scan, and the intensity should be normalized with the number of scans,

$$\hat{I}_s(x, t) = \frac{1}{N_{scans}(t)} \sum_{i=1}^N K_{H(x_i)}(x - x_i)e^{-(t-t_i)/t_f} \quad (3)$$

Counting the number of scans over an exponentially weighted time window can be made with an AR-scheme,

$$\begin{cases} \Delta t_k & = t_k - t_{k-1} \\ N_{scans}(t_k) & = e^{-\Delta t_k/t_f} N_{scans}(t_{k-1}) + 1 \end{cases} \quad (4)$$

where  $t_k$  is the time of scan  $k$ . By this scheme, old observations and old scans at the same time instant are given the same exponential weight.

*Computational aspects:* It is important that the sums above are calculated efficiently. Observations are suitably maintained in a grid in the horizontal plane, either in Cartesian or polar coordinates, and at every grid cell the observations are put in a queue. In some cells the detection rate is higher, and the sums in (2) and (3) are more costly to evaluate. Therefore, cell-wise component management is required, and the following options can be considered:

- truncation: the tail of the queue is cut of so that the size is limited to say  $n \cdot 10$  observations per cell. (Note that the normalization with the number of scans in (3) will become conditioned on the cell. Moreover, cutting the tail in (4) must be compensated for.)
- pruning: aged components have low weight and should be pruned as well. A simple rule is to prune components with  $t - t_i < 3 \cdot t_f$ .
- merging: given Gaussian kernels, the sums above are Gaussian mixtures. Similar to GM-PHD and the method in [3], we can use merging to reduce the number of components.

Merging operations and evaluation of merging conditions are expensive, and the gain should be substantial to motivate merging. In this work the component management is restricted to the first two options, i.e., truncation and pruning.

#### IV. LOCAL ORIENTATION

Given the sums in (2) and (3), the problem is now to adaptively select the kernel bandwidth matrix  $H(x_i)$  for each recorded detection  $x_i$ . We base this selection on an estimate of local orientation in a 2D histogram,  $g$ , of observations over a Cartesian or polar grid. For every histogram grid cell, a local orientation estimate is computed, and the estimate is mapped to a local kernel bandwidth matrix, see Section V. A detection falling inside a grid cell inherits the local bandwidth matrix of the cell. The grid for computing local orientation herein is conveniently matched with the grid discussed for computational aspects in Section III above.

The histogram  $g$  can be regarded as an image, see the example in Figure 1 (a grid resolution of 200 meters is used is the example, and anti-aliasing is performed, see the end of this section). Local 2D-orientation of image data at a certain sample  $i, j$  is suitably represented with a symmetric  $2 \times 2$  matrix  $T$ , similar to a covariance matrix (the matrix is also denoted an orientation tensor or structure tensor in the literature) [5]. Locally, images are often relatively constant in one direction, and have the significant variation in the other direction. Given an eigenvalue decomposition,

$$T = \lambda_1 e_1 e_1^T + \lambda_2 e_2 e_2^T, \quad (5)$$

the significant direction of the variation corresponds to the eigenvector  $e_1$  with the largest eigenvalue  $\lambda_1$ , where  $\lambda_1 > \lambda_2$ . If  $\lambda_1 \gg \lambda_2$ , the variation in direction of  $e_2$  is comparatively insignificant. If  $\lambda_1 = \lambda_2$ , the orientation is isotropic. The Frobenius norm  $\|T\| = \sqrt{\sum T_{mn}} = \sqrt{\sum \lambda_i^2}$  indicates the overall energy in the local variation.

There are several methods to estimate the matrix  $T$  and we have followed a method based on directional filters described in e.g., [5], [6]. Only an overview of the essential parts are given here. It is advised to use a set of directional quadrature filters specified in the 2D Fourier domain,

$$\begin{cases} F_k(\omega) = R(\rho)(\hat{\omega} \cdot \hat{n}_k)^2 & \omega \cdot \hat{n}_k > 0 \\ F_k(\omega) = 0 & \omega \cdot \hat{n}_k \leq 0 \end{cases} \quad (6)$$

where  $\omega = [\omega_1, \omega_2]^T$ ,  $\rho = \|\omega\|$ ,  $R(\rho)$  is a radial bandpass function common to the set of filters,  $\hat{\omega}$  is the unit vector in the direction of  $\omega$ , and  $\hat{n}_k$  are filter specific directional vectors. Corresponding filters in the spatial domain are  $f_k = \mathcal{IDFT}(F_k)$ . A motivation behind quadrature filters is that they operate both as edge and line detectors. An interpretation is that they have superposed even and an odd parts which cancel in the negative hemisphere and coincide in the positive hemisphere. The even part transforms to a real and even part in the spatial domain with the ability to detect lines, and the odd part transforms to a complex and odd part in the spatial domain with the ability to detect edges. The filter localizes edges and lines equally well.

To measure direction in 2D, three filters are enough, and it is suitable to distribute the directions  $\hat{n}_k$  evenly,

$$\begin{aligned}\hat{n}_1 &= [1 \ 0]^T \\ \hat{n}_2 &= [-\frac{1}{2} \ \frac{\sqrt{3}}{2}]^T \\ \hat{n}_3 &= [-\frac{1}{2} \ -\frac{\sqrt{3}}{2}]^T.\end{aligned}$$

Figure 2 shows a contour plot of three quadrature filters in the Fourier domain given these directions and with a log-normal radial function, see filter selection below.

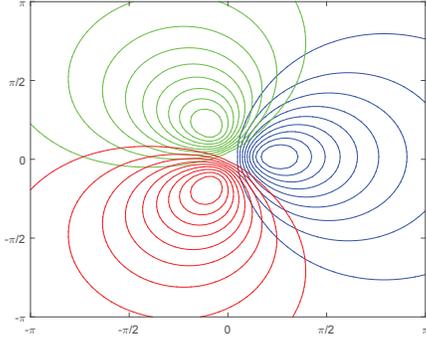


Figure 2. Contour plot in the Fourier domain of three quadrature filters following (6) with the directions  $\hat{n}_1, \hat{n}_2, \hat{n}_3$  specified in the text, and with a log-normal radial function  $R(\rho)$ , see (8).

The set of symmetric matrices  $\hat{N}_k = \hat{n}_k \hat{n}_k^T$  form a non-orthogonal basis for the space of symmetric 2D matrices. The scalar product in this space is defined as

$$\langle N, M \rangle = \sum_{i,j} N_{ij} M_{ij}.$$

Introduce a dual basis  $\{\tilde{N}_k\}$  to  $\{\hat{N}_k\}$  such that

$$\langle \hat{N}_i, \tilde{N}_j \rangle = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

With the selection of  $\tilde{N}_k$  above, the dual matrices are

$$\tilde{N}_k = \frac{4}{3} \hat{N}_k - \frac{1}{3} I$$

An arbitrary symmetric matrix can now be written  $M = \sum_k m_k \hat{N}_k$ , where the coordinates are calculated as  $\langle M, \tilde{N}_k \rangle = \langle \sum_l m_l \hat{N}_l, \tilde{N}_k \rangle = m_k$ , and therefore  $M = \sum_k \langle M, \tilde{N}_k \rangle \hat{N}_k$ . Likewise, we have  $M = \sum_k \langle M, \hat{N}_k \rangle \tilde{N}_k$ .

Assume a simple, one-dimensional, local signal neighborhood,  $s(\xi) = g(\xi^T \hat{x})$  (the Fourier transform of  $s$  is along a line in the  $\hat{x}$  direction). It is possible to show that the magnitude of the filter response from quadrature filter  $k$  is  $q_k = A(\hat{x}^T \hat{n}_k)^2$ , where  $A = |\frac{1}{2\pi} \int R(\rho) S(\rho) d\rho|$  and  $S(\rho)$  is the local signal spectrum in the direction of  $\hat{x}$ . There is no directional dependence in  $A$ . If we write the local orientation given the simple signal neighborhood as  $T = A \hat{x} \hat{x}^T$ , the filter responses are  $q_k = A(\hat{x}^T \hat{n}_k)^2 = A \langle \hat{x} \hat{x}^T, \hat{n}_k \hat{n}_k^T \rangle = \langle T, \hat{N}_k \rangle$ . Finally, given the dual basis we have

$$T = \sum_k \langle T, \hat{N}_k \rangle \tilde{N}_k = \sum_k q_k \tilde{N}_k. \quad (7)$$

In conclusion, to estimate local orientations we do the following sequence of operations:

- 1) run the three spatial, complex filters over the 2D array  $g$ , i.e., compute the convolutions  $f_k * g$  for  $k \in \{1, 2, 3\}$ ,
- 2) take the local magnitudes of their responses point-wise,  $q_k(i, j) = |(f_k * g)(i, j)|$ , for all  $i, j$ ,
- 3) use  $q_k(i, j)$  as weights in the dual basis according to (7) and we arrive at the local orientation  $T(i, j)$ .

Below, some practical considerations are discussed.

*Filter selection:* A suggestion in [5] is to use a log-normal filter for the radial filter function,

$$R(\rho) = e^{-\frac{4}{B^2 \ln^2} \ln^2(\rho/\rho_0)}, \quad (8)$$

where  $B$  is the relative bandwidth and  $\rho_0$  is the center frequency. In the evaluations, we have used  $\rho_0 = 0.7$  and  $B = 3$ . Varying the parameters has little effects on the final results.

The directional log-normal filters  $F_k(\omega)$  are specified in the Fourier domain. There is no known inverse transform to get  $f_k$ , and we use optimization to generate spatial kernels [5]. That is, the samples of  $f_k$  are tuned to get a good match to  $F_k(\omega)$  in the Fourier domain,

$$f_k^* = \arg \min_{f_k} \|F_k(\omega) - \mathcal{DFT}\{f_k\}(\omega)\|.$$

We have universally used a kernel size of 15x15 grid cells for  $f_k$ .

*Grid selection:* To handle background at a certain scale, the grid resolution is selected to match that scale rather than altering the design of the filter kernels. However, the grid size is limited for computational reasons. Given a Cartesian grid, we aim at a maximum of approximately  $500 \times 500$  samples. For a system with visible background up to 50km, a reasonable grid size is then  $200 \times 200$  meters (used in the Figure 1).

In a polar grid, the resolution scales automatically with range. For a moving system though, the polar grid must continuously be translated such that the sensor is at the origin. Given the Cartesian grid, sensor motion over short time scales can simply be subtracted. Eventually, the sensor has moved a certain distance from the grid center and the grid must be translated to the new position.

*Anti-aliasing:* Aliasing is a problem when working with gridded data. Anti-aliasing is herein handled in a standard fashion by 1) forming the histogram  $g$  on a grid with double resolution, 2) applying a Gaussian low-pass filter with bandwidth  $\omega_0 = \pi/4$ , and 3) down-sampling  $g$  by a factor 2. The quadrature filters are applied to the down sampled grid.

*Averaging:* In [5], low-pass filtering is applied to the coefficients of the estimated orientation matrices. Rapid changes in the orientation are suppressed. Additionally, the consistency of the orientation estimate around corners is improved. A corner does not have a unique orientation, and the averaging of the orientation from two perpendicular edges results in an isotropic orientation. The averaging is applied to the elements of  $T(i, j)$  separately. Thus, if

$$T(i, j) = \begin{bmatrix} t_1(i, j) & t_2(i, j) \\ t_2(i, j) & t_3(i, j) \end{bmatrix} \quad (9)$$

the 2D arrays  $t_1$ ,  $t_2$ , and  $t_3$  are averaged separately. An isotropic filter of small size is appropriate.

Figure 3 shows an example of estimated orientation for a line shape and a rectangular shape. Filter selections are as above. One can observe that the energy, i.e., the length of the vectors, is greater close to the shapes.

*Temporal filtering:* For the same reasons as for the kernel evaluations, the histogram grid  $g$  should be subject to exponential forgetting. At every scan, the histogram is multiplied with a forgetting factor,  $g(i, j) := e^{-\Delta t_k / t_f} g(i, j)$  before new data enters the histogram.

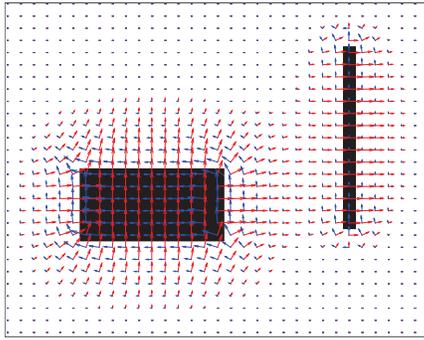


Figure 3. Example of estimated orientation for a line object and a rectangular object. The vector pairs at each point  $i, j$  illustrate the eigenvalue decomposition of the estimated local orientation,  $T(i, j)$ . The vectors with red colors correspond to  $\sqrt{\lambda_1} e_1$  (large eigenvalue), and the vectors with blue colors correspond to  $\sqrt{\lambda_2} e_2$  (small eigenvalue).

## V. MAPPING OF ORIENTATION TO FILTER KERNEL

The estimated orientation  $T(i, j)$  at position  $i, j$  should be mapped to a bandwidth matrix  $H(i, j)$  at the same grid position so that it can be used in (2). Unfortunately, we do not have a super-level modeling framework for guiding us to interpret  $T(i, j)$  and optimize a response  $H(i, j)$ . Thus, the mapping is subject to ad hoc considerations. First, a fundamental observation is that  $H(i, j)$  should have an inverse relation to  $T(i, j)$ . That is, a large local variation in one direction should be met with a small filter bandwidth in the same direction. The reasoning can be applied to each eigenvalue and eigenvector separately. The following exponential mapping empirically produces reasonable results,

$$\begin{cases} \gamma_i &= f_{max}^2 e^{-\frac{\lambda_i^f}{c}} + f_{min}^2 \\ H &= \gamma_1 e_1 e_1^T + \gamma_2 e_2 e_2^T. \end{cases} \quad (10)$$

The constants  $c$ ,  $f_{min}$ ,  $f_{max}$  are design parameters, and  $[\lambda_1^f, \lambda_2^f] = f([\lambda_1, \lambda_2])$  are normalized eigenvalues so that  $\lambda_i^f \in [0, 1]$  (see below). In the evaluations we have used  $c = 0.15$ , so at  $\lambda_i^f = 1$  the attenuation of  $f_{max}^2$  in (10) is  $e^{-\lambda_i^f / c} \approx 0.001$ , meaning that  $f_{min}^2$  is essentially selected as kernel size in the  $e_i$  direction. That is, the bandwidth is high in

the  $e_i$  direction. Contrary, at  $\lambda_i^f = 0$  we get essentially  $f_{max}^2$ , the larger kernel size, and the lower bandwidth. It is important that the selection of  $f_{max}$  matches the extent and bandwidth of the directional filter kernels. A too large  $f_{max}$  smooths the signal beyond the support of the directional kernels. For example, a ridge in data would leak to a neighboring region where the transient of the orientation estimate from the ridge is essentially zero. Given the log-normal kernels with size  $15 \times 15$  specified above, we use  $f_{min} \approx 0.7\Delta$  and  $f_{max} \approx 3\Delta$  with some variations, where  $\Delta$  is the grid resolution.

*Normalization:* By selecting a mapping of  $\lambda_1$  and  $\lambda_2$  to  $\lambda_1^f$  and  $\lambda_2^f$  we can decide what is a prominent structure, and what is not. Given filter selections according to above, including anti-aliasing, a single observation yields a maximum filter response in orientation energy  $\|T\|$  of approximately 0.05. A single observation is clearly no structure and should result in the largest kernel,  $f_{max}$ , isotropically. Perhaps two or three observations at the same position is the beginning of structure. Based on this reasoning, we may set  $\lambda_1^f$  and  $\lambda_2^f$  to 0 for  $\|T\| < \alpha$  with say  $\alpha = 0.1$ .

At the upper limit, it is important to avoid clipping effects around the high peaks, i.e., the high peaks give plateaus with  $\lambda_1^f = 1$ . The high bandwidth would otherwise be used in too large regions around the peaks yielding undesirable ripple. This is an argument for scaling the filter responses so that the highest energy peak,  $\max_{i,j} \|T(i, j)\|$ , results in  $\lambda_1^f = 1$ . The smaller eigenvalue,  $\lambda_2$ , is preferably scaled the same amount as  $\lambda_1$  so that the shape of  $T$  is maintained. A mapping condensing the reasoning herein is,

$$\begin{cases} \lambda_1^f(i, j) &= \max\left(\frac{\|T(i, j)\| - \alpha}{(\max_{k,l} \|T(k, l)\|) - \alpha}, 0\right) \\ \lambda_2^f(i, j) &= \lambda_2(i, j) \frac{\lambda_1^f(i, j)}{\lambda_1(i, j)}. \end{cases} \quad (11)$$

The mapping might be too conservative in promoting structure. In [5], specific functions are provided such that they can be tuned to enhance or suppress image features. Note that in image enhancement, minor distortions which are invisible to the eye are acceptable while they may be not in the context of density estimation.

An alternative to the uniform normalization over the entire 2D array in (11) is to use regional normalization, for example accomplished by a low pass filtered regional max.

*Filter kernels in measurement coordinates:* The target tracker requires intensity expressed in the measurement space, i.e., range, bearing, elevation, possibly radial velocity, and radar cross section (RCS). The method herein adaptively computes kernel bandwidth matrices only in the horizontal plane, either in Cartesian or polar coordinates. In the other dimensions, there is no adaptation at this point, and we select fixed bandwidths. In practice, the local neighborhood imposed by a horizontally adapted kernel implies that the correlations to and between other dimensions are typically low. A bandwidth matrix then has the following structure



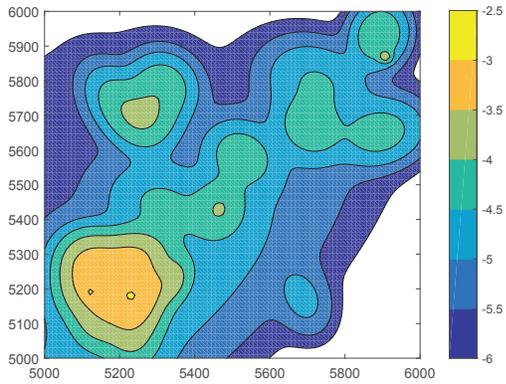


Figure 5. Example of estimated intensity from the GM-PHD filter. The color scale is logarithmic and ranges from  $10^{-6}$  to  $10^{-2.5}$ , see the color bar.

is set to 0.9. The result is normalized with the number of scans, also AR-filtered by the same parameter, see (4). Figure 6 shows an example of estimated intensity, given exactly the same data that Figure 5 is based on. Comparing with the GM-PHD, the estimated intensity has more distinct transitions from high to low intensity, and the intensity levels in respective region are overall close to the true values ( $10^{-5}$  and  $10^{-3}$ ).

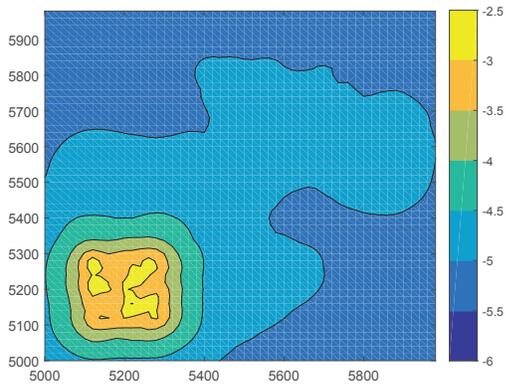


Figure 6. Example of estimated intensity from the inverse intensity estimator. The color scale is logarithmic and ranges from  $10^{-6}$  to  $10^{-2.5}$ , see the color bar.

*Locally adaptive KIE:* The corresponding result for KIE on the same data set is shown in Figure 7. The grid density is here  $\Delta = 20$  meters, and the time constant is 30 scans. In comparison with Figure 6, the transition region between the high and the low clutter areas is even sharper. Excluding the transition region, the estimated intensities inside the two areas are close to the true values. KIE manages to spatially and temporally smooth the estimate to a much higher degree than the GM-PHD with given parameters.

The kernels in KIE have limited support, implying that the intensity is low where there is no data. This effect can be seen on the border of the figure. The inverse intensity estimator on the other hand may extend the estimate farther. Reducing the grid resolution improves the support while affecting the sharpness in transitions.

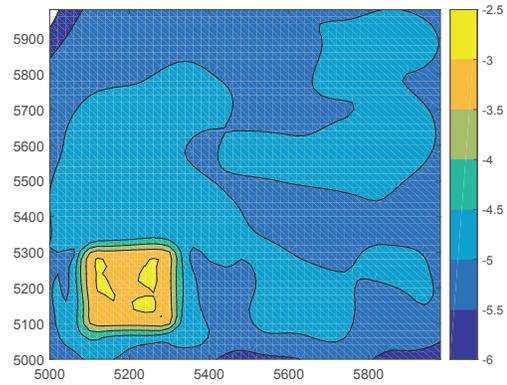


Figure 7. Example of estimated intensity from the locally adaptive KIE. The color scale is logarithmic and ranges from  $10^{-6}$  to  $10^{-2.5}$ , see the color bar.

### B. Demonstration on real data

We have also applied KIE on recorded radar data from an experimental radar system, see the histogram in Figure 1. The background in this data set is highly non-homogeneous, and our implementation of the inverse intensity estimator fails to pickup the local variations, see the contour plot in Figure 8 (the data set contains approximately 300 scans, and the forgetting factor is set to 0.95).

Figure 9 shows the corresponding results for KIE. The method picks up peaks well and has a fair representation of details. The contours nicely follow the edges of regions with increased intensity. Computationally, the method is fast and real time feasible for high data rates. Note that online operation only requires the intensity estimator (3) to be evaluated point-wise for every new observation. Moreover, the filtering operations for calculating local orientations and adaptive kernels can be carried out at a moderate rate with regular intervals.

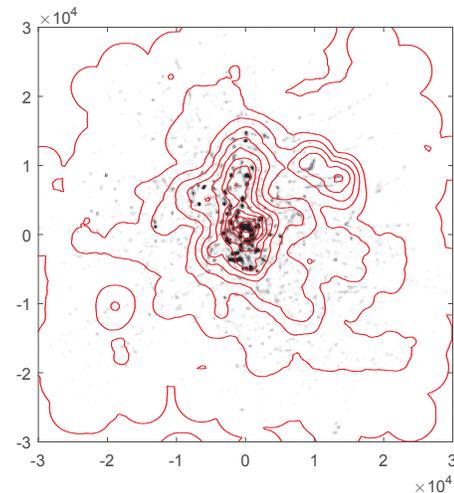


Figure 8. Isocontours of the estimated intensity from the inverse intensity estimator given the data in Figure 1. The contour intervals are logarithmic.

*Example of KIE in other dimensions:* The intensity estimation handles other dimensions by extending the kernel

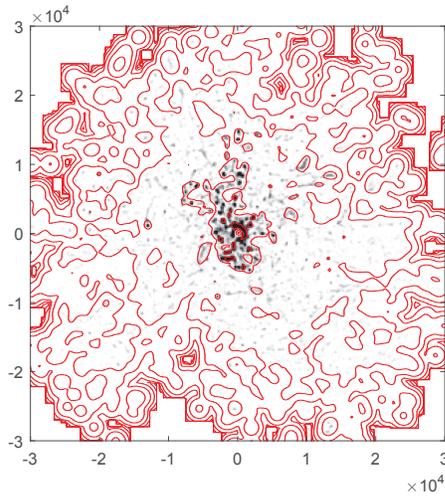


Figure 9. Isocontours of the estimated intensity from the locally adaptive KIE presented herein, given the data in Figure 1. The contour intervals are logarithmic.

from 2D to ND using the diagonal bandwidth matrix according to (12). We can extract data in arbitrary subspaces. Conditional probabilities are easily calculated given KDE (1). For example, let  $\sigma$  denote RCS. Then the conditional probability of RCS at position  $x, y$  is expressed by

$$p(\sigma|x, y) = \frac{p(x, y, \sigma)}{\int p(x, y, \sigma) d\sigma}.$$

Independence in the bandwidth matrix implies that we can write

$$\begin{aligned} \hat{f}(x, y, \sigma) &= \frac{1}{N} \sum_{i=1}^N K_{H_{xy,i}} \left( \begin{bmatrix} x - x_i \\ y - y_i \end{bmatrix} \right) \cdot K_{H_{RCS}}(\sigma - \sigma_i) \\ &= \sum_{i=1}^N w_{xy,i} \cdot K_{H_{RCS}}(\sigma - \sigma_i) \end{aligned}$$

where

$$w_{xy,i} = \frac{1}{N} K_{H_{xy,i}} \left( \begin{bmatrix} x - x_i \\ y - y_i \end{bmatrix} \right).$$

An estimate of the conditional probability density is therefore

$$\hat{f}(\sigma|x, y) = \frac{\sum_{i=1}^N w_{xy,i} \cdot K_{H_{RCS}}(\sigma - \sigma_i)}{\sum_{i=1}^N w_{xy,i}}.$$

Figure 10 shows an example of estimated conditional probability density of RCS at a given position. The weighted histogram is also given for comparison, i.e., sample  $i$  is weighted with  $w_{xy,i}$ . Such a diagram is helpful for analysis of background properties in the vicinity of a position.

## VII. CONCLUSIONS

In this report, a new method for estimation of clutter intensity is presented. The method is based on kernel density estimation, and in particular estimation of local orientation in the horizontal plane which provide bandwidths for the kernels.

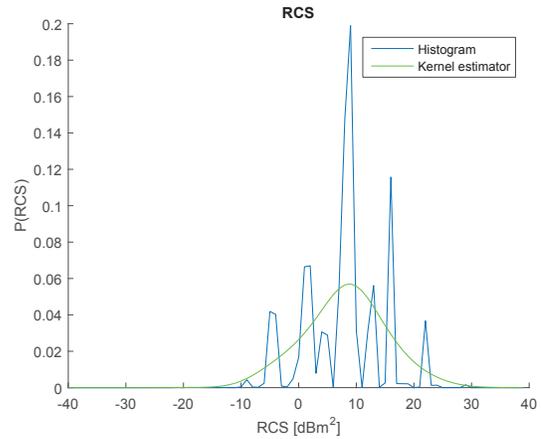


Figure 10. Example of conditional probability of RCS at a certain position.

The local orientation estimates reflect the underlying structure of background data, and the estimation thus adapts to the recorded background.

The method has been implemented and tested on both synthetic and recorded radar data. The results indicate that the method performs well in comparison to the other methods in the evaluation. The simplicity of the method further imply that there is a good chance that the method will be robust for many different kinds of background. Finally, the method is computationally feasible for real-time processing.

## REFERENCES

- [1] D. Barton. *Modern Radar System Analysis*. Artech House, 1988.
- [2] S.S. Blackman. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.
- [3] X. Chen, R. Tharmarasa, T. Kirubarajan, and M. Pelletier. Online clutter estimation using a Gaussian kernel density estimator for target tracking. *Proceedings of the 14th International Conference on Information Fusion*, 2011.
- [4] X. Chen, R. Tharmarasa, M. Pelletier, and T. Kirubarajan. Integrated clutter estimation and target tracking using Poisson point process. *IEEE Transactions on Aerospace and Electronic Systems*, 48(2), 2012.
- [5] G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1994.
- [6] L. Haglund. *Adaptive Multidimensional Filtering*. PhD thesis, Linköping, Sweden, 1992.
- [7] T. Hanselmann, D. Musicki, and M. Palaniswami. Adaptive target tracking in slowly changing clutter. *Proceedings of the 9th International Conference on Information Fusion*, 2006.
- [8] R. Jonsson, J. Degerman, D. Svensson, and J. Wintenby. Multi target tracking with background discrimination using PHD filters. *Proceedings of the 15th International Conference on Information Fusion*, 2012.
- [9] W. C. Kim, D. Musicki, T. L. Song, and J. S. Bae. A multi scan clutter density estimator. *Proceedings of the 16th International Conference on Information Fusion*, 2013.
- [10] R. Mahler. Multi-target Bayes filtering via first-order multi-target moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39, 2003.
- [11] R. Mahler. CPHD and PHD filters for unknown backgrounds II: multitarget filtering in dynamic clutter. *Proceedings of SPIE*, 7330, 2009.
- [12] B. W. Silverman. *Density estimation for statistics and data analysis*. *Monographs on Statistics and Data Analysis*, Chapman and Hall, 1986.
- [13] B. Vo and W. Ma. The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Aerospace and Electronic Systems*, 54(11), 2006.