On Probability of Target Presence Considering Nonlinear Spatial Sampling and Quantization Properties of a Planar Acoustic Sensor Array Used for Target Localization and Tracking Outside the Array Plane

Research Visit: University of California, San Diego (UCSD)
Technical Report (Excerpt)

by
Hannes Pessentheiner

Gernot Kubin, Univ.-Prof. Dipl.-Ing. Dr.techn.
Signal Processing and Speech Communication Laboratory
Graz University of Technology
Inffeldgasse 16c/EG, 8010 Graz, Austria

Bhaskar D. Rao, Prof. Ph.D
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, CA 92093-0407, USA

Contents

List of Symbols

N	Number of dimensions.
$f(\cdot)$	(Non)linear function.
$p(\cdot)$	Probability distribution.
r	Radius/distance between center of coordinate system and target.
$\mathbf{r}$	Vector depending on r.
ϕ	Azimuth angle.
θ	Azimuth angle.
N_ϕ	Number of azimuth angles.
N_θ	Number of elevation angles.
$N_{\phi,\theta}$	Number of of angles after permutation of azimuth and elevation angles.
$\widetilde{\phi}$	Azimuthal angular difference.
$\widetilde{\theta}$	Elevation angular difference.
$\Delta\phi$	Azimuthal step-size.
$\Delta\theta$	Elevation step-size.
$\boldsymbol{\Psi}$	Polar/spherical coordinate vector.
h	Height.
x, y, z	Cartesian coordinates.
x_ϕ	x depending on and represented by ϕ.
$\mathbf{x}$	Coordinate vector.
C	Cell.
C_Σ	Set of cell lengths/areas/volumes/hyper-volumes.
$\mathcal{C}$	Set of cells.
I	Interval.
$\mathcal{I}$	Set of intervals.
ϵ	Mesh-size of a grid.
V_C	Size or (hyper-)volume of a cell.
$\mathbb{R}$	Real numbers.
$\mathbb{Z}$	Whole numbers.
$\mathbb{N}$	Natural numbers.
$\mathcal{B}$	Domain.
$\mathcal{D}$	Region.
$\widetilde{\mathcal{D}}$	Sub-region of $\mathcal{D}$.
$\mathcal{S}$	Target region.
ε	Infinitesimal small value.
m	Particle mass.
k	Discrete time variable.
t	Continuous time variable.
Δs	Line segment of $\mathcal{S}$.
ΔT	Discrete time step.
ϕ_m	Angular interval's lower bound (azimuth).
ϕ_p	Angular interval's upper bound (azimuth).
u_m	Integral's lower bound.

u_p	Integral's upper bound.
ζ_0	Integral's lower bound.
ζ_1	Integral's upper bound.
$\mathbf{J}$	Jacobian matrix.
$\mathbf{U}$	Matrix of eigenvectors.
$\boldsymbol{\Lambda}$	Diagonal matrix of eigenvalues.
$\mathbf{A}$	Part one of factorized covariance matrix.
$\mathbf{B}$	Inverse of $\mathbf{A}$.
η	Noise variable.
β	Damping coefficient/Langevin parameter.
$\hat{v}$	Steady-state root-mean-square velocity.
$\mathcal{N}$	Normal distribution.
Φ	Random variable of azimuth angles.
Θ	Random variable of elevation angles.
X, Y, Z	Random variables of Cartesian coordinates.
w	Kernel weight.
μ	Mean.
σ	Standard deviation.
$\boldsymbol{\Sigma}$	Covariance matrix.
$\mathbf{x}_k$	State vector at time-step k.
$\mathbf{z}_k$	Measurement vector at time-step k.
λ	Parameter set containing mean, variance, and/or weights.

Abstract

In practice, a common way to cope with nonlinear problems is to make them linear, but often without thinking about the adverse effects. These effects are negligible for scenarios that are subject to specific constraints, e.g., limited scopes or unnatural object motions. But for industrial partners it's hard or even impossible to stick to these constraints if they are interested in producing robust implementations and devices, without tight limitations, that are simple to operate and flexible in application. Thus, it is necessary to introduce a more accurate and reasonable solution that considers realistic constraints but requires time-consuming and sophisticated derivations and investigations. Since the beginning of the DIRHA (Distant-speech Interaction for Robust Home Applications) project, a single planar acoustic sensor array mounted on the ceiling has become more and more attractive for distant speech recognition. Ignoring the nonlinear spatial sampling and spatial quantization properties of such a sensor array causes—depending on the scenario—a decrease in (multi-)target localization accuracy or a false prediction of (multi-)target transitions in case of tracking, just to mention a few. This report is about the nonlinear properties of an arbitrary planar acoustic sensor array used for localizing and tracking (multiple) moving targets outside the array plane. It introduces a mathematical framework that describes the nonlinearities and how to calculate the shapes and volumes of the nonlinear quantization cells to avoid linearization and, as a consequence, to increase a distant speech recognition's accuracy. Beside that, it shows how to generate a probability density function based on a uniform distribution or a multi-variate Gaussian mixture model; it is a prerequisite to compute the probability of target presence. Furthermore, the report describes how to assign the probability of target presence to a nonuniform cell depending on its size.

Keywords: spatial sampling, spatial quantization, planar acoustic sensor array, distant speech recognition, (multi-)target localization, (multi-)target tracking.

Chapter 1

Introduction

In practice, a common way to cope with nonlinear problems is to make them linear, but often without thinking about the adverse effects. These effects are negligible for scenarios that are subject to specific constraints, e.g., limited scopes, unnatural object motions, nonrealistic physical conditions, etc. But for industrial partners it's hard or even impossible to stick to these constraints if they are interested in producing robust implementations and devices, without tight limitations, that are simple to operate and flexible in application. Thus, it is necessary to introduce a more accurate, reasonable, and economical solution that considers realistic constraints but requires time-consuming, sophisticated, and challenging derivations and investigations.

Since the beginning of the DIRHA [1] (Distant-speech Interaction for Robust Home Applications) project, a single planar acoustic sensor array mounted on the ceiling has become more and more attractive for distant speech recognition. Ignoring the nonlinear spatial sampling and spatial quantization properties of such a sensor array causes—depending on the scenario—a decrease in (multi-)target localization accuracy or a false prediction of (multi-)target transitions in case of tracking, just to mention a few.

In this report, we focus on the nonlinear properties of an arbitrary planar acoustic sensor array used for localizing and tracking (multiple) moving targets outside the array plane. We introduce a mathematical framework that describes the nonlinearities and how to calculate the shapes and volumes of the nonlinear quantization cells to avoid linearization and, as a consequence, to increase a distant speech recognition's accuracy and to make it more robust. We especially focus on a realistic three-dimensional case where the shape corresponds to an oblique pyramidal frustum. Beside that, we show how to generate a probability density function based on a uniform distribution or a multi-variate Gaussian mixture model; it is a prerequisite to compute the probability of target presence. Furthermore, we describe how to assign the probability of target presence to a nonuniform cell depending on its size.

Chapter 2

Spatial Sampling

Let's assume a spherical coordinate system with a fixed radius r and azimuth and elevation angles $\{\phi \mid 0 \leq \phi < 360,\ \phi \in \mathbb{Z}\}$ and $\{\theta \mid 0 \leq \theta \leq 180, \theta \in \mathbb{Z}\}$, respectively, that are finite in number, e.g., $N_\phi = 360$ and $N_\theta = 181$. Then, the permutation of all angles that span a domain $\mathcal{B}$ yields a cardinality of $|\mathcal{B}| = N_{\phi,\theta} = 65\,160$. In other words, $\mathcal{B}$ spans a grid on a spherical surface consisting of $N_{\phi,\theta} = 65\,160$ grid points. We will later refer to these elements as spatial sampling points. If the space of interest is a grid on a hemispherical surface with angles

$$\{(\phi, \theta) \mid 0 \leq \phi < 360,\ 90 \leq \theta \leq 180,\ \phi, \theta \in \mathbb{Z}\}, \tag{2.1}$$

than the number of elements reduces to $N_{\phi,\theta} = 32\,760$ and, as a consequence, the spatial sampling rate decreases. According to the definitions above, we can represent these elements as unsigned 16-bit integer values that require less memory and processing power than their alternatives with larger word-size. Considering the assumptions mentioned before, the resulting total number of integer angles $N_{\phi,\theta}$ in a spherical coordinate system still covers the spatial resolution requirements for successful distant and close-talking target *direction* detection [2]. A small number of angles or a small word-size becomes even more important in case of resource-intensive target detection and tracking, where the word-size particularly affects the computation time.

Example 1 (Sensor Array Positioning for Target Detection in $\mathbb{R}^3$). *In a real environment, a target detector based on a single acoustic sensor array and the assumption of impinging plane waves cannot determine a target's distance r but its direction (ϕ, θ). In case of an undefined distance r, the detector spatially samples directions (ϕ, θ) defined in (2.1) instead of points on a spherical grid (see Figure 2.1). Let's assume that the center of the planar acoustic sensor array lies in the center of the coordinate system. A sampling element can be represented by an infinitesimal long ray originating from the center of the coordinate system and going through the grid points on a spherical grid with undefined distance r. If we place an acoustic sensor array in the middle of a room, the detector needs to sample $N_{\phi,\theta} = 65\,160$ directions to cover all directions (ϕ, θ) in $\mathbb{R}^3$, whereas a sensor array mounted on a ceiling needs to sample $N_{\phi,\theta} = 32\,760$ directions only. This requires less computational resources. Moreover, this avoids spatial aliasing caused by the up- down–ambiguity of the planar sensor array. And besides that, we can increase the spatial resolution on the lower hemisphere without exceeding $N_{\phi,\theta} = 65\,160$.* ■

There is a growing attention in decreasing an algorithm's computation time and increasing a system's performance by upgrading algorithms and implementations without improving its hardware. Still, the system must be accurate, powerful, energy-efficient, and reasonable in price—prerequisites for, e.g., (ambient) assisted living environments [1], where old or disabled people cannot raise the financial means for expensive hardware. Ways to fulfill these prerequisites are to use inexpensive hardware, to improve the algorithms' performances, or to find inconsistencies in algorithms and implementations. One common inconsistency in three-dimensional target *direction* detection and tracking is to assume a constant target distance for all directions, which causes wrong transition model steady-state–velocities for moving targets. The steady-state–velocities play an important role in target prediction, especially by applying transition models (e.g., the Langevin model) when using equally spaced angles and planar sensor arrays. This assumption is correct if targets are moving around the planar acoustic sensor array, within the sensor array plane, and without changing the distance [2] [3] [4]. However, in real scenarios targets are moving horizontally through space, not on a (hemi)spherical surface.

Now, let's focus on the problematic of spatial sampling in target detection and tracking with a single planar sensor array. Let's assume that a spatial filter samples a space $\mathbb{R}^N$ in equidistant N-dimensional sampling intervals, which results in an N-dimensional uniform grid with point lattices

$$\mathcal{B} = (\epsilon_1\mathbb{Z}) \times \cdots \times (\epsilon_n\mathbb{Z}) \times \cdots \times (\epsilon_N\mathbb{Z}) \subset \mathbb{R}^N,$$

where ϵ_n is the mesh size of a grid $\mathcal{B}$ in the n-th direction [5] and constant for all $\mathbb{Z}$. An N-dimensional sampling interval is defined as the cartesian product of N one-dimensional intervals

$$I = I_1 \times \cdots \times I_n \times \cdots \times I_N \subset \mathbb{R}^N,$$

where $a_n \leq I_n \leq a_n + \epsilon_n$ with $a_n \in (\epsilon_n\mathbb{Z})$. Similar to the interval I, an N-dimensional quantization cell is defined as the cartesian product of N one-dimensional cells

$$C = C_1 \times \cdots \times C_n \times \cdots \times C_N \subset \mathbb{R}^N, \tag{2.2}$$

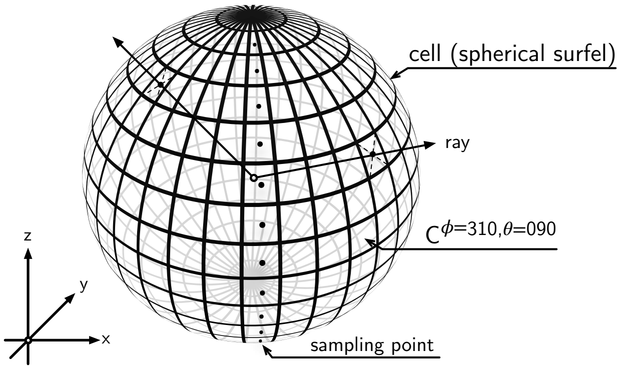

Figure 2.1: Sphere with equidistant sampling points (black dots) spanning a grid (here: vertical grid segment) and surface elements representing cells $C^{(i)}$. A ray going through a surface element represents possible sampling points for a fixed set of angles (ϕ, θ) and varying radius r, i.e., it illustrates a direction.

where $a_n - \epsilon_n/2 \leq C_n \leq a_n + \epsilon_n/2$ with $a_n \in (\epsilon_n\mathbb{Z})$. Thus, the N-dimensional intervals $\forall i : I^{(i)} \in \mathcal{I}$ are equidistant, and the corresponding quantization cells $C^{(i)} \in \mathcal{C}$ are equal in size. Both $\mathcal{I}$ and $\mathcal{C}$ are sets. A direction with $\epsilon_n = 0$ is defined as a continuous one: $(\epsilon_n\mathbb{Z}) = \mathbb{R}$ and $\lim\limits_{\epsilon_n \to 0} C_n = 0$.

Now, let's assume a spatial filter that samples a region $\mathcal{D}_1 \in \mathbb{R}^N$ in a way that its subregions introduced by sampling, i.e., N-dimensional cells of $\mathcal{D}_1$, are equal in size. If $\mathcal{D}_2 \subset \mathbb{R}^N$ is a nonlinearly distorted or transformed version of $\mathcal{D}_1$, i.e., $f : D_1 \subset \mathbb{R}^N \to D_2 \subset \mathbb{R}^N$, where $f(\cdot)$ is a nonlinear function, then the same spatial filter samples $\mathcal{D}_2$ in a nonlinear way, so that the N-dimensional cells are unequal in size.

Example 2 (Linear and Nonlinear Spatial Sampling of a Line Embedded in $\mathbb{R}^2$). *Let's assume a spatial filter that samples an arc linearly with elevation angles $120 \leq \theta \leq 180$ and step-size $\Delta\theta = 10$, a fixed azimuth angle ϕ, and a radius $r = h$ (see Figure 2.2). The black dots on the arc, it is labeled as region $\mathcal{D}_1 \in \mathbb{R}$ embedded in $\mathbb{R}^2$, represent the sampling points after sampling region $\mathcal{D}_1$. Each arc-segment between two sampling points has the same size, i.e., each interval and cell (arc not labeled in figure) is equal in size. This is due to the spatial sampling of the arc with radius $r = h$ based on a discrete set of equally-spaced elevation angles. Now, let's transform $\mathcal{D}_1$ in a way that we obtain the tangent $\mathcal{D}_2 \in \mathbb{R}$ of the arc at point $(r = h, \theta = 180)$. We call this tangent the target axis. The same spatial filter samples $\mathcal{D}_2$ in a nonlinear way, so that we obtain unequal intervals $I^{(i)}$ and cells $C^{(i)}$. The smaller the angular difference $\widetilde{\theta}_\theta$, the larger its corresponding interval $I^{(i)}$ and cell $C^{(i)}$.* ■

Example 3 (Linear and Nonlinear Spatial Sampling of a Disc Embedded in $\mathbb{R}^3$). *Let's assume a spatial filter that samples a circle linearly with azimuth angles $0 \leq \phi < 360$ and step-size $\Delta\theta = 10$, and a radius $r \in \{r_1, r_2\}$. The radii $\{r_i \mid r_i = (-h)\cos(\theta)^{-1},\ \theta_i = \{140, 130\},\ i = \{1, 2\}\}$ point on a two-dimensional region $\mathcal{D}_2$, the target plane (see Figure 2.3). The cells aligned on a circle for a fixed θ and varying ϕ are equal in size. This*

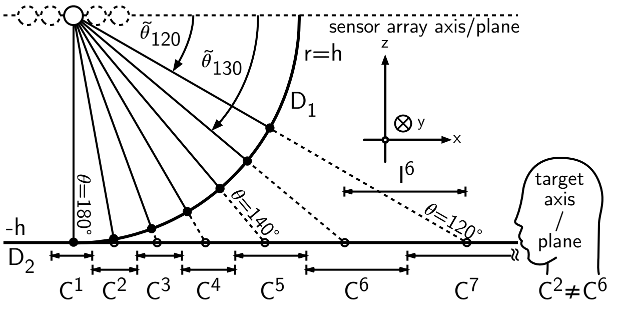

Figure 2.2: Region $\mathcal{D}_1$ represents the arc, whereas $\mathcal{D}_2$ represents the target axis. The black dots and circles on the arc and target axis, respectively, illustrate sampling points that span a grid on both regions. Variable $\widetilde{\theta}_\theta$ represents the angular difference between the sensor array axis and an arbitrary elevation angle θ. Variables $C^{(i)}$ and $I^{(i)}$ represent a cell and an interval related to a certain θ.

is due to the linear sampling of a circle with linearly spaced azimuth angles ϕ. However, if we consider both radii in $\mathcal{D}_2$, the cells' size and interval on a circle are constant but grow for increasing radii. This means that the spatial filter samples a circle linearly, but a disc nonlinearly. ■

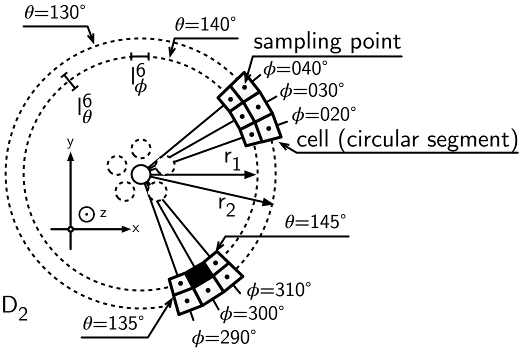

Figure 2.3: The black dot in the centroid of each circular segment illustrates a sampling point that span a grid on region $\mathcal{D}_2$, which is the target plane. The circular segments grow in size for an increasing radius.

Chapter 3

Spatial Resolution

The resolution of a spatial filter depends the spatial sampling interval or cell size: the smaller it is, the higher the spatial resolution. In practice, finite computational resources limit the spatial resolution to an upper bound. However, the more computational resources are available, the smaller we can set the sampling intervals to obtain a better spatial resolution—without loosing realtime performance. This enables a more accurate position assignment of a detected target in space. The following example illustrates the effect of a limited resolution in case of target detection.

> **Example 4 (Limited Spatial Sampling Resolution).** *A target detector—a spatial filter—linearly samples a region $D \subset \mathbb{R}^N$, and splits it into a set $\mathcal{C}$ of equally sized cells C. The centroid of each cell represents a sampling point in $\mathcal{D}$. In case of an infinite high resolution, the target detector would be able to sample each point in $\mathcal{D}$; thus, C would be infinitesimal small. In case of a limited resolution, the target detector samples a finite number of points in $\mathcal{D}$, and $V_C > \varepsilon$, where V_C is the size or hyper-volume of C, and ε is an infinitesimal small number. After sampling $\mathcal{D}$ with a limited resolution, the target detector determines the activity but not the number of targets in each cell. If there are two targets at different positions within C, the filter would only detect a single target activity. To detect both targets, the sampling resolution has to be increased until both targets are located in different cells.* ■

In the remaining report, we will use a limited angular resolution as defined in (2.1).

Chapter 4

Nonlinear Spatial Sampling of a Line Embedded in $\mathbb{R}^2$

Let's start with a theoretical and simple scenario about sampling a one-dimensional region, more specifically, a straight line embedded in $\mathbb{R}^2$. Our goal is to derive a mathematical expression for the radius that points on cells that split the straight line into segments. We will need these segments later to set up a mask that provides information about the probability of target presence.

For now, let's assume a moving target on a straight line parallel to the sensor array axis and call this line the target axis (see Figure 2.2). A sampling point is the intersection point of the target axis and a straight line originating from the center of the coordinate system, which is the center of the sensor array, and running in a direction ϕ. Thus, we need to know the target's height, which is given in case of, e.g., a wheelchair user.

Both, the sensor array and target axis, lie within a two-dimensional region defined as $\mathcal{D} = \{(x, y) \mid x \in \mathbb{R},\ y \in \mathbb{R}_{<0}\}$, where (x, y) are Cartesian coordinates. We convert the Cartesian to polar coordinates (r, ϕ) according to

$$x = r\cos(\phi) \tag{4.1}$$

$$y = r\sin(\phi) \tag{4.2}$$

with $\{(r, \phi) \mid r \in \mathbb{R}_{\geq 0},\ 180 < \phi < 360\}$. The sum of the squared cartesian coordinates is

$$x^2 + y^2 \overset{y=(-h)}{=} x^2 + (-h)^2 = r^2(\cos^2(\phi) + \sin^2(\phi)) = r^2,$$

where

$$y = -h,\ h \geq 0. \tag{4.3}$$

Its square root yields the radius

$$r = \sqrt{x^2 + (-h)^2} \tag{4.4}$$

pointing to $\mathbf{x} = (x, -h)^T$. Variable h represents the Euclidean norm of vector $\mathbf{h}$, which is a vector perpendicular to the region $\widetilde{\mathcal{D}} = \{(x, y) \mid x \in \mathbb{R},\ y = 0,\ \widetilde{\mathcal{D}} \subset \mathcal{D}\}$—the sensor array axis—pointing to $\mathbf{x} = (x, -h)^T$. We combine (4.1) and (4.3) to form a vector

$$\mathbf{r} = (r\cos(\phi), -h)^T = (\sqrt{x^2 + (-h)^2}\cos(\phi), -h)^T.^1 \tag{4.5}$$

Now, we define a parameter set $\mathcal{P} = \{h, \phi, \Delta\phi\}$ that comprises the azimuthal step size $\Delta\phi$. Additionally, we define the azimuthal cell boundaries $\phi_p = \phi + \Delta\phi/2$ and $\phi_m = \phi - \Delta\phi/2$ as well as cell C—it is an Euclidean distance measure at the same time—

$$C(\mathcal{P}) = \|\mathbf{r}(\phi_m) - \mathbf{r}(\phi_p)\|, \tag{4.6}$$

where $\mathbf{r}(\phi_p)$ and $\mathbf{r}(\phi_m)$ are vectors pointing in different directions ϕ_p and ϕ_m, respectively. Inserting (4.5) into (4.6) yields

$$C(\mathcal{P}) = \left\| \begin{pmatrix} \sqrt{x_{\phi_m}^2 + (-h)^2}\cos(\phi_m) \\ -h \end{pmatrix} - \begin{pmatrix} \sqrt{x_{\phi_p}^2 + (-h)^2}\cos(\phi_p) \\ -h \end{pmatrix} \right\| \tag{4.7}$$

with $x_{\phi_p} = r\cos(\phi_p)$ and $x_{\phi_m} = r\cos(\phi_m)$. We omit the y-components in (4.7)—their difference is always zero—due to simplicity. Therefore, we replace the Euclidean distance $\|\cdot\|$ by the modulus $|\cdot|$. Now, we substitute x in a way that C becomes independent of variable x. Therefore, we assume $y = (-h) = r_\phi \sin(\phi)$ and consider (4.4) for a general θ to obtain

$$(-h) = \sqrt{x_\phi^2 + (-h)^2}\sin(\phi), \tag{4.8}$$

and solve (4.8) for x_ϕ so that

$$x_\phi = (-h)\sqrt{\sin(\phi)^{-2} - 1} = (-h)\frac{\sqrt{1 - \sin(\phi)^2}}{\sin(\phi)}$$

which yields

$$x_\phi = (-h)\tan(\phi)^{-1}. \tag{4.9}$$

The index in x_ϕ refers to the dependency on ϕ. Now, we can rewrite $\mathbf{r}$ according to

$$\boxed{\mathbf{r}(h, \phi) = (-h)\left(\tan(\phi)^{-1}, 1\right)^T.} \tag{4.10}$$

Inserting (4.9) into (4.7) and considering the modulus yields

$$C(\mathcal{P}) = h\left(\sqrt{\sin^{-2}(\phi_m)}\cos(\phi_m) - \sqrt{\sin^{-2}(\phi_p)}\cos(\phi_p)\right). \tag{4.11}$$

After applying the trigonometric identities we get

$$\boxed{C(\mathcal{P}) = h\left(\tan(\phi_m)^{-1} - \tan(\phi_p)^{-1}\right)}, \tag{4.12}$$

which is a mathematical expression to calculate the cells on a straight line.

Another way to derive C is to solve a line integral over a uniform scalar field $f(\mathbf{r}(h, \phi)) = 1$, which will be necessary in the next section. Assume that the straight line S can be partitioned into sub-lines Δs_i. We can compute S by summing over all infinite small sub-lines according to

$$S = \lim_{\Delta s_i \to 0} \sum_{i=1}^{N} f(\mathbf{r}(h, \phi)) \cdot \Delta s_i = \lim_{\Delta s_i \to 0} \sum_{i=1}^{N} \Delta s_i \tag{4.13}$$

[1] We use $\mathbf{r}$ instead of $\mathbf{x}$ to represent a point in $\mathbb{R}^2$ in polar coordinates.

where N is the total number of line segments. The distance between subsequent points on the straight line is

$$\Delta s_i = |\mathbf{r}(h, \phi_i + \Delta\phi) - \mathbf{r}(h, \phi_i)| \approx \left| \frac{\mathrm{d}\mathbf{r}(h, \phi_i)}{\mathrm{d}\phi_i} \right| \Delta\phi. \tag{4.14}$$

Substituting (4.14) into (4.13) yields

$$S = \lim_{\Delta s_i \to 0} \sum_{i=1}^{N} \left| \frac{\mathrm{d}\mathbf{r}(h, \phi_i)}{\mathrm{d}\phi_i} \right| \Delta\phi,$$

the Riemann sum for

$$S = \int \left| \frac{\mathrm{d}\mathbf{r}(h, \phi)}{\mathrm{d}\phi} \right| \mathrm{d}\phi.$$

To calculate the length of a cell C, we need to set the limits of the integral according to

$$C = \int\limits_{\phi_m}^{\phi_p} \left| \frac{\mathrm{d}\mathbf{r}(h, \phi)}{\mathrm{d}\phi} \right| \mathrm{d}\phi. \tag{4.15}$$

Inserting

$$\begin{aligned}
\left| \frac{\mathrm{d}\mathbf{r}(h, \phi)}{\mathrm{d}\phi} \right| &= \left| (-h) \frac{\mathrm{d}}{\mathrm{d}\phi} \begin{pmatrix} \cos(\phi)\sin(\phi)^{-1} \\ 1 \end{pmatrix} \right| \\
&= \left| (-h) \begin{pmatrix} \dfrac{-\sin(\phi)\sin(\phi) - \cos(\phi)\cos(\phi)}{\sin(\phi)^2} \\ 0 \end{pmatrix} \right| \\
&= \frac{h}{\sin(\phi)^2}
\end{aligned}$$

into (4.15) yields

$$C = \int\limits_{\phi_m}^{\phi_p} \frac{h}{\sin(\phi)^2} \mathrm{d}\phi = (-h) \left[\cot(\phi_p) - \cot(\phi_m) \right],$$

which results in

$$\boxed{C = h \left(\tan(\phi_m)^{-1} - \tan(\phi_p)^{-1} \right),} \tag{4.16}$$

and which is identical to (4.12).

Now, we can split the target axis into segments by applying (4.16), and with (4.10) we are able to calculate the distance between the origin of the coordinate system and each point on the target axis.

Chapter 5

Nonlinear Spatial Sampling of a Plane Embedded in $\mathbb{R}^3$

Now, let's continue with a more realistic scenario. Our goal is to derive a mathematical expression for the radius that points on cells that split the plane embedded in $\mathbb{R}^3$ into segments.

Let's assume a moving target on a plane—the target plane—that is parallel to the sensor array plane. A sampling point is the intersection point of the target plane and a straight line originating from the center of the coordinate system, which is the center of the sensor array, and running in a direction (ϕ, θ). Once again, we need to know the target's height.

Let's consider the centroid of a sensor array as the origin of the coordinate system. We define a region $\mathcal{D} = \{(x, y, z) \mid x \in \mathbb{R},\ y \in \mathbb{R},\ z \in \mathbb{R}_{<0}\}$ and convert the cartesian to spherical coordinates (r, ϕ, θ) according to

$$x = r\sin(\theta)\cos(\phi) \tag{5.1}$$

$$y = r\sin(\theta)\sin(\phi) \tag{5.2}$$

$$z = r\cos(\theta) \tag{5.3}$$

with $\{(r, \phi, \theta) \mid r \in \mathbb{R}_{\geq 0},\ 0 \leq \phi < 360,\ 90 < \theta \leq 180\}$. Taking into account that

$$z = -h,\ h \geq 0 \tag{5.4}$$

yields a radius according to

$$r = \sqrt{x^2 + y^2 + (-h)^2} \tag{5.5}$$

pointing to $\mathbf{x} = (x, y, -h)^T$. Variable h represents the Euclidean norm of vector $\mathbf{h}$, which is a vector perpendicular to the region $\widetilde{\mathcal{D}} = \{(x, y, z) \mid x \in \mathbb{R},\ y \in \mathbb{R},\ z = 0,\ \widetilde{\mathcal{D}} \subset \mathcal{D}\}$—the sensor array plane—pointing to $\mathbf{x} = (x, y, z)^T$. We combine (5.1), (5.2), and (5.4) to form a vector:

$$\mathbf{r} = \begin{pmatrix} r\sin(\theta)\cos(\phi) \\ r\sin(\theta)\sin(\phi) \\ -h \end{pmatrix} = \begin{pmatrix} \sqrt{x^2 + y^2 + (-h)^2}\sin(\theta)\cos(\phi) \\ \sqrt{x^2 + y^2 + (-h)^2}\sin(\theta)\sin(\phi) \\ -h \end{pmatrix}. \tag{5.6}$$

Now, we define a parameter set $\mathcal{P} = \{h, \phi, \theta, \Delta\phi, \Delta\theta\}$ that comprises the azimuthal and elevation step sizes $\Delta\phi$ and $\Delta\theta$. Additionally, we define the angular cell boundaries

$\theta_p = \theta + \Delta\theta/2$, $\theta_m = \theta - \Delta\theta/2$, $\phi_p = \phi + \Delta\phi/2$, and $\phi_m = \phi - \Delta\phi/2$. Instead of defining an Euclidean distance measure, we have to use integrals to determine the size of a two-dimensional cell C. First, we need calculate the norm of (5.6):

$$r(h,\phi,\theta) = \|\mathbf{r}(h,\phi,\theta)\| = \left\| \begin{pmatrix} \sqrt{x^2+y^2+(-h)^2}\sin(\theta)\cos(\phi) \\ \sqrt{x^2+y^2+(-h)^2}\sin(\theta)\sin(\phi) \\ -h \end{pmatrix} \right\|. \tag{5.7}$$

Let's assume that $z = (-h) = r\cos(\theta)$ and consider this in (5.5) to obtain

$$(-h) = \sqrt{x^2+y^2+(-h)^2}\cos(\theta)$$

or

$$(-h)\cos(\theta)^{-1} = \sqrt{x^2+y^2+(-h)^2} \tag{5.8}$$

after multiplying $\cos(\theta)^{-1}$ on both sides. Inserting (5.8) into (5.7) results in

$$\begin{aligned} r(h,\phi,\theta) &= \left\| \begin{pmatrix} (-h)\cos(\phi)\cos^{-1}(\theta)\sin(\theta) \\ (-h)\sin(\phi)\cos^{-1}(\theta)\sin(\theta) \\ -h \end{pmatrix} \right\| = \left\| \begin{pmatrix} (-h)\cos(\phi)\tan(\theta) \\ (-h)\sin(\phi)\tan(\theta) \\ -h \end{pmatrix} \right\| \\ &= \sqrt{(-h)^2\cos(\phi)^2\tan(\theta)^2 + (-h)^2\sin(\phi)^2\tan(\theta)^2 + (-h)^2} \\ &= \sqrt{(-h)^2\left(\cos(\phi)^2\tan(\theta)^2 + \sin(\phi)^2\tan(\theta)^2 + 1\right)} \\ &= \sqrt{(-h)^2\left(\tan(\theta)^2 + 1\right)} \\ &= (-h)\sqrt{\tan(\theta)^2 + 1} \end{aligned}$$

which yields

$$\boxed{r(h,\theta) = (-h)\cos(\theta)^{-1}} \tag{5.9}$$

that does not depend on ϕ anymore. Now, we have derived the distance between the origin of the coordinate system and each point on the target plane. To obtain an expression for the corresponding cells, we have to calculate the area

$$\begin{aligned} C(\mathcal{P}) &= \int\limits_{\phi_m}^{\phi_p} \int\limits_{r(h,\theta_p)}^{r(h,\theta_m)} r(h,\theta)\mathrm{d}r\mathrm{d}\phi \\ &= \int\limits_{\phi_m}^{\phi_p} \frac{r(h,\theta_m)^2 - r(h,\theta_p)^2}{2}\mathrm{d}\phi \\ &= (\phi_p - \phi_m)\,\frac{r(h,\theta_m)^2 - r(h,\theta_p)^2}{2} \\ &= \Delta\phi\frac{r(h,\theta_m)^2 - r(h,\theta_p)^2}{2} \\ &= \frac{\Delta\phi}{2}\left((-h)^2\cos(\phi_m)^{-2} - (-h)^2\cos(\phi_p)^{-2}\right) \\ &= \frac{\Delta\phi h^2}{2}\left(\cos(\phi_m)^{-2} - \cos(\phi_p)^{-2}\right) \\ &= \frac{\Delta\phi h^2}{2}\left(1+\tan(\phi_m)^2 - \left(1+\tan(\phi_p)^2\right)\right), \end{aligned} \tag{5.10}$$

which results in the area of a cell

$$\boxed{C(\mathcal{P}) = \frac{\Delta\phi h^2}{2}\left(\tan^2(\theta_m) - \tan^2(\theta_p)\right).}$$

Now, we can split the target plane into segments by applying (5.10), and with (5.9) we are able to calculate the distance between the origin of the coordinate system and each point on the target plane.

Chapter 6

Probability of Target Presence

Many target trackers are based on a (recursive) Bayesian filter [6]. Their performance depends on transition models and likelihood functions, among others. In the previous chapters, we derived expressions that enable us to improve the transition model's prediction step by properly assigning a distance between the sensor array and a point on a target plane.

To improve the quality of the likelihood function, we introduce a probability of target presence, which can be used to reduce clutter or any other inferences occurring at places with a low probability density of target presence, e.g., places with non-removable furniture or parts of a room that are not or hardly accessible; no one can go through a pillar or wall, or walk on a conference room table or a bookshelf, just to name but a few. Moreover, a target does not move (close) along the wall or tend to stay at corners. Thus, the probability density of target presence at such places is low. It is higher at frequently visited places, e.g., a chair next to an office desk, or a toilette, a window, etc. We can model the probability density by a Gaussian mixture model, which simplifies calculations and yields closed-form solutions when we split the region of interest into cells and assign a probability[1]to each pair of angles (ϕ, θ).

Figure 6.1 depicts the probability of target presence modeled by weighted Gaussian kernels or a uniform distribution. The uniform distribution enables us to precisely set the boundaries of the region of interest—the probability density becomes zero at specified regions, e.g., walls and obstacles—, but we cannot define different probability densities within this region. By contrast, the Gaussian mixture distribution enables us to define non-uniform probability densities within this region, but, due to the definition of the Gaussian function, we are not able to set precise boundaries where the probability becomes zero if they are exceeded. However, we can model an abrupt decrease of probability density, but then we need to use a high number of Gaussian kernels.

In the upcoming chapters, we will focus on the Gaussian mixture distribution. Still, for the sake of completeness, we derive the closed-form solution for a uniform distribution.

[1] We assign a probability to each cell, but we compile a probability density describing the probability of target presence in a room.

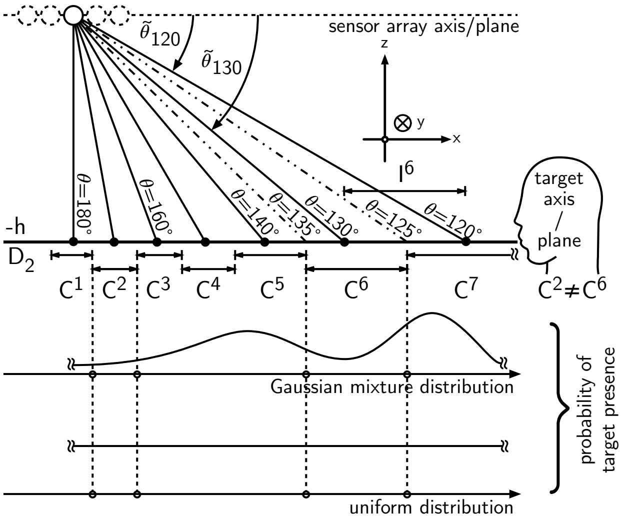

Figure 6.1: A Gaussian mixture distribution and a uniform distribution model the probability of target presence.

The probability distribution of a recursive Bayesian filtering update [6] is

$$p(\mathbf{x}_k|\mathbf{z}_k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{k-1})}{p(\mathbf{z}_k|\mathbf{z}_{k-1})},$$

where $\mathbf{x}$ and $\mathbf{z}$ represents the state and the measurement vector, respectively, and $p(\cdot)$ is a probability distribution. Without going into detail, let's focus on the likelihood function $p(\mathbf{z}_k|\mathbf{x}_k)$ only. A sensor array measures the target position or direction and provides this information as a measurement distorted by sensor noise and other interferences. Both, the sensor noise and the interferences, can be modeled by likelihood functions that faithfully reflect the measurement and transition models in a probabilistic way. One way to fuse the likelihood function and the probability of target presence is to multiply both functions together and consider the result in the Bayesian filtering update [6]

$$p(\mathbf{x}_k|\mathbf{z}_k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)\widetilde{p}(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{k-1})}{p(\mathbf{z}_k|\mathbf{z}_{k-1})},$$

where $\widetilde{p}(\mathbf{z}_k|\mathbf{x}_k)$ represents the probability of target presence.

Our main goal is to compute the probability of target presence for each cell and, thus, for each set of angles (ϕ, θ). To do so, we have to split the region of interest into cells, and we need to integrate the probability density over each cell to obtain its probability.

Chapter 7

Probability of Target Presence of a Linear Cell Embedded in $\mathbb{R}^2$

Let's start with a simple scenario: a target is moving on a straight line, which is split into several cells. Eq. (4.12) represents a one-dimensional cell C for a set of parameters $\mathcal{P}$ without including the probability of target presence. Thus, we have to rewrite (4.15) as follows:

$$p(\mathcal{P}) = \int\limits_{S \subset \mathbb{R}^1} f_X(x;\lambda)\mathrm{d}x = \int\limits_{\phi_m}^{\phi_p} f_\Phi(\mathbf{\Psi};\lambda) \left| \frac{\mathrm{d}r(h,\phi)}{\mathrm{d}\phi} \right| \mathrm{d}\phi \tag{7.1}$$

where $f_X(x;\lambda)$ is a probability density function in Cartesian coordinates and $f_\phi(\mathbf{\Psi};\lambda)$ in polar coordinates depending on $\mathbf{\Psi} = (r(h,\phi)\cos(\phi), r(h,\phi)\sin(\phi))^T$. Set λ contains the mean μ and variance σ.

7.1 Uniform Distribution

In case of a uniform probability of target presence, we set $\lambda = \{C_\Sigma\}$ and $f(\mathbf{\Psi};\lambda) = 1/C_\Sigma$, where C_Σ is the sum of all cell lengths $\{C^{(i)}\}_{i=1}^{N_{\phi,\theta}}$, i.e., the length of the target axis. Inserting the uniform density function in (7.1) and solving

$$p(\mathcal{P}) = \int\limits_{\phi_m}^{\phi_p} f_\Phi(\mathbf{\Psi};\lambda) \left| \frac{\mathrm{d}r(h,\phi)}{\mathrm{d}\phi} \right| \mathrm{d}\phi \quad = \int\limits_{\phi_m}^{\phi_p} \frac{1}{C_\Sigma} \left| \frac{\mathrm{d}r(h,\phi)}{\mathrm{d}\phi} \right| \mathrm{d}\phi = \frac{1}{C_\Sigma} \int\limits_{\phi_m}^{\phi_p} \frac{h}{\sin(\phi)^2}\mathrm{d}\phi$$

yields

$$\boxed{p(\mathcal{P}) = \frac{h}{C_\Sigma}\left(\tan(\phi_m)^{-1} - \tan(\phi_p)^{-1}\right),}$$

the probability of target presence of a cell, i.e., a line segment with parameters $\mathcal{P}$.

7.2 Gaussian Mixture Distribution

In case of a Gaussian mixture distribution, we set $\lambda = \{\mu_i, \sigma_i, w_i\}_{i=1}^{M}$, where M is the number of Gaussian kernels, μ_i and σ_i represents the mean and the variance of the Gaussian kernel with index i, respectively. The parameter w_i is its corresponding weight. Let

$$f_X(x;\lambda) = \sum_{i=1}^{M} w_i \cdot \mathcal{N}(x;\mu_i,\sigma_i), \tag{7.2}$$

be a one-dimensional Gaussian mixture probability density function. The Gaussian distribution in Cartesian coordinates is defined as

$$\mathcal{N}(x;\mu_i,\sigma_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \mathrm{e}^{-\dfrac{(x-\mu_i)^2}{2\sigma_i^2}}. \tag{7.3}$$

Replacing x with x_ϕ defined in (4.9) we get

$$\begin{aligned}
p(\mathcal{P}) &= \int\limits_{\phi_m}^{\phi_p} f_\Phi(\mathbf{\Psi};\lambda) \left|\frac{\mathrm{d}r(h,\phi)}{\mathrm{d}\phi}\right| \mathrm{d}\phi \\
&= \int\limits_{\phi_m}^{\phi_p} \left(\sum_{i=1}^{M} w_i \cdot \mathcal{N}(x_\phi;\mu_i,\sigma_i)\right) \left|\frac{\mathrm{d}r(h,\phi)}{\mathrm{d}\phi}\right| \mathrm{d}\phi \\
&= \sum_{i=1}^{M} w_i \int\limits_{\phi_m}^{\phi_p} \left(\mathcal{N}(x_\phi;\mu_i,\sigma_i)\right) \left|\frac{\mathrm{d}r(h,\phi)}{\mathrm{d}\phi}\right| \mathrm{d}\phi \\
&= \sum_{i=1}^{M} \frac{1}{\sqrt{2\pi\sigma_i^2}} \int\limits_{\phi_m}^{\phi_p} \mathrm{e}^{-\dfrac{(x_\phi-\mu_i)^2}{2\sigma_i^2}} \left|\frac{\mathrm{d}r(h,\phi)}{\mathrm{d}\phi}\right| \mathrm{d}\phi \\
&= \sum_{i=1}^{M} \frac{1}{\sqrt{2\pi\sigma_i^2}} \int\limits_{\phi_m}^{\phi_p} \mathrm{e}^{-\dfrac{\left((-h)\tan(\phi)^{-1}-\mu_i\right)^2}{2\sigma_i^2}} \frac{h}{\sin(\phi)^2} \mathrm{d}\phi.
\end{aligned}$$

Substituting $(-h)\cot(\phi) - \mu_i = u_i$ where

$$\begin{aligned}
\frac{\mathrm{d}u_i}{\mathrm{d}\phi} &= (-h)\frac{\mathrm{d}\cot(\phi)}{\mathrm{d}\phi} \\
&= (-h)\frac{(-\sin(\phi)\sin(\phi) - \cos(\phi)\cos(\phi))}{\sin(\phi)^2} \\
&= h\frac{(\sin(\phi)^2 + \cos(\phi)^2)}{\sin(\phi)^2} \\
&= \frac{h}{\sin(\phi)^2}
\end{aligned}$$

yields the normalized Gaussian mixture functions

$$\boxed{p(\mathcal{P}) = \sum_{i=1}^{M} \frac{1}{\sqrt{2\pi\sigma_i^2}} \int\limits_{u_m^{(i)}}^{u_p^{(i)}} \mathrm{e}^{-\dfrac{u_i^2}{2\sigma_i^2}} \mathrm{d}u_i}$$

with integral boundaries

$$\begin{aligned} u_p^{(i)} &= (-h)\cot(\phi_p) - \mu_i \\ u_m^{(i)} &= (-h)\cot(\phi_m) - \mu_i \end{aligned}$$

Now, we can compute each cell's probability and use it as a probability of target presence to improve Bayesian filtering.

Chapter 8

Probability of Target Presence of a Planar Cell Embedded in $\mathbb{R}^3$

Now, let's assume a moving target on a plane embedded in $\mathbb{R}^3$. The plane is split into two-dimensional cells C. To compute a cell's probability, we need to calculate

$$p(\mathcal{P}) = \iint\limits_{S \subset \mathbb{R}^2} f_{X,Y}(\mathbf{x}; \lambda) \mathrm{d}\mathbf{x} = \int\limits_{\phi_m}^{\phi_p} \int\limits_{r(h,\theta_p)}^{r(h,\theta_m)} f_{\Phi,\Theta}(\mathbf{\Psi}; \lambda) r(h, \theta) \mathrm{d}r \mathrm{d}\phi \tag{8.1}$$

where $f_{\Phi,\Theta}(\mathbf{\Psi}; \lambda)$ is a probability density function in Cartesian coordinates depending on $\mathbf{\Psi} = (x = r\cos(\phi), y = r\sin(\phi))^T$. Set λ contains the mean vector μ and covariance matrix $\mathbf{\Sigma}$.

8.1 Uniform Distribution

In case of a uniform probability of target presence, we set $\lambda = \{C_\Sigma\}$ and $f(\mathbf{\Psi}; \lambda) = 1/C_\Sigma$, where C_Σ is the sum of all cell areas $\{C^{(i)}\}_{i=1}^{N_{\phi,\theta}}$, i.e., the area of the target plane. Inserting the uniform density function in (8.1) results in

$$\boxed{p(\mathcal{P}) = \int\limits_{\phi_m}^{\phi_p} \int\limits_{r(h,\theta_p)}^{r(h,\theta_m)} \frac{1}{C_\Sigma} r(h, \theta) \mathrm{d}r \mathrm{d}\Phi = \frac{\Delta\phi h^2}{2C_\Sigma} \left(\tan(\theta_m)^2 - \tan(\theta_p)^2\right),}$$

the probability of target presence of a cell, i.e., a plane segment with parameters $\mathcal{P}$.

8.2 Gaussian Mixture Distribution

In case of a Gaussian mixture distribution, we set $\lambda = \{\mu_i, \mathbf{\Sigma}_i, w_i\}_{i=1}^{M}$, where M is the number of Gaussian kernels, μ_i and $\mathbf{\Sigma}_i$ represents the mean and the variance of the Gaussian kernel with index i, respectively. The parameter w_i is its corresponding weight. Let

$$f_{X,Y}(\mathbf{x}; \lambda) = \sum_{i=1}^{M} w_i \cdot \mathcal{N}(\mathbf{x}; \mu_i, \mathbf{\Sigma}_i), \tag{8.2}$$

be a two-dimensional Gaussian mixture probability density function. The multivariate (here: bivariate) Gaussian distribution in Cartesian coordinates is defined as

$$\mathcal{N}(\mathbf{x};\mu_i,\mathbf{\Sigma}_i)=\frac{1}{2\pi|\mathbf{\Sigma}_i|^{1/2}}\mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\mu_i)^T\mathbf{\Sigma}_i^{-1}(\mathbf{x}-\mu_i)} \tag{8.3}$$

with $\mathbf{x}=(x,y)^T$. We simplify the calculations by converting the given distribution into polar coordinates and normalizing it. Before doing so, we factorize the covariance matrix $\mathbf{\Sigma}_i$, which simplifies the forthcoming coordinate transform from Cartesian to polar coordinates.

8.2.1 Covariance Factorization

Let us assume a factorable and invertible covariance matrix

$$\mathbf{\Sigma}=\mathbf{U\Lambda U}^T=\underbrace{\mathbf{U\Lambda}^{\frac{1}{2}}}_{\mathbf{A}}(\mathbf{U\Lambda}^{\frac{1}{2}})^T=\mathbf{AA}^T$$

and $\mathbf{\Sigma}^{-1}=\mathbf{A}^{-T}\mathbf{A}^{-1}$ for some invertible matrix $\mathbf{A}$, where

$$\mathbf{\Sigma}=\{\mathbf{AA}^T\mid\mathbf{A}\in\mathbb{R}^{2\times2},\ \mathbf{A}=\mathbf{A}^T,\ \mathbf{x}^T\mathbf{Ax}>0,\ \mathbf{x}\in\mathbb{R}^2,\ \mathbf{x}\neq\mathbf{0}\},$$

which yields

$$\mathcal{N}(\mathbf{x};\mu_i,\mathbf{\Sigma}_i)=\frac{1}{2\pi|\mathbf{A}_i\mathbf{A}_i^T|^{1/2}}\mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\mu_i)^T\mathbf{A}_i^{-T}\mathbf{A}_i^{-1}(\mathbf{x}-\mu_i)}.$$

By choosing the eigendecomposition as a factorization criterion, the columns of $\mathbf{U}$ are unit eigenvectors and $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues. Combining the Cartesian version of (8.1), as well as (8.2) and (8.3) yields

$$\begin{aligned}p(\mathcal{P})&=\iint\limits_{S\subset\mathbb{R}^2}\left(\sum_{i=1}^{M}w_i\cdot\mathcal{N}(\mathbf{x};\mu_i,\mathbf{A}_i\mathbf{A}_i^T)\right)\mathrm{d}x\mathrm{d}y\\&=\sum_{i=1}^{M}w_i\iint\limits_{S\subset\mathbb{R}^2}\Big(\mathcal{N}(\mathbf{x};\mu_i,\mathbf{A}_i\mathbf{A}_i^T)\Big)\mathrm{d}x\mathrm{d}y\\&=\sum_{i=1}^{M}\frac{w_i}{2\pi|\mathbf{A}_i\mathbf{A}_i^T|^{1/2}}\iint\limits_{S\subset\mathbb{R}^2}\mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\mu_i)^T\mathbf{A}_i^{-T}\mathbf{A}_i^{-1}(\mathbf{x}-\mu_i)}\mathrm{d}x\mathrm{d}y,\end{aligned}\tag{8.4}$$

where $\mathbf{x}=(x,y)^T$.

8.2.2 Conversion to Polar Coordinates

Now, we convert (8.4) to polar coordinates by substituting

$$\begin{aligned}x&=r(h,\theta)\cos(\phi)\\y&=r(h,\theta)\sin(\phi),\end{aligned}$$

To change the variables, we need to calculate the Jacobian matrix of $\mathbf{x}$ with its first-order partial derivatives

$$\mathbf{J}=\begin{pmatrix}\frac{\partial x}{\partial r(h,\theta)}&\frac{\partial x}{\partial\phi}\\\frac{\partial y}{\partial r(h,\theta)}&\frac{\partial y}{\partial\phi}\end{pmatrix}=\begin{pmatrix}\cos(\phi)&-r(h,\theta)\sin(\phi)\\\sin(\phi)&r(h,\theta)\cos(\phi)\end{pmatrix}$$

and its determinant

$$
\begin{aligned}
|\det(\mathbf{J})| &= |\cos(\phi) r(h,\theta)\cos(\phi) - (-r(h,\theta))\sin(\phi)\sin(\phi)| \\
&= r(h,\theta)\left|\cos(\phi)^2 + \sin(\phi)^2\right| \\
&= r(h,\theta)
\end{aligned}
$$

which yields $|\det(\mathbf{J})|\mathrm{d}r\mathrm{d}\phi = r(h,\theta)\mathrm{d}r\mathrm{d}\phi = \mathrm{d}x\mathrm{d}y$. After applying the substitution due to the coordinate transform we get

$$
\begin{aligned}
p(\mathcal{P}) = \sum_{i=1}^{M} \frac{w_i}{2\pi|\mathbf{A}_i\mathbf{A}_i^T|^{1/2}} \int\limits_{\phi_m}^{\phi_p} \int\limits_{r(h,\theta_p)}^{r(h,\theta_m)} r(h,\theta) \times \ldots \\
\ldots \times \mathrm{e}^{-\frac{1}{2}\begin{pmatrix} r(h,\theta)\cos(\phi) - \mu_x^{(i)} \\ r(h,\theta)\sin(\phi) - \mu_y^{(i)} \end{pmatrix}^T \mathbf{A}_i^{-T}\mathbf{A}_i^{-1} \begin{pmatrix} r(h,\theta)\cos(\phi) - \mu_x^{(i)} \\ r(h,\theta)\sin(\phi) - \mu_y^{(i)} \end{pmatrix}} \mathrm{d}r\mathrm{d}\phi.
\end{aligned}
\tag{8.5}
$$

8.2.3 Mean Adjusting

Now, we consider mean adjusting [7] in (8.5) by substituting $\mathbf{u}_i = \mathbf{A}_i^{-1}(\mathbf{r}(h,\theta) - \mu_i)$ with $\mathbf{r}(h,\theta) = (r(h,\theta)\cos(\phi), r\sin(\phi))^T$ and $\mu_i = \left(\mu_x^{(i)}, \mu_y^{(i)}\right)^T$, where

$$
\begin{aligned}
\mathbf{u}_i = \begin{pmatrix} u_1^{(i)} \\ u_2^{(i)} \end{pmatrix} &= \mathbf{A}_i^{-1}\begin{pmatrix} r(h,\theta)\cos(\phi) - \mu_x^{(i)} \\ r(h,\theta)\sin(\phi) - \mu_y^{(i)} \end{pmatrix} = \mathbf{B}_i \begin{pmatrix} r(h,\theta)\cos(\phi) - \mu_x^{(i)} \\ r(h,\theta)\sin(\phi) - \mu_y^{(i)} \end{pmatrix} \\
&= \begin{pmatrix} b_{11}^{(i)}(r(h,\theta)\cos(\phi) - \mu_x^{(i)}) + b_{12}^{(i)}(r(h,\theta)\sin(\phi) - \mu_y^{(i)}) \\ b_{21}^{(i)}(r(h,\theta)\cos(\phi) - \mu_x^{(i)}) + b_{22}^{(i)}(r(h,\theta)\sin(\phi) - \mu_y^{(i)}) \end{pmatrix}
\end{aligned}
$$

with

$$
\mathbf{B}_i = \mathbf{A}_i^{-1} = \frac{1}{|\mathbf{A}_i|}\begin{pmatrix} a_{22}^{(i)} & -a_{12}^{(i)} \\ -a_{21}^{(i)} & a_{11}^{(i)} \end{pmatrix},
$$

which is the inverse of $\mathbf{A}_i$. The Jacobian matrix of $\mathbf{u}_i$ is defined as

$$
\mathbf{J}_i = \begin{pmatrix} \frac{\partial u_1^{(i)}}{\partial r(h,\theta)} & \frac{\partial u_1^{(i)}}{\partial \phi} \\ \frac{\partial u_2^{(i)}}{\partial r(h,\theta)} & \frac{\partial u_2^{(i)}}{\partial \phi} \end{pmatrix} = \begin{pmatrix} b_{11}^{(i)}\cos(\phi) + b_{12}^{(i)}\sin(\phi) & -b_{11}^{(i)} r(h,\theta)\sin(\phi) + b_{12}^{(i)} r(h,\theta)\cos(\phi) \\ b_{21}^{(i)}\cos(\phi) + b_{22}^{(i)}\sin(\phi) & -b_{21}^{(i)} r(h,\theta)\sin(\phi) + b_{22}^{(i)} r(h,\theta)\cos(\phi) \end{pmatrix}
$$

with its determinant

$$
|\det(\mathbf{J}_i)| = r(h,\theta)\cdot\left|b_{11}^{(i)}b_{22}^{(i)} - b_{12}^{(i)}b_{21}^{(i)}\right| = r(h,\theta)\cdot|\mathbf{B}_i| = r(h,\theta)\cdot|\mathbf{A}_i^{-1}| = r(h,\theta)\cdot|\mathbf{A}_i|^{-1},
$$

which yields

$$
\mathrm{d}u_1^{(i)}\mathrm{d}u_2^{(i)} = |\det(\mathbf{J}_i)|\mathrm{d}r\mathrm{d}\phi = r(h,\theta)\cdot|\mathbf{A}_i|^{-1}\mathrm{d}r\mathrm{d}\phi = \frac{r(h,\theta)}{|\mathbf{A}_i\mathbf{A}_i^T|^{1/2}}\mathrm{d}r\mathrm{d}\phi.
$$

8.2.4 A Cell's Probability

After applying the substitution and considering a single Gaussian kernel for reasons of clarity and comprehensibility, we get

$$
\begin{aligned}
\widetilde{p}(\mathcal{P}) &= \frac{1}{2\pi} \int\limits_{\zeta_{1,0}}^{\zeta_{1,1}} \int\limits_{\zeta_{2,0}}^{\zeta_{2,1}} \mathrm{e}^{-\dfrac{1}{2}\mathbf{u}^T\mathbf{u}} \mathrm{d}u_1 \mathrm{d}u_2 \\
&= \frac{1}{2\pi} \int\limits_{\zeta_{1,0}}^{\zeta_{1,1}} \int\limits_{\zeta_{2,0}}^{\zeta_{2,1}} \mathrm{e}^{-\dfrac{1}{2}\left(u_1^2 + u_2^2\right)} \mathrm{d}u_1 \mathrm{d}u_2 \\
&= \frac{1}{2\pi} \int\limits_{\zeta_{1,0}}^{\zeta_{1,1}} \left(\frac{\mathrm{e}^{-\dfrac{1}{2}(u_1^2 + \zeta_{2,1}^2)}}{-\zeta_{2,1}} - \frac{\mathrm{e}^{-\dfrac{1}{2}(u_1^2 + \zeta_{2,0}^2)}}{-\zeta_{2,0}} \right) \mathrm{d}u_1 \\
&= \frac{1}{2\pi} \left(\left[\frac{\mathrm{e}^{-\dfrac{1}{2}(\zeta_{1,1}^2 + \zeta_{2,1}^2)}}{(-\zeta_{1,1})(-\zeta_{2,1})} - \frac{\mathrm{e}^{-\dfrac{1}{2}(\zeta_{1,0}^2 + \zeta_{2,1}^2)}}{(-\zeta_{1,0})(-\zeta_{2,1})} \right] - \left[\frac{\mathrm{e}^{-\dfrac{1}{2}(\zeta_{1,1}^2 + \zeta_{2,0}^2)}}{(-\zeta_{1,1})(-\zeta_{2,0})} - \frac{\mathrm{e}^{-\dfrac{1}{2}(\zeta_{1,0}^2 + \zeta_{2,0}^2)}}{(-\zeta_{1,0})(-\zeta_{2,0})} \right] \right) \\
&= \frac{1}{2\pi} \left(\frac{\mathrm{e}^{-\dfrac{1}{2}(\zeta_{1,1}^2 + \zeta_{2,1}^2)}}{\zeta_{1,1} \cdot \zeta_{2,1}} - \frac{\mathrm{e}^{-\dfrac{1}{2}(\zeta_{1,0}^2 + \zeta_{2,1}^2)}}{\zeta_{1,0} \cdot \zeta_{2,1}} - \frac{\mathrm{e}^{-\dfrac{1}{2}(\zeta_{1,1}^2 + \zeta_{2,0}^2)}}{\zeta_{1,1} \cdot \zeta_{2,0}} + \frac{\mathrm{e}^{-\dfrac{1}{2}(\zeta_{1,0}^2 + \zeta_{2,0}^2)}}{\zeta_{1,0} \cdot \zeta_{2,0}} \right) \\
&= \frac{1}{2\pi} \sum_{0 \leq j_1, j_2 \leq 1} (-1)^{j_1 + j_2} \frac{\mathrm{e}^{-\dfrac{1}{2}(\zeta_{1,j_1}^2 + \zeta_{2,j_2}^2)}}{\zeta_{1,j_1} \cdot \zeta_{2,j_2}} .
\end{aligned}
$$

Considering all kernels yields

$$
\boxed{p(\mathcal{P}) = \sum_{i=1}^{M} \frac{w_i}{2\pi} \sum_{0 \leq j_1, j_2 \leq 1} (-1)^{j_1 + j_2} \frac{\mathrm{e}^{-\dfrac{1}{2}\left({\zeta_{1,j_1}^{(i)}}^2 + {\zeta_{2,j_2}^{(i)}}^2\right)}}{\zeta_{1,j_1}^{(i)} \cdot \zeta_{2,j_2}^{(i)}}}
$$

with $j_n \in \{0, 1\}$,

$$
\zeta_{n,0}^{(i)} = b_{n,1}^{(i)} \left(r(h, \Theta_p) \cos(\Phi_m) - \mu_x^{(i)} \right) + b_{n,2}^{(i)} \left(r(h, \Theta_p) \sin(\Phi_m) - \mu_y^{(i)} \right)
$$

and

$$
\zeta_{n,1}^{(i)} = b_{n,1}^{(i)} \left(r(h, \Theta_m) \cos(\Phi_p) - \mu_x^{(i)} \right) + b_{n,2}^{(i)} \left(r(h, \Theta_m) \sin(\Phi_p) - \mu_y^{(i)} \right) .
$$

Now, we can compute each cell's probability and use it as a probability of target presence to improve Bayesian filtering.

Chapter 9

Probability of Target Presence of a Frustum-Shaped Cell Embedded in $\mathbb{R}^3$

Now, let's focus on a realistic scenario: a target is moving within a cuboid spanning a region S, which is split into several cells. To calculate the probability of target presence for a three-dimensional cell C, we have to solve

$$p(C|S;\mathcal{P}) = \iiint\limits_{S\subset\mathbb{R}^3} f_{X,Y,Z}(\mathbf{x};\lambda)\mathrm{d}\mathbf{x} \tag{9.1}$$

$$= \int\limits_{\phi_m}^{\phi_p}\int\limits_{\theta_m}^{\theta_p}\int\limits_{r(h-\Delta h,\theta)}^{r(h+\Delta h,\theta)} f_{H,\Phi,\Theta}(\mathbf{\Psi};\lambda)r(h,\theta)^2\sin(\theta)\mathrm{d}r\mathrm{d}\phi\mathrm{d}\theta \tag{9.2}$$

where $f_{H,\Phi,\Theta}(\mathbf{\Psi};\lambda)$ is a probability density function in spherical coordinates depending on $\mathbf{\Psi} = (r(h,\theta)\sin(\theta)\cos(\phi), r(h,\theta)\sin(\theta)\sin(\phi), r(h,\theta)\cos(\theta))^T$. Set λ contains the mean vector μ and covariance matrix $\mathbf{\Sigma}$. But this time, we need to specify the covariance matrix

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & 0 \\ \sigma_{21} & \sigma_{22} & 0 \\ 0 & 0 & \sigma_{33} \end{pmatrix},$$

if we assume that $f_{X,Y,Z}(\mathbf{\Psi};\lambda) = f_{X,Y}(\mathbf{\Psi};\lambda_1)\cdot f_Z(\mathbf{\Psi};\lambda_2)$, i.e., the random variable Z is independent of X and Y. Thus, $f_Z(\mathbf{\Psi};\lambda_2)$ is constant on a plane that is parallel to the array plane. It is noteworthy to mention that we consider a finite region—the cuboid—in $\mathbb{R}^3$ only; but the Gaussian mixture distribution is defined everywhere on $\mathbb{R}^3$. However, we assume that the target moves within a certain region S. Thus, we have to calculate the conditional probability $p(C|S) = p(C\cap S)/p(S)$, where region $S\subset\mathbb{R}^3$ represents the cuboid, C is an oblique pyramidal frustum-shaped cell and a subregion of S (see Figure 9.1). We will use this notation when we apply the Gaussian mixture distribution.

9.1 Uniform Distribution

In case of a uniform probability of target presence, we set $\lambda = \{C_\Sigma\}$ and $f(\mathbf{\Psi};\lambda) = 1/C_\Sigma$, where C_Σ is the sum of all cell volumes $\{C^{(i)}\}_{i=1}^{N_{\phi,\theta}}$, i.e., the cuboid. Thus,

$$\begin{aligned}
p(C;\mathcal{P}) &= \int\limits_{\phi_m}^{\phi_p}\int\limits_{\theta_m}^{\theta_p}\int\limits_{r(h-\Delta h,\theta)}^{r(h+\Delta h,\theta)} \frac{1}{C_\Sigma} r^2(h,\theta)\sin(\theta)\mathrm{d}r\mathrm{d}\phi\mathrm{d}\theta \\
&= \frac{1}{C_\Sigma}\int\limits_{\phi_m}^{\phi_p}\int\limits_{\theta_m}^{\theta_p} \frac{r(h+\Delta h,\theta)^3 - r(h-\Delta h,\theta)^3}{3}\sin(\theta)\mathrm{d}\phi\mathrm{d}\theta \\
&= \frac{(-1)}{C_\Sigma}\int\limits_{\phi_m}^{\phi_p}\int\limits_{\theta_m}^{\theta_p} \left((h+\Delta h)^3\cos(\theta)^{-3} - (h-\Delta h)^3\cos(\theta)^{-3}\right)\frac{\sin(\theta)}{3}\mathrm{d}\phi\mathrm{d}\theta \\
&= \frac{(h+\Delta h)^3 - (h-\Delta h)^3}{3C_\Sigma}\int\limits_{\phi_m}^{\phi_p}\int\limits_{\theta_m}^{\theta_p} \frac{\sin(\theta)}{-\cos(\theta)^{-3}}\mathrm{d}\phi\mathrm{d}\theta.
\end{aligned}$$

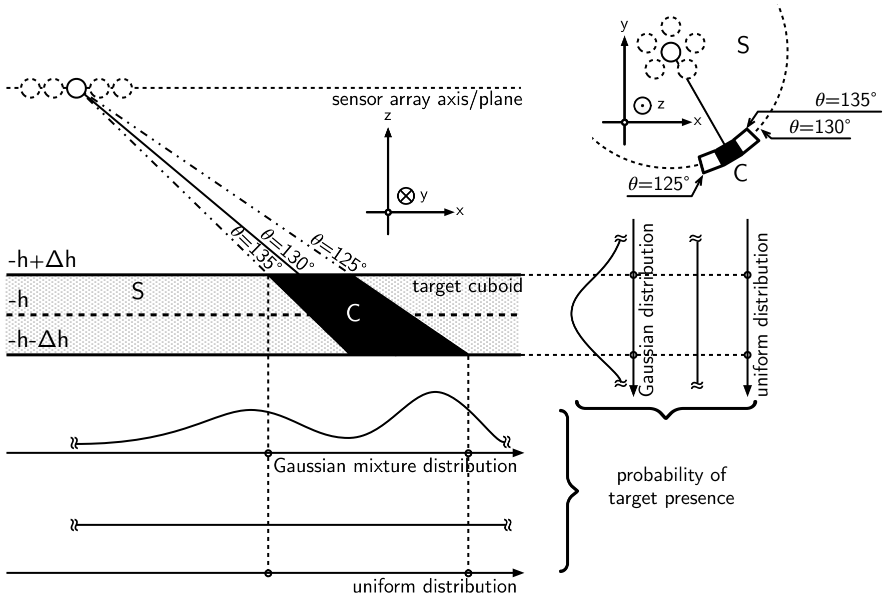

Figure 9.1: A cell C (black area) within the target cuboid S (shaded area) for $\theta = 130°$.

Substituting $u = \cos(\theta)$ with $\mathrm{d}\theta = -\mathrm{d}u/\sin(\theta)$ results in

$$
\begin{aligned}
p(C;\mathcal{P}) &= \frac{(h+\Delta h)^3 - (h-\Delta h)^3}{3C_\Sigma} \int\limits_{\phi_m}^{\phi_p} \int\limits_{\upsilon_m}^{\upsilon_p} \frac{1}{u^3} \mathrm{d}\phi \mathrm{d}\theta \\
&= \frac{(h+\Delta h)^3 - (h-\Delta h)^3}{6C_\Sigma} \int\limits_{\phi_m}^{\phi_p} \left(\frac{1}{\upsilon_m^2} - \frac{1}{\upsilon_p^2} \right) \mathrm{d}\phi \mathrm{d}\phi \mathrm{d}\theta \\
&= \frac{\Delta\phi \left((h+\Delta h)^3 - (h-\Delta h)^3 \right)}{6C_\Sigma} \left(\frac{1}{\upsilon_m^2} - \frac{1}{\upsilon_p^2} \right).
\end{aligned}
$$

Considering $u = \cos(\theta)$ leads to

$$
\begin{aligned}
p(C;\mathcal{P}) &= \frac{\Delta\phi \left((h+\Delta h)^3 - (h-\Delta h)^3 \right)}{6C_\Sigma} \left(\frac{1}{\cos(\theta_m)^2} - \frac{1}{\cos(\theta_p)^2} \right) \\
&= \frac{\Delta\phi \left((h+\Delta h)^3 - (h-\Delta h)^3 \right)}{6C_\Sigma} \left(\sec(\theta_m)^2 - \sec(\theta_p)^2 \right)
\end{aligned}
$$

with

$$
\boxed{p(C;\mathcal{P}) = \frac{\Delta\phi \left((h+\Delta h)^3 - (h-\Delta h)^3 \right)}{6C_\Sigma} \left(\tan(\theta_m)^2 - \tan(\theta_p)^2 \right)}
$$

as the final result.

9.2 Gaussian Mixture Distribution

In case of a Gaussian mixture distribution, the parameter set $\lambda = \{\mu_i, \mathbf{\Sigma}_i, w_i\}_{i=1}^{M}$, where M is the number of Gaussian kernels, μ_i and $\mathbf{\Sigma}_i$ represents the mean vector and the covariance matrix of the Gaussian kernel with index i, respectively. The parameter w_i is the corresponding weight for this kernel. Thus, let

$$
f(\mathbf{x};\lambda) = \sum_{i=1}^{M} w_i \cdot \mathcal{N}(\mathbf{x};\mu_i,\mathbf{\Sigma}_i), \tag{9.3}
$$

be a two-dimensional Gaussian mixture probability density function. The multivariate (here: bivariate) Gaussian distribution in Cartesian coordinates is defined as

$$
\mathcal{N}(\mathbf{x};\mu_i,\mathbf{\Sigma}_i) = \frac{1}{(2\pi)^{3/2}|\mathbf{\Sigma}_i|^{1/2}} \mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\mu_i)^T \mathbf{\Sigma}_i^{-1}(\mathbf{x}-\mu_i)} \tag{9.4}
$$

with $\mathbf{x} = (x, y, z)^T$. We can simplify the calculations by converting the given distribution into polar coordinates and normalizing it. Before doing so, we can factorize the covariance matrix $\mathbf{\Sigma}$, which simplifies the forthcoming substitution.

9.2.1 Covariance Factorization

Let us assume a factorable and invertible covariance matrix

$$
\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T = \underbrace{\mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}}_{\mathbf{A}}(\mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}})^T = \mathbf{A}\mathbf{A}^T
$$

and $\mathbf{\Sigma}^{-1} = \mathbf{A}^{-T}\mathbf{A}^{-1}$ for some invertible matrix $\mathbf{A}$, where

$$
\mathbf{\Sigma} = \{\mathbf{A}\mathbf{A}^T \mid \mathbf{A} \in \mathbb{R}^{3\times 3},\ \mathbf{A} = \mathbf{A}^T,\ \mathbf{x}^T\mathbf{A}\mathbf{x} > 0,\ \mathbf{x} \in \mathbb{R}^3,\ \mathbf{x} \neq \mathbf{0}\},
$$

which yields

$$\mathcal{N}(\mathbf{x};\mu_i,\mathbf{\Sigma}_i)=\frac{1}{(2\pi)^{3/2}|\mathbf{A}_i\mathbf{A}_i^T|^{1/2}}\mathrm{e}^{-\dfrac{1}{2}(\mathbf{x}-\mu_i)^T\mathbf{A}_i^{-T}\mathbf{A}_i^{-1}(\mathbf{x}-\mu_i)}.$$

By choosing the eigendecomposition as a factorization criterion, the columns of $\mathbf{U}$ are unit eigenvectors and $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues. Combining (9.1), (9.3), and (9.4) yields

$$\begin{aligned}
p(C|S;\mathcal{P}) &= \frac{1}{p(S)}\iiint\limits_{C\subset\mathbb{R}^3}\left(\sum_{i=1}^{M} w_i\cdot\mathcal{N}(\mathbf{x};\mu_i,\mathbf{A}_i\mathbf{A}_i^T)\right)\mathrm{d}x\mathrm{d}y\\
&= \frac{1}{p(S)}\sum_{i=1}^{M} w_i\iiint\limits_{C\subset\mathbb{R}^3}\left(\mathcal{N}(\mathbf{x};\mu_i,\mathbf{A}_i\mathbf{A}_i^T)\right)\mathrm{d}x\mathrm{d}y\\
&= \frac{1}{p(S)}\sum_{i=1}^{M}\frac{w_i}{(2\pi)^{3/2}|\mathbf{A}_i\mathbf{A}_i^T|^{1/2}}\iiint\limits_{C\subset\mathbb{R}^3}\mathrm{e}^{-\dfrac{1}{2}(\mathbf{x}-\mu_i)^T\mathbf{A}_i^{-T}\mathbf{A}_i^{-1}(\mathbf{x}-\mu_i)}\mathrm{d}x\mathrm{d}y,
\end{aligned}\tag{9.5}$$

with

$$p(S)=\sum_{i=1}^{M}\frac{w_i}{(2\pi)^{3/2}|\mathbf{A}_i\mathbf{A}_i^T|^{1/2}}\iiint\limits_{S\subset\mathbb{R}^3}\mathrm{e}^{-\dfrac{1}{2}(\mathbf{x}-\mu_i)^T\mathbf{A}_i^{-T}\mathbf{A}_i^{-1}(\mathbf{x}-\mu_i)}\mathrm{d}x\mathrm{d}y,$$

where $\mathbf{x}=(x,y,z)^T$

9.2.2 Conversion to Spherical Coordinates

Now, we convert (9.5) to spherical coordinates by substituting

$$\begin{aligned}
x &= r(h,\theta)\sin(\theta)\cos(\phi)\\
y &= r(h,\theta)\sin(\theta)\sin(\phi)\\
z &= r(h,\theta)\cos(\theta)
\end{aligned}$$

where $r=\sqrt{x^2+y^2+(-h)^2}$, $\phi=\arctan(y/x)$, and $\theta=\arccos(-h/r(h,\theta))$. To change the variables, we need to calculate the Jacobian matrix with its first-order partial derivatives

$$\begin{aligned}
\mathbf{J} &= \begin{pmatrix} \frac{\partial x}{\partial r(h,\theta)} & \frac{\partial x}{\partial\phi} & \frac{\partial x}{\partial\theta}\\ \frac{\partial y}{\partial r(h,\theta)} & \frac{\partial y}{\partial\phi} & \frac{\partial y}{\partial\theta}\\ \frac{\partial z}{\partial r(h,\theta)} & \frac{\partial z}{\partial\phi} & \frac{\partial z}{\partial\theta}\end{pmatrix}\\
&= \begin{pmatrix} \sin(\theta)\cos(\phi) & -r(h,\theta)\sin(\theta)\sin(\phi) & r(h,\theta)\cos(\theta)\cos(\phi)\\ \sin(\theta)\sin(\phi) & r(h,\theta)\sin(\theta)\cos(\theta) & r(h,\theta)\cos(\theta)\sin(\phi)\\ \cos(\theta) & 0 & -r(h,\theta)\sin(\theta)\end{pmatrix}
\end{aligned}$$

and its determinant

$$|\det(\mathbf{J})|=r(h,\theta)^2\sin(\theta),$$

which yields $|\det(\mathbf{J})|\mathrm{d}r\mathrm{d}\phi\mathrm{d}\theta = r(h,\theta)^2\sin(\theta)\mathrm{d}r\mathrm{d}\phi\mathrm{d}\theta = \mathrm{d}x\mathrm{d}y$. After applying the substitution we get

$$p(C|S;\mathcal{P}) = \frac{1}{p(S)}\sum_{i=1}^{M}\frac{w_i}{(2\pi)^{3/2}|\mathbf{A}_i\mathbf{A}_i^T|^{1/2}}\int\limits_{\phi_m}^{\phi_p}\int\limits_{\theta_m}^{\theta_p}\int\limits_{r(h-\Delta h,\theta)}^{r(h+\Delta h,\theta)} r(h,\theta)^2\sin(\theta)\times\ldots$$

$$\ldots\times \mathrm{e}^{-\frac{1}{2}\begin{pmatrix} r(h,\theta)\sin(\theta)\cos(\phi)-\mu_x^{(i)} \\ r(h,\theta)\sin(\theta)\sin(\phi)-\mu_y^{(i)} \\ r(h,\theta)\cos(\theta)-\mu_z^{(i)}\end{pmatrix}^T \mathbf{A}_i^{-T}\mathbf{A}_i^{-1}\begin{pmatrix} r(h,\theta)\sin(\theta)\cos(\phi)-\mu_x^{(i)} \\ r(h,\theta)\sin(\theta)\sin(\phi)-\mu_y^{(i)} \\ r(h,\theta)\cos(\theta)-\mu_z^{(i)}\end{pmatrix}}\mathrm{d}r\mathrm{d}\phi\mathrm{d}\theta \tag{9.6}$$

9.2.3 Mean Adjusting

Now, we consider mean adjusting [7] by substituting $\mathbf{u}_i = \mathbf{A}_i^{-1}(\mathbf{r}(h,\theta)-\mu_i)$ with $\mathbf{r}(h,\theta) = (r(h,\theta)\sin(\theta)\cos(\phi), r(h,\theta)\sin(\theta)\sin(\phi), r(h,\theta)\cos(\theta))^T$ and $\mu_i = \left(\mu_x^{(i)},\mu_y^{(i)},\mu_y^{(i)}\right)^T$ in (9.6), where

$$\mathbf{u}_i = \begin{pmatrix} u_1^{(i)} \\ u_2^{(i)} \\ u_3^{(i)}\end{pmatrix} = \mathbf{A}_i^{-1}\begin{pmatrix} r(h,\theta)\sin(\theta)\cos(\phi)-\mu_x^{(i)} \\ r(h,\theta)\sin(\theta)\sin(\phi)-\mu_y^{(i)} \\ r(h,\theta)\cos(\theta)-\mu_z^{(i)}\end{pmatrix} = \mathbf{B}_i\begin{pmatrix} r(h,\theta)\sin(\theta)\cos(\phi)-\mu_x^{(i)} \\ r(h,\theta)\sin(\theta)\sin(\phi)-\mu_y^{(i)} \\ r(h,\theta)\cos(\theta)-\mu_z^{(i)}\end{pmatrix}$$

$$= \begin{pmatrix} b_{11}^{(i)}(r(h,\theta)\sin(\theta)\cos(\phi)-\mu_x^{(i)}) + b_{12}^{(i)}(r(h,\theta)\sin(\theta)\sin(\phi)-\mu_y^{(i)}) + b_{13}^{(i)}(r(h,\theta)\cos(\theta)-\mu_z^{(i)})) \\ b_{21}^{(i)}(r(h,\theta)\sin(\theta)\cos(\phi)-\mu_x^{(i)}) + b_{22}^{(i)}(r(h,\theta)\sin(\theta)\sin(\phi)-\mu_y^{(i)}) + b_{23}^{(i)}(r(h,\theta)\cos(\theta)-\mu_z^{(i)}) \\ b_{31}^{(i)}(r(h,\theta)\sin(\theta)\cos(\phi)-\mu_x^{(i)}) + b_{32}^{(i)}(r(h,\theta)\sin(\theta)\sin(\phi)-\mu_y^{(i)}) + b_{33}^{(i)}(r(h,\theta)\cos(\theta)-\mu_z^{(i)})\end{pmatrix}$$

with $\mathbf{B}_i = \mathbf{A}_i^{-1}$, which is the inverse of $\mathbf{A}_i$. The Jacobian matrix is defined as

$$\mathbf{J} = \begin{pmatrix} \frac{\partial x}{\partial r(h,\theta)} & \frac{\partial x}{\partial\phi} & \frac{\partial x}{\partial\theta} \\ \frac{\partial y}{\partial r(h,\theta)} & \frac{\partial y}{\partial\phi} & \frac{\partial y}{\partial\theta} \\ \frac{\partial z}{\partial r(h,\theta)} & \frac{\partial z}{\partial\phi} & \frac{\partial z}{\partial\theta}\end{pmatrix},$$

and the absolute value of the determinant is

$$|\det(\mathbf{J}_i)| = |\mathbf{B}_i| r(h,\theta)^2\sin(\theta),$$

which yields

$$\mathrm{d}u_1^{(i)}\mathrm{d}u_2^{(i)}\mathrm{d}u_3^{(i)} = |\det(\mathbf{J}_i)|\mathrm{d}r\mathrm{d}\phi\mathrm{d}\theta = |\mathbf{B}_i| r(h,\theta)^2\sin(\theta)\mathrm{d}r\mathrm{d}\phi\mathrm{d}\theta = \frac{r(h,\theta)^2\sin(\theta)}{|\mathbf{A}_i\mathbf{A}_i^T|^{1/2}}\mathrm{d}r\mathrm{d}\phi\mathrm{d}\theta.$$

9.2.4 A Cell's Probability

After applying the substitution and considering a single kernel and focussing on the intersection of $C \cap S$ for reasons of clarity and comprehensibility, we get

$$
\begin{aligned}
p(C \cap S) &= \frac{1}{(2\pi)^{3/2}} \int\limits_{\zeta_{1,0}}^{\zeta_{1,1}} \int\limits_{\zeta_{2,0}}^{\zeta_{2,1}} \int\limits_{\zeta_{3,0}}^{\zeta_{3,1}} \mathrm{e}^{-\frac{1}{2}\mathbf{u}^T\mathbf{u}} \mathrm{d}u_1 \mathrm{d}u_2 \mathrm{d}u_3 \\
&= \frac{1}{(2\pi)^{3/2}} \int\limits_{\zeta_{1,0}}^{\zeta_{1,1}} \int\limits_{\zeta_{2,0}}^{\zeta_{2,1}} \int\limits_{\zeta_{3,0}}^{\zeta_{3,1}} \mathrm{e}^{-\frac{1}{2}\left(u_1^2 + u_2^2 + u_3^3\right)} \mathrm{d}u_1 \mathrm{d}u_2 \mathrm{d}u_3 \\
&= \frac{1}{(2\pi)^{3/2}} \int\limits_{\zeta_{1,0}}^{\zeta_{1,1}} \int\limits_{\zeta_{2,0}}^{\zeta_{2,1}} \left(\frac{\mathrm{e}^{-\frac{1}{2}(u_1^2 + u_2^2 + \zeta_{3,1}^2)}}{-\zeta_{3,1}} - \frac{\mathrm{e}^{-\frac{1}{2}(u_1^2 + u_2^2 + \zeta_{3,0}^2)}}{-\zeta_{3,0}} \right) \mathrm{d}u_1 \mathrm{d}u_2 \\
&= \frac{1}{(2\pi)^{3/2}} \int\limits_{\zeta_{1,0}}^{\zeta_{1,1}} \left(\frac{\mathrm{e}^{-\frac{1}{2}(u_1^2 + \zeta_{2,1}^2 + \zeta_{3,1}^2)}}{(-\zeta_{2,1})(-\zeta_{3,1})} - \frac{\mathrm{e}^{-\frac{1}{2}(u_1^2 + \zeta_{2,0}^2 + \zeta_{3,1}^2)}}{(-\zeta_{2,0})(-\zeta_{3,1})} - \dots \right. \\
&\qquad \left. \dots - \frac{\mathrm{e}^{-\frac{1}{2}(u_1^2 + \zeta_{2,1}^2 + \zeta_{3,0}^2)}}{(-\zeta_{2,1})(-\zeta_{3,0})} + \frac{\mathrm{e}^{-\frac{1}{2}(u_1^2 + \zeta_{2,0}^2 + \zeta_{3,0}^2)}}{(-\zeta_{2,0})(-\zeta_{3,0})} \right) \mathrm{d}u_1 \\
&= \frac{1}{(2\pi)^{3/2}} \left(\frac{\mathrm{e}^{-\frac{1}{2}(\zeta_{1,1}^2 + \zeta_{2,1}^2 + \zeta_{3,1}^2)}}{(-\zeta_{1,1})(-\zeta_{2,1})(-\zeta_{3,1})} - \frac{\mathrm{e}^{-\frac{1}{2}(\zeta_{1,0}^2 + \zeta_{2,1}^2 + \zeta_{3,1}^2)}}{(-\zeta_{1,0})(-\zeta_{2,1})(-\zeta_{3,1})} - \frac{\mathrm{e}^{-\frac{1}{2}\zeta_{1,1}^2 + \zeta_{2,0}^2 + \zeta_{3,1}^2)}}{(-\zeta_{1,1})(-\zeta_{2,0})(-\zeta_{3,1})} + \dots \right. \\
&\qquad \dots + \frac{\mathrm{e}^{-\frac{1}{2}\zeta_{1,0}^2 + \zeta_{2,0}^2 + \zeta_{3,1}^2)}}{(-\zeta_{1,0})(-\zeta_{2,0})(-\zeta_{3,1})} - \frac{\mathrm{e}^{-\frac{1}{2}(\zeta_{1,1}^2 + \zeta_{2,1}^2 + \zeta_{3,0}^2)}}{(-\zeta_{1,1})(-\zeta_{2,1})(-\zeta_{3,0})} + \frac{\mathrm{e}^{-\frac{1}{2}(\zeta_{1,0}^2 + \zeta_{2,1}^2 + \zeta_{3,0}^2)}}{(-\zeta_{1,0})(-\zeta_{2,1})(-\zeta_{3,0})} + \dots \\
&\qquad \left. \dots + \frac{\mathrm{e}^{-\frac{1}{2}(\zeta_{1,1}^2 + \zeta_{2,0}^2 + \zeta_{3,0}^2)}}{(-\zeta_{1,1})(-\zeta_{2,0})(-\zeta_{3,0})} - \frac{\mathrm{e}^{-\frac{1}{2}(\zeta_{1,0}^2 + \zeta_{2,0}^2 + \zeta_{3,0}^2)}}{(-\zeta_{1,0})(-\zeta_{2,0})(-\zeta_{3,0})} \right) \\
&= \frac{1}{(2\pi)^{3/2}} \left(-\frac{\mathrm{e}^{-\frac{1}{2}(\zeta_{1,1}^2 + \zeta_{2,1}^2 + \zeta_{3,1}^2)}}{\zeta_{1,1}\zeta_{2,1}\zeta_{3,1}} + \frac{\mathrm{e}^{-\frac{1}{2}(\zeta_{1,0}^2 + \zeta_{2,1}^2 + \zeta_{3,1}^2)}}{\zeta_{1,0} \cdot \zeta_{2,1} \cdot \zeta_{3,1}} + \frac{\mathrm{e}^{-\frac{1}{2}\zeta_{1,1}^2 + \zeta_{2,0}^2 + \zeta_{3,1}^2)}}{\zeta_{1,1} \cdot \zeta_{2,0} \cdot \zeta_{3,1}} - \dots \right. \\
&\qquad \dots - \frac{\mathrm{e}^{-\frac{1}{2}\zeta_{1,0}^2 + \zeta_{2,0}^2 + \zeta_{3,1}^2)}}{\zeta_{1,0} \cdot \zeta_{2,0} \cdot \zeta_{3,1}} + \frac{\mathrm{e}^{-\frac{1}{2}(\zeta_{1,1}^2 + \zeta_{2,1}^2 + \zeta_{3,0}^2)}}{\zeta_{1,1} \cdot \zeta_{2,1} \cdot \zeta_{3,0}} - \frac{\mathrm{e}^{-\frac{1}{2}(\zeta_{1,0}^2 + \zeta_{2,1}^2 + \zeta_{3,0}^2)}}{\zeta_{1,0} \cdot \zeta_{2,1} \cdot \zeta_{3,0}} - \dots \\
&\qquad \left. \dots - \frac{\mathrm{e}^{-\frac{1}{2}(\zeta_{1,1}^2 + \zeta_{2,0}^2 + \zeta_{3,0}^2)}}{\zeta_{1,1} \cdot \zeta_{2,0} \cdot \zeta_{3,0}} + \frac{\mathrm{e}^{-\frac{1}{2}(\zeta_{1,0}^2 + \zeta_{2,0}^2 + \zeta_{3,0}^2)}}{\zeta_{1,0} \cdot \zeta_{2,0} \cdot \zeta_{3,0}} \right) \\
&= \frac{1}{(2\pi)^{3/2}} \sum_{0 \le j_1, j_2, j_3 \le 1} (-1)^{j_1+j_2+j_3} \frac{\mathrm{e}^{-\frac{1}{2}(\zeta_{1,j_1}^2 + \zeta_{2,j_2}^2 + \zeta_{3,j_3}^2)}}{\zeta_{1,j_1} \cdot \zeta_{2,j_2} \cdot \zeta_{3,j_3}}.
\end{aligned}
$$

Considering all kernels and the conditional probability yields

$$\boxed{p(C|S;\mathcal{P}) = \frac{1}{p(S)} \sum_{i=1}^{M} \frac{w_i}{(2\pi)^{3/2}} \sum_{0\leq j_1,j_2,j_3\leq 1} (-1)^{j_1+j_2+j_3} \frac{\mathrm{e}^{-\frac{1}{2}\left({\zeta^{(i)}_{1,j_1}}^2 + {\zeta^{(i)}_{2,j_2}}^2 + {\zeta^{(i)}_{3,j_3}}^2\right)}}{\zeta^{(i)}_{1,j_1} \cdot \zeta^{(i)}_{2,j_2} \cdot \zeta^{(i)}_{3,j_3}}}$$

with $j_n \in \{0,1\}$,

$$\begin{aligned}
\zeta^{(i)}_{n,0} = {} & b^{(i)}_{n,1}\left(r(h-\Delta h,\theta)\sin(\theta_m)\cos(\phi_m) - \mu^{(i)}_x\right) \\
& + b^{(i)}_{n,2}\left(r(h-\Delta h,\theta)\sin(\theta_m)\sin(\phi_m) - \mu^{(i)}_y\right) \\
& + b^{(i)}_{n,3}\left(r(h-\Delta h,\theta)\cos(\theta_m) - \mu^{(i)}_z\right)
\end{aligned}$$

and

$$\begin{aligned}
\zeta^{(i)}_{n,1} = {} & b^{(i)}_{n,1}\left(r(h+\Delta h,\theta)\sin(\theta_p)\cos(\phi_p) - \mu^{(i)}_x\right) \\
& + b^{(i)}_{n,2}\left(r(h+\Delta h,\theta)\sin(\theta_p)\sin(\phi_p) - \mu^{(i)}_y\right) \\
& + b^{(i)}_{n,3}\left(r(h+\Delta h,\theta)\cos(\theta_p) - \mu^{(i)}_z\right),
\end{aligned}$$

and with

$$\boxed{p(S;\mathcal{P}) = \sum_{i=1}^{M} \frac{w_i}{(2\pi)^{3/2}} \sum_{0\leq j_1,j_2,j_3\leq 1} (-1)^{j_1+j_2+j_3} \frac{\mathrm{e}^{-\frac{1}{2}\left({\zeta^{(i)}_{1,j_1}}^2 + {\zeta^{(i)}_{2,j_2}}^2 + {\zeta^{(i)}_{3,j_3}}^2\right)}}{\zeta^{(i)}_{1,j_1} \cdot \zeta^{(i)}_{2,j_2} \cdot \zeta^{(i)}_{3,j_3}}}$$

with $j_n \in \{0,1\}$,

$$\zeta^{(i)}_{n,0} = b^{(i)}_{n,1}\left(r(h-\Delta h,\theta) - \mu^{(i)}_x\right) + b^{(i)}_{n,2}\left(-\mu^{(i)}_y\right) + b^{(i)}_{n,3}\left(-\mu^{(i)}_z\right)$$

and

$$\zeta^{(i)}_{n,1} = b^{(i)}_{n,1}\left(-\mu^{(i)}_x\right) + b^{(i)}_{n,2}\left(-\mu^{(i)}_y\right) + b^{(i)}_{n,3}\left(r(h+\Delta h,\theta) - \mu^{(i)}_z\right).$$

Now, we can compute the cell's probability and use it in many different ways, e.g., as a confidence measure in target detection and tracking.

Chapter 10

Probability of Target Presence in $\mathbb{R}^N$

Just for completeness, a more general solution for an N dimensional multivariate Gaussian mixture distribution with hyper-spherical coordinates

$$\begin{aligned}
u_1 &= r\sin(\phi_1)\sin(\phi_2)\cdots\sin(\phi_{N-2})\cos(\phi_{N-1})\\
u_2 &= r\sin(\phi_1)\sin(\phi_2)\cdots\sin(\phi_{N-2})\sin(\phi_{N-1})\\
&\vdots\\
u_{N-2} &= r\sin(\phi_1)\sin(\phi_2)\cos(\phi_3)\\
u_{N-1} &= r\sin(\phi_1)\cos(\phi_2)\\
u_N &= r\cos(\phi_1)
\end{aligned}$$

with

$$r=\sqrt{\sum_{n=1}^{N}u_n^2},\ \phi_1=\cos^{-1}\left(\frac{u_1}{\sqrt{\sum\limits_{n=1}^{N}u_n^2}}\right),\ \cdots,\ \phi_{N-1}=\cos^{-1}\left(\frac{u_{N-1}}{\sqrt{\sum\limits_{n=N-1}^{N}u_n^2}}\right)$$

is

$$p(C|S;\mathcal{P})=\frac{1}{p(S)}\sum_{i=1}^{M}\frac{w_i}{(2\pi)^{N/2}}\sum_{0\le j_1,j_2,\ldots,j_N\le 1}(-1)^{j_1+\ldots+j_N}\frac{\mathrm{e}^{-\frac{1}{2}\left(\sum\limits_{n=1}^{N}\zeta_{n,j_n}^{(i)\ 2}\right)}}{\prod\limits_{n=1}^{N}\zeta_{n,j_n}^{(i)}}.$$

with $j_n\in\{0,1\}$,

$$\zeta_{n,0}^{(i)}=\sum_{k=1}^{N}b_{n,k}^{(i)}\left(u_k(\phi_m)-\mu_k^{(i)}\right)$$

and

$$\zeta_{n,1}^{(i)}=\sum_{k=1}^{N}b_{n,k}^{(i)}\left(u_k(\phi_p)-\mu_k^{(i)}\right).$$

The scaling factor $p(S;\mathcal{P})$ is analogous to the scaling factor in the previous section except that we have to consider the higher-dimensional extension and its corresponding angles. However, the use of a four- or higher-dimensional solution goes beyond the scope of this report.

Chapter 11

Conclusion & Outlook

We introduced a mathematical framework that describes the nonlinear spatial sampling and quantization properties of a single planar acoustic sensor array mounted on the ceiling of a room. Furthermore, we answered the question how to calculate the shapes and volumes of the nonuniform spatial quantization cells to avoid assuming false linear physical conditions and, as a consequence, to increase a distant speech recognition system's accuracy. Beside that, we show how to generate a probability density function based on a uniform distribution or a multi-variate Gaussian mixture model; it is a prerequisite to compute the probability of target presence. Moreover, we describe how to assign the probability of target presence to a nonuniform cell depending on its size. Considering the nonlinearities can improve a target tracker's performance by assigning the correct distance- or cell-dependent steady-state velocity to a transition model. The use of probability of target presence, which is assigned to each nonuniform cell, can improve the tracking performance due to a modified more accurate likelihood function. In case of a target localizer, we can reject measured clutter in cells featuring a probability of target presence lower then a given threshold.

In our upcoming publications we want to show the increase in performance of a target localizer and tracker by considering these nonlinearities in theoretical and real-world experiments. In addition, we want to generate a general framework for different sensor arrays—one-, two-, and three-dimensional—and different scenarios.

Bibliography

[1] Fondazione Bruno Kessler, "Distant-speech Interaction for Robust Home Applications," http://dirha.fbk.eu, April 2014.

[2] T. Habib and H. Romsdorfer, "Auditory inspired methods for localization of multiple concurrent speakers," *Computer Speech and Language*, vol. 27, pp. 634–659, 2013.

[3] I. Marković and I. Petrović, "Speaker localization and tracking with a microphone array on a mobile robot using von mises distribution and particle filtering," *Robotics and Autonomous Systems*, vol. 58, no. 11, pp. 1185–1196, 2010.

[4] A. Masnadi-Shirazi and B. D. Rao, "An ICA-SCT-PHD Filter Approach for Tracking and Separation of Unknown Time-Varying Number of Sources," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 828–841, April 2013.

[5] D. Matthes, "Discrete Surfaces and Coordinate Systems: Approximation Theorems and Computation," Ph.D. dissertation, School of Mathematics and Natural Sciences, Technical University of Berlin, December 2003.

[6] R. Mahler, *Statistical Multisource-Multitarget Information Fusion.* Artech House, 2007.

[7] C. B. Do, "The Multivariate Gaussian Distribution," Section Notes, Lecture on Machine Learning, CS 229, Stanford University, October 2008.