A Noise Reduction Method based upon Statistical Analysis for the Detection of Weak Signals in Discrete Data

Dissertation

zur

Erlangung des Doktorgrades (Dr.rer.nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität

zu Bonn

vorgelegt von

Andrea Löhr

aus

Arienheller

Bonn 2003

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Universität Bonn

Referent: Prof. Dr. U.-G. Meißner
Korreferent: Prof. Dr. R. G. Winkler
Tag der Promotion:

To **O** *and* **E** *and* **S**

Contents

List of Figures

1 Introduction

We describe a new method for the detection of statistic anomalies in discrete data. Its aim is the separation of weak signals from heavy noise. The method combines several means of statistical data analysis to a mapping of data sub-samples that can serve as an indicator of statistical anomalies. It is built to form a simple algorithm that requires computation load only linearly proportional to the number of data. It is thus very useful, if vast amounts of data have to be processed very quickly. Its aim is the fast detection of a weak signal and the reduction of false alarm detections at the same time. The algorithm was developed for the detection of trajectories in discrete three dimensional data that are significantly contaminated with false data and is widely applicable.
The specific problem we developed this method for, arises in the field of tracking: we want to find trajectories of moving objects in a set of discrete data.

In many engineering applications (e.g. surveillance, guidance, navigation, robotics, system control, bubble-chamber experiments, image processing) sensors, single or networks, are used to collect information on time varying quantities (e.g. kinematic parameters, feature information). This process can more generally be described as the tracking of time varying signal parameters. The state of a stochastically driven dynamical system is to be estimated from a series of sensor data sets received at discrete instants of time.
Tracking is the processing of measurements obtained from an object in order to maintain an estimate of its state at the time of the measurement:

1. kinematic components: position, velocity, acceleration

2. non-kinematic components: signal amplitude, spectral characteristic, feature information.

Measurements are the noise corrupted observations related to the state of an object.
The measurements of interest, i.e. the input data of the tracking processing unit or tracker are not raw data points but pre-processed outputs of signal-processing units and detection subsystems. The main objective of the tracking unit is to associate data of subsequent measurements that belong to the same object. The tracker has to establish the state trajectory estimated from a set of measurements associated with the same object, i.e. to establish a *track*. A complex data association logic is necessary to sort out the data into the general categories of objects of interest, recurrent sources that are of no interest and false signals with little or no correlation in time.
The categories of data association (what is being associated with what) are:

measurement-to-measurement:	track initiation
measurement-to-track:	track maintenance
track-to-track:	track fusion

Once tracks are formed and confirmed so that background and false objects are reduced one can estimate the kinematic parameters, future predicted position and object classification characteristics for each track. Often, data association systems use filtering techniques for the recursive object state estimation of moving objects.

In addition to the inaccuracy of measurements (which can usually be modeled by additive noise) there is also an uncertainty associated with the measurements related to their origin. The crux of multi-object tracking is to carry out the data association for measurements whose origin is not only unknown in the first place but also uncertain due to random false alarms in the detection process or due to spurious data.
This leads to the important and challenging task of tracking in high false alarm environment. The tracking objective of collecting data from the sensor's field of view, find objects of interest and establish an estimated state trajectory is rendered significantly more difficult under the presence of huge amounts of uncertain measurements. Data association conflicts arise with the standard estimation algorithms. In environments where spurious measurements occur frequently these methods lead to poor results: they are not only extremely slow when processing huge amounts of data they also lead to a great number of false alarms. The high sensitivity of modern detection systems can lead to huge data loads, the processing of which takes computational time of $O(N^2)$ or higher when N data have to be tested for association with $N-1$ data.

The most common data association methods and estimation algorithms include the nearest neighbor method, where the data point nearest in some sense to the previous is being associated, as well as the sequential extraction of trajectories using sequential likelihood tests and signal strength information (e.g. [vKeuk1996], [vKeuk1998]).
Many mathematical ideas relevant to the design of tracking systems can be discussed in terms of statistical data processing that basically makes use of Bayes' rule ([Stone1999]). In these terms, tracking consists of an iterative updating scheme for conditional probability densities that describes the object state on grounds of all available information. The iterative state estimation, also referred to as filtering, often implies methods like the Kalman filter (e.g. [Bar-Shalom], [Steven1988], [Harp2002], [Davidson]) for single dynamics models, $\alpha-\beta$-filter (e.g. [Kanetos]), Median filter ([Wu]), probabilistic data association filter (PDAF) (e.g. [Bar-Shalom], [Colegrove2002]). The easiest model for the underlying system dynamics is a linear model that can be well and sufficiently handled with a Kalman filter. For non-linear systems this filter has been further developed to the extended Kalman filter ([Bar-Shalom]). There are even more complex models taking into consideration not only multiple possible forms of object motion but also the interaction of them (IMM, interacting multiple model) (cf. e.g. [Koch2000]).
These methods have in common, that in the course of subsequent measurements they choose the best (in terms of likelihood) data association hypothesis and this decision is irrevocable. There are more sophisticated approaches that delay the final assignment in polyvalent situations and maintain several association hypotheses until the next measurement(s) has (have) been obtained. These are the so-called Multi-Hypotheses-Tracking (MHT) methods that have become widely popular with the increasing computational possibilities (e.g. [Bar-Shalom]). Implementations of these methods may make use of the above mentioned filtering techniques but also use Lagrange relaxation ([Winkler2000]) or joint probabilistic data association (JPDAF).
Lately many approaches try to make use of image processing methods such as the Radon

or especially the Hough transformation and all its variates (standard, probabilistic, randomized, hierarchical), all of which proved unsuitable because of too high an error rate, e.g. [McLaughlin1998], [Binias2000], [Kern2000], [Wolf2000]. These methods can well be applied for feature detection in images but trajectories in discrete data do not provide enough information to produce secure and unique results. Another major disadvantage of both methods is, that one cannot search for any feature but has to declare the specific feature to be looked for. Of course, one can apply the transformation consecutively for all possible trajectory features (straight line, curves), but this also implies a huge computational effort.

With all these methods data association conflicts arise and increase if a high false alarm background is to be taken into account. These conflicts lead to ambiguous correlations and false or non detection of trajectories. Thus the detection of trajectories in noise corrupted discrete data is by no means trivial, if the sensor data are of uncertain origin or if there are uncertainties as to the dynamics of the underlying system.

Considering three dimensional discrete data containing a vast amount of uncorrelated spurious data points that lead to false detections and also containing a weak signal belonging to a trajectory consisting of a small amount of uncorrelated data we have found a new method to separate this weak signal from the noise corrupted information. We avoid the usage of common data association methods and introduce a new quick detector of such signals that requires computational time of only $\mathrm{O}(N)$.

This method makes use of statistic quantities that differ significantly if calculated for correlated or uncorrelated data.

Assuming we have noise consisting of entirely uncorrelated data and a signal consisting of correlated data: We subdivide the data set into sub-samples and map these cut-outs to certain statistical quantities. The expected values of these quantities differ for correlated and uncorrelated data. The mappings are then suitable to serve as an indicator of statistical anomalies, i.e. as a distinguisher between data sub-samples containing entirely uncorrelated data or correlated and uncorrelated.

We like to point out that the method is not suitable to extract the trajectory itself, it can rather reduce the noise around trajectories. It can select small portions out of a volume of three dimensional discrete data that contain a trajectory with a certain, calculable probability. It can thus reduce the data set to be searched for trajectories to the ‘interesting’ bits and serves as a noise-reducer. With this method we can improve the performance of common tracking systems in finding trajectories in high false alarm environment by reducing the noise background. Applications of which can range from bubble-chamber experiments and tracking with radar systems to finding comets in the night sky.

2 Basic principle of the noise reduction method

2.1 Division of data into local cut-outs

Tracking an object means to observe and estimate from these observations the time varying parameters of interest, mainly the location of the object. We want to find trajectories left by moving objects in discrete three dimensional data. The three dimensions will then be two spatial ones and the time. If we collect data that have been received at discrete instants of time over a certain time interval and the data contain a moving object e.g. linearly and straight then the trajectory consists of data points that are linearly correlated in time and space. Noise and disturbances have their origin in measurement insufficiencies or observational inaccuracies. In addition to this kind of noise we often have to deal with spurious data whose origin is unknown. These spurious data points lead to data association conflicts and result in false or non detections when looking for trajectories.
Our aim is to extract the few correlated data points that form the trajectory out of large amounts of uncorrelated spurious data points (noise) quickly, with good detection probability and at a low false alarm rate. The method we developed only requires the three dimensional co-ordinates of the data under discussion and is thus of great use, if detection methods that also require additional information (e.g. signal amplitude) fail to work.

Our idea is to treat an area containing ambiguous data in sets of integrated layers of one discrete co-ordinate such as the time, if we want to find out about the time varying parameters. We process the area in subsets that we call **local cut-outs**. These cut-outs are three dimensional sub-samples or chunks of the data into which the data volume under discussion is being divided. The data chunks contain integrated information from several time-layers. If the data contain a trajectory then there are data points that are correlated from layer to layer so forming the trajectory. Contrary to these data points belonging to the trajectory the spurious data (noise) are uncorrelated and assumed to be rectangularly distributed in each of the three dimensions. (Note: this assumption is only being made and valid in each local cut-out, but not globally for the entire data set. Should this assumption not be fulfilled, we suggest, as will be shown later, a simple co-ordinate transformation.)

2.2 Mapping the cut-outs to statistical quantities

Having divided the data set into numerous cut-outs, we apply a mapping to all data in each cut-out, that allows us to distinguish between cut-outs containing correlated information or exclusively uncorrelated information.

In Fig. 1 the basic principle of the method is visualized: The topmost picture shows two components of the three dimensional data. The x-axis marks a position, the y-axis marks the time. The noise data have been generated randomly and contain a trajectory in their center. The object to this trajectory does not move in x-direction (it moves in the second spatial direction, which is not shown here). The trajectory data follow a noisy trajectory representing the measurement inaccuracies that usually occur in real detection systems. The number of data points belonging to the trajectory is equal to the number of time-layers in this data set, as each object is detected only once per layer (unlikely exceptions possible). In this realistic but nevertheless artificial scenario the trajectory can be seen with the naked eye, which needs not be the case. The ratio of noise data to trajectory data (i.e. the signal to noise ratio) is 18 to 432.

The red circle represents a local cut-out, indicated in the cut-out are a few time layers and the geometric center. All data in this cut-out are now being collected and mapped to a single statistic quantity, which in this example is the second-greatest eigenvalue of the covariance matrix calculated from all data dimensions. The cut-out's central co-ordinate is on the x-axis of Fig. 1, II and the calculated eigenvalue on the y-axis. We now process all data from the data set by moving the cut-out along the x-axis in small increments. The next cut-out on the way to process all data would have (t, x+Δx) as central co-ordinate.

Mapping the entire data set in such a way leads to Fig. 1, II. As can be seen, the chosen eigenvalue is a very sensitive quantity: as soon as the cut-out contains correlated data, it decreases significantly. The smallest value occurs, when the trajectory runs right through the cut-out's center; but even before, the eigenvalue decreases when only part of the trajectory lies within the cut-out.

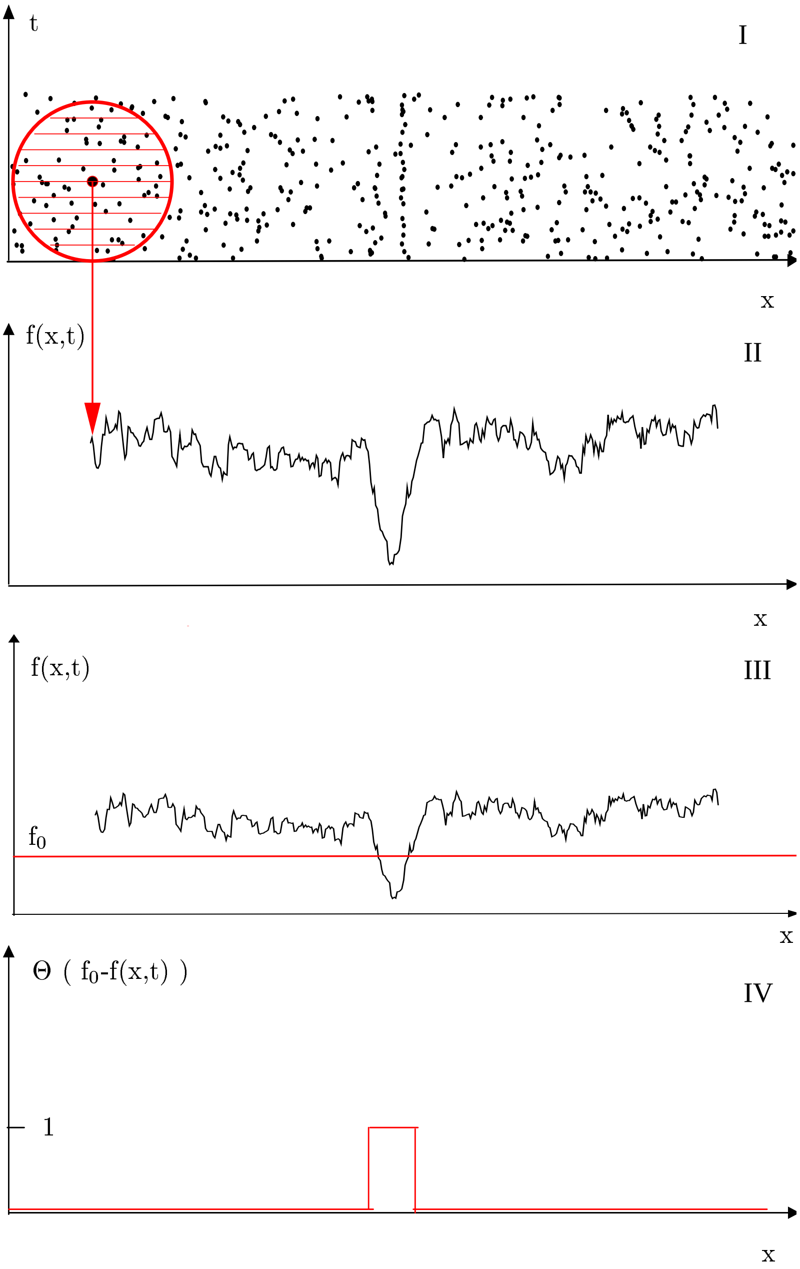

Figure 1: **I Data:** Two components (x,t) of the three dimensional data showing noise and a trajectory. Red is a local cut-out containing numerous plots, indicated are the time-layers representing the different observation times. **II Mapping:** The cut-outs are moved across the data set. The data points they contain are mapped to a single value. As can be seen, there are statistical quantities that behave significantly different on uncorrelated and correlated data. Here, an eigenvalue of the covariance matrix is significantly smaller, if a cut-out contains a trajectory. **III Threshold:** The problem of distinguishing the trajectory from noise data has now changed to finding a constant noise level in the mapped results. We can then define a threshold f_0, below which we claim the presence of correlated data. **IV Decision:** According to this threshold f_0 we decide whether the data in a cut-out contain a trajectory using

$$\Theta(f_0\text{-}f(x,t)) = \begin{cases} 1 \text{ (trajectory)} & \text{if } f(x,t) < f_0 \\ 0 \text{ (noise alone)} & \text{if } f(x,t) > f_0. \end{cases}$$

2.3 Cut-out configurations

While moving the cut-outs over a large data set various data configurations can lie within these sub-samples. Fig. 2 shows various ways in which a cut-out can be composed: it may contain only noise (a), noise and a trajectory straight through its center (b, i) or noise and parts of the trajectory (b, ii). For the detection of a trajectory, case (b, i) is obviously the best configuration, as in this case the trajectory and thus all correlated data lie completely within the cut-out. Due to the spherical geometry of the cut-out, the number of correlated data points lying within decreases with increasing distance of the trajectory from the diagonal position in the cut-out.

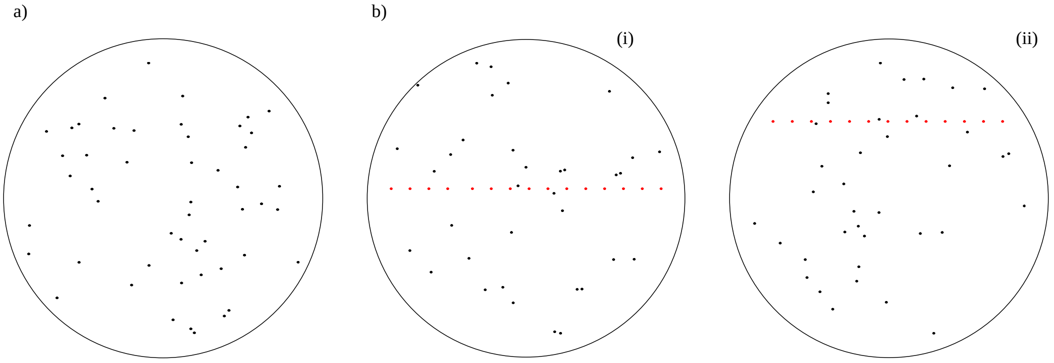

Figure 2: Local cut-out a) noise, b) noise with trajectory (i) offset 0,(ii) offset greater 0.

Fig. 1, III shows how we now have to set a value f_0 to serve as a threshold: if the calculated value $f(x,t)$ is greater than this threshold f_0, we can exclude that the cut-out under discussion contains correlated data. If the calculated eigenvalue is smaller than this threshold f_0, we assume the area in the cut-out contains more than only noise, e.g. a trajectory.
We then use the unit function of Heaviside (Fig. 1, IV) to make a decision whether the cut-out contains a trajectory or noise alone:

$$\Theta\ (\mathrm{f}_0\text{-f(x,t)})= \begin{cases} 1\ \text{(trajectory)} & \text{if f(x,t)<f}_0 \\ 0\ \text{(noise alone)} & \text{if f(x,t)>f}_0. \end{cases}$$

We can thus extract from a huge amount of data those sub-samples that contain all correlated data with a certain probability.

In the following chapters we will study in full detail, which mappings are suitable for the detection of correlated data, how they perform under various conditions. We want to find the best combination of all mappings to build a reliable trajectory detector in high false alarm environments. Finally, we will give an example for an interesting application of the method and show how and that it works on real data.

3 Analytical and simulative investigations on mappings

3.1 Composition of cut-outs

As described in the previous chapter the data to be processed are being divided into local spherical portions (cut-outs). For each of these cut-outs we investigate the hypothesis H_0:

The cut-out contains uncorrelated data, i.e. noise.

We now have to find and investigate a mapping of the cut-out that allows us to find out, if the hypothesis is true or false.
Such a criterion is e.g. (as will be shown) the trace of the covariance matrix.
For this mapping or detection criterion we can obtain characteristical features from analytical calculations as well as Monte Carlo simulation. We will regard the possible situations (models): noise, trajectory alone and noise plus trajectory. For the Monte Carlo simulation we will compose the models mentioned above according to the following:

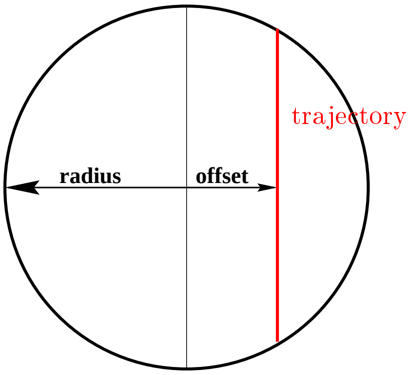

Figure 3: Example of cut-out configuration ‘noise and trajectory’

A local spherical cut-out is generated as shown on the left: firstly, we choose the trajectory position to be a diameter, i.e. the trajectory runs directly through the cut-out’s center (offset 0). Then, the trajectory is moved towards the edge of the sphere while the increment between the correlated data points remains fixed, i.e. with increasing offset there are less correlated data points in the cut-out. In addition to the trajectory we generate random noise within the cut-out. The total number of data points per portion remains fixed. Later, we will also vary some features of these models like position of the trajectory, noisiness of trajectory, number of trajectories, ratio of correlated data to noise etc..

3.2 Analytical models for the mapping *Trace of the covariance matrix*

In the following we will find some characteristics for the criterion *trace of the covariance matrix* (which is equal to the sum of all eigenvalues of the covariance matrix) through some analytical calculations.

3.2.1 Modeling the noise

We regard the noise in a local cut-out as a random sample of a rectangularly distributed random vector (X, Y, T).

Firstly, we investigate a single random vector.

The first order moment of a function of random numbers $g(X,Y,T)$ is the expectation value

$$E(g(X,Y,T)) = \int\int\int g(x,y,t)\, f(x,y,t)\, \mathrm{d}x\, \mathrm{d}y\, \mathrm{d}t \ , \tag{3.1}$$

where $f(x,y,t)$ is the common probability density function. In our case of a random vector (X,Y,T) that is rectangularly distributed within a sphere of radius R_{sphere}, this function is

$$f(x,y,t) = \left(\frac{4}{3}\pi R^3_{sphere}\right)^{-1} = \frac{1}{V_{sphere}} \ . \tag{3.2}$$

Due to reasons of symmetry $(E(X), E(Y), E(T))$ is the sphere's center, thus zero in the sphere's co-ordinate system.
The central second order moments with respect to the sphere's center are

$$\sigma_i^2 = E(X_i^2) \qquad \sigma_{ij} = E(X_i X_j) \tag{3.3}$$

Where σ_X^2 is the variance of the random number X (Y, T respectively) and σ_{XY} the covariance of X and Y (all other combinations of X, Y, T respectively) and

$$\sigma_{XX} = \sigma_X^2 \qquad \sigma_{XY} = \sigma_{YX} \tag{3.4}$$

The matrix

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots \\ \sigma_{21} & \sigma_2^2 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

is the covariance matrix.

For easier calculation we switch to a representation in polar co-ordinates and calculate the covariance matrix.
With radius vector r, geographical length φ $(-\pi < \varphi \leq \pi)$ and polar distance ϑ $(0 \leq \vartheta \leq \pi)$ the co-ordinates and volume element are

$$\begin{aligned} x &= r \sin\vartheta \cos\varphi \qquad (3.5) \\ y &= r \sin\vartheta \sin\varphi \\ t &= r \cos\vartheta \\ dV &= dx\ dy\ dt = r^2 \cdot \sin\vartheta\, \mathrm{d}r\, \mathrm{d}\vartheta\, \mathrm{d}\varphi \end{aligned}$$

Then the diagonal covariance matrix elements are

$$\begin{aligned} E(X^2) &= \frac{1}{V_{sphere}} \cdot \int_0^{R_{sphere}} \int_0^{\pi} \int_{-\pi}^{\pi} r^4 \sin^3\vartheta \cos^2\varphi\, \mathrm{d}r\, \mathrm{d}\vartheta \mathrm{d}\,\varphi \qquad (3.6) \\ &= \frac{1}{5} R^2_{sphere} \end{aligned}$$

and analogous

$$\begin{aligned} E(Y^2) &= \frac{1}{V_{sphere}} \cdot \int_0^{R_{sphere}} \int_0^{\pi} \int_{-\pi}^{\pi} r^4 \sin^3\vartheta \sin^2\varphi\, \mathrm{d}r\, \mathrm{d}\vartheta\, \mathrm{d}\varphi = \frac{1}{5} R^2_{sphere} \qquad (3.7) \\ E(T^2) &= \frac{1}{V_{sphere}} \cdot \int_0^{R_{sphere}} \int_0^{\pi} \int_{-\pi}^{\pi} r^4 \cos^2\vartheta \sin\vartheta\, \mathrm{d}r\, \mathrm{d}\vartheta\, \mathrm{d}\varphi = \frac{1}{5} R^2_{sphere} \ . \qquad (3.8) \end{aligned}$$

The trace of the covariance matrix is

$$E(X^2) + E(Y^2) + E(T^2) = \frac{3}{5} R^2_{sphere} \ . \tag{3.9}$$

The trace can be calculated in a more simple way using polar co-ordinates:

$$E(X^2) + E(Y^2) + E(T^2) = E(R^2) \tag{3.10}$$

$$E(R^2) \ = \ \frac{1}{V_{sphere}} \cdot \int_0^{R_{sphere}} \int_0^{\pi} \int_{-\pi}^{\pi} r^4 \cdot \sin\vartheta \, \mathrm{d}r \, \mathrm{d}\vartheta \, \mathrm{d}\varphi = \frac{3}{5} R^2_{sphere} \ . \tag{3.11}$$

For dispersion $D(X)$ and standard deviation σ_X of a random number X in general we know

$$D(X) = \sigma_X^2 = E(X^2) - (E(X))^2 \tag{3.12}$$

$$D(a \cdot X) = a^2 \cdot D(X) \tag{3.13}$$

and for the random vector in particular

$$D(R^2) = E(R^4) - E(R^2)^2 \ . \tag{3.14}$$

With

$$E(R^4) = \frac{1}{V_{sphere}} \cdot \int_0^{R_{sphere}} \int_0^{\pi} \int_{-\pi}^{\pi} r^6 \cdot \sin\vartheta \, \mathrm{d}r \, \mathrm{d}\vartheta \, \mathrm{d}\varphi = \frac{3}{7} R^4_{sphere} \tag{3.15}$$

dispersion and standard deviation become

$$\begin{aligned} D(R^2) \ &= \ E(R^4) - E(R^2)^2 \\ &= \ \frac{3}{7} R^4_{sphere} - \frac{9}{25} R^4_{sphere} \\ &= \ \frac{12}{175} \cdot R^4_{sphere} \end{aligned} \tag{3.16}$$

and

$$\sigma_{R^2} = \sqrt{D(R^2)} = \sqrt{\frac{12}{175}} \cdot R^2_{sphere} \ . \tag{3.17}$$

We now transgress from one single random vector (X, Y, T) to the sample of rectangularly distributed random vectors (X_i, Y_i, T_i) we call noise.
As statistic we choose the trace of the covariance matrix (*trace*), that results from a sub-sample of N random vectors:

$$\begin{aligned} trace \ &= \ \frac{1}{N} \sum_{i=1}^{N} X_i^2 + \frac{1}{N} \sum_{i=1}^{N} Y_i^2 + \frac{1}{N} \sum_{i=1}^{N} T_i^2 \\ &= \ \frac{1}{N} \sum_{i=1}^{N} R_i^2 \ . \end{aligned} \tag{3.18}$$

The expected value of this statistic is then

$$\begin{aligned} E(trace) &= E\left(\frac{1}{N}\sum_{i=1}^{N} R_i^2\right) \\ &= \frac{1}{N}\sum_{i=1}^{N} E(R_i^2) \\ &= E(R^2) \\ &= \frac{3}{5}R_{sphere}^2 \ , \end{aligned} \tag{3.19}$$

Dispersion and standard deviation of this statistic are

$$\begin{aligned} D(trace) &= D\left(\frac{1}{N}\sum_{i=1}^{N} R_i^2\right) \\ &= \frac{1}{N^2}\cdot\sum_{i=1}^{N} D(R_i^2) \\ &= \frac{1}{N}D(R^2) \\ &= \frac{1}{N}\cdot\frac{12}{175}\cdot R_{sphere}^4 \end{aligned} \tag{3.20}$$

$$\sigma_{trace} = \sqrt{D(trace)} = \frac{1}{\sqrt{N}}\cdot\frac{2}{5}\cdot\sqrt{\frac{3}{7}}\cdot R_{sphere}^2 \ . \tag{3.21}$$

3.2.2 Modeling the signal

We assume a trajectory through the center of a spherical portion. Contrary to the noise vectors which are distributed rectangularly, the data points belonging to this trajectory are not. We adapt the analytical modeling of the trajectory according to the simulative generation described in full detail in Appendix B. The first data point of the trajectory is chosen on the diameter in a way that corresponds to its rectangular distribution on the diameter. All other data points correlated to this first one are obtained by adding a multiple of the three dimensional increment to this first data point. As statistic we chose the trace of the covariance matrix (*trace*).

Let R_1 be the first data point belonging to the trajectory, d the increment between two adjacent correlated data points and $N = 2\cdot k+1$ the number of all data points belonging to the trajectory. Then the position of the correlated data points can be obtained with respect to the first one through

$$P_{2i} = i\cdot d + R_1 \qquad \text{and} \qquad P_{2i+1} = i\cdot d - R_1 \ . \tag{3.22}$$

The trace is then

$$trace = \frac{1}{N}\sum_{i=1}^{N} R_i^2 \tag{3.23}$$

$$\begin{aligned}
&= \frac{1}{N}\left((d^2 \cdot 2 \cdot \sum_{i=1}^{k} i^2) + N \cdot R_1^2\right)\\
&= \frac{1}{N}\left(\frac{1}{6} \cdot k(k+1)(2k+1) + N \cdot R_1^2\right)\\
&= \frac{k(k+1)}{3} d^2 + R_1^2
\end{aligned} \tag{3.24}$$

The expected value for the trace of this statistic is

$$\begin{aligned}
E(trace) &= \frac{k(k+1)}{3} d^2 + E(R_1^2) \qquad (3.25)\\
&= \frac{k(k+1)}{3} d^2 + \frac{1}{3}\left(\frac{d}{2}\right)^2\\
&= \frac{k(k+1) + \frac{1}{4}}{3} \cdot d^2\\
&= \frac{(k+\frac{1}{2})^2}{3} d^2\\
&= \frac{1}{3}\left(\frac{N}{2} d\right)^2
\end{aligned} \tag{3.26}$$

With

$$d = \frac{2R_{sphere}}{N}$$

$$E(trace) = \frac{1}{3} \cdot R_{sphere}^2 \ . \tag{3.27}$$

Thus dispersion $D(trace)$ and standard deviation σ_{trace} are derived to

$$\begin{aligned}
D(trace) &= D(R_1^2) = E((R_1^2)^2) - (E(R_1^2))^2 \qquad (3.28)\\
&= \frac{1}{5}\left(\frac{d}{2}\right)^4 - \left(\frac{1}{3}\left(\frac{d}{2}\right)^2\right)^2\\
&= \frac{4}{45} \cdot \left(\frac{d}{2}\right)^4
\end{aligned} \tag{3.29}$$

and again setting

$$d = \frac{2R_{sphere}}{N}$$

$$D(trace) = \frac{4}{45} \cdot \left(\frac{R_{sphere}}{N}\right)^4 \tag{3.30}$$

$$\sigma_{trace} = \sqrt{D(R^2)} = \sqrt{\frac{4}{45}} \cdot \frac{R_{sphere}^2}{N^2} \ . \tag{3.31}$$

3.2.3 Modeling signal plus noise

Finally, we regard a local cut-out containing noise as well as a trajectory through its center. There are N_C random vectors representing noise and N_T data points belonging to the trajectory, while the total number N of data points is $N = N_C + N_T$. As statistic we choose as before the sum of the diagonal elements of the covariance matrix.

The expected value of this statistic is

$$\begin{aligned} E(trace) &= E\left(\frac{1}{N}\sum_{i=1}^{N_T} R_{T\ i}^2\right) + E\left(\frac{1}{N}\sum_{i=1}^{N_C} R_{C\ i}^2\right) \\ &= \frac{1}{N}\cdot\left(N_T\cdot E(R_T^2) + N_C\cdot E(R_C^2)\right) \\ &= \frac{1}{N_T+N_C}\cdot\left(N_T\cdot\frac{R_{sphere}^2}{3} + N_C\cdot\frac{3}{5}\cdot R_{sphere}^2\right) \\ &= \frac{R_{sphere}^2}{N_T+N_C}\cdot\left(\frac{N_T}{3}+\frac{3}{5}\cdot N_C\right)\ . \end{aligned} \tag{3.32}$$

Thus dispersion $D(trace)$ and standard deviation σ_{trace} are

$$\begin{aligned} D(trace) &= D\left(\frac{1}{N}\sum_{i=1}^{N_T} R_{T\ i}^2 + \frac{1}{N}\sum_{i=1}^{N_C} R_{C\ i}^2\right) \\ &= \frac{R_{sphere}^4}{N^2}\cdot\left(\frac{4}{45}\cdot\frac{1}{N_T^2}+\frac{12}{175}\cdot N_C\right)\ , \end{aligned} \tag{3.33}$$

$$\sigma_{trace} = \sqrt{D(trace)} = \frac{R_{sphere}^2}{N}\cdot\sqrt{\frac{2}{45}}\cdot\sqrt{\frac{4}{45}\cdot\frac{1}{N_T^2}+\frac{12}{175}\cdot N_C}\ . \tag{3.34}$$

We were thus able to prove analytically that the expected value for the trace of the covariance matrix of a cut-out containing exclusively uncorrelated data differs from that of a cut-out containing a certain number of correlated data. This difference depends on the ratio of correlated to uncorrelated data, i.e. the ratio of data points from the trajectory to noise data, as well as the total number of data points per cut-out.
To investigate the quality of this criterion as a verifier of the hypothesis ‘noise’ we will use Monte Carlo simulations as described in the following.

3.3 Monte Carlo Simulation for the mapping *Trace of the covariance matrix*

We will investigate the previously analytically examined criterion *trace of the covariance matrix* which is identical with the *sum of the diagonal elements of the covariance matrix*; in addition we will also examine each eigenvalue individually.
The models are set up like this:

- Noise: The local cut-out is a sphere of radius R_{sphere} and always contains 50 randomly generated noise plots. (cf. Fig. 2 a)).

- Trajectory: Trajectories running through the center of the cut-out are incremented by 3.5 in both spatial directions and by 5 in the third (time) direction. In order not to prefer one of the spatial directions the trajectory is then rotated randomly. The trajectory is also generated slightly noisy; a random value generated within a certain noise radius (2) is added to each data point belonging to the trajectory.
  For trajectories not being a cut-out diameter, the according data points are incremented by 5 in two dimensions (one spatial and time) and are then positioned in the third direction with various offsets through the cut-out. Due to the constant increment of the trajectory data their number is reduced with the offset increasing (cf. Fig. 2 b)(ii)).
  For more detailed information concerning the generation of the trajectory also confer Appendix B.

- Noise and Trajectory: The number of plots per cut-out is always 50. Thus with increasing offset the ratio noise to correlated data increases simultaneously (cf. Fig. 2 b)).

As we originally started the simulation with the prospect of an application to data obtained with a radar system to find trajectories of moving objects therein, the explicit numbers used here are related to situations that occur when processing radar data. Especially the signal to noise ratio has been chosen accordingly. Nevertheless, this should not a restriction but rather an example.

We made simulations of 10^6 cut-outs each for the models ‘noise alone’ and ‘noise plus trajectory’, where for the latter we moved the trajectory through the cut-out as described in Fig. 3. For each cut-out we calculated the three dimensional covariance matrix and its three eigenvalues. Thus we derived distributions of these values.
We then compared the distributions for the models ‘noise alone’ and ‘noise plus trajectory’ and were able to prove the previously calculated difference between them. We also calculated the eigenvalues of the model ‘noise plus trajectory’ for various compositions concerning the position of the trajectory in the cut-out to see how a trajectory can best be detected.
For the cut-outs we choose a radius of 50; every cut-out contains 50 data points altogether, an average of 15 of which are correlated, if the cut-out contains a trajectory through its center.

In the following Fig. 4 a selection from the results for three different offsets is presented: the diagrams show the probability density function of the eigenvalues. Black is the distribution for the model ‘noise alone’, red is the distribution for the model ‘noise and trajectory’. Fig. 4 (left panel) shows the individual eigenvalues, from left to right: smallest, middle and greatest eigenvalue. Fig. 4 (right panel) shows the sum of the eigenvalues. The trajectory’s offset from the cut-out’s center increases top (offset 0) down (offset 10 and 25=$R_{sphere}/2$).
One can clearly see the difference in the distributions belonging to the model ‘noise and trajectory’ (black) and the model ‘noise’ (red) for all criteria, if the trajectory runs through the cut-out’s center, i.e. offset 0.
This difference vanishes slowly with increasing offset, until for offset 25, i.e. $R_{sphere}/2$ the distributions of greatest and middle eigenvalue can no longer be distinguished. Also, the distributions for *sum of the eigenvalues* assimilate with increasing offset.
For even greater offsets from the center (no figure shown) none of the criteria shows a significant

difference between the two models 'noise' and 'noise plus trajectory'.

From these results we can clearly see that the eigenvalues of the covariance matrix as well as their sum is a reliable criterion to distinguish between the two models 'noise' and 'noise plus trajectory', if the trajectory runs through the cut-out's center. Also, they can be of use as trajectory detectors for smaller trajectory offsets up to an offset of $R_{sphere}/2$.

The conclusion we can draw immediately from the decreasing reliability with increasing trajectory offset from the cut-out's center is: while searching a large noise area for trajectories we can detect them best, when the trajectory runs through the cut-out's center. Thus the distance between adjacent cut-outs' centers should not exceed $R_{sphere}/4$ in every direction. This overlap of adjacent cut-out's of at least $7R_{sphere}/4$ in each direction guarantees that every trajectory runs through (or very close to) the center of at least one cut-out. This should ensure best exploitation of the capabilities of these criteria and improve the performance of the algorithm.

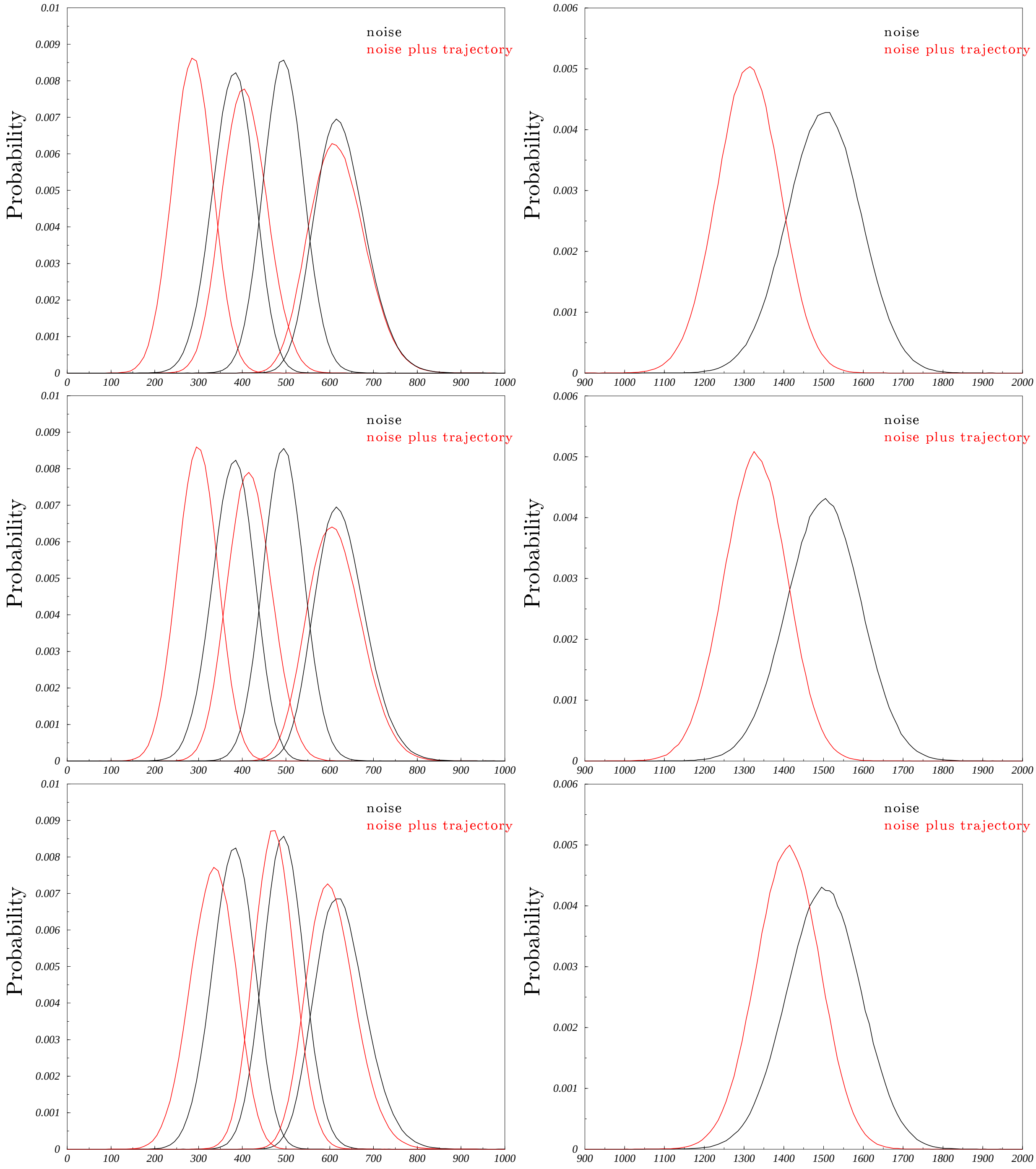

Figure 4: Left: Probability density function obtained with Monte Carlo simulation for the (from left to right) smallest, middle and greatest eigenvalue. Right: Probability density function obtained with Monte Carlo simulation of *sum of all eigenvalues*. The models are 'noise' (black) and 'noise plus trajectory' (red), the trajectory offsets from the center are (top down): 0, 10, 25 (i.e. $R_{sphere}/2$).

3.4 Evaluation of the mappings

We have shown in the previous section that the eigenvalues of the covariance matrix can serve as a trajectory detector. Now we will analyze how well the criterion works under various conditions.

To obtain a measure for the criterion's quality and sufficiency in performance we will take into account the statistical type I and II errors of the distributions introduced in the previous section.

While testing a hypothesis two errors can occur, the type I and II errors which for a hypothesis H_0 are defined as:

1. **Type I error:** Dropping the null hypothesis H_0, although H_0 is correct.
2. **Type II error:** Not dropping the null hypothesis H_0, although H_0 is wrong.

The probability for the type I error is called the **level of significance**, the probability of the type II error is the **power of the test**.

In our case the null hypothesis H_0 is the assumption that the cut-out under discussion contains exclusively noise.
Thus the type I error corresponds to the probability of detecting a non-existing trajectory and the type II error to the probability of not detecting an existing trajectory. Here, the type I error is the false alarm probability (P_{fa}), and the type II error is 1 minus the detection probability (P_d); in the following we will use P_d instead of the type II error.

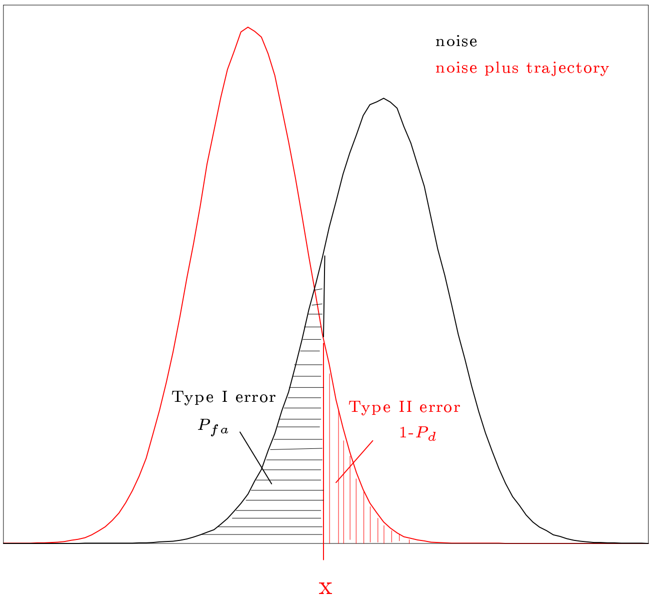

Figure 5: The statistical type I and II errors can be obtained from the probability density distributions and the threshold value x can be set accordingly.

We will also use these errors to fix the threshold described in Fig. 1, that shall be the value against which the calculated mapping from an unknown set of data will be compared in order to make the decision, if the hypothesis is true or false. This threshold is also shown as the value x in Fig. 5, defining the according errors.

With a volume of 10^6 events we generate cut-outs of radius R_{sphere}=50 with and without trajectory for each offset from 0 to 49. We then make a statistic for various mappings like eigenvalues of the covariance matrix, their sum or other combinations. The results corresponding to the probability of the respective event, are being sorted into a histogram. We can then compare the probability density functions of the two models 'noise' and 'noise plus trajectory' by calculating the power of the test for various fixed levels of significance (thus fixing the threshold value) in dependence from the trajectory position in the cut-out.

3.4.1 Analytical calculations of the statistical type I and II errors

As before the criterion *trace of the covariance matrix* allows analytical calculation of the type I and II errors.
To do so, we assume normal distribution of the criterion. This is legitimate because of the central limit theorem: the criterion *trace of the covariance matrix* is the sum of many independent and identically distributed variates.

To prove this assumption we obtain distributions for the criterion through analytical calculation and via Monte Carlo simulation. For the analytical calculation we calculate the expected value and dispersion (first and second order geometric moments) as described in Section 3.2 and use these values to draw a Gaussian curve. The calculations of third and fourth order geometric moments, i.e. skewness and kurtosis are only done for the model 'noise'.

The results are presented in Table 1 and graphically in Fig. 6.

	noise		noise plus trajectory	
	calculated	simulated	calculated	simulated
1^{st} order moment expected value	1500	1500.03	1309.33	1310.52
2^{nd} order moment dispersion	8571.43	8593.24	6121.09	6227.83
standard deviation	92.58	92.70	78.24	78.92
3^{rd} order moment skewness	-0.048	-0.0436	-	-0.0517
4^{th} order moment kurtosis	-0.0166	-0.0145	-	-0.0243

Table 1: Parameters of probability density distribution as calculated analytically and obtained via Monte Carlo simulation.

As can be seen from Table 1, the skewness, a measure for the degree of asymmetry of a distribution around its mean, is almost zero, as expected for a Gaussian distribution. The kurtosis measure for the non-gaussianity, giving the flatness (< 0) or peakedness (> 0) of a distribution, also vanishes. These two values clearly prove the almost gaussianity of the distributions for the models 'noise'.

As can be seen from Fig. 6 the difference between the distribution obtained by Monte Carlo simulation (red) and the normal distribution obtained with analytical values (black) is imperceptible, assuring that the simulation is in total agreement with the calculation.

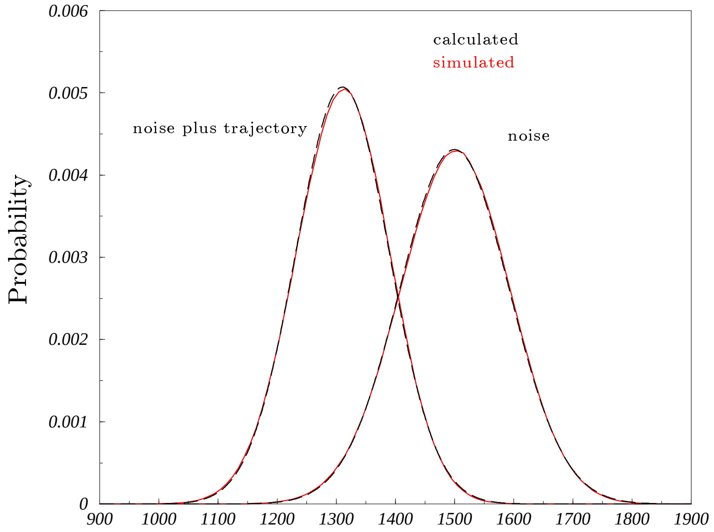

Figure 6: In comparison, the distributions obtained via analytical calculation (black) and Monte Carlo simulation (red) for the models 'noise' (right) and 'noise plus trajectory' (left); criterion is the *sum of all eigenvalues*.

As a next step we obtain the type II error for some levels of significance analytically. As the distribution function

$$\phi(x) = \int_{-\infty}^{x} g(u)\mathrm{d}u$$

cannot be dealt with analytically, we have to use tabulated values to do so.

For the normal distribution $N(E, \sigma)$ we know the probability density function to be

$$g(x) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{1}{2} \cdot \left(\frac{x-E}{\sigma}\right)^2} . \tag{3.35}$$

Expected value E and standard deviation σ have been calculated in Section 3.2 for the models 'noise' and 'noise plus trajectory' (offset 0) depending on the cut-out's radius and the number of data points. Therefore we know ϕ_{noise} and $\phi_{noise+trajectory}$.

The type I and II errors are tabulated for the standardized normal distribution $N(0, 1)$ (cf. e.g. [Bron1977]). By means of a simple transformation we can standardize the normal distribution $N(E, \sigma)$:

$$\begin{aligned} g(x') &= \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{x'^2}{2}} \\ x' &= \frac{x-E}{\sigma} \\ x &= x' \cdot \sigma + E\ . \end{aligned} \tag{3.36}$$

The type I error or false alarm probability of 10% corresponds to 10% of the area under the distribution for the model 'noise' (c.f. Fig. 5).

$$\begin{aligned} P_{fa} &= \int_{-\infty}^{x} g_{noise}(u)\mathrm{d}u = \phi_{noise}(x) \\ \Longrightarrow x &= \phi_{noise}^{-1}(P_{fa}) \end{aligned} \tag{3.37}$$

We are looking for

$$x' = \phi_{noise}^{-1}(10\%)\ .$$

The value for x that marks the point where 10% of the area under the standardized normal distribution are reached can be looked up (cf. e.g. [Bron1977]). With the transformation given in equation 3.36 the corresponding x' of the $N(E,\sigma)$-distribution can be obtained.
The detection probability P_d=1-type II error can be obtained from the distribution for 'noise plus trajectory'

$$P_d = \int_{-\infty}^{x} g_{noise+trajectory}(u)\mathrm{d}u = \phi_{noise+trajectory}(x) \tag{3.38}$$

$$\Longrightarrow P_d = \phi_{noise+trajectory}\left(\phi_{noise}^{-1}(P_{fa})\right) \tag{3.39}$$

For this value x we now determine via equation 3.36 the respective x' for the distribution belonging to the model 'noise plus trajectory' and look up, which area under the distribution this value corresponds to. The so obtained value is the probability for the type II error.

For the actual calculation we use the equations for expected value and standard deviation from Section 3.2 as well as the following values: R_{sphere}=50, N_C=50 noise data points or N_C=35 noise data and N_T=15 trajectory data points. Thus

Level of significance P_{fa}	5%	10%	15%	20%
Power of the test P_d	29,8% 70,2%	18,1% 81,9%	12,3 % 87,7%	8,53% 91,47%

Table 2: Detection Probability P_d for several false alarm rates P_{fa}; criterion is the *sum of all eigenvalues*

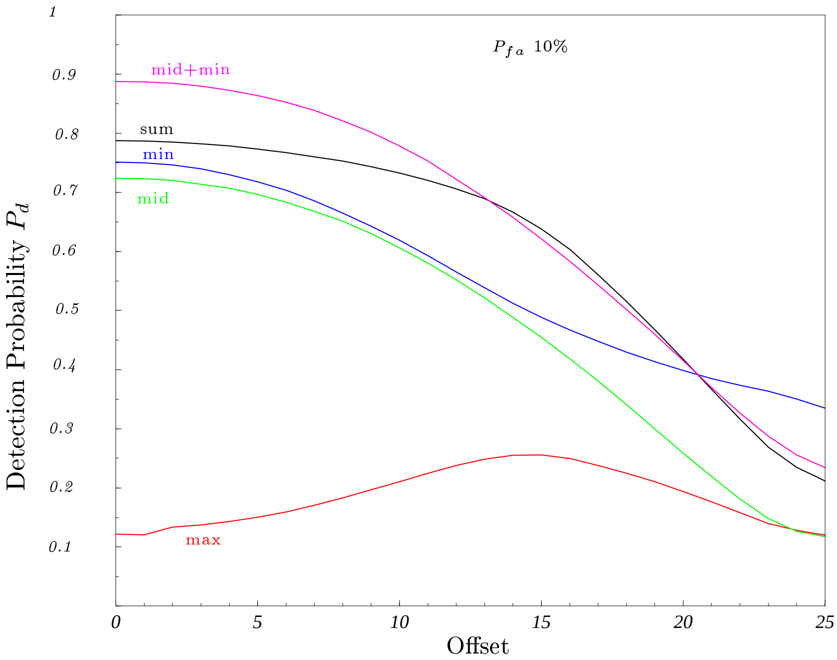

Figure 7: The detection probability P_d against the offset of the trajectory in the cut-out for a constant false alarm rate P_{fa} of 10%. Criteria are *sum of all eigenvalues* (black), *greatest* (red), *middle*, *smallest* (blue), *sum of middle and smallest* (magenta) eigenvalue.

3.4.2 The statistical type I and II errors obtained via Monte Carlo Simulation

Based upon the values from Table 2 we choose a threshold value x corresponding to a significance level for the statistical test of 10%.
To obtain the type I and II errors from Monte Carlo simulation the assumption of normal distribution that was necessary for the analytical calculation is obsolete. The exact calculation of the area and errors respectively is performed on the probability density function in terms of a histogram.

Firstly, we investigate the behaviour of all criteria *trace of the covariance matrix* and *eigenvalues of covariance matrix* individually for a constant false alarm probability P_{fa} of 10%. Taken into consideration the so obtained results we also calculate *sum of smallest and middle eigenvalue.*
As can be seen from Fig. 7 the criterion *sum of smallest and middle eigenvalue* proves to be the best detection criterion with 88% detection probability for a trajectory with offset 0. The next best criteria are *sum of all eigenvalues* (P_d 78%), *smallest* (P_d 75%) and *middle eigenvalue* (P_d 72.5%). The *greatest eigenvalue* turns out to be a very bad detection criterion (P_d 12%). Apart from the latter, the detection probability decreases for all criteria with increasing offset of the trajectory.

3.4.3 Evaluation of mappings by means of the Receiver Operating Characteristics

To summarize the results obtained until now we make use of a common graphical method to document the quality and applicability of the criteria: we plot the detection probability P_d

against the false alarm probability P_{fa} to obtain **Receiver Operating Characteristics** for each criterion.

These ROCs are commonly used to describe the performance of detection systems of any kind. The ideal receiver or detector would have a single point as ROC: P_d=1, P_{fa}=0. For real detection systems the ROC has a typical form as shown in Fig. 8: with increasing false alarm probability the detection probability increases also. This means, that in real detection systems, there is no such point where maximum detection probability coincides with minimum false alarm probability. One has to find an acceptable compromise between the two, as drawn in Fig. 8: in this example we would choose the detection threshold in a way that fixes P_{fa} to 10% and then deal with the P_d of 81%. We would thus only detect 81% of the existing trajectories and the probability of claiming to have found a trajectory where there is none is 10%.

To say, which of the errors type I or type II is the more important one, or which one to minimize first is highly dependant on the application.

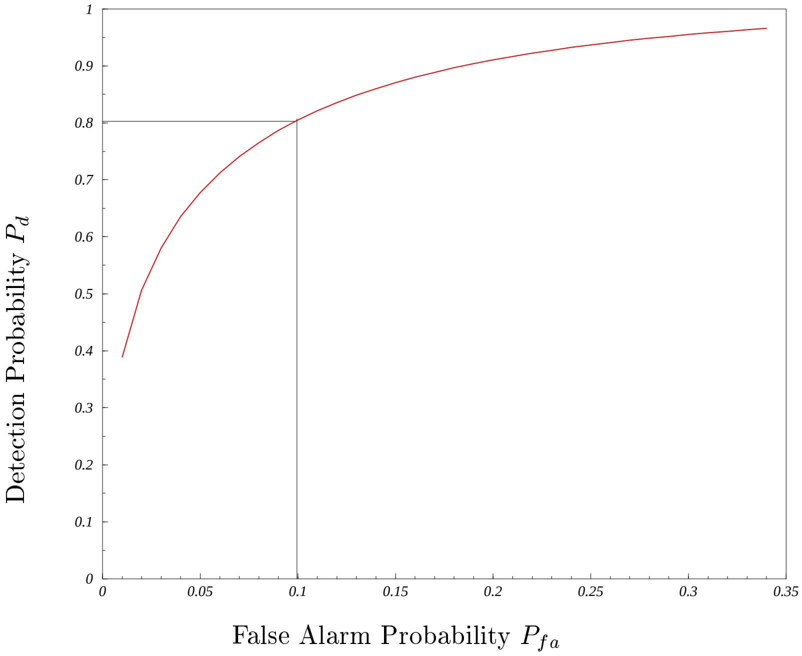

Figure 8: Example of a **R**eceiver **O**perating **C**haracteristic

In the following figures Fig. 9, 10, 11 we show the Receiver Operating Characteristics for the criteria previously investigated on: smallest and middle eigenvalue of covariance matrix, their sum and the sum of all eigenvalues. We do not show the greatest eigenvalue, as it is not a sufficient criterion as could be seen from Fig. 4 and Fig. 7. We also present results for various trajectory positions in the cut-out. The highest P_{fa} we considered is 35%, anything beyond this value is definitely not suitable for any detection system.

As can be seen from Fig. 9 the criterion performing best is the *sum of middle and smallest eigenvalue*: for a P_{fa} of 10% it shows P_d 87.5%, if the trajectory runs diagonally through the cut-out. Second best criterion is the *sum of all eigenvalues* with P_d 80% at P_{fa} 10%, followed by the *smallest* and *middle eigenvalue* with P_d of 74% and 71% respectively.

The detection probabilities for all offsets at P_{fa} 10% are listed in Table 3.

P_d for P_{fa} 10%	offset 0	offset $R_{sphere}/5$	offset $rad/2$
sum of smallest and middle eigenvalue	87.5%	77.5%	24%
sum of all eigenvalues	80%	71%	23%
smallest eigenvalue	74%	61%	32.5%
middle eigenvalue	71%	59%	13%

Table 3: The detection probability of all eigenvalue criteria in dependence on several offsets for a constant false alarm rate of 10%.

As can be seen from Table 3 above, all criteria perform best for a trajectory through the cut-out's center and decrease in detection ability the farther the trajectory is moved towards the cut-out's edge. Also, the farther the trajectory is moved away from the center, the less plots the trajectory contains.
So the conclusion we can draw is like before, that while processing a data set the cut-outs will need to overlap. Also, we will have to investigate the performance for various signal to noise ratios.

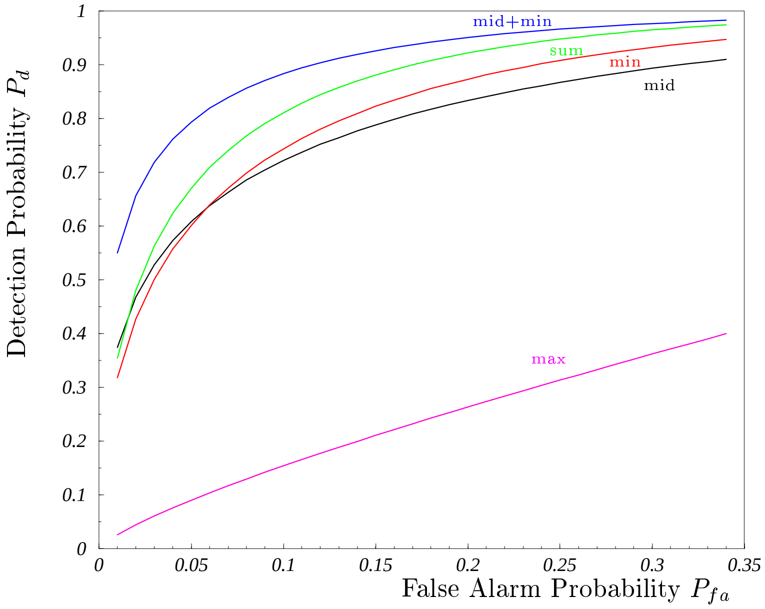

Figure 9: Operating Characteristic for a trajectory with offset 0 to the cut-out's center. Mappings are *sum of smallest and middle eigenvalue* (blue), *sum of all eigenvalues* (green), *smallest* (red), *middle* (black) and *largest* (magenta) eigenvalue.

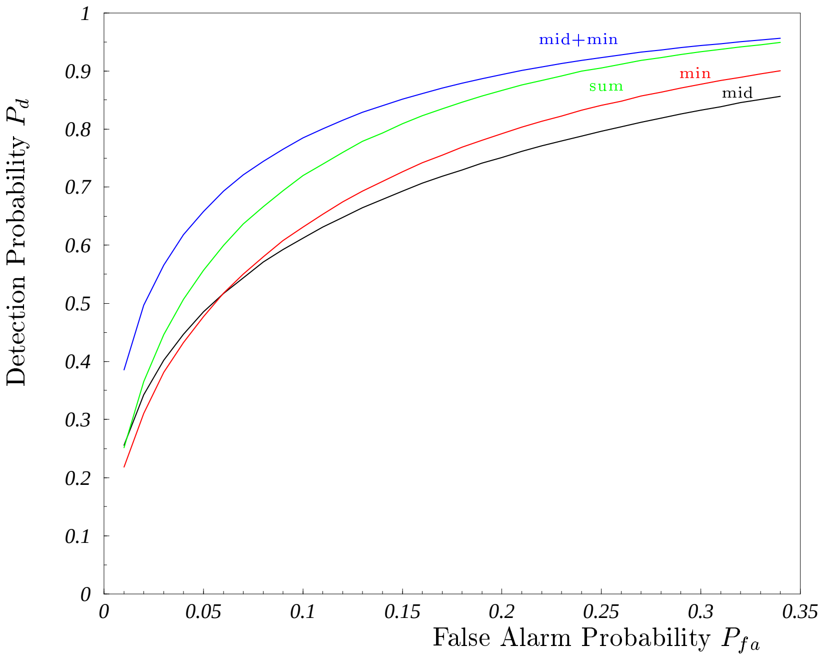

Figure 10: Operating Characteristic for an offset of 20% of the cut-out's radius. Mappings are *sum of smallest and middle eigenvalue* (blue), *sum of all eigenvalues* (green), *smallest* (red) and *middle* (black) eigenvalue.

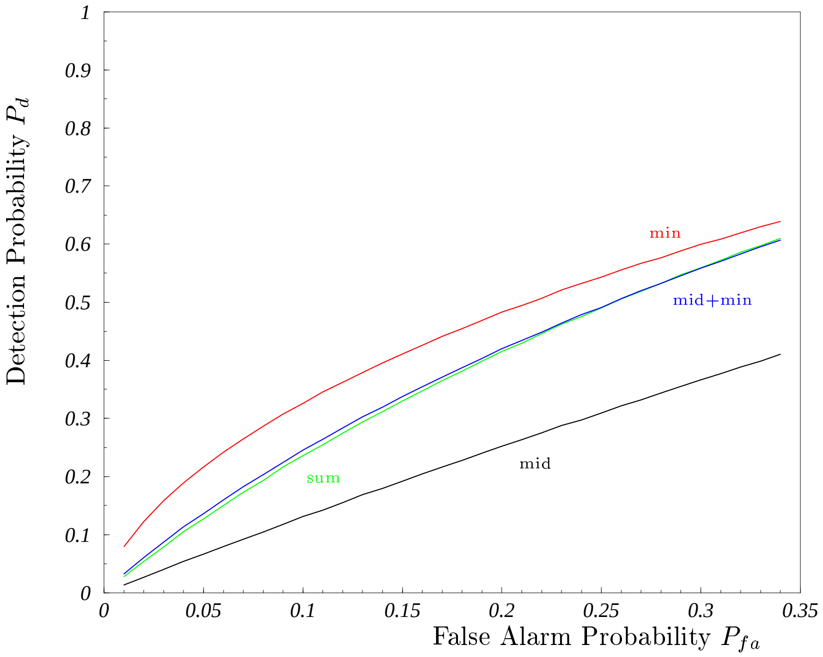

Figure 11: Operating Characteristic for an offset of 50% of the cut-out's radius. Mappings are *sum of smallest and middle eigenvalue* (blue), *sum of all eigenvalues* (green), *smallest* (red) and *middle* (black) eigenvalue.

3.5 Evaluation of the mappings for various cut-out configurations

Hitherto we have only investigated cut-outs with one trajectory consisting of a fixed number of trajectory plots.

In the following we will create two additional models for cut-outs:

1. Cut-out with two parallel trajectories at various distances to each other.
2. Cut-out with two crossing trajectories under various angles to each other.

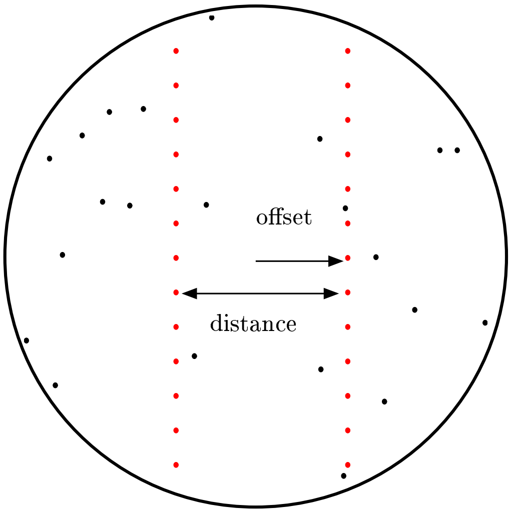

Figure 12: Example of a cut-out containing two parallel trajectories at distance 2×offset to each other.

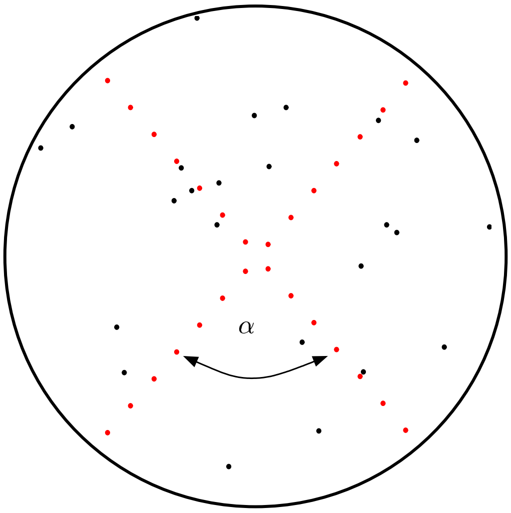

Figure 13: Example of a cut-out containing two trajectories crossing at an angle α.

3.5.1 Cut-out containing two parallel trajectories

To investigate the performance of some criteria in case that a cut-out contains two parallel trajectories, we model cut-outs with two trajectories of 15 correlated and 35 uncorrelated data points (offset 0). The configuration is shown in Fig. 12. We consider two parallel trajectories being moved away from each other. In dependence from the trajectories' distance in the cut-out we investigate the detection probability for a constant false alarm rate of 10%. We choose the best eigenvalue criteria, which are *sum of all eigenvalues* and *sum of smallest and middle eigenvalue* .

The result is shown in Fig. 14. As expected for small distances two trajectories actually are treated in the way as would be one trajectory consisting of twice as many correlated data. As shown in the previous section, the greater the ratio correlated to uncorrelated data the higher the detection probability. In this case the detection probability even increases to 1 for small offsets of $10 - 15$ for each trajectory (note: the trajectories' offset is half their distance).

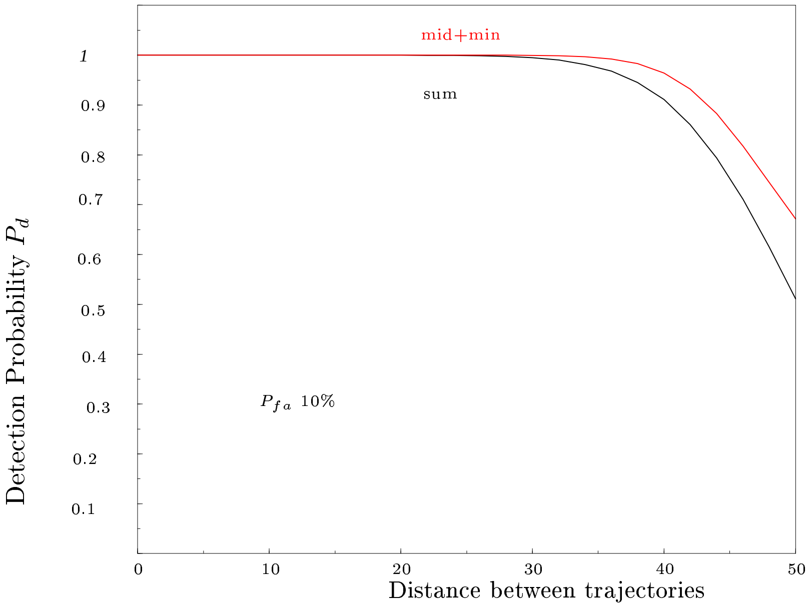

Figure 14: The detection probability P_d against the distance of two parallel trajectories in a cut-out for the mappings *sum of all eigenvalues* (black) and *sum of smallest and middle eigenvalue* (red) the graphs of which are identical for distances of 0 to 20..

3.5.2 Cut-out containing two crossing trajectories

To investigate the performance of some criteria in case that a cut-out contains two crossing trajectories, we model cut-outs with two trajectories of 15 correlated and 35 uncorrelated data points (offset 0). The configuration is shown in Fig. 13. We generate the trajectories crossing under various angles to each other and investigate the detection probability for a constant false alarm rate of 10%. We choose the best eigenvalue criteria, which are *sum of all eigenvalues* and *sum of smallest and middle eigenvalue.* The result is shown in Fig. 15.

As expected for all angles two trajectories are being treated as would be one trajectory containing twice as many correlated data, when the eigenvalue-criteria are applied (*sum of smallest and middle eigenvalue* results in the same graph as *sum of all eigenvalues* and is thus not shown in the diagram); thus the detection probability is 1 for all angles.

3.5.3 Comparison of the geometric and central moments

Hitherto we have exclusively calculated the covariance matrix with respect to the cut-outs' geometrical center (geom.c.), thus using geometric moments. We will now see, whether the covariance calculated with respect to the cut-outs' center of momentum (central moments) can improve the sufficiency of the criterion. For this test we use the criterion *sum of all eigenvalues* and compare the two cases. The result is presented in Fig. 16.

As can be seen, the detection probability is slightly larger for smaller offsets, if we use the geometric moments. But at the same time the detection probability decreases more rapidly

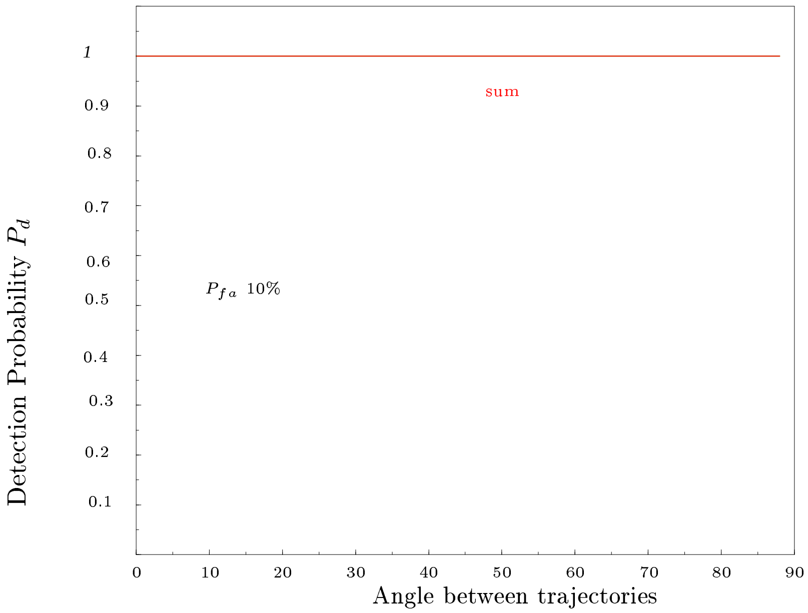

Figure 15: The detection probability P_d against the angle between two crossing trajectories for the mappings *sum of all eigenvalues* and *sum of smallest and middle eigenvalue* which both result in the same graph as shown above.

with increasing offset. Overall, the resulting difference between both ways of calculating the covariance matrix is quite significant in the important region of offsets 0 to 15. We thus choose the method performing better and calculate the moments with respect to the geometrical center.

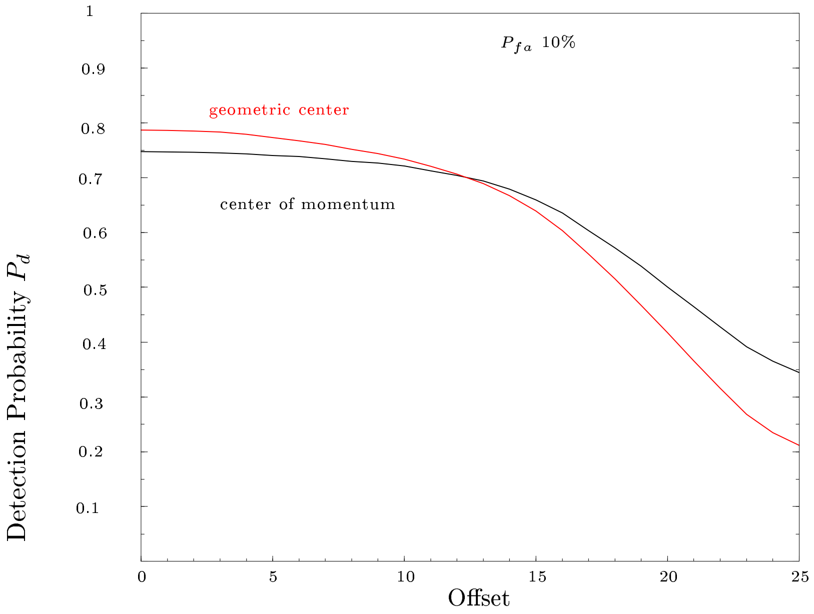

Figure 16: The detection probability P_d against the offset of the trajectory in a cut-out for a constant P_{fa} of 10%. Mapping is the *sum of all eigenvalues*. The covariance matrix was calculated with respect to the cut-out's center of momentum (black) and geometrical center (red).

3.5.4 Adding noise to the trajectory data points

During observations all data are subject to observational disturbances. The data quality is mainly influenced by the intrinsic system limitations and inaccuracies in the data processing chain. As a result, data points are only exact up to a certain accuracy. This means, that observed trajectories of linearly moving objects , are not exactly linear but disturbed to some degree.

We will now examine how the criteria *sum of all eigenvalues* and *sum of smallest and middle eigenvalue* behave under various levels of noise and also add noise to the correlated data.
The disturbance in linearity of motion influences the increment of the noisy correlated data: it is no longer constant in the three dimensions. To imitate this disturbance, we add to each trajectory vector a random vector that has been generated within a certain radius, the noise radius (cf. Fig. 17). We let the noise radius vary between 2% and 10% of the cut-out's radius. Thus the increment between the trajectory vectors also varies slightly.
For a detailed description of the generation of the trajectory and the noise also confer Appendix B.

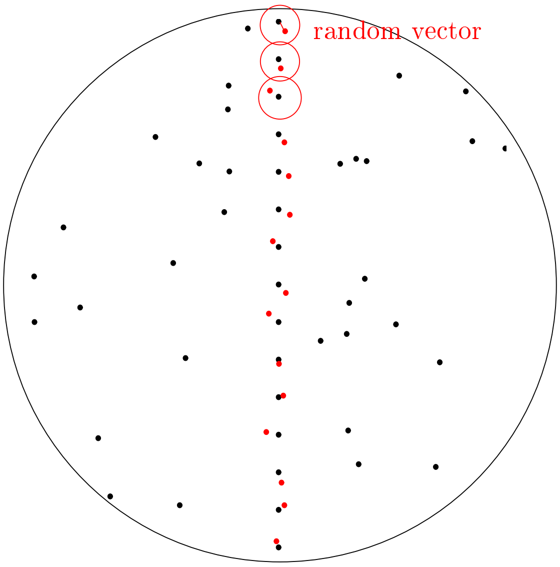

Figure 17: In a noisy trajectory a random vector r drawn in a sphere of radius *noiseradius* is added to each data point to form the noisy trajectory as drawn in red. The degree of noisiness depends on the *noiseradius* indicated by the red circles; the larger it is the more noise is added to the plots.

We investigate on noise radii of 0 (undisturbed trajectory), 2, 5, and 10 (corresponding to 0%, 4%, 10% and 20% of the cut-out's radius). In Fig. 18 and Fig. 19 one can see that with increasing noise radius the detection probability decreases only slightly. Over all, one can say that the criteria both remain reliable even if the trajectory is heavily disturbed.

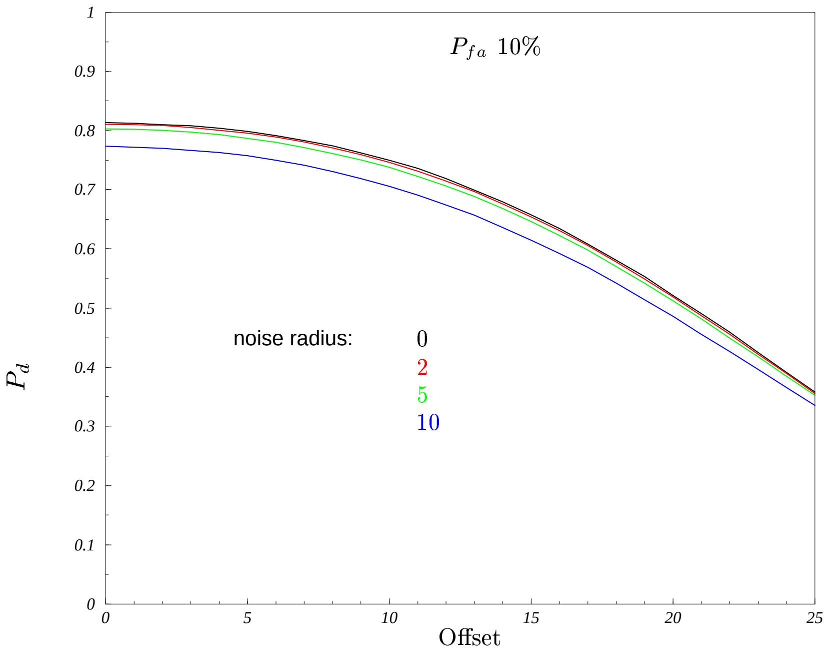

Figure 18: The detection probability P_d against the offset of the trajectory in a cut-out for a constant P_{fa} 10%. Mapping is the *sum of all eigenvalues*; noise radii are 0 (black), 2 (red), 5 (green) and 10 (blue).

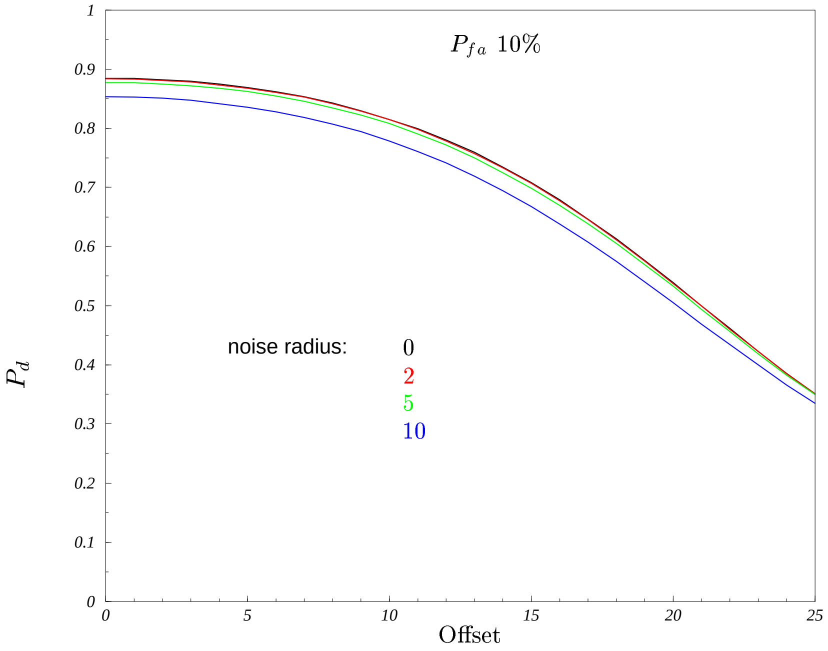

Figure 19: The detection probability P_d against the offset of the trajectory in a cut-out for a constant P_{fa} of 10%. Mapping is the *sum of smallest and middle eigenvalue*; noise radii are 0 (black), 2 (red), 5 (green), 10 (blue).

3.5.5 detection of a trajectory consisting of less data points

In order to investigate the performance of the two best criteria in case the trajectory contains less than 15 data points, we model a cut-out with 8 correlated and 42 uncorrelated data points. In dependence on the trajectory offset in the cut-out we investigate the detection probability for a constant false alarm rate of 10%. We chose the best eigenvalue criteria, which are *sum of all eigenvalues* and *sum of smallest and middle eigenvalue.* We expect a worsening in performance: the eigenvalues are used as a measure for the nature of the local data distribution. For calculating the covariance matrix, each data co-ordinate is used once. Hence, the more data points divert from a random distribution the greater is the difference between the eigenvalues for cut-outs containing noise and noise plus trajectory. The result is shown in Fig. 20.

As one can see very clearly the performance of the criteria has decreased significantly: for a trajectory through the center of the cut-out the detection probability is less than 50% for the best criterion and decreases heavily with growing trajectory-offset from the center.

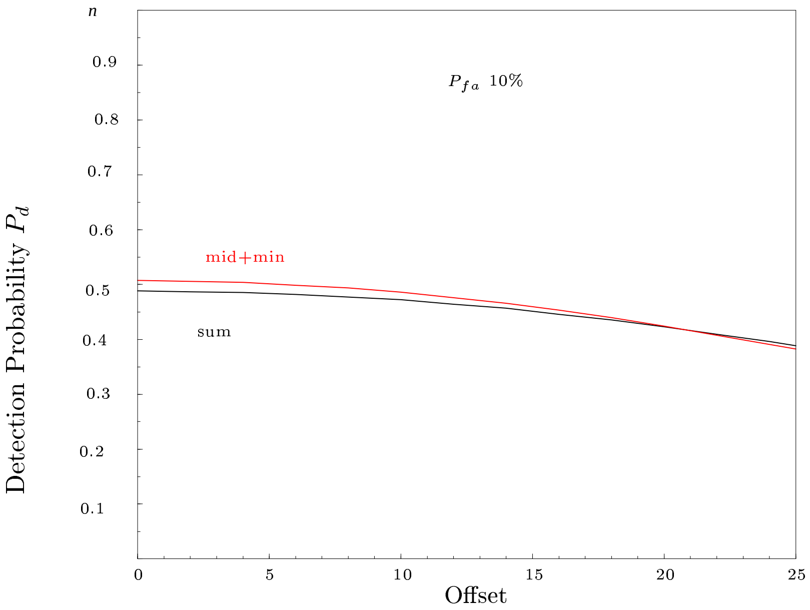

Figure 20: The detection probability P_d against the offset of a trajectory consisting of 8 correlated data points in a cut-out with 42 noise data points for the mappings *sum of all eigenvalues* (black) and *sum of smallest and middle eigenvalue* (red).

For the application to real data this result is utterly important. We will have to investigate whether there is an optimum ratio of correlated to uncorrelated data, which could stabilize or even improve the performance of the criteria.

3.5.6 Varying the signal to noise ratio

As seen while considering trajectories at various offsets from the cut-outs' center, another factor to be investigated is the ratio of correlated to uncorrelated data. As noise is not globally homogeneously distributed and the noise density may not be constant over the entire data set, we also have to examine this possible detraction.

For that reason we generate various numbers of noise data for a fixed number of correlated data. We choose a total number of 35, 50 (as before), 70 and 100 data per cut-out; for a trajectory through the cut-out's geometric center (offset 0) the number of data points is 15 and the noise radius is 2.
In Fig. 21 we present the results for the mapping *sum of all eigenvalues.* The criterion worsens relevantly with increasing noise density.

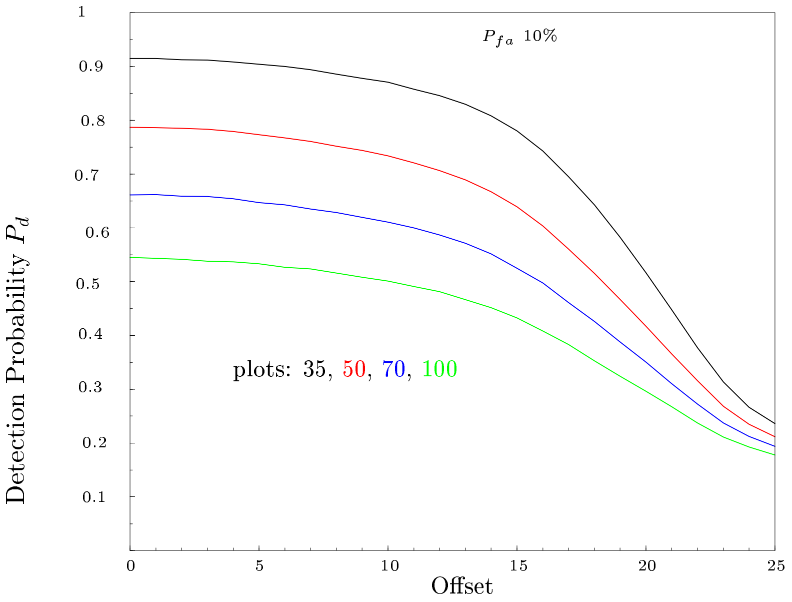

Figure 21: The detection probability P_d against the offset of a trajectory in a cut-out for a constant false alarm probability P_{fa} of 10%. Mapping is the *sum of all eigenvalues.* The total number of data points is 35 (black), 50 (red), 70 (blue) and 100 (green).

We also investigate trajectories through the center of the cut-outs with an average of 15 data points while we vary the number of noise data added from 20 to 100. Another way of illustrating the performance of the criteria is shown in Fig. 22 and Fig. 23: we obtained expected value and standard deviation of the probability density distributions for both cases noise and noise plus trajectory. We present the expected values in dependence of the total number of data points per cut-out. The error bars to the expected value represent the standard deviation of the distributions. The quality of the criteria can be estimated by the distance of the two curves: the farther they are apart the more suitable the criterion as a distinguisher between noise and noise plus trajectory. As can be seen, all criteria worsen with increasing noise.

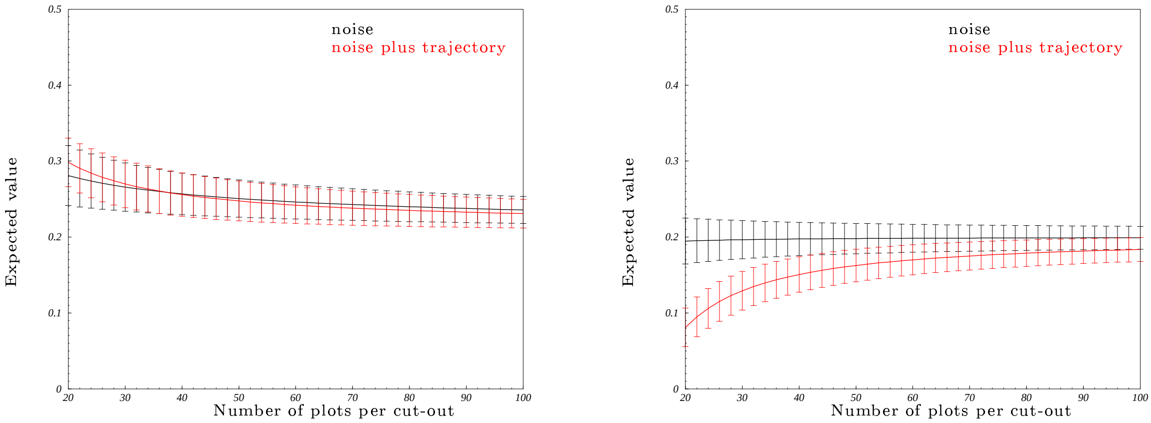

Figure 22: The expected values of the probability density distributions for the eigenvalue criteria in dependence of the plot number per cut-out for the model ‘noise plus trajectory’ (red) and ‘noise’ (black). Left: mapping is the *greatest eigenvalue of covariance matrix*; right: mapping is the *middle eigenvalue of covariance matrix.*

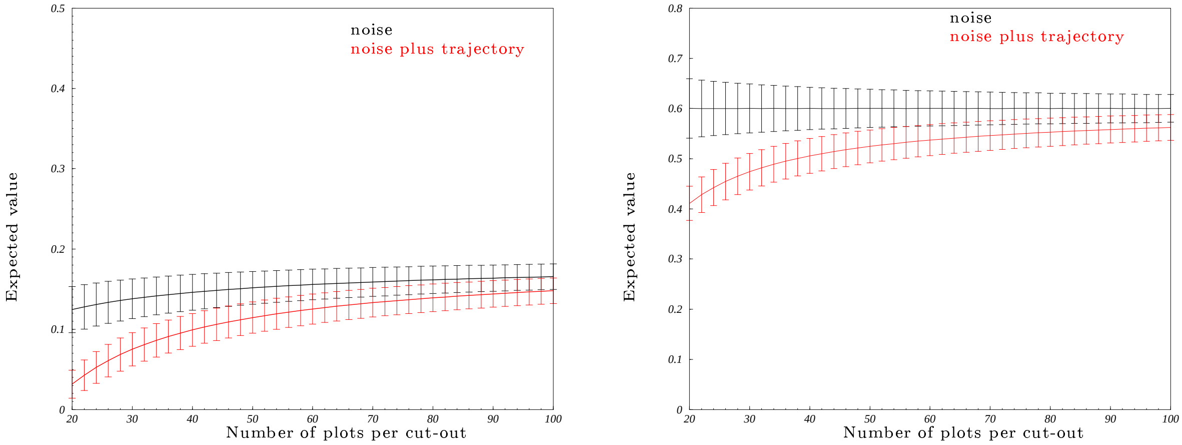

Figure 23: The expected values of the probability density distributions for the eigenvalue criteria in dependence of the plot number per cut-out for the model ‘noise plus trajectory’ (red) and ‘noise’ (black). Left: mapping is the *smallest eigenvalue of covariance matrix*; right: mapping is the *sum of all eigenvalues of the covariance matrix.*

Considering these results, it might be of advantage for an application to set an upper limit to the number of data points per cut-out. (Of course, the number of data points has also a lower limit, as we require a minimum number of correlated data to detect it.)

3.6 Derivation of a weighted combination of the eigenvalue mappings

We have now investigated the performance of some criteria, each showing advantages and disadvantages. We have stuck to the *sum of all eigenvalues* and the *sum of smallest and middle eigenvalue*, which until now are the best criteria. In order to improve the performance of the criteria, we now try to find the optimum combination of all three eigenvalues.
We have also tried to improve the performance by using logical combinations of the eigenvalue criteria, such as:

$$\text{if } \; a \cdot max > x \;\wedge\; b \cdot mid > y \;\wedge\; c \cdot min > z \; \text{ then decide for noise}$$

or

$$\text{if } \; a \cdot mid > x \;\vee\; b \cdot min > y \; \text{ then decide for noise}$$

without achieving a major improvement. We thus try a multidimensional optimization forming a weighted combination of the single eigenvalue mappings.

So far, we have regarded the eigenvalues in terms of $1 \cdot max + 1 \cdot mid + 1 \cdot min$ or $1 \cdot mid + 1 \cdot min$ and investigated their performance weighting them all with the factor 1.
Taking into account that the smallest and middle eigenvalue are better criteria than the largest, but in order not to enounce the largest eigenvalue entirely, it is reasonable to weight each eigenvalue individually. We will thus regard the combination of eigenvalues in terms of

$$\alpha \cdot max + \beta \cdot mid + \gamma \cdot min \; .$$

To find the best combination means to find values for the parameters α, β and γ, so that the false alarm probability is minimized and the detection probability is maximized at the same time.
As we use the receiver operating characteristic to measure the quality of a criterion we want to obtain the detection probability for fixed false alarm rates. This means, we have to obtain the detection probability P_d as a function of α, β, γ for a certain the false alarm probability P_{fa}:

$$P_d = f(\alpha, \beta, \gamma) \; .$$

This means, for any given P_{fa} we search for α, β, γ so that

$$P_d = f(\alpha, \beta, \gamma) \overset{!}{=} Max \; ,$$

or as the type II error is 1-P_d

$$TypeII = 1 - f(\alpha, \beta, \gamma,) \overset{!}{=} Min \; .$$

In order to do so we use the **Downhill Simplex Method** (cf. Appendix C), a common minimization/maximization method often used, if neither the function itself nor its derivates are known.

First of all, we generate the data. This is done by simulating 1.) 10^7 cut-outs containing noise and 2.) 10^7 cut-outs containing noise and a trajectory through their center like before. Then for each cut-out the three eigenvalues are calculated and stored in two data-arrays (as sketched in Tab. 4):

1. max, mid, min for noise
2. max, mid, min for noise plus trajectory

cut-out no.	1	2	3	...	10^7
	$\max_1$	$\max_2$	$\max_3$	...	$\max_{10^7}$
	mid_1	mid_2	mid_3	...	mid_{10^7}
	$\min_1$	$\min_2$	$\min_3$	...	$\min_{10^7}$

Table 4: Example of the required data array.

We can then obtain for any combination of α, β, γ

from 1.: $\mathrm{P}_{fa}(\alpha, \beta, \gamma)$ false alarm probability

from 2.: $\mathrm{P}_{d}(\alpha, \beta, \gamma)$ detection probability

We investigate the eigenvalues of each cut-out and decide for 'noise plus trajectory', if

$$\alpha \cdot min + \beta \cdot mid + \gamma \cdot max < 1.0 \tag{3.40}$$

For any given combination of α, β, γ we check equation 3.40 for every cut-out in data array 1. We then count the numbers of times we decide for 'noise plus trajectory' and because we generated all cut-outs in this array exclusively with noise, this number is the number of false alarms. Its ratio to the total number of cut-outs in the array gives us the false alarm probability $P_{fa}(\alpha, \beta, \gamma)$.
Accordingly for the same combination of α, β, γ we check equation 3.40 for data array 2. We count the number of times we decided correctly for 'noise plus trajectory' and set this number in relation to the total number of cut-outs in the array. This ratio is then the detection probability $P_d(\alpha, \beta, \gamma)$.
We can thus obtain values of the function $P_d = f(\alpha, \beta, \gamma)$ from the two data arrays.

We have to maximize a function $f(\alpha, \beta, \gamma)$, that we do not know explicitly, but whose function values we can obtain as explained above. Especially, we want to maximize it under the condition, that we do so for a chosen P_{fa}. This means, for a certain P_{fa} we want

$$P_d(\alpha, \beta, \gamma) = P_d^*(\alpha, \beta, \gamma) - a \cdot |P_{fa}^*(\alpha, \beta, \gamma) - P_{fa}| \overset{!}{=} Max \; . \tag{3.41}$$

We choose a value for P_{fa}, try an α, β, γ combination and obtain $P_d^*(\alpha, \beta, \gamma)$ and $P_{fa}^*(\alpha, \beta, \gamma)$ as described below. If for this specific parameter combination the calculated P_{fa}^* deviates much from the chosen P_{fa}, the calculated P_d^* is worsened by a $a \times |P_{fa}^* - P_{fa}|$, where a is a factor influencing the strictness of the 'punishment'. So on its search for the maximum we simultaneously force the algorithm to achieve a certain false alarm probability P_{fa}.
The actual function we do want to maximize is hence equation 3.41.

To do so we now start the Downhill Simplex Method.
As an input the algorithm is given a so-called *simplex*, which is in n dimensions a geometric

figure consisting of $n + 1$ points. In this case it would be a tetrahedron. For this input simplex we choose four α, β, γ combinations and obtain the P_{fa} as well as the function value. From then on the algorithm finds its own way along the function on the search for the global minimum.
A detailed description of the Downhill Simplex Method and the way we applied it can be found in Appendix C.

As this method weights all eigenvalues independently it will perform better or at least not worse than the *sum of all eigenvalues* and also than the *sum of smallest and middle eigenvalue.*

The resulting parameters and results are shown in Table 5. As can be seen, the largest eigenvalue has been weighted the least, as it was proven to be the worst single criterion of all eigenvalues. But, one can also notice that with increasing P_{fa} the weight of the largest eigenvalue also increases. Conferring the operating characteristic of the eigenvalues as shown in Fig. 9, one can see, that in the region of P_{fa} 15-35% the P_d still increases in the same way for the largest eigenvalue, whereas for the other two eigenvalues it increases only very slightly. Thus it is sensible to put more weight on the largest eigenvalue for a higher P_{fa}.

The receiver operating characteristic is shown in Fig. 24.
As can be seen, the parameter combination weighting the eigenvalues individually is slightly but clearly better than our two best criteria so far.
For a false alarm probability of 10% it achieves a detection probability of 89.96%, which is 2.5% better than the criterion *sum of smallest and middle eigenvalue* performs.

This also indicates, how important the individual weighting is while searching for the optimum combination of several single criteria.

3.7 Investigations on other mappings independent from the eigenvalue mappings

Apart from the eigenvalues of the covariance matrix and their weighted combination we have found additional mappings suitable to distinguish between the cut-out models ‘noise’ and ‘signal plus noise’. These criteria that will be introduced and investigated in the following are independent from the covariance matrix. They will thus behave in a different way and can be used even when the eigenvalue mappings fail to work.

3.7.1 Mapping *Dispersion of cosine*

As another mapping of a cut-out capable of distinguishing between the two models

1. The cut-out contains only noise.

2. The cut-out contains noise and trajectory.

we try to make use of the angle between e.g. the time axis and each plot-vector in a cut-out. If a cut-out contained the trajectory of a linearly moving object, the trajectory plots should all have the same angle to the three axes, especially to the time axis (cf. Fig. 25).

α	β	γ	P_{fa}	P_d
0.689806	2.69738	3.20716	0.01367	0.614602
0.631516	2.67498	3.24607	0.019998	0.674926
0.713465	2.5063	3.1953	0.030006	0.738312
0.665007	2.53706	3.16136	0.040004	0.780468
0.645584	2.5577	3.09876	0.050001	0.812543
0.646078	2.47176	3.15733	0.06	0.837062
0.680128	2.35982	3.19627	0.069999	0.85691
0.785598	2.32395	3.00264	0.08	0.874111
0.775479	2.34812	2.9495	0.09	0.888027
0.855749	2.24366	2.90833	0.1	0.899681
0.814567	2.26298	2.92558	0.11	0.909907
0.821996	2.27641	2.86409	0.12	0.918559
0.783539	2.24701	2.94614	0.13	0.926115
0.860416	2.18852	2.86017	0.14	0.932959
0.74721	2.3357	2.84066	0.15	0.93881
0.780142	2.24213	2.88437	0.16	0.944166
0.823062	2.27221	2.74452	0.17	0.949202
0.824634	2.23846	2.76603	0.18	0.953534
0.811256	2.24174	2.76504	0.19	0.957449
0.801666	2.27474	2.71818	0.2	0.960792
0.915937	2.16364	2.64702	0.21	0.964198
0.899562	2.13485	2.69635	0.22	0.96706
0.850495	2.25678	2.60189	0.23	0.969635
0.8792	2.13374	2.69935	0.24	0.972146

Table 5: Results of the maximization of the function $P_d = f(\alpha, \beta, \gamma, P_{fa})$, where α, β, γ are the weighting factors for the largest, middle and smallest eigenvalue respectively.

As the angle itself is not rectangularly distributed, but is the same for all plots belonging to one trajectory and so is its cosine. We cannot make use of the angle itself, but of its cosine.
Firstly we show that the cosine of the angle between noise vectors and one axis is rectangularly distributed. Thus, the cosine in the cut-out is rectangularly distributed, if the cut-out contains only noise (as shown below) and is not rectangularly distributed, if the cut-out contains a trajectory. The detection of such a diversion from the rectangular distribution could suite as a trajectory detector.

We regard noise as random vectors (x, y, t) or (r, ϑ, ϕ) in polar co-ordinates, which seem more suitable for the calculation. We have to prove that the mapping $g(r, \vartheta, \varphi)$, that maps a cut-out to an interval [a,b]

$$g(r, \vartheta, \varphi) : \longrightarrow \quad [a, b] \tag{3.1}$$

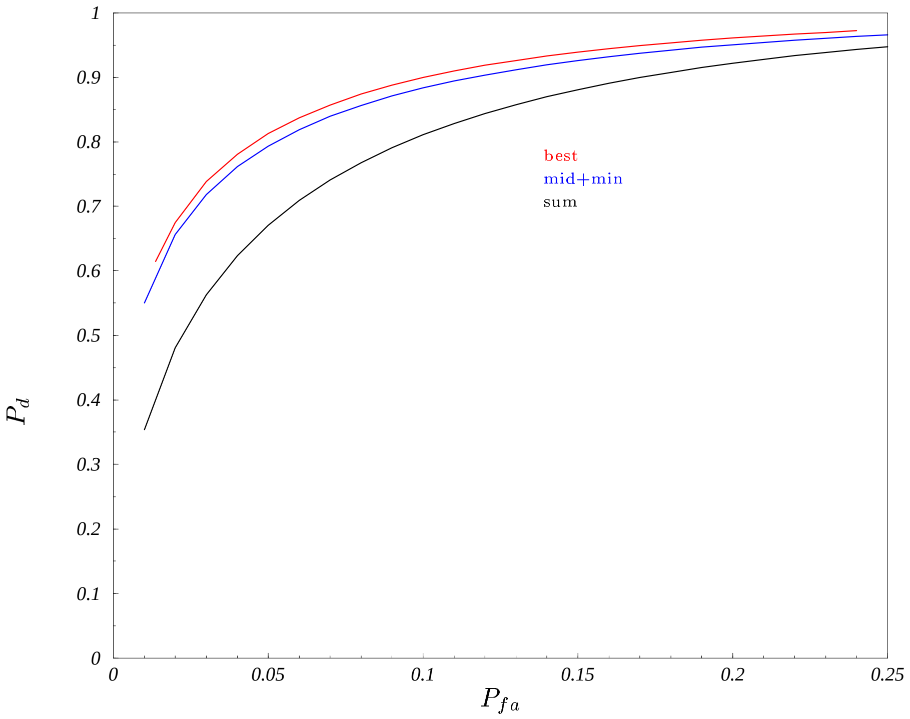

Figure 24: Receiver Operating Characteristics for a trajectory through the cut-outs' center. Mappings are *sum of all eigenvalues* (black), it sum of smallest and middle eigenvalue (blue) and the *weighted combination of eigenvalues* (red).

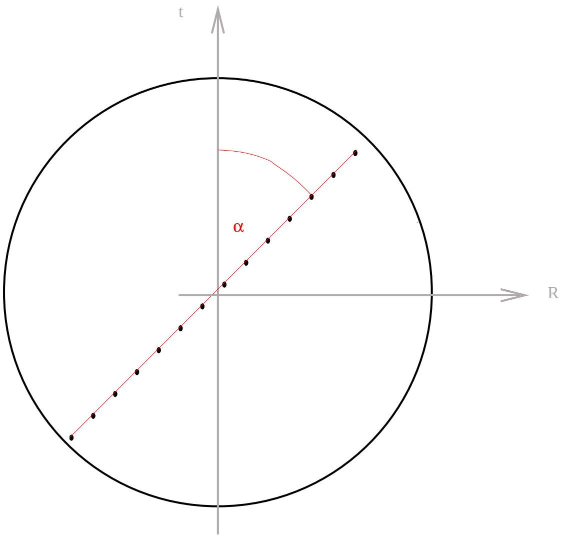

Figure 25: The angle between the time axis and plot-vectors of a trajectory in a cut-out.

is rectangularly distributed, i.e. fulfills the following condition:

$$Prob\{\alpha \leq g \leq \beta\} = \frac{\beta - \alpha}{b - a} \tag{3.2}$$

As the cosine maps to the interval [-1,1] we have to show that:

$$Prob\{\alpha \leq g \leq \beta\} = \frac{\beta - \alpha}{2} \ . \tag{3.3}$$

$$Prob\{\alpha \leq g \leq \beta\} = \int\int\int_{\{(r,\vartheta,\varphi)|\alpha \leq g(r,\vartheta.\varphi) \leq \beta\}} f(r,\vartheta,\varphi)\ g(r,\vartheta,\varphi)\ r^2\ \sin\vartheta \mathrm{d}r\, \mathrm{d}\vartheta\, \mathrm{d}\varphi\ , \tag{3.4}$$

where $f(r,\vartheta,\varphi)$ is the probability density function, in our case of a spherical cut-out of radius R_{sphere}

$$f(x,y,t) = \left(\frac{4}{3}\pi R^3_{sphere}\right)^{-1} = \frac{1}{V_{sphere}} \tag{3.5}$$

and

$$g(r,\vartheta,\varphi) = \cos\vartheta\ . \tag{3.6}$$

$$\begin{aligned} Prob\{\alpha \leq g \leq \beta\} &= \frac{1}{V_{sphere}} \cdot \int_0^{R_{sphere}} \int_{\arccos(\beta)}^{\arccos(\alpha)} \int_{-\pi}^{\pi} r^2\ \sin\vartheta \mathrm{d}r\, \mathrm{d}\vartheta\, \mathrm{d}\varphi \\ &= \frac{3}{4 \cdot \pi \cdot R^3_{sphere}} \cdot \frac{1}{3} \cdot R^3_{sphere} \cdot 2\pi \cdot \int_{\cos^{-1}(\beta)}^{\cos^{-1}(\alpha)} \sin\vartheta\ \mathrm{d}\vartheta \end{aligned} \tag{3.7}$$

Substituting

$$u = \cos\vartheta\ , \qquad \text{and} \qquad \mathrm{d}u = -\sin\vartheta\, \mathrm{d}\vartheta \tag{3.8}$$

$$\begin{aligned} Prob\{\alpha \leq g \leq \beta\} &= -\frac{1}{2} \cdot \int_\beta^\alpha du \\ &= \frac{\beta - \alpha}{2} \end{aligned} \tag{3.9}$$

Thus equation 3.2 is proven to be fulfilled: the cosine is rectangularly distributed on the interval [-1,1].

Having shown the rectangular distribution of the cosine, we now want to find an apt criterion to detect a trajectory. Firstly, we expect the dispersion of a distribution containing noise plus trajectory to be smaller than the one for noise alone, because the single value caused by the trajectory should occur more often according to the number of correlated data points. We thus try to make use of the dispersion of the cosine of the angle between the data vectors and one cut-out axis.

As before we make a Monte Carlo simulation with a volume of 10^6 events for this criterion for the two models 'noise' and 'noise plus trajectory' for trajectory offset 0 first.
In Fig. 26 we present the result and show, that indeed the distributions for the dispersion differ for the two models. The dispersion does not seem to be too good a criterion but yet, a criterion. Note: for the randomly generated diagonal trajectory we investigate in case the offset is 0, the angle to all directions (spatial and time) behaves in the same way; we thus only present the result for the angle to the time-axis.

In Fig. 27 we present the detection probability P_d against the trajectory offset for three constant false alarm probabilities P_{fa} 10%, 15%, 20%.
The maximum detection probability for P_{fa} 10% is ≈50% (for comparison: *sum of smallest and middle eigenvalue* obtained P_d 87.5%).
Thus it is not suitable to serve as a stand-alone detection criterion. But its advantages consist on one hand of its being completely independent from the eigenvalue criteria and on the other hand on the stability against the trajectory position in the cut-out. The detection probability is almost constant up to an offset of $R_{sphere}/2$. In combination with other criteria it might thus help to guarantee a more stable performance.

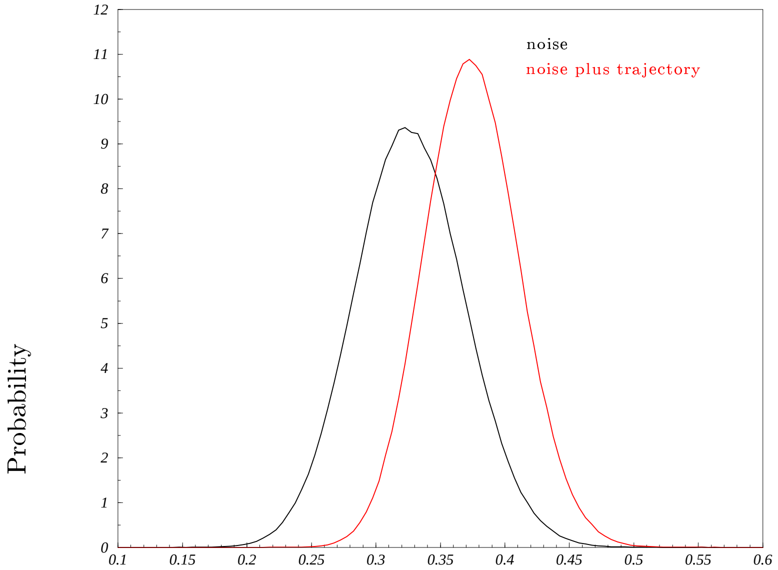

Figure 26: The probability density distribution obtained via Monte Carlo simulation for the mapping *dispersion of the cosine of the angle between data vector and time axis* for offset 0 for the models 'noise' (black) and 'noise plus trajectory' (red).

The Receiver Operating Characteristics for the criterion is shown in Fig. 28 in comparison to our standard criterion, the *sum of smallest and middle eigenvalue*. Clearly, the criterion *dispersion* is far from being as good as the *eigenvalue* criterion, but as this criterion is completely independent from the covariance matrix we hope to improve performance of a combination of all criteria.

For an application to real data we want to find out about how the criterion works for various ratios of correlated to uncorrelated data points. Like before we obtained the probability density distributions of the criterion for the models 'noise' and 'noise plus trajectory'. In Fig. 29 we show the expected values of the distributions in dependence of the number of data per cut-out. The trajectory used is a diagonal trajectory through the cut-out's center and the average number of correlated data is always 15. The error bars correspond to the standard deviation of the distributions.
As can be seen, the deviation of the two distributions is one σ only for 20 to 22 data points per cut-out, according to a signal to noise ratio of 3 to 2.1.

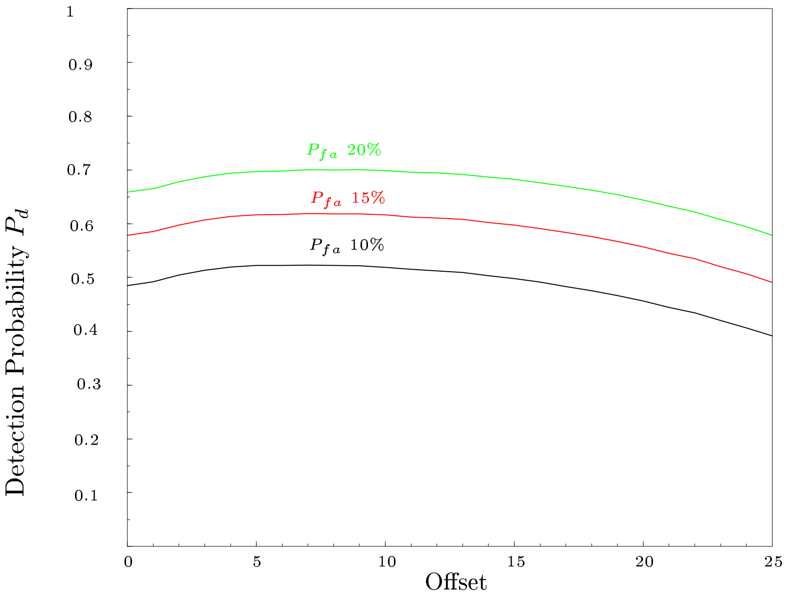

Figure 27: The detection probability P_d against offset of trajectory in cut-out for P_{fa} 10% (black), 15% (red) and 20% (green). Mapping is the *dispersion of the cosine of the angle between data vector and time axis.*

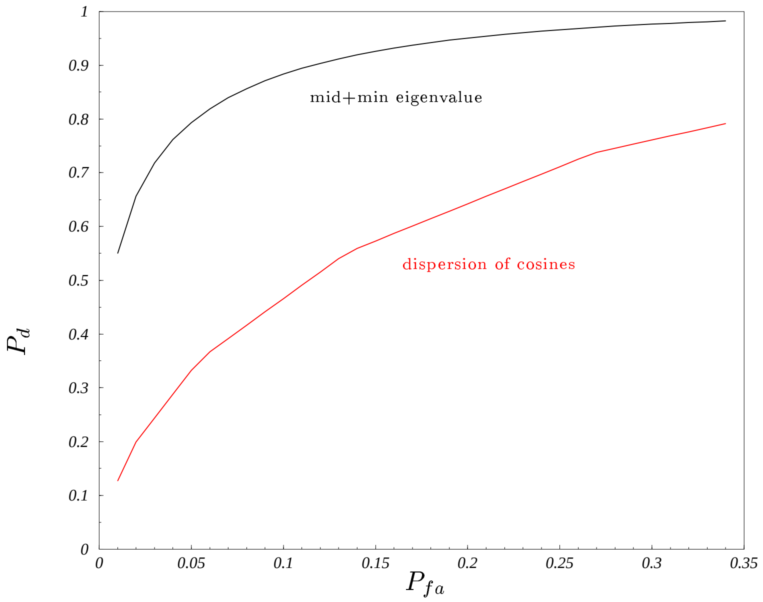

Figure 28: Receiver Operating Characteristics. Mapping is the *dispersion of the cosine of the angle between data vector and time axis* (red) in comparison to the *sum of smallest and middle eigenvalues* (black).

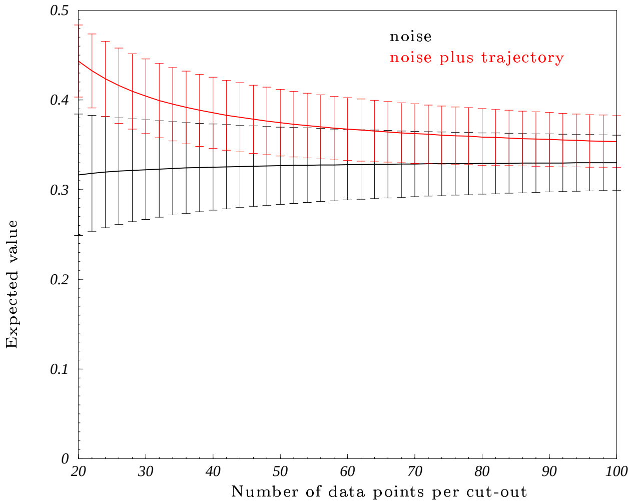

Figure 29: Expected values of the probability density distributions for the criterion *dispersion of cosine* in dependence of the data volume per cut-out for the model ‘noise plus trajectory’ (red) is and ‘noise’ (black).

3.7.2 Mapping *Kolmogorov-Smirnov one-sample test*

Another approach of making use of that angle is a Kolmogorov-Smirnov one-sample test. This test is capable of deciding whether one sample is distributed in a certain fashion. It is thus a way of testing a cut-out for the rectangular distribution of the cosine of the angle between the data vectors and the axes. For noise, the cosine is - as shown above - rectangularly distributed, whereas it is not, if the cut-out contains correlated data like a trajectory.

The Kolmogorov-Smirnov test measures the difference between the theoretically assumed distribution and the distribution calculated from the sample under discussion.

The theoretical distribution function is the straight-line on the interval [-1,1] with slope $\frac{1}{2}$ and intercept $\frac{1}{2}$. We choose 7 points for which we check the difference and obtain the greatest difference. The values for the theoretical distribution can easily be calculated. The values for the distribution in the cut-out can be derived by sorting all incoming values into a histogram with 7+1 bins. The derived values for the distribution is the number of values per bin divided by the total number of data in the cut-out. Thus we can easily compare the greatest difference between the assumed and the true distribution.

We made the test for the cosine of the angles to all three axes and combinations: x is the result for the angle to one axis (all axes lead to the same result for the investigated diagonal trajectory), xy is the sum of results for the angle to the x-axis and to the y-axis; xyt is the product of results for the angle to all axes. As can be seen from Fig. 32 the Kolmogorov-Smirnov one-sample test is an even worse detector than the previously investigated dispersion.

Nevertheless, it is another criterion that can probably serve to improve the result for all criteria.

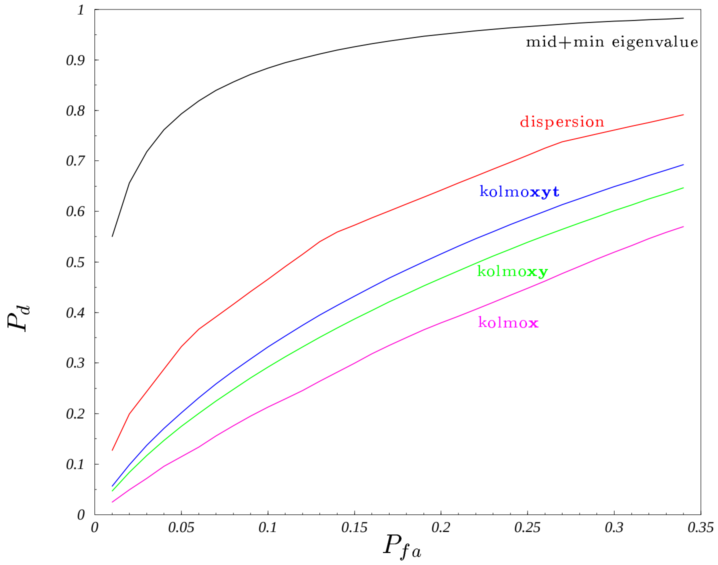

Figure 30: Receiver Operating Characteristics. Mapping is the *Kolmogorov-Smirnov test* for the cosine of the angle between the data vectors and one axis (magenta), two axes (green), all three axes (blue) in comparison to the *dispersion of the cosine* (red) and the *sum of smallest and middle eigenvalues* (black).

Again, we investigate about how the criterion works for various signal to noise ratios. In Fig.

31 we show the expected values of the probability density distribution for the criterion in dependence of the number of data points per cut-out. As can be seen, the two distributions do not differ significantly.

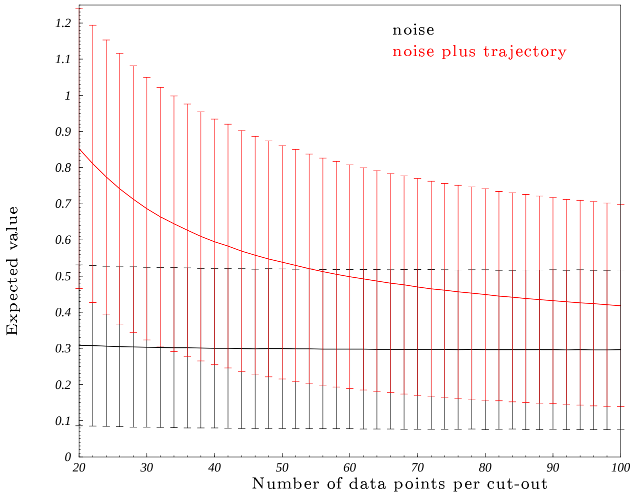

Figure 31: The expected values of the probability density distributions for the mapping *Kolmogorov-Smirnov test* in dependence on the number of data points per cut-out for the model 'noise plus trajectory' (red) is and 'noise' (black).

3.7.3 Mapping *Linear regression*

Another criterion that could suit to detect the difference between cut-outs containing noise or noise plus trajectory could be a linear regression.

We try to detect a correlation between the data in a cut-out. As we are looking for the trajectory of a linearly moving object we can try to fit a straight line into the data contained in a cut-out. This can be done by applying a linear regression. The quality of the fit is found through the least squares method, that measures the deviation from the straight line: it should be the smaller the more data in a cut-out are correlated linearly.

In detail this means, regarding two random variables X and Y, we want to find out about the nature of the functional connection $Y = f(X)$ between them. X and Y are two components of the measured plot positions, e.g. time and range.
In case of linear motion of an object the functional connection also is linear. We thus apply a linear regression to the data set assuming that for any but fixed value x the random number Y is normally distributed and has an expected value of

$$E(Y) = m \cdot x + b \; . \tag{3.10}$$

This relation indicates that the random number Y depends on average linearly on the fixed value x. As slope m and intercept b are unknown one can estimate their values applying the least square method to the sample represented by the data contained in a cut-out:

$$\sum_{i=1}^{n} \left[y_i - (m \cdot x_i + b)^2 \right] \overset{!}{=} \text{Min} \tag{3.11}$$

The estimates for m and b are then

$$m_{est} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{3.12}$$

$$b_{est} = \overline{y} - m_{est} \cdot \overline{x} \tag{3.13}$$

$$\text{where} \qquad \overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \text{and} \qquad \overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \ .$$

The sums in the equations 3.12 and 3.13 can be rewritten in terms of sums of all co-ordinates, not using the mean. One advantage is hence, that in a combination with the eigenvalue criteria this criterion would simply use the sums calculated anyhow and not need much additional computation time and load.

We thus obtain obtain slope m_{est} and intercept b_{est}, insert these values into the sum in equation 3.11 and calculate the sum of the least squares. This sum will serve as the criterion to distinguish between not linearly correlated data ('noise') and linearly correlated data ('noise plus trajectory').

As we expect the sum of the least squares to be smaller for cut-outs containing a trajectory, we applied two two dimensional linear regressions in each cut-out. We tested for correlation between the time component and each space component.

The result is presented in Fig. 32. The results for the two single regressions are identical for the diagonal trajectory, so we combined the two regressions to their sum.
As their individual performance is identical, it makes no sense to minimize their combination as we did with the single eigenvalues (cf. Section 3.6). We thus weighted them with the same factor 1 each.

The criterion turns out to perform surprisingly well. For a false alarm probability of 10% the detection probability reaches 80%. In Fig. 32 it is shown comparison to our best criterion so far, the *sum of smallest and middle eigenvalue*, the *sum of regressions.*

Again, we investigate about how the criterion works for various ratios of correlated to uncorrelated data. In Fig. 33 we show the expected values of the probability density distribution for the criterion in dependence of the number of data points per cut-out. The error bars are according to the standard deviations. As can be seen by the large gap between the two curves, the probability density distributions of this criterion for the cases 'noise' and 'noise plus trajectory' are significantly different. The expected values for the two cases differ for more than one σ for sub-sample volumes of up to 52 data points. The data consist of an average of 15 correlated plots forming the signal and 37 uncorrelated noise data, thus representing a signal to noise ratio of 0.41.

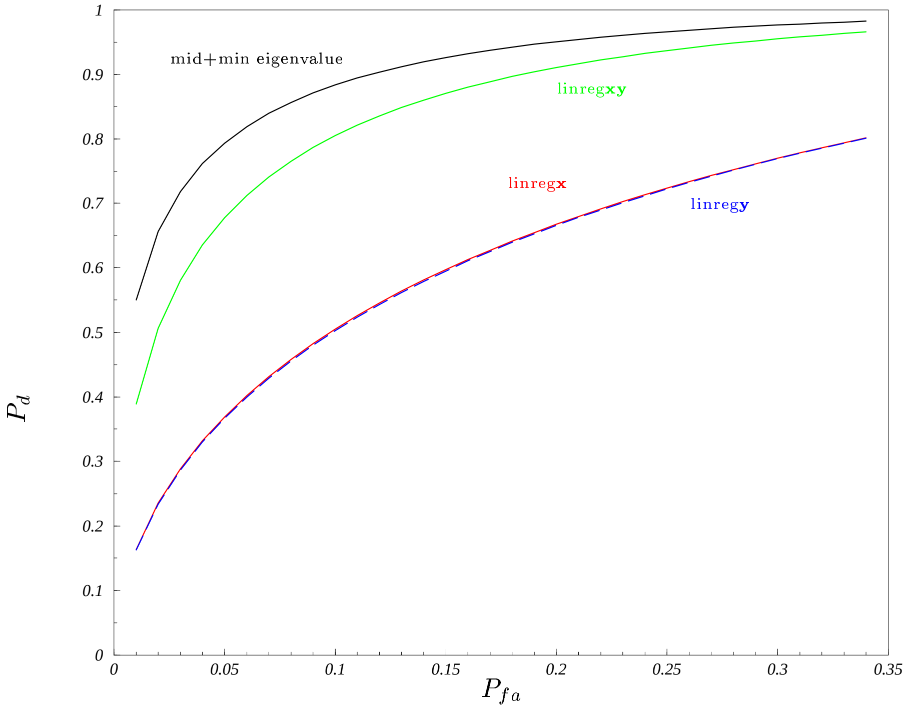

Figure 32: Receiver Operating Characteristics. Mapping is the *linear regression* between time axis and one spatial axis (red) or the other spatial axis (blue) and combination of both spatial axes (green) in comparison to the *sum of smallest and middle eigenvalue* (black).

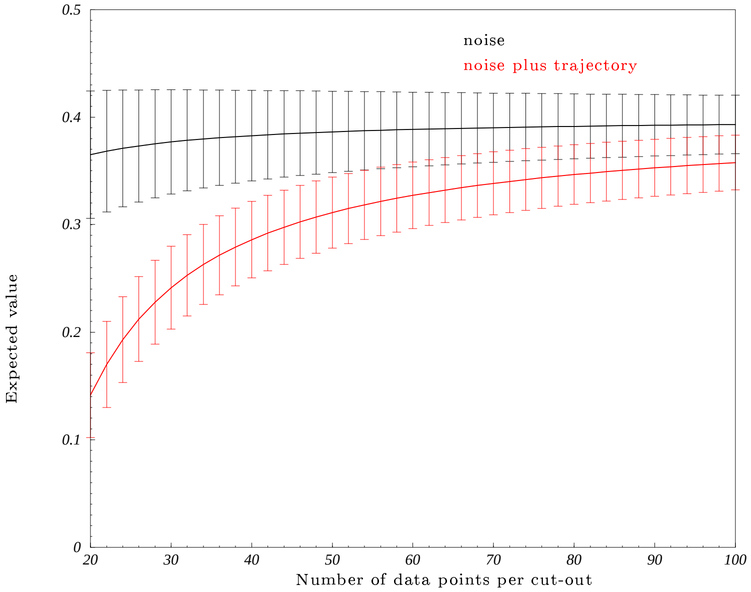

Figure 33: The expected values of the probability density distributions for the criterion *linear regression* in dependence on the number of data points per cut-out for the model ‘noise plus trajectory’ (red) is and ‘noise’ (black).

3.8 Obtaining a weighted combination of all mappings

Finally we combine all criteria investigated until now to find the optimum combination.

We combine the following criteria

1. f_1*(cut-out)*: smallest eigenvalue of the covariance matrix
2. f_2*(cut-out)*: middle eigenvalue of the covariance matrix
3. f_3*(cut-out)*: greatest eigenvalue of the covariance matrix
4. f_4*(cut-out)*: linear regression between x and t plus linear regression between y and t
5. f_5*(cut-out)*: product of Kolmogorov-Smirnov-tests in each direction
6. f_6*(cut-out)*: dispersion of the cosine of the angle between data vectors and time-axis

to one six dimensional criterion

$$f\,(\textit{cut-out}) = \sum_{i=1}^{6} \alpha_i \cdot f_i(cut - out)\ .$$

The criterion will decide for 'noise plus trajectory', if f*(cut-out)*$<$ 1.0 and elsewhere for 'noise'.

We will find the six weighting parameters α_i for the best combination analogous to Section 3.6, but this time the maximization via the Downhill Simplex Method (cf. Appendix C) is done in six dimensions.

The result of the receiver operating characteristic is shown in Fig. 34. The obtained combination is by far better than each of the single criteria.
We list some values for the P_d in Table 6. E.g. for a P_{fa} of 5% we reach a P_d of 90%a P_{fa} and for a P_{fa} of 10% we reach a P_d of 96%.
This is a very good result indeed.

α_1	α_2	α_3	α_4	α_5	α_6	P_{fa}	P_d
12.257	10.5808	5.02779	-5.16858	-0.319143	-4.63608	0.011945	0.714697
12.0046	9.90245	5.07341	-5.0872	-0.285429	-4.84322	0.05	0.904136
12.2024	10.2249	5.55704	-5.39628	-0.307908	-5.37498	0.1	0.959906
10.7323	8.65733	4.84814	-4.64481	-0.247326	-4.46535	0.15	0.979845
11.5364	9.73153	5.39314	-5.31047	-0.263039	-5.19323	0.2	0.989226

Table 6: The results of the maximization of P_d using a weighted combination of all mappings; α_i are the weighting factors for the single mappings f_i*(cut-out)* listed above.

The probability density functions for offsets 0 and 20% of R_{sphere} are shown in Fig. 35.
For offset 0 the two distributions only overlap very few, clearly indicating that the values for

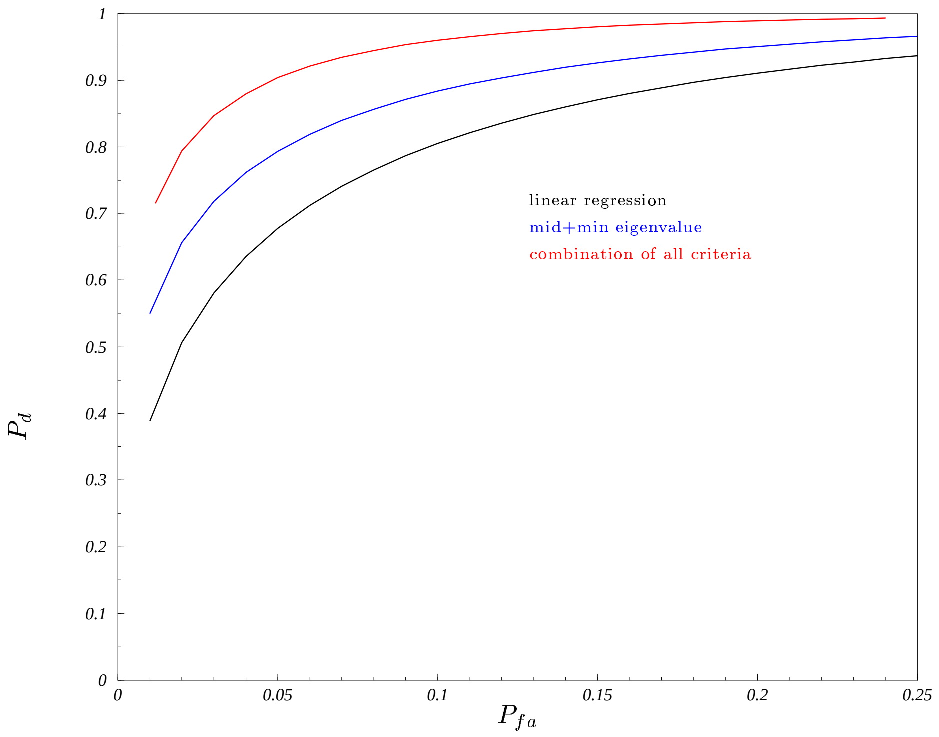

Figure 34: Receiver Operating Characteristics. The weighted combination of *all criteria* (red), compared to the *sum of smallest and middle eigenvalue* (blue) and *linear regression* (black).

the combined criteria are very different indeed, if the cut-out contains either noise alone or noise plus trajectory.

Especially in comparison to Fig. 4 in Section 3.3 where we have shown the probability density functions of the eigenvalue criteria one can see what huge improvement has been made concerning the reliability of the detection criterion.

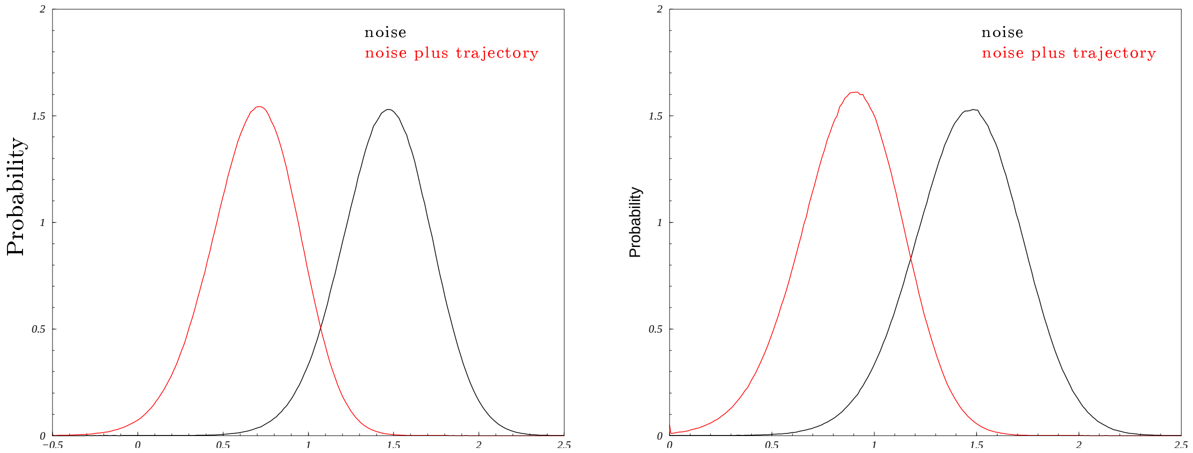

Figure 35: The probability density density for the combination of all mappings for a diagonal trajectory (left) and a trajectory at offset 10 (i.e. 20% of R_{sphere}) from the cut-out's center (right).

As discussed previously in Section 3.7.1, we expect the relatively weak criteria as e.g. the *dispersion of cosine* (cf. Fig. 26) to contribute to the improvement of the performance when all criteria are combined.
We now want to see if and in which degree this weak criterion actually does improve the detection probability.

We reran the minimization algorithm in five dimensions for all criteria but (i) the *dispersion of cosine* and (ii) the *Kolmogorov-Smirnov test.* The resulting receiver operating characteristics are shown in Fig. 36 and some of its values are given in Table 7.
Both results are worse in performance than the combination of all criteria. The ROC missing the Kolmogorov-Smirnov test achieves better results than the one missing the dispersion of the cosine. That means, that the contribution of the Kolmogorov-Smirnov test to the combination of all criteria is smaller than the one of the dispersion of the cosine. This is, of course, in perfect agreement with the ROCs of the single criteria given in Fig. 28 and Fig. 32. Also, from Table 6 one can see that for the multidimensional maximization the Kolmogorov-Smirnov test (α_5) is weighted far less than the dispersion of the cosine (α_6), i.e. $\alpha_5 \ll \alpha_6$.
Comparing the values in Table 7 for the P_{fa}=10% we can state that the *dispersion of the cosine* improves the detection probability in combination with all other criteria for about 4%. The Kolmogorov-Smirnov test improves for the same P_{fa} the detection probability for about 1.5%. Both criteria, each on its own very weak, do indeed contribute to the improvement of the detector when combined with all other criteria in not a negligible way.

P_{fa}	P_d		
	all criteria	*without Kolmogorov-Smirnov test*	*without dispersion of cosine*
1.2%	71.5%	68.4%	64.7%
5%	90.4%	87.6%	84.5%
10%	96.0%	94.4%	92.1%
15%	98.0%	97.0%	95.1%
20%	98.9%	98.3%	97.1%

Table 7: Results of the maximization of P_d for the combination of all mappings in comparison to all but the *dispersion of cosine* and all but the *Kolmogorov-Smirnov test* .

For an application to real data it is utterly important to know how many correlated data points are needed for a satisfying performance of the combined criterion. As the number of correlated data points per cut-out is equal to the number of time-layers required for the accumulation of data out of which we want to detect trajectories. Naturally, we want to do so as quickly as possible. Hence we want the best performance for as few correlated data points possible.

To compare the reliability of the combined criterion for different numbers of correlated data we obtained several receiver operating characteristics shown in Fig. 37: the curves in black belong to the combined criteria for an **average** number of correlated data of (top down) 12.7, 11.0, 10.0, 9.0, 7.8. (The odd numbers result from the way we generate the trajectory; for details cf. Appendix B). Clearly the performance worsens with decreasing number of correlated data. For comparative reasons we have also drawn the ROCs of the *sum of all eigenvalues*

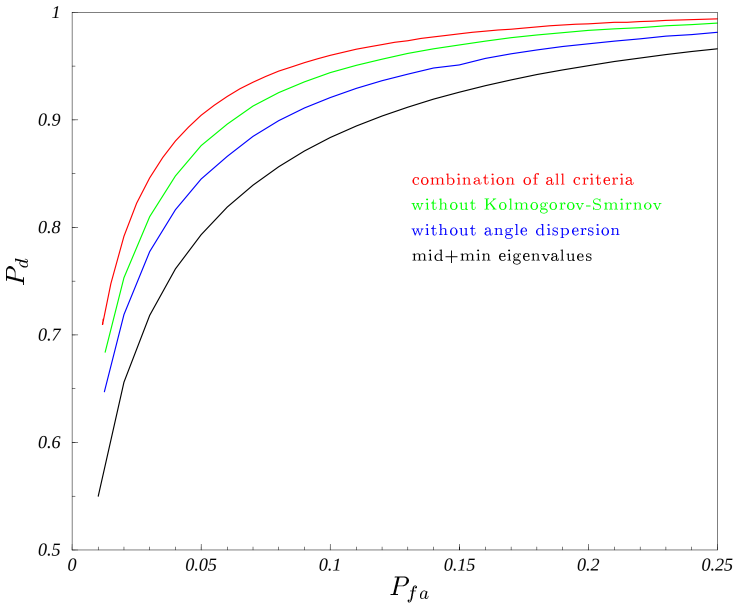

Figure 36: Receiver Operating Characteristics. Combination of *all criteria* (red), compared to the weighted combination of all **without** *Kolmogorov-Smirnov test* (green), *dispersion of cosine* (blue) and *sum of smallest and middle eigenvalue* (black).

(green) and the *sum of smallest and middle eigenvalue* (red). With the combined criteria we can achieve the same performance as these two criteria but with less correlated data necessary.

Also, we investigate about how the criterion performs for a constant number of correlated (15) and different numbers of uncorrelated data points (5-80). In Fig. 38 we show the expected values of the probability density distribution for the criterion in dependence of the number of data per cut-out. The error bars are according to the standard deviations. As can be seen by the large gap between the two curves, the probability density distributions of this criterion for the cases ‘noise’ and ‘noise plus trajectory’ are significantly different. The expected values for the two cases differ for more than one σ for sub-sample volumes of up to 88 data points, 15 of which belong to the trajectory and 73 of which are noise. The signal to noise ratio is hence 0.21.

For the application to real this result is important in so far as the noise is not distributed homogeneously over the entire area the number of data per cut-out may vary strongly. Knowing the dependence of the criterion’s performance on the number of data per cut-out one can adapt the strictness of the detection significance accordingly: the more data per cut out the greater the significance has to be.

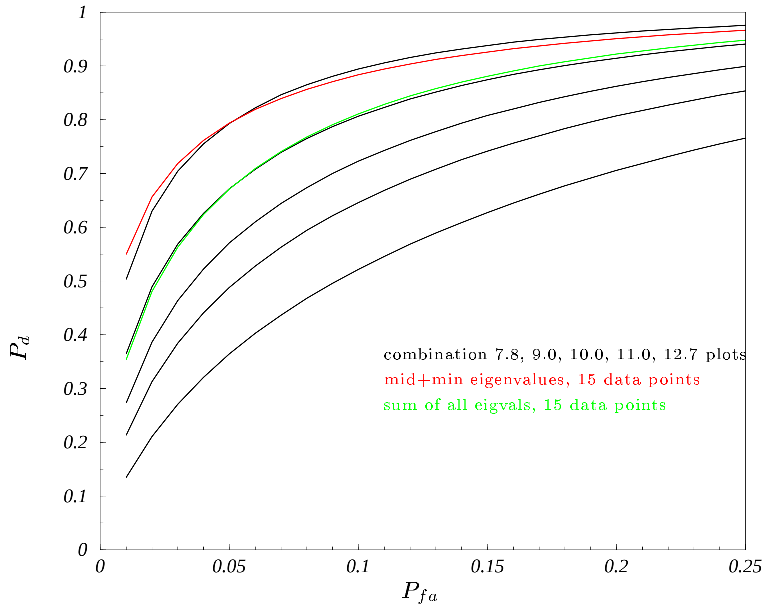

Figure 37: Receiver operating characteristic for the weighted combination of all criteria for less correlated data. Average number of correlated data is (top down) 12.7, 11.0, 10.0, 9.0, 7.8. Red is the *sum of smallest and middle eigenvalue* and green *sum of all eigenvalues* with 15 correlated data points.

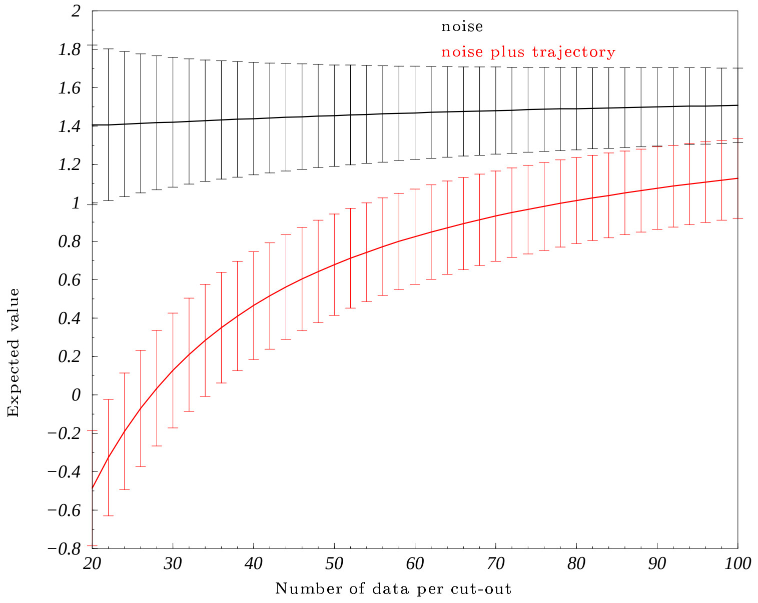

Figure 38: The expected values of the probability density distributions for the *weighted combination of all criteria* in dependence on the number of data per cut-out for the model ‘noise plus trajectory’ (red) is and ‘noise’ (black).

4 Two ways of implementing the method for the usage on real data

In this chapter we want to describe two possible implementations of our method for the processing of an unknown set of real data. We will describe two ways of doing so. Also, we will study the performance of the algorithm and possible resulting modifications.

4.1 Intrinsic implementation

As a first possible implementation we can apply the algorithm in a way, that all required values are calculated exclusively from the set of data under discussion. No external information like those obtained through the simulations is necessary. This implies not only the calculation of the cut-out radius, and the grid over which the sub-sample is being moved while processing the data. Also, the threshold for the detection criterion is also calculated online. As we have seen before, the analytical calculation of the expected values is only possible for the *sum of all eigenvalues*. Thus this implementation method is limited to this one criterion.

After reading the set of data from a file and calculations the adaptive quantities mentioned above, and according to the covered volume and cut-out radius the starting cut-out is obtained. For this first unknown sub-sample the detection criterion is calculated and the resulting value is tested for its multiple-sigma deviation from the expected value of the criterion's distribution for the model 'noise'. The latter depends on the cut-out radius (cf. Section 3.2) and is calculated immediately after the derivation of the radius. With these two values available we can then make a decision as to the content of the sub-sample: we could e.g. decide for 'noise plus trajectory', if the calculated value differed more than $n\sigma$ from the expected value for 'noise' and failing which we would decide for 'noise'. The algorithm will finally return all data belonging to suspicious sub-samples.

The advantage of this method is, that it only uses the actual set of data under discussion for the adaptive calculation of all parameters. It needs no information from simulations or elsewhere. The main disadvantage is, as shown before, the required calculation of the expected value and standard deviation. This calculation can only be done for the criterion *sum of all eigenvalues*, which is not the best criterion.

4.2 Implementation using externally obtained information

A second way of implementing the algorithm could profit from the results of our previous studies. It explicitely makes use of the parameters we have obtained for the weighted combination of all single detection criteria. To set the detection threshold it is necessary to pre-select a P_d-P_{fa} combination, because the according weighting parameters are required for the calculation of the threshold value.

As before, the data to be investigated are being read from a file and stored in an array. The data-dependent quantities as cut-out radius and grid are calculated as before. Previously

we have chosen a detection probability (or false alarm probability) before starting the data processing. Then the according parameters stored in e.g. an internal database would be used to calculate the criterion. This database would e.g. contain the parameters from Table 6 for the combination of all criteria. Analogous to the simulation, the criterion would then decide according to the unit function of Heaviside. If the selected criterion is fulfilled we will decide for 'noise', failing which we will decide 'noise plus trajectory' and return the selected data contained in the corresponding sub-sample .
It is not necessary to use especially this combination of *all* criteria. One can combine all criteria she renders suitable for the specific situation.

The main advantage of this method is, that the operator can adapt the P_d-P_{fa} combination according to specific situations, like extraordinarily heavy or exceptionally inhomogeneous noise. This high flexibility can contribute to the stable performance of the method in many different situations. The disadvantage is that a database has to be used which contains parameters gained from simulations. Although the simulations are being made according to the conditions known about the real system they still only consider an average situation. There may be extreme conditions under which the data deviate exceptionally strongly from the assumed average situation so that the algorithm may fail. But if we want to use criteria other than the *sum of all eigenvalues*, this method is an apt way to do so.
Of course it is recommendable to rerun the minimization method with data simulated in accordance with the type of real data planned to be processed and not regard the values we obtained as suitable for all kinds of data.

We also want to emphasize that our algorithm is not capable of detecting the actual trajectory itself. It can only find and select small sub-samples of the original data that contain a trajectory with a certain probability.

In the following we will give examples of both ways of implementing the algorithm to various sets of real data.

5 Application to real data: Radar

As a first application to real data we chose a set of radar data.
We want to find trajectories of moving objects in a set of data recorded with a rotating sensor. Radar tracking applications include e.g. airport and harbour surveillance, navigation system on ships or maritime patrol for search and rescue operations.

A common radar system obtains data with a rotating antenna which sends electro-magnetic radiation and receives its reflections by objects. The radiation is emitted in pulses and the emission alternates with the reception. During one rotation (*scan*) the antenna sends a pulse in a direction, receives all echos from objects, memorizes the information (position, amplitude, Doppler-shift etc.), continues to rotate, sends, receives, stores and so on. The data obtained contain the time of the measurement and the position of the echo. The position consists of the rotation angle of the antenna (azimuth) and the distance of the reflecting object (range), obtained through the run time of the signal. Every data point has thus (at least) three dimensions.
If two data points $(t_1, \vec{x}_1)$ and $(t_2, \vec{x}_2)$ belong to a linearly moving object, detected at time t_1 during scan no. 1 and time t_2 during scan no. 2, we expect a detection of the same object at time t_3 in scan no. 3. Linear motion means, that the corresponding velocity vector $\vec{v}$ remains unchanged:

$$(\vec{x}_2 - \vec{x}_1)/(t_2 - t_1) = \vec{v} = (\vec{x}_3 - \vec{x}_2)/(t_3 - t_2) \ .$$

If we collect data over several scans and the object is for example a ship moving linearly and straight across the ocean, the trajectory are data points that are linearly correlated in time and space. Noise and disturbances have their origin in antenna insufficiencies, observational inaccuracies or reflections of objects one is not interested in detecting (*clutter* from e.g. clouds, rain, waves etc.).
An important and challenging task is radar tracking in high false alarm environment. The tracking objective of collecting sensor data from the field of view, find objects of interest and establish an estimated state trajectory is rendered significantly more difficult under the presence of huge amounts of uncertain measurements. Common data association methods are not only very slow for large amounts of data they also lead to a great number of false alarms. The high bandwidth and sensitivity of modern radar systems can lead to huge data loads, the processing of which still has to be performed in real time.
In addition to these difficulties a special challenge is the detection of dim objects in the described environments of high false alarm. A main feature in estimating the state trajectory of an object is the reflected signal amplitude that depends on the object size, its surface and its distance. There are objects of interest e.g. common ships whose backscattering cross section is clearly larger than that of clutter produced by reflections of waves. However, there also are objects of interest whose backscattering cross section lies in the range of that of clutter, e.g. small ships, boats or castaways. The detection of dim targets can be enhanced by lowering the detection threshold or increasing the sensor sensitivity. Both options create the necessity to process a greater number of false or undesired returns. E.g. the sensitivity of modern radar systems is such that birds (and even bees) may be detected. By reducing the non-detection of dim targets with such measures the previously described computation time problem in

processing huge amounts of data is being enlarged.
Further difficulties arise when detecting slowly moving objects. An important indicator while detecting moving objects with a radar system is the object velocity measured through the Doppler shift of the received frequency with respect to the emitted frequency. Usually the velocity of interesting moving objects (e.g. ships) differs significantly from that of unwanted reflections and allows the easy detection of moving objects. Under certain conditions though, the Doppler velocity of the moving object (e.g. slow ships, castaways) is of the same size as that of the noise (e.g. waves). In this case the detectability of moving objects with common methods that use the Doppler velocity is severely limited.
Often, the above mentioned problems occur in combination, making the detection even more difficult.

The field of tracking small and slow objects in high false alarm environments is a challenging and much worked upon field. The reduction of false alarms and clutter is a strong aim of the pre-tracking processing chain. Many attempts are being made to reduce the false alarms at least to a constant predictable rate (CFAR - constant false alarm rate), e.g. [Rohling1983], [Hofele2002], [Galati2002], [Gessler2002]. Although this method is twenty-five years old and widely used it is still not sufficient as the many works in this area as well as the necessity of special clutter treatment in following processing units prove. Many methods try to improve the data association process by modeling the background noise ([Gray2002]). There are studies on statistical models for the spatial distributions of clutter, assuming Rayleigh or log-normal distribution ([Kassotakis2001]). Other methods imply models of the ellipticity of the scattered wave from sea clutter ([Pusone02]) or the backscattering Doppler spectrum as a function of wind induced motion of sea waves ([Pusone2000]). These models are strong functions of the prevailing sea condition, which is highly variable and unpredictable. They hence suffer from influences out of our control. Others struggle to great expense to make use of the Stokes vectors obtained through the polarization measurements and develop a theoretical model for the characterization of the polarimetric features of the radar signal ([Pusone2002], [Lightart2002]). Yet, these methods have not even been tested on real data (private communication with E. Pusone) and the optimal polarization to be used for these studies is also still under discussion ([Carrea2002]).
The method we developed to detect trajectories in discrete data is applicable on radar data as a detector of moving objects in high false alarm environment that makes neither use of the Doppler velocity nor data association techniques nor global clutter models.

5.1 General adaptions and application to radar data

Before we can process radar data some preparatory procedures have to be automated, e.g. the adaptive calculation of the cut-out's radius and the gauge on which the cut-out is moved over the data set. According to the data, their number and the field they cover we want these quantities to be adaptively derived.

To derive the cut-out's radius for a set of radar data is not trivial. Several aspects and inherent restrictions have to be considered:

1. Co-ordinates (azimuth, range, time): the azimuth ranges from 0 to 2π, range from 0 to several hundred kilometers and time from 0 until the end of the recording. Thus the metric $x^2 + y^2 + t^2 \leq R^2$ that has been used before to divide the data set into sub-samples has to be modified.

2. In order to detect a track with our method at least 15 scans per cut-out are required; thus the radius in direction of the time is already given (cf. also Fig. 39).

3. At the same time the cut-outs should contain an average number of about 40 plots.

Firstly, we are looking for a co-ordinate transformation and a reasonable metric:

$$\begin{aligned} Azimuth &\longrightarrow x = f_{Az} \cdot Azimuth \\ Range &\longrightarrow y = f_{Ra} \cdot Range \\ Time &\longrightarrow t = f_T \cdot Time \end{aligned} \tag{5.1}$$

$$(f_{Az} \cdot Azimuth)^2 + (f_{Ra} \cdot Range)^2 + (f_T \cdot Time)^2 \leq R_{sphere}^2 \ . \tag{5.2}$$

Dimensions of the factors are:

$$\begin{aligned} f_{Az} &\quad \text{in} \quad \frac{1}{rad} \\ f_{Ra} &\quad \text{in} \quad \frac{1}{m} \\ f_T &\quad \text{in} \quad \frac{1}{s} \end{aligned}$$

Thus the transformed co-ordinates as well as R_{sphere} are dimensionless.
The time scaling is given by the number of scans required to detect a track (15):

$$\begin{aligned} R_{sphere} &= f_T \cdot \frac{1}{2} \cdot T_{sphere} \\ f_T &= \frac{R_{sphere}}{\frac{1}{2} \cdot T_{sphere}} \end{aligned} \tag{5.3}$$

where the time radius $\frac{1}{2} \cdot T_{sphere}$ corresponds to e.g. 10 scans (cf. Fig. 39).
In order not to distort the spherical radar coverage area too much we demand validity of the following relation (cf. Fig.40):

$$\frac{L}{\Delta Ra} = \frac{\Delta Az \cdot Ra_m}{\Delta Ra} = \frac{\Delta Az \cdot f_{Az}}{\Delta Ra \cdot f_{Ra}} \tag{5.4}$$

$$\Longrightarrow Ra_m = \frac{f_{az}}{f_{Ra}} \tag{5.5}$$

If we set $f_{Az} = 1$ we obtain

$$f_{Ra} = \frac{1}{Ra_m} \ . \tag{5.6}$$

We find the cut-out's radius through the following relation, N is the total number of plots per cut-out:

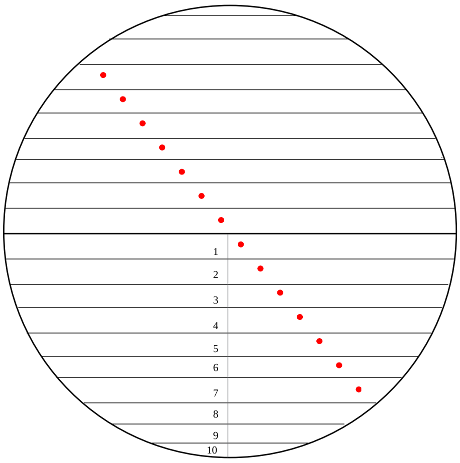

Figure 39: Estimation of the number of scans (10) per cut-out (corresponding to the radius in time) required to detect a track consisting of 15 plots (red).

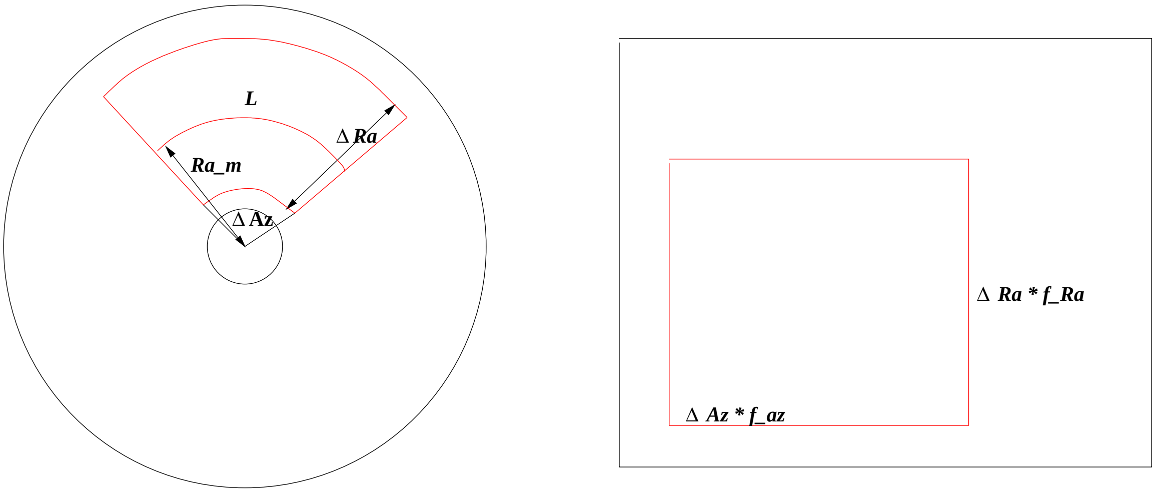

Figure 40: Illustration of the co-ordinate transformation necessary for the application to radar data to fulfill the demand for local rectangular distribution.

$$
\begin{aligned}
\frac{N_{sphere}}{N_{tot}} &= \frac{V_{sphere}}{V_{to}} \qquad (5.7)\\
&= \frac{\frac{4}{3}\cdot\pi\cdot R_{sphere}^3}{(x_{max}-x_{min})\cdot(y_{max}-y_{min})\cdot(t_{max}-t_{min})}\\
&= \frac{4\cdot\pi\cdot T_{sphere}\cdot R_{sphere}^2}{3\cdot f_{Az}\cdot f_{Ra}\cdot(Az_{max}-Az_{min})(Ra_{max}-Ra_{min})(T_{max}-T_{min})}
\end{aligned}
$$

With $f_{Az}=1$ and $f_{Ra}=\frac{1}{Ra_m}$ (see above) we derive

$$R_{sphere}^2 = \frac{3}{4 \cdot \pi} \cdot \frac{N_{sphere}}{N_{tot} \cdot Ra_m} \cdot \frac{(Az_{max} - Az_{min})(Ra_{max} - Ra_{min})(T_{max} - T_{min})}{T_{sphere}} \; . \tag{5.8}$$

5.1.1 Intrinsic Implementation

We will firstly implement the first method described in Section 4.1 and the *sum of all eigenvalues* as criterion.

Available are sea clutter data obtained with an EADS naval radar system. The recorded data cover a time of 32 minutes, out of which we chose 70 seconds to process with the new algorithm. 70 seconds correspond to 20 scans, so we can use a reliable cut-out radius of R_{sphere}=10 scans.

The original data set is shown in Fig. 41. The track is visible in the center running diagonally from north-west to south-east.

Also, the reason for the previously described co-ordinate transformation becomes visible: the clutter it not equally distributed in azimuth for all range values. For smaller distances from the sensor the plot density is higher than for greater distances. As most of the clutter origins in the reflections by waves this effect of decreasing clutter density with increasing distance is a shading effect: The radar system is positioned at some height over sea and looks down onto the water. Due to this aspect angle waves that are closer to the ship shield waves farther away from the sensor. In order to correct this effect we apply the transformation as described in Fig. 40.

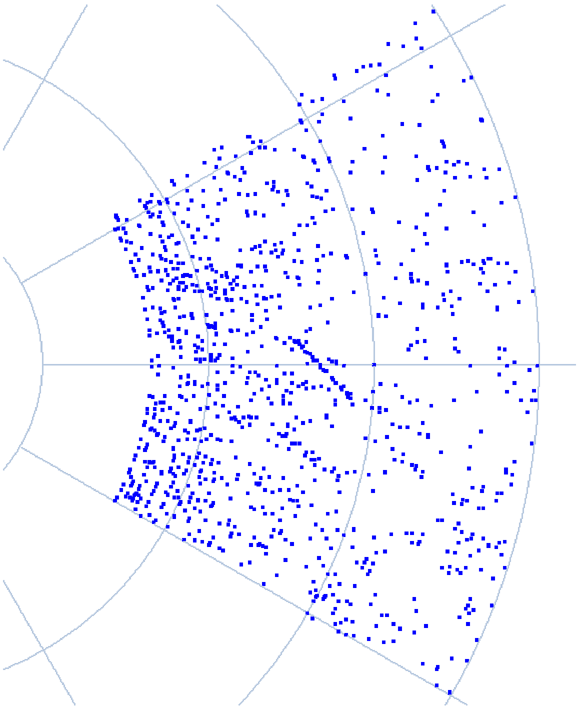

Figure 41: Real data from a naval radar system recorded over 20 scans. The diagonal track is clearly visible in the center.

After the co-ordinate-transformation the data look as shown in Fig. 42. The track is still visible in the center. But now, the clutter plots are equally distributed in x and y. The red circle indicates the size of a cut-outs in which the data are being processed. After the transformation the cut-out is a sphere, here shown in two dimensions. Its size has here been chosen in a way that the average number of plots per cut-out is 45.

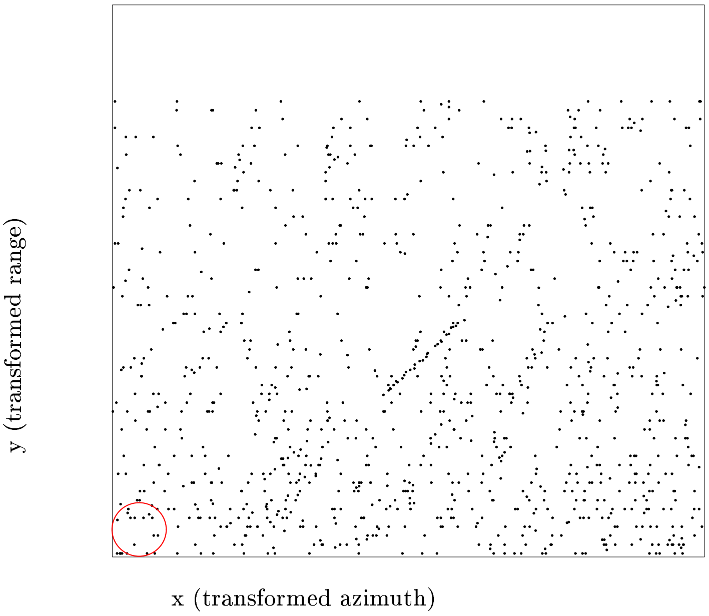

Figure 42: Data set from Fig. 41 after the co-ordinate transformation. The track is visible in the center. Indicated by the red circle is the size of the now spherical cut-outs.

The transformed data set is now being processed as previously described: the cut-outs are moved across the data set with an overlap of $\frac{5}{3}R_{sphere}$. In order to illustrate how densely the cut-outs are placed and to show that every plot is processed several times in different cut-outs we show the corresponding configuration in Fig. 43 *I*.

The clutter is not homogeneously distributed over the entire area. Since we only chose the average number of plots per cut-out here to be 45, the various cut-outs may contain very different numbers of plots (cf. Fig. 43, e.g. *II, III, IV*). Due to this differences there may occur artifacts: if a cut-out contains e.g. only one plot, it is very likely to be marked to contain a statistical anomaly; the distribution clearly deviates from the local normal distribution we assumed for clutter. In order to avoid false alarms caused by such and alike artifacts we also take into account the number of plots per cut-out. We modify the detection criterion to the following:

The cut-outs are marked to contain a track, if either of the following conditions are fulfilled:

- If the cut-out contains between 15 and 30 plots plots and the difference between the expected value for the selection criterion and calculated value is $30\,\sigma$ (which is a very strict empirically derived value).
- If the cut-out contains more than 30 plots and the difference between the expected value for the selection criterion and the calculated value is at least $3\,\sigma$.

If the number of plots per cut-out is less than 15 plots, the cut-out is not being processed at all. In this case the sample size is too small to make a reasonable prediction.

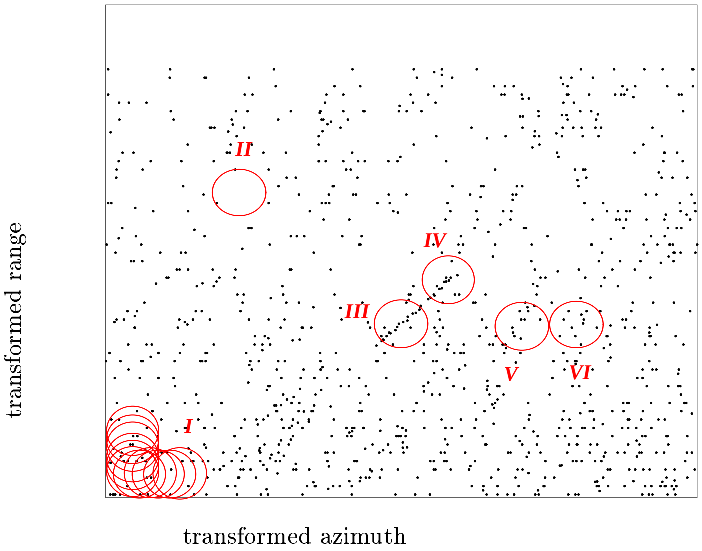

Figure 43: Data set from Fig. 41 after the co-ordinate transformation. The track is visible in the center. The red circles are examples for the now spherical cut-outs.

In addition to this we implement another restrictive measure: the less plots a cut-out contains the stricter the criterion is applied: the required significance is 30σ for tracks with a number of plots between 15 and 30. The chosen 30σ is an empirically derived value that leads to the best avoidance of false detections. Only for plot numbers of more than 30 we can expect the criterion to work sufficiently reliable. Since we still want to avoid false alarms we demand a significance of 3σ, based on our experience while studying real data.
The selection criterion in this case is the *sum of all eigenvalues*, as it is the only criterion we can calculate the expected value for.

The data contain 1061 plots and are divided into 221 cut-outs with an average number of 45 plots per sub-sample. There are 17 cut-outs in azimuth-direction and 13 in range direction. The overlap is $\frac{5}{3}R_{sphere}$ in each direction. The result is presented in Fig. 44; blue are the original data, red are the selected plots that lie within cut-outs marked to be suspicious of containing a track.

The algorithm chose only three cut-outs to be suspicious of containing statistical anomalies. It detected the track with one cut-out, namely the central one containing most track plots. In the outer regions we note two detections by adjacent cut-outs. Most probably they are false alarms, because we cannot see any anomaly in the plot distribution. Thus we have two false detections out of a total of three.
On the other hand, the algorithm detected the track reliably and only selected 110 out of 1061 which is only 10.4% of the original data.

One striking phenomenon that can be observed in Fig. 44: In some areas there are selected plots (red) next to unmarked plots (blue). This can in fact be the case, if the plots lie spatially close together, but not in time. We must not forget that we regard 20 time layers. For a plot to

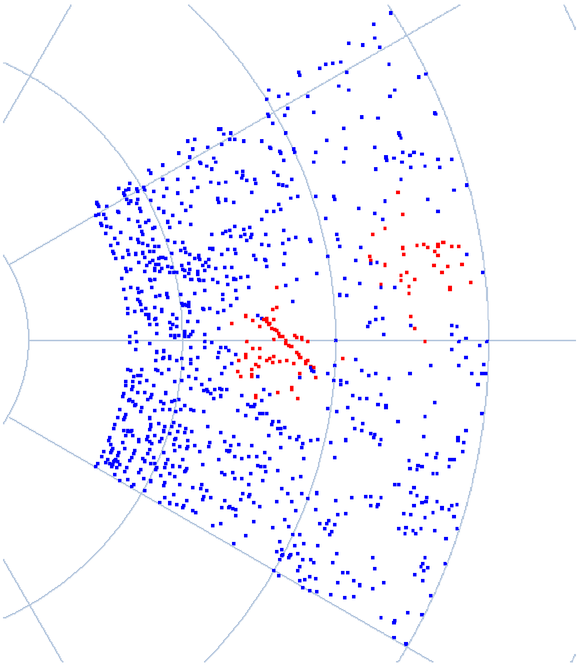

Figure 44: Real data from a naval radar system recorded over 20 scans. Blue are the original data, red are the selected plots from the suspicious cut-outs. Mapping is the *sum of all eigenvalues*.

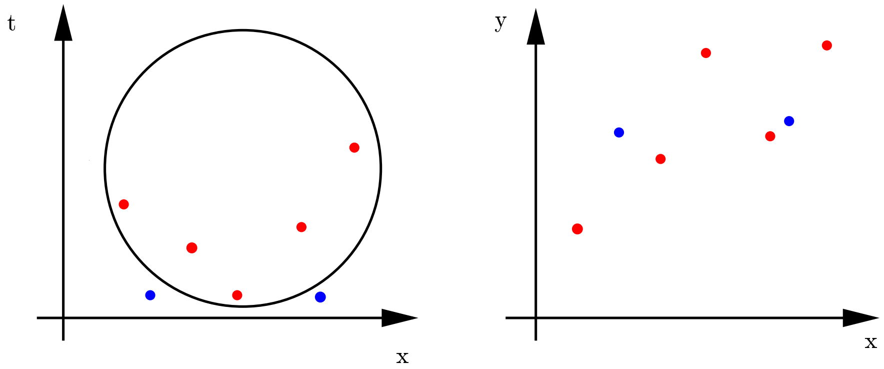

Figure 45: Suspicious cut-out with selected data (red) and adjacent blue plots, that do not lie within the cut-out as they would be seen in the x-y-illustration.

lie within one spherical cut-out it has to fulfill the equation $x^2 + y^2 + t^2 < R^2_{sphere}$.
If plots in the extreme time-layers (topmost or bottommost) lie spatially close to each other, they might not lie within the same cut-out. Thus, in the azimuth-range-illustration adjacent plots do not necessarily have the same color. This is illustrated in Fig. 45.

5.1.2 Implementation using externally obtained parameters

Another way of implementing the algorithm can use a combination of all (or several selected) criteria as described in Section 4.2. The method would no longer calculate the selection threshold intrinsically but use values that have been obtained through the simulations we have made before. Before, the detection threshold was defined by a multiple-sigma deviation from the calculable expected value for the model 'clutter'. Now, the detection threshold would be defined by an external parameter combination that sets the detection probability and false alarm rate to fixed and known values. The algorithm would decide for 'clutter plus anomaly', if

$$f(cut-out) = \sum_{i=1}^{n} \alpha_i \cdot f_i(cut-out) < 1.0 \ ,$$

where $f_i(cut-out)$ are the individual mappings building the combination. For the combinations of criteria possible and reasonable we would obtain the respective weighting parameters α_i as before through multidimensional minimization (cf. Chapter 3.8) and store them in an internal database.

The system operator would choose a suitable detection probability and a detection criterion. According to that the algorithm would use the respective parameters to define the detection threshold.

As before the cut-out size and the processing grid is obtained according to the data set. It leads to the same results as before: 1061 plots are divided into 221 sub-samples, 17 in azimuth-direction and 13 in range direction, the average number of plots per cut-out is 45. The co-ordinate transformation is also performed as before. The criterion now is the previously optimized *combination of all criteria*. The selected P_d is 70%, corresponding to a P_{fa} of 1%. The result is shown in Fig. 46.

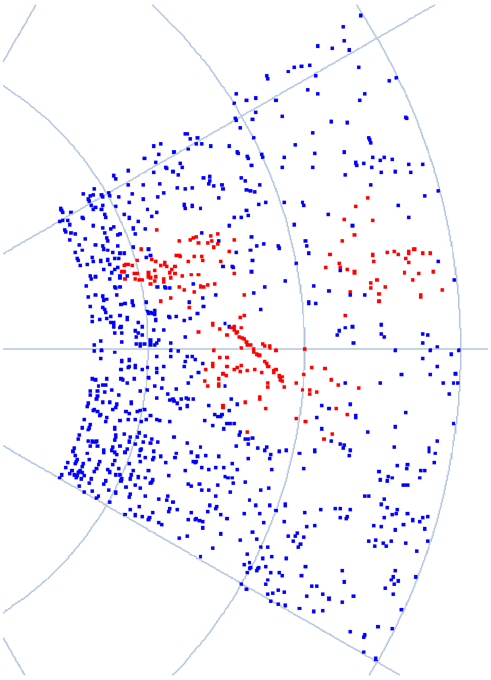

Figure 46: Real data from a naval radar system recorded over 20 scans. Red are the original plots, blue are the centers of the suspicious cut-outs. Mapping is the *combination of all criteria*

The algorithm selected twelve cut-outs to contain statistical anomalies, corresponding to 5.4% of all cut-outs.

Eight selected overlapping cut-outs clearly contain and thus detected the track. Eventually, to reduce the multiple detections of adjacent cut-outs one could consider reducing their overlap.

Two cut-outs left of the track are clearly false alarms. Two false cut-outs out of a total of 221 correspond exactly to the 1% false alarm probability we have selected before.

Right of the track are two more detections. At first sight we would say they were false alarms, because we do not see any statistical anomaly with the naked eye. But as can be seen from Fig. 44, the previous implementation of the algorithm using the *sum of all eigenvalues* as detection criterion has also selected this region. It did so with a significance of 3σ. Considering the diligently investigated reliability of both the *sum of all eigenvalues* and the *combination of all criteria* as detectors we cannot fully exclude an anomaly in the twice marked region of the dataset.

Thus we would count 4 - 6 false alarms corresponding to a P_{fa} of 1.8%-2.7% which corresponds to the P_{fa} of 1% we have selected previously within errors.

The algorithm has selected 210 plots out of 1061, which is 19.8% of the original data and almost twice as much as the intrinsic method selected.

A comparison of the two ways of using the algorithm is fairly difficult.

At first sight, the intrinsic method seems to perform far better, because it shows less detections and still selected the track. It does not detect the track several times (multiple detection means more selected data) and also shows less false alarms. On the other hand, because of the large overlap multiple selection by adjacent cut-outs only confirm the detection. Also, multiple detection enlarges the probability that if one cut-out did not detect an anomaly (due to P_d ¡ 1) its neighbours do not miss it as well. Thus, the single detection might not a stable detector.

In addition to that, the required 30σ deviation for cut-outs with less than 30 plots has been chosen especially strict in order to avoid false alarms.

The implementation using externally obtained values behaves absolutely predictably. It detected the track without any doubt and did not exceed the chosen false alarm probability of 1%. Although the illustration (cf. Fig. 46) suggests that a vast amount of data have been selected to be suspicious, in fact only 5.5% of all cut-outs have been marked. Not to forget the fact, that adjacent cut-outs contain very much the same data.

Nevertheless, this method has clearly selected more cut-outs and thus more data than the intrinsic method.

5.1.3 Application to further sets of radar data

For further comparison we have chosen another set of radar data from the same radar system as before. These data have been recorded on a different day under different weather conditions and thus show different clutter structures.

We will now process the entire area of radar coverage in azimuth direction. As the clutter density decreases with increasing distance from the sensor it is not necessary to process the entire coverage area in range direction. At greater distances from the sensor the common processing algorithms can handle the decreased number of clutter plots.

Again we have selected 20 scans out of a larger data set[1]. The total number of plots is 3146. The average number of plots per sub-sample has been set to 40, which is less than before. Since we noted dense multiple detection by adjacent cut-outs with the last data set, we now increase their overlap to $\frac{3}{2}R_{sphere}$. This leads to a division of the data set into a total number of 342 cut-outs, 56 in azimuth and 6 in range direction.

For the intrinsic implementation all parameters remain unchanged. The result is shown in Fig. 47. The algorithm selected three cut-outs and a total of only 80 plots, which corresponds to 2.5% of the original data. As we can see the method clearly failed to detect a track at azimuth 45° . It detected one anomaly visible at azimuth 120° and noted one false alarm at 250° .

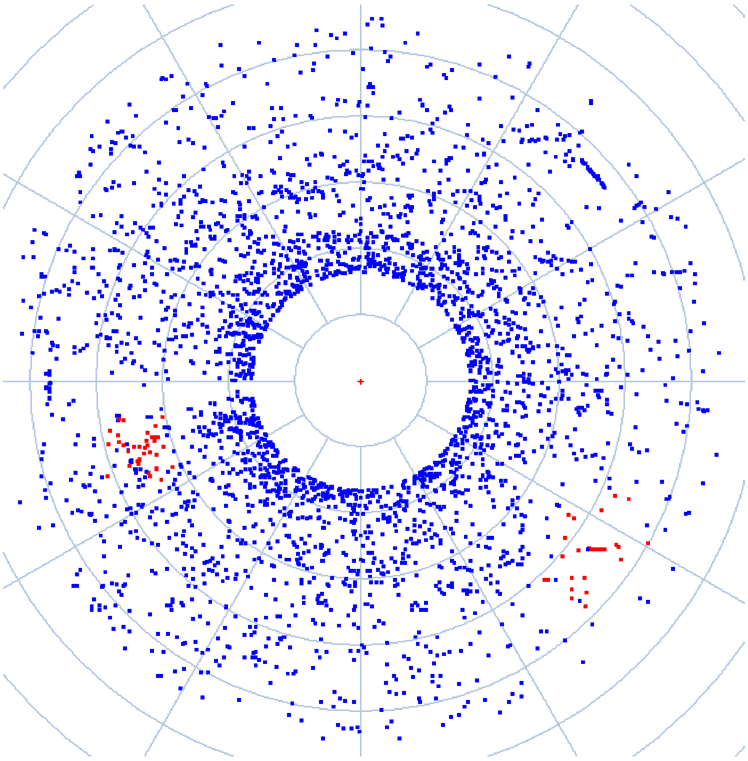

Figure 47: Real data recorded over 20 scans. Blue are the original plots, red are the plots from the suspicious cut-outs. Mapping is the *sum of all eigenvalues.*

[1] Between azimuth 260° and 280° there is a visible gap in the data. This is the position where the ship mast stands. Since the radar antenna is not the highest point on this ship its view is shielded by the mast and for a certain azimuth a small region is shaded. The effects vanishes with increasing distance.

Since we are not satisfied with this performance we reset the average number of plots per cut-out from 40 to 45. We thus increased the size of the cut-outs and decreased their number. We then rerun the algorithm. The data are now divided into 53 (before: 56) cut-outs in azimuth direction and 6 (before: 6) in range direction, which means a total of 318 (before 342) cut-outs. The algorithm has now selected four cut-outs with 145 plots which are 4.6% of the original data. The new result is shown in Fig. 48. This time, the algorithm has detected the track at azimuth 45° and as before, the one at 120° . We still have to note one false detection, this time at 345° . This time, the performance has improved compared to the situation before. The false alarm rate remained constant but the detection probability has increased significantly. We conclude that for the intrinsic implementation a higher average number of plots per cut-out is of advantage.

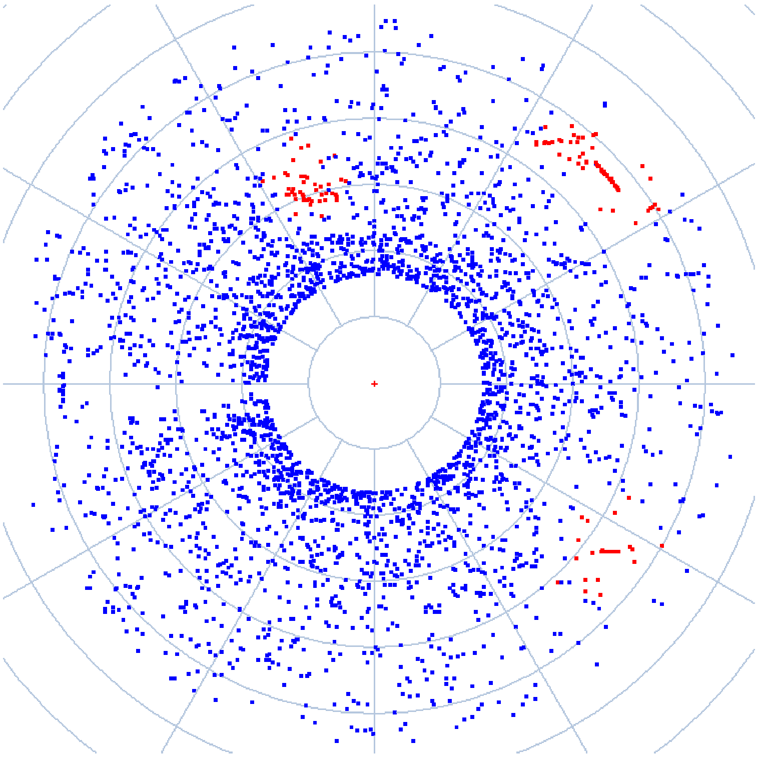

Figure 48: Real data recorded with a naval radar system over 20 scans. Blue are the original plots, red are the plots from the suspicious cut-outs. Criterion is the *sum of all eigenvalues.*

We now process the same data set with the external parameter method using the weighted *combination of all criteria*. We chose a $P_d - P_{fa}$ combination of 80% and 2% respectively. The average number of plots per-cut-out is set to 40, the resulting number of cut-outs is 342. In Fig. 49 the result is shown. The algorithm selected 16 suspicious cut-outs which are 4.7% of all cut-outs. The number of selected plots is 489 out of 3146, corresponding to 15.5% of all data. Clearly the tracks at azimuth 45° and 120° have been detected reliably. Also, at azimuth 250° there seems to be an anomaly, which looks like an interrupted track. Plot miss in this area can result from the mast shielding explained before. If the ship has made a movement

that included a rotation around its own axis, the mast shielding area has also moved since the radar data are displayed with respect to the north direction. In our data we see a relatively large shielded area, indicating that the ship has not rotated for a while. Also, the regions next to the completely shielded area are somewhat patchy indicating that the ship indeed may have performed a slight rotational movement. Thus the detected anomaly most probably belongs to the track of a moving object.
Hence we note six detections out of 16 cut-outs and ten false alarms. This corresponds to 2.9% P_{fa} which is in good agreement with the P_{fa} we set to 2%.

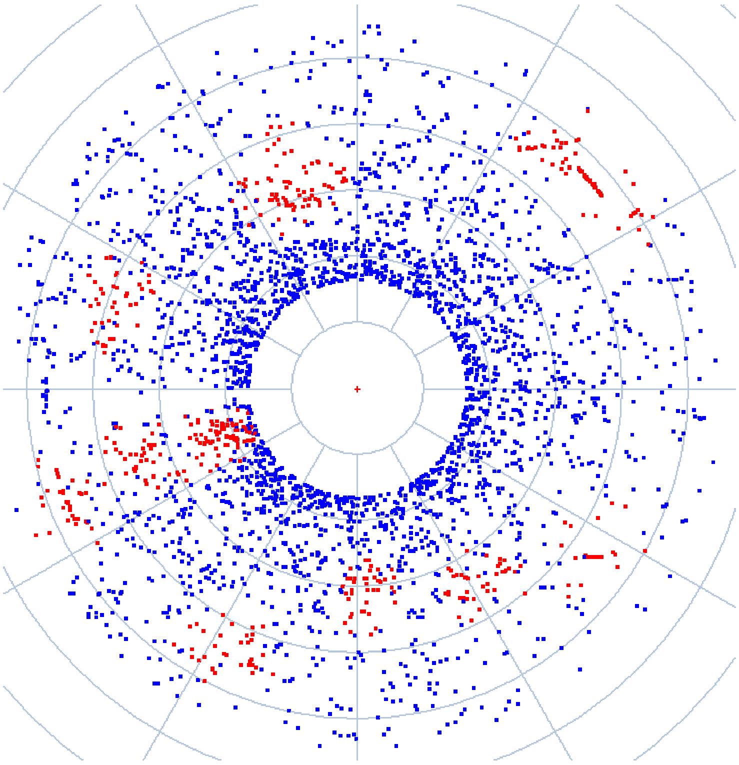

Figure 49: Real data recorded with a naval radar system over 20 scans. Blue are the original plots, red are the plots from the suspicious cut-outs. Mapping is the *weighted combination of all criteria.*

For another application data from a different radar system are available. The data have been recorded in June 2003 with another 3D EADS radar sensor.
We again select 20 scans containing 3034 plots and apply the intrinsic method. The average number of plots per cut-out is set to 45, based on our recent experience, the overlap is $\frac{5}{3}R_{sphere}$. The data are divided into 756 cut-outs, 63 in azimuth direction and 12 in range direction. We note seven selected cut-outs which are 0.93% of all sub-samples. These cut-outs contain 219 plots, which correspond to 7.2% of all plots. The results is shown in Fig. 50.

As is at once visible, two anomalies have not or only partially been detected; the track at azimuth 90° has only been detected during the manoeuvre; although, it is not important which part of a track is being detected by the algorithm, as long as it is detected at all. Also, the track at azimuth 330° at the inner dead zone edge has not been detected. Only the track at azimuth 220° was detected reliably. The number of selected data is very small and thus showing only few false detections. However, the price for the low false alarm probability is a very low detection probability.

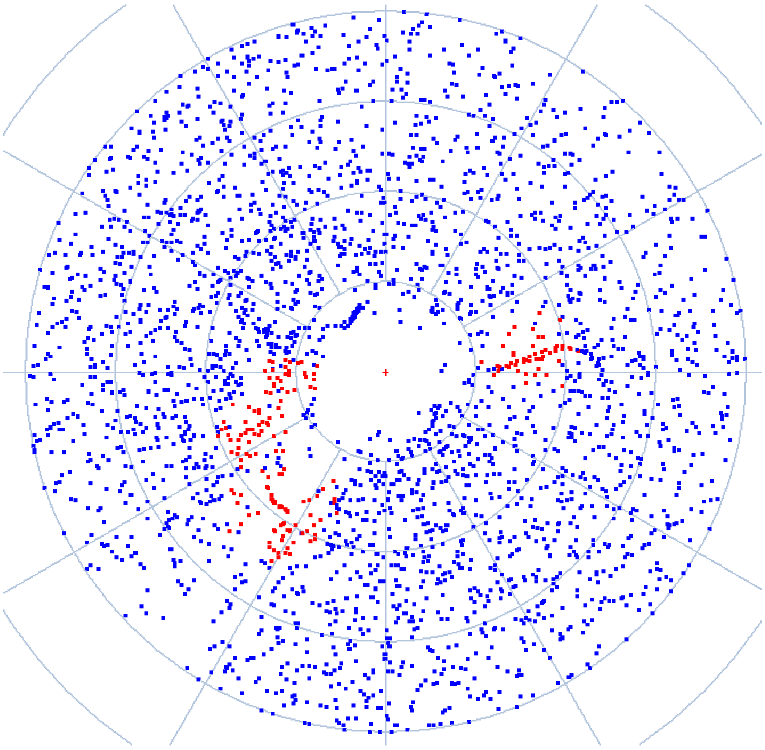

Figure 50: Real data recorded with a naval radar system over 20 scans. Blue are the original plots, red are the plots from the suspicious cut-outs. Mapping is the *sum of all eigenvalues.*

For the external parameter method we chose the average number of plots per cut-out also to be 45, but let the overlap only be $\frac{3}{2}R_{sphere}$. We set the P_{fa} to 1% corresponding to a P_d of 70%. The 3034 plots were divided into 336 sub-samples, 42 in azimuth and eight in range direction. Selected to be suspicious of containing an anomaly have been 19 sub-samples which are 5.6% of all. These cut-outs contain 575 plots which are 18.95% of the total number of plots. The result is presented in Fig. 51.

The external parameter method selected twice as many plots as the intrinsic method, but also found all tracks that can be seen with the naked eye. Thus the detection probability is by far higher than that of the intrinsic method. Of course, also the number of false detections has increased. But still, the algorithm selected less than one fifth of the original data which is a notable reduction.

Contrary to the intrinsic method, the track at azimuth 90° has been detected completely, even in the part where the object performs a manoeuvre.

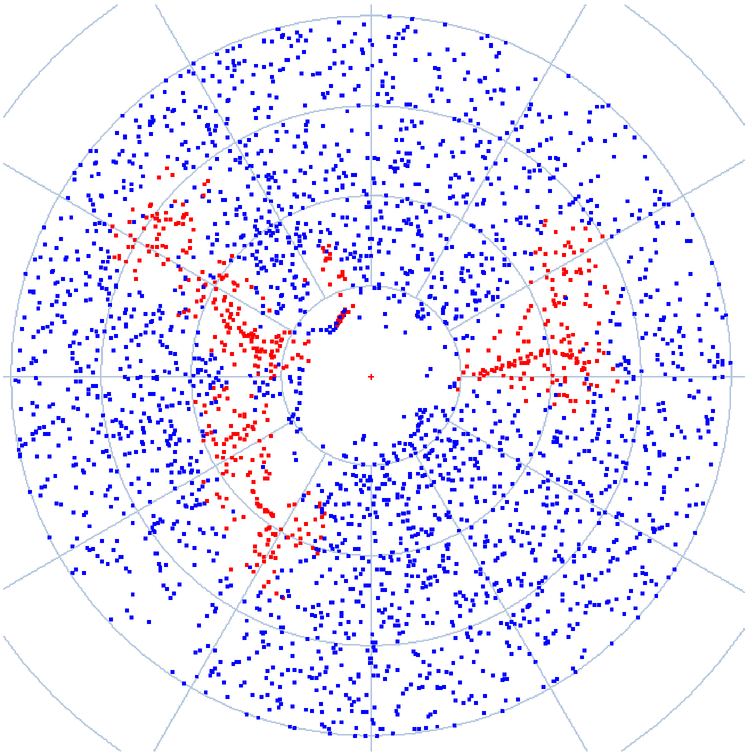

Figure 51: Radar data recorded with a naval radar system over 20 scans. Blue are the original plots, red are the plots from the suspicious cut-outs. Criterion is the *weighted combination of all criteria.*

5.2 Minimum required data integration time

Hitherto we have applied the implementations to radar data sets that had been collected over 20 scans. The number of scans was chosen as our estimate as to when the statistical deviation was significant enough to make a calculable difference. This estimation was, of course, based upon our knowledge of the characteristics of real data.

Nevertheless, the aim of our method is the fast detection of objects within the radar field of view with highest possible detection probability and the reduction of false alarms at the same time. We have already optimized the detection probability and simultaneously reduced the false alarm rate. The last remaining task is the reduction of the data integration time. We want to find the least number of scans that have to be collected in order to make a reliable statement as to the presence of an object of interest.

The tracking objective in radar surveillance systems is the fastest possible detection of objects of interest. This is important in many ways: for any naval radar to avoid collisions while cruising and for orientation while shipping close to the coast or into a harbour; for harbour surveillance systems for the optimized co-ordination in industrial harbours; for maritime patrols on search and rescue missions to find castaways as quickly as possible; for military ships to see the foe as soon as possible; for airport surveillance radars to avoid accidents and co-ordinate all flights. Thus the time required until a detection is being made with sufficient significance is next to the detection probability and the false alarm rate another critical parameter in radar surveillance systems of any kind and should be as few as possible.

In order to fulfill the task of fastest possible detection we have applied the algorithm to the previous data set, reducing the number of scans collected. As the intrinsic method already failed to work sufficiently even on 20 time-layers and we do not expect its performance to improve with less time-layers, we only apply the external parameter method. Fig. 52 and Fig. 53 give representations of the results.

The time layers were reduced by simultaneously removing the first and the last time layer of the initial 20-scan data set. Of course, the less time scans, the less data points are contained in the set. The previously detected tracks lie relatively central in time and remain so, due to the symmetrical reduction technique applied. The average number of plots per cut-out was set to 45, the overlap of adjacent cut-outs has been increased to $\frac{5}{3}R_{sphere}$, detection Probability P_d was set to 70%, coinciding with a false alarm rate P_{fa} of 1%.

Table 8 gives the data parameters, concerning number of plots, number of sub-samples, number of selected sub-samples and plots.

The average number of plots per scan is 150, thus the difference of plots between two adjacent figures is approximately 300. Table 8 reveals that with decreasing plot number the number of sub-samples decreases also. At the same time their radius increases, because the average number of plots per cut-out remains unchanged.

Surprisingly, the number of selected cut-outs per data set is relatively constant. It varies only from 4.3% to 6.1% with an average value of 5.6%. Also, the number of plots contained in these cut-outs, i.e. the selected data points is roughly one fourth of the original data in all cases. It varies from exceptionally low 13.4% for six time-layers to 26.6% with an average of 23.3%.

scans	plots	cut-outs (az*range)		cut-out radius	selected cut-outs		selected plots	
18	2721	60*11	660	0.285	32	4.9%	682	25.1%
16	2387	55*10	550	0.304	35	6.4%	636	26.6%
14	2087	51*9	459	0.325	27	5.9%	542	26.0%
12	1792	47*7	329	0.351	20	6.1%	451	25.2%
10	1495	43*6	258	0.384	14	5.4%	323	21.7%
8	1193	37*5	185	0.430	11	6.0%	304	25.5%
6	880	31*3	93	0.501	4	4.3%	118	13.4%

Table 8: Results of radar data application with reduced integration time.

Considering the results shown in the following figures Fig. 52 and Fig. 53 we note that in all but one case two anomalies were detected and one was not, this indicates that the detection probability and the false alarm rate both remain extraordinarily constant. Only the data set containing as few as six scans shows exceptions to the average values given above, thus most probably forming the lower edge of applicability of the algorithm to radar data.

In Fig. 52 we present the results for 20, 18, 16 and 14 scans (from top left to bottom right).

For 20 scans we used (as before) the overlap of $\frac{3}{2}R_{sphere}$. To obtain better results with the decreasing amount of data we set the overlap to $\frac{5}{3}R_{sphere}$ for all other data sets.
As can be seen from Fig. 52 the two anomalies at azimuth 90° (an object performing a manoeuvre) and in the mast-shaded region at 220° have been detected for all data sets. The track at azimuth 330° very close to the sensor has not been detected for data containing less than 18 scans. As the anomaly lies at the innermost edge of the processed area this also means that it lies at a great offset from the center of the innermost cut-out. Hence, only very few plots will lie within one cut-out, obviously, too few plots.
Again, the constancy of the detection probability can be confirmed. At the same time, we can also see the reduction of the total number detections, indicating a constant false alarm probability.

In Fig. 53 the number of time-layers was reduced to as far as six scans. Still, both anomalies at azimuth 90° and 220° have been detected easily in all data sets.

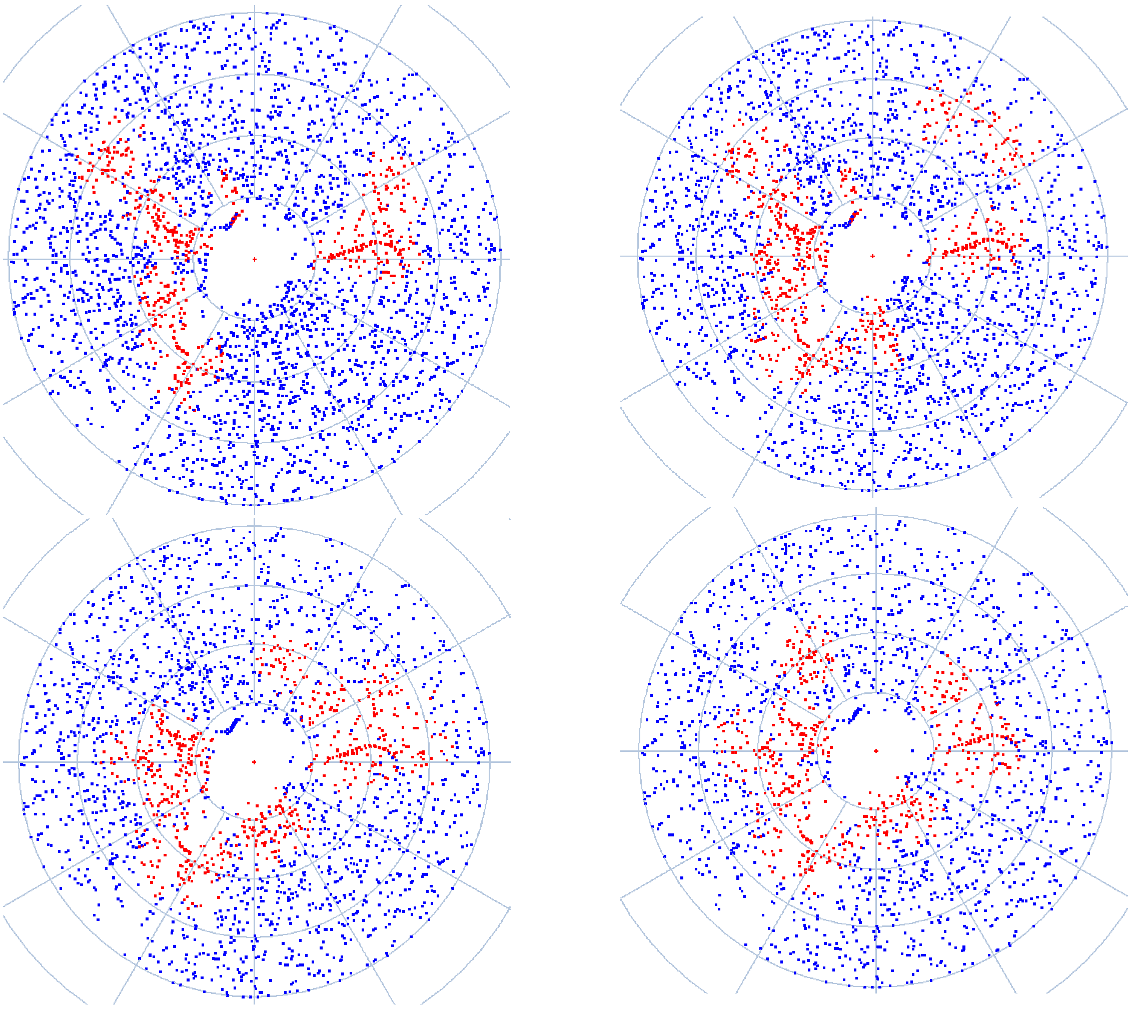

Figure 52: Radar data containing 20, 18, 16 and 14 time-layers (top left to bottom right), mapping is the *weighted combination of all criteria*. Blue are the original data, red are the selected plots.

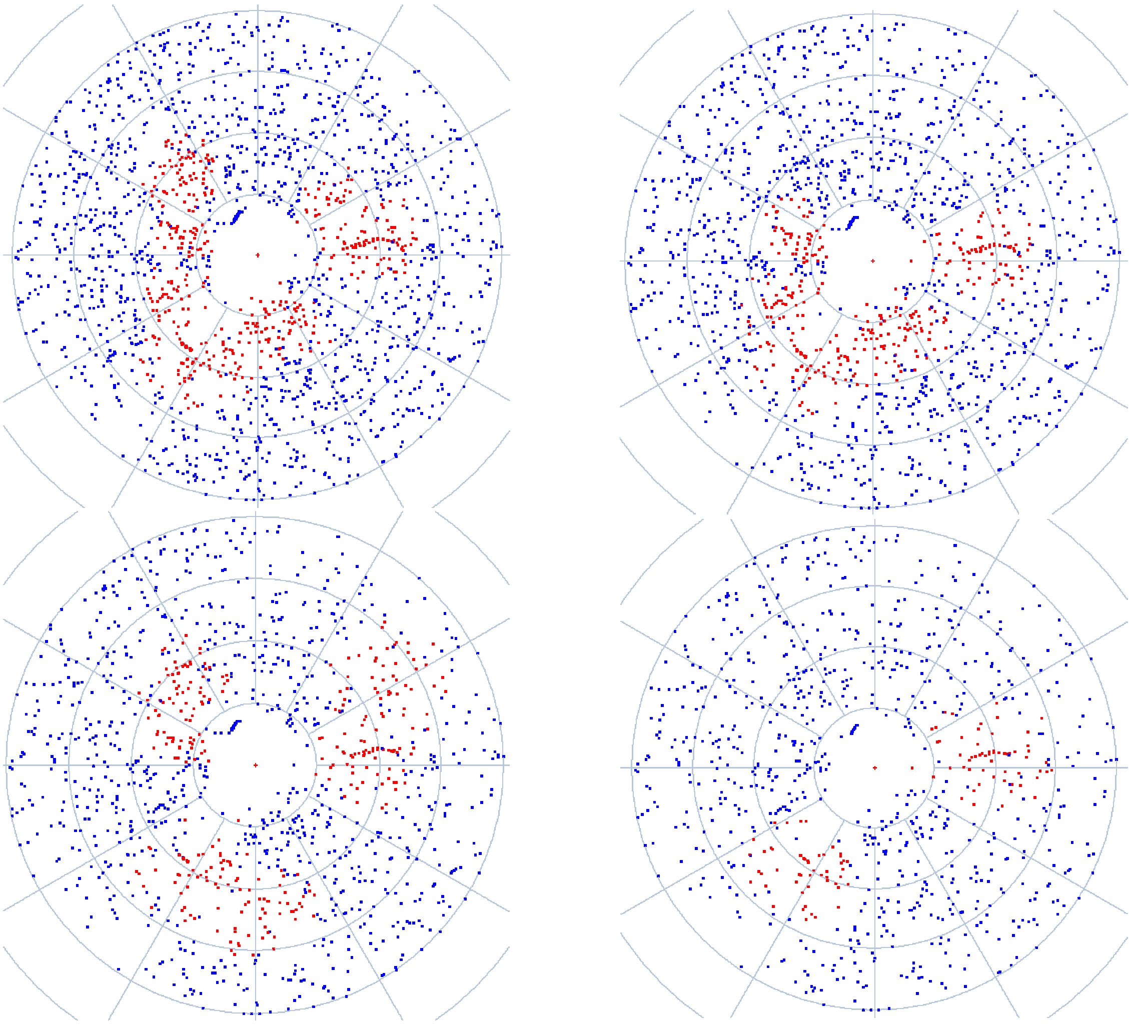

Figure 53: Radar data containing 12, 10, 8 and 6 time-layers (top left to bottom right), mapping is the *weighted combination of all criteria.* Blue are the original data, red are the selected plots.

In the set containing only six scans one can hardly see any correlation or anomaly with the naked eye. Yet, the method could not detect the one anomaly at low range and azimuth 330° . For 14, 8 and 6 time-layers we reran the algorithm using the parameters according to 90% detection probability and 5% false alarm rate, but could not achieve any improvements as to this specific anomaly.

5.3 Maximum detectable object velocities

In this section we want to discuss which objects according to which object velocity can be detected with our method. Due to the cut-outs' fixed size and the demand for n scans to be the radius in time direction, there is an upper limit to the detectable object velocity. If an object travels with $x\frac{m}{s}$ and the cut-out radius is R_{sphere}, the object will only be detected, if $n \cdot scantime \cdot x \leq R_{sphere}$ i.e. if its trajectory lies entirely within one cut-out.

In order to give an idea of the object velocities that can be detected with our method, we made a small statistic using the real data that have also been processed in the previous chapters as well as some other data sets. For this statistic we set the average number of plots per cut-out to 45, which has proved to be a reasonable value for the processing of real data. This is the only parameter that has any influence on the size of the cut-outs apart from the clutter-density.

For cut-outs containing 20 scans we obtained and average radius of 0.27 (after transformation). This corresponds to an azimuth area of 15° and an average radius in range of 600 m. According to the covered time of 60 s this means the maximum detectable (radial) object velocity is 600 m/60 s=10 $\frac{m}{s}$. The velocity spectrum of sea clutter typically ranges from 0 - 15 $\frac{m}{s}$. With an average detectable velocity of 10 $\frac{m}{s}$ we thus lie within the velocity spectrum of clutter. We hence can detect exactly those objects that do not differ from clutter in Doppler velocity and can thus not be detected with common tracking methods.

For further information we also use the results from Table 8. We use the obtained radii and integration times to obtain the maximum detectable object velocity for every integration time.

scans	radius	range $[m]$	maximum object velocity $[\frac{m}{s}]$
18	0.285	627	12
16	0.304	670	14
14	0.325	715	17
12	0.351	770	21
10	0.384	846	28
8	0.430	947	40
6	0.501	1100	60

Table 9: Maximum detectable object velocities in dependence on the number of scans per cut-out.

As can be seen from Table 9 the cut-out radii increase with decreasing number of scans, as the average number of plots per cut-out remains fixed. Simultaneously the maximum range also increases and with it the maximum object velocity. This means the less scans we integrate and the higher we set the average number of plots per cut-out the higher is the maximum detectable object velocity.

Of course, with decreasing number of scans per sub-sample the more important it is to provide a sufficient overlap of adjacent sub-samples in order to guarantee that even the trajectories of fast objects lie diagonally in at least one cut-out which is the position providing the highest detection probability.

Objects that are interesting to be tracked with a naval naval radar systems include buoys (0 $\frac{m}{s}$), castaways (0 - 0.5 $\frac{m}{s}$), small boats (3 - 5 $\frac{m}{s}$), small ships (4 - 7 $\frac{m}{s}$), average ships (3 - 9 $\frac{m}{s}$), military ships (8 - 17 $\frac{m}{s}$), helicopters (20 - 80 $\frac{m}{s}$), aero-planes (80 - 200 $\frac{m}{s}$) and - for military applications - missiles ($\approx$ 600 $\frac{m}{s}$). Apart from the latter two our method is capable of detecting all interesting objects and especially those invisible to common systems in high false alarm environments.

g to be tracked with a naval naval radar systems include buoys (0 $\frac{m}{s}$), castaways (0 - 0.5 $\frac{m}{s}$),

small boats (3 - 5 $\frac{m}{s}$), small ships (4 - 7 $\frac{m}{s}$), average ships (3 - 9 $\frac{m}{s}$), military ships (8 - 17 $\frac{m}{s}$), helicopters (20 - 80 $\frac{m}{s}$), aero-planes (80 - 200 $\frac{m}{s}$) and - for military applications - missiles ($\approx$ 600 $\frac{m}{s}$). Apart from the latter two our method is capable of detecting all interesting objects and especially those invisible to common systems in high false alarm environments.

5.4 Functionality of our method in a radar tracking system

As already mentioned in the introduction there are two major and common disadvantages of existing methods for track detection or track initialization in strongly disturbed surroundings. The first disadvantage is, that most methods use measured object velocity and signal intensity as main detection means. The spectrum of both is similar to that of clutter. Hence, common distinguishing methods are of few use in disturbed environments. The second is the large computation time they require. The data association methods are based on finding an association between one data point and all other possibly correlated ones. If N data points are to be associated to $N-1$ others this leads to computational time of $\geq O(N^2)$. Thus these methods are not unrestrictedly suitable for real-time applications.

Contrary to other methods we make neither use of Doppler velocity nor signal strength. We also make no attempt concerning data association. Instead we are able to detect sub-sets of integrated data that contain other than clutter alone, most probably the trajectory of a moving object.
The advantage that arises from these contrasts is that our method can be used exactly then when the common ones fail. Vast amounts of clutter have no influence whatsoever on the performance of our method. As we do not associate data the computational time required is only of O(N), which makes the method especially suitable to process huge sets of data as they occur in clutter areas.
Another advantage is the high flexibility of the method. When using the external parameter implementation as suggested on grounds of reasons found in the previous sections (full exploitation of all criteria, best possible P_d-P_{fa} combination) the operator can chose the parameters and the according detection probability and false alarm probability at her own will. If the operator e.g. increases the detection probability she does so knowing exactly how much the false alarm rate will increase. It is fully at the operator's license to choose the P_d-P_{fa} compromise she finds most apt for the current situation. Additionally, the once chosen combination may be changed at any time. Another parameter that can be changed during the operation is the overlap of adjacent sub-samples. The larger this overlap, the more data have to be processed and the longer it takes to achieve a result. If only a quick estimate is needed, the overlap can be reduced in one or all dimensions; if exact predictions are required, the overlap can be increased as necessary. In addition to all this, also the area to be investigated can be chosen freely. Since the sub-samples are formed in adaption to the data to be processed it is possible to investigate any area in detail, if wished. Another surplus is that the average number of plots per cut-out can also be used as a free parameter. This would allow an additional adaption to the actual clutter density. If the clutter density is very low, the average number of plots per cut-out can be reduced within reasonable limits in order not to enlarge the cut-outs too much. Because if they are too large in range direction, the investigated area may not be large enough to allow for several overlapping cut-outs. As the overlap is an important feature of the method it is highly

recommendable to achieve its realization. It may in certain situations be hence of advantage to vary the average number of plots per cut-out according to the actual clutter density. Of course, not all or even none of these parameters needs to be implemented as such, this is just an optional possibility. There are also reasonable default values for all parameters with which the method will also work fine.
What seems to be of disadvantage with our method is that it requires integrated data from several scans. But as we have shown in the previous section, we can make remarkably good statements as to the presence of object trajectories after only six scans. Common methods for track initialization in clutter also demand six ore even more subsequent associations of certain quality before they initiate the object trajectory, in order to avoid false associations and false tracks. We thus do not require more time than the standard methods, although, of course, our method works the better the more scans are available. The number of scans to be integrated can also be an operator definable parameter.

In comparison to other track initiation methods we must, however, admit, that our method does not exactly initiate object trajectories. It does not even find the trajectories themselves but only the small area defined by the cut-out within which the trajectory most probably lies. Hence, the method cannot be used for track initiation itself.

The application is implemented into an EADS naval radar system as a *noise reduction method.* The radar data processing chain works as show in Fig. 54: The antenna sends and receives an analogue signal, that undergoes various preprocessing measures (e.g. amplification, mixing to lower frequencies, digitalization). The preprocessing unit provides the digitalized signal as input data for the tracking device in which the data association is being performed. The last unit in the processing chain is the displaying device that shows the results from the tracker.
In the same figure a close-up of the tracking unit is being shown: Incoming digital data (plots) are tested for possible *association* with already existing tracks. If a plot can be assigned to an existing track it is used to *update* the according track. If the plot cannot be assigned to any existing track, it is stored in a database with other data from previous measurements that could also not be associated with the existing tracks. Within this database the data are again tested for correlation. If there are correlations and data belong to one object a new track is being *initiated.*
The updated or newly established tracks are sent to the displaying unit where they are visualized and made available for the operator.

It is exactly the track initiation that has to deal with the mentioned data association problems. We thus want our algorithm to help reducing the noise, i.e. the uncorrelated clutter data there. The input data for our noise reduction method is the database of un-associated data from the last scans upon which the track initiation algorithms work. Our noise reduction algorithm will then process these data parallel to the common initiation methods. It will select the suspicious data sub-sets that (with certain probability) contain the tracks formed by all hitherto un-detected objects. The selected data are then additional input data for the common initiation unit, available after several scans of data integration. The noise reduction method will be applied multiply: one quick-checking version, processing data from only very few (n) scans, and in addition a more diligent version working on a larger number (m) of scans. Both versions will make their results available to the common initiation every n and m scans. There, the tracks are established without difficulty from the reduced number of previously un-associated plots. Finally, the new tracks are sent to the displaying unit.

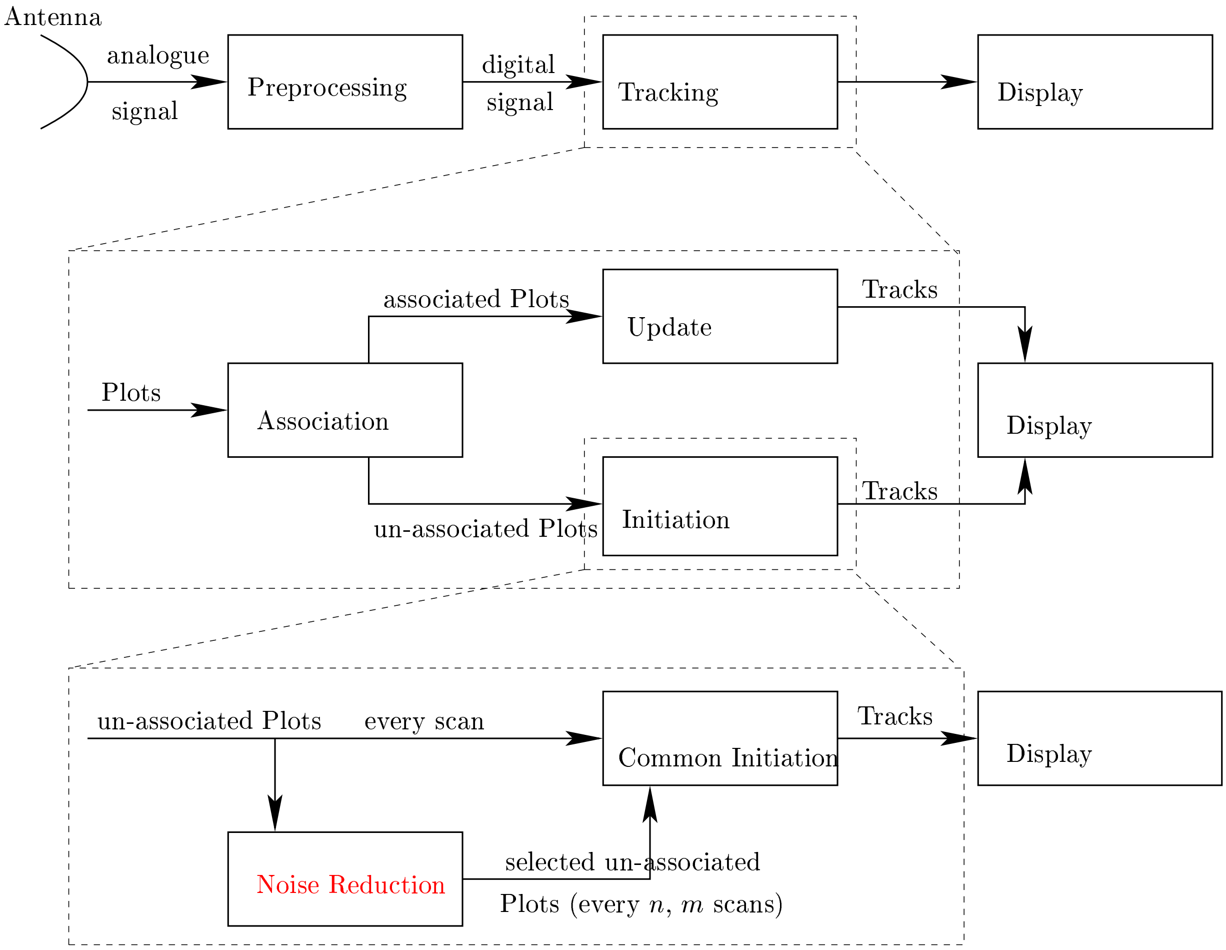

Figure 54: Possible implementation of the algorithm into the data processing chain.

6 Summary

While searching sets of discrete data contaminated heavily by noise for weak signals of correlated data points we have made use of statistic quantities to detect correlations between data points. The signal consists of linearly correlated three dimensional discrete data, forming the trajectory. The noise data are independent, uniformly distributed three dimensional random vectors, originating in spurious data whose origin is not known to us.
The method is widely usable whenever the state of a stochastically driven dynamical system is to be estimated from a series of sensor data sets received at discrete instants of time, i.e. when three dimensional discrete noise corrupted data are being searched for trajectories.

We divide the data set into representative sub-samples and map the data contained in such a sub-set to one value. Mappings we investigated include the eigenvalues of the covariance matrix, the dispersion of the angle between time and spatial axes, the Kolmogorov-Smirnov one-sample test and linear regression. Using Monte Carlo simulation we obtained probability density distributions for all mappings and for both cases 'noise' and 'noise plus trajectory' each. As we test the data sub-sets for the hypothesis 'noise', we obtained the statistical type I and II errors as an indicator for the quality of a mapping. The type I error corresponds to the false alarm probability, i.e. the detection of a non-existent trajectory, and the type II error corresponds to 1 minus the detection probability, which is the probability of not detecting an existent trajectory. The aim is to achieve the highest possible detection probability while reducing the false alarm probability at the same time. We obtained receiver operating characteristics (ROC) which shows the detection probability as a function of the false alarm rate. This curve is an indicator of the quality of any detection system. In order to obtain the ROC we simulated 10^7 cut-outs containing 'noise' and 'noise plus trajectory', mapped each sub-sample to one value and compared the probability density distributions for the two cases. From these distributions we obtained the statistical type I and II errors corresponding to the false alarm probability and 1-detection probability.
To maximize the detection probability while reducing the false alarm probability at the same time we composed a weighted combination of all mappings. Each single mapping contributes to the weighted combination according to its quality. Even mappings that are clearly not suitable as a stand-alone detector contributes and improves the performance of the combination.

A huge unknown data set is now processed as the following: We divide the data set into representative sub-samples the size of which is adapted to the volume and density of the data set in a way such that the average number of data points per sub-sample and the expected average signal to noise ratio are constant. Also, we assume the noise to be *locally* uniformly distributed. Should this be not the case, we suggest a simple data transformation in terms of stretching or compressing the data dimensions until the demand for *local* uniform distribution is fulfilled..
We start mapping the first sub-sample to one value using the weighted combination of all data according to the chosen P_d-P_{fa} values. We compare the calculated value with a previously fixed threshold (here: 1.0) and decide that the sub-sample contains only noise, if the value exceeds this threshold. The next sub-sample is composed with a certain overlap of the previous one in all data dimensions, in order to achieve highest possible detection probability.

So out of a vast amount of data, we select small sub-samples that with given/chosen probabilities contain all statistical anomalies caused by the correlation of data points which belong to a trajectory.

The method has been implemented as an algorithm on static data sets. It only requires computation time of $O(N)$, where N is the number of data points to be processed. It is thus very fast and suitable for real-time applications in radar systems.
We understand that it is not a stand-alone track initiation device. The implementation we think of is as an asset or pre-processing to the standard tracking method: in high false alarm environment the algorithm could pre-select suspicious regions and return the selected data to the tracking input. The significantly reduced number of noise data could be re-processed with the common methods. We can thus help to improve the performance of common tracking systems in high false alarm environment as well as decrease the computational time required when huge amounts of data are to be processed.

An application to radar data obtained with a naval rada system have shown satisfying results to prove the applicability of the method to real data.
Radar tracking in high false alarm environment is an important and challenging field that encounters many difficulties. The tracking objective of collecting sensor data from the field of view, find objects of interest and establish an estimated state trajectory is rendered significantly more difficult under the presence of huge amounts of uncertain measurements. The three major problems of track detection in disturbed environment are:

1. The data association of a vast number of uncertain detections is not only very slow it also leads to a great number of false alarms.
2. The backscattered signal amplitude, an important distinguishing feature while detecting objects, is of the same size for the objects of interest and the noise for small and radar-dim objects.
3. The Doppler shifted frequency spectrum to measure the object velocity is the same for slowly moving objects of interest and the uninteresting returns (clutter).

While implementing and applying our method all these problems were of no significance and could be undermined.

Further applications we think of, could include the detection of comets in the night sky: in integrated observations from several nights the fixed stars would accord to noise while moving objects like planets or comets had left a trajectory that could quickly be found.

A The software - an object oriented program in C++

In order to process the data, simulated as well as real we have designed, written and used object oriented software. The programming language used is C++. In the following we will shortly describe the design of the program as well as the main functionalities.

We have designed seven classes to generate the required data for the Monte Carlo simulations and to calculate the required mathematics. The 25 most important parameters that were changed very often can be edited with a parameter file, in order to save re-compiling of the code.
For Monte Carlo simulation we had to generate the input data. The main data types we defined are the three dimensional data vectors and the cut-outs. The data vectors are randomly drawn within a defined radius using the C++ intern random number generator. For this specific data type we also defined operations like addition/subtraction to calculate the difference between two data vectors (to obtain the geometric moments) and comparative means. The other major data type are the cut-outs that are filled with many of the random vectors. We defined various cut-out configurations: cut-outs containing noise alone, noise plus diagonal trajectory, noisy trajectory, two trajectories (crossing and parallel), trajectory at offset from the center. This class also contains the calculation of all mappings and the comparative method that finally decides whether a cut-out is suspicious of containing a trajectory or not.
For the probability distributions obtained by Monte Carlo simulation we defined a histogram class, that sorts the calculated values into bins; from two histograms of the according distributions for the two models ‘noise’ and ‘noise plus trajectory’ we defined a functionality to calculate the statistical type I and II errors. Another class was defined for the calculations concerning the covariance matrix. It defines objects of the matrix type. The data points of one cut-out are stored by the sum of their mean quadratic distance to the cut-out’s center (i.e. the covariance) in each direction. These covariances are the input of the matrix class which then uses the Householder algorithm for tridiagonalisation and returns the eigenvalues.
For minor mathematical operations we have an additional class containing self-defined mathematical tools.
The processing of real data made another class necessary, as for the adaptive calculations it is no longer sufficient to store the incoming data by their covariances, but we have to store each incoming data point in an array. Knowing their number and the volume thy cover we can then calculate the data-adaptive parameters, perform the co-ordinate transformation, process the data and return the selected data points.
Overall, we have defined the following classes containing the listed objects and described methods:

1. `plot`

   - object: three dimensional random vector `plot`
   - methods:
     creates three dimensional vector
     calculates distance to another vector
     compares size of two vectors
     creates three dimensional vector within spherical limits of given radius

2. `orb`

   - object: local spherical sub-sample `cut-out`
   - methods:
     creates sphere of given radius
     fills sphere with given number of random vectors
     fills sphere with chosen configuration (diagonal trajectory, noisy trajectory, trajectory offset from center, two parallel trajectories, two crossing trajectories, trajectory of any number of data, randomly rotated diagonal trajectory)
     calculates mappings of sub-sample (eigenvalues of covariance matrix, dispersion of cosine of the angle between data vectors and chosen axis, Kolmogorov-Smirnov test of cosine of the angle between data vector and given axis, linear regression in given dimension)
     decides, whether cut-out is suspicious using intrinsic method
     decides, whether cut-out is suspicious using external parameter method
     return data points belonging to suspicious cut-outs

3. `global_data`

   - object: data array `global_data`
   - methods:
     creates dynamical data array
     fills the array with random vectors
     fills array with data read from a file
     calculates parameters for co-ordinate transformation
     performs co-ordinate transformation
     calculates grid on which the cut-outs are moved over the data

4. `histogram`

   - object: histogram `histogram`
   - methods:
     creates histogram with $n-2$ bins between two values min and max plus two bins for all values smaller than min or greater than max
     inserts values into according bins
     creates probability density distribution
     calculates expected value, dispersion, skewness and kurtosis of distribution
     calculates intersection of two histograms assuming Gaussian distribution; in order to solve the problem of polyvalence, as there may occur multiple intersections, the intersection is also calculated using polygons that connect the histogram bins; the polygon intersection closest to the Gaussian one is returned.
     calculates statistical type I and II errors
     calculates value of mapping corresponding to certain type I ore type II error

5. `matrix`
   - object: symmetrical matrix `matrix`
   - methods:
     creates symmetrical data array
     uses Householder method for tridiagonalisation
     calculates eigenvalues
     assigns the attributes min, mid max to eigenvalues according to their value

6. `parameter`
   - object: changeable values `parameter`
   - methods:
     reads parameters from parameter file
     writes parameters to parameter file

7. `mathtools`
   - methods:
     returns the greater of two values
     returns the smaller of two values
     signum function
     pythagoras (returns square root of sum of two squared values)

B Generating the trajectory

The way the trajectory is simulated in the cut-outs has great effect on the performance of the criteria.
As our investigations are based on statistical analyses, it is utmost important to ensure that the way the trajectory is made up has no influence whatsoever on the result.

Our first approach was to choose the center of the cut-out as a first data point of trajectory and according to the number of correlated data points we wanted to lie in the cut-out we chose the three dimensional increment between the plots. The same was done if the trajectory lay at a certain offset from the center. The respective cut-out configurations are displayed in Fig. 56.

We encountered a weird phenomenon while investigating on the greatest eigenvalue: up to a certain offset its mean for the model 'noise plus trajectory' was **smaller** than the mean for the model 'noise' (cf. Fig. 55). For the next offset incremented by 1, it was just the other way around: the mean for the model 'noise plus trajectory' was **greater** than the mean for the model 'noise' (cf. Fig. 55). This phenomenon occurred for several offsets.

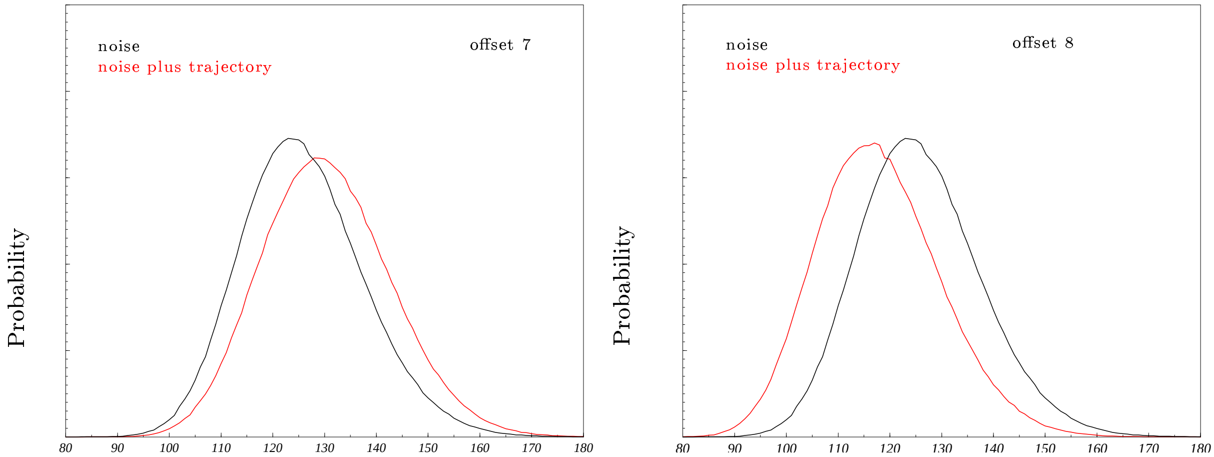

Figure 55: Probability density function for the greatest eigenvalue of the covariance matrix . Left panel: trajectory at offset 7 from the center. The mean for the model 'noise plus trajectory' (red) is **smaller** than the mean for the model 'noise' (black). Right panel: trajectory at offset 8 from the center. The mean for the model 'noise plus trajectory' (red) is **greater** than the mean for the model 'noise' (black).

Another phenomenon occurred while investigating on the criterion *sum of smallest and middle eigenvalue* in dependence on various trajectory offsets: the graph we obtained was not continuous (cf. Fig. 56). At certain offsets the detection probability decreased enormously. This phenomenon occurred only with the undisturbed trajectory.

We found out that these effects occur due to the way we generated the trajectory: while moving the trajectory towards greater offsets, at certain offsets correlated data points do not lie in the cut-out any more. This is illustrated by the red trajectory in Fig. 56. Thus the ratio correlated to uncorrelated data decreases in discrete steps for the same offsets in every cut-out. As shown in Section 3.5.4 a decrease of this ratio leads to a decrease of the detection probability.

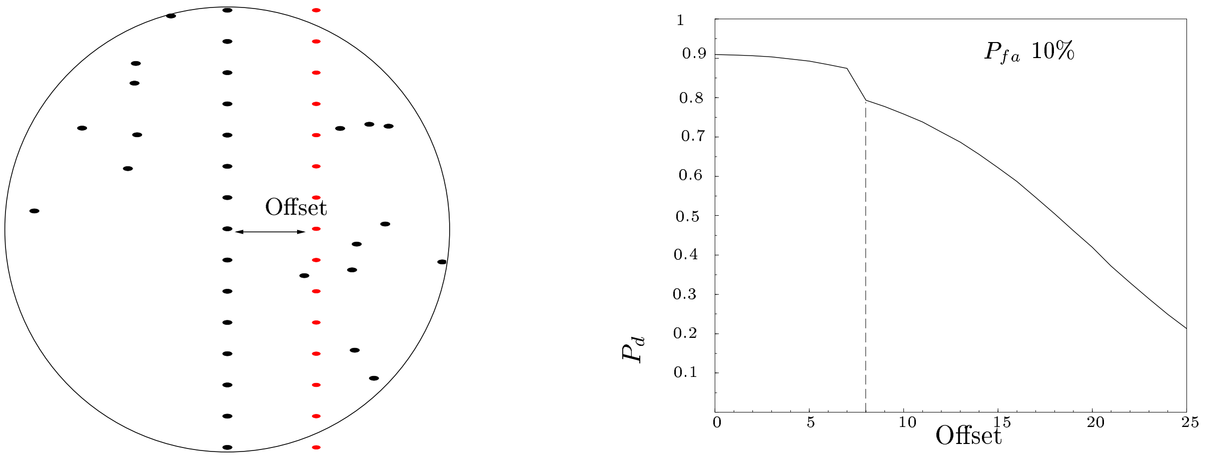

Figure 56: Left: Undisturbed trajectory through center and for one offset (red). For this certain offset two plots of the original trajectory do not lie within the cut-out any more. Right: The detection probability P_d shows discontinuities for a certain offset. The reason for the sudden decrease in P_d is the change in the ratio of correlated to uncorrelated data as shown left.

In order to remove these artefacts we changed the fashion in which we generate the trajectory to a more random one.
As first trajectory data point we no longer choose the center of the cut-out. We start the trajectory at the (three dimensional) point

$$first\_trajectory\_plot = centre_{Sphere} + offset_{Trajectory} - \Delta \cdot (R_{sphere}/\Delta_{max} + random\_num) \ ,$$

where Δ is the three dimensional increment vector between the correlated data points and Δ_{max} its largest component.

We thus choose the center of the cut-out ($centre_{Sphere}$), move to the offset position ($+offset_{Trajectory}$) and move this data point outside the cut-out: we divide the cut-out's radius (R_{sphere}) by the maximum increment (Δ_{max}) and obtain the maximum number of data points that can lie in the cut-out in the according dimension. To this number we add a random number ($random\_num$) $\in [0, 1]$) and multiply the result with the increment vector Δ.
To the previous offset position this new vector $\Delta^* = \Delta \cdot (R_{sphere}/\Delta_{max} + random\_num)$ is added. The result is then the $first\_trajectory\_plot$.
This measure guarantees that the starting value for the trajectory is different for every single cut-out. As shown in Fig. 57 this leads to the effect that the number of correlated data per cut-out varies. So now the ratio of correlated to uncorrelated data is not constant for a constant offset. This measure also inhibits the rapid changes in the signal to noise ratio at certain offsets mentioned above.
The detection probability P_d is now a smooth function of the trajectory offset (cf. Fig. 57).
In addition to this for trajectories through the cut-outs center we rotated the trajectory for a random angle in order not to prefer one direction.
As a clean trajectory does not represent real data very well, we add noise to every trajectory we generated. We do this by not only adding noise to the first trajectory point but also by adding a three dimensional random vector to every trajectory point. This random vector is generated

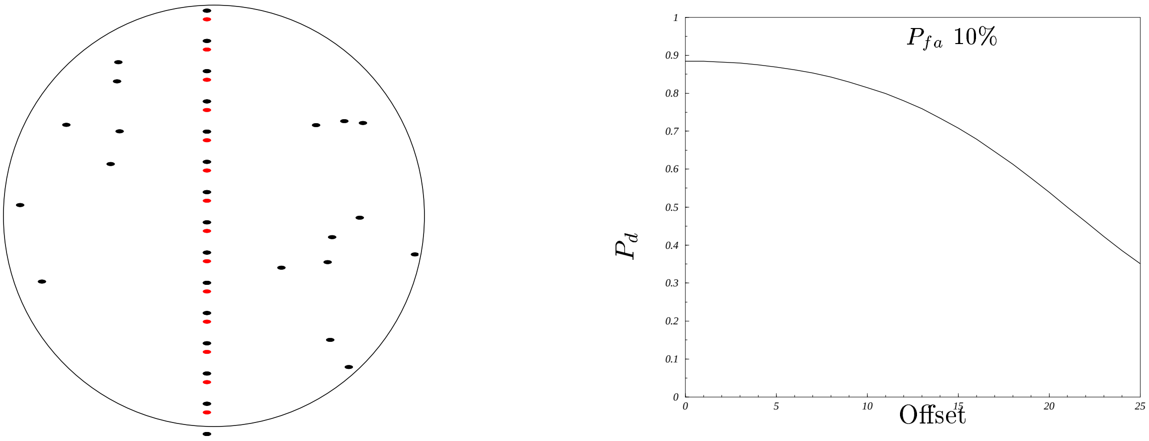

Figure 57: Left: Cut-out containing clean trajectories through the center whose first point was generated randomly. The number of correlated data for the two trajectories is not identical. Right: The detection probability P_d is now a smooth function of the offset due to the varying signal to noise as shown left.

within a certain radius, the noise radius.

The procedure is illustrated in Fig. 58, where black is the clean trajectory and red the noisy trajectory. The noise radius is indicated by the red circles. In the topmost circle the random vector is represented by the red line.

For our simulations we used a *noise radius* of 4% of R_{sphere}.

In Fig. 58 we show what noisy trajectories look like. The additional noise amplifies the effect we already provoked with the noisy first trajectory point: the number of correlated data that lie within the cut-out for a fixed offset is about the same but varies randomly for about ± 2. This can also be seen from Fig. 58: two noisy trajectories are offset at the same distance from the center and yet the number of correlated data in the cut-out differs by 1.

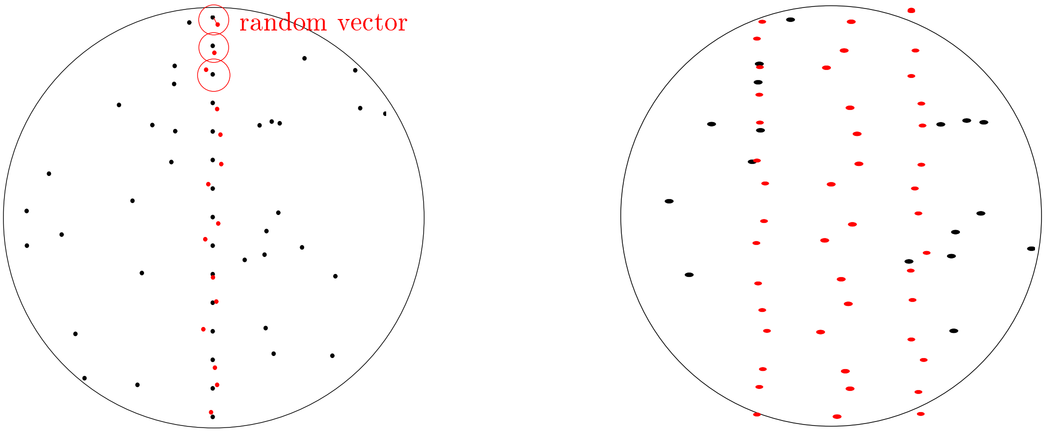

Figure 58: Left: In a noisy trajectory a three dimensional random vector (red) generated in a sphere of radius *noise radius* is added to each trajectory point to form the noisy trajectory as drawn in red. The degree of noisiness depends on the *noise radius* indicated by the red circles; the larger it is the noisier the resulting trajectory. Right: Cut-out containing noisy trajectories through the center and at the same offset on each side of the center. The number of correlated data for the two offset-trajectories is not identical.

This simulation comes closer to reality and at the same time removes the artefacts described previously.

C The Downhill Simplex minimization method

To build the best detector possible we combine n detection criteria to one n-dimensional criterion:

$$f(cut-out) = \sum_{i=1}^{n} \alpha_i \cdot f_i(cut-out) \ .$$

For each cut-out we calculated the values for the single criteria which form an n-dimensional topography in which we want to find the global minimum, or at least the best local minimum. We thus have to find the combination of weighting parameters α_i that results in maximum P_d under the demand of simultaneously achieving a given P_{fa}. To find this maximum (equal to the minimum of the type II error) we use the Downhill Simplex Minimization Method.

The Downhill Simplex Method (cf. [Nel69]) is a method to minimize (or maximize) a multidimensional function without computing first derivatives. It makes almost no assumptions about the function under discussion and is said to be extremely robust.

First of all we generate the data: we generate 10^6 cut-outs with noise alone and 10^6 cut-outs with noise and trajectory. For each cut-out we calculate the n single criteria, like eigenvalues, dispersion of cosine etc and store the values in two data arrays, consisting of $10^6 \times n$ entries each.

To start the algorithm, it needs a first simplex as input. A *simplex* in n dimensions is the geometrical figure consisting of $n+1$ points in any $n+1$ dimensional subspace, e.g. in three dimensions the simplex is a tetrahedron.

The only but utterly important influence one has on this algorithm is the choice of the starting values. After starting the algorithm it makes its own way through the data.

We developed a rather sophisticated version of drawing a 'grid' of random numbers across the range of the function. We selected the one with the smallest function value found and built the simplex around this value. We drew approximately $n{\cdot}10^4$ random numbers and were able - in comparison to earlier tries - to reduce the probability that the algorithm got stuck at a local minimum. Whenever that happens, the resulting graph shows zig-zag features, falling back to lower values for the P_d that clearly belonged to a single criterion.

This effect is shown in Fig. 59 where we see in black the criterion *sum of smallest and middle eigenvalue* which we wanted to minimize in combination with the *dispersion of the cosine of the angle between plot and time axis.*

The first try without choosing the starting values through the random-grid led to the result drawn in red. The curve shows strong zig-zag features whenever the algorithm got stuck at a local minimum. E.g. at the marked points x, y and z it only found some local minima of the eigenvalue criterion instead of the best combination. In some cases it even got stuck at worse local minima.

For the second try we drew 5000 random plots and chose the smallest one of them to have the starting simplex built around. This led to the clearly improved result drawn in blue. The smooth optimum curve we would like to reach is the dashed line drawn in green.

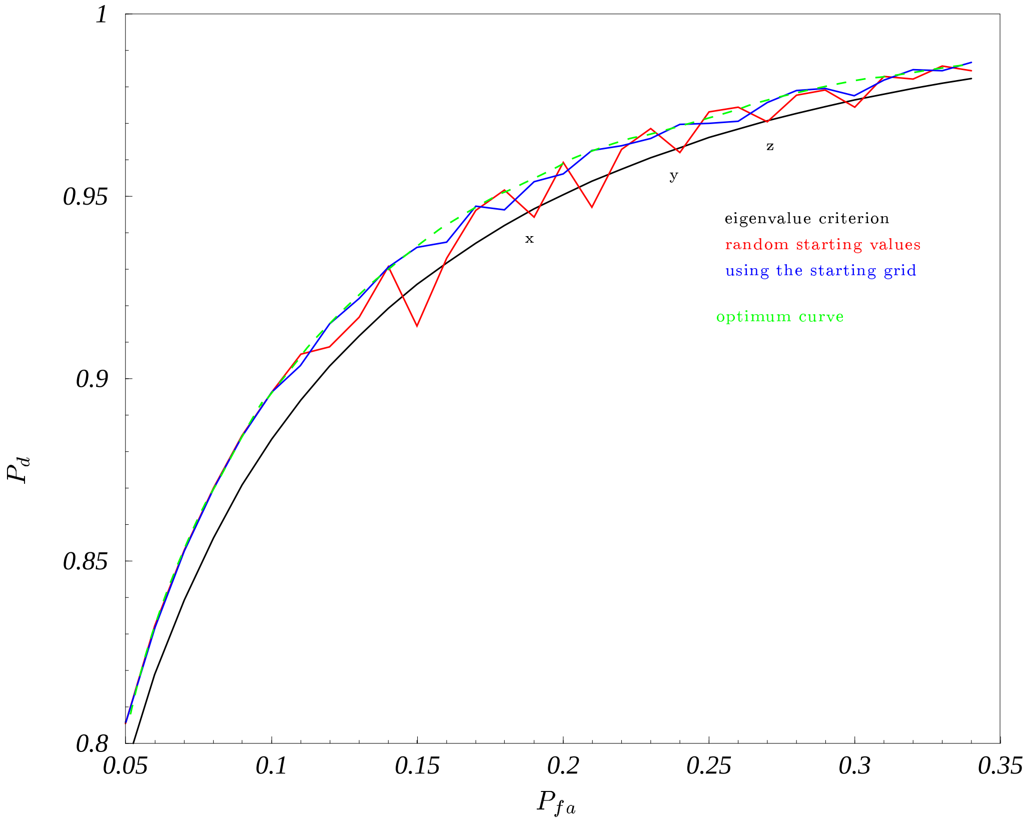

Figure 59: Example of ROC during the minimization with the Downhill Somplex method.

Additional inaccuracies occured when setting the starting values for the weighting parameters. The function to be minimized uses a detection criterion of the form

$$f(cut - out) = \sum_{i=1}^{n} \alpha_i \cdot f_i(cut - out) < 1.0 \ .$$

The data we investigate consist of the two $n+1$ dimensional arrays mentioned above, consisting of all n investigated criteria for 10^6 cut-outs. In order to set the false alarm probability, we give a starting guess for the n parameters α_i, apply the equation above to the data without trajectory and count the number of times we detect a non existent trajectory in the array 'noise' and count the number of times we failed to detect the trajectory in the array 'noise plus trajectory'. Set to the total number of cut-outs the latter gives us the type II error (1-P_d) and the former gives us the P_{fa} for the one combination of α_i.

In order to avoid any distortion of the result we do not calculate the plain type II error. We also consider the possibility, that the type I error has not been set exactly to what we wanted it to be.

We thus set the function to be maximized

$$P_d(\alpha_i) = P_d^*(\alpha_i) - a \cdot |P_{fa}^*(\alpha_i) - P_{fa}| \overset{!}{=} Max \ .$$

We choose a value for P_{fa}, try an α_i combination and obtain $P_d^*(\alpha_i)$ and $P_{fa}^*(\alpha_i)$ as described below. If for this specific parameter combination the calculated P_{fa}^* deviates from the chosen P_{fa}, the calculated P_d^* is worsened by a $a \times |P_{fa}^* - P_{fa}|$, where a is a factor influencing the strictness of the 'punishment'.

This means the larger the difference between the demanded and the achieved type I error the greater the resulting type II error and hence the smaller the detection probability P_d. So on its search for the maximum we simultaneously force the algorithm to achieve a certain false alarm probability P_{fa}.

In order to also reach the highest accuracy, for each type I error we vary the factor a from a relatively small value (e.g. 0.2) in several steps to a large value (e.g. 10).

For further discussion we present in Table 10 some of the results of the six dimensional combination of all criteria (cf. Section 3.8).

α_1	α_2	α_3	α_4	α_5	α_6	P_{fa}	P_d
12.257	10.5808	5.02779	-5.16858	-0.319143	-4.63608	0.011945	0.714697
13.0501	11.2988	5.41918	-5.65139	-0.338247	-5.03245	0.011609	0.709454
13.3082	11.5374	5.73987	-5.91139	-0.341478	-5.29209	0.015001	0.747771
12.0066	10.1791	5.01208	-5.05341	-0.309003	-4.62778	0.02	0.791634
12.2551	10.4296	4.97326	-5.06217	-0.308648	-4.87497	0.025	0.822311
13.7142	11.4836	5.60797	-6.02076	-0.347459	-5.45966	0.03	0.846126
13.0522	11.0136	5.28806	-5.74338	-0.327337	-5.11662	0.035	0.865058
13.4454	11.0133	5.72202	-5.92702	-0.339521	-5.41909	0.04	0.880129
13.3389	11.0471	5.79025	-5.89744	-0.33925	-5.51947	0.045	0.893097
12.0046	9.90245	5.07341	-5.0872	-0.285429	-4.84322	0.05	0.904136
12.461	9.98779	5.23863	-5.21182	-0.307615	-5.05988	0.055	0.913378
12.8092	10.3152	5.60528	-5.59914	-0.321756	-5.25086	0.06	0.921538
13.0456	10.7204	5.63688	-5.76871	-0.344348	-5.41132	0.065	0.928708
12.164	9.53036	5.20921	-5.09715	-0.319008	-4.89116	0.07	0.934852
13.0794	10.3787	5.58758	-5.56799	-0.322256	-5.49694	0.075	0.94017
12.0391	9.69058	5.23266	-5.11195	-0.31655	-4.98559	0.08	0.945256
12.9332	10.4082	5.73626	-5.65171	-0.330874	-5.52363	0.085	0.949339
11.9342	10.1083	5.17614	-5.05629	-0.338783	-5.20661	0.09	0.953018
13.0715	10.2679	5.5084	-5.51771	-0.330445	-5.52929	0.095	0.956687
12.2024	10.2249	5.55704	-5.39628	-0.307908	-5.37498	0.1	0.959906

Table 10: Results of the maximization of P_d using a combination of all criteria; α_i are the weighting factors for the six single criteria $f_i(cut - out)$.

As can be seen from Table 10 the above mentioned deviation from the wanted P_{fa} to the achieved P_{fa}^* occurred for the three smallest P_{fa} values, which actually should have been 0.5%, 1% and 1.5%, but resulted in 1.1945%, 1.1609% and 1.5001%. Nevertheless, due to the corrective "punishment" we implemented we need not fear to have distorted the P_d to the better.

What can also be noticed is that some of the weighting parameters vary strongly. For example, for P_{fa} 7.5% and 8% the difference between the according values for α_1 is 9%.

Having thus taken into consideration and eliminated all problems we encountered, we then start the actual Downhill Simplex Method:
After having set the apt starting simplex the algorithm makes its own way through the n-dimensional landscape on the search for the global minimum.
Firstly, for the starting simplex the method checks, at which point the function is highest, meaning worst. It then projects this point through the opposite face of the simplex (reflection) and checks, if this point is better. If it is, then the highest point undergoes the same treatment and so on, until the smallest value is found.
If it is not, it expands in the same direction (expansion) for a given factor (e.g. 2) and finds another value. If this is smaller than the previous one, the simplex contracts to its original volume and performs the next reflection; if it is not better, the next expansion is performed.
Thus, the algorithm consists mainly of the three steps reflection, expansion and contraction. If the ratio of highest to lowest value for one simplex lies underneath a variable threshold (e.g. 0.001), the global minimum is claimed to be found and its values returned.

The major advantage of this minimization method is that is makes no assumptions about the function under discussion, because we do not even know the function; it does not, as many other methods do, require the derivatives of the function, which we certainly do not know either; it is said to be fairly robust and not to make too many mistakes; the influence one has by setting the best staring values is fairly big and enabled us to eliminate almost all difficulties we came across.
The only disadvantage we can think of, is that it is really slow.

One last problem we encountered is the computational limitation: the two data sets we generate consist of two arrays, each containing - according to the number of criteria - up to 6 floating numbers at a length of 10^6 iterations. The computational devices at hand could not deal with an even larger array of e.g. 10^7 iterations.
Taking into account that we generate two of these huge arrays, then, for 50 different false alarm probabilities, we generate 50000 random numbers at the beginning and check for an unknown number of minimization steps $5 \cdot 10^6$ times the 6 criteria, one can imagine that it takes quite a while ($\approx$ 60 hours) and quite a lot of memory. So unfortunately we could not enlarge the volume of the statistic.

References

[Alt2000] Altmann, J.: 2000, Ausgelagerte Trackinitialisierung, Projektnotiz

[Bar-Shalom] Bar-Shalom, Y., Li, X.-R.: 1983, Estimation and Tracking, Principles, Techniques and Software, ArtechHouse Inc.

[Binias2000] Binias, G.: 2000, Ein Zielspurextraktionsverfahren für OLPI-Antennensystemdaten auf der Grundlage von Hough-Transformationen, FGAN, FHR-Bericht 27

[Black1999] Blackman, S., Popoli, R.: Design and Analysis of Modern Tracking Systems, Artech House

[Bron1977] Bronstein, I.N., Semendjajew, K.A., Musiol, G., Mühlig, H.: Taschenbuch der Mathematik, Verlag Harry Deutsch

[Carrea2002] Carrea, L., Wanielik, G.: 2002, Interpretation of the Polar Decomposition of the Scattering Matrix for Polarimetric Imaging Data, GRS 2002

[Colegrove2002] Colegrove, S. B., Davey, S. J.: 2002, PDAF with Multiple Clutter Regions and Target Models, IEEE July 2002

[Davidson] Davidson, F., Grenier, D., Bosse, E.: 1998, Study of fusion architecture for target tracking with Kalman filtering, Rapport annuel d'activites

[Galati2002] Galatai, G., Leonardi, M.: 2002, Constant-False-Alarm-Rate Techniques for Surface Movement Radar, GRS 2002

[Graf1998] Graf, U., Henning, H.-J., Stange, K., Wilrich, P.-Th.: 1998, Formeln und Tabellen der angewandten mathematischen Statistik, 3. Auflage, Springer Verlag

[Gray2002] Gray, I., Spratt, B.: 2002 Advanced Tracking Data Association Techniques, GRS 2002

[GRS2002] Proceedings zum German Radar Symposium 2002, Deutsche Gesellschaft für Ortung und Navigation

[Harp2002] Harp, G. R.: 2002, 3-Parameter Kalman Filter for Antenna Motion Model

[Hew97,1] Hew, P.C.: 1997, Geometric and Central Moments, Diary, Department of Mathematics, The University of Western Australia, Nedlands

[Hew97,2] Hew, P.C.: 1997, The Spot Algorithm, Diary, Department of Mathematics, The University of Western Australia, Nedlands

[Hofele2002] Hofele, F. X.: 2002, An Innovative Simplification in Processing the CASH-CFAR, GRS 2002

[Kassotakis2001] Kassotakis, I., Uzunoglou, N.: 2002, Sea Clutter Measurements at Low Grazing Angles in Correlation with Prevailing Weather and Atmospheric Condition Using L and S Band Surveillance Radars, GRS 2002

[Kanetos] Kanetos, M., Woodbridge, K., Baker, C.: Tracking optimisation for multi-function radar, University College London department of Electronic and Electrical Engineering

[Kern2000] Kern, A.: 2000, Tracking in der Radaradatenverarbeitung mit Hilfe von bildverarbeitenden Methoden, Diplomarbeit, DaimlerChrysler Aerospace Ulm

[Koch1999] Koch, W.: 1999, Overview of problems and Techniques in Target Tracking, IEE

[Koch2000] Koch, W.: 2000, Fixed-Interval retrodiction Approach to Bayesian IMM-MHT for Maneuvering Multiple Targets, IEEE Vol. 36 No. 1, Jan 2000

[Gessler2002] Gessler, R., Kölle, H.-G., Ludloff, A., Mannes, J.: 2002, Improved CFAR Algorithms using modern hardware, GRS 2002

[vKeuk1996] van Keuk, G.: 1996, Sequential Extraction of Target-Tracks, FGAN, FFM-Bericht 461

[vKeuk1998] van Keuk, G.: 199, Sequential Track-Extraction Related to Sensor Data Fusion, FGAN, FFM-Bericht 483

[Lightart2002] Lightart, L., Tatarinov, V., Tatarinov, S., Pusone, E.G.: 2002, An effective Polarimetric Detection of Small-Scale Man-Made Radar Objects on the Sea Surface, International Conference on Microwaves, Radar and Wireless Communications

[Lim] Lim, G., Alder, M.d., deSilva, C.J.: Syntactic Pattern Classification of Moving Objects in a Domestic Environment, Centre for Intelligent Information Processing Systems, The University of Western Australia, Nedlands

[Loehr] A. Löhr, W. Schrodi: 2002, German Radar Symposium, Proceedings, 'New Approaches to Tracking, Track Initialization and Tracking in Clutter Areas', 77

[Lu] Ludloff, A.: 1993, Praxiswissen Radar und Radarsignalverarbeitung, Vieweg

[McL97] McLaughlin, R.A., Alder, M.D.: 1997, Technical Report, The Hough Transform versus the UpWrite, The University of Western Australia, Nedlands

[McL2000] McLaughlin, R.A.: 2000, Dissertation, University of Western Australia, Nedlands

[McLaughlin1998] McLaughlin, R.A.: 1998, Randomized Hough Transform: Improved Ellipse Detection with Comparison, Technical Report, The University of Western Australia, Nedlands

[Nel69] Nelder, J.A., Mead, R.: 1965, Computer Journal, vol. 7, pp. 308-313

[Pre92] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: 1992, Numerical Recipes in C, Second Edition, Cambridge University Press

[Pru2000] Prunsch, H.: 2000, Grundlagen der Radartechnik, Carl-Crantz-Gesellschaft

[Pusone2002] Pusone, E. G., van Genderen, P., Sinttrujien, J.S.: 2002, Detection of small objects at sea based on frequency analysis of the Stokes vector, GRS 2002

[Pusone2000] Pusone, E. G., van Genderen, P., Sinttrujien, J.S.: A predictive model of sea backscatter Doppler spectrum at X-band frequencies, at low grazing angles, NATO conference on clutter, 2000

[Pusone02] Pusone, E. G., van Genderen, P., Tatarinov, V., Lightart, L. P.: 2002 A computer simulation of a statistical model for detection of radar objects at sea, GRS 2002

[Rohling1983] Rohling, H.: 1983, Radar CFAR Thresholding in Clutter and Multiple Target Environment, IEEE AES-19, p.608-621

[Skolnik1981] Skolnik, M. I.: 1981, Introduction into Radar Systems, 2. ed., McGraw-Hill, Singapore

[Steven1988] Steven, G. F.: 1988, Kalman Filtering - Noise-Corrupted Signal Processing, Electronics & Wireless World, Nov 1988

[Stone1999] Stone, L. D., Barlow, C. A., Corwin, T.L.: Bayesian Multiple Target Tracking, Artech House

[Stu82] Stummel, F., Hainer, K.: 1982, Praktische Mathematik, Teubner

[Winkler2000] Winkler, R. G., van Keuk, G.: 2000, Multi-Hypothesen-Tracking von Formationen: Zuordnungsformulierung und suboptimale Lösung durch Lagrange-Relaxation, FGAN, FKIE-Bericht 24

[Wolf2000] Wolf, G. L.: 2000, Objektextraktion aus Radardaten mittels digitaler Bildverarbeitung, Diplomarbeit DaimlerChrysler Aerospace Ulm

[Wu] Wu, W.-R., Chang, D.-C.: Target Tracking by Using the Median Filter as a Preprocessor, department of Communication Engineering National Chiao Tung University, Taiwan

Acknowledgments

First of all I would like to thank my supervisor Prof. U.-G. Meißner who majorly helped to enable this thesis with his indefatigable support. He is always willing to help and always able to inspire.

I heartily thank Prof. R. G. Winkler who was kindly willing to examine and report on this thesis.

I am also pleased to thank Dr. W. Schrodi and Mr. J. Semmelrodt at EADS for their support as heads of department.

A special thank goes to Christoph who has been a supportive and entertaining room mate and always knew what the ladies wanted.

I also thank my colleagues Ulrich Lode, Jürgen Altmann, Ronald Brauch, Bernd Deutsch, Klaus Gille, Heinz Löw and Jürgen Wittlinger for the friendly and informal atmosphere and especially Dr. W. Wittekind for the informative and helpful discussions.

In addition I would like to thank Stefan Albert and PJ for succour in hard times and also Michaela Kraus, Sascha Florin and Bernd Weferling for helping with 102 things.

Finally, I want to thank Dirk for his friendship. Without him the past three years would have been incredibly less fruitful, less funny and less strenuous.