UNIVERSITÀ DEGLI STUDI DI TRENTO

DEPARTMENT INFORMATION ENGINEERING AND COMPUTER SCIENCE

MASTER OF SCIENCE IN TELECOMMUNICATIONS ENGINEERING

THESIS

VIPROT: A VISUAL PROBABILISTIC MODEL FOR MOVING INTEREST POINTS CLUSTERS TRACKING

Thesis Advisors:
Nicola Conci, University of Trento
Radu Horaud, INRIA Rhône-Alpes

Student:
Alessio Xompero

Co-Advisors :
Xavier Alameda-Pineda and Silèye Ba
INRIA Rhône-Alpes

ACADEMIC YEAR 2013-2014

Contents

Acknowledgments

First of all, I would like to thank Prof. Radu Patrice Horaud for having accepted me in his team, PERCEPTION, at INRIA Grenoble Rhône-Alpes research center, and for having given me the opportunity to realize this master thesis, connected with the project carried out there. Secondly, I would like to thank Nicola Conci, who has been my supervisor at University of Trento. Then, many thanks go to Xavier-Alameda Pineda and Silèye Ba, who ere always available for any clarification during the supervision of this project and gave me advice for understanding the new topics and for improving over time. My stay in Grenoble and at INRIA was really fine thanks to all people of PERCEPTION team (Vincent, Israel, Cosmin, Adrian, Dionyssos, Quentin, Konstantinos, Kaustubh, Georgios, Maxime, Xiaofei, Nathalie, Soraya) and all others I met there. Finally, I want to give thanks to my girlfriend, all my friends (from collegues at university to Salesiani people to all others), and my family for having encouraged me all the time.

Chapter 1

INTRODUCTION

Object tracking is a challenging task widely investigated over years with both militaristic and civilian applications. Tracking is defined as the estimation of the dynamics of a moving object, or in other words as the reconstruction of the smooth trajectories of the objects in a scene.

In the last two decades, computers and video cameras have become more and more affordable and offering high computational performances. Mobile phones have become more sophisticated and spread among people around the world. They embed cameras that allow people to easily take several photos and videos and to share them across Internet. These are great stimuli for the formulation of new paradigms for the research in image and video processing and analysis. One of the topics that has attracted a lot of attention is the reconstruction of object motion in videos, which aims at extracting the trajectories and following the motion of different entities. Object tracking can be applied in various applications such as human-computer interaction, human-robot interaction, traffic monitoring, automated surveillance, motion-based recognition, and it can involve tasks such as gesture recognition or eye gaze tracking.

However, dealing with the motion of multiple objects in video sequences is still an open problem, due to, for instance, the loss of information caused by projection of the 3D world on a 2D image, object representation, feature selection, object de-

tection, data association, variable number of objects, noise in the image processing, complex trajectories, object identification, illumination changes, and occlusions.

This master thesis describes the modelling and developing of a tracking algorithm carried out within the PERCEPTION team at INRIA Rhône-Alpes in Grenoble.

PERCEPTION and VHIA The PERCEPTION team, directed by Dr. Radu Horaud, is a research group which "*investigates and implements computational models for mapping images and sounds onto meaning and onto actions.*"[1] One of the main research interests of the PERCEPTION group is defined by the VHIA[2] (Vision and Hearing in Action) project (funded by the ERC):

> "*VHIA studies the fundamentals of audio-visual perception in the context of human-robot interaction and its objective is to elaborate a holistic computational paradigm of perception and of perception-action loops. VHIA will launch and achieve a unique fine coupling between methodological findings and proof-of-concept implementations using the consumer humanoid NAO, manufactured in France*[3]*. The proposed multi-modal approach is in strong contrast with current computational paradigms that are based on uni-modal biological theories. VHIA bets on the high-risk idea that in the next decades robot technology will have a considerable social and economical impact and that there will be millions of humanoids, in our homes, schools, and offices, which will be able to naturally communicate with us.*"

The Master Project The project stems from a preliminary work, consisting of a method able to detect and localize objects on the basis of audio-visual information [Alameda-Pineda et al., 2011], but it lacked of a way to link the sources over

[1] `http://team.inria.fr/perception/`

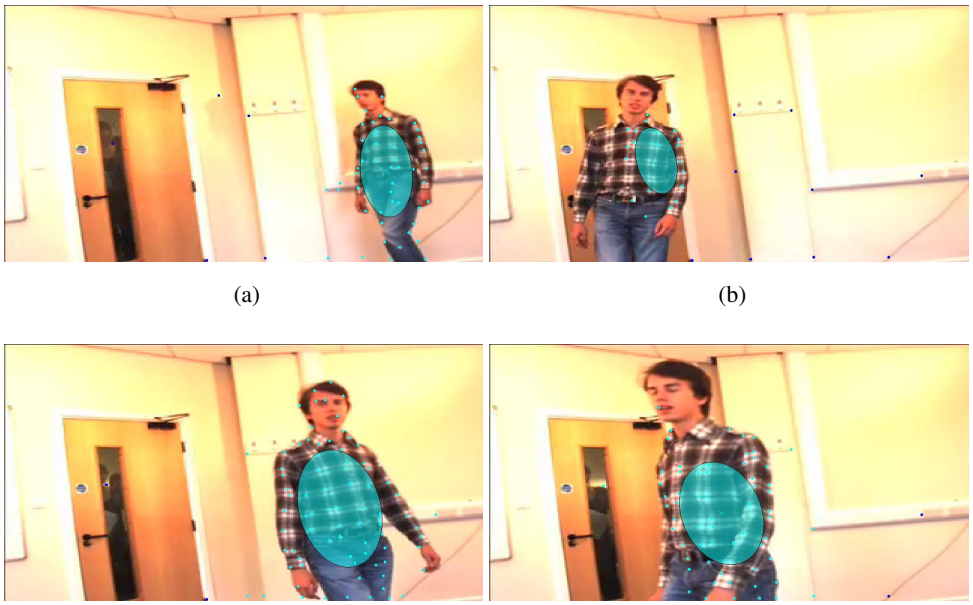
[2] `http://team.inria.fr/perception/vhia`

[3] `http://www.aldebaran-robotics.com/fr`

time. The goal of the master project was to propose a new model able to combine the clustering of visual observations with the tracking of localized sources. Thus, we derived the associated Expectation-Maximization (EM) algorithm for a fixed number of sources. The EM is an iterative algorithm consisting of two phases. The resulting E step did not have closed-form solution. For the sake of saving computing power, and since an approximation is required in any case, we chose to factorize the E-step into two steps, using a variational approximation. The two sub-steps are concerned with observation clustering, on one side, and object tracking, on the other side. The latter is based on the forward-backward algorithm, which allows to smooth the trajectories. The resulting M step, instead, maximizes the log-likelihood, providing with the estimation of the model's parameters.

Thus, the algorithm is tested on different scenarios by using both synthetic and real-world data. While we generated the first ones, we took the second ones from existing datasets and we had to annotate them in order to make appropriate evaluations. The general task of object tracking is to estimate the motion of an object in a certain area, but with multiple objects we need to deal with other issues, like occlusion, object entering or exiting, and object-to-object association over time. In our model, we assumed that the number of objects is known in advance and fixed over time, thus the model does not work in scenarios in which the previous events occur, but it is able to track objects in simple ones. An example is reported in Fig. 1.1 In order to increase the robustness of the original algorithm, we added some extensions to the initial model, such as outliers and velocity, and we made a comparison among the different versions of the algorithm to verify the improvements.

To summarize, the main tasks in which I was involved during this master project were: i) to derive and implement the EM algorithm of the proposed probabilistic graphical model, ii) to test and evaluate it, iii) to extend it to improve the performance and to deal with outliers, and iv) to annotate three real-word video sequences for the evaluation.



(a) (b)

(c) (d)

Figure 1.1: Example of results obtained with the method described in this thesis. The person is represented by a colored elliptical which is the covariance matrix resulting from clustering of observations, as well depicted here. We report some frames to show how the resulting tracking fits the motion of the person.

Thesis organization Chapter 2 presents a brief survey on object tracking from the definition of the general problem to the description of the most common techniques. Chapter 3 describes the proposed probabilistic generative model and the derived algorithm. Some extensions to improve the algorithm are also presented, as well as an explanation of the initialization problem. Experiments using synthetic and real-world data are reported in Chapter 4, making also a comparison among the different versions of the algorithm. Finally, a discussion about limitations and possible future work is presented in Chapter 5. In the Appendices the detailed derivations of the associated EM algorithms can be found.

Chapter 2

STATE OF THE ART

Here, we provide a briefly survey of the literature regarding object tracking and data clustering, which are the tasks involved in the method that we will describe later.

2.1 Object Tracking

Object tracking is a problem widely studied within the field of computer vision, as it is the basis for models for human-computer interaction (HCI), motion-based recognition, traffic monitoring, etc..[Yilmaz et al., 2006].

Generally and in a simple form, tracking can be defined as an estimation of the path of an object within an area of interest due to its movements over time. As reported in [Yilmaz et al., 2006], the aim of an object tracker is to retrieve the trajectory of an object over time by locating its position in every frame of the video. Besides, they defined an object as anything that is of interest for further analysis. For instance, in the area of radar applications an object is referred as a target, and as a source in the audio environment. So, we can find various definitions in the literature that depend on the context. Nevertheless, we will use here all these terms interchangeably together with the term 'object'.

Usually, in order to track an object, it is useful to represent it based on different

characteristics, like shape or appearance. Nevertheless, in this project we only focused on information about object positions and same observations are generated in order to not be meaningful in terms of characterization of a source. So, an object will be described in simplest form as we can think: just as as single point. Obviously, this point could be anything: the centroid of an object,the face of a person or simply the lips of a face. However, the description of objects in terms of more complex representation than a point could be issued in future works.

2.1.1 Visual Tracking

In the beginning, tracking was a problem addressed in military applications to discover and follow enemies with radars. This did not require advanced features to represent the object and only position of a point was enough. Within the computer vision field, instead, we can deal with images that provide much more information and they often are referred to as visual tracking. Together with tracking, we need to define the object representation, the features to extract and an algorithm for detection. [Yilmaz et al., 2006] published a survey about object tracking, providing many details and references, and we report here some concepts.

Object representation: an object can be represented through its shape or appearance. In terms of shape, we can describe an object as a single point, multiple points, a rectangular or elliptical patch, part-based multiple patches, skeleton, contour, control points on contour, and silhouette. In terms of appearance, instead, we can use probability densities, templates, active models, or multi-view models. The choice of which representation is to be used depends on the application domain and the content of the image. For example, if an object is very small in a certain frame, it could be more appropriate to use a single point representation.

Feature selection: strictly linked to object representation, it is really important because it allows to distinguish different objects in the feature space. For this

reason, the uniqueness of the visual feature is the most desirable property, also for tracking algorithms. From images we can extract color information, edges, optical flow, or textures. Many implementations have been proposed on the basis of only one or a combination of these features. Besides, many algorithms have been investigated in order to obtain an automatic feature extraction.

Object detection: once the object representation and the features are selected, we need a mechanism to detect objects either in every frame or when it appears for the first time in the video. Methods can use the information available either in a single frame or from a sequence of frames. The first approach is the most common. We list, now, some common techniques, subdivided into categories, always according to [Yilmaz et al., 2006]. Among point detectors, Harris point [Harris and Stephens, 1988], Kanade-Lucas-Tomasi features [Lucas and Kanade, 1981] and [Shi and Tomasi, 1994], and Scale Invariant Feature Transform [Lowe, 2003] detectors are very well-known; mean-shift was proposed in [Comaniciu and Meer, 1999] as a segmentation technique; mixture of Gaussians [Stauffer and Grimson, 2000] is used to model the background , and Support Vector Machines [Papageorgiou et al., 1998] or Adaptive Boosting [Viola et al., 2003] are mechanisms to learn discriminative object models (e.g. faces, pedestrians, ...) in a supervised way.

2.1.2 The Tracking Model

Only after deciding the characteristics of an object and a method to detect it, we can address the problem of tracking. Tracking has been investigating for a long time: the most famous approach was proposed by Kalman in 1960 [Kalman, 1960]. The easiest way to describe an object is to use a point and tracking can be modelled as

a system of equations from the physics:

$$x_k = f_k\left(x_{k-1}, u_{k-1}, v_{k-1}\right)$$
$$z_k = h_k\left(x_k, w_k\right)$$

where $\{x_k \in \mathbb{R}^{n_x}, k \in \mathbb{N}\}$ is the discrete or continuous time state sequence, f_k is a time dependent function, u_{k-1} is the (known) control input which is not necessarily present, and $\{v_{k-1} \in \mathbb{R}^{n_v}, k \in \mathbb{N}\}$ is an independent and identically distributed (i.i.d) process noise sequence, n_x, n_u, and n_v are the dimensions of the state, input, and process noise vectors respectively, and $\mathbb{N}$ is the set of natural numbers. z_k is the measurement at time k, $\{h_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \rightarrow \mathbb{R}^{n_z}\}$ is again a function, $\{w_k, k \in \mathbb{N}\}$ is an i.i.d. measurement noise sequence, and n_z and n_w are the dimensions of the measurement and measurement noise vectors z_k and w_k.

[Bishop, 2006] uses graphical models to represent this kind of system and the state is described by a latent variable. The latter could be discrete or continuous. While the model is indicated as a hidden Markov model (HMM) in the first case, in the second case if both the latent variable and the observed variables are Gaussian (with a linear-Gaussian dependence of the conditional distributions on their parents), then it refers to the well-know linear dynamical system (LDS).

2.1.3 Kalman Filter

As we said before, in 1960, R.E. Kalman proposed a solution to the continuous-linear (and extended to the discrete domain) filtering problem, which became very famous due to its efficiency and simplicity [Kalman, 1960]. He derived a set of recursive equations under some assumptions, that provides an optimal solution to the Bayesian approach for the tracking problem. Under Gaussian and linearity assumption, the general problem to solve is to estimate the state of a linear dynamical system with the following equation:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{2.1}$$

with a measurement modelled as:

$$z_k = Hx_k + v_k \tag{2.2}$$

The random variables w_k and v_k follow a zero mean Gaussian distribution with covariances that might change over time or with each measurement:

$$p\left(w\right) \sim \mathcal{N}\left(0, Q\right) \tag{2.3}$$

$$p\left(v\right) \sim \mathcal{N}\left(0, R\right) \tag{2.4}$$

They represent the process and measurement noise, respectively. The u_{k-1} variable represents a control input signal, which is not always available or included in the formula.

The algorithm of the Kalman filter is based on two recursive steps: the prediction and the correction. The first one is responsible to make a forward projection in time of the current state,$\hat{x}_{k|k-1}$, and error covariance estimates, $P_{k|k-1}$, which will become a priori information for the next time step. The time update equations are:

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} \tag{2.5}$$

$$P_{k|k-1} = AP_{k-1|k-1}A^{\top} + Q \tag{2.6}$$

The second one, instead, makes a correction of the prediction based on the actual measurement, providing a posteriori estimate (a noisy measurement, $x_{k|k}$) which will be the feedback for a new prediction. The matrix K_k is chosen to be the gain that minimizes the a posteriori error covariance, $P_{k|k}$. The equations for the measurement update are:

$$K_k = P_{k|k-1}H_k^{\top}\left(H_kP_{k|k-1}H_k^{\top} + R\right)^{-1} \tag{2.7}$$

$$x_{k|k} = \hat{x}_{k|k-1} + K_k\left(z_k - H_k\hat{x}_{k|k-1}\right) \tag{2.8}$$

$$P_{k|k} = \left(I - K_kH_k\right)P_{k|k-1} \tag{2.9}$$

Then, the cycle starts again as reported in Fig. 2.1.3.

Kalman filter is a very powerful algorithm because it gives the optimal Bayesian solution to the state estimation problem under linearity and Gaussian assumptions, and it can estimate past, present, and even future states. A practical introduction to the Kalman filter with many more details, such as description, derivation and discussions can be found in [Welch and Bishop, 1995].

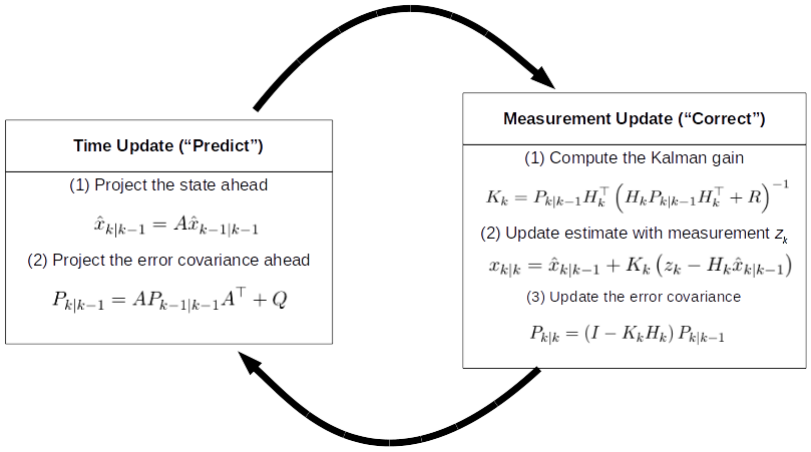


Figure 2.1: The Kalman filter cycle. The *time update* projects the current state estimate ahead in time. The *measurement update* adjusts the projected estimate by an actual measurement at that time. Equations are depicted as well. (This picture is similar to the one in [Welch and Bishop, 1995])

2.1.4 Beyond the Kalman Filter

Before we briefly described the well-known Kalman filter, which is an optimal solution to the Bayesian approach for the tracking problem and it is reliable and efficient for many applications. However, it relies on some restrictive assumptions, such as linearity, Gaussian and uni-modality, which does not always fit the problem that it is addressing. In the case that linearity property does not hold for the process to be estimated, another version of the Kalman filter was proposed: the Extended Kalman filter (EKF). It is based on a linearisation of the current mean and covariance around the working point by means of the first-order Taylor Series expansion. A well-described explanation can be found in [Welch and Bishop,

1995]. Over the years, the EKF became a standard tool for the estimation of the state and/or the parameters of a non-linear system; but, it has some drawbacks such as the difficulty to implement, the difficulty to tune, and it is suitable only for some systems, which can be linearised over time. In the latter, if the assumption of local linearity is violated, it can produce highly unstable filters. Another approach was proposed by [Julier and Uhlmann, 1997], called Unscented Kalman Filter (UKF), in which the state distribution is represented but not restricted to a Gaussian Random Variable (GRV) and it is specified by a minimal set of chosen sample points. "*These sample points capture the true mean and the true covariance of the GRV, and when propagated through the true non-linear system, captures the posterior mean and covariance accurately to the 3rd order Taylor series expansion for any non-linearity*" [Wan and Van Der Merwe, 2000]. The idea behind this algorithm is based on a method for calculating the statistics of a random variable which undergoes a non-linear transformation, called Unscented Transformation [Julier and Uhlmann, 1997]. Nevertheless, some assumptions are still considered, such as Gaussian probability density function and uni-modality of the distribution. There are many problems and applications that cannot rely on them, so we need other approaches for handling multivariate data and non-linear/non-Gaussian processes. Particle filters are well-known techniques suitable for this kind of problem. "*The key idea is to represent the required posterior density function by a set of random samples with associated weights and to compute estimates based on these samples and weights*" [Arulampalam et al., 2002]. We do not go in detail about these concepts here and we refer you to literature for futher information. A comparison of the methods on the basis of the properties of the system is reported in Tab. 2.1.4.

	KF	EKF	UKF	PF
Gaussian	X	X	X	
Uni-modal	X	X	X	
Multi-modal				X
Symmetric	X	X		
Linearity	X			

Table 2.1: Qualitative comparison of the most common tracking techniques based on the main properties of the system. KF: Kalman Filter; EKF: Extendend Kalman Filter; UKF: Unscented Kalman Filter; PF: Particle Filter

2.2 Multiple-Target Tracking

Tracking has been a challenging problem for a long time and it has not been completely solved yet. In the beginning, the objective was to target a single object and we explained the most famous algorithm, also known as Kalman filter, and some other approaches. Because the tracking problem was initially issued for radar applications in the military field, an object was observed as a measurement detected by a sensor and algorithms could work fine in scenarios with only the observation available. When it began to use multi-sensors, new approaches able to distinguish between noisy measurements, or clutter, and the object were developed. The complexity of the problem increased when algorithms should take into account multiple targets, especially in clutter environments.

Introducing clutter and multiple objects arose various issues to solve: determination of the number of sources, which could be time-varying, estimate the data association among objects across consecutive time frames, and estimate the data association between measurements generated from objects and false alarms. Dealing with multiple objects means the number of sources over time is not fixed: one or more object could appear or disappear in a considered region and sometimes only for some period or for the whole period the algorithm is running. In computer

vision applications, there is also a special case, called occlusion, in which the object is completely or partially not visible, but it is still present in the scene. This phenomenon is due to the projection of the 3D real world in a 2D image, making a loss of information.

As far as the data association is concerned, we will explain it in more detail later, but we report here a common assumption which has been used in many tracking algorithms: one object can generate only one measurement per time and one measurement is originated from only one source. Even though it is very common, other algorithms started to rely on a different assumption: more measurements could be associated to an object, talking of extended object in this case.

These are the major issues that a multiple object tracking algorithm try to address, but the problem remains a big challenge. All algorithms proposed until now partially solve the tracking problem, often because of the assumptions, and they can be apply in specific contexts.

So, in this section we explain briefly the data association problem and we describe some well-know multiple-object tracking algorithms.

2.2.1 Data Association problem

One of the main problems that came up when people started dealing with the tracking of multiple objects was to determine a correspondence between the new measurements available in the current time step and the predicted objects coming from the previous instant. Since tracking is a technique widely used in different domains and what has to be tracked is referred with different names, the same happens for the correspondence problem as well. While it is called motion correspondence in computer vision field, in radar application it is well-known as data association problem. As it is well explained in [Cox, 1993], the data association is computed between the prediction and the update stages in order to assign a measurement to a tracker and to go on with the update. Even if it could firstly seem easy to solve, it was and it is a hard problem due to many reasons and that let people to propose

several approaches in many years. In detail, ambiguities may arise when trackers made their predictions and when measurements become available. For instance, it cannot distinguish between an occlusion and an object disappearance when there are no observations around the prediction done by a tracker; or, we cannot say if unexpected measurements are either only false alarms or generated by a new object appeared in the scene; or, if more than one measurement could be available for each prediction, there is no information about which one may be correct and what is the origin of the others. Finally, some measurements could be matched to more than one single track, but this do not satisfy a general assumption in tracking which states that a measurement originates from only one single source [Cox, 1993]. As regards the data association, we do not enter in more detail and we suggest to have a look at references. Now, we go on with the description of some algorithms.

2.2.2 Nearest Neighbour Standard Filter

The first straightforward solution to associate new measurements to tracks is to select the closest one to the predicted measurement. This kind of approach relies on the nearest neighbour search, which is an optimization problem for finding most similar points, or also known as proximity. The latter is based on a dissimilarity function that is defined as a distance metric. Some examples of distance metrics could be the well-known Euclidean distance, the Mahalanobis distance, and others. So, the choice of the closest point is carried out by minimizing the dissimilarity function.

The Nearest Neighbour Standard filter (NNSF) is the implementation of the described algorithm for tracking multiple objects, but it works only under some assumptions. In general, the filter can track any number of targets, but this number should be known in advance and constant in time. The Mahalanobis distance usually is chosen and is calculated as described in the following:

$$\mathrm{d}^2 = \nu_k^T S_k^{-1} \nu_k$$

where S_k is the measurement prediction covariance matrix and ν is the innovation

$(\hat{z}_{k|k-1} - z_n)$.

The main advantage is the low computational complexity, but different and obvious problems can arise. First of all, the nearest neighbour is not always the correct measurement, and then performances of the filter can rapidly degrade with the increasing of mismatches. Another non-trivial problem is when a measurement falls inside the gate of two or more tracks and some of them determine that measurement to be the nearest neighbour to associate. The gate of a track corresponds to an area around the prediction in which measurements could be validated as possible candidates for the correct association.

If the constraint of one measurement to one target is imposed, one measurement can not belong to two targets in the same time and a decision criterion has to be chosen. Here, the Global Nearest Neighbour (GNN) is obtained. One way to overcome the ambiguity problem is to choose the best combination among all possible combinations inside the validation matrix. However, this solution could be feasible only when dealing with very few number of tracks and measurements, due to the computational time (for instance, six or seven objects and/or six or seven measurements are acceptable values to compute all combinations fast enough; higher values could require the algorithm to work for some hours or even more).

2.2.3 Probabilistic Data Association Filter

According to [Cox, 1993], there is a class of suboptimal algorithms, which requires to limit the ambiguity and the high computational problems of other algorithms, such as the NNSF. The *probabilistic data association* filter (PDAF), is part of this class and its main feature is to use weights of all measurements associated to a track. These weights are the probabilities to associate each measurement to a track. Indeed, this filter takes its name from the described property. In reality, the PDAF is not a MTT technique, but it deals with only one track and multiple measurements. Besides, it assumes a linear dynamical system, where both state and measurement are modelled as normal distribution, and this allows to derive the

well-known Kalman filter for the tracking algorithm. So, the PDAF considers the existence of only a single target whose track has been initialized; it defines a validation region to select measurements taking into account; it computes the probabilities corresponding to events that validated measurements are target-originated, and a linear Kalman filter algorithm is applied to each of the validated measurements. We refer to [Bar-Shalom and Fortmann, 1988] for more details and derivations.

2.2.4 Joint Probabilistic Data Association Filter

In realistic scenarios, having only one object to track is not so common, then the *joint probabilistic data association* filter, (JPDAF), is a straightforward extension of the PDAF. It takes into account a fixed number of targets in a clutter environment and it computes the measurement-to-target association as a joint probability across targets. As an extension of PDAF, some assumptions are the same: the last measurements are used and each object has its own linear dynamic and measurement model. This means that linear Kalman filters can be used for tracking, and data association is solved by the joint probabilities. Besides, keeping in mind the basic assumption that each target can generate only one measurement, and each measurement can only originate from one target, this constraint can reduce some computations in the method. Again, details about this technique can be found in [Bar-Shalom and Fortmann, 1988]. JPDAF have been widely used and many improvements and combination with other techniques have been proposed; however, linearity and fixed number of objects are relevant limitations in many scenarios.

2.2.5 Multiple Hypothesis Filter

The multiple-hypothesis filter, (MHF), or also known as multiple-hypothesis tracker, (MHT), is another algorithm proposed in [Reid, 1978]. The purpose is always to track multiple targets in a cluttered environment, but now also to handle objects that enter or exit in the field of view (FOV) or occlusions. The approach is an iterative algorithm based on a set of feasible hypothesis that becomes a node at each

iteration. Each hypothesis consists of a collection of disjoint tracks, and for each of them a prediction of each object's position in the next frame is made. When new measurements become available in the next frame, an association between prediction and actual measurements is computed on the basis of a distance measure, like Mahalanobis distance. As [Cox, 1993] reports: "each measurement may either (1) belong to a previously known geometric feature or (2) be the start of a new geometric feature or (3) be a false alarm, or (4) not belong to any geometric feature, determining the deletion of the geometric feature". All of these generate a set of events that becomes a children hypothesis node in the tree, available for the next iteration.

Because of hypothesis, this tracker postpones the data association decision, but it can result in an exponentially growth of the hypothesis tree and in high computational load. For this reason, different solutions were proposed: one is track and hypothesis pruning [Reid, 1978], and the other is clustering [Blackman, 2004]. The first one is based on a combination of "N-scan-back" algorithm and a simple lower limit probability threshold [Cox, 1993]. (Scan refers to military/radar fields, and frames, instead, to videos). The second one aims at subdividing a set of measurements in smaller sets, making the implementation more efficient.

2.2.6 Probabilistic MHT

[Streit and Luginbuhl, 1995] proposed a new algorithm that tried to solve the high computational complexity problem of the standard MHT due to an hard-association model which enumerates all possible associations between measurements and targets, and evaluate which is the best.

The main idea was to modify the measurement model relying on a probabilistic approach, in which the assignment is provided by a discrete random variable. This changed the assignment from "hard" to "soft" because of probabilities behind the discrete random variables. Besides, the joint estimation of target states and assignment probabilities was carried out by means of an Expectation-Maximization

algorithm. Other techniques, such as JPDAF, Maximum Likelihood Data Association, Gaussian Sum Tracking Filters and Centroid Group Tracking, which faced similar approaches, were compared with PMHT in the report itself.

In [Willett et al., 1995], Willet and Rago analysed the performance of the PMHT compared with the one of the JPDAF in details, while in [Willett et al., 2002] issues, problems and solutions related to PMHT were discussed, always comparing with PDAF. In particular, ten different implementations were analysed underlying weaknesses and strengths for each of them. In the end, they stated the superiority of PMHT with respect to PDAF was not shown, and the preference of one of the two depends on the environment situation: while PMHT could fit better if it is benign and a track is unlikely to be lost, otherwise PDAF could be applied when lost tracks are more relevant.

2.2.7 Probabilistic Hypothesis Density Filter

Until the beginning of 2000s, most of proposed techniques tried to use an explicit association between measurements and targets in order to solve the multiple-target tracking problem. But, even if many improvements and alternatives have been proposed, this explicit association involves a combinatorial problem which results in high computational load; besides, these technique cannot deal with the varying number of targets. However, different approaches such as random finite sets (RFS) or symmetric measurement equations became relevant from 2000s, because they rely on formulations which avoid explicit associations. Especially, the RFS approach was very interesting due to its formulations of collection of individual targets and of individual of observations as set-valued state and set-valued observation, respectively. This allowed to take the problem back into a Bayesian filtering framework, generating novel and promising filters such as multiple-target Bayes filter and the probability hypothesis density filter (PHD).

The latter was investigated and proposed by Mahler in 2002. His purpose was to formulate a recursion in the multiple-target domain which emulates the propa-

gation of the first moment of the single-target state in the Kalman filter equations and it alleviates the computational intractability in the multiple target Bayes filter as well. So, he came up with a solution that propagates the first-order statistical moment, or intensity, of the RFS of states in time, avoiding so the combinatorial problem of the techniques based on an explicit data-association strategy. Nevertheless, even if this recursion has not closed-form solution due to the multiple integrals involved, because of its attractive features many implementations, solutions and improvements have been proposed in the years. For example, Sidenbladh implemented a PHD based method using particle filters, Lin et al. added a track labelling, and Vo et al. proposed an implementation based on Sequential Monte Carlo methods.

Meanwhile, Mahler investigated and proposed an improvement which propagates both the first-order statistics in the state of targets and the entire probability distribution on the number of targets, called cardinality distribution. In this case, the recursion is referred as cardinalized PHD (CPHD). Again, many implementations and improvements have been investigated based on it.

2.2.8 The Gaussian Mixture Probability Hypothesis Density Filter

Among all implementations regarding the PHD filter, a noteworthy one is the Vo and Ma's Gaussian Mixture PHD. They showed that under linear and Gaussian assumptions and when the initial prior intensity of the RFS of targets is a Gaussian mixture, then the posterior intensity as well is a Gaussian mixture at any time. What is relevant is they derived a closed-form recursion for the weights, means and covariances of the constituent Gaussian components of the posterior intensity. They also handled the issue of the growing number of components by using a pruning method. Finally, similarly to implementations of Kalman filter under less constraint assumptions, such as the Extended Kalman Filter and the Unscented Kalman Filter, they derived formulas for an Extended Kalman PHD filter and an Unscented Kalman PHD filter. According to them, these implementations could be

good candidates in highly non-linear and non Gaussian models.

2.3 Clustering

In the beginning of the chapter we explained how to represent an object and a way to do it is by means of multiple points. When dealing with multiple objects, it could be useful to associate points to proper objects and statistical analysis like clustering can be applied to solve this task. Precisely, clustering is the task of grouping a set of data (objects, measurements,...) in such a way that they belong to the same group, or cluster, on the basis of some similarity properties. Many algorithms have been proposed and implemented in different fields such as pattern recognition, machine learning, image analysis, and information retrieval. A precise definition of "cluster" cannot be found and this is the reason why there are so many algorithms. There is a wide literature about clustering, but here we want to focus and present briefly two very famous algorithms: k-means and Gaussian mixture model.

2.3.1 K-means Algorithm

The K-means algorithm is a non-probabilistic technique which partitions a set of data points into K groups, where K is usually given, on the basis of the squared Euclidean distance as a measure of dissimilarity between a data point and a prototype vector, which could be the center of a cluster. In detail, we can think to have a data set $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$ consisting of *N* observations of a random *D*-dimensional Euclidean variable x, and to find an assignment of these data points to clusters in order to minimize the objective function above-mentioned and given by:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2 \tag{2.10}$$

r_{nk} is a binary indicator variable that assigns every data point uniquely to one, and only one, of the clusters, making an 'hard' assignment. The goal of the k-means

algorithm is to find values for $\{r_{nk}\}$ and the $\{\boldsymbol{\mu}_k\}$ in order to minimize the objective function, and this is carried out through an iterative procedure, well-known as Expectation-Maximization algorithm, in which each iteration consists of two stages to minimize J with respect to each variable, keeping the other fixed. This procedure is repeated until the convergence of the algorithms is reached or a fixed number of iterations exceeds. This algorithm is widely used for many applications, such as for instance image segmentation and compression or as initialization in Gaussian mixture models. However, there are some limitations like the used measure, that is the nearest cluster, is the most appropriate as hard assignment. Besides, the basic algorithm is not so fast and different improvements have been proposed. K-medoids is a variant which is based on a more general dissimilarity measure. More details and references are given in [Bishop, 2006].

2.3.2 Gaussian Mixture Model

Another way to cluster a set of data points is by means of mixture of Gaussians, which is simply a linear superposition of Gaussian components and it is well explained in [Bishop, 2006] in terms of graphical probabilistic models. Since the formulation for the Gaussian mixture distribution is:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \tag{2.11}$$

we can introduce a K-dimensional binary discrete latent variable $\boldsymbol{z}$, which determines a 1-of-K scheme and each element is equal to 1 or 0, respecting the constraint that the sum of vector is still one. Thanks to this hidden variable, it is possible to derive equivalent formulations which allow to deal with a joint distribution between the observation and the latent variable $p\left(\boldsymbol{x}, \boldsymbol{z}\right)$, instead of only the the marginal distribution $p\left(\boldsymbol{x}\right)$. Besides, it is possible to define the conditional probability $p\left(\boldsymbol{z}|\boldsymbol{x}\right)$, which leads to derive the *responsibilities* by using Bayes' theorem. In reality, this corresponds to the first phase of the expectation-maximization algorithm which is derived from the previous definitions and formulations. "*The EM algorithm is an*

elegant and powerful method for finding maximum likelihood solutions for models with latent variables." Mixture of Gaussians with an hidden variable relies within the maximum likelihood framework and there is a maximization problem due to the presence of a summation over *k* inside the logarithm. Moreover, the presence of singularities could bring to collapsing problem of a Gaussian component, and it is worth dealing properly with it as well. EM algorithm is an approach that allows to maximize the likelihood function with respect to the parameters in an iteratively way until a convergence criterion is reached. After an initialization of parameters, which are the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixing coefficient π_k, and that could be done with a K-means algorithm, firstly the responsibilities are evaluated by using the current parameters values, and then all parameters are estimated again using the current responsibilities and keeping all other parameters fixed a turn. At each iteration, the log likelihood is evaluated and the convergence of either the parameters or the log likelihood is verified. The EM algorithm requires more iterations to reach convergence and more computations as well with respect to the K-means algorithm, but due to the probabilistic framework , it makes a 'soft' assignment, reflecting better the level of uncertainty in the assignment itself, and it can deal with outliers as well, with a proper formulation. However, there are some limitations: it must avoid singularities of the likelihood function otherwise a Gaussian component can collapse onto a particular data point; and it does not guarantee to converge global maxima of the log likelihood function, since multiple local maxima exist.

2.4 Group Tracking

A particular problem is to track an extended object or a group of objects. In this case, either many measurements belong to the same object or some objects are moving together, so measurements can refer to this group as a single object. For instance, an aircraft team consists of a certain amount of military planes that move together in the sky; generated measurements seem to belong to the same object,

even if it is not so, and you can treat the group as a single entity. Often, a way to deal with a set of measurements that belong to the same entity (object/group) is clustering. So, group tracking is simply a combination of clustering and tracking, and the latter uses results from the first one as measurements. Example of group tracking are: [Clark and Godsill, 2007], [Ishiguro et al., 2008], [Pece, 2002], and [Feldmann et al., 2011].

Chapter 3

PROBABILISTIC TRACKING MODEL

3.1 Modelling Objects' Positions

The main objective of the work is to estimate the track of some sources which generate multiple measurements at each instant time. We want to describe the previous statement on the basis of a probabilistic graphical model, as shown in Fig. 3.1. The number of sources is denoted by N and it assumed to be known in advance and constant for the moment. Besides, it is considered a temporal window of length T. The number of observations, instead, depends on time, and it is denoted by K_t.

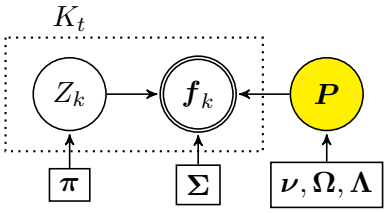


Figure 3.1: The graphical model to solve.

Sources' Position $\boldsymbol{P}_{t,n}$ is an hidden and temporal variable which denotes the position of the n^{th} source at time t. Each source is supposed to follow a zero-

order Gaussian model. Indeed:

$$\boldsymbol{P}_{t,n}|\boldsymbol{P}_{t-1,n} \sim \mathcal{N}\left(\boldsymbol{P}_{t,n}; \boldsymbol{P}_{t-1,n}, \boldsymbol{\Lambda}_n\right).$$

Since a temporal variable has an initial point, we report its initial model here:

$$\boldsymbol{P}_{1,n} \sim \mathcal{N}\left(\boldsymbol{P}_{1,n}; \boldsymbol{\nu}_n, \boldsymbol{\Omega}_n\right)$$

The temporal order of the variable $\boldsymbol{P}$ is represented in Fig. 3.2

Assignment Z denotes an hidden variable that assigns each observation $f_{t,k}$ at time t to sources. The number of assignment variables at time t corresponds to the number of observations in that time. Each assignment variable follows a multinomial distribution:

$$Z_{tk} \sim \mathcal{M}\left(Z_{tk}; N, \boldsymbol{\pi}_t\right).$$

Observations $\boldsymbol{f}_{t,k}$ Each observation at time t follows a normal distribution:

$$\boldsymbol{f}_{tk}|Z_{tk} = n, \boldsymbol{P}_t \sim \mathcal{N}\left(\boldsymbol{f}_{tk}; \boldsymbol{P}_{tn}, \boldsymbol{\Sigma}_{t,n}\right).$$

In the end, the set of all model's parameters, which will be estimated, is:

$$\theta = \underbrace{\{\boldsymbol{\nu}_n, \boldsymbol{\Omega}_n, \boldsymbol{\Lambda}_n\}_{n=1}^{N}}_{\theta_{\boldsymbol{P}}} \cup \underbrace{\{\boldsymbol{\pi}_t\}_{t=1}^{T}}_{\theta_{\boldsymbol{\pi}}} \cup \underbrace{\{\boldsymbol{\Sigma}_{tn}\}_{t,n=1}^{T,N}}_{\theta_{\boldsymbol{f}}}$$

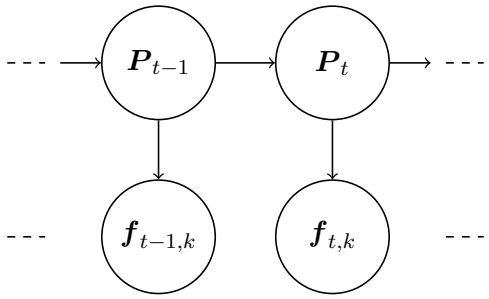


Figure 3.2: The temporally linked sources based on positions.

3.1.1 The Expectation-Maximization algorithm

Starting from the above described graphical model, we can derive the well-known iterative Expectation-Maximization algorithm in order to solve the problem.

The Q function First of all, we write the expected complete-data log-likelihood:

$$\mathcal{Q}\left(\theta, \theta^{(r)}\right) = \mathrm{E}_{p\left(Z, \boldsymbol{P} | \boldsymbol{f}, \theta^{(r)}\right)} \left\{\ln p\left(Z, \boldsymbol{P}, \boldsymbol{f} | \theta\right)\right\}. \tag{3.1}$$

where we compute $p\left(Z, \boldsymbol{P} | \boldsymbol{f}, \theta^{(r)}\right)$ during the E step, and during the M step, instead, we compute:

$$\theta^{(r+1)} = \arg\max_{\theta} \mathcal{Q}\left(\theta, \theta^{(r)}\right).$$

Since the EM algorithm is iterative, $\theta^{(r)}$ corresponds to the model's parameters previously estimated.

3.1.1.1 Expectation step

The Variational Approach We can notice that the E step of this graphical model has not a computationally tractable closed-form solution. Indeed, the assignment variable makes the E step combinatorial, and therefore computationally too expensive. To overcome the problem, we use the variational approach with the following factorization:

$$q\left(Z, \boldsymbol{P}\right) \propto \prod_{t,k=1}^{T,K_t} q\left(Z_{t,k}\right) \prod_{n=1}^{N} q\left(\boldsymbol{P}_n\right) \tag{3.2}$$

by using the exponential family as approximation:

$$q\left(Z_{tk}\right) = \exp\left(\mathrm{E}_{q\left(\boldsymbol{P}, Z \setminus Z_{t,k}\right)} \left\{\ln p\left(\boldsymbol{f}_{t,k}, Z_{t,k}, \boldsymbol{P}_t | \theta^{(r)}\right)\right\}\right), \quad \forall t, k, \tag{3.3}$$

$$q\left(\boldsymbol{P}_n\right) = \exp\left(\mathrm{E}_{q(\boldsymbol{P}_{1:n:N}) q(Z)} \left\{\ln p\left(\boldsymbol{f}, Z, \boldsymbol{P} | \theta^{(r)}\right)\right\}\right), \quad \forall n. \tag{3.4}$$

From here, some formulas and derivations are reported to explain the algorithm in detail. We do not enter in the complete derivation of each of them, but they can be found in appendix A.

E-Z step We analyse the two terms of the factorization. The first one, $q\left(Z_{t,k}\right)$, will be denote as E-Z step and its computation leads to the following result:

$$\begin{aligned}\eta_{tkn}^{(r+1)} \propto \quad & \pi_{tn} \exp\left(-\ln\left|\boldsymbol{\Sigma}_{tn}^{(r)}\right|^{\frac{1}{2}} - \frac{1}{2}\mathrm{tr}\left(\left(\boldsymbol{\Sigma}_{tn}^{(r)}\right)^{-1}\left(\boldsymbol{\Psi}_{tn}^{(r)} + \boldsymbol{\mu}_{tn}^{(r)}\left(\boldsymbol{\mu}_{tn}^{(r)}\right)^{\top}\right)\right)\right. \\ & \left. -\frac{1}{2}\left(\boldsymbol{f}_{tk} - 2\boldsymbol{\mu}_{tn}^{(r)}\right)^{\top}\left(\boldsymbol{\Sigma}_{tn}^{(r)}\right)^{-1}\boldsymbol{f}_{tk}\right). \end{aligned} \tag{3.5}$$

E-S step The computation of the second term, $q(\boldsymbol{P}_n)$, and denoted as E-S step, provides the following result:

$$\begin{aligned} q\left(\boldsymbol{P}_n\right) \propto \quad & \mathcal{N}\left(\boldsymbol{P}_{1n}; \boldsymbol{\nu}_n^{(r)}, \boldsymbol{\Omega}_n^{(r)}\right)\prod_{t=2}^{T}\mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{P}_{t-1n}, \boldsymbol{\Lambda}_n^{(r)}\right) \\ & \prod_{t=1}^{T}\mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{\mu}_{tn}^{\iota(r+1)}, \boldsymbol{\Psi}_{tn}^{\iota(r+1)}\right). \end{aligned} \tag{3.6}$$

So, we can state that $q(\boldsymbol{P}_n)$ is a multivariate normal. Therefore, we want to compute $q(\boldsymbol{P}_{tn})$ due to the temporal order of the variable. However, it is not obvious to extract it from the above result. Direct marginalization will naturally lead to forward-backward recursions. Indeed, the marginalization of (3.6) can be written as a product of two marginalizations representing the *past* and the *future* with respect to $q(\boldsymbol{P}_n)$:

$$\begin{aligned} q(\boldsymbol{P}_{tn}) \propto & \int \mathcal{N}\left(\boldsymbol{P}_{1n}; \boldsymbol{\nu}_n^{(r)}, \boldsymbol{\Omega}_n^{(r)}\right)\prod_{s=2}^{t}\mathcal{N}\left(\boldsymbol{P}_{sn}; \boldsymbol{P}_{s-1n}, \boldsymbol{\Lambda}_n^{(r)}\right) \\ & \prod_{s=1}^{t}\mathcal{N}\left(\boldsymbol{P}_{sn}; \boldsymbol{\mu}_{sn}^{\iota(r+1)}, \boldsymbol{\Psi}_{sn}^{\iota(r+1)}\right)\mathrm{d}\boldsymbol{P}_{1:t-1n} \qquad \times \end{aligned} \tag{3.7}$$

$$\begin{aligned} & \int \prod_{s=t+1}^{T}\mathcal{N}\left(\boldsymbol{P}_{sn}; \boldsymbol{P}_{s-1n}, \boldsymbol{\Lambda}_n^{(r)}\right) \\ & \prod_{s=t+1}^{T}\mathcal{N}\left(\boldsymbol{P}_{sn}; \boldsymbol{\mu}_{sn}^{\iota(r+1)}, \boldsymbol{\Psi}_{sn}^{\iota(r+1)}\right)\mathrm{d}\boldsymbol{P}_{t+1:Tn} \end{aligned} \tag{3.8}$$

The quantity on (3.7) will be denoted $\phi\left(\boldsymbol{P}_{tn}\right)$ and the one on (3.8), $\beta\left(\boldsymbol{P}_{tn}\right)$. With this notation we can get forward-backward recursions for ϕ and β:

$$\phi\left(\boldsymbol{P}_{tn}\right) = \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{\mu}_{tn}^{\iota(r+1)}, \boldsymbol{\Psi}_{tn}^{\iota(r+1)}\right) \int \phi\left(\boldsymbol{P}_{t-1n}\right) \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{P}_{t-1n}, \boldsymbol{\Lambda}_{n}^{(r)}\right) \mathrm{d}\boldsymbol{P}_{t-1n} \tag{3.9}$$

$$\beta\left(\boldsymbol{P}_{tn}\right) = \int \beta\left(\boldsymbol{P}_{t+1n}\right) \mathcal{N}\left(\boldsymbol{P}_{t+1n}; \boldsymbol{P}_{tn}, \boldsymbol{\Lambda}_{n}^{(r)}\right) \mathcal{N}\left(\boldsymbol{P}_{t+1n}; \boldsymbol{\mu}_{t+1n}^{\iota(r+1)}, \boldsymbol{\Psi}_{t+1n}^{\iota(r+1)}\right) \mathrm{d}\boldsymbol{P}_{t+1n} \tag{3.10}$$

By combining the forward and backward recursions we get:

$$\begin{aligned} q\left(\boldsymbol{P}_{tn}\right) &\propto \phi\left(\boldsymbol{P}_{tn}\right) \beta\left(\boldsymbol{P}_{tn}\right) \\ &\propto \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{\mu}_{tn}^{\phi(r+1)}, \boldsymbol{\Psi}_{tn}^{\phi(r+1)}\right) \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{\mu}_{tn}^{\beta(r+1)}, \boldsymbol{\Psi}_{tn}^{\beta(r+1)}\right) \end{aligned}$$

which leads to:

$$\boxed{q\left(\boldsymbol{P}_{tn}\right) = \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{\mu}_{tn}^{(r+1)}, \boldsymbol{\Psi}_{tn}^{(r+1)}\right)} \tag{3.11}$$

with

$$\left(\boldsymbol{\Psi}_{tn}^{(r+1)}\right)^{-1} = \left(\boldsymbol{\Psi}_{tn}^{\beta(r+1)}\right)^{-1} + \left(\boldsymbol{\Psi}_{tn}^{\phi(r+1)}\right)^{-1} \tag{3.12}$$

$$\boldsymbol{\mu}_{tn}^{(r+1)} = \boldsymbol{\Psi}_{tn}^{(r+1)} \left(\left(\boldsymbol{\Psi}_{tn}^{\beta(r+1)}\right)^{-1} \boldsymbol{\mu}_{tn}^{\beta(r+1)} + \left(\boldsymbol{\Psi}_{tn}^{\phi(r+1)}\right)^{-1} \boldsymbol{\mu}_{tn}^{\phi(r+1)} \right) \tag{3.13}$$

The coupled joint distribution When dealing with the M step, we will need the distribution $q\left(\boldsymbol{P}_{tn}, \boldsymbol{P}_{t-1n}\right)$ for the estimation of $\boldsymbol{\Lambda}_n$. If we marginalize (3.6) with respect to all variables except $\boldsymbol{P}_{tn}$ and $\boldsymbol{P}_{t-1n}$, we obtain the well-known formula:

$$\begin{aligned} q\left(\boldsymbol{P}_{tn}, \boldsymbol{P}_{t-1n}\right) &\propto \phi\left(\boldsymbol{P}_{t-1n}\right) \iota\left(\boldsymbol{P}_{tn}\right) T\left(\boldsymbol{P}_{tn}, \boldsymbol{P}_{t-1n}\right) \beta\left(\boldsymbol{P}_{tn}\right) \\ &\propto \mathcal{N}\left(\boldsymbol{P}_{t-1n}; \boldsymbol{\mu}_{t-1n}^{\phi(r+1)}, \boldsymbol{\Psi}_{t-1n}^{\phi(r+1)}\right) \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{\mu}_{tn}^{\iota(r+1)}, \boldsymbol{\Psi}_{tn}^{\iota(r+1)}\right) \\ &\quad \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{P}_{t-1n}, \boldsymbol{\Lambda}_{n}^{(r)}\right) \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{\mu}_{tn}^{\beta(r+1)}, \boldsymbol{\Psi}_{tn}^{\beta(r+1)}\right) \\ &\propto \mathcal{N}\left(\boldsymbol{P}_{tn}, \boldsymbol{P}_{t-1n}; \boldsymbol{\mu}_{tn}^{\zeta(r+1)}, \boldsymbol{\Psi}_{tn}^{\zeta(r+1)}\right) \end{aligned} \tag{3.14}$$

with

$$\boldsymbol{\Psi}_{tn}^{\zeta(r+1)} = \begin{pmatrix} \left(\boldsymbol{\Psi}_{tn}^{\iota(r+1)}\right)^{-1} + \left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1} + \left(\boldsymbol{\Psi}_{tn}^{\beta(r+1)}\right)^{-1} & -\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1} \\ -\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1} & \left(\boldsymbol{\Psi}_{t-1n}^{\phi(r)}\right)^{-1} + \left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1} \end{pmatrix}^{-1} \tag{3.15}$$

$$\boldsymbol{\mu}_{tn}^{\zeta(r+1)} = \boldsymbol{\Psi}_{tn}^{\zeta(r+1)} \begin{pmatrix} \left(\boldsymbol{\Psi}_{tn}^{\beta(r+1)}\right)^{-1} \boldsymbol{\mu}_{tn}^{\beta(r+1)} + \left(\boldsymbol{\Psi}_{tn}^{\iota(r+1)}\right)^{-1} \boldsymbol{\mu}_{tn}^{\iota(r+1)} \\ \left(\boldsymbol{\Psi}_{t-1n}^{\phi(r+1)}\right)^{-1} \boldsymbol{\mu}_{t-1n}^{\phi(r+1)} \end{pmatrix} \quad (3.16)$$

3.1.1.2 Maximization step

In the Maximization step, we have to evaluate the new model parameters from 3.1:

$$\theta^{(r+1)} = \arg\max_{\theta} \mathcal{Q}\left(\theta, \theta^{(r)}\right).$$

Again, all derivations can be found in appendix A as in the previous step.

If we develop 3.1, we obtain

$$\begin{aligned} \mathcal{Q}\left(\theta, \theta^{(r)}\right) = & \sum_{t,k=1}^{T,K_t} \mathrm{E}_{q(Z_{tk})} \left\{\ln p\left(Z_{tk}|\theta\right)\right\} + \mathrm{E}_{q(Z_{tk})q(\boldsymbol{P}_t)} \left\{\ln p\left(\boldsymbol{f}_{tk}|Z_{tk}, \boldsymbol{P}_t, \theta\right)\right\} \\ & + \sum_{n=1}^{N} \left[\sum_{t=2}^{T} \mathrm{E}_{q(\boldsymbol{P}_{tn}, \boldsymbol{P}_{t-1n})} \left\{\ln p\left(\boldsymbol{P}_{t-1n}|\boldsymbol{P}_{tn}, \theta\right)\right\} + \mathrm{E}_{q(\boldsymbol{P}_{1n})} \left\{\ln p\left(\boldsymbol{P}_{1n}|\theta\right)\right\} \right] \end{aligned}$$

As we can see in the above formula, there are four terms. Each of them depends on different parameters and as we know when taking the derivative with respect to one of parameters all others become zero, we can skip them.

The parameter π

$$\pi_{tn} = \frac{\sum_{k=1}^{K_t} \eta_{tkn}^{Z(r+1)}}{K_t} \quad (3.17)$$

The parameter $\boldsymbol{\Sigma}$

$$\boldsymbol{\Sigma}_{tn}^{(r+1)} = \boldsymbol{\Psi}_{tn} + \frac{\sum_{k=1}^{K_t} \eta_{ktn} \left(\boldsymbol{\mu}_{tn} - \boldsymbol{f}_{tk}\right) \left(\boldsymbol{\mu}_{tn} - \boldsymbol{f}_{tk}\right)^{\top}}{\sum_{k=1}^{K_t} \eta_{ktn}} \quad (3.18)$$

The parameters $\boldsymbol{\Lambda}$

$$\boldsymbol{\Lambda}_{n}^{(r+1)} = \frac{1}{T-1} \sum_{t=2}^{T} \mathbf{E}_{tn}^{(r+1)}. \quad (3.19)$$

where $\mathbf{E}_{tn}^{(r+1)}$ is the energy matrix, which comes from the coupled joint distribution $q\left(\boldsymbol{P}_{tn}, \boldsymbol{P}_{t-1n}\right)$ and has the following expression:

$$\begin{aligned}\mathbf{E}_{tn}^{(r+1)} = & \left(\boldsymbol{\Psi}_{tn}^{\zeta(r+1)}\right)_{11} + \left(\boldsymbol{\mu}_{tn}^{\zeta(r+1)}\right)_1 \left(\boldsymbol{\mu}_{tn}^{\zeta(r+1)}\right)_1^\top + \left(\boldsymbol{\Psi}_{tn}^{\zeta(r+1)}\right)_{22} \\ & + \left(\boldsymbol{\mu}_{tn}^{\zeta(r+1)}\right)_2 \left(\boldsymbol{\mu}_{tn}^{\zeta(r+1)}\right)_2^\top - \left(\boldsymbol{\Psi}_{tn}^{\zeta(r+1)}\right)_{12} - \left(\boldsymbol{\mu}_{tn}^{\zeta(r+1)}\right)_1 \left(\boldsymbol{\mu}_{tn}^{\zeta(r+1)}\right)_2^\top \\ & - \left(\boldsymbol{\Psi}_{tn}^{\zeta(r+1)}\right)_{21} - \left(\boldsymbol{\mu}_{tn}^{\zeta(r+1)}\right)_2 \left(\boldsymbol{\mu}_{tn}^{\zeta(r+1)}\right)_1^\top\end{aligned}$$

The parameter $\boldsymbol{\nu}$

$$\boldsymbol{\nu}_n = \boldsymbol{\mu}_{1n}^{(r+1)} \tag{3.20}$$

The parameter $\boldsymbol{\Omega}$

$$\boldsymbol{\Omega}_n = \boldsymbol{\Psi}_{1n}^{(r+1)} \tag{3.21}$$

After having dealt with the initial graphical model, we now want to make some extensions, such as solve the outliers/clutter problem, and add velocity for smoothness of trajectories.

3.2 Clutter Model

In this section, we present the adding of clutter to make the model more realistic. Clutter is everything that looks like observations, but it is not generated by an object. So, it refers to all spurious observations and we assumed that it follows an uniform distribution.

Observations Differently from the previous assumption, now:

$$\boldsymbol{F}_{tk}|Z_{tk}=n,\boldsymbol{P}_t\sim\begin{cases}\mathcal{N}\left(\boldsymbol{F}_{tk};\boldsymbol{P}_{tn},\boldsymbol{\Sigma}_n\right) & \text{if } n\leq N\\ \mathcal{U}\left(\boldsymbol{F}_{tk};A\right) & \text{if } n=N+1\end{cases}$$

where A is the area taken into account, with width and height fixed.

Assignment variables In this case clutter is consider as generated from another source and Z_{tk} still follows a multinomial:

$$Z_{tk}\sim\mathcal{M}\left(Z_{tk};N+1,\boldsymbol{\pi}_t\right).$$

As far as the Expectation-Maximization algorithm is concerned, the derived formulas remain almost the same, except for the E-Z step and the π parameter in the M step. All others are indirectly affected from the previous ones.

E-Z step

$$\eta_{tkn}^{(r+1)}\propto\begin{cases}\pi_{tn}\exp\left[-\ln\left|\boldsymbol{\Sigma}_{tn}^{(r)}\right|^{\frac{1}{2}}-\frac{1}{2}\text{tr}\left(\left(\boldsymbol{\Sigma}_{tn}^{(r)}\right)^{-1}\left(\boldsymbol{\Psi}_{tn}^{(r)}+\boldsymbol{\mu}_{tn}^{(r)}\left(\boldsymbol{\mu}_{tn}^{(r)}\right)^{\top}\right)\right)-\right.\\ \left.\quad\frac{1}{2}\left(\boldsymbol{f}_{tk}-2\boldsymbol{\mu}_{tn}^{(r)}\right)^{\top}\left(\boldsymbol{\Sigma}_{tn}^{(r)}\right)^{-1}\boldsymbol{f}_{tk}\right] & \text{if } n\leq N\\ \pi_{tn}\frac{1}{|A|} & \text{if } n=N+1\end{cases}\tag{3.22}$$

where $|A|$ is the area taken into account. In the the Maximization step, the expected complete-data log likelihood function and factorization done are not affected by the adding of the clutter, except for the parameter π which now takes into account the extra class. However, the final formula of the parameter is the same.

3.3 Modelling Objects' Velocities

In tracking it is common to take velocity into account for the state in order to have a more accurate estimation of the trajectories. Here we extend the model with a further hidden variable related to the velocity as shown in Fig. 3.3.

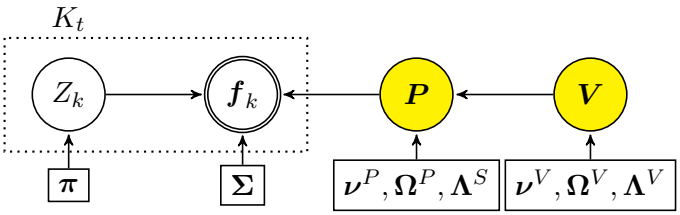


Figure 3.3: The extended graphical model to solve with the velocity hidden variable.

In this case, source positions at each time step do not depend only on the previous positions, but now also on the velocity in the current time, as reported in Fig. 3.4.

Sources $\boldsymbol{P}_{tn}$ denotes the position of the n^{th} source at time t. Each source is supposed to follow a zero-order Gaussian model as in the initial graphical model. However, the mean takes current velocity into account as well, here. Moreover, $\boldsymbol{V}_{tn}$ denotes the velocity of the n^{th} source at time t and again, each source is supposed to follow a first-order Gaussian model.

$$\boldsymbol{P}_{tn}|\boldsymbol{P}_{t-1n} \sim \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{P}_{t-1n} + \boldsymbol{V}_{tn}, \boldsymbol{\Lambda}_n^S\right)$$
$$\boldsymbol{V}_{tn}|\boldsymbol{V}_{t-1n} \sim \mathcal{N}\left(\boldsymbol{V}_{tn}; \boldsymbol{V}_{t-1n}, \boldsymbol{\Lambda}_n^V\right)$$

These two formulas can be combined in order to deal with only one Gaussian:

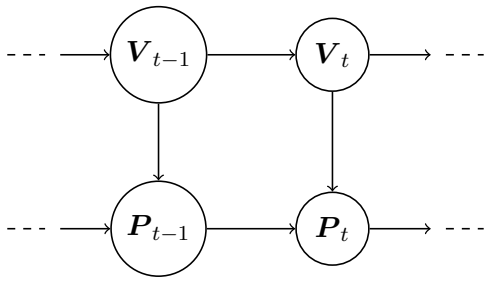


Figure 3.4: The temporally linked sources based on positions and velocities.

$$\begin{pmatrix} \boldsymbol{P}_{tn} \\ \boldsymbol{V}_{tn} \end{pmatrix} \Bigg| \begin{pmatrix} \boldsymbol{P}_{t-1n} \\ \boldsymbol{V}_{t-1n} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{P}_{tn} \\ \boldsymbol{V}_{tn} \end{pmatrix}; \mathbf{A} \begin{pmatrix} \boldsymbol{P}_{t-1n} \\ \boldsymbol{V}_{t-1n} \end{pmatrix}, \mathbf{\Lambda} \right)$$

with

$$\mathbf{\Lambda}^{-1} = \begin{pmatrix} \mathbf{\Lambda}_P^{-1} & -\mathbf{\Lambda}_P^{-1} \\ -\mathbf{\Lambda}_P^{-1} & \mathbf{\Lambda}_P^{-1} + \mathbf{\Lambda}_V^{-1} \end{pmatrix} \qquad \text{and} \qquad \mathbf{A} = \mathbf{\Lambda} \begin{pmatrix} \mathbf{\Lambda}_P^{-1} & 0 \\ -\mathbf{\Lambda}_P^{-1} & \mathbf{\Lambda}_V^{-1} \end{pmatrix}$$

In appendix D, we proof that $\mathbf{\Lambda}$ is invertible and this let us show:

$$\mathbf{A} = \begin{pmatrix} \mathbf{\Lambda}_P + \mathbf{\Lambda}_V & \mathbf{\Lambda}_V \\ \mathbf{\Lambda}_V & \mathbf{\Lambda}_V \end{pmatrix} \begin{pmatrix} \mathbf{\Lambda}_P^{-1} & 0 \\ -\mathbf{\Lambda}_P^{-1} & \mathbf{\Lambda}_V^{-1} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{I} \\ 0 & \mathbf{I} \end{pmatrix}$$

So, we can notice that $\mathbf{A}$ is fixed and we do not need to estimate it.

In all the set of parameters is:

$$\theta = \underbrace{\left\{\boldsymbol{\nu}_n^V, \mathbf{\Omega}_n^V, \mathbf{\Lambda}_n^V\right\}_{n=1}^N}_{\theta_{\boldsymbol{V}}} \cup \underbrace{\left\{\boldsymbol{\nu}_n^P, \mathbf{\Omega}_n^P, \mathbf{\Lambda}_n^P\right\}_{n=1}^N}_{\theta_{\boldsymbol{P}}} \cup \underbrace{\left\{\boldsymbol{\pi}_t\right\}_{t=1}^T}_{\theta_{\boldsymbol{\pi}}} \cup \underbrace{\left\{\mathbf{\Sigma}_{tn}\right\}_{t,n=1}^{T,N}}_{\theta_{\boldsymbol{f}}}$$

3.3.1 EM Algorithm Revisited

The $\mathcal{Q}$ function First of all, we write the expected complete-data log-likelihood:

$$\mathcal{Q}\left(\theta, \theta^{(r)}\right) = \mathrm{E}_{p\left(Z, \boldsymbol{P}, \boldsymbol{V} | \boldsymbol{f}, \theta^{(r)}\right)} \left\{\ln p\left(Z, \boldsymbol{P}, \boldsymbol{V}, \boldsymbol{f} | \theta\right)\right\}. \tag{3.23}$$

where we compute $p\left(Z, \boldsymbol{P}, \boldsymbol{V} | \boldsymbol{f}, \theta^{(r)}\right)$ during the E step, and during the M step,instead, we compute:

$$\theta^{(r+1)} = \arg\max_{\theta} \mathcal{Q}\left(\theta, \theta^{(r)}\right).$$

The Variational Approach Again, as we saw previously, we can notice that the E step of this graphical model has not a computationally tractable closed-form solution and we can use the variational approach to overcome the problem. Now, the factorization is:

$$q\left(Z, \boldsymbol{P}, \boldsymbol{V}\right) = \prod_{t,k=1}^{T,K_t} q\left(Z_{tk}\right) \prod_{n=1}^{N} q\left(\boldsymbol{P}_n, \boldsymbol{V}_n\right) \tag{3.24}$$

by using the exponential family as approximation:

$$q\left(Z_{tk}\right) \propto \exp\left(\mathrm{E}_{q(\boldsymbol{P}_t, \boldsymbol{V}_t)}\left\{\ln p\left(\boldsymbol{f}_{tk}, Z_{tk}, \boldsymbol{P}_t, \boldsymbol{V}_t | \theta^{(r)}\right)\right\}\right), \quad \forall t, k, \tag{3.25}$$

$$q\left(\boldsymbol{P}_n, \boldsymbol{V}_n\right) \propto \exp\left(\mathrm{E}_{q(\boldsymbol{P}_{1:n:N}, \boldsymbol{V}_{1:n:N})q(Z)}\left\{\ln p\left(\boldsymbol{f}, Z, \boldsymbol{P}, \boldsymbol{V} | \theta^{(r)}\right)\right\}\right), \quad \forall n. \tag{3.26}$$

3.3.1.1 Expectation Step

E-Z step As far as concerned the E-Z step with respect to the first term in the variational, the probability inside the logarithm does not depend on $\boldsymbol{V}_n$ and so the result is the same as we saw in the beginning 3.5.

E-S step The derivation of all formulas in this phase follows the same way we did for the model without velocity. So, for the details, see appendix. Here, we report what is worthy. First of all, we can write the second term in the variational in the following way:

$$\begin{aligned} q\left(\boldsymbol{P}_n, \boldsymbol{V}_n\right) \propto \quad & p\left(\boldsymbol{V}_n | \theta^{(r)}\right) p\left(\boldsymbol{P}_n | \boldsymbol{V}_n, \theta^{(r)}\right) \times \\ & \exp\left(\mathrm{E}_{q(\boldsymbol{P}_{1:n:N}, \boldsymbol{V}_{1:n:N})q(Z)}\left\{\ln p\left(\boldsymbol{f} | Z, \boldsymbol{P}, \theta^{(r)}\right)\right\}\right) \end{aligned} \tag{3.27}$$

As regards the first two probabilities in (3.27), we can make some considerations:

$$p\left(\boldsymbol{V}_n|\theta^{(r)}\right) \sim \mathcal{N}\left(\boldsymbol{V}_{1n};\boldsymbol{\nu}_n^{V^{(r)}},\boldsymbol{\Omega}_n^{V^{(r)}}\right)\prod_{t=2}^{T}\mathcal{N}\left(\boldsymbol{V}_{tn};\boldsymbol{V}_{t-1n},\boldsymbol{\Lambda}_n^{V^{(r)}}\right)$$

$$p\left(\boldsymbol{P}_n|\boldsymbol{V}_n,\theta^{(r)}\right) \sim \mathcal{N}\left(\boldsymbol{P}_{1n};\boldsymbol{\nu}_n^{P^{(r)}}+\boldsymbol{V}_{1n},\boldsymbol{\Omega}_n^{P^{(r)}}\right)\prod_{t=2}^{T}\mathcal{N}\left(\boldsymbol{P}_{tn};\boldsymbol{P}_{t-1n}+\boldsymbol{V}_{tn},\boldsymbol{\Lambda}_n^{P^{(r)}}\right)$$

$$\sim \mathcal{N}\left(\boldsymbol{P}_{1n};\boldsymbol{\nu}_n^{P^{(r)}},\boldsymbol{\Omega}_n^{P^{(r)}}\right)\prod_{t=2}^{T}\mathcal{N}\left(\boldsymbol{P}_{tn};\boldsymbol{P}_{t-1n}+\boldsymbol{V}_{tn},\boldsymbol{\Lambda}_n^{P^{(r)}}\right)$$

We can consider initialization independent for both positions and velocities, and this is the reason why we ignore the dependence on $\boldsymbol{V}$ for the first time step in the above formula. Besides, we can combine for convenience the two above probabilities in the following one:

$$\begin{aligned} p\left(\boldsymbol{P}_n,\boldsymbol{V}_n|\theta^{(r)}\right) \sim \quad & \mathcal{N}\left(\begin{pmatrix}\boldsymbol{P}_{1n}\\ \boldsymbol{V}_{1n}\end{pmatrix};\boldsymbol{\nu}_n^{(r)},\boldsymbol{\Omega}_n^{(r)}\right) \\ & \prod_{t=2}^{T}\mathcal{N}\left(\begin{pmatrix}\boldsymbol{P}_{tn}\\ \boldsymbol{V}_{tn}\end{pmatrix};\mathbf{A}\begin{pmatrix}\boldsymbol{P}_{t-1n}\\ \boldsymbol{V}_{t-1n}\end{pmatrix},\boldsymbol{\Lambda}\right) \end{aligned} \tag{3.28}$$

with

$$\boldsymbol{\nu}_n^{(r)} = \begin{pmatrix}\boldsymbol{\nu}_n^{P^{(r)}}\\ \boldsymbol{\nu}_n^{V^{(r)}}\end{pmatrix} \qquad \text{and} \qquad \boldsymbol{\Omega}_n^{(r)} = \begin{pmatrix}\boldsymbol{\Omega}_n^{P^{(r)}} & 0\\ 0 & \boldsymbol{\Omega}_n^{V^{(r)}}\end{pmatrix}$$

From here, we denote $\begin{pmatrix}\boldsymbol{P}_{tn}\\ \boldsymbol{V}_{tn}\end{pmatrix}$ with $\boldsymbol{S}_{tn}$.

So, similarly to the initial model:

$$\begin{aligned} q\left(\boldsymbol{S}_n\right) \quad \propto \quad & \mathcal{N}\left(\boldsymbol{S}_{1,n};\boldsymbol{\nu}_n^{(r)},\boldsymbol{\Omega}_n^{(r)}\right)\prod_{t=2}^{T}\mathcal{N}\left(\boldsymbol{S}_{t,n};\mathbf{A}\boldsymbol{S}_{t-1,n},\boldsymbol{\Lambda}\right) * \\ & * \prod_{t=1}^{T}\mathcal{N}\left(\boldsymbol{S}_{t,n};\boldsymbol{\mu}_{t,n}^{\iota(r+1)},\boldsymbol{\Psi}_{t,n}^{\iota(r+1)}\right) \end{aligned} \tag{3.29}$$

with

$$\left(\boldsymbol{\Psi}_{t,n}^{\iota(r+1)}\right)^{-1} = \left(\begin{array}{c|c} \frac{\left(\sum_{k=1}^{K_t}\eta_{t,k,n}^{(r+1)}\right)\left(\boldsymbol{\Sigma}_{t,n}^{(r)}\right)^{-1}}{0} & \begin{array}{c}0\\ \hline 0\end{array}\end{array}\right)$$

$$\boldsymbol{\mu}_{t,n}^{\iota(r+1)} = \left(\begin{array}{c} \frac{\sum_{k=1}^{K_t}\eta_{t,k,n}^{(r+1)}\boldsymbol{f}_{t,k}}{\sum_{k=1}^{K_t}\eta_{t,k,n}^{(r+1)}} \\ \hline ? \end{array}\right).$$

Since we have no knowledge about velocity, we want zero value in the precision matrix and this results in infinity values into $\boldsymbol{\Psi}_{t,n}^{\iota(r+1)}$, which is a block matrix (see [Petersen and Pedersen, 2012]). Due to zero values in precision matrix, we can assign every values to the block referring to velocity in $\boldsymbol{\mu}_{t,n}^{\iota(r+1)}$. At the light of this result, we are certain that $q(\boldsymbol{S}_n)$ is a multivariate normal. Because of the temporal term, it is not obvious to extract $q(\boldsymbol{S}_{t,n})$ from here. Direct marginalization will naturally lead to forward-backward recursions. Indeed, the marginalization of 3.29 can be written as a product of two marginalizations representing the *past* and the *future* with respect to $\boldsymbol{S}_{t,n}$:

$$\begin{aligned} q\left(\boldsymbol{S}_n\right) \quad \propto \quad & \int \mathcal{N}\left(\boldsymbol{S}_{1,n}; \boldsymbol{\nu}_n^{(r)}, \boldsymbol{\Omega}_n^{(r)}\right) \prod_{s=2}^{t} \mathcal{N}\left(\boldsymbol{S}_{t,n}; \mathbf{A}\boldsymbol{S}_{s-1,n}, \boldsymbol{\Lambda}\right) * \\ & * \prod_{s=1}^{t} \mathcal{N}\left(\boldsymbol{S}_{t,n}; \boldsymbol{\mu}_{s,n}^{\iota(r+1)}, \boldsymbol{\Psi}_{s,n}^{\iota(r+1)}\right) \mathrm{d}\boldsymbol{S}_{1:t-1,n} * \qquad (3.30) \\ & * \int \prod_{s=t+1}^{T} \mathcal{N}\left(\boldsymbol{S}_{s,n}; \mathbf{A}\boldsymbol{S}_{s-1,n}, \boldsymbol{\Lambda}\right) * \\ & * \prod_{s=t+1}^{T} \mathcal{N}\left(\boldsymbol{S}_{s,n}; \boldsymbol{\mu}_{s,n}^{\iota(r+1)}, \boldsymbol{\Psi}_{s,n}^{\iota(r+1)}\right) \mathrm{d}\boldsymbol{S}_{t+1:t,n} \qquad (3.31) \end{aligned}$$

The quantity on (3.30) will be denoted $\phi\left(\boldsymbol{S}_{t,n}\right)$ and the one on (3.31), $\beta\left(\boldsymbol{S}_{t,n}\right)$.

With this notation we can get forward-backward recursions for ϕ and β:

$$\begin{aligned}\phi\left(\boldsymbol{S}_{t,n}\right) &= \mathcal{N}\left(\boldsymbol{S}_{t,n};\boldsymbol{\mu}_{t,n}^{\iota(r+1)},\boldsymbol{\Psi}_{t,n}^{\iota(r+1)}\right)\int\phi\left(\boldsymbol{S}_{t-1,n}\right)* \\ &\quad *\mathcal{N}\left(\boldsymbol{S}_{t,n};\mathbf{A}\boldsymbol{S}_{t-1,n},\boldsymbol{\Lambda}\right)\mathrm{d}\boldsymbol{S}_{t-1,n} \qquad (3.32)\\ \beta\left(\boldsymbol{S}_{t,n}\right) &= \int\beta\left(\boldsymbol{S}_{t+1,n}\right)\mathcal{N}\left(\boldsymbol{S}_{t+1,n};\mathbf{A}\boldsymbol{S}_{t,n},\boldsymbol{\Lambda}\right)* \\ &\quad *\mathcal{N}\left(\boldsymbol{S}_{t+1,n};\boldsymbol{\mu}_{t+1,n}^{\iota(r+1)},\boldsymbol{\Psi}_{t+1,n}^{\iota(r+1)}\right)\mathrm{d}\boldsymbol{S}_{t+1,n} \qquad (3.33)\end{aligned}$$

By combining the forward and backward recursions we get:

$$\begin{aligned}q\left(\boldsymbol{S}_{t,n}\right) &\propto \phi\left(\boldsymbol{S}_{t,n}\right)\beta\left(\boldsymbol{S}_{t,n}\right) \\ &\propto \mathcal{N}\left(\boldsymbol{S}_{t,n};\boldsymbol{\mu}_{t,n}^{\phi(r+1)},\boldsymbol{\Psi}_{t,n}^{\phi(r+1)}\right)\mathcal{N}\left(\boldsymbol{S}_{t,n};\boldsymbol{\mu}_{t,n}^{\beta(r+1)},\boldsymbol{\Psi}_{t,n}^{\beta(r+1)}\right)\end{aligned}$$

which leads to:

$$\boxed{q\left(\boldsymbol{S}_{t,n}\right)=\mathcal{N}\left(\boldsymbol{S}_{t,n};\boldsymbol{\mu}_{t,n}^{(r+1)},\boldsymbol{\Psi}_{t,n}^{(r+1)}\right)} \qquad (3.34)$$

with

$$\left(\boldsymbol{\Psi}_{t,n}^{(r+1)}\right)^{-1}=\left(\boldsymbol{\Psi}_{t,n}^{\beta(r+1)}\right)^{-1}+\left(\boldsymbol{\Psi}_{t,n}^{\phi(r+1)}\right)^{-1} \qquad (3.35)$$

$$\boldsymbol{\mu}_{t,n}^{(r+1)}=\boldsymbol{\Psi}_{t,n}^{(r+1)}\left(\left(\boldsymbol{\Psi}_{t,n}^{\beta(r+1)}\right)^{-1}\boldsymbol{\mu}_{t,n}^{\beta(r+1)}+\left(\boldsymbol{\Psi}_{t,n}^{\phi(r+1)}\right)^{-1}\boldsymbol{\mu}_{t,n}^{\phi(r+1)}\right) \qquad (3.36)$$

For the complete derivation of forward and backward recursions, see appendix C.

The coupled joint distribution When dealing with the M step, we will need the distribution $q\left(\boldsymbol{S}_{t,n},\boldsymbol{S}_{t-1,n}\right)$ for the estimation of $\boldsymbol{\Lambda}_n$. If we marginalize (3.29) with respect to all variables except $\boldsymbol{S}_{t,n}$ and $\boldsymbol{S}_{t-1,n}$, we obtain the well-known formula:

$$\begin{aligned}q\left(\boldsymbol{S}_{t,n},\boldsymbol{S}_{t-1,n}\right) &\propto \phi\left(\boldsymbol{S}_{t-1,n}\right)\iota\left(\boldsymbol{S}_{t,n}\right)T\left(\boldsymbol{S}_{t,n},\boldsymbol{S}_{t-1,n}\right)\beta\left(\boldsymbol{S}_{t,n}\right) \\ &\propto \mathcal{N}\left(\boldsymbol{S}_{t-1,n};\boldsymbol{\mu}_{t-1,n}^{\phi(r+1)},\boldsymbol{\Psi}_{t-1,n}^{\phi(r+1)}\right)\mathcal{N}\left(\boldsymbol{S}_{t,n};\boldsymbol{\mu}_{t,n}^{\iota(r+1)},\boldsymbol{\Psi}_{t,n}^{\iota(r+1)}\right)* \\ &\quad *\mathcal{N}\left(\boldsymbol{S}_{t,n};\mathbf{A}\boldsymbol{S}_{t-1,n},\boldsymbol{\Lambda}_n^{(r)}\right)\mathcal{N}\left(\boldsymbol{S}_{t,n};\boldsymbol{\mu}_{t,n}^{\beta(r+1)},\boldsymbol{\Psi}_{t,n}^{\beta(r+1)}\right) \qquad (3.37)\end{aligned}$$

where, the covariance is:

$$\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}=\begin{pmatrix}\left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{11} & \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{12}\\ \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{21} & \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{22}\end{pmatrix}^{-1} \qquad (3.38)$$

with

$$\begin{aligned}
\left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{11} &= \left(\boldsymbol{\Psi}_{t,n}^{\iota(r+1)}\right)^{-1} + \left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1} + \left(\boldsymbol{\Psi}_{t,n}^{\beta(r+1)}\right)^{-1} \\
\left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{12} &= -\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}\mathbf{A} \\
\left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{21} &= -\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}\mathbf{A} \\
\left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{22} &= \left(\boldsymbol{\Psi}_{t-1,n}^{\phi(r)}\right)^{-1} + \mathbf{A}^{\top}\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}\mathbf{A}
\end{aligned}$$

and the mean is:

$$\boldsymbol{\mu}_{t,n}^{\zeta(r+1)} = \boldsymbol{\Psi}_{t,n}^{\zeta(r+1)} \begin{pmatrix} \left(\boldsymbol{\Psi}_{t,n}^{\beta(r+1)}\right)^{-1}\boldsymbol{\mu}_{t,n}^{\beta(r+1)} + \left(\boldsymbol{\Psi}_{t,n}^{\iota(r+1)}\right)^{-1}\boldsymbol{\mu}_{t,n}^{\iota(r+1)} \\ \left(\boldsymbol{\Psi}_{t-1,n}^{\phi(r+1)}\right)^{-1}\boldsymbol{\mu}_{t-1,n}^{\phi(r+1)} \end{pmatrix} \tag{3.39}$$

Here, we make some considerations. The coupled joint distribution following a Gaussian distribution:

$$q\left(\boldsymbol{S}_{t,n}, \boldsymbol{S}_{t-1,n}\right) \propto \mathcal{N}\left(\begin{pmatrix} \boldsymbol{S}_{t,n} \\ \boldsymbol{S}_{t-1,n} \end{pmatrix}; \boldsymbol{\mu}_{t,n}^{\zeta(r+1)}, \boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right) \tag{3.40}$$

3.3.1.2 Maximization Step

In the M step, we have to evaluate the new model's parameters from the expected complete-data log-likelihood function:

$$\theta^{(r+1)} = \arg\max_{\theta} \mathcal{Q}\left(\theta, \theta^{(r)}\right).$$

Starting from the expected complete-data log-likelihood function and the factorization done in the previous step, we can write the first one in the following way:

$$\begin{aligned}
\mathcal{Q}\left(\theta, \theta^{(r)}\right) \quad = \quad & \sum_{t,k=1}^{T,K_t} \mathrm{E}_{q(Z_{tk})}\left\{\ln p\left(Z_{tk}|\theta\right)\right\} + \mathrm{E}_{q(Z_{tk})q(\boldsymbol{S}_t)}\left\{\ln p\left(\boldsymbol{f}_{tk}|Z_{tk}, \boldsymbol{S}_t, \theta\right)\right\} + \\
& + \sum_{n=1}^{N}\sum_{t=2}^{T} \mathrm{E}_{q(\boldsymbol{P}_{t,n}, \boldsymbol{P}_{t-1,n})}\left\{\ln p\left(\boldsymbol{P}_{t-1,n}|\boldsymbol{P}_{t,n}, \theta\right)\right\} + \\
& + \sum_{n=1}^{N} \mathrm{E}_{q(\boldsymbol{P}_{1,n})}\left\{\ln p\left(\boldsymbol{P}_{1,n}|\theta\right)\right\}
\end{aligned} \tag{3.41}$$

In 3.41 there is a linear combination of four terms in which each depends on only one different parameter: taking the derivative with respect to one parameter, only

one term is taken into account and the others are skipped, because they become zero. Since the first two terms do not depend on $\boldsymbol{S}$, their evaluation is totally equal to results found for the model with only positions. Neverthless, we report here their definitions and results, and we compute the remains parameters.

The parameter π

$$\pi_{t,n} = \frac{\sum_{k=1}^{K_t} \eta_{t,k,n}^{Z(r+1)}}{K_t} \tag{3.42}$$

The parameter $\boldsymbol{\Sigma}$

$$\boldsymbol{\Sigma}_{t,n}^{(r+1)} = \boldsymbol{\Psi}_{t,n} + \frac{\sum_{k=1}^{K_t} \eta_{t,k,n} \left(\boldsymbol{\mu}_{t,n} - \boldsymbol{f}_{t,k}\right) \left(\boldsymbol{\mu}_{t,n} - \boldsymbol{f}_{t,k}\right)^{\top}}{\sum_{k=1}^{K_t} \eta_{t,k,n}} \tag{3.43}$$

The parameters $\boldsymbol{\Lambda}_n^P$ and $\boldsymbol{\Lambda}_n^V$

$$\boldsymbol{\Lambda}_n^{P^{(r+1)}} = \frac{1}{T-1} \sum_{t=2}^{T} \mathbf{E}_{t,n}^{P^{(r+1)}} \tag{3.44}$$

$$\boldsymbol{\Lambda}_n^{V^{(r+1)}} = \frac{1}{T-1} \sum_{t=2}^{T} \mathbf{E}_{t,n}^{V^{(r+1)}} \tag{3.45}$$

where $\mathbf{E}_{t,n}^{S^{(r+1)}}$ and $\mathbf{E}_{t,n}^{V^{(r+1)}}$ being the energy matrices with the following expression:

$$
\begin{aligned}
\mathbf{E}_{t,n}^{P^{(r+1)}} =& \mathbf{Q}_{t,n}^{11} + \mathbf{Q}_{t,n}^{33} + \mathbf{Q}_{t,n}^{44} - 2\mathbf{Q}_{t,n}^{13} - 2\mathbf{Q}_{t,n}^{14} + 2\mathbf{Q}_{t,n}^{34} - 2\mathbf{Q}_{t,n}^{12} + 2\mathbf{Q}_{t,n}^{14} + \\
&+ 2\mathbf{Q}_{t,n}^{23} - 2\mathbf{Q}_{t,n}^{34} + 2\mathbf{Q}_{t,n}^{24} - 2\mathbf{Q}_{t,n}^{44} + \mathbf{Q}_{t,n}^{44} + \mathbf{Q}_{t,n}^{22} - 2\mathbf{Q}_{t,n}^{24} = \\
=& \mathbf{Q}_{t,n}^{11} + \mathbf{Q}_{t,n}^{33} + \mathbf{Q}_{t,n}^{22} - 2\mathbf{Q}_{t,n}^{13} - 2\mathbf{Q}_{t,n}^{12} + 2\mathbf{Q}_{t,n}^{23} = \\
=& \left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{11} + \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_1 \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_1^\top + \left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{22} + \\
&+ \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_2 \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_2^\top + \left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{33} + \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_3 \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_3^\top - \\
&- \left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{12} - \left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{21} - \left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{13} - \left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{31} - \\
&- \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_1 \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_2^\top - \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_2 \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_1^\top - \\
&- \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_3 \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_1^\top - \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_1 \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_3^\top + \\
&+ \left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{23} + \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_2 \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_3^\top + \left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{32} + \\
&+ \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_3 \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_2^\top
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{E}_{t,n}^{V^{(r+1)}} =& \left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{22} + \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_2 \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_2^\top + \left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{44} + \\
&+ \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_4 \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_4^\top - \left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{14} - \left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{41} - \\
&- \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_1 \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_4^\top - \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_4 \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_1^\top
\end{aligned}
$$

where the subscribes here denote the quarter in the correspondant dimension.

The parameter $\boldsymbol{\nu}_n^P$

$$
\boldsymbol{\nu}_n^P = \left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_1 \tag{3.46}
$$

The parameter $\mathbf{\Omega}_n^P$

$$
\mathbf{\Omega}_n^P = \left(\mathbf{\Psi}_{1,n}^{(r+1)}\right)_{11} \tag{3.47}
$$

The parameter $\boldsymbol{\nu}_n^V$

$$
\boldsymbol{\nu}_n^V = \left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_2 \tag{3.48}
$$

The parameter $\mathbf{\Omega}_n^V$

$$\mathbf{\Omega}_n^V = \left(\mathbf{\Psi}_{1,n}^{(r+1)}\right)_{22} \tag{3.49}$$

3.4 Initialization

As it is well-known, an EM algorithm requires an initialization. Previously, we described the method and we derived all formulations regarding the Expectation-Maximization algorithm. In this case, due to the variational approach, the expectation step is split into two parts, making so the initialization hard to deal with. Having a look at derivations, each step needs parameters coming from the other two and this means we have to initialize at least the parameters of two step out of three before starting iterations of the algorithm. A guess of model's parameters (output of the maximization step) is not enough. Before analysing different solutions to initialize the EM algorithm, we recall the general flow of the algorithm in Fig. 3.5. The purpose of the method is to learn the model's parameter, $\{\boldsymbol{\pi}, \boldsymbol{\Sigma}, \boldsymbol{\nu}, \boldsymbol{\Omega}, \boldsymbol{\Lambda}\}$, as

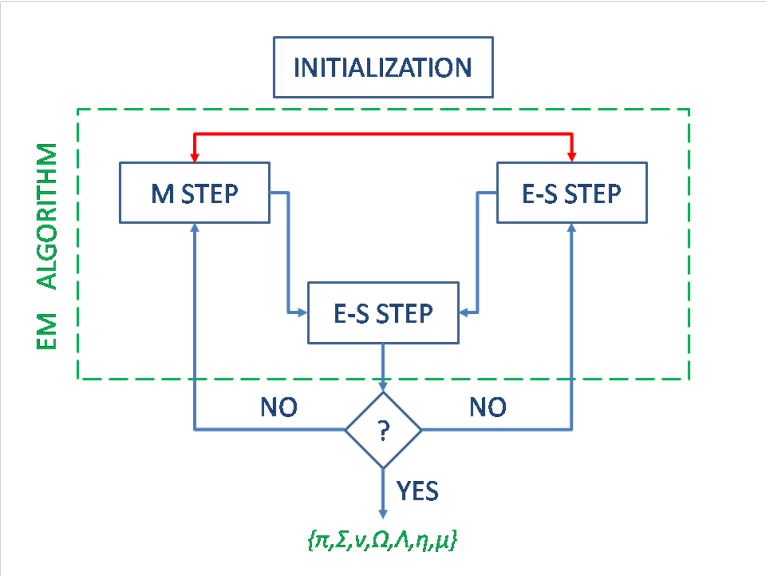
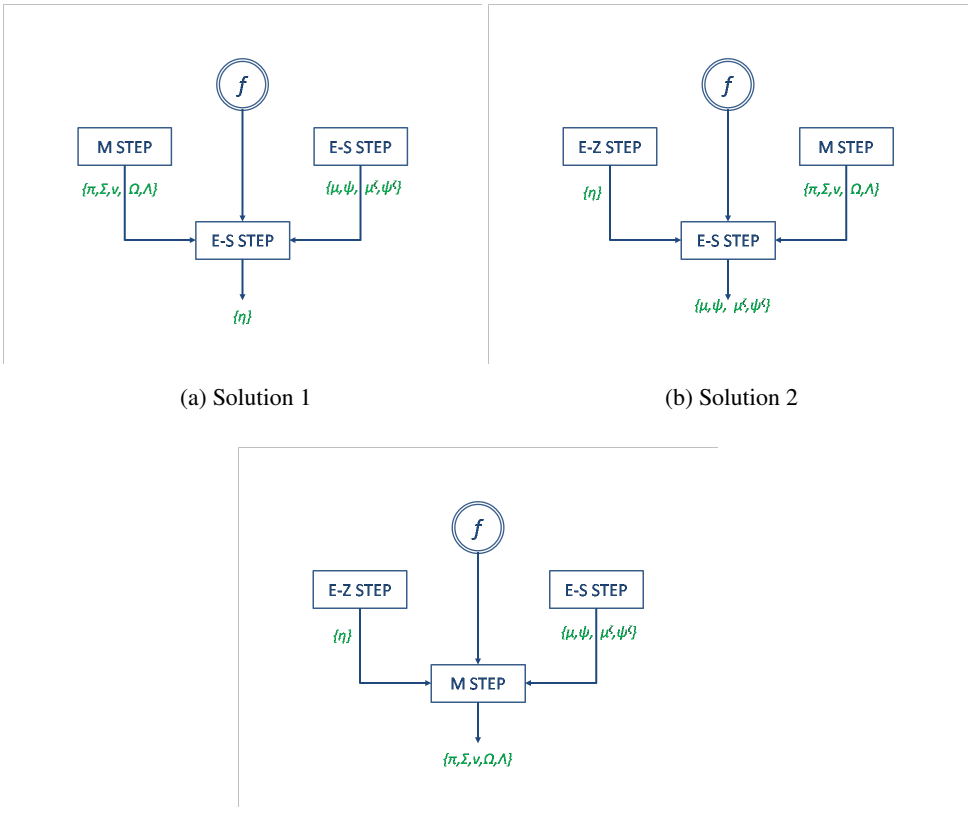


Figure 3.5: Flowchart of the EM algorithm.

well the posterior probabilities that each observation could be assigned to a certain track or source, $\{\eta\}$, and an estimation of the hidden track of each object, $\{\boldsymbol{\mu}\}$. As it is represented, each iterations involves the three steps in a order which has to define in advance. For example, either it can start with the E-S step and then goes on with the E-Z step and the M step, or it can start with the E-Z step and

then goes on with the E-S step and the M step. Nevertheless, in the end of each iteration we want to evaluate the convergence of the algorithm to decide to stop it or to continue with another iteration. Usually, the likelihood is used to evaluate the algorithm convergence, but this method has a closed-form solution that is not computationally tractable and so what we can do is to evaluate the convergence of the model's parameters or to stop the algorithm after a fixed number of iterations. There is no link between the initialization box and the rest of the flowchart because of the problem explained in the beginning. Now, we will illustrate and explain the different options in details.

The possible solutions are shown in Fig. 3.6:



(a) Solution 1

(b) Solution 2

(c) Solution 3

Figure 3.6: Three different ways to initialize the EM algorithm.

The first option is to initialize the model's parameters and the hidden tracks with

means and covariances, $\{\mu, \Psi\}$. This means to make a random guess on the parameters and to use multiple Kalman filters for the track-estimations. But, these two involves some problems. Firstly, the random guess of model's parameters could be very sensitive because it could generate results completely different from what expected, simply due to a wrong initialization of parameters themselves. Secondly, the multiple Kalman filters which are based on the formulas derived in the algorithm, require the initialization of the model's parameters, the observations and the responsibilities, which are the output of the E-Z step that we want to run after the initialization. A representation of the problem is depicted in Fig. 3.7

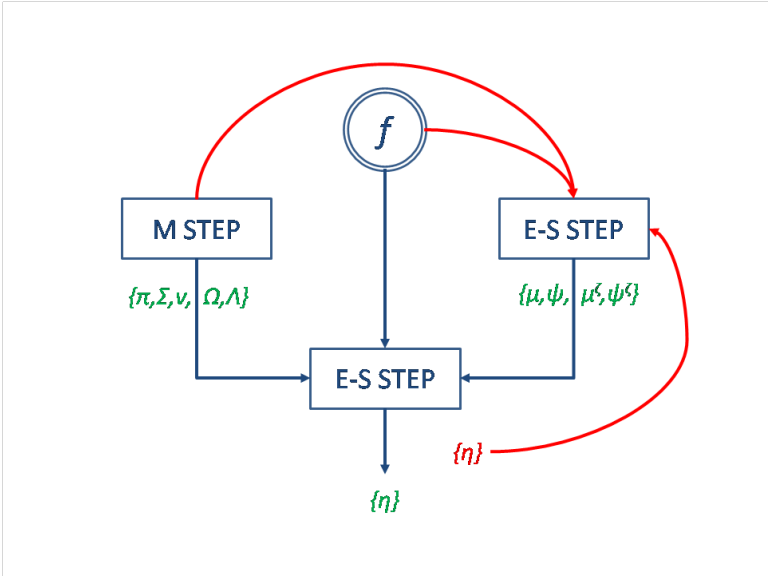


Figure 3.7: Problem related to the first initialization setting. The initialization of E-S step requires the output of the configuration, the responsibilities, beyond the observations and some parameters from the M step.

The second option is to initialize the model's parameters again and the responsibilities. To do so, a clustering technique could be used joint to a guess of some model's parameters. For example, k-means or GMM algorithm is a solution which requires an initial guess of the mixing coefficients, π, and it can provide the responsibilities and covariances of each cluster/object. This operation might be run for each interval/frame of all sequence to use and from the results it is possible to

make a guess of all other model's parameters. Here, the main problem is that data association among sources over time is not solved within the algorithm itself and so it is necessary to provide with it already in the initialization phase by means of a mechanism like distances (i.e. Euclidean or Mahalanobis). Less important is to decide with which phase going on after E-S step: it can continue with either M step or E-Z step.

In the end, the last option is to initialize the whole expectation step through a clustering technique, such as k-means or GMM, and Kalman filters. While the process for clustering is the same as above with the purpose to identify areas which should correspond to object, the Kalman filters, instead, have the task to make a first guess of tracks for each object. Again, we need a mechanism to associate properly objects over time, because the algorithm does not deal with it. Some model's parameters are computed directly through the clustering technique as before, but some of them, such as $\mathbf{\Lambda}$, are required and a guess has to be done.

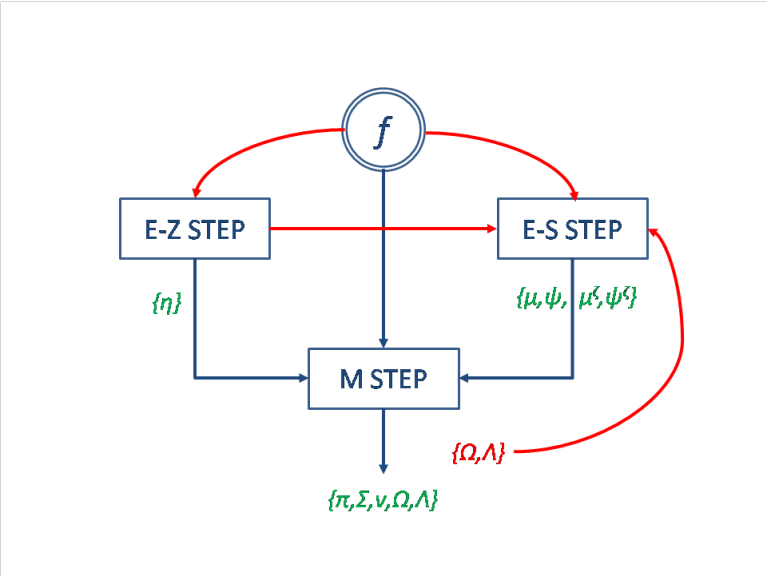


Figure 3.8: Problem related to the third initialization setting. In this case, you need the output of the E-Z step plus a guess of two model's parameters to run the E-S step. Only after these two operations the algorithm can go on with the M step and the next iterations.

The initialization of an EM algorithm usually should be not more relevant or

even complex than the algorithm itself. For sure, it is a critical point that can affect the algorithm and its convergence, but if it solves the problem by itself, you do not need the EM algorithm anymore. Comparing with the three options and according to the issues explained above for each of them, the first one seems to be the worst, because it is based on the output and a random guess of the model's parameters; the last becomes enough complex, since it requires both clustering and Kalman filters plus a guess of some parameters; and the second one, instead, seems to be more appropriate, because it involves only one technique (clustering) and a guess of a few parameters.

Chapter 4

MODEL EVALUATION

Since we added some extensions to our initial model, firstly we want to introduce different versions of the algorithm, which we used to evaluate the model.

Method #1: corresponds to the initial version of the model, so the derived algorithm takes only object position into account.

Method #2: introduces the first extension in the model, outliers.

Method #3: adds the second extension into the model, velocity.

Method #4: combines both outliers and velocity into the algorithm.

All evaluations are done for each method, leading a comparison among them in order to understand the usefulness of extensions.

4.1 Simulated Data

To demonstrate the performance of the implemented method, several simulated scenarios are created with synthetic data generated partially following the graphical model. The advantage of using simulated data is that it allows observations and hidden trajectories to be created easily and with different problems, from simpler to more difficult ones. Besides, this allows to have the ground truth, which are

useful for comparison and accuracy tests. While the trajectories are generated from deterministic functions, observations are randomly generated.

4.1.1 Data Generation

The observation generation process depicted by the graphical model in the previous chapter cannot be totally followed because it does not produce object motions similar to smooth curves, which are easier to analyse. Indeed, the Gaussian distribution assumption for temporally linked sources produces motions with random points at each time step, but we expect them to follow curves with some smoothness, such as linearity, quadratic or higher order. To emulate these motions, we chose a certain amount of points, which depends on the order of the curve to generate, within a fixed region of $x \in [-200; 200]$ and $y \in [-200; 200]$, and by means of polynomial fitting we extracted 25 points, which correspond to object position at each time step. Here an example of used points to generate a linear, a quadratic and a 3-order curves:

Track # 1 Initial point: $[175, -120]$, Final point: $[-60, 180]$

Track # 2 Initial point: $[-180, 130]$, Middle Point: $[-85, -10]$,
Final point: $[-40, -190]$

Track # 3 Initial point: $[-190, -175]$, Second Point: $[-100, -20]$,
Third point: $[50, 35]$ Final point: $[165, -195]$

Thus, except for the sources generation, the observations are generated by following the graphical model: from the position of the object in that moment together with the initialization of model's parameters $\mathbf{\Sigma}$ and mixing coefficients π. Because part of the algorithm consists of determining the association observations-to-object, we generated a variable quantity of around 100 measurements per object at every time-step. The choice of so many observations per object was aim at making the clustering part work properly, avoiding the case of objects disappearing from the scene and causing a fail of the EM algorithm itself. This issue could be addressed

in a future study. Besides, we took into account the presence of outliers (observations not generated by moving objects, or false alarms) for each scenario. So, the same scenario was generated both with outliers and without outliers. The number of outliers per interval equals to the 10% of the total number of observations generated by objects per interval.

4.1.2 Evaluation

Since the algorithm itself is complex, even the evaluation requires many things to take into account. We need to make sure that the EM converges and is improving over iterations to work properly; we need to evaluate the assignment observation-to-object, and we need to define metrics for measuring the error between positions and velocities, if considered. Besides, tracking requires to keep the identity of the object over time, so we also need a metric to make sure that we are not facing a switching problem.

Model's Parameters Convergence

As far as concerned the evaluation of the Expectation-Maximization algorithm, even though we used the variational approximation of the log-likelihood, we cannot evaluate its convergence, but alternatively we can evaluate the convergence of the model's parameters. Here, we want to analyse two different cases: the first one is about keeping fixed the parameter Λ, and the second one, instead, is related to the estimation of the same parameter. As regards all other parameters, they are always estimated. Besides, we put only figures concerning one scenario, because the behaviour is similar among all of them. The analysis is carried out by fixing the number of iteration to 100. In Fig. 4.1, the convergence takes into account all parameters, and each of them is computed by the average of norm distance between two consecutive iterations and among all intervals for one object. The graph is similar for the other objects.

Parameter π_n

$$\epsilon_{\pi}^{r+1} = \frac{\sum |\pi_{tn}^{r+1} - \pi_{tn}^{r}|}{T}$$

Parameter $\mathbf{\Sigma}_n$

$$\epsilon_{\mathbf{\Sigma}}^{r+1} = \frac{\sum ||\mathbf{\Sigma}_{tn}^{r} - \mathbf{\Sigma}_{tn}^{r+1}||}{T}$$

Parameter $\mathbf{\Lambda}_n$

$$\epsilon_{\mathbf{\Lambda}}^{r+1} = ||\mathbf{\Lambda}_{n}^{r} - \mathbf{\Lambda}_{n}^{r+1}||$$

Parameter $\boldsymbol{\nu}_n$

$$\epsilon_{\boldsymbol{\nu}}^{r+1} = ||\boldsymbol{\nu}_{n}^{r} - \boldsymbol{\nu}_{n}^{r+1}||$$

Parameter $\mathbf{\Omega}_n$

$$\epsilon_{\mathbf{\Omega}}^{r+1} = ||\mathbf{\Omega}_{n}^{r} - \mathbf{\Omega}_{n}^{r+1}||$$

As far as concerned the last three formula, they are worth for both positions and velocities parameters.

This comparison is helpful to understand that we are not able to estimate the parameter Λ, because it creates problems of convergence and it affects all other parameters at the same time. Adding a prior to this parameter might be part of a future work.

EM iterations

The EM algorithm is an iterative technique and in Fig. 4.2 we display the behaviour that usually occurs over iterations, smoothing the track closer to the real one. From the figures it is possible to notice that just after few iterations the best tracks it is able to estimate are reached.

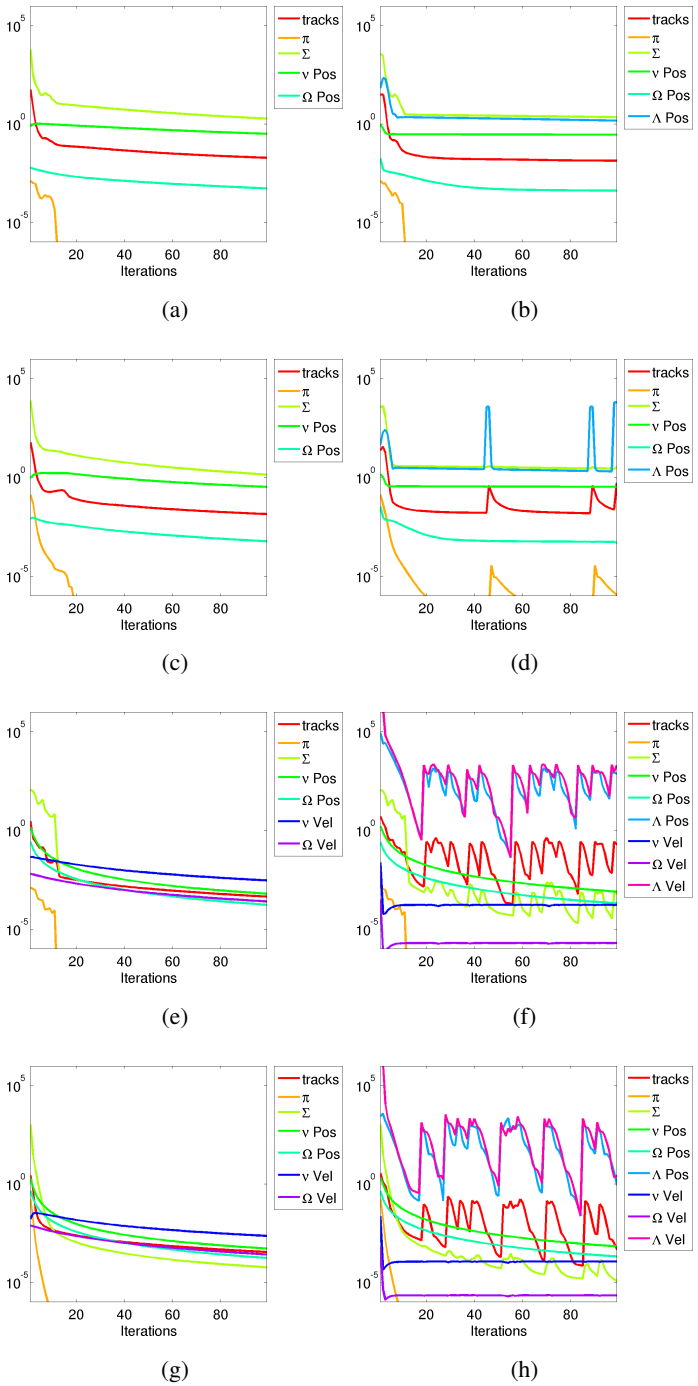
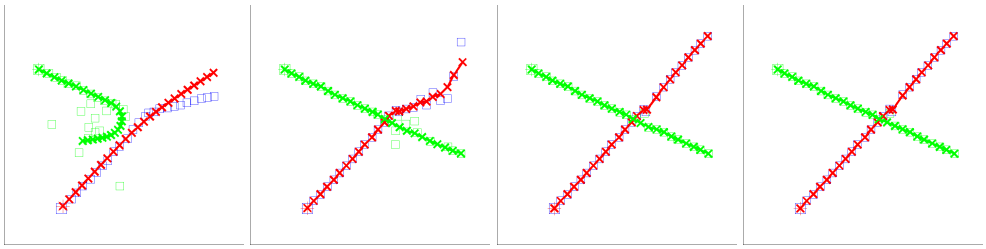


Figure 4.1: Model's parameters convergence. Every row corresponds to the results of each method respectively. While column 1 shows the convergence keeping lambda fixed, the other takes the lambda estimation into account and the linked problem.



(a) Iter 1 (b) Iter 10 (c) Iter 30 (d) Iter 50

Figure 4.2: EM iterations.

Assignment

As above-mentioned, each scenario is represented both without and with outliers. Besides, we derived the algorithm in the previous chapter in four versions, each one taking into account different things: 1) positions, 2) positions and outliers, 3) positions and velocities, and 4) positions, outliers and velocities. Because part of the model is related to the association of observations to objects (clustering phase), we can split the methods into two groups: methods 1 and 3 do not consider the outliers, methods 2 and 4 do. For the evaluation we can use only methods 1 and 2, and it's the same for the others. Defining a metric to measure the association is not easy, thus we report in Fig. 4.3 a qualitative comparison. As you can see, the first column represents the scenario without outliers and the second one with outliers. While the ground-truth of each case is depicted in the first row, the second and the third rows correspond respectively to method 1 and 2. As you can imagine, using a method that embeds the outliers does not improve the association problem in a scenario where outliers are not present. The method 1 is already sufficient to have a good clustering. Unlikely, in the other case, the method 1 starts decreasing the performances and it associates outliers to objects, affecting then the estimation of the object position. By introducing and modelling the outliers in the algorithm, referring to it as method 2, the association problem is solved again.

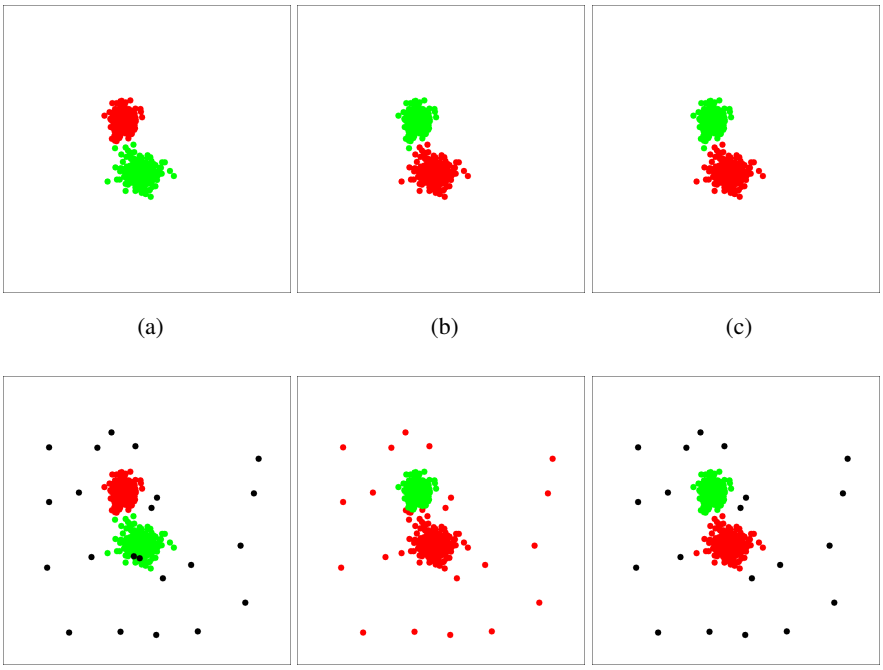


Figure 4.3: The association observation-to-object in the instant time, T = 10. While the 1st row corresponds to the scenario without outliers, the 2nd one corresponds to the same with outliers. In the first column the ground-truth observations are depicted. Second and third columns corresponds to method-1 (algorithm with only positions) and method-2 (algorithm with positions and velocities) respectively.

Error and Accuracy Metrics

Dealing with tracking, we need to evaluate some error metrics to understand how efficient is the algorithm.

Position RMSE: how far estimated position components are from ground-truth position ones for each object and at each interval.

$$\epsilon^{pos}(t) = \sqrt{\frac{\sum_{d=1}^{D}\left(x_d^{est}(t) - x_d^{gt}(t)\right)^2}{D}}$$

where D is the dimension of cartesian component vector.

Speed RMSE: how far estimated speed components are from ground-truth speed

ones for each object and at each interval. It is computed only for the algorithm versions that take velocity into account.

$$\epsilon^{spe}(t) = \sqrt{\frac{\sum_{d=1}^{D}\left(\dot{x}_d^{est}(t) - \dot{x}_d^{gt}(t)\right)^2}{D}}$$

where D is again the dimension of cartesian component vector.

Velocity Error: velocity is more accurate than speed because it takes direction into account. The error between estimation and ground-truth is computed through the angle α between the two vectors.

$$< \boldsymbol{v}^{gt}, \boldsymbol{v}^{est} >= 1 - \cos\alpha$$

If alpha is zero, there is no error; on the other hand, the maximum error is equal to 2 and it means the two vectors have opposite directions.

Mean and Standard Deviation: we also evaluated the error between estimation and ground-truth tracks as an Euclidean distance at each interval, and then we computed the mean and the standard deviation of these errors for each object, as well a global one.

$$\bar{m_\epsilon} = \frac{1}{T}\sum_{t=1}^{T} d < (x,y)_t^{est}; (x,y)_t^{gt} >$$

$$\bar{sd_\epsilon} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left[\left(d < (x,y)_t^{est}; (x,y)_t^{gt} > -\bar{m_\epsilon}\right)^2\right]}$$

Identity Accuracy: it measures a spatial-temporal relation between estimated objects $\mathcal{E}$ and ground-truth objects $\mathcal{GT}$, evaluating the constant tracking of an object by a specific estimation over time. To do so, we computed the Euclidean distance between all $\mathcal{E}$s and all $\mathcal{GT}$s at each time step, generating an association matrix. Inside the association matrix, we can retrieve many configurations that map each $\mathcal{GT}$ to at most one $\mathcal{E}$ and vice versa. Since

every association corresponds to a distance value, we can compute a cost function, which is the sum of all distance values within each configuration. The configuration with the minimum cost determines the $\mathcal{E}$ identity assigned to each $\mathcal{GT}$. Taking the initial identity of each $\mathcal{GT}$ as reference, we define the identity accuracy as the ratio between the number of time steps that the identities associated to that $\mathcal{GT}$ correspond to the initial one, and the total number of time steps.

All metrics are computed for each method. In addition, the error distribution is computed for each metric in order to make easier the comparison among methods and to understand if the improvements are relevant. The mean and the standard deviation of the norm between the estimated and ground-truth position/velocity vectors per object are computed, as well the aggregate between objects.

4.1.3 Scenarios

Now, we describe scenarios used to test the four methods and they are depicted in Fig. 4.4. All scenarios contain all observations generated by objects over time, hidden trajectories, and outliers. For testing, another version of each scenario without outliers was considered.

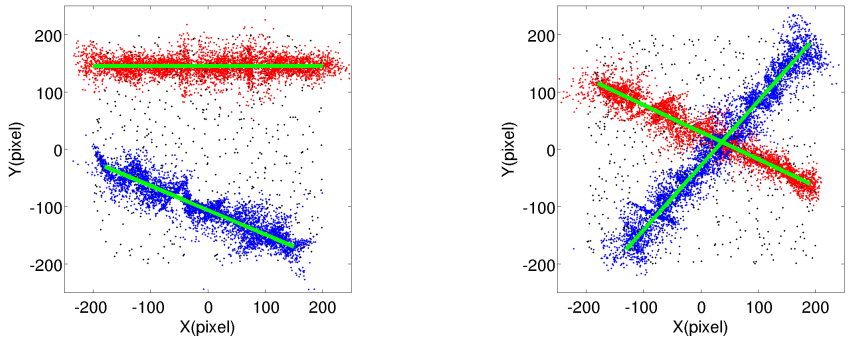
Scenario #1: is taking into account two linear tracks which never cross (see Fig. 4.4a).
It is the simplest scenario to make sure the tracker is working. The trajectories follow these functions: Track-1: $x = 16t - 200, \quad y = t - 145$
Track-2: $x = 13.2t - 180, \quad y = -5.6t - 30$

Scenario #2 considers the case of crossing between two objects (see Fig. 4.4b). The crossing problem is very common in multiple target tracking and it is closed to the switching problem. Dealing with this scenario allows to verify the effectiveness of the tracker and its capability to keep the identities. Indeed, the estimation of object paths does not provide enough information about the algorithm correctness.

Scenario #3 represents a situation similar to the previous one where the two objects do not cross, but they come nearer to each other and then they go away again (see Fig. 4.4c).

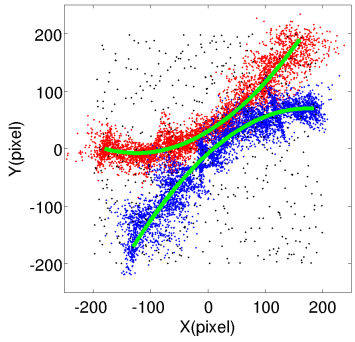
Scenario #4 takes into consideration a more complex scenario with three objects (see Fig. 4.4d). Each o them moves following a n-th order curve, with n equal to 1, 2, or 3, and crossing each other. It's helpful to understand if the algorithm still works by increasing the number of objects within the scene.

All trajectories were built by using polynomial fitting, after defining the necessary points for each curve. All points are reported in Tab. 4.1.

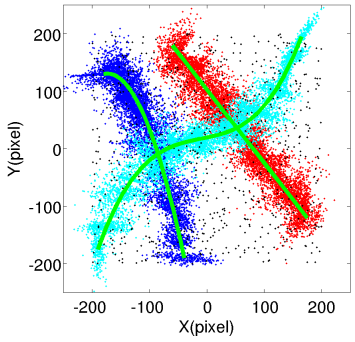


(a) Scenario #1

(b) Scenario #2



(c) Scenario #3



(d) Scenario #4

Figure 4.4: Tested scenarios: (a) two independent linear tracks, (b) two linear crossing trajectories, (c) two 2nd-order curves that come closer and then go away without crossing, and (d) three different curves which also cross each other. Object-related observations are depicted for all intervals, as well outliers.

Scenario	Track	Initial Point	Second Point	Third Point	Final Point
#1	Track-1	$[-200, -145]$	-	-	$[200, 145]$
	Track-2	$[-180, -30]$	-	-	$[150, -170]$
#2	Track-1	$[-180, 115]$	-	-	$[190, -60]$
	Track-2	$[-130, -175]$	-	-	$[190, 185]$
#3	Track-1	$[-180, 0]$	$[0, 30]$	-	$[160, -90]$
	Track-2	$[-130, -175]$	$[50, 30]$	-	$[185, 70]$
#4	Track-1	$[175, -120]$	-	-	$[-60, 180]$
	Track-2	$[-180, 130]$	$[-85, -10]$	-	$[-40, -190]$
	Track-3	$[-190, -175]$	$[-100, -20]$	$[50, 35]$	$[165, -195]$

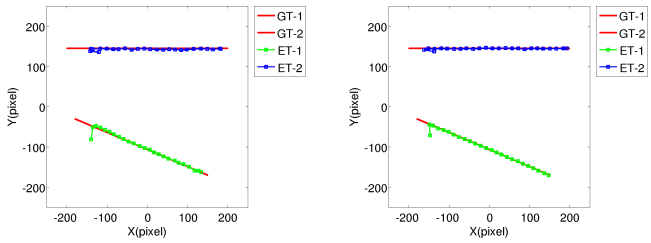
Table 4.1: Points used to build trajectories by means of polynomial fitting for each scenario.

As we previously discussed, the model's parameter convergence is not reported because it is similar for all scenarios to those we depicted, showing the same Λ estimation problem. Apart from that and assignment, we now go in detail of evaluation of each scenario taking into account all other metrics.

First of all, we reported in Fig. 4.5,Fig. 4.6, Fig. 4.7, and Fig. 4.8 the final estimation of all tracks for each scenario and for each method. Ground-truth trajectories are also depicted in order to see how estimated tracks match with them. The comparison is among the four methods, and we can roughly state a qualitative result: by introducing outliers or/and velocity in the model, we obtain a better smoothness of hidden trajectories with respect to method 1, which considers only object positions. This seems to be true in almost all scenarios, and method 4, which takes into account both outliers and velocity, seems to work better than others; indeed, estimated tracks look almost equal to their respective ground-truth. We can notice that if there is an error in the estimation of the initial point of a trajectory, the first two methods, which do not take into account the velocity, are not able to correct it over iterations. Unlikely, by modelling velocity in the algorithm we can correct this error. When modelling outliers too, the mean of clusters corresponds almost

perfectly to the source position within the track. Method 1 and 3, which do not consider the outliers, show some ambiguities when objects come nearer, and it can be seen in Fig. 4.6a, Fig. 4.6c, Fig. 4.7a, Fig. 4.7c, Fig. 4.8a, and Fig. 4.8c. So, the combination of outliers and velocity seems to solve both crossing and initial point problems.

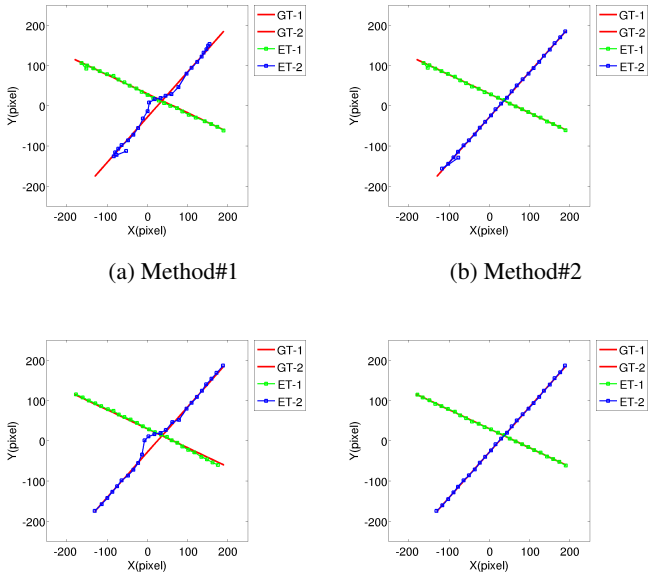
We have to make some more considerations about scenario 3 in Fig. 4.7. Except for the fourth method, the final estimation of others is really far from the right one, showing a strange behaviour when coming closer each other. To better understand it, we reported a sequence of images in Tab. 4.2 and Tab. 4.3, which represents the clustering estimation over some time steps, comparing the methods at the same time. We can notice that method 1 and 3 are not able to deal with outliers, producing a unique cluster for the two object when come closer and another one corresponding to false alarms. Such an error propagates until the end, generating a trajectory stuck around a point as we can see in Fig. 4.7a. As regards method 2, the clustering problem occurs later and in a different way. It is able to distinguish the two objects longer than method 1 and method 3, but after that it produces again only one cluster for the two objects, generating a trajectory in the middle between the two real ones. Another cluster, instead, is estimated from the subset of outliers and it has a mean point every time in a different position in the area. Due to this highly change of the estimated position in the clustering, the tracker tried to correct it as much as possible, providing a final trajectory totally different from the real one. Nevertheless, the combination of velocity and outliers in method 4 allows the algorithm to estimate almost correct trajectories with respect to the real ones.



(a) Method#1 (b) Method#2

(c) Method#3 (d) Method#4

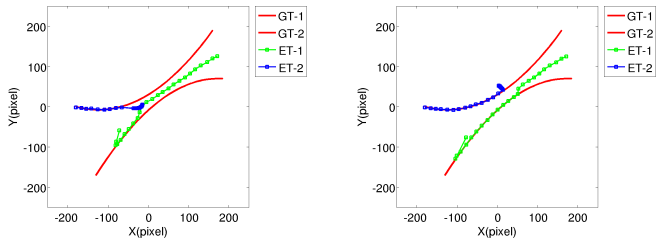
Figure 4.5: Comparison among methods of estimated tracks with their respective ground-truth for scenario-1.



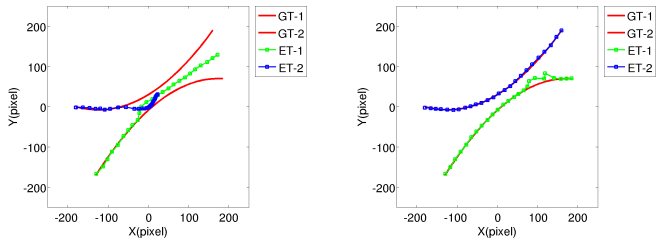
(a) Method#1 (b) Method#2

(c) Method#3 (d) Method#4

Figure 4.6: Comparison among methods of estimated tracks with their respective ground-truth for scenario-2.

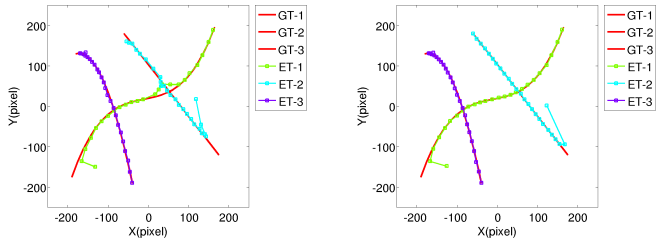


(a) Method#1 (b) Method#2

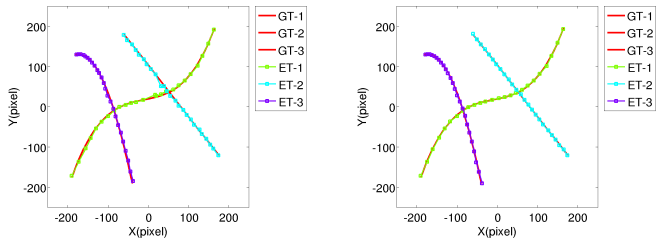


(c) Method#3 (d) Method#4

Figure 4.7: Comparison among methods of estimated tracks with their respective ground-truth for scenario-3.



(a) Method#1 (b) Method#2



(c) Method#3 (d) Method#4

Figure 4.8: Comparison among methods of estimated tracks with their respective ground-truth for scenario-4.

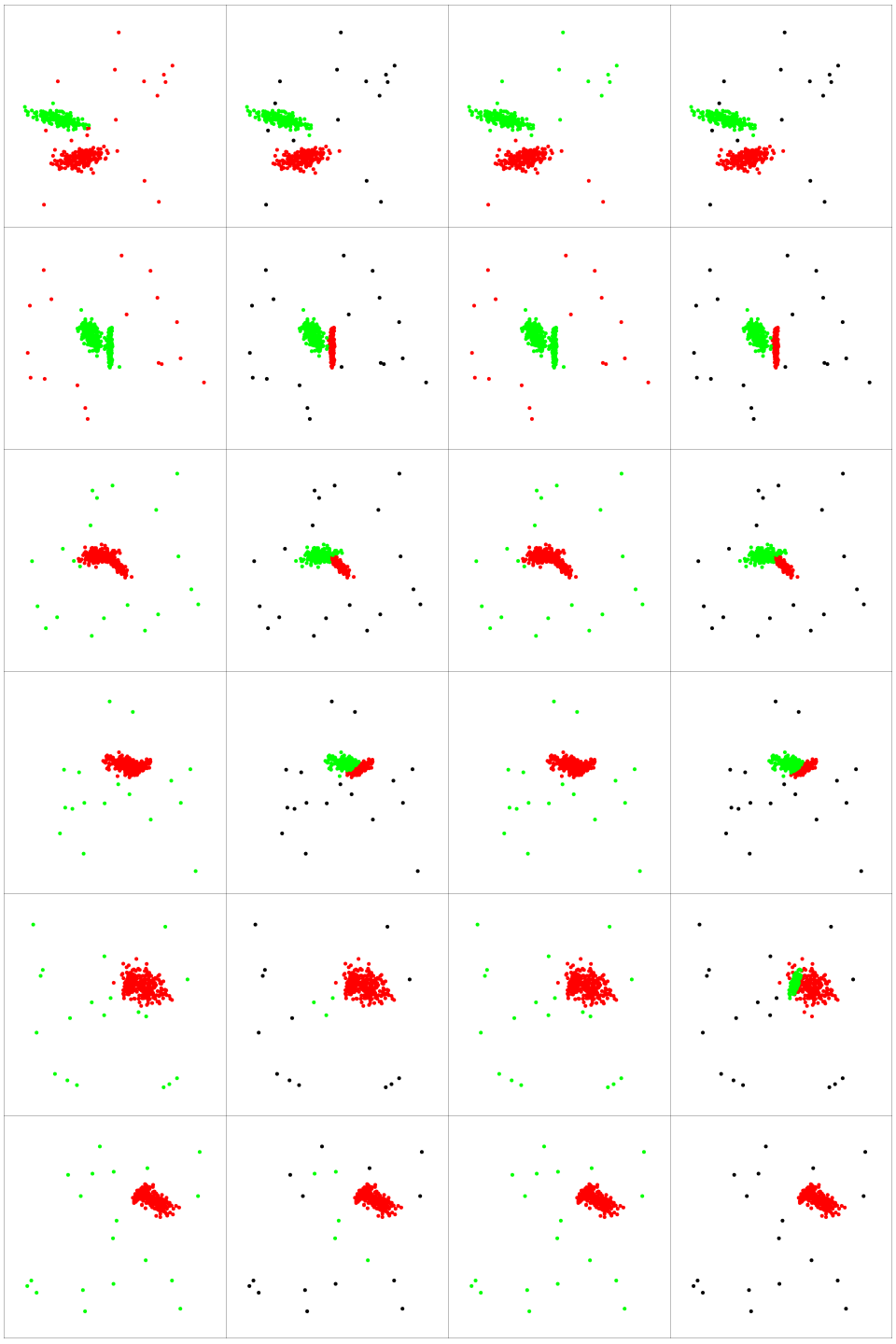


Table 4.2: A comparison of clustering among the four methods along columns in scenario 3. Each row represents intervals #5, #10, #11, #15, #16, #18.

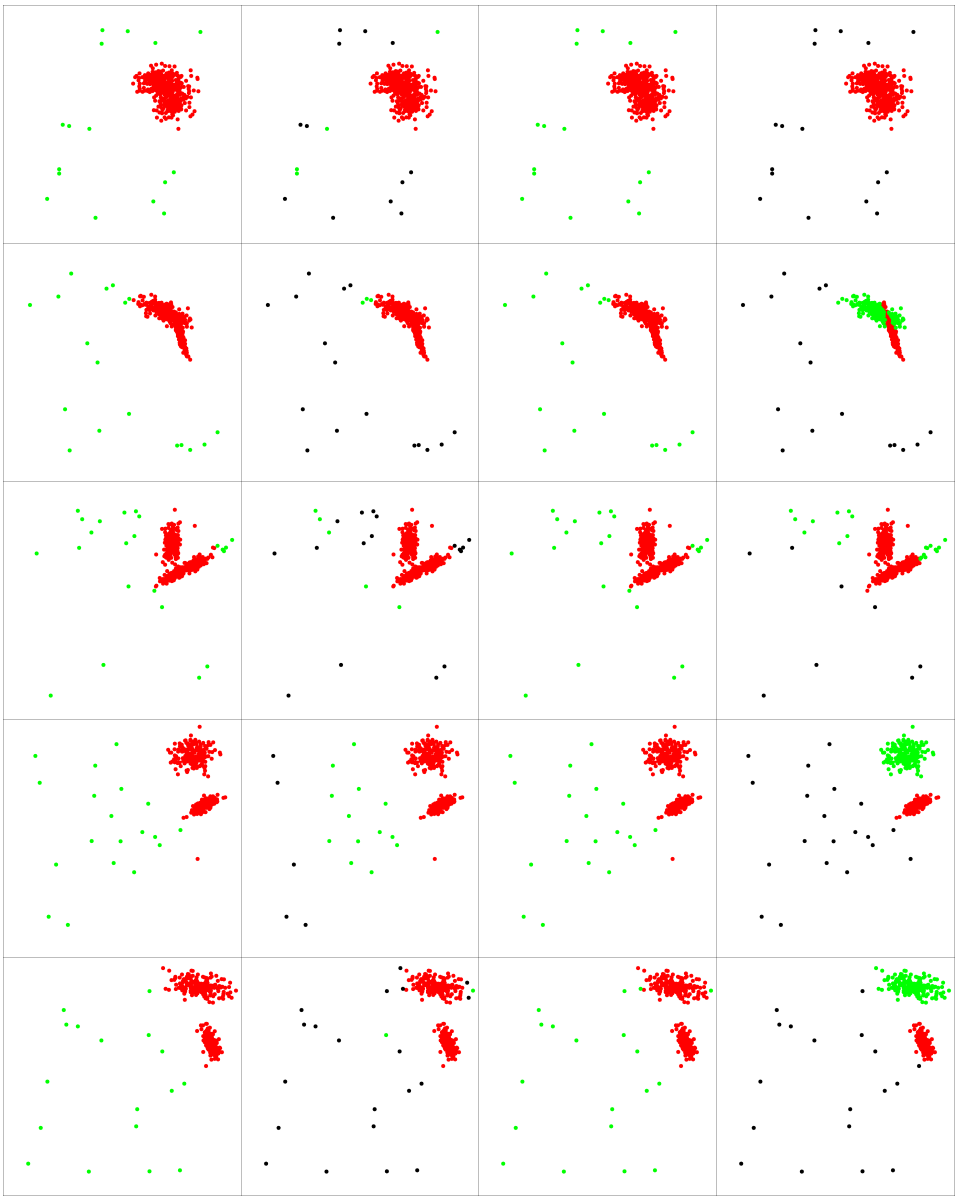
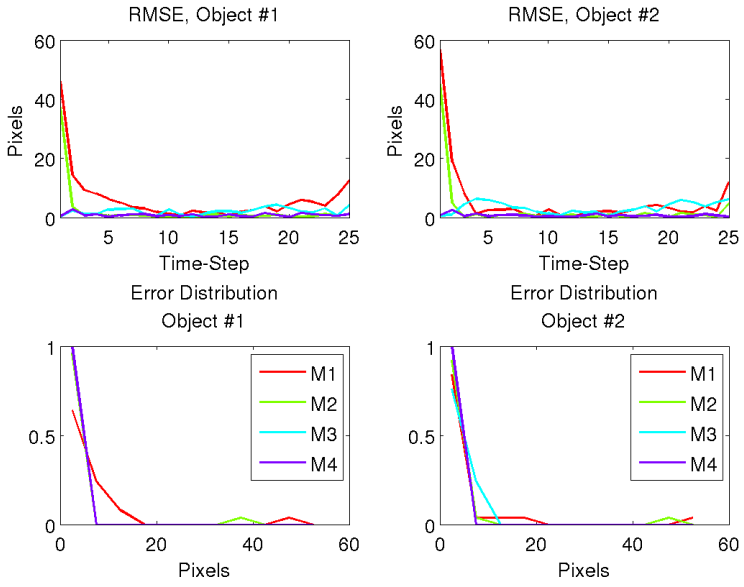


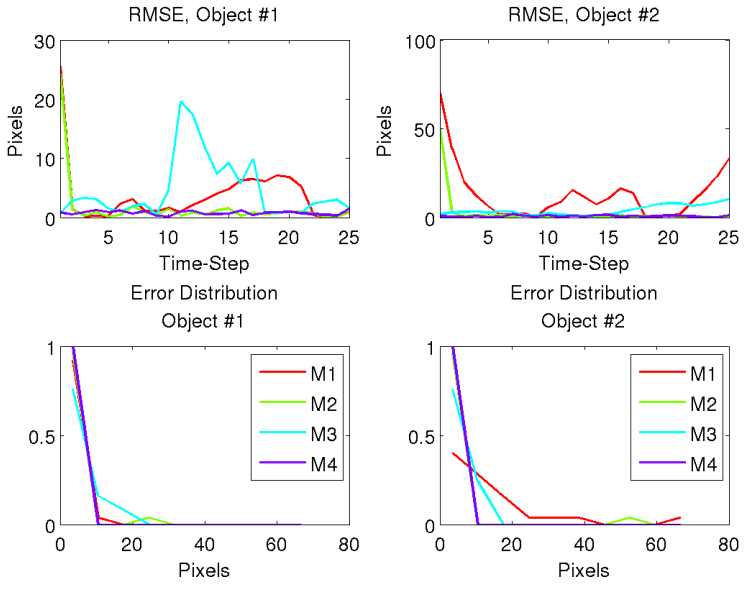
Table 4.3: A comparison of clustering among the four methods along columns in scenario 3. Each row represents intervals #19, #20, #21, #24, #25.

Now, we want to analyse these estimations more accurately through the error metrics, which let us have a quantitative comparison among methods.
In Fig. 4.9 and Fig. 4.10, we reported the root mean square error (RMSE) between the estimated and ground-truth object positions at each interval. The error distribution of the metric is also depicted below every graph, leading sometimes a more

quickly understanding of the comparison. It is noteworthy that there is a smaller error for the methods 3 and 4, which take velocity into account, in almost all graphs. Besides, the performances are improved when the model takes outliers into account. The Tab. 4.4 allows us to understand better these considerations.We computed the mean and the variance of the absolute difference between ground-truth and estimated positions per object, as well the global ones. Data shows method 2, which introduces outliers in the algorithm, and method 3, which introduces velocity, have a better performance with respect to the initial method, which takes only object position into account, but a comparison between the two is not relevant. Indeed, method 2 has generally a smaller error, but higher variance with respect to method 3. Instead, modelling both within the algorithm allows to have a very good improvement of performance in all scenarios, both in terms of mean error and in terms of variance. As far as concerned the object positions, we can conclude that adding both outliers and object velocity in our initial model could improve a lot the performance of EM algorithm.

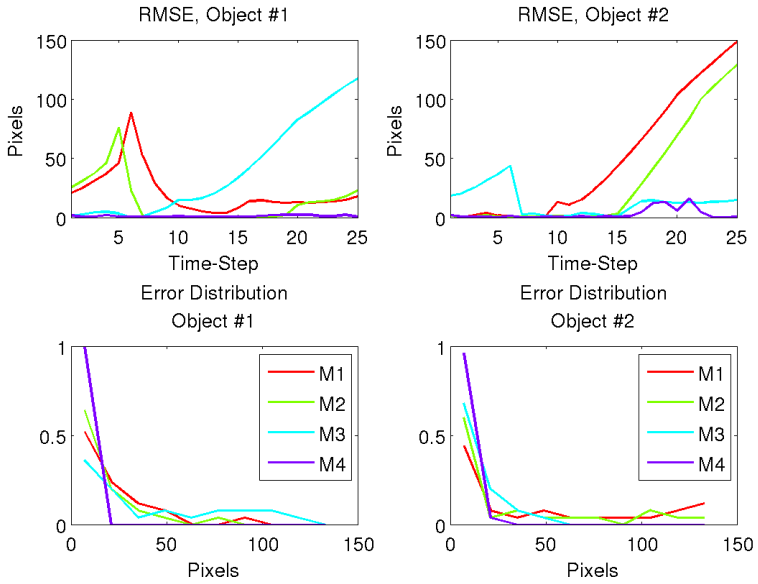


(a) Scenario-1

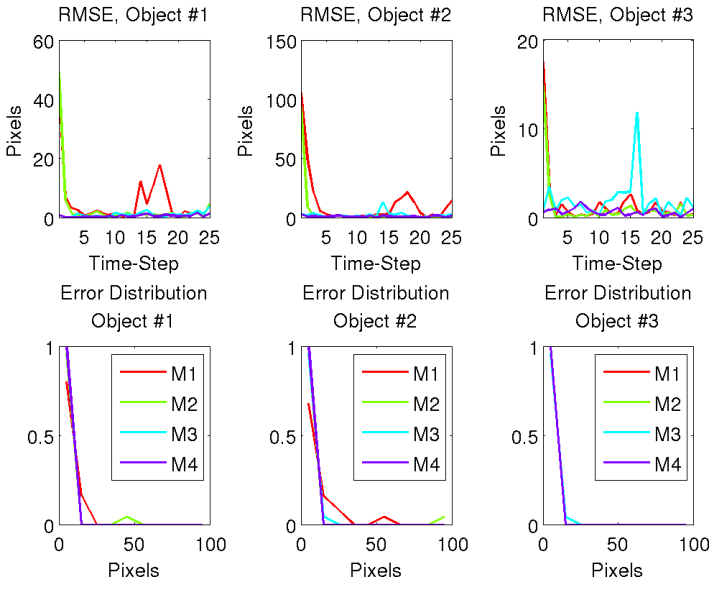


(b) Scenario-2

Figure 4.9: RMSE between ground-truth and estimated object positions with respective error distribution for scenarios 1 and 2. The metric is computed for every object and all methods.



(a) Scenario-3



(b) Scenario-4

Figure 4.10: RMSE between ground-truth and estimated object positions with respective error distribution for scenarios 3 and 4. The metric is computed for every object and all methods.

Scenario	Metric	Method 1	Method 2	Method 3	Method 4
#1	mean # 1	5.67 ± 9.31	2.14 ± 7.28	1.85 ± 1.48	0.70 ± 0.67
	mean # 2	4.67 ± 11.69	2.28 ± 8.97	2.82 ± 2.44	0.59 ± 0.57
	global mean	5.17 ± 10.53	2.21 ± 8.13	2.33 ± 2.06	0.65 ± 0.62
#2	mean # 1	3.18 ± 5.41	1.61 ± 4.70	4.32 ± 5.40	0.72 ± 0.45
	mean # 2	12.76 ± 15.68	2.70 ± 9.69	3.30 ± 3.64	0.67 ± 0.58
	global mean	7.97 ± 12.62	2.15 ± 7.60	3.81 ± 4.61	0.70 ± 0.52
#3	mean # 1	19.32 ± 19.84	12.30 ± 18.69	42.00 ± 47.23	0.96 ± 0.81
	mean # 2	50.92 ± 58.11	31.44 ± 47.26	11.70 ± 12.73	2.60 ± 4.44
	global mean	35.12 ± 46.03	21.87 ± 37.03	26.85 ± 37.63	1.78 ± 3.28
#4	mean # 1	4.95 ± 9.50	2.96 ± 9.64	0.92 ± 0.98	0.48 ± 0.46
	mean # 2	10.94 ± 22.55	4.30 ± 18.49	1.87 ± 2.54	0.87 ± 0.73
	mean # 3	1.31 ± 3.49	0.96 ± 2.95	1.76 ± 2.38	0.59 ± 0.46
	global mean	5.73 ± 14.72	2.74 ± 12.16	1.52 ± 2.12	0.64 ± 0.59

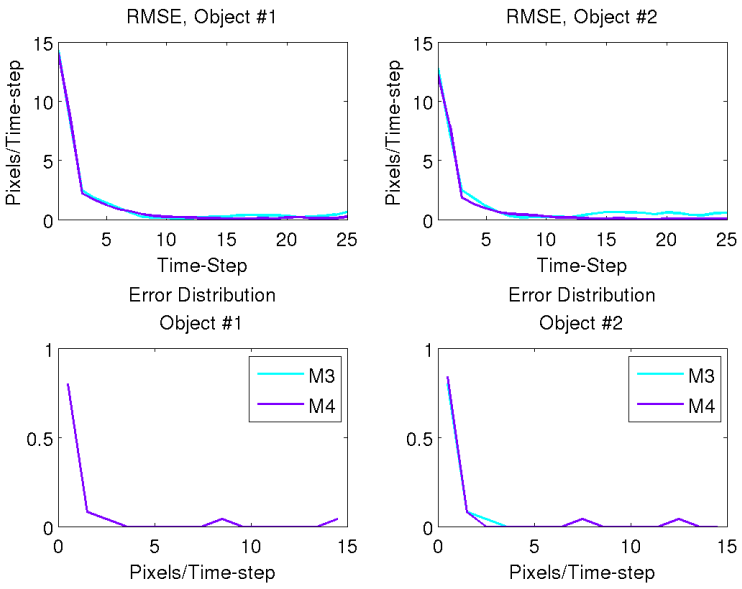
Table 4.4: Mean error and variance of the absolute difference between ground-truth and estimated object position vectors for each scenario. An aggregate mean error is also computed.

Since we introduced object velocity for methods 3 and 4, we also need to evaluate the error of this quantity with respect to its ground-truth. In this case, we computed the metric in two different ways: one is the root mean square error between the estimation and the ground-truth quantities, and the other is the angle error between the two quantities again. As we said, the second one is more precise because it takes the direction into account. The error distributions is always displayed for both, below every graph. So, Fig. 4.11, Fig. 4.12, Fig. 4.13, and Fig. 4.14 illustrate the comparison between the two metrics for scenario 1, scenario 2, scenario 3, and scenario 4, respectively. We notice the method 4, which also embeds outliers, works better than method 3 everywhere. This is confirmed by the values in Tab. 4.5. Again, the mean and the variance of the absolute difference between ground-truth and estimated velocity vectors are used. Improvements are not so large as the case of object positions, but still corroborate the previous considerations. Besides, we can see how velocity angle error is approximately zero

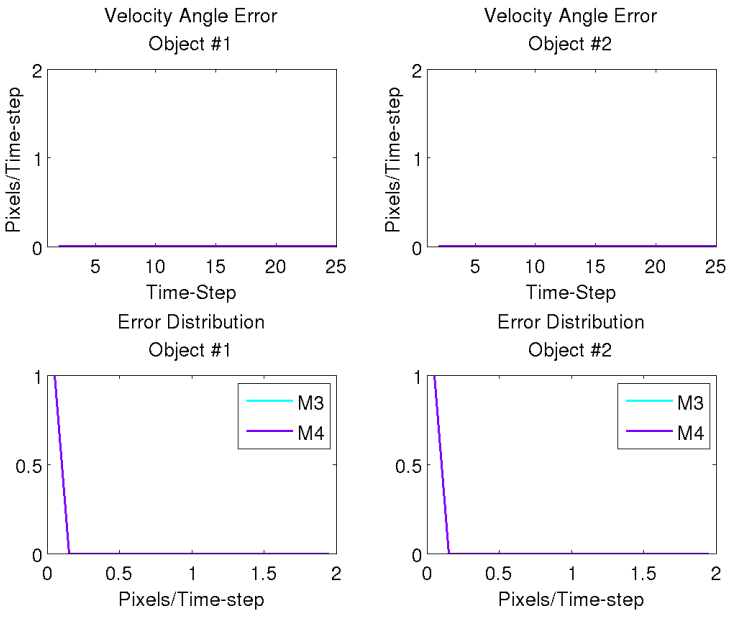
almost everywhere, even when RMSE is not.

Scenario	Metric	Method 3	Method 4
#1	mean # 1	1.03 ± 3.26	0.93 ± 3.25
	mean # 2	1.23 ± 2.81	1.00 ± 2.81
	global mean	1.13 ± 3.03	0.97 ± 3.02
#2	mean # 1	1.22 ± 2.82	0.68 ± 2.80
	mean # 2	0.89 ± 2.35	0.56 ± 2.32
	global mean	1.05 ± 2.59	0.62 ± 2.56
#3	mean # 1	6.53 ± 4.99	0.77 ± 2.09
	mean # 2	2.41 ± 3.54	1.23 ± 2.92
	global mean	4.47 ± 4.78	1.00 ± 2.53
#4	mean # 1	1.02 ± 1.34	0.92 ± 1.29
	mean # 2	3.00 ± 4.74	2.87 ± 4.72
	mean # 3	0.65 ± 2.23	0.51 ± 2.21
	global mean	1.55 ± 3.27	1.43 ± 3.25

Table 4.5: Mean error and variance of the absolute difference between ground-truth and estimated object velocity vectors for each scenario. An aggregate mean error is also computed.

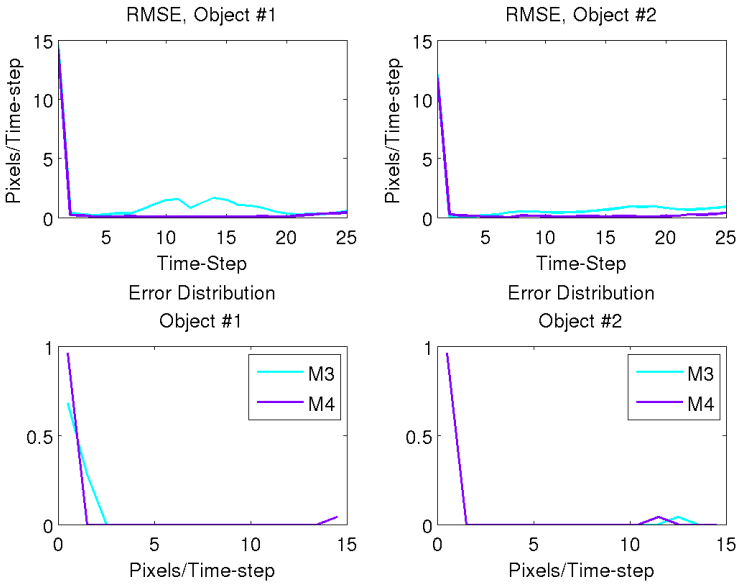


(a) Velocity RMSE and error distribution per object

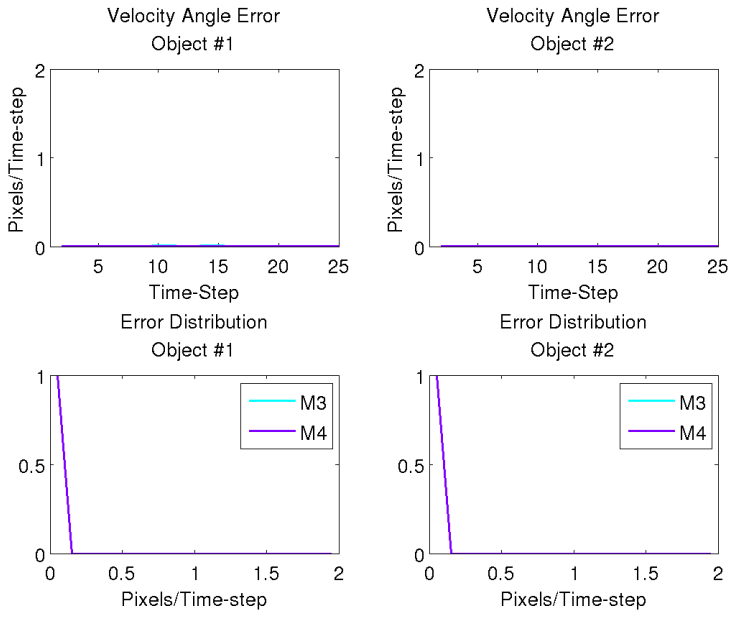


(b) Velocity angle error and error distribution per object

Figure 4.11: Scenario 1: (a) RMSE between ground-truth and estimated object velocities with respective error distribution. (b) Velocity angle error. Metrics are computed for every object and all methods.

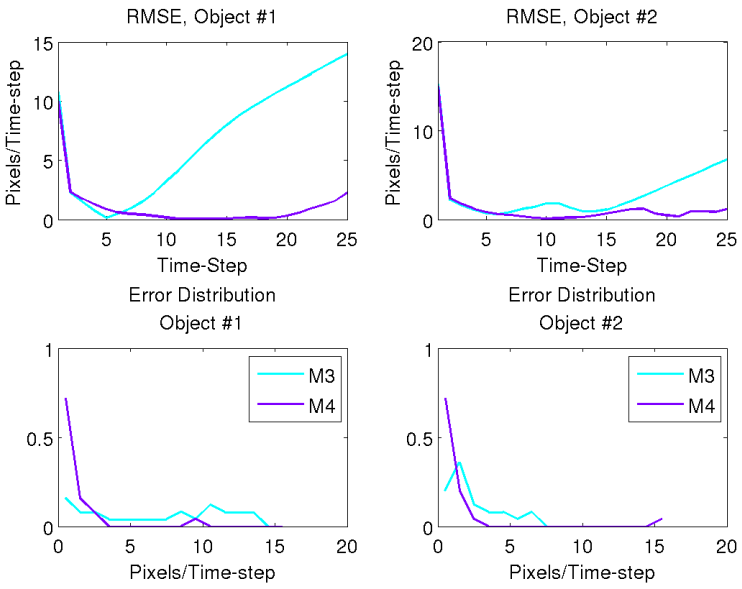


(a) RMSE of velocity

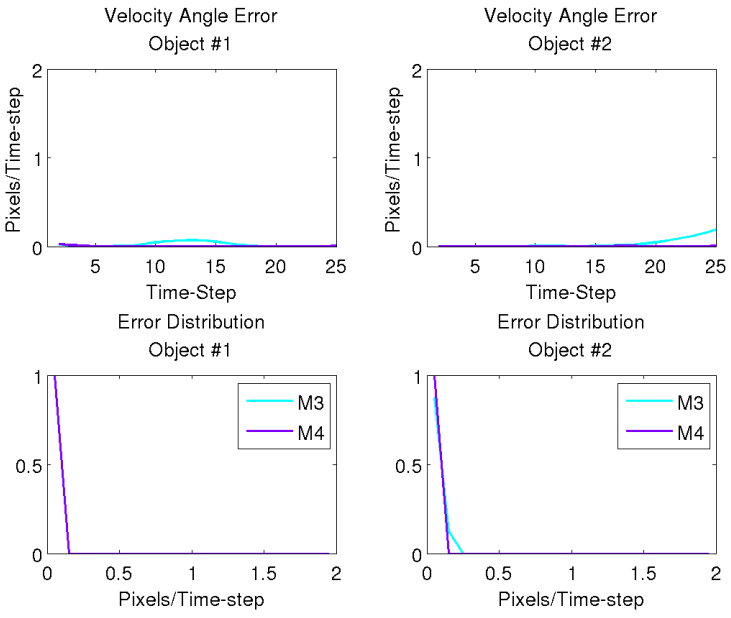


(b) Velocity angle error

Figure 4.12: Scenario 2: (a) RMSE between ground-truth and estimated object velocities with respective error distribution. (b) Velocity angle error. Metrics are computed for every object and all methods.

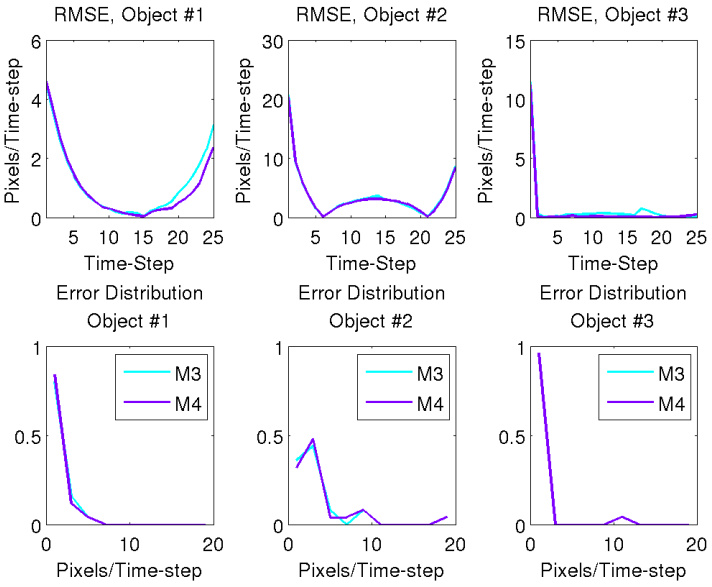


(a) RMSE of velocity

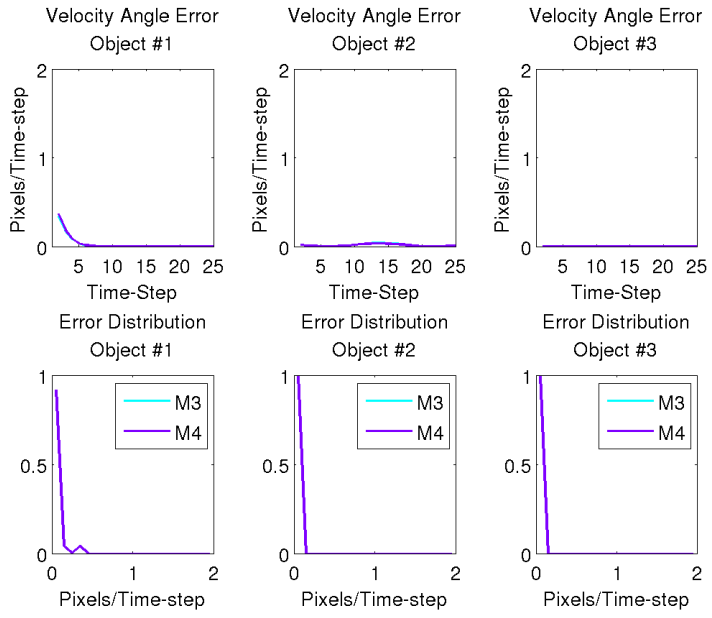


(b) Velocity angle error

Figure 4.13: Scenario 3: (a) RMSE between ground-truth and estimated object velocities with respective error distribution. (b) Velocity angle error. Metrics are computed for every object and all methods.



(a) RMSE of velocity



(b) Velocity angle error

Figure 4.14: Scenario 4: (a) RMSE between ground-truth and estimated object velocities with respective error distribution. (b) Velocity angle error. Metrics are computed for every object and all methods.

Nevertheless, all error metrics considered until now do not provide us a complete evaluation of our algorithm. They are necessary, but not sufficient. Tracking requires to keep correctly the identity of objects, and measured errors do not embed such information, even if we can notice larger values which could tell us something is going wrong. For this reason, we computed the identity accuracy: the percentage of time that an estimated track corresponds to its ground-truth. Since for scenarios 1, 2 and 4 we succeeded with 100% in all methods, we report results of only scenario 3 in Tab. 4.6. Although the method 1 shows a good performance, the combination of outliers and velocity within the algorithm allowed to reach the maximum percentage.

Object	Method 1	Method 2	Method 3	Method 4
# 1	0.80	0.84	0.76	1.00
# 2	0.80	0.84	0.76	1.00

Table 4.6: Accuracy of the identity. In this table, only scenario 3 is reported, because the values for the others are equal to 1.

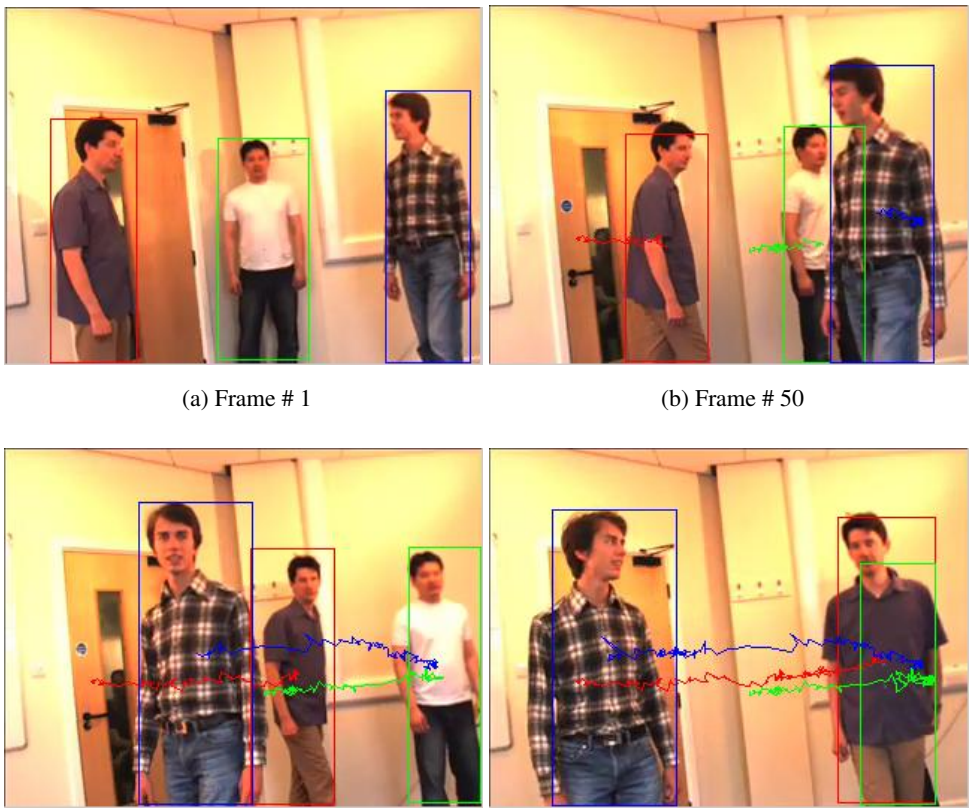
In conclusion, we showed our algorithm works in different simulated scenarios, and the introduction of outliers and object velocity within the model increases remarkably the performance. Modelling false alarms is particularly useful in scenario with outliers inside. Then, thanks to both outliers and velocity, we are able to solve the identity switching problem and to obtain very nice smoothed curves close to the true ones.

4.2 Real Data

The algorithm is not only tested on simulated scenarios, but also on some realistic videos. Two of them are taken from the published Computational Audio-Visual Analysis (CAVA) dataset (recorded at The University of Sheffield, UK) and another from a dataset recorderd directly at INRIA Rône-Alpes in Grenoble. The latter is split into two sub-sequences in order to evaluate two different cases. The CAVA dataset is a unique collection of three synchronised data streams obtained from a binaural microphone pair, a stereoscopic camera pair and a head tracking device [Arnaud et al., 2008]. Details about the acquisition system and the post-processing of data are provided in the paper. For the purpose of this project, only a portion of the video stream and the sequence from one camera are used. The other video was recorded according to the same purpose that means with multiple streams. In this case, the POPEYE system is used. In all videos, people are recorded in such a way to be fully visible or that half of body is visualized.

4.2.1 Annotation

We annotated every video by hand in order to have the ground-truth for evaluating the results. The annotation is carried out by drawing a box for each person and for each frame of the whole sequence. Thus, we are able to reconstruct the path of each of them directly from the box, by computing the center of the box. Boxes are also useful to evaluate the accuracy of the identities and clusters. Since the operation is done by hand, both boxes and path are not so precise, but they are good enough for the evaluation. Besides, 2D images are a projection of the 3D real world and the movement of people coming closer to the camera or in the opposite way results in different sizes of boxes. So, boxes are affected by scaling and translation transformations over time, and not by rotation. In Fig. 4.15, some frames with boxes and tracks are depicted from one video of the CAVA dataset.



(a) Frame # 1 (b) Frame # 50

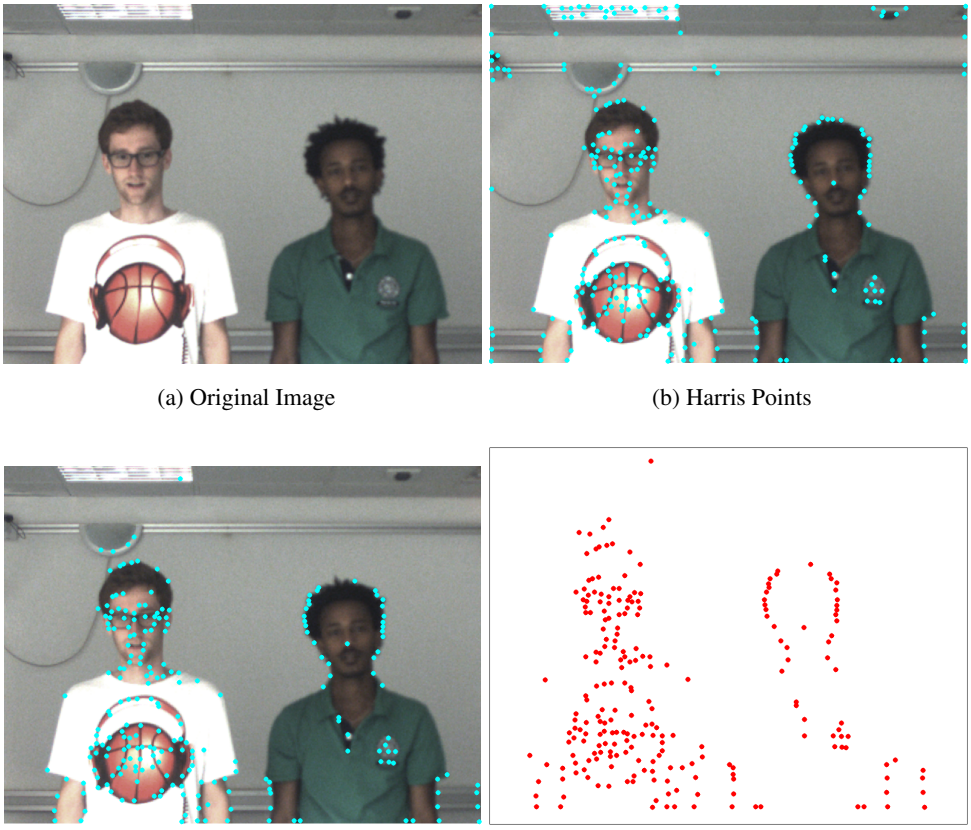
(c) Frame # 100 (d) Frame # 150

Figure 4.15: Example of annotation in some frames of a video, showing both boxes and central positions. Besides, the history of positions for each object are depicted and they are well-seen in (c) and (d).

4.2.2 Data Processing and Evaluation

Harris points and motion filtering Since our algorithm does not work directly on each image of the video sequence, we need to generate observations. These observations represent interest points which should rely on people present in the scene. To do so, we decided to use the Harris corner detector [Harris and Stephens, 1988], which generates many observations inside the figure. However, we would like to have interest points related only to people an so we need a mechanism to preserve only those. Our approach is to filter them on the basis of motion. Because we want to track moving object and videos contain people moving around a room,

then all Harris points generated above people are not in the same position across frames. So, we chose to filter all static points over 5 or 10 frames. However, not all points which do not rely on people are filtered because there could be areas affected by illumination which generated points every time in different positions. In this case, it is very useful for us to have a method which takes into account the outliers. In Fig. 4.16, there is a visual representation of this process.



(a) Original Image (b) Harris Points

(c) Motion filtering (d) Observations

Figure 4.16: Process to generate observations from a video sequence. Example of one frame.

Metrics Evaluating multi-object tracking is a challenging task and various papers have been proposed over years with the purpose to define general metrics. [Smith et al., 2005] is an interesting reference paper. In visual tracking, we have to deal with extend objects that can be represented by a shape like a rectangular, as we did

in our annotation. These rectangular boxes are referred to as *ground truth objects ($\mathcal{GT}$)* for our evaluation. The outputs of the tracker are elliptical covariances, which are referred to as *estimates ($\mathcal{E}s$)*. The central point of these $\mathcal{E}$s corresponds to the actual position of objects. [Smith et al., 2005] defines recall and precision as two fundamental measures to test the coverage between $\mathcal{GT}$ and $\mathcal{E}$, requiring high values for both. In our case, we define the **object coverage accuracy** as the ratio between the number of frames that each $\mathcal{GT}$ is covered successfully by at most one $\mathcal{E}$ and the total number of frames. An object is successfully covered when the central point of $\mathcal{E}$ is inside the rectangular box of the associated $\mathcal{GT}$, and we call it coverage test. So, this is the new metric which we added to all previous metrics, defined for simulated scenarios, and that we used for the evaluation of next scenarios.

4.2.3 Scenarios

Now, we illustrate the four recorded video sequences under test.

Scenario #1: TTOS (Tracking test; one speaker) video sequence from the CAVA dataset; "one speaker, walking while speaking continuously though the whole scene; the speaker moves in front of the camera and passes behind; he reappears from the right, and turns to the cameras". Since our algorithm is not able to deal directly with exit or enter of a person in the scene, we decided to take into account part of the video. This is an easy case to be sure algorithm is working properly a video too.

Scenario #2: a segment of a sequence recorded at INRIA in which two people move front and back. We are interested in evaluating the algorithm and making sure any identity switch appears.

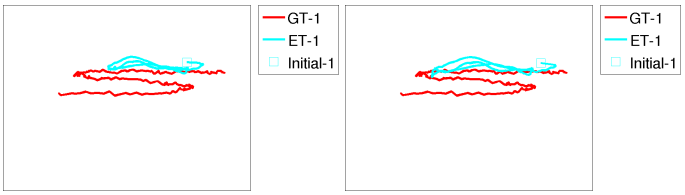
Scenario #3: a second segment of the previous sequence in which the two people switch their position. In this case, the problem of occlusion arises when they are switching, and since we have not modelled any mechanism yet to deal

with it, we want only to analyse if the algorithm can recover the identity after the occlusion itself. Indeed. we recall that occlusion is still a very tricky problem and does not allow to understand what happens during the whole period that it occurs.

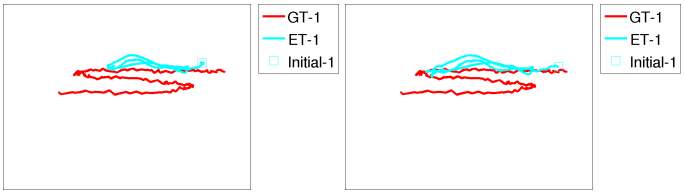
Scenario #4: CTMS3 (Clustering test; multiple speakers) video sequence from the CAVA dataset. We considered only the initial part in which three actors move around in the scene. It is a more complex scenario than the previous one with still occlusions, but more interesting trajectories to reconstruct.

As we did for simulated scenarios, we now evaluate these four scenarios taken from real recorded videos.

First of all, in Fig. 4.17, Fig. 4.18, Fig. 4.19, and Fig. 4.20, we reported the estimation of trajectories for all methods and for each scenario, respectively. As well, the annotated ground-truth trajectories are depicted. Except for the first two scenarios where estimated and ground-truth tracks are not so far from each other because no occlusion occurs, Fig. 4.19 and Fig. 4.20 show the estimated tracks are affected by this problem: the estimated tracks tend to come back in their original position or follow the path of another person. Just having a loot at this graphs, the error between the estimation and ground-truth trajectories should not be so small, but it will increase a lot in presence of occlusions.

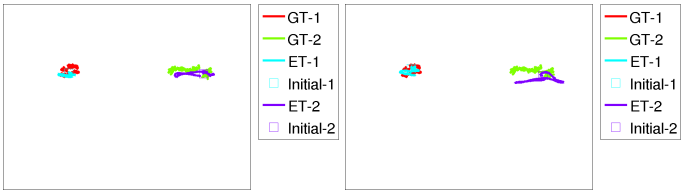


(a) Method #1 (b) Method #2

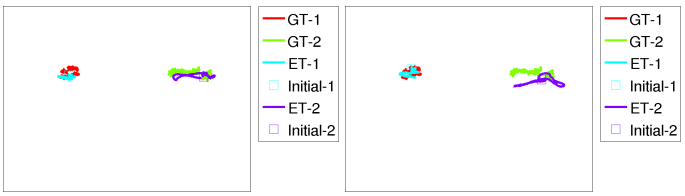


(c) Method #3 (d) Method #4

Figure 4.17: Comparison of the final estimated track with the ground-truth among all proposed methods

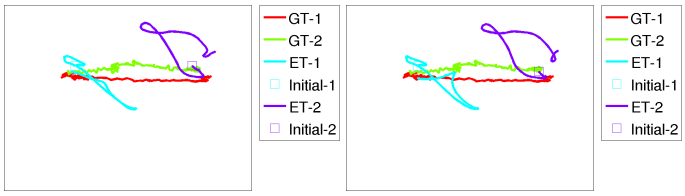


(a) Method 1 (b) Method 2

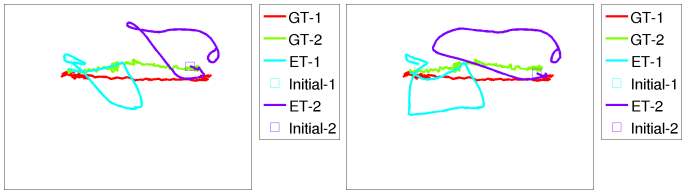


(c) Method 3 (d) Method 4

Figure 4.18: Comparison of the final estimated track with the ground-truth among all proposed methods

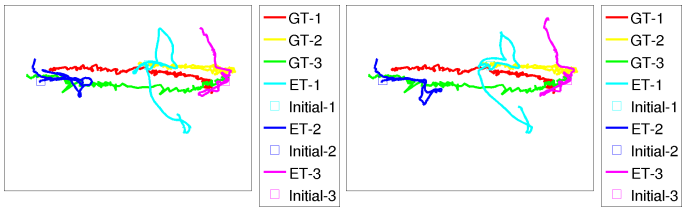


(a) Method 1 (b) Method 2

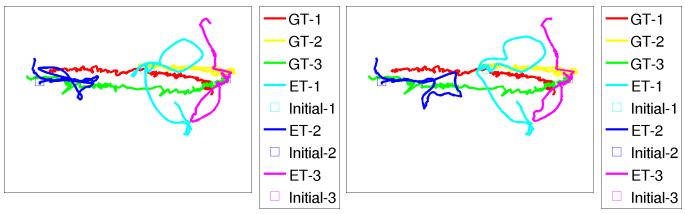


(c) Method 3 (d) Method 4

Figure 4.19: Comparison of the final estimated track with the ground-truth among all proposed methods



(a) Method 1 (b) Method 2

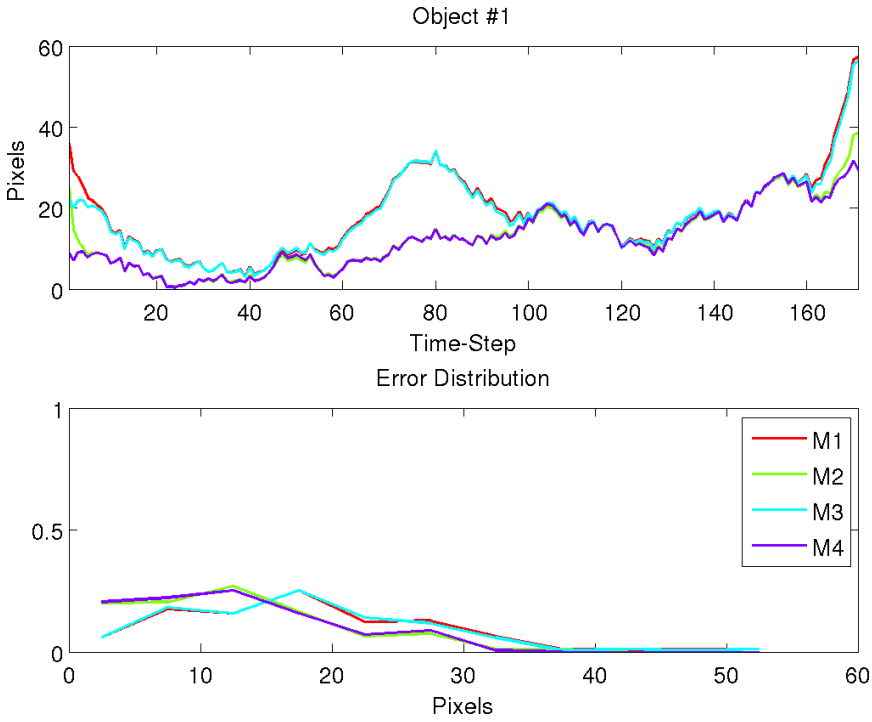


(c) Method 3 (d) Method 4

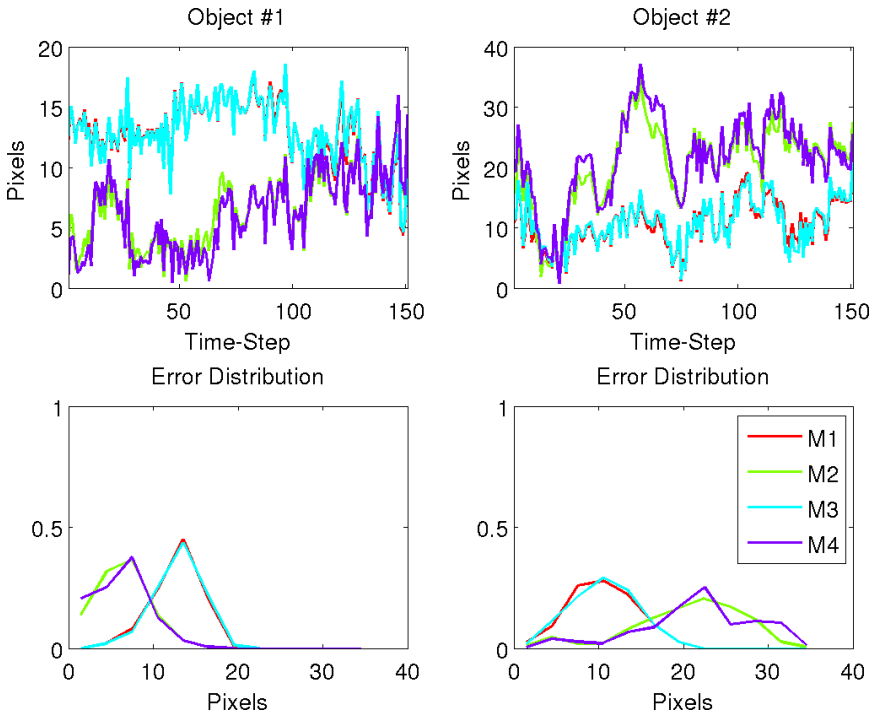
Figure 4.20: Comparison of the final estimated track with the ground-truth among all proposed methods

As in the case of synthetic data, we computed the root mean square error between ground-truth and estimated object positions for each method, and we reported results in Fig. 4.21 and Fig. 4.22 for all scenarios. Again, the error distribution is displayed to understand which method works better. As we expected,

we can immediately notice that errors start being high in terms of pixel. Although scenarios 1 and 2 show errors in the range of 0-20 pixels and we can state they are produced by the imprecision during the annotation, differently higher errors are in scenarios 3 and 4, and they confirm the wrong estimation caused by occlusions. This is a consideration worth for all methods: indeed, we cannot state there is a method that works better than others. So, we can have a look at values in Tab. 4.7. They correspond to the mean and variance of the absolute difference between ground-truth and estimated object position vectors for each scenario. A global value that does not differentiate among objects is also computed. Values confirm that outliers and velocity extensions into the algorithm do not increase the performance, apart from the easier scenarios where occlusion does not occur and we can notice an improvement by modelling outliers.

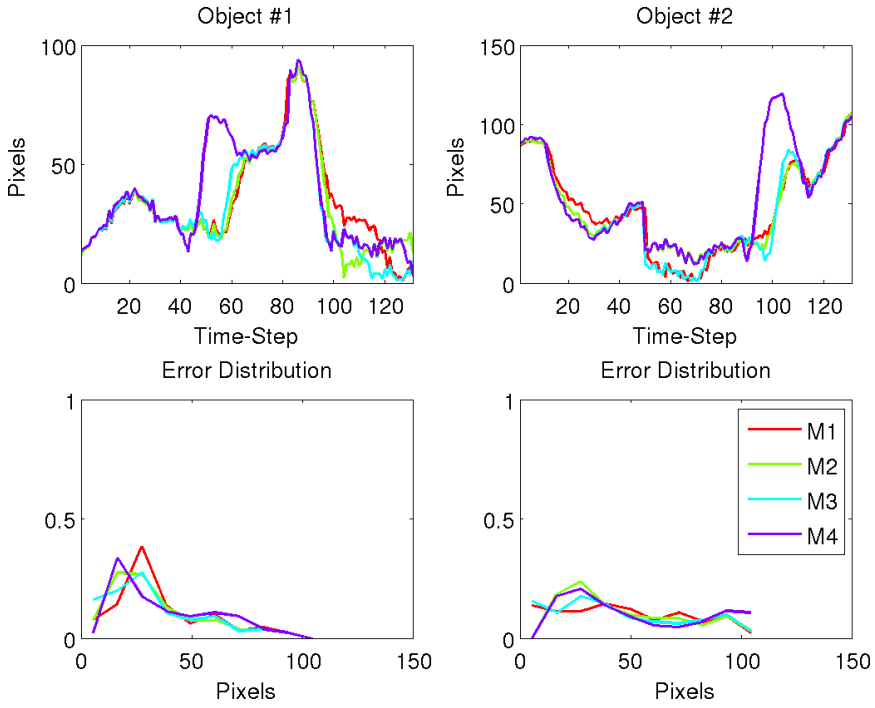


(a) Scenario #1

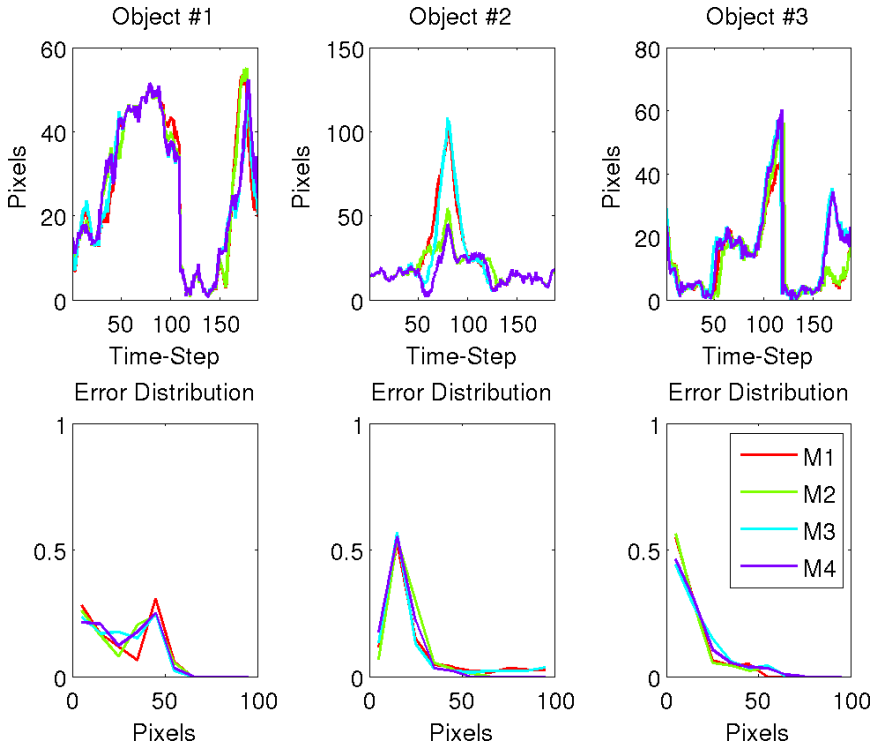


(b) Scenario #2

Figure 4.21: RMSE between ground-truth and estimated object positions with respective error distribution for scenarios 1 and 2. The metric is computed for every object and all methods.



(a) Scenario #3



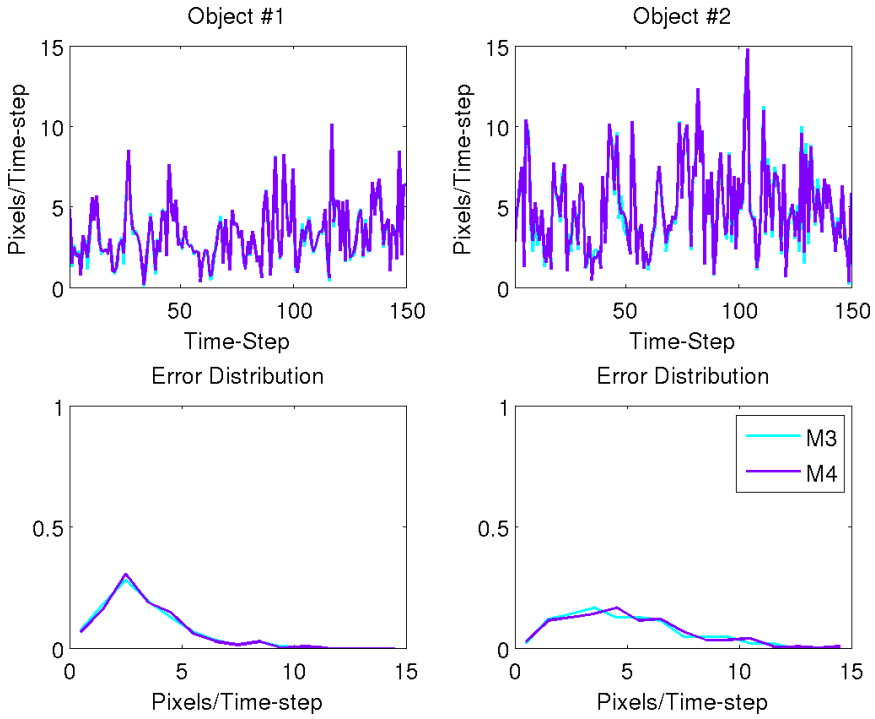
(b) Scenario #4

Figure 4.22: RMSE between ground-truth and estimated object positions with respective error distribution for scenarios 3 and 4. The metric is computed for every object and all methods.

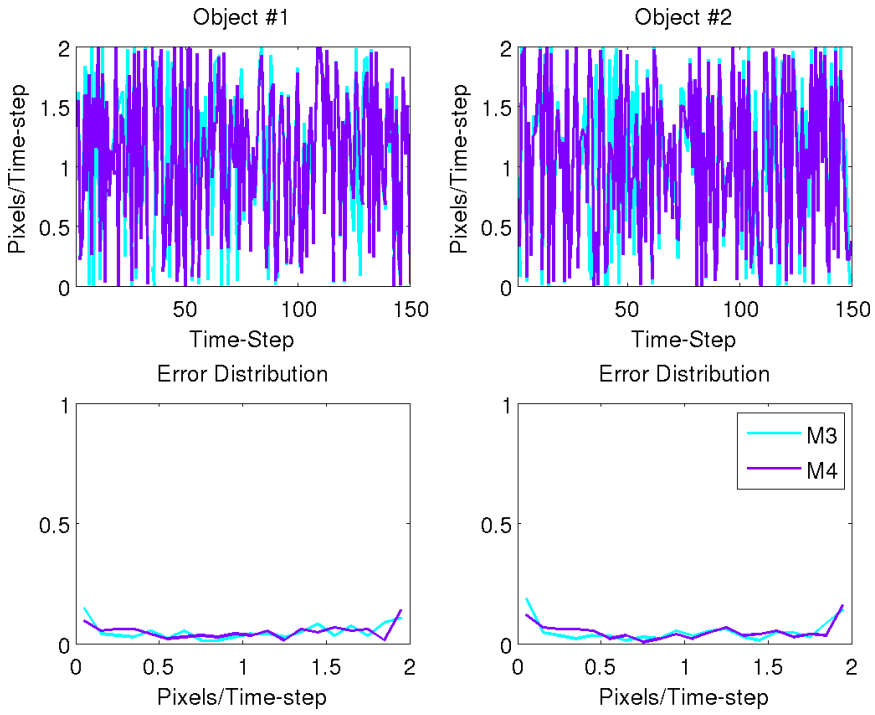
Scenario	Metric	Method 1	Method 2	Method 3	Method 4
#1	mean #1	25.76 ± 13.99	17.63 ± 11.30	25.17 ± 13.53	17.17 ± 10.78
	global mean	25.76 ± 13.99	17.63 ± 11.30	25.17 ± 13.53	17.17 ± 10.78
#2	mean #1	18.12 ± 3.89	9.04 ± 4.23	18.17 ± 3.87	8.76 ± 4.63
	mean #2	14.81 ± 5.41	28.89 ± 9.58	15.00 ± 5.52	30.35 ± 10.03
	global mean	16.46 ± 4.99	18.97 ± 12.39	16.58 ± 5.01	19.56 ± 13.33
#3	mean #1	48.96 ± 29.93	47.05 ± 29.79	45.42 ± 31.77	52.83 ± 31.80
	mean #2	65.08 ± 40.60	66.16 ± 36.99	62.52 ± 42.05	74.71 ± 44.88
	global mean	57.02 ± 36.50	56.61 ± 34.86	53.97 ± 38.17	63.77 ± 40.34
#4	mean #1	36.94 ± 25.04	37.50 ± 24.49	35.95 ± 22.85	36.88 ± 23.10
	mean #2	35.32 ± 31.04	27.78 ± 12.53	33.47 ± 31.51	23.59 ± 11.23
	mean #3	16.89 ± 15.80	17.20 ± 18.03	21.45 ± 19.24	20.63 ± 19.13
	global mean	29.72 ± 26.34	27.49 ± 20.69	30.29 ± 25.82	27.03 ± 19.77

Table 4.7: Mean error and variance of the absolute difference between ground-truth and estimated object position vectors for each scenario. An aggregate mean error is also computed.

As far as concerned the speed RMSE and velocity error, we reported only one graph because all scenarios show a similar behaviour, and we analyse them through Tab. 4.8. As we can see in Fig. 4.23, the errors oscillate a lot for both method 3 and method 4. This is strictly connected to position errors, generating speed and velocity estimations far from the real ones. We can notice that velocity errors reach values equal to two, which means our estimation is totally wrong and in the opposite direction with respect to the ground-truth. Looking at values in Tab. 4.8, we can observe any improvement is provided with the adding of outliers in method 4, and they are higher than results in simulated scenarios, where the errors are smaller than 1.



(a) Velocity RMSE and error distribution per object



(b) Velocity angle error and error distribution per object

Figure 4.23: Scenario 2: (a) RMSE between ground-truth and estimated object velocities with respective error distribution. (b) Velocity angle error. Metrics are computed for every object and all methods.

Scenario	Metric	Method #3	Method #4
#1	mean #1	4.43 ± 2.55	4.64 ± 2.68
#2	mean #1	4.53 ± 2.55	4.60 ± 2.57
	mean #2	6.84 ± 3.85	6.92 ± 3.85
	global mean	5.69 ± 3.46	5.76 ± 3.47
#3	mean #1	8.79 ± 5.65	8.48 ± 4.85
	mean #2	8.28 ± 5.43	9.08 ± 6.15
	global mean	8.53 ± 5.53	8.78 ± 5.53
#4	mean #1	3.98 ± 2.42	4.00 ± 2.42
	mean #2	4.15 ± 2.76	4.13 ± 2.72
	mean #3	3.62 ± 2.27	3.66 ± 2.31
	global mean	3.92 ± 2.50	3.93 ± 2.49

Table 4.8: Mean error and variance of the absolute difference between ground-truth and estimated object velocity vectors for each scenario. An aggregate mean error is also computed.

Evaluation is not limited only to position and velocity estimations, but even to identity and in-box accuracy, which we defined in the beginning.

In Tab. 4.9, we can see the identity accuracy results concerned scenario 3 and scenario 4. Since there is only one person in scenario 1 and no occlusion occurs in scenario 2, the identity switching problem does not appear, showing a perfect identity accuracy for all methods. So, we do not report the results for this two scenarios. An example for scenario 1 is shown in Tab. 4.11 reporting some frames for all methods and depicting the covariance and observations. On the other hand, we can notice that accuracy decreases in scenario 3 because the two people inverted their position in the scene, presenting a small period of occlusion. During and after that moment, an error occurs in the identification decreasing our values of accuracy. However, the two object are strictly connected to each other, as we can see in the same value of accuracy for both. In this case, we can state that a higher value of accuracy depends on the quantity of time that people are sepa-

rated before changing position. Besides, we do not notice any relevant difference among methods. Again, an example of some frames regarding this scenario is reported in Tab. 4.12. Scenario 4 represents a more complex scene involving three people and some occlusions. Performance decreases considerably due to this multiple occlusions. Indeed, the algorithm has not any mechanism to recover from this problem, leading the accuracy to take into account only the initial quantity of time that a person is correctly identified before of an occlusion (except for the case in which another error occurs leading a correct identification again). For instance, in Tab. 4.13, this is clearly visible because the three different coloured clusters are not covering the same people from the beginning to the end. Again, there are no relevant improvements among methods, meaning that the extensions to the algorithm do not affect the performance in terms of identification.

Scenario	Object	Method 1	Method 2	Method 3	Method 4
#3	# 1	0.62	0.63	0.63	0.63
	# 2	0.62	0.63	0.63	0.63
#4	# 1	0.36	0.36	0.38	0.37
	# 2	0.42	0.42	0.42	0.42
	# 3	0.36	0.36	0.38	0.37

Table 4.9: Identity Accuracy

In-box accuracy is slightly different from identity one, because it considers only if the central point of an estimation is inside the box of the associated ground-truth. Indeed, we can notice in Tab. 4.10 that there is a lower performance than identity accuracy in scenario 1; it means that even if the person is correctly identified all the time, the estimated position does not properly rely on the area defined for the ground-truth. However, this is an effect of the used method: lower performances are caused by methods 1 and 3 which do not take outliers into account; methods 2 and 4, instead, reach again the maximum accuracy. So, we can state the clustering affects results for in-box accuracy. Having a look at other more complex

scenarios, which involve more people and occlusions, we see results that are comparable to those related to identity, meaning that occlusion is a relevant problem for this algorithm, independently of the used method and reducing considerably the performance. Again, we can see some examples in Tab. 4.11, Tab. 4.12, and Tab. 4.13: there are various frames in which covariances are not covering correctly or are bigger than the people behind.

Scenario	Object	Method 1	Method 2	Method 3	Method 4
#1	#1	0.89	0.99	0.91	1.00
#2	#1	1.00	1.00	1.00	1.00
	#2	1.00	1.00	1.00	1.00
#3	#1	0.67	0.67	0.69	0.69
	#2	0.55	0.55	0.55	0.45
#4	#1	0.36	0.37	0.39	0.39
	#2	0.29	0.31	0.33	0.40
	#3	0.38	0.38	0.38	0.38

Table 4.10: In-box accuracy.

Table 4.11: Scenario 1. Some frames with the results of clustering for each method in rows.

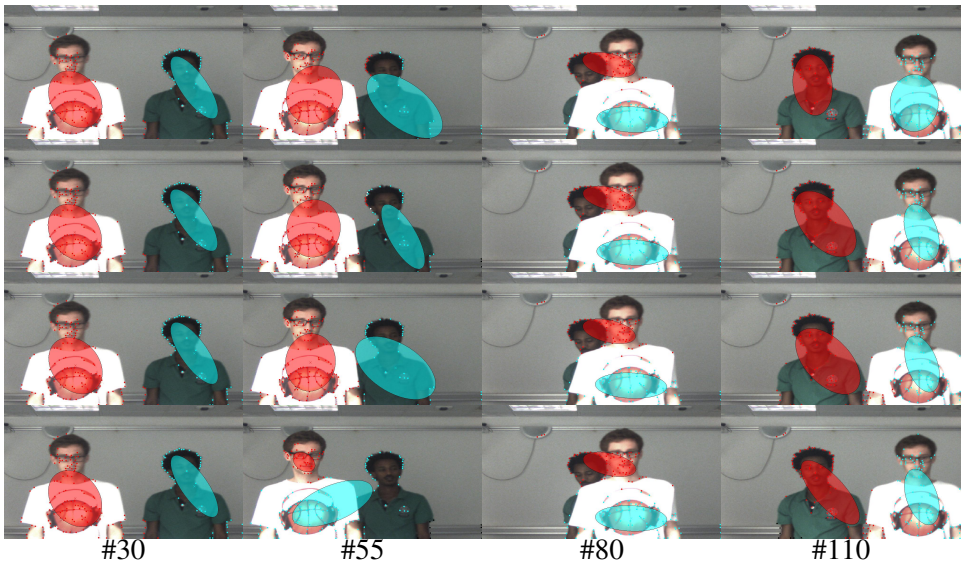


Table 4.12: Scenario 3. Some frames with the results of clustering and tracking for each method per rows.

Table 4.13: Scenario 4. Some frames with the results of clustering and tracking for each method per rows.

As we can see in the errors related to estimated positions, identity and in-box accuracies, the implemented algorithm and its variants do not achieve good results. Since our proposed model does not take into account occlusions, we have

expected these lower performances in real data. Indeed, occlusion is a chaning in the number of object visible in the scene, and this is in contrast to our assumption of fixed number of sources. Nevertheless, the algorithm works well in ideal cases, and improvements are provided with the extensions of outlier and velocity. Limitations and conclusions about the proposed model are described in detail in the next chapter.

Chapter 5

CONCLUSIONS

Achievements The aim of this project is to track multiple sources by means of moving interest points generated by the objects themselves and without knowing which points they generated. Doing so, a probabilistic graphical model is proposed and the Expectation-Maximization algorithm is derived. We found that a closed form solution for the E step of the expected complete-data log-likelihood is not computationally tractable, due to the assignment hidden variable which makes the computation combinatorial. To overcome the problem, we decided to use a variational approach, which lets us factorize the expected complete-data log-likelihood and to determine a closed-form solution. All the derived formulas are provided in appendices.

Thus, thanks to this proposed method we are able: (i) to estimate the object-observation assignments that are not known; (ii) to reconstruct the track of all objects over time; (iii) and, to learn the model's parameters, concerning both the assignment and the dynamics parts. In addition, we modeled outliers and velocity to improve both the observation-to-object association in cluttered scenarios and the estimation of trajectories, and we derived the respectively EM algorithms (positions and outliers, positions and velocities, position and outliers and velocities).

For the testing phase, we used both scenarios with synthetic data and real videos, and the evaluation of performances was carried out for every derived al-

gorithm, comparing them to analyze the improvements.

Limitations However, there are some issues and drawbacks. The main one is the absence of an object identity association method over time, because the model assumes an already existing data association which has to be provided in the initialization of the algorithm. Secondly and in contrast to our assumptions, the incapability to deal with an unknown and variable number of objects over time is a relevant issue to solve in tracking. Then, we had to deal with the convergence of the model's parameters. As we described in a proper section, we noticed an unexpected behaviour in the estimation of Λ, which is the covariance of the dynamics of each source and allows objects to move far away from the current positions. Since we would not allow that, we think to add a prior over the parameter. Moreover, the Gaussian assumption seems to not fit properly with the observations generated from real videos, and it could be interesting to integrate a different distribution into the model. Finally, occlusion is another relevant issue in real videos and we did not model it in our proposed method.

5.1 Future Work

Starting from these considerations, we can illustrate several ways to address the described issues and to improve the method itself. As far as the variable number of objects is concerned, it could be interesting to integrate a birth/death process and investigate the Probability Hypothesis Density filter, which is used in the last decade to solve this particular issue. Obviously, we have to think about a way to integrate the variational approach used in this paper and the framework which the PHD filter relies on. As regards the occlusion, we have already started to analyse the possibility of adding color information into interest points. This can help to keep the identity of each object and to recover from an identity switching problem. Fig. 5.1 shows that color could be useful because the identity of people is kept. A raw test was made. We compared three different approaches to associate

clusters: the first one uses only spatial distance, such as the Mahalanobis distance, between two successive frames, and as we did in our model; the second one uses intersections of the actual histogram of each cluster and its reference, taken as the histogram of the first frame; and, the last is based on a combination of the two with a weight alpha that can give more relevance to one of the two. In the figure, we report the two more interesting instants: before and after the occlusion. While the bar-plot provides the confidence to be associated to each object, the image below of it shows the interest points associated to the person. Using Mahalanobis distance, the confidence of association is high both before and after the occlusion without having two clusters associated to the same object, but showing an identity switching. The color histogram intersection method shows less strong confidence than Mahalanobis, but keeping two complementary associations higher than the others: it means that cluster 1 is associated to object 2 with higher confidence and at the same time, cluster 2 is associated to object 1 with higher confidence to be associated to object 2. In addition, having a look at the images, the object identity is kept. The combination of the two, instead, does not provide a useful result for the moment, so, even though the color looks to be helpful, we need to understand how to integrate correctly the histogram intersection approach in the current one.

Nevertheless, we have to extend the method in order to take into account these color features.

Many ways can be explored by means of moving interest points. Not only color, but also motion information could be embedded, and this allows to both have a better clustering and improve the EM algorithm through the velocity extension. Motion features embed information about direction, and this allows to have no ambiguity between closer points with different motion when clustering.

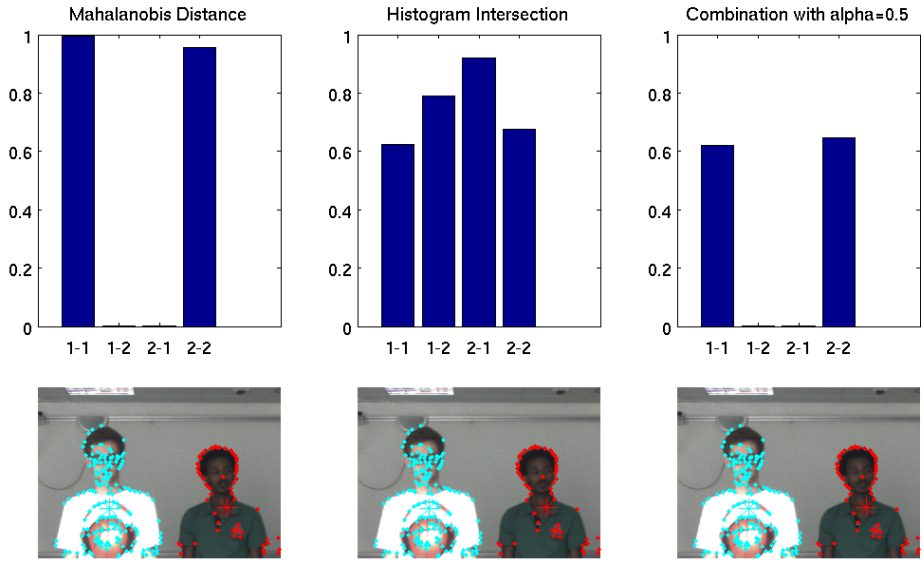
Until now, 2D-points have been used, but we can extend to 3D-points and exploit the depth information, which could be useful to solve problems like occlusion.

As far as the Gaussian distribution problem is concerned, two techniques can be investigated. One could be the convex hull which allows to add again all interest

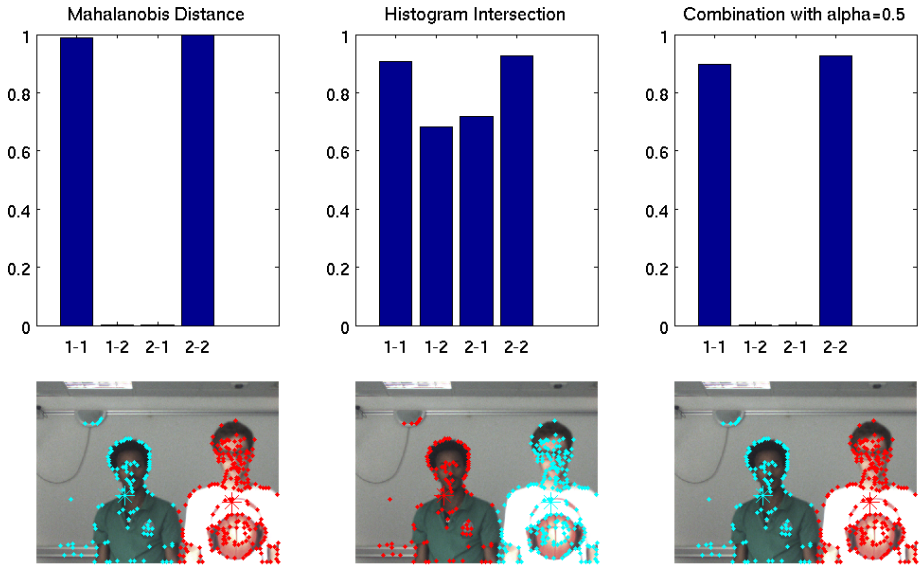
points which rely inside a cluster, but they were removed by the motion filtering. It means that all points could have a visibility parameter which is disabled by the motion filtering, but it could be enabled again after the assignment part, in the case of points are inside the area described by the covariance. This could solve some issues like occlusion or assignment (combined with other solution, such as color features), by having more information about each object. The other one is the regression analysis that allows to learn the relationship of two clusters over time: they refer to the same object, but the generated observations are not the same, resulting in a different covering area; for instance, while interest points could be generated from the whole body of a person in the previous frame, in the current one, an occlusion could start and only a smaller portion of the body, e.g. the head, generates observations. In reality, we know where the center of the cluster should be and adopting a regression technique we can reconstruct the relationship between the two clusters.

Looking at the observations in real scenarios, we noticed the amount of interest points is almost the same over time, except when an occlusion occurs. So, it could be interesting to make an evaluation of the statistics of observations in order to determine the number of objects in the scene.

In the end, it could be nice making the method real-time, either adopting only the forward recursion or using a short sliding window in which the forward-backward recursions can run. Nowadays, cameras can record videos at rate of 25 fps, which permits to slide a window of half or less frames without affecting the online perception.



(a) Before occlusion



(b) After occlusion

Figure 5.1: Comparison of three different approaches for the object association problem between two successive frames before and after the occlusion issue: Mahalanobis distance, histogram intersection, and combination of the two with a weight alpha. The bar-plots represent the confidence of association between the current cluster and the objects.

Appendix A

Derivation of EM: Position

A.1 E-step

Starting from the factorization proposed in chapter 3 for the variational approach:

$$q\left(\boldsymbol{Z},\boldsymbol{P}\right)=\prod_{t,k=1}^{T,K_t}q\left(\boldsymbol{Z}_{tk}\right)\prod_{n=1}^{N}q\left(\boldsymbol{P}_n\right).$$

we now derive the two parts of the expectation step.

A.1.1 E-Z step

The computation of $q\left(\boldsymbol{Z}_{tk}\right)$ is straightforward:

$$\begin{aligned}\eta_{tkm}^{(r+1)}=q\left(Z_{tk}=m\right)&\propto\exp\left(\ln p\left(Z_{tk}=m\right)+\mathrm{E}_{q(\boldsymbol{P}_m)}\left\{\ln p\left(\boldsymbol{f}_{tk}|Z_{tk}=m,\boldsymbol{P}_m,\theta^{(r)}\right)\right\}\right)\\&\propto\pi_{tm}\exp\left(\mathrm{E}_{q(\boldsymbol{P}_{tm})}\left\{\ln\mathcal{N}\left(\boldsymbol{f}_{tk};\boldsymbol{P}_{tm},\boldsymbol{\Sigma}_{tm}^{(r)}\right)\right\}\right)\\&\propto\exp\left[-\ln\left|\boldsymbol{\Sigma}_{tm}^{(r)}\right|^{\frac{1}{2}}-\frac{1}{2}\mathrm{tr}\left(\left(\boldsymbol{\Sigma}_{tm}^{(r)}\right)^{-1}\left(\boldsymbol{\Psi}_{tm}^{(r)}+\boldsymbol{\mu}_{tm}^{(r)}\left(\boldsymbol{\mu}_{tm}^{(r)}\right)^{\top}\right)\right)\right.\\&\qquad\left.-\frac{1}{2}\left(\boldsymbol{f}_{tk}-2\boldsymbol{\mu}_{tm}^{(r)}\right)^{\top}\left(\boldsymbol{\Sigma}_{tm}^{(r)}\right)^{-1}\boldsymbol{f}_{tk}\right]\end{aligned}$$

where we used that $q\left(\boldsymbol{P}_{tm}\right) \sim \mathcal{N}\left(\mu_{tm}^{(r)}, \boldsymbol{\Psi}_{tm}^{(r)}\right)$ (see Section A.1.2). Therefore we write:

$$\eta_{tkm}^{(r+1)} \propto \pi_{tm} \exp\left(-\ln\left|\boldsymbol{\Sigma}_{tm}^{(r)}\right|^{\frac{1}{2}} - \frac{1}{2}\mathrm{tr}\left(\left(\boldsymbol{\Sigma}_{tm}^{(r)}\right)^{-1}\left(\boldsymbol{\Psi}_{tm}^{(r)} + \boldsymbol{\mu}_{tm}^{(r)}\left(\boldsymbol{\mu}_{tm}^{(r)}\right)^{\top}\right)\right)\right.$$
$$\left. -\frac{1}{2}\left(\boldsymbol{f}_{tk} - 2\boldsymbol{\mu}_{tm}^{(r)}\right)^{\top}\left(\boldsymbol{\Sigma}_{tm}^{(r)}\right)^{-1}\boldsymbol{f}_{tk}\right). \tag{A.1}$$

A.1.2 E-P step

In this step we want to compute $q(\boldsymbol{P}_n)$, to further on compute $q(\boldsymbol{P}_{tn})$. Using the proposed factorization we know that:

$$q\left(\boldsymbol{P}_n\right) \propto \exp\left(\mathrm{E}_{q(\boldsymbol{P}_{1:n:N})q(\boldsymbol{Z})}\left\{\ln p\left(\boldsymbol{f}, \boldsymbol{Z}, \boldsymbol{P}|\theta^{(r)}\right)\right\}\right).$$

Let us first analyse term inside the expectation:

$$\ln p\left(\boldsymbol{f}, \boldsymbol{Z}, \boldsymbol{P}|\theta^{(r)}\right) = \ln\left(p\left(\boldsymbol{f}|\boldsymbol{P}, \boldsymbol{Z}, \theta^{(r)}\right) p\left(\boldsymbol{P}|\theta^{(r)}\right) p\left(\boldsymbol{Z}|\theta^{(r)}\right)\right).$$

We notice that the last term does not depend on $\boldsymbol{P}_n$, and can therefore be removed from the computation. Similarly, because $p\left(\boldsymbol{P}\right)$ is separable on n, we can remove all other sources:

$$q\left(\boldsymbol{P}_n\right) \propto p\left(\boldsymbol{P}_n|\theta^{(r)}\right) \exp\left(\mathrm{E}_{q(\boldsymbol{P}_{1:n:N})q(\boldsymbol{Z})}\left\{\ln p\left(\boldsymbol{f}|\boldsymbol{Z}, \boldsymbol{P}, \theta^{(r)}\right)\right\}\right). \tag{A.2}$$

We are now ready to take the expectation:

$$
\begin{aligned}
& \mathrm{E}_{q(\boldsymbol{P}_{1:n:N})q(\boldsymbol{Z})}\left\{\ln p\left(\boldsymbol{f}|\boldsymbol{Z},\boldsymbol{P},\theta^{(r)}\right)\right\} \\
[\boldsymbol{f} \text{ i.i.d}] &= \sum_{t,k=1}^{T,K_t} \mathrm{E}_{q(\boldsymbol{P}_{1:n:N})q(\boldsymbol{Z})}\left\{\ln p\left(\boldsymbol{f}_{tk}|\boldsymbol{Z},\boldsymbol{P},\theta^{(r)}\right)\right\} \\
[\boldsymbol{f}_{tk}|\boldsymbol{P}_t,\boldsymbol{Z}_{tk}.] &= \sum_{t,k=1}^{T,K_t} \mathrm{E}_{q(\boldsymbol{P}_{1:n:N})q(\boldsymbol{Z})}\left\{\ln p\left(\boldsymbol{f}_{tk}|\boldsymbol{Z}_{tk},\boldsymbol{P}_t,\theta^{(r)}\right)\right\} \\
&= \sum_{t,k=1}^{T,K_t} \mathrm{E}_{q(\boldsymbol{P}_{t1:n:N})q(\boldsymbol{Z}_{tk})}\left\{\ln p\left(\boldsymbol{f}_{tk}|\boldsymbol{Z}_{tk},\boldsymbol{P}_t,\theta^{(r)}\right)\right\} \\
&= \sum_{t,k=1}^{T,K_t}\sum_{m=1}^{N} \eta_{tkm}^{(r+1)}\mathrm{E}_{q(\boldsymbol{P}_{t1:n:N})}\left\{\ln p\left(\boldsymbol{f}_{tk}|\boldsymbol{Z}_{tk}=m,\boldsymbol{P}_t,\theta^{(r)}\right)\right\} \\
&= \sum_{t,k=1}^{T,K_t}\sum_{m=1}^{N} \eta_{tkm}^{(r+1)}\mathrm{E}_{q(\boldsymbol{P}_{t1:n:N})}\left\{\ln p\left(\boldsymbol{f}_{tk}|\boldsymbol{Z}_{tk}=m,\boldsymbol{P}_{tm},\theta^{(r)}\right)\right\} \\
&= \sum_{t,k=1}^{T,K_t}\sum_{m=1,m\neq n}^{N} \eta_{tkm}^{(r+1)}\mathrm{E}_{q(\boldsymbol{P}_{tm})}\left\{\ln p\left(\boldsymbol{f}_{tk}|\boldsymbol{Z}_{tk}=m,\boldsymbol{P}_{tm},\theta^{(r)}\right)\right\} \\
&\qquad + \sum_{t,k=1}^{T,K_t} \eta_{tkn}^{(r+1)}\ln p\left(\boldsymbol{f}_{tk}|\boldsymbol{Z}_{tk}=n,\boldsymbol{P}_{tn},\theta^{(r)}\right) \\
&\overset{\boldsymbol{P}_n}{=} \sum_{t,k=1}^{T,K_t} \eta_{tkn}^{(r+1)}\ln\mathcal{N}\left(\boldsymbol{f}_{tk};\boldsymbol{P}_{tn},\boldsymbol{\Sigma}_{tn}^{(r)}\right) \\
&\overset{\boldsymbol{P}_n}{=} \sum_{t,k=1}^{T,K_t} \ln\mathcal{N}\left(\boldsymbol{f}_{tk};\boldsymbol{P}_{tn},\frac{\boldsymbol{\Sigma}_{tn}^{(r)}}{\eta_{tkn}^{(r+1)}}\right)
\end{aligned}
\tag{A.3}
$$

Replacing the previous result into (A.2) we write:

$$
\begin{aligned}
q\left(\boldsymbol{P}_n\right) \propto \quad & \mathcal{N}\left(\boldsymbol{P}_{1n};\boldsymbol{\nu}_n^{(r)},\boldsymbol{\Omega}_n^{(r)}\right)\prod_{t=2}^{T}\mathcal{N}\left(\boldsymbol{P}_{tn};\boldsymbol{P}_{t-1n},\boldsymbol{\Lambda}_n^{(r)}\right)\times \\
& \times\prod_{t=1}^{T}\prod_{k=1}^{K_t}\mathcal{N}\left(\boldsymbol{f}_{tk};\boldsymbol{P}_{tn},\frac{\boldsymbol{\Sigma}_{tn}^{(r)}}{\eta_{tkn}^{(r+1)}}\right)
\end{aligned}
\tag{A.4}
$$

We also notice that:

$$
\prod_{k=1}^{K_t}\mathcal{N}\left(\boldsymbol{f}_{tk};\boldsymbol{P}_{tn},\frac{\boldsymbol{\Sigma}_n^{(r)}}{\eta_{tkn}^{(r+1)}}\right) \overset{\boldsymbol{P}_n}{\propto} \mathcal{N}\left(\boldsymbol{P}_{tn};\boldsymbol{\mu}_{tn}^{\iota(r+1)},\boldsymbol{\Psi}_{tn}^{\iota(r+1)}\right),
$$

with

$$\boldsymbol{\mu}_{tn}^{\iota(r+1)} = \frac{\sum_{k=1}^{K_t} \eta_{tkn}^{(r+1)} \boldsymbol{f}_{tk}}{\sum_{k=1}^{K_t} \eta_{tkn}^{(r+1)}} \qquad \boldsymbol{\Psi}_{tn}^{\iota(r+1)} = \left(\sum_{k=1}^{K_t} \eta_{tkn}^{(r+1)}\right)^{-1} \boldsymbol{\Sigma}_{tn}^{(r)}.$$

Therefore, (A.4) rewrites:

$$\begin{aligned} q\left(\boldsymbol{P}_n\right) \propto \quad & \mathcal{N}\left(\boldsymbol{P}_{1n}; \boldsymbol{\nu}_n^{(r)}, \boldsymbol{\Omega}_n^{(r)}\right) \prod_{t=2}^{T} \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{P}_{t-1n}, \boldsymbol{\Lambda}_n^{(r)}\right) \times \\ & \times \prod_{t=1}^{T} \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{\mu}_{tn}^{\iota(r+1)}, \boldsymbol{\Psi}_{tn}^{\iota(r+1)}\right). \end{aligned} \tag{A.5}$$

At the light of this result, we are certain that $q(\boldsymbol{P}_n)$ is a multivariate normal. Because of the temporal term, it is not obvious to extract $q(\boldsymbol{P}_{tn})$ from here. Direct marginalization will naturally lead to forward-backward recursions.

The marginal distribution - forward/backward Let us compute $q(\boldsymbol{P}_{tn})$, by marginalizing (A.5):

$$\begin{aligned} q(\boldsymbol{P}_{tn}) \propto \quad & \int \mathcal{N}\left(\boldsymbol{P}_{1n}; \boldsymbol{\nu}_n^{(r)}, \boldsymbol{\Omega}_n^{(r)}\right) \prod_{s=2}^{T} \mathcal{N}\left(\boldsymbol{P}_{sn}; \boldsymbol{P}_{s-1n}, \boldsymbol{\Lambda}_n^{(r)}\right) \times \\ & \times \prod_{s=1}^{T} \mathcal{N}\left(\boldsymbol{P}_{sn}; \boldsymbol{\mu}_{sn}^{\iota(r+1)}, \boldsymbol{\Psi}_{sn}^{\iota(r+1)}\right) \mathrm{d}\boldsymbol{P}_{1:t:Tn} \end{aligned}$$

We can rewrite this marginalization into a product of two marginalizations representing the *past* and the *future* with respect to $\boldsymbol{P}_{tn}$:

$$\begin{aligned} q(\boldsymbol{P}_{tn}) \propto \quad & \int \mathcal{N}\left(\boldsymbol{P}_{1n}; \boldsymbol{\nu}_n^{(r)}, \boldsymbol{\Omega}_n^{(r)}\right) \prod_{s=2}^{t} \mathcal{N}\left(\boldsymbol{P}_{sn}; \boldsymbol{P}_{s-1n}, \boldsymbol{\Lambda}_n^{(r)}\right) \times \\ & \times \prod_{s=1}^{t} \mathcal{N}\left(\boldsymbol{P}_{sn}; \boldsymbol{\mu}_{sn}^{\iota(r+1)}, \boldsymbol{\Psi}_{sn}^{\iota(r+1)}\right) \mathrm{d}\boldsymbol{P}_{1:t-1n} \end{aligned} \tag{A.6}$$

$$\int \prod_{s=t+1}^{T} \mathcal{N}\left(\boldsymbol{P}_{sn}; \boldsymbol{P}_{s-1n}, \boldsymbol{\Lambda}_n^{(r)}\right) \prod_{s=t+1}^{T} \mathcal{N}\left(\boldsymbol{P}_{sn}; \boldsymbol{\mu}_{sn}^{\iota(r+1)}, \boldsymbol{\Psi}_{sn}^{\iota(r+1)}\right) \mathrm{d}\boldsymbol{P}_{t+1:Tn} \tag{A.7}$$

The quantity on (A.6) will be denoted $\phi\left(\boldsymbol{P}_{tn}\right)$ and the one on (A.7), $\beta\left(\boldsymbol{P}_{tn}\right)$. With this notation we can get forward-backward recursions for ϕ and β:

$$\begin{aligned}\phi\left(\boldsymbol{P}_{tn}\right) = & \quad \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{\mu}_{tn}^{\iota(r+1)}, \boldsymbol{\Psi}_{tn}^{\iota(r+1)}\right) \int \phi\left(\boldsymbol{P}_{t-1n}\right) \times \\ & \times \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{P}_{t-1n}, \boldsymbol{\Lambda}_{n}^{(r)}\right) \mathrm{d}\boldsymbol{P}_{t-1n} \qquad \text{(A.8)} \\ \beta\left(\boldsymbol{P}_{tn}\right) = & \int \beta\left(\boldsymbol{P}_{t+1n}\right) \mathcal{N}\left(\boldsymbol{P}_{t+1n}; \boldsymbol{P}_{tn}, \boldsymbol{\Lambda}_{n}^{(r)}\right) \times \\ & \times \mathcal{N}\left(\boldsymbol{P}_{t+1n}; \boldsymbol{\mu}_{t+1n}^{\iota(r+1)}, \boldsymbol{\Psi}_{t+1n}^{\iota(r+1)}\right) \mathrm{d}\boldsymbol{P}_{t+1n} \qquad \text{(A.9)}\end{aligned}$$

We will now compute the recursions for ϕ and β. Let us assume that $\phi\left(\boldsymbol{P}_{t-1n}\right)$ and $\beta\left(\boldsymbol{P}_{t+1n}\right)$ are Gaussian and compute $\phi\left(\boldsymbol{P}_{tn}\right)$ and $\beta\left(\boldsymbol{P}_{tn}\right)$ from them. Therefore, we set

$$\begin{aligned}\phi\left(\boldsymbol{P}_{t-1n}\right) &= \mathcal{N}\left(\boldsymbol{P}_{t-1n}; \boldsymbol{\mu}_{t-1n}^{\phi(r+1)}, \boldsymbol{\Psi}_{t-1n}^{\phi(r+1)}\right) \\ \beta\left(\boldsymbol{P}_{t+1n}\right) &= \mathcal{N}\left(\boldsymbol{P}_{t+1n}; \boldsymbol{\mu}_{t+1n}^{\beta(r+1)}, \boldsymbol{\Psi}_{t+1n}^{\beta(r+1)}\right)\end{aligned}$$

Forward recursion

$$\begin{aligned}\phi(\boldsymbol{P}_{t,n}) \propto & \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{\mu}_{tn}^{\iota(r+1)}, \boldsymbol{\Psi}_{tn}^{\iota(r+1)}\right) \int \mathcal{N}\left(\boldsymbol{P}_{t-1n}; \boldsymbol{\mu}_{t-1n}^{\phi(r+1)}, \boldsymbol{\Psi}_{t-1n}^{\phi(r+1)}\right) \times \\ & \times \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{P}_{t-1n}, \boldsymbol{\Lambda}_{n}^{(r)}\right) \mathrm{d}\boldsymbol{P}_{t-1,n} \\ \propto & \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{\mu}_{tn}^{\iota(r+1)}, \boldsymbol{\Psi}_{tn}^{\iota(r+1)}\right) \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{\mu}_{t-1n}^{\phi(r+1)}, \boldsymbol{\Psi}_{t-1n}^{\phi(r+1)} + \boldsymbol{\Lambda}_{n}^{(r)}\right) \\ \propto & \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{\mu}_{tn}^{\phi(r+1)}, \boldsymbol{\Psi}_{tn}^{\phi(r+1)}\right) \qquad \text{(A.10)}\end{aligned}$$

where

$$\left(\boldsymbol{\Psi}_{tn}^{\phi(r+1)}\right)^{-1} = \left(\boldsymbol{\Psi}_{tn}^{\iota(r+1)}\right)^{-1} + \left(\boldsymbol{\Psi}_{t-1n}^{\phi(r+1)} + \boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1} \qquad \text{(A.11)}$$

$$\boldsymbol{\mu}_{tn}^{\phi(r+1)} = \boldsymbol{\Psi}_{tn}^{\phi(r+1)} \left(\left(\boldsymbol{\Psi}_{tn}^{\iota(r+1)}\right)^{-1} \boldsymbol{\mu}_{tn}^{\iota(r+1)} + \left(\boldsymbol{\Psi}_{t-1n}^{\phi(r+1)} + \boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1} \boldsymbol{\mu}_{t-1n}^{\phi(r+1)}\right) \qquad \text{(A.12)}$$

Backward recursion

$$\begin{aligned}\beta\left(\boldsymbol{P}_{tn}\right) \propto & \int \mathcal{N}\left(\boldsymbol{P}_{t+1n}; \boldsymbol{\mu}_{t+1n}^{\beta(r+1)}, \boldsymbol{\Psi}_{t+1n}^{\beta(r+1)}\right) \mathcal{N}\left(\boldsymbol{P}_{t+1n}; \boldsymbol{P}_{tn}, \boldsymbol{\Lambda}_{n}^{(r)}\right) \times \\ & \times \mathcal{N}\left(\boldsymbol{P}_{t+1n}; \boldsymbol{\mu}_{t+1n}^{\iota(r+1)}, \boldsymbol{\Psi}_{t+1n}^{\iota(r+1)}\right) \mathrm{d}\boldsymbol{P}_{t+1n} \\ \propto & \int \mathcal{N}\left(\boldsymbol{P}_{t+1n}; \boldsymbol{m}, \boldsymbol{\Phi}\right) \exp\left(-\frac{1}{2}\left(\left(-\left\|\boldsymbol{m}\right\|_{\boldsymbol{\Phi}} + \left\|\boldsymbol{\mu}_{t+1n}^{\beta(r+1)}\right\|_{\boldsymbol{\Psi}_{t+1n}^{\beta(r+1)}}\right.\right.\right. \\ & \left.\left.\left. + \left\|\boldsymbol{P}_{tn}\right\|_{\boldsymbol{\Lambda}_{n}^{(r)}} + \left\|\boldsymbol{\mu}_{t+1n}^{\iota(r)}\right\|_{\boldsymbol{\Psi}_{t+1n}^{\iota(r+1)}}\right)\right) \mathrm{d}\boldsymbol{P}_{t+1n} \\ \propto & \exp\left(-\frac{1}{2}\left(-\left\|\boldsymbol{m}\right\|_{\boldsymbol{\Phi}} + \left\|\boldsymbol{\mu}_{t+1n}^{\beta(r+1)}\right\|_{\boldsymbol{\Psi}_{t+1n}^{\beta(r+1)}}\right.\right. \\ & \left.\left. + \left\|\boldsymbol{P}_{tn}\right\|_{\boldsymbol{\Lambda}_{n}^{(r)}} + \left\|\boldsymbol{\mu}_{t+1n}^{\iota(r)}\right\|_{\boldsymbol{\Psi}_{t+1n}^{\iota(r+1)}}\right)\right)\end{aligned}$$

with

$$\boldsymbol{\Phi}^{-1} = \left(\boldsymbol{\Psi}_{t+1n}^{\beta(r+1)}\right)^{-1} + \left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1} + \left(\boldsymbol{\Psi}_{t+1n}^{\iota(r+1)}\right)^{-1}$$

$$\boldsymbol{m} = \boldsymbol{\Phi}\left(\left(\boldsymbol{\Psi}_{t+1n}^{\beta(r+1)}\right)^{-1} \boldsymbol{\mu}_{t+1n}^{\beta(r+1)} + \left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1} \boldsymbol{P}_{tn} + \left(\boldsymbol{\Psi}_{t+1n}^{\iota(r)}\right)^{-1} \boldsymbol{\mu}_{t+1n}^{\iota(r+1)}\right)$$

The terms of the exponential depending on $\boldsymbol{P}_{tn}$ are two (we do not take the $-\frac{1}{2}$ into account): the quadratic term

$$\left(\boldsymbol{P}_{tn}\right)^{\top}\left(\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1} - \left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1} \boldsymbol{\Phi}\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}\right) \boldsymbol{P}_{tn},$$

and the linear term

$$-2\left(\boldsymbol{P}_{tn}\right)^{\top}\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1} \boldsymbol{\Phi}\left(\left(\boldsymbol{\Psi}_{t+1n}^{\beta(r+1)}\right)^{-1} \boldsymbol{\mu}_{t+1n}^{\beta(r+1)} + \left(\boldsymbol{\Psi}_{t+1n}^{\iota(r+1)}\right)^{-1} \boldsymbol{\mu}_{t+1n}^{\iota(r+1)}\right).$$

The quadratic term sets the covariance matrix of the distribution:

$$\boldsymbol{\Psi}_{tn}^{\beta(r+1)} = \boldsymbol{\Lambda}_{n}^{(r)} + \left(\left(\boldsymbol{\Psi}_{t+1n}^{\beta(r+1)}\right)^{-1} + \left(\boldsymbol{\Psi}_{t+1n}^{\iota(r+1)}\right)^{-1}\right)^{-1} \tag{A.13}$$

Together with the linear term, we set the mean to:

$$\boldsymbol{\mu}_{tn}^{\beta(r+1)} = \boldsymbol{\Psi}_{tn}^{\beta(r+1)}\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1} \boldsymbol{\Phi}\left(\left(\boldsymbol{\Psi}_{t+1n}^{\beta(r+1)}\right)^{-1} \boldsymbol{\mu}_{t+1n}^{\beta(r+1)} + \left(\boldsymbol{\Psi}_{t+1n}^{\iota(r)}\right)^{-1} \boldsymbol{\mu}_{t+1n}^{\iota(r+1)}\right)$$

which can be simplified down to:

$$\boldsymbol{\mu}_{tn}^{\beta(r+1)} = \left(\left(\boldsymbol{\Psi}_{t+1n}^{\beta(r+1)} \right)^{-1} + \left(\boldsymbol{\Psi}_{t+1n}^{\iota(r+1)} \right)^{-1} \right)^{-1} \left(\left(\boldsymbol{\Psi}_{t+1n}^{\beta(r+1)} \right)^{-1} \boldsymbol{\mu}_{t+1n}^{\beta(r+1)} + \left(\boldsymbol{\Psi}_{t+1n}^{\iota(r+1)} \right)^{-1} \boldsymbol{\mu}_{t+1n}^{\iota(r+1)} \right) \tag{A.14}$$

Thus obtaining:

$$\beta\left(\boldsymbol{P}_{tn}\right) \propto \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{\mu}_{tn}^{\beta(r+1)}, \boldsymbol{\Psi}_{tn}^{\beta(r+1)}\right) \tag{A.15}$$

The final marginal distribution By combining the forward and backward recursions we get:

$$\begin{aligned} q\left(\boldsymbol{P}_{tn}\right) &\propto \phi\left(\boldsymbol{P}_{tn}\right) \beta\left(\boldsymbol{P}_{tn}\right) \\ &\propto \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{\mu}_{tn}^{\phi(r+1)}, \boldsymbol{\Psi}_{tn}^{\phi(r+1)}\right) \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{\mu}_{tn}^{\beta(r+1)}, \boldsymbol{\Psi}_{tn}^{\beta(r+1)}\right) \end{aligned}$$

which leads to:

$$\boxed{q\left(\boldsymbol{P}_{tn}\right) = \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{\mu}_{tn}^{(r+1)}, \boldsymbol{\Psi}_{tn}^{(r+1)}\right)} \tag{A.16}$$

with

$$\left(\boldsymbol{\Psi}_{tn}^{(r+1)}\right)^{-1} = \left(\boldsymbol{\Psi}_{tn}^{\beta(r+1)}\right)^{-1} + \left(\boldsymbol{\Psi}_{tn}^{\phi(r+1)}\right)^{-1} \tag{A.17}$$

$$\boldsymbol{\mu}_{tn}^{(r+1)} = \boldsymbol{\Psi}_{tn}^{(r+1)} \left(\left(\boldsymbol{\Psi}_{tn}^{\beta(r+1)}\right)^{-1} \boldsymbol{\mu}_{tn}^{\beta(r+1)} + \left(\boldsymbol{\Psi}_{tn}^{\phi(r+1)}\right)^{-1} \boldsymbol{\mu}_{tn}^{\phi(r+1)} \right) \tag{A.18}$$

The coupled joint distribution When dealing with the M step, we will need the distribution $q\left(\boldsymbol{P}_{tn}, \boldsymbol{P}_{t-1n}\right)$ for the estimation of $\boldsymbol{\Lambda}_n$. If we marginalize (A.5) with respect to all variables except $\boldsymbol{P}_{tn}$ and $\boldsymbol{P}_{t-1n}$, we obtain the well-known formula:

$$\begin{aligned} q\left(\boldsymbol{P}_{tn}, \boldsymbol{P}_{t-1n}\right) &\propto \phi\left(\boldsymbol{P}_{t-1n}\right) \iota\left(\boldsymbol{P}_{tn}\right) T\left(\boldsymbol{P}_{tn}, \boldsymbol{P}_{t-1n}\right) \beta\left(\boldsymbol{P}_{tn}\right) \\ &\propto \mathcal{N}\left(\boldsymbol{P}_{t-1n}; \boldsymbol{\mu}_{t-1n}^{\phi(r+1)}, \boldsymbol{\Psi}_{t-1n}^{\phi(r+1)}\right) \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{\mu}_{tn}^{\iota(r+1)}, \boldsymbol{\Psi}_{tn}^{\iota(r+1)}\right) \times \\ &\quad \times \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{P}_{t-1n}, \boldsymbol{\Lambda}_n^{(r)}\right) \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{\mu}_{tn}^{\beta(r+1)}, \boldsymbol{\Psi}_{tn}^{\beta(r+1)}\right) \end{aligned}$$

If we develop the terms in the exponential, we realize that the quadratic term is:

$$\boldsymbol{P}_{t-1n}^{\top}\left(\left(\boldsymbol{\Psi}_{t-1n}^{\phi(r)}\right)^{-1}+\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}\right)\boldsymbol{P}_{t-1n}+\boldsymbol{P}_{tn}^{\top}\left(\left(\boldsymbol{\Psi}_{tn}^{\iota(r+1)}\right)^{-1}+\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}+\left(\boldsymbol{\Psi}_{tn}^{\beta(r+1)}\right)^{-1}\right)\boldsymbol{P}_{tn}-2\boldsymbol{P}_{t-1n}^{\top}\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}\boldsymbol{P}_{tn}$$

leading to the following covariance matrix for the joint distribution:

$$\boldsymbol{\Psi}_{tn}^{\zeta(r+1)}=\begin{pmatrix}\left(\boldsymbol{\Psi}_{tn}^{\iota(r+1)}\right)^{-1}+\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}+\left(\boldsymbol{\Psi}_{tn}^{\beta(r+1)}\right)^{-1} & -\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}\\ -\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1} & \left(\boldsymbol{\Psi}_{t-1n}^{\phi(r)}\right)^{-1}+\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}\end{pmatrix}^{-1} \tag{A.19}$$

The linear term is:

$$-2\left(\boldsymbol{\mu}_{t-1n}^{\phi(r+1)}\right)^{\top}\left(\boldsymbol{\Psi}_{t-1n}^{\phi(r+1)}\right)^{-1}\boldsymbol{P}_{t-1n}-2\left(\left(\boldsymbol{\mu}_{tn}^{\beta(r+1)}\right)^{\top}\left(\boldsymbol{\Psi}_{tn}^{\beta(r+1)}\right)^{-1}+\left(\boldsymbol{\mu}_{tn}^{\iota(r+1)}\right)^{\top}\left(\boldsymbol{\Psi}_{tn}^{\iota(r+1)}\right)^{-1}\right)\boldsymbol{P}_{tn}$$

and therefore set the mean of the joint distribution as:

$$\boldsymbol{\mu}_{tn}^{\zeta(r+1)}=\boldsymbol{\Psi}_{tn}^{\zeta(r+1)}\begin{pmatrix}\left(\boldsymbol{\Psi}_{tn}^{\beta(r+1)}\right)^{-1}\boldsymbol{\mu}_{tn}^{\beta(r+1)}+\left(\boldsymbol{\Psi}_{tn}^{\iota(r+1)}\right)^{-1}\boldsymbol{\mu}_{tn}^{\iota(r+1)}\\ \left(\boldsymbol{\Psi}_{t-1n}^{\phi(r+1)}\right)^{-1}\boldsymbol{\mu}_{t-1n}^{\phi(r+1)}\end{pmatrix} \tag{A.20}$$

A.2 M step

The parameter π

$$\begin{aligned}\mathcal{Q}\left(\pi,\pi^{(r)}\right)&=\sum_{t,k=1}^{T,K_t}\mathrm{E}_{q(\boldsymbol{Z}_{tk})}\left\{\ln p\left(\boldsymbol{Z}_{tk}|\pi_{tn}\right)\right\}\\&=\sum_{t,k=1}^{T,K_t}\mathrm{E}_{q(\boldsymbol{Z}_{tk})}\left\{\ln\prod_{n=1}^{N}\pi_{tn}^{\boldsymbol{Z}_{tkn}}\right\}\\&=\sum_{t,k=1}^{T,K_t}\sum_{n=1}^{N}\mathrm{E}_{q(\boldsymbol{Z}_{tk})}\left\{\boldsymbol{Z}_{tkn}\right\}\ln\pi_{tn}\\&=\sum_{t,k=1}^{T,K_t}\sum_{n=1}^{N}\eta_{tkn}^{\boldsymbol{Z}(r+1)}\ln\pi_{tn}\end{aligned}$$

Now, we can choose between having equal or different parameters for each time step and, in general, it depends on observations. So, we set the derivative of the above function with respect to π to zero and rearranging we obtain the formula for the parameter.

First case (different parameters):

$$\frac{\partial \mathcal{Q}\left(\pi, \pi^{(r)}\right)}{\partial \pi_{tn}} = \sum_{k=1}^{K_t} \eta_{tkn}^{\boldsymbol{Z}(r+1)} \frac{1}{\pi_{tn}} = 0$$

and we obtain,

$$\boxed{\pi_{tn} = \frac{\sum_{k=1}^{K_t} \eta_{tkn}^{\boldsymbol{Z}(r+1)}}{K_t}} \tag{A.21}$$

Second case (equal parameters):

$$\frac{\partial \mathcal{Q}\left(\pi, \pi^{(r)}\right)}{\partial \pi_{n}} = \sum_{k,t=1}^{K_t,T} \eta_{tkn}^{\boldsymbol{Z}(r+1)} \frac{1}{\pi_{n}} = 0$$

and we obtain,

$$\boxed{\pi_{n} = \frac{\sum_{k,t=1}^{K_t,T} \eta_{tkn}^{\boldsymbol{Z}(r+1)}}{\sum_{t=1}^{T} K_t}} \tag{A.22}$$

The parameter $\boldsymbol{\Sigma}$ Secondly, we take into into account the second term of the developed Q-function. So, we define:

$$\begin{aligned}
\mathcal{Q}\left(\boldsymbol{\Sigma}, \Psi^{(r)}\right) &= \sum_{t,k=1}^{T,K_t} \mathrm{E}_{q(\boldsymbol{Z}_t k) q(\boldsymbol{P}_t)} \left\{\ln p\left(\boldsymbol{f}_{tk} | \boldsymbol{Z}_{tk}, \boldsymbol{P}_t, \boldsymbol{\Sigma}\right)\right\} \\
&= \sum_{t,k=1}^{T,K_t} \mathrm{E}_{q(\boldsymbol{P}_t)} \left\{ \sum_{m=1}^{N} \eta_{tkm} \ln \mathcal{N}\left(\boldsymbol{f}_{tk}; \boldsymbol{Z}_{tk}, \boldsymbol{P}_{tm}, \boldsymbol{\Sigma}_{tm}\right) \right\} \\
&= \sum_{m=1}^{N} \sum_{t,k=1}^{T,K_t} \eta_{tkm} \mathrm{E}_{q(\boldsymbol{P}_{tm})} \left\{\ln \mathcal{N}\left(\boldsymbol{P}_{tm}; \boldsymbol{Z}_{tk}, \boldsymbol{f}_{tk}, \boldsymbol{\Sigma}_{tm}\right)\right\}
\end{aligned}$$

and develop it,

$$\mathcal{Q}\left(\boldsymbol{\Sigma}_n, \Psi^{(r)}\right) = \sum_{t,k=1}^{T,K_t} \eta_{tkn} \left[\frac{1}{2}\ln|\boldsymbol{\Sigma}_{tn}^{-1}| - \frac{1}{2}\left(tr\left(\boldsymbol{\Sigma}_{tn}^{-1}\left(\boldsymbol{\Psi}_{tn} + \boldsymbol{\mu}_{tm}\boldsymbol{\mu}_{tn}^{T}\right)\right)\right.\right.$$
$$\left.\left. -2\boldsymbol{\mu}_{tn}^{\top}\boldsymbol{\Sigma}_{tn}^{-1}\boldsymbol{f}_{tk} + \boldsymbol{f}_{tn}\boldsymbol{\Sigma}_{tn}^{-1}\boldsymbol{f}_{tk}\right)\right]$$

It is worth to notice that the $\boldsymbol{\Sigma}$ is a symmetric matrix and before taking the derivative, we need to split the term $-2\boldsymbol{\mu}_{tn}^{\top}\boldsymbol{\Sigma}_{tn}^{-1}\boldsymbol{f}_{tk}$ in order to obtain the correct one later. Now, we take the derivative and set it to zero:

$$\frac{\partial\mathcal{Q}_{\boldsymbol{\Sigma}_n,\boldsymbol{\Psi}^{(r)}}}{\partial\boldsymbol{\Sigma}_n^{-1}} = \frac{1}{2}\sum_{k=1}^{K_t}\eta_{ktn}\left[\boldsymbol{\Sigma}_{tn} - \left(\boldsymbol{\Psi}_{tn} + \boldsymbol{\mu}_{tn}\boldsymbol{\mu}_{tn}^{\top} - \boldsymbol{\mu}_{tn}^{\top}\boldsymbol{f}_{tk} - \boldsymbol{f}_{tk}\boldsymbol{\mu}_{tn}^{\top}\right.\right.$$
$$\left.\left. +\boldsymbol{f}_{tk}\boldsymbol{f}_{tk}^{\top}\right)\right] = 0$$

Rearranging:

$$\boxed{\boldsymbol{\Sigma}_{tn}^{(r+1)} = \boldsymbol{\Psi}_{tn} + \frac{\sum_{k=1}^{K_t}\eta_{ktn}\left(\boldsymbol{\mu}_{tn} - \boldsymbol{f}_{tk}\right)\left(\boldsymbol{\mu}_{tn} - \boldsymbol{f}_{tk}\right)^{\top}}{\sum_{k=1}^{K_t}\eta_{ktn}}} \tag{A.23}$$

or

$$\boxed{\boldsymbol{\Sigma}_{n}^{(r+1)} = \frac{\sum_{t,k=1}^{T,K_t}\eta_{ktn}\left[\boldsymbol{\Psi}_{tn} + \left(\boldsymbol{\mu}_{tn} - \boldsymbol{f}_{tk}\right)\left(\boldsymbol{\mu}_{tn} - \boldsymbol{f}_{tk}\right)^{\top}\right]}{\sum_{t,k=1}^{T,K_t}\eta_{ktn}}} \tag{A.24}$$

The parameters $\mathbf{\Lambda}_n$ We will now derive the update equation for $\mathbf{\Lambda}_n$. The terms of the $\mathcal{Q}$ function related to the transition matrix are:

$$\begin{aligned}
\mathcal{Q}_{\mathbf{\Lambda}_n} &= \sum_{t=2}^{T} \mathrm{E}_{q(\boldsymbol{P}_{tn}, \boldsymbol{P}_{t-1n})} \left\{ \ln p\left(\boldsymbol{P}_{tn} | \boldsymbol{P}_{t-1n}, \theta\right) \right\} \\
&= \sum_{t=2}^{T} \mathrm{E}_{q(\boldsymbol{P}_{tn}, \boldsymbol{P}_{t-1n})} \left\{ \ln \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{P}_{t-1n}, \mathbf{\Lambda}_n\right) \right\} \\
&\overset{\mathbf{\Lambda}_n}{=} \sum_{t=2}^{T} \mathrm{E}_{q(\boldsymbol{P}_{tn}, \boldsymbol{P}_{t-1n})} \left\{ \ln \left|\mathbf{\Lambda}_n\right|^{-1/2} - \left\| \boldsymbol{P}_{tn} - \boldsymbol{P}_{t-1n} \right\|_{\mathbf{\Lambda}_n} \right\} \\
&= \frac{1}{2} \left((T-1) \ln \left| \left(\mathbf{\Lambda}_n\right)^{-1} \right| - \sum_{t=2}^{T} \mathrm{E}_{q(\boldsymbol{P}_{tn}, \boldsymbol{P}_{t-1n})} \left\{ \left\| \boldsymbol{P}_{tn} - \boldsymbol{P}_{t-1n} \right\|_{\mathbf{\Lambda}_n} \right\} \right)
\end{aligned} \tag{A.25}$$

If we now analyse the expectation term:

$$\begin{aligned}
&\mathrm{E}_{q(\boldsymbol{P}_{tn}, \boldsymbol{P}_{t-1n})} \left\{ \left\| \boldsymbol{P}_{tn} - \boldsymbol{P}_{t-1n} \right\|_{\mathbf{\Lambda}_n} \right\} \\
&= \quad \mathrm{E}_{q(\boldsymbol{P}_{tn}, \boldsymbol{P}_{t-1n})} \left\{ \boldsymbol{P}_{tn}^{\top} \left(\mathbf{\Lambda}_n\right)^{-1} \boldsymbol{P}_{tn} - \boldsymbol{P}_{t-1n}^{\top} \left(\mathbf{\Lambda}_n\right)^{-1} \boldsymbol{P}_{tn} - \boldsymbol{P}_{tn}^{\top} \left(\mathbf{\Lambda}_n\right)^{-1} \boldsymbol{P}_{t-1n} \right. \\
&\qquad \left. + \boldsymbol{P}_{t-1n}^{\top} \left(\mathbf{\Lambda}_n\right)^{-1} \boldsymbol{P}_{t-1n} \right\} \\
&= \mathrm{E}_{q(\boldsymbol{P}_{tn}, \boldsymbol{P}_{t-1n})} \left\{ \mathrm{tr} \left(\left(\mathbf{\Lambda}_n\right)^{-1} \left[\boldsymbol{P}_{tn} \boldsymbol{P}_{tn}^{\top} - \boldsymbol{P}_{t-1n} \boldsymbol{P}_{tn}^{\top} - \boldsymbol{P}_{tn} \boldsymbol{P}_{t-1n}^{\top} + \boldsymbol{P}_{t-1n} \boldsymbol{P}_{t-1n}^{\top} \right] \right) \right\} \\
&= \mathrm{tr} \left(\left(\mathbf{\Lambda}_n\right)^{-1} \left[\mathrm{E}_{q(\boldsymbol{P}_{tn}, \boldsymbol{P}_{t-1n})} \left\{ \boldsymbol{P}_{tn} \boldsymbol{P}_{tn}^{\top} - \boldsymbol{P}_{t-1n} \boldsymbol{P}_{tn}^{\top} - \boldsymbol{P}_{tn} \boldsymbol{P}_{t-1n}^{\top} + \boldsymbol{P}_{t-1n} \boldsymbol{P}_{t-1n}^{\top} \right\} \right] \right) \\
&= \mathrm{tr} \left(\left(\mathbf{\Lambda}_n\right)^{-1} \mathbf{E}_{tn}^{(r+1)} \right)
\end{aligned}$$

with $\mathbf{E}_{tn}^{(r+1)}$ begin the energy matrix with the following expression:

$$\begin{aligned}
\mathbf{E}_{tn}^{(r+1)} = &\left(\mathbf{\Psi}_{tn}^{\zeta(r+1)} \right)_{11} + \left(\boldsymbol{\mu}_{tn}^{\zeta(r+1)} \right)_1 \left(\boldsymbol{\mu}_{tn}^{\zeta(r+1)} \right)_1^{\top} + \left(\mathbf{\Psi}_{tn}^{\zeta(r+1)} \right)_{22} \\
&+ \left(\boldsymbol{\mu}_{tn}^{\zeta(r+1)} \right)_2 \left(\boldsymbol{\mu}_{tn}^{\zeta(r+1)} \right)_2^{\top} - \left(\mathbf{\Psi}_{tn}^{\zeta(r+1)} \right)_{12} - \left(\boldsymbol{\mu}_{tn}^{\zeta(r+1)} \right)_1 \left(\boldsymbol{\mu}_{tn}^{\zeta(r+1)} \right)_2^{\top} \\
&- \left(\mathbf{\Psi}_{tn}^{\zeta(r+1)} \right)_{21} - \left(\boldsymbol{\mu}_{tn}^{\zeta(r+1)} \right)_2 \left(\boldsymbol{\mu}_{tn}^{\zeta(r+1)} \right)_1^{\top},
\end{aligned}$$

where the subscribes here denote the first half or the second half in the correspondent dimension. Therefore, we can rewrite (A.25) as:

$$\mathcal{Q}_{\mathbf{\Lambda}_n} = \frac{1}{2} \left((T-1) \ln \left| \left(\mathbf{\Lambda}_n\right)^{-1} \right| - \sum_{t=2}^{T} \mathrm{tr} \left(\left(\mathbf{\Lambda}_n\right)^{-1} \mathbf{E}_{tn}^{(r+1)} \right) \right).$$

Taking now the derivative and cancelling it:

$$\mathbf{\Lambda}_n^{(r+1)} = \frac{1}{T-1}\sum_{t=2}^{T}\mathbf{E}_{tn}^{(r+1)}. \tag{A.26}$$

We notice that if the matrices $\mathbf{E}_{tn}^{(r+1)}$ are symmetric and positive definite, then $\mathbf{\Lambda}_n^{(r+1)}$ will also be. From its definition it is obvious that they are symmetric. But are they positive definite? Well, Let's see. Let us first observe that the matrix $\mathbf{\Psi}_{tn}^{\zeta(r+1)}$ is positive definite. Indeed, its inverse is:

$$\left(\mathbf{\Psi}_{tn}^{\zeta(r+1)}\right)^{-1} = \mathbf{M} + \mathbf{\Delta} \otimes \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

begin $\mathbf{M}$ a block-diagonal matrix composed of two covariance matrices of the same size as $\mathbf{\Delta}$, which is always a covariance matrix. We notice that for any $\boldsymbol{x} = (\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top)^\top$:

$$\boldsymbol{x}^\top \left(\mathbf{\Psi}_{tn}^{\zeta(r+1)}\right)^{-1} \boldsymbol{x} =$$

$$\boldsymbol{x}^\top \mathbf{M} \boldsymbol{x} + \boldsymbol{x}^\top \mathbf{\Delta} \otimes \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \boldsymbol{x} > \tilde{\boldsymbol{x}}^\top \mathbf{\Delta} \otimes \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \tilde{\boldsymbol{x}} \geq 0$$

with $\tilde{\boldsymbol{x}} = (\boldsymbol{x}_1^\top, -\boldsymbol{x}_2^\top)^\top$. Because $\mathbf{\Psi}_{tn}^{\zeta(r+1)}$ is symmetric and positive definite, $\mathbf{\Psi}_{tn}^{\zeta(r+1)} + \boldsymbol{\mu}_{tn}^{\zeta(r+1)}\left(\boldsymbol{\mu}_{tn}^{\zeta(r+1)}\right)^\top$ is also symmetric and positive definite. It is easy to see that, for any symmetric and positive definite matrix $\mathbf{P} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{12}^\top & \mathbf{P}_{22} \end{pmatrix}$ the matrix $\mathbf{P}_{11} + \mathbf{P}_{22} - \mathbf{P}_{12} - \mathbf{P}_{12}^\top$ is also positive definite. Indeed, for a given vector $\boldsymbol{y}$ we notice that:

$$\boldsymbol{y}^\top \left(\mathbf{P}_{11} + \mathbf{P}_{22} - \mathbf{P}_{12} - \mathbf{P}_{12}^\top\right) \boldsymbol{y} = \begin{pmatrix} \boldsymbol{y} \\ -\boldsymbol{y} \end{pmatrix}^\top \mathbf{P} \begin{pmatrix} \boldsymbol{y} \\ -\boldsymbol{y} \end{pmatrix} > 0,$$

because $\mathbf{P}$ is a positive definite. Therefore, $\mathbf{E}_{tn}^{(r+1)}$ are symmetric and positive definite matrices and in all, $\mathbf{\Lambda}_n^{(r+1)}$ is also.

The parameters $(\boldsymbol{\nu}_n, \boldsymbol{\Omega}_n)$ Finally, we evaluate the 4^{th} term of the developed Q-function.

From A.16 and A.18, we already know as well:

$$q\left(\boldsymbol{P}_{1n}\right) = \mathcal{N}\left(\boldsymbol{P}_{1n}; \boldsymbol{\mu}_{1n}^{(r+1)}, \boldsymbol{\Psi}_{1n}^{(r+1)}\right)$$

and we define and develop

$$\begin{aligned}
&= \sum_{n=1}^{N} \mathrm{E}_{q(\boldsymbol{P}_{1n})} \left\{\ln p\left(\boldsymbol{P}_{1n} | \boldsymbol{\nu}_n, \boldsymbol{\Omega}_n\right)\right\} \\
&= \sum_{n=1}^{N} \mathrm{E}_{q(\boldsymbol{P}_{1n})} \left\{\ln \mathcal{N}\left(\boldsymbol{P}_{1n}; \boldsymbol{\nu}_n, \boldsymbol{\Omega}_n\right)\right\} \\
&= \sum_{n=1}^{N} \mathrm{E}_{q(\boldsymbol{P}_{1n})} \left\{-\ln 2\pi^{D/2} + \frac{1}{2}\ln\left|\left(\boldsymbol{\Omega}_n\right)^{-1}\right| - \frac{1}{2}\left[\left(\boldsymbol{P}_{1n} - \boldsymbol{\nu}_n\right)^{\top} \boldsymbol{\Omega}_n \left(\boldsymbol{P}_{1n} - \boldsymbol{\nu}_n\right)\right]\right\} \\
&= -\ln 2\pi^{D/2} + \sum_{n=1}^{N} \left[\frac{1}{2}\ln\left|\left(\boldsymbol{\Omega}_n\right)^{-1}\right| - \frac{1}{2}\mathrm{E}_{q(\boldsymbol{P}_{1n})} \left\{\left(\boldsymbol{P}_{1n} - \boldsymbol{\nu}_n\right)^{\top} \boldsymbol{\Omega}_n \left(\boldsymbol{P}_{1n} - \boldsymbol{\nu}_n\right)\right\}\right] \\
&= -\ln 2\pi^{D/2} + \sum_{n=1}^{N} \left[\frac{1}{2}\ln\left|\left(\boldsymbol{\Omega}_n\right)^{-1}\right| - \frac{1}{2}\left[\mathrm{tr}\left(\boldsymbol{\Omega}_n^{-1}\left(\boldsymbol{\Psi}_{1n}^{(r+1)} + \boldsymbol{\mu}_{1n}^{(r+1)}\left(\boldsymbol{\mu}_{1n}^{(r+1)}\right)^{\top}\right)\right)\right.\right. \\
&\qquad \left.\left. -2\left(\boldsymbol{\mu}_{1n}^{(r+1)}\right)^{\top} \boldsymbol{\Omega}_n^{-1}\boldsymbol{\nu}_n + \boldsymbol{\nu}_n^{\top}\boldsymbol{\Omega}_n^{-1}\boldsymbol{\nu}_n\right]\right]
\end{aligned}$$

To compute $(\boldsymbol{\nu}_n, \boldsymbol{\Omega}_n)$, we set the derivative of the above function with respect to each of the parameters to zero and rearranging, we obtain the formulas.

$$\frac{\partial \mathcal{Q}_{\boldsymbol{\nu}_n, \boldsymbol{\Omega}_n}}{\partial \boldsymbol{\nu}_n} = -2\boldsymbol{\Omega}_n^{-1}\left(\boldsymbol{\mu}_{1n}^{(r+1)}\right) + 2\boldsymbol{\Omega}_n^{-1}\boldsymbol{\nu}_n = 0$$

$$\boxed{\boldsymbol{\nu}_n = \boldsymbol{\mu}_{1n}^{(r+1)}} \tag{A.27}$$

$$\frac{\partial \mathcal{Q}_{\boldsymbol{\nu}_n, \boldsymbol{\Omega}_n}}{\partial \boldsymbol{\Omega}_n^{-1}} = \frac{\boldsymbol{\Omega}_n}{2} - \frac{1}{2}\left[\boldsymbol{\Psi}_{1n}^{(r+1)} + \boldsymbol{\mu}_{1n}^{(r+1)}\left(\boldsymbol{\mu}_{1n}^{(r+1)}\right)^{\top} - 2\boldsymbol{\mu}_{1n}^{(r+1)}\boldsymbol{\nu}_n^{\top} + \boldsymbol{\nu}_n\boldsymbol{\nu}_n^{\top}\right] = 0$$

$$\boxed{\boldsymbol{\Omega}_n = \boldsymbol{\Psi}_{1n}^{(r+1)} + \boldsymbol{\mu}_{1n}^{(r+1)}\left(\boldsymbol{\mu}_{1n}^{(r+1)}\right)^{\top} - 2\boldsymbol{\mu}_{1n}^{(r+1)}\boldsymbol{\nu}_n^{\top} + \boldsymbol{\nu}_n\boldsymbol{\nu}_n^{\top} \overset{(A.27)}{=} \boldsymbol{\Psi}_{1n}^{(r+1)}} \tag{A.28}$$

Appendix B

Derivation of EM: Clutter

In chapter 3, we described a first extension of the proposed method: we modelled the clutter. Here, we show the complete derivations of all formulas for the Expectation-Maximization algorithm.

The Clutter Model

Observations

$$\boldsymbol{F}_{tk}|Z_{tk}=n,\boldsymbol{P}_t\sim\begin{cases}\mathcal{N}\left(\boldsymbol{F}_{tk};\boldsymbol{P}_{tn},\boldsymbol{\Sigma}_n\right) & \text{if } n\leq N\\ \mathcal{U}\left(\boldsymbol{F}_{tk};A\right) & \text{if } n=N+1\end{cases}$$

where A is the area taken into account, with width and height fixed.

Assignment variables

$$Z_{tk}\sim\mathcal{M}\left(Z_{tk};N+1,\boldsymbol{\pi}_t\right).$$

EM algorithm

E-Z step

$$
\begin{aligned}
\eta_{tkn}^{(r+1)} &= q\left(Z_{tk}=n\right) \\
&\propto \exp\left(\ln p\left(Z_{tk}=n\right)+\mathrm{E}_{q(\boldsymbol{S}_n)}\left\{\ln p\left(\boldsymbol{f}_{tk}|Z_{tk}=n,\boldsymbol{S}_n,\theta^{(r)}\right)\right\}\right) && \text{(B.1)} \\
&\propto \begin{cases} \pi_{tn}\exp\left(\mathrm{E}_{q(\boldsymbol{S}_{tn})}\left\{\ln\mathcal{N}\left(\boldsymbol{f}_{tk};\boldsymbol{S}_{tn},\boldsymbol{\Sigma}_{tn}^{(r)}\right)\right\}\right) & \text{if } n\leq N \\ \pi_{tn}\exp\left(\mathrm{E}_{q(\boldsymbol{S}_{tn})}\left\{\ln\mathcal{U}\left(\boldsymbol{f}_{tk};a,b\right)\right\}\right) & \text{if } n=N+1 \end{cases} && \text{(B.2)} \\
&\propto \begin{cases} \pi_{tn}\exp\left[-\ln\left|\boldsymbol{\Sigma}_{tn}^{(r)}\right|^{\frac{1}{2}}-\frac{1}{2}\mathrm{tr}\left(\left(\boldsymbol{\Sigma}_{tn}^{(r)}\right)^{-1}\left(\boldsymbol{\Psi}_{tn}^{(r)}+\boldsymbol{\mu}_{tn}^{(r)}\left(\boldsymbol{\mu}_{tn}^{(r)}\right)^{\top}\right)\right)-\right. \\ \left.\quad\frac{1}{2}\left(\boldsymbol{f}_{tk}-2\boldsymbol{\mu}_{tn}^{(r)}\right)^{\top}\left(\boldsymbol{\Sigma}_{tn}^{(r)}\right)^{-1}\boldsymbol{f}_{tk}\right] & \text{if } n\leq N \\ \pi_{tn}\frac{1}{|A|} & \text{if } n=N+1 \end{cases} && \text{(B.3)}
\end{aligned}
$$

where $|A|$ is the area taken into account.

M step

In the the M step, the expected complete-data log-likelihood function, recalling below, and factorization done are not affected by the adding of the clutter, except for the parameter π.

$$
\begin{aligned}
\mathcal{Q}\left(\theta,\theta^{(r)}\right) &= \sum_{t,k=1}^{T,K_t}\mathrm{E}_{q(Z_{tk})}\left\{\ln p\left(Z_{tk}|\theta\right)\right\}+\mathrm{E}_{q(Z_{tk})q(\boldsymbol{P}_t)}\left\{\ln p\left(\boldsymbol{f}_{tk}|Z_{tk},\boldsymbol{P}_t,\theta\right)\right\}+ \\
&\quad+\sum_{n=1}^{N}\sum_{t=2}^{T}\mathrm{E}_{q(\boldsymbol{P}_{tn},\boldsymbol{P}_{t-1n})}\left\{\ln p\left(\boldsymbol{P}_{t-1n}|\boldsymbol{P}_{tn},\theta\right)\right\}+ \\
&\quad+\sum_{n=1}^{N}\mathrm{E}_{q(\boldsymbol{P}_{1n})}\left\{\ln p\left(\boldsymbol{P}_{1n}|\theta\right)\right\} && \text{(B.4)}
\end{aligned}
$$

We can notice the function consists of four terms in which the uniform distribution appears during derivations, but it can be ignored after taking the derivative.

The parameter π

From the equation B.4, we can define:

$$\begin{aligned}
\mathcal{Q}\left(\pi, \pi^{(r)}\right) &= \sum_{t,k=1}^{T,K_t} \mathrm{E}_{q(Z_{tk})} \left\{\ln p\left(Z_{tk}|\pi_{tn}\right)\right\} \\
&= \sum_{t,k=1}^{T,K_t} \mathrm{E}_{q(Z_{tk})} \left\{\ln \prod_{n=1}^{N+1} \pi_{tn}^{Z_{tkn}}\right\} \\
&= \sum_{t,k=1}^{T,K_t} \sum_{n=1}^{N+1} \mathrm{E}_{q(Z_{tk})} \left\{Z_{tkn}\right\} \ln \pi_{tn} \\
&= \sum_{t,k=1}^{T,K_t} \sum_{n=1}^{N+1} \eta_{tkn}^{Z(r+1)} \ln \pi_{tn} \qquad \text{(B.5)}
\end{aligned}$$

Unlikely the initial method, now this term takes into account an extra class, which coincides with the clutter.

In order to obtain formula of the model's parameter π, we have to set the derivative of the previous term to zero and rearranging by means of appropriate Lagrange multiplier. We can take into account two cases of the parameter: time-invariant and time-dependant.

Time-invariant parameters

$$\frac{\partial \mathcal{Q}\left(\pi, \pi^{(r)}\right)}{\partial \pi_n} = \sum_{k,t=1}^{K_t,T} \eta_{tkn}^{Z(r+1)} \frac{1}{\pi_n} = 0 \quad \Rightarrow \quad \boxed{\pi_n = \frac{\sum_{k,t=1}^{K_t,T} \eta_{tkn}^{Z(r+1)}}{\sum_{t=1}^{T} K_t}} \qquad \text{(B.6)}$$

Time-dependant parameters

$$\frac{\partial \mathcal{Q}\left(\pi, \pi^{(r)}\right)}{\partial \pi_{tn}} = \sum_{k=1}^{K_t} \eta_{tkn}^{Z(r+1)} \frac{1}{\pi_{tn}} = 0 \quad \Rightarrow \quad \boxed{\pi_{tn} = \frac{\sum_{k=1}^{K_t} \eta_{tkn}^{Z(r+1)}}{K_t}} \qquad \text{(B.7)}$$

Appendix C

Derivation of EM: Velocity

In chapter 3, we described an extension of the proposed model with the introduction of the velocity. Here, we report the complete derivations of all presented formulas.

C.1 The Velocity Model

Sources

$$\boldsymbol{P}_{tn}|\boldsymbol{P}_{t-1,n} \sim \mathcal{N}\left(\boldsymbol{P}_{tn}; \boldsymbol{P}_{t-1,n} + \boldsymbol{V}_{tn}, \boldsymbol{\Lambda}_n^P\right)$$
$$\boldsymbol{V}_{tn}|\boldsymbol{V}_{t-1,n} \sim \mathcal{N}\left(\boldsymbol{V}_{tn}; \boldsymbol{V}_{t-1,n}, \boldsymbol{\Lambda}_n^V\right)$$

These two formulas can be combined in order to deal with only one Gaussian:

$$\left(\begin{array}{c}\boldsymbol{P}_{t,n}\\ \boldsymbol{V}_{t,n}\end{array}\right)\Bigg|\left(\begin{array}{c}\boldsymbol{P}_{t-1,n}\\ \boldsymbol{V}_{t-1,n}\end{array}\right) \sim \mathcal{N}\left(\left(\begin{array}{c}\boldsymbol{P}_{t,n}\\ \boldsymbol{V}_{t,n}\end{array}\right); \mathbf{A}\left(\begin{array}{c}\boldsymbol{P}_{t-1,n}\\ \boldsymbol{V}_{t-1,n}\end{array}\right), \boldsymbol{\Lambda}\right)$$

As far as concerned the covariance matrix $\boldsymbol{\Lambda}$, we describe it in chapter 3 and a proof of its invertibility is shown in appendix D.

In all the set of parameters is:

$$\theta = \underbrace{\left\{\boldsymbol{\nu}_n^V, \boldsymbol{\Omega}_n^V, \boldsymbol{\Lambda}_n^V\right\}_{n=1}^N}_{\theta_{\boldsymbol{V}}} \cup \underbrace{\left\{\boldsymbol{\nu}_n^P, \boldsymbol{\Omega}_n^P, \boldsymbol{\Lambda}_n^P\right\}_{n=1}^N}_{\theta_{\boldsymbol{P}}} \cup \underbrace{\left\{\boldsymbol{\pi}_t\right\}_{t=1}^T}_{\theta_{\boldsymbol{\pi}}} \cup \underbrace{\left\{\boldsymbol{\Sigma}_{t,n}\right\}_{t,n=1}^{T,N}}_{\theta_{\boldsymbol{f}}}$$

C.2 EM Algorithm

The $\mathcal{Q}$ function We report here the expected complete-data log-likelihood function:

$$\mathcal{Q}\left(\theta, \theta^{(r)}\right) = \mathrm{E}_{p\left(Z,\boldsymbol{P},\boldsymbol{V}|\boldsymbol{f},\theta^{(r)}\right)}\left\{\ln p\left(Z, \boldsymbol{P}, \boldsymbol{V}, \boldsymbol{f}|\theta\right)\right\}. \tag{C.1}$$

As far as concerned this functions, all considerations, such as closed-form which is non computationally tractable, variational approach and E-Z step, were done in the chapter 3. Here, we go in detail of E-S step and M step with respectively formulas.

C.2.1 Expectation step

E-S step

In this step we want to compute $q(\boldsymbol{P}_n, \boldsymbol{V}_n)$, to further on compute $q(\boldsymbol{P}_{t,n}, \boldsymbol{V}_{t,n})$. Using the proposed factorization we know that:

$$q\left(\boldsymbol{P}_n, \boldsymbol{V}_n\right) \propto \exp\left(\mathrm{E}_{q\left(\boldsymbol{P}_{1:N\setminus n}, \boldsymbol{V}_{1:N\setminus n}\right)q(Z)}\left\{\ln p\left(\boldsymbol{f}, Z, \boldsymbol{P}, \boldsymbol{V}|\theta^{(r)}\right)\right\}\right).$$

Let us first analyse the term inside the expectation:

$$\ln p\left(\boldsymbol{f}, Z, \boldsymbol{P}, \boldsymbol{V}|\theta^{(r)}\right) = \ln\left(p\left(\boldsymbol{f}|\boldsymbol{P}, \boldsymbol{V}, Z, \theta^{(r)}\right) p\left(\boldsymbol{P}|\boldsymbol{V}, \theta^{(r)}\right) p\left(\boldsymbol{V}|\theta^{(r)}\right) p\left(Z|\theta^{(r)}\right)\right).$$

We notice that the last term does not depend neither on $\boldsymbol{P}_n$ nor on $\boldsymbol{V}_n$, and can therefore be removed from the computation. As well, $p\left(\boldsymbol{f}|\boldsymbol{P}, \boldsymbol{V}, Z, \theta^{(r)}\right)$ does not depend on $\boldsymbol{V}$, and can be simplify in $p\left(\boldsymbol{f}|\boldsymbol{P}, Z, \theta^{(r)}\right)$. Similarly, because $p\left(\boldsymbol{P}\right)$ and $p\left(\boldsymbol{V}\right)$ are separable on n, we can remove all other sources:

$$\begin{aligned} q\left(\boldsymbol{P}_n, \boldsymbol{V}_n\right) \quad \propto \quad & p\left(\boldsymbol{V}_n|\theta^{(r)}\right) p\left(\boldsymbol{P}_n|\boldsymbol{V}_n, \theta^{(r)}\right) * \\ & * \exp\left(\mathrm{E}_{q\left(\boldsymbol{P}_{1:N\setminus n}, \boldsymbol{V}_{1:N\setminus n}\right)q(Z)}\left\{\ln p\left(\boldsymbol{f}|Z, \boldsymbol{P}, \theta^{(r)}\right)\right\}\right) \end{aligned} \tag{C.2}$$

As far as concerned the expectation in C.2, the computation is the same as we did for the model that takes only positions into account (see appendix A). Here, we report only the result:

$$\mathrm{E}_{q\left(\boldsymbol{P}_{1:N\setminus n}\right)q(Z)}\left\{\ln p\left(\boldsymbol{f}|Z, \boldsymbol{P}, \theta^{(r)}\right)\right\} \overset{\boldsymbol{P}_n}{=} \sum_{t,k=1}^{T,K_t} \ln \mathcal{N}\left(\boldsymbol{f}_{t,k}; \boldsymbol{P}_{t,n}, \frac{\boldsymbol{\Sigma}_{t,n}^{(r)}}{\eta_{t,k,n}^{(r+1)}}\right)$$

As regards the other probabilities in C.2, we can make some considerations:

$$p\left(\boldsymbol{V}_n|\theta^{(r)}\right) \sim \mathcal{N}\left(\boldsymbol{V}_{1,n};\boldsymbol{\nu}_n^{V^{(r)}},\boldsymbol{\Omega}_n^{V^{(r)}}\right)\prod_{t=2}^{T}\mathcal{N}\left(\boldsymbol{V}_{t,n};\boldsymbol{V}_{t-1,n},\boldsymbol{\Lambda}_n^{V^{(r)}}\right) \quad \text{(C.3)}$$

$$\begin{aligned} p\left(\boldsymbol{P}_n|\boldsymbol{V}_n,\theta^{(r)}\right) &\sim \mathcal{N}\left(\boldsymbol{P}_{1,n};\boldsymbol{\nu}_n^{P^{(r)}}+\boldsymbol{V}_{1,n},\boldsymbol{\Omega}_n^{P^{(r)}}\right)* \\ &\quad *\prod_{t=2}^{T}\mathcal{N}\left(\boldsymbol{P}_{t,n};\boldsymbol{P}_{t-1,n}+\boldsymbol{V}_{t,n},\boldsymbol{\Lambda}_n^{P^{(r)}}\right) \\ &\sim \mathcal{N}\left(\boldsymbol{P}_{1,n};\boldsymbol{\nu}_n^{P^{(r)}},\boldsymbol{\Omega}_n^{P^{(r)}}\right)* \\ &\quad *\prod_{t=2}^{T}\mathcal{N}\left(\boldsymbol{P}_{t,n};\boldsymbol{P}_{t-1,n}+\boldsymbol{V}_{t,n},\boldsymbol{\Lambda}_n^{P^{(r)}}\right) \end{aligned} \quad \text{(C.4)}$$

We can consider initialization independent for both positions and velocities, and this is the reason why we ignore the dependence on $\boldsymbol{V}$ for the first time step in C.3 and C.4. Besides, we can combined them for convenience in the following one:

$$\begin{aligned} p\left(\boldsymbol{P}_n,\boldsymbol{V}_n|\theta^{(r)}\right) &\sim \mathcal{N}\left(\begin{pmatrix}\boldsymbol{P}_{1,n}\\ \boldsymbol{V}_{1,n}\end{pmatrix};\boldsymbol{\nu}_n^{(r)},\boldsymbol{\Omega}_n^{(r)}\right)* \\ &\quad *\prod_{t=2}^{T}\mathcal{N}\left(\begin{pmatrix}\boldsymbol{P}_{t,n}\\ \boldsymbol{V}_{t,n}\end{pmatrix};\mathbf{A}\begin{pmatrix}\boldsymbol{P}_{t-1,n}\\ \boldsymbol{V}_{t-1,n}\end{pmatrix},\boldsymbol{\Lambda}\right) \end{aligned} \quad \text{(C.5)}$$

with

$$\boldsymbol{\nu}_n^{(r)} = \begin{pmatrix}\boldsymbol{\nu}_n^{P^{(r)}}\\ \boldsymbol{\nu}_n^{V^{(r)}}\end{pmatrix} \qquad \text{and} \qquad \boldsymbol{\Omega}_n^{(r)} = \begin{pmatrix}\boldsymbol{\Omega}_n^{P^{(r)}} & 0\\ 0 & \boldsymbol{\Omega}_n^{V^{(r)}}\end{pmatrix}$$

From here, we denote $\begin{pmatrix}\boldsymbol{P}_{t,n}\\ \boldsymbol{V}_{t,n}\end{pmatrix}$ with $\boldsymbol{S}_{t,n}$.

Replacing all above results into (C.2) we write:

$$\begin{aligned} q\left(\boldsymbol{S}_n\right) &\propto \mathcal{N}\left(\boldsymbol{S}_{1,n};\boldsymbol{\nu}_n^{(r)},\boldsymbol{\Omega}_n^{(r)}\right)\prod_{t=2}^{T}\mathcal{N}\left(\boldsymbol{S}_{t,n};\mathbf{A}\boldsymbol{S}_{t-1,n},\boldsymbol{\Lambda}\right)* \\ &\quad *\prod_{t=1}^{T}\prod_{k=1}^{K_t}\mathcal{N}\left(\boldsymbol{f}_{t,k};\boldsymbol{P}_{t,n},\frac{\boldsymbol{\Sigma}_{t,n}^{(r)}}{\eta_{t,k,n}^{(r+1)}}\right) \end{aligned} \quad \text{(C.6)}$$

In order to keep the dimensions consistent, we write:

$$\prod_{k=1}^{K_t} \mathcal{N}\left(\boldsymbol{f}_{t,k}; \boldsymbol{P}_{t,n}, \frac{\boldsymbol{\Sigma}_n^{(r)}}{\eta_{t,k,n}^{(r+1)}}\right) \overset{\boldsymbol{P}_n}{\propto} \mathcal{N}\left(\boldsymbol{S}_{1,n}; \boldsymbol{\mu}_{t,n}^{\iota(r+1)}, \boldsymbol{\Psi}_{t,n}^{\iota(r+1)}\right),$$

with

$$\left(\boldsymbol{\Psi}_{t,n}^{\iota(r+1)}\right)^{-1} = \left(\begin{array}{c|c} \left(\sum_{k=1}^{K_t} \eta_{t,k,n}^{(r+1)}\right)\left(\boldsymbol{\Sigma}_{t,n}^{(r)}\right)^{-1} & 0 \\ \hline 0 & 0 \end{array}\right)$$

$$\boldsymbol{\mu}_{t,n}^{\iota(r+1)} = \left(\begin{array}{c} \dfrac{\sum_{k=1}^{K_t} \eta_{t,k,n}^{(r+1)} \boldsymbol{f}_{t,k}}{\sum_{k=1}^{K_t} \eta_{t,k,n}^{(r+1)}} \\ \hline ? \end{array}\right).$$

Since we have no knowledge about velocity, we want zero value in the precision matrix and this results in infinity values into $\boldsymbol{\Psi}_{t,n}^{\iota(r+1)}$, which is a block matrix (see [Petersen and Pedersen, 2012]). Due to zero values in precision matrix, we can assign every values to the block referring to velocity in $\boldsymbol{\mu}_{t,n}^{\iota(r+1)}$. Finally, (C.6) rewrites:

$$\begin{aligned} q\left(\boldsymbol{S}_n\right) \quad \propto \quad & \mathcal{N}\left(\boldsymbol{S}_{1,n}; \boldsymbol{\nu}_n^{(r)}, \boldsymbol{\Omega}_n^{(r)}\right) \prod_{t=2}^{T} \mathcal{N}\left(\boldsymbol{S}_{t,n}; \mathbf{A}\boldsymbol{S}_{t-1,n}, \boldsymbol{\Lambda}\right) * \\ & * \prod_{t=1}^{T} \mathcal{N}\left(\boldsymbol{S}_{t,n}; \boldsymbol{\mu}_{t,n}^{\iota(r+1)}, \boldsymbol{\Psi}_{t,n}^{\iota(r+1)}\right) \end{aligned} \tag{C.7}$$

At the light of this result, we are certain that $q(\boldsymbol{S}_n)$ is a multivariate normal. Because of the temporal term, it is not obvious to extract $q(\boldsymbol{S}_{t,n})$ from here. Direct marginalization will naturally lead to forward-backward recursions.

The marginal distribution - forward/backward Let us compute $q(\boldsymbol{S}_{t,n})$, by marginalizing (C.7):

$$\begin{aligned} q\left(\boldsymbol{S}_n\right) \propto \quad & \int \mathcal{N}\left(\boldsymbol{S}_{1,n}; \boldsymbol{\nu}_n^{(r)}, \boldsymbol{\Omega}_n^{(r)}\right) \prod_{s=2}^{T} \mathcal{N}\left(\boldsymbol{S}_{s,n}; \mathbf{A}\boldsymbol{S}_{s-1,n}, \boldsymbol{\Lambda}\right) * \\ & * \prod_{s=1}^{T} \mathcal{N}\left(\boldsymbol{S}_{s,n}; \boldsymbol{\mu}_{s,n}^{\iota(r+1)}, \boldsymbol{\Psi}_{s,n}^{\iota(r+1)}\right) \mathrm{d}\boldsymbol{S}_{1:t\setminus s,n} \end{aligned}$$

We can rewrite this marginalization into a product of two marginalizations representing the *past* and the *future* with respect to $\boldsymbol{S}_{t,n}$:

$$\begin{aligned} q\left(\boldsymbol{S}_n\right) \quad \propto \quad & \int \mathcal{N}\left(\boldsymbol{S}_{1,n}; \boldsymbol{\nu}_n^{(r)}, \boldsymbol{\Omega}_n^{(r)}\right) \prod_{s=2}^{t} \mathcal{N}\left(\boldsymbol{S}_{t,n}; \mathbf{A}\boldsymbol{S}_{s-1,n}, \boldsymbol{\Lambda}\right) * \\ & * \prod_{s=1}^{t} \mathcal{N}\left(\boldsymbol{S}_{t,n}; \boldsymbol{\mu}_{s,n}^{\iota(r+1)}, \boldsymbol{\Psi}_{s,n}^{\iota(r+1)}\right) \mathrm{d}\boldsymbol{S}_{1:t-1,n} * \qquad \text{(C.8)} \\ & * \int \prod_{s=t+1}^{T} \mathcal{N}\left(\boldsymbol{S}_{s,n}; \mathbf{A}\boldsymbol{S}_{s-1,n}, \boldsymbol{\Lambda}\right) * \\ & * \prod_{s=t+1}^{T} \mathcal{N}\left(\boldsymbol{S}_{s,n}; \boldsymbol{\mu}_{s,n}^{\iota(r+1)}, \boldsymbol{\Psi}_{s,n}^{\iota(r+1)}\right) \mathrm{d}\boldsymbol{S}_{t+1:t,n} \qquad \text{(C.9)} \end{aligned}$$

The quantity on (C.8) will be denoted $\phi\left(\boldsymbol{S}_{t,n}\right)$ and the one on (C.9), $\beta\left(\boldsymbol{S}_{t,n}\right)$. With this notation we can get forward-backward recursions for ϕ and β:

$$\phi\left(\boldsymbol{S}_{t,n}\right) = \mathcal{N}\left(\boldsymbol{S}_{t,n}; \boldsymbol{\mu}_{t,n}^{\iota(r+1)}, \boldsymbol{\Psi}_{t,n}^{\iota(r+1)}\right) \int \phi\left(\boldsymbol{S}_{t-1,n}\right) \mathcal{N}\left(\boldsymbol{S}_{t,n}; \mathbf{A}\boldsymbol{S}_{t-1,n}, \boldsymbol{\Lambda}\right) \mathrm{d}\boldsymbol{S}_{t-1,n} \tag{C.10}$$

$$\beta\left(\boldsymbol{S}_{t,n}\right) = \int \beta\left(\boldsymbol{S}_{t+1,n}\right) \mathcal{N}\left(\boldsymbol{S}_{t+1,n}; \mathbf{A}\boldsymbol{S}_{t,n}, \boldsymbol{\Lambda}\right) \mathcal{N}\left(\boldsymbol{S}_{t+1,n}; \boldsymbol{\mu}_{t+1,n}^{\iota(r+1)}, \boldsymbol{\Psi}_{t+1,n}^{\iota(r+1)}\right) \mathrm{d}\boldsymbol{S}_{t+1,n} \tag{C.11}$$

We will now compute the recursions for ϕ and β. Let us assume that $\phi\left(\boldsymbol{S}_{t-1,n}\right)$ and $\beta\left(\boldsymbol{S}_{t+1,n}\right)$ are Gaussian and compute $\phi\left(\boldsymbol{S}_{t,n}\right)$ and $\beta\left(\boldsymbol{S}_{t,n}\right)$ from them. Therefore, we set

$$\begin{aligned} \phi\left(\boldsymbol{S}_{t-1,n}\right) &= \mathcal{N}\left(\boldsymbol{S}_{t-1,n}; \boldsymbol{\mu}_{t-1,n}^{\phi(r+1)}, \boldsymbol{\Psi}_{t-1,n}^{\phi(r+1)}\right) \\ \beta\left(\boldsymbol{S}_{t+1,n}\right) &= \mathcal{N}\left(\boldsymbol{S}_{t+1,n}; \boldsymbol{\mu}_{t+1,n}^{\beta(r+1)}, \boldsymbol{\Psi}_{t+1,n}^{\beta(r+1)}\right) \end{aligned}$$

Forward recursion

$$\begin{aligned} \phi(\boldsymbol{S}_{t,n}) \quad \propto \quad & \mathcal{N}\left(\boldsymbol{S}_{t,n}; \boldsymbol{\mu}_{t,n}^{\iota(r+1)}, \boldsymbol{\Psi}_{t,n}^{\iota(r+1)}\right) * \\ & \int \mathcal{N}\left(\boldsymbol{S}_{t-1,n}; \boldsymbol{\mu}_{t-1,n}^{\phi(r+1)}, \boldsymbol{\Psi}_{t-1,n}^{\phi(r+1)}\right) \mathcal{N}\left(\boldsymbol{S}_{t,n}; \mathbf{A}\boldsymbol{S}_{t-1,n}, \boldsymbol{\Lambda}_n^{(r)}\right) \mathrm{d}\boldsymbol{S}_{t-1,n} \\ \propto \quad & \mathcal{N}\left(\boldsymbol{S}_{t,n}; \boldsymbol{\mu}_{t,n}^{\iota(r+1)}, \boldsymbol{\Psi}_{t,n}^{\iota(r+1)}\right) \mathcal{N}\left(\boldsymbol{S}_{t,n}; \mathbf{A}\boldsymbol{\mu}_{t-1,n}^{\phi(r+1)}, \mathbf{A}\boldsymbol{\Psi}_{t-1,n}^{\phi(r+1)}\mathbf{A}^{\top} + \boldsymbol{\Lambda}_n^{(r)}\right) \\ \propto \quad & \mathcal{N}\left(\boldsymbol{S}_{t,n}; \boldsymbol{\mu}_{t,n}^{\phi(r+1)}, \boldsymbol{\Psi}_{t,n}^{\phi(r+1)}\right) \qquad \text{(C.12)} \end{aligned}$$

where

$$\left(\boldsymbol{\Psi}_{t,n}^{\phi(r+1)}\right)^{-1} = \left(\boldsymbol{\Psi}_{t,n}^{\iota(r+1)}\right)^{-1} + \left(\mathbf{A}\boldsymbol{\Psi}_{t-1,n}^{\phi(r+1)}\mathbf{A}^{\top} + \boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}$$

$$\boldsymbol{\mu}_{t,n}^{\phi(r+1)} = \boldsymbol{\Psi}_{t,n}^{\phi(r+1)}\left(\left(\boldsymbol{\Psi}_{t,n}^{\iota(r+1)}\right)^{-1}\boldsymbol{\mu}_{t,n}^{\iota(r+1)} + \left(\boldsymbol{\Psi}_{t-1,n}^{\phi(r+1)}\mathbf{A}^{\top} + \mathbf{A}^{-1}\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}\boldsymbol{\mu}_{t-1,n}^{\phi(r+1)}\right)$$

Backward recursion

$$\begin{aligned}
\beta\left(\boldsymbol{S}_{t,n}\right) &\propto \int \mathcal{N}\left(\boldsymbol{S}_{t+1,n};\boldsymbol{\mu}_{t+1,n}^{\beta(r+1)},\boldsymbol{\Psi}_{t+1,n}^{\beta(r+1)}\right)\mathcal{N}\left(\boldsymbol{S}_{t+1,n};\mathbf{A}\boldsymbol{S}_{t,n},\boldsymbol{\Lambda}_{n}^{(r)}\right)* \\
&\quad *\mathcal{N}\left(\boldsymbol{S}_{t+1,n};\boldsymbol{\mu}_{t+1,n}^{\iota(r+1)},\boldsymbol{\Psi}_{t+1,n}^{\iota(r+1)}\right)\mathrm{d}\boldsymbol{S}_{t+1,n} \\
&\propto \int \mathcal{N}\left(\boldsymbol{S}_{t+1,n};\boldsymbol{m},\boldsymbol{\Phi}\right)\exp\left(-\frac{1}{2}\left(-\left\|\boldsymbol{m}\right\|_{\boldsymbol{\Phi}} + \left\|\boldsymbol{\mu}_{t+1,n}^{\beta(r+1)}\right\|_{\boldsymbol{\Psi}_{t+1,n}^{\beta(r+1)}} + \right.\right. \\
&\quad \left.\left. + \left\|\mathbf{A}\boldsymbol{S}_{t,n}\right\|_{\boldsymbol{\Lambda}_{n}^{(r)}} + \left\|\boldsymbol{\mu}_{t+1,n}^{\iota(r)}\right\|_{\boldsymbol{\Psi}_{t+1,n}^{\iota(r+1)}}\right)\right)\mathrm{d}\boldsymbol{S}_{t+1,n} \\
&\propto \exp\left(-\frac{1}{2}\left(-\left\|\boldsymbol{m}\right\|_{\boldsymbol{\Phi}} + \left\|\boldsymbol{\mu}_{t+1,n}^{\beta(r+1)}\right\|_{\boldsymbol{\Psi}_{t+1,n}^{\beta(r+1)}} + \left\|\mathbf{A}\boldsymbol{S}_{t,n}\right\|_{\boldsymbol{\Lambda}_{n}^{(r)}} + \right.\right. \\
&\quad \left.\left. + \left\|\boldsymbol{\mu}_{t+1,n}^{\iota(r)}\right\|_{\boldsymbol{\Psi}_{t+1,n}^{\iota(r+1)}}\right)\right) \qquad \text{(C.13)}
\end{aligned}$$

with

$$\boldsymbol{\Phi}^{-1} = \left(\boldsymbol{\Psi}_{t+1,n}^{\beta(r+1)}\right)^{-1} + \left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1} + \left(\boldsymbol{\Psi}_{t+1,n}^{\iota(r+1)}\right)^{-1}$$

$$\boldsymbol{m} = \boldsymbol{\Phi}\left(\left(\boldsymbol{\Psi}_{t+1,n}^{\beta(r+1)}\right)^{-1}\boldsymbol{\mu}_{t+1,n}^{\beta(r+1)} + \left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}\mathbf{A}\boldsymbol{S}_{t,n} + \left(\boldsymbol{\Psi}_{t+1,n}^{\iota(r)}\right)^{-1}\boldsymbol{\mu}_{t+1,n}^{\iota(r+1)}\right)$$

The terms of the exponential depending on $\boldsymbol{S}_{t,n}$ are two (we do not take the $-\frac{1}{2}$ into account): the quadratic term

$$\left(\boldsymbol{S}_{t,n}\right)^{\top}\mathbf{A}^{\top}\left(\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1} - \left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}\boldsymbol{\Phi}\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}\right)\mathbf{A}\boldsymbol{S}_{t,n},$$

and the linear term

$$-2\left(\boldsymbol{S}_{t,n}\right)^{\top}\mathbf{A}^{\top}\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}\boldsymbol{\Phi}\left(\left(\boldsymbol{\Psi}_{t+1,n}^{\beta(r+1)}\right)^{-1}\boldsymbol{\mu}_{t+1,n}^{\beta(r+1)} + \left(\boldsymbol{\Psi}_{t+1,n}^{\iota(r+1)}\right)^{-1}\boldsymbol{\mu}_{t+1,n}^{\iota(r+1)}\right).$$

The quadratic term sets the covariance matrix of the distribution:

$$\boldsymbol{\Psi}_{t,n}^{\beta(r+1)} = \mathbf{A}^{-1}\left(\boldsymbol{\Lambda}_{n}^{(r)} + \left(\left(\boldsymbol{\Psi}_{t+1,n}^{\beta(r+1)}\right)^{-1} + \left(\boldsymbol{\Psi}_{t+1,n}^{\iota(r+1)}\right)^{-1}\right)^{-1}\right)\left(\mathbf{A}^{\top}\right)^{-1}. \qquad \text{(C.14)}$$

Together with the linear term, we set the mean to:

$$\begin{aligned}\boldsymbol{\mu}_{t,n}^{\beta(r+1)} &= \boldsymbol{\Psi}_{t,n}^{\beta(r+1)}\mathbf{A}^{\top}\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}\boldsymbol{\Phi} * \\ &\quad * \left(\left(\boldsymbol{\Psi}_{t+1,n}^{\beta(r+1)}\right)^{-1}\boldsymbol{\mu}_{t+1,n}^{\beta(r+1)} + \left(\boldsymbol{\Psi}_{t+1,n}^{\iota(r)}\right)^{-1}\boldsymbol{\mu}_{t+1,n}^{\iota(r+1)}\right) \\ &= \mathbf{A}^{-1}\left(\left(\boldsymbol{\Psi}_{t+1,n}^{\beta(r+1)}\right)^{-1} + \left(\boldsymbol{\Psi}_{t+1,n}^{\iota(r+1)}\right)^{-1}\right)^{-1} * \\ &\quad * \left(\left(\boldsymbol{\Psi}_{t+1,n}^{\beta(r+1)}\right)^{-1}\boldsymbol{\mu}_{t+1,n}^{\beta(r+1)} + \left(\boldsymbol{\Psi}_{t+1,n}^{\iota(r+1)}\right)^{-1}\boldsymbol{\mu}_{t+1,n}^{\iota(r+1)}\right) \quad \text{(C.15)}\end{aligned}$$

Thus obtaining:

$$\beta\left(\boldsymbol{S}_{t,n}\right) \propto \mathcal{N}\left(\boldsymbol{S}_{t,n}; \boldsymbol{\mu}_{t,n}^{\beta(r+1)}, \boldsymbol{\Psi}_{t,n}^{\beta(r+1)}\right) \quad \text{(C.16)}$$

The final marginal distribution By combining the forward and backward recursions we get:

$$\begin{aligned}q\left(\boldsymbol{S}_{t,n}\right) &\propto \phi\left(\boldsymbol{S}_{t,n}\right)\beta\left(\boldsymbol{S}_{t,n}\right) \\ &\propto \mathcal{N}\left(\boldsymbol{S}_{t,n}; \boldsymbol{\mu}_{t,n}^{\phi(r+1)}, \boldsymbol{\Psi}_{t,n}^{\phi(r+1)}\right)\mathcal{N}\left(\boldsymbol{S}_{t,n}; \boldsymbol{\mu}_{t,n}^{\beta(r+1)}, \boldsymbol{\Psi}_{t,n}^{\beta(r+1)}\right)\end{aligned}$$

which leads to:

$$\boxed{q\left(\boldsymbol{S}_{t,n}\right) = \mathcal{N}\left(\boldsymbol{S}_{t,n}; \boldsymbol{\mu}_{t,n}^{(r+1)}, \boldsymbol{\Psi}_{t,n}^{(r+1)}\right)} \quad \text{(C.17)}$$

with

$$\left(\boldsymbol{\Psi}_{t,n}^{(r+1)}\right)^{-1} = \left(\boldsymbol{\Psi}_{t,n}^{\beta(r+1)}\right)^{-1} + \left(\boldsymbol{\Psi}_{t,n}^{\phi(r+1)}\right)^{-1} \quad \text{(C.18)}$$

$$\boldsymbol{\mu}_{t,n}^{(r+1)} = \boldsymbol{\Psi}_{t,n}^{(r+1)}\left(\left(\boldsymbol{\Psi}_{t,n}^{\beta(r+1)}\right)^{-1}\boldsymbol{\mu}_{t,n}^{\beta(r+1)} + \left(\boldsymbol{\Psi}_{t,n}^{\phi(r+1)}\right)^{-1}\boldsymbol{\mu}_{t,n}^{\phi(r+1)}\right) \quad \text{(C.19)}$$

The coupled joint distribution When dealing with the M step, we will need the distribution $q\left(\boldsymbol{S}_{t,n}, \boldsymbol{S}_{t-1,n}\right)$ for the estimation of $\boldsymbol{\Lambda}_n$. If we marginalize (C.7) with respect to all variables except $\boldsymbol{S}_{t,n}$ and $\boldsymbol{S}_{t-1,n}$, we obtain the well-known formula:

$$\begin{aligned}q\left(\boldsymbol{S}_{t,n}, \boldsymbol{S}_{t-1,n}\right) &\propto \phi\left(\boldsymbol{S}_{t-1,n}\right)\iota\left(\boldsymbol{S}_{t,n}\right)T\left(\boldsymbol{S}_{t,n}, \boldsymbol{S}_{t-1,n}\right)\beta\left(\boldsymbol{S}_{t,n}\right) \\ &\propto \mathcal{N}\left(\boldsymbol{S}_{t-1,n}; \boldsymbol{\mu}_{t-1,n}^{\phi(r+1)}, \boldsymbol{\Psi}_{t-1,n}^{\phi(r+1)}\right)\mathcal{N}\left(\boldsymbol{S}_{t,n}; \boldsymbol{\mu}_{t,n}^{\iota(r+1)}, \boldsymbol{\Psi}_{t,n}^{\iota(r+1)}\right) * \\ &\quad * \mathcal{N}\left(\boldsymbol{S}_{t,n}; \mathbf{A}\boldsymbol{S}_{t-1,n}, \boldsymbol{\Lambda}_{n}^{(r)}\right)\mathcal{N}\left(\boldsymbol{S}_{t,n}; \boldsymbol{\mu}_{t,n}^{\beta(r+1)}, \boldsymbol{\Psi}_{t,n}^{\beta(r+1)}\right) \quad \text{(C.20)}\end{aligned}$$

If we develop the terms in the exponential, we realize that the quadratic term is:

$$\begin{aligned}
&\boldsymbol{P}_{t-1,n}^{\top}\left(\left(\boldsymbol{\Psi}_{t-1,n}^{\phi(r)}\right)^{-1}+\mathbf{A}^{\top}\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}\mathbf{A}\right)\boldsymbol{P}_{t-1,n}+\\
&+\boldsymbol{P}_{t,n}^{\top}\left(\left(\boldsymbol{\Psi}_{t,n}^{\iota(r+1)}\right)^{-1}+\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}+\left(\boldsymbol{\Psi}_{t,n}^{\beta(r+1)}\right)^{-1}\right)\boldsymbol{P}_{t,n}-\\
&-2\boldsymbol{P}_{t-1,n}^{\top}\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}\mathbf{A}\boldsymbol{P}_{t,n}
\end{aligned}$$

leading to the following covariance matrix for the joint distribution:

$$\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}=\begin{pmatrix}\left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{11} & \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{12}\\ \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{21} & \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{22}\end{pmatrix}^{-1} \tag{C.21}$$

with

$$\begin{aligned}
\left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{11}&=\left(\boldsymbol{\Psi}_{t,n}^{\iota(r+1)}\right)^{-1}+\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}+\left(\boldsymbol{\Psi}_{t,n}^{\beta(r+1)}\right)^{-1}\\
\left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{12}&=-\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}\mathbf{A}\\
\left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{21}&=-\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}\mathbf{A}\\
\left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{22}&=\left(\boldsymbol{\Psi}_{t-1,n}^{\phi(r)}\right)^{-1}+\mathbf{A}^{\top}\left(\boldsymbol{\Lambda}_{n}^{(r)}\right)^{-1}\mathbf{A}
\end{aligned}$$

The linear term is:

$$\begin{aligned}
&-2\left(\left(\boldsymbol{\mu}_{t,n}^{\beta(r+1)}\right)^{\top}\left(\boldsymbol{\Psi}_{t,n}^{\beta(r+1)}\right)^{-1}+\left(\boldsymbol{\mu}_{t,n}^{\iota(r+1)}\right)^{\top}\left(\boldsymbol{\Psi}_{t,n}^{\iota(r+1)}\right)^{-1}\right)\boldsymbol{S}_{t,n}-\\
&-2\left(\boldsymbol{\mu}_{t-1,n}^{\phi(r+1)}\right)^{\top}\left(\boldsymbol{\Psi}_{t-1,n}^{\phi(r+1)}\right)^{-1}\boldsymbol{S}_{t-1,n}
\end{aligned}$$

and therefore set the mean of the joint distribution as:

$$\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}=\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\begin{pmatrix}\left(\boldsymbol{\Psi}_{t,n}^{\beta(r+1)}\right)^{-1}\boldsymbol{\mu}_{t,n}^{\beta(r+1)}+\left(\boldsymbol{\Psi}_{t,n}^{\iota(r+1)}\right)^{-1}\boldsymbol{\mu}_{t,n}^{\iota(r+1)}\\ \left(\boldsymbol{\Psi}_{t-1,n}^{\phi(r+1)}\right)^{-1}\boldsymbol{\mu}_{t-1,n}^{\phi(r+1)}\end{pmatrix} \tag{C.22}$$

Here, we make some considerations. The coupled joint distribution following a Gaussian distribution:

$$q\left(\boldsymbol{S}_{t,n},\boldsymbol{S}_{t-1,n}\right)\propto\mathcal{N}\left(\begin{pmatrix}\boldsymbol{S}_{t,n}\\ \boldsymbol{S}_{t-1,n}\end{pmatrix};\boldsymbol{\mu}_{t,n}^{\zeta(r+1)},\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right) \tag{C.23}$$

with

$$\begin{pmatrix} \boldsymbol{S}_{t,n} \\ \boldsymbol{S}_{t-1,n} \end{pmatrix} = \begin{pmatrix} \boldsymbol{P}_{t,n} \\ \boldsymbol{V}_{t,n} \\ \boldsymbol{P}_{t-1,n} \\ \boldsymbol{V}_{t-1,n} \end{pmatrix}, \qquad \boldsymbol{\mu}_{t,n}^{\zeta(r+1)} = \begin{pmatrix} \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_1 \\ \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_2 \\ \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_2 \\ \left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_4 \end{pmatrix}$$

and

$$\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)} = \begin{pmatrix} \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{11} & \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{12} & \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{13} & \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{14} \\ \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{21} & \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{22} & \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{23} & \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{24} \\ \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{31} & \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{32} & \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{33} & \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{34} \\ \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{41} & \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{42} & \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{43} & \left(\boldsymbol{\Psi}_{t,n}^{\zeta(r+1)}\right)_{44} \end{pmatrix}$$

Indeed, all these parameters are block matrices.

C.2.2 Maximization step

In the M step, we have to evaluate the new model's parameters from the expected complete-data log-likelihood function:

$$\theta^{(r+1)} = \arg\max_{\theta} \mathcal{Q}\left(\theta, \theta^{(r)}\right).$$

The developed Q-function Starting from the expected complete-data log-likelihood function and the factorization done in the previous step, we can write the first one in the following way:

$$\begin{aligned} \mathcal{Q}\left(\theta, \theta^{(r)}\right) &= \mathrm{E}_{q(Z_{t,k})q(\boldsymbol{S}_t)}\left\{\ln p\left(Z, \boldsymbol{S}_t, \boldsymbol{f}|\theta\right)\right\} \\ &= \sum_{t,k=1}^{T,K_t} \mathrm{E}_{q(Z_{tk})}\left\{\ln p\left(Z_{tk}|\theta\right)\right\} + \mathrm{E}_{q(Z_{tk})q(\boldsymbol{S}_t)}\left\{\ln p\left(\boldsymbol{f}_{tk}|Z_{tk}, \boldsymbol{S}_t, \theta\right)\right\} + \\ &\quad + \sum_{n=1}^{N}\sum_{t=2}^{T} \mathrm{E}_{q(\boldsymbol{P}_{t,n}, \boldsymbol{P}_{t-1,n})}\left\{\ln p\left(\boldsymbol{P}_{t-1,n}|\boldsymbol{P}_{t,n}, \theta\right)\right\} + \\ &\quad + \sum_{n=1}^{N} \mathrm{E}_{q(\boldsymbol{P}_{1,n})}\left\{\ln p\left(\boldsymbol{P}_{1,n}|\theta\right)\right\} \end{aligned} \tag{C.24}$$

In C.24 there is a linear combination of four terms in which each depends on only one different parameter: taking the derivative with respect to one parameter, only one term is taken into account and the others are skipped, because they become zero. Since the first two terms do not depend on $\boldsymbol{S}$, their evaluation is totally equal to results found for the model with only positions. Neverthless, we report here their definitions and results, and we compute the remains parameters.

The parameter π

$$\mathcal{Q}\left(\pi, \pi^{(r)}\right) = \sum_{t,k=1}^{T,K_t} \mathrm{E}_{q\left(Z_{t,k}\right)} \left\{\ln p\left(Z_{t,k}|\pi_{t,n}\right)\right\} \qquad \Longrightarrow \qquad \boxed{\pi_{t,n} = \frac{\sum_{k=1}^{K_t} \eta_{t,k,n}^{Z(r+1)}}{K_t}}$$

The parameter $\boldsymbol{\Sigma}$

$$\mathcal{Q}\left(\boldsymbol{\Sigma}, \Psi^{(r)}\right) = \sum_{t,k=1}^{T,K_t} \mathrm{E}_{q(Z_t,k)q(\boldsymbol{P}_t)} \left\{\ln p\left(\boldsymbol{f}_{t,k}|Z_{t,k}, \boldsymbol{P}_t, \boldsymbol{\Sigma}\right)\right\}$$

$$\Downarrow$$

$$\boxed{\boldsymbol{\Sigma}_{t,n}^{(r+1)} = \boldsymbol{\Psi}_{t,n} + \frac{\sum_{k=1}^{K_t} \eta_{t,k,n} \left(\boldsymbol{\mu}_{t,n} - \boldsymbol{f}_{t,k}\right) \left(\boldsymbol{\mu}_{t,n} - \boldsymbol{f}_{t,k}\right)^{\top}}{\sum_{k=1}^{K_t} \eta_{t,k,n}}}$$

The parameters $\mathbf{\Lambda}_n^P$ and $\mathbf{\Lambda}_n^V$ We will now derive the update equations for $\mathbf{\Lambda}_n^P$ and $\mathbf{\Lambda}_n^V$. The terms of C.24 related to the transition matrix are:

$$\begin{aligned}
\mathcal{Q}_{\mathbf{\Lambda}_n^P,\mathbf{\Lambda}_n^V} &= \sum_{t=2}^{T} \mathrm{E}_{q(\boldsymbol{S}_{t,n},\boldsymbol{S}_{t-1,n})} \left\{ \ln p\left(\boldsymbol{S}_{t,n} | \boldsymbol{S}_{t-1,n}, \theta\right) \right\} \\
&= \sum_{t=2}^{T} \mathrm{E}_{q(\boldsymbol{S}_{t,n},\boldsymbol{S}_{t-1,n})} \left\{ \ln \mathcal{N}\left(\boldsymbol{S}_{t,n}; \mathbf{A}\boldsymbol{S}_{t-1,n}, \mathbf{\Lambda}_n\right) \right\} \\
&\overset{\mathbf{\Lambda}_n^P,\mathbf{\Lambda}_n^V}{=} \sum_{t=2}^{T} \mathrm{E}_{q(\boldsymbol{S}_{t,n},\boldsymbol{S}_{t-1,n})} \left\{ \ln |\mathbf{\Lambda}_n|^{-1/2} - \left\| \boldsymbol{S}_{t,n} - \mathbf{A}\boldsymbol{S}_{t-1,n} \right\|_{\mathbf{\Lambda}_n} \right\} \\
&= -\frac{1}{2} \left((T-1) \ln |\mathbf{\Lambda}_n| + \sum_{t=2}^{T} \mathrm{E}_{q(\boldsymbol{S}_{t,n},\mathbf{A}\boldsymbol{S}_{t-1,n})} \left\{ \left\| \boldsymbol{S}_{t,n} - \mathbf{A}\boldsymbol{S}_{t-1,n} \right\|_{\mathbf{\Lambda}_n} \right\} \right)
\end{aligned} \tag{C.25}$$

First of all, we analyze:

$$\begin{aligned}
\ln |\mathbf{\Lambda}_n| &= -\ln \left| (\mathbf{\Lambda}_n)^{-1} \right| = -\ln \left(\left| \left(\mathbf{\Lambda}_n^S\right)^{-1} \right| * \left| \left(\mathbf{\Lambda}_n^V\right)^{-1} \right| \right) = \ln \left(\left|\mathbf{\Lambda}_n^S\right| * \left|\mathbf{\Lambda}_n^V\right| \right) \\
&= \ln \left|\mathbf{\Lambda}_n^S\right| + \ln \left|\mathbf{\Lambda}_n^V\right|
\end{aligned}$$

Then,

$$\mathbf{A}\boldsymbol{S}_{t-1,n} = \mathbf{A} \begin{pmatrix} \boldsymbol{P}_{t-1,n} \\ \boldsymbol{V}_{t-1,n} \end{pmatrix} = \begin{pmatrix} \boldsymbol{P}_{t-1,n} + \boldsymbol{V}_{t-1,n} \\ \boldsymbol{V}_{t-1,n} \end{pmatrix}$$

$$\Rightarrow \left(\boldsymbol{S}_{t,n} - \mathbf{A}\boldsymbol{S}_{t-1,n}\right)^\top \left(\mathbf{\Lambda}_n\right)^{-1} \left(\boldsymbol{S}_{t,n} - \mathbf{A}\boldsymbol{S}_{t-1,n}\right) =$$

$$\begin{aligned}
&= \left(\begin{pmatrix} \boldsymbol{P}_{t,n} \\ \boldsymbol{V}_{t,n} \end{pmatrix} - \begin{pmatrix} \boldsymbol{P}_{t-1,n} + \boldsymbol{V}_{t-1,n} \\ \boldsymbol{V}_{t-1,n} \end{pmatrix} \right)^\top \begin{pmatrix} \left(\mathbf{\Lambda}_n^S\right)^{-1} & -\left(\mathbf{\Lambda}_n^S\right)^{-1} \\ -\left(\mathbf{\Lambda}_n^S\right)^{-1} & \left(\mathbf{\Lambda}_n^S\right)^{-1} + \left(\mathbf{\Lambda}_n^V\right)^{-1} \end{pmatrix} * \\
&\quad * \left(\begin{pmatrix} \boldsymbol{S}_{t,n} \\ \boldsymbol{V}_{t,n} \end{pmatrix} - \begin{pmatrix} \boldsymbol{S}_{t-1,n} + \boldsymbol{V}_{t-1,n} \\ \boldsymbol{V}_{t-1,n} \end{pmatrix} \right) =
\end{aligned}$$

$$\begin{aligned}
&= \left(\boldsymbol{P}_{t,n} - \boldsymbol{P}_{t-1,n} - \boldsymbol{V}_{t-1,n}\right)^\top \left(\mathbf{\Lambda}_n^P\right)^{-1} \left(\boldsymbol{P}_{t,n} - \boldsymbol{P}_{t-1,n} - \boldsymbol{V}_{t-1,n}\right) - \\
&\quad - 2 \left(\boldsymbol{P}_{t,n} - \boldsymbol{P}_{t-1,n} - \boldsymbol{V}_{t-1,n}\right)^\top \left(\mathbf{\Lambda}_n^P\right)^{-1} \left(\boldsymbol{V}_{t,n} - \boldsymbol{V}_{t-1,n}\right) + \\
&\quad + \left(\boldsymbol{V}_{t,n} - \boldsymbol{V}_{t-1,n}\right)^\top \left(\left(\mathbf{\Lambda}_n^P\right)^{-1} + \left(\mathbf{\Lambda}_n^V\right)^{-1} \right) \left(\boldsymbol{V}_{t,n} - \boldsymbol{V}_{t-1,n}\right)
\end{aligned}$$

We now take the expectations:

$$\text{tr}\left(\left(\mathbf{\Lambda}_n^S\right)^{-1}\mathbf{E}_{t,n}^{S^{(r+1)}}\right)+\text{tr}\left(\left(\mathbf{\Lambda}_n^V\right)^{-1}\mathbf{E}_{t,n}^{V^{(r+1)}}\right)$$

with $\mathbf{E}_{t,n}^{S^{(r+1)}}$ and $\mathbf{E}_{t,n}^{V^{(r+1)}}$ being the energy matrices with the following expression:

$$\begin{aligned}
\mathbf{E}_{t,n}^{P^{(r+1)}} =&\mathbf{Q}_{t,n}^{11}+\mathbf{Q}_{t,n}^{33}+\mathbf{Q}_{t,n}^{44}-2\mathbf{Q}_{t,n}^{13}-2\mathbf{Q}_{t,n}^{14}+2\mathbf{Q}_{t,n}^{34}-2\mathbf{Q}_{t,n}^{12}+2\mathbf{Q}_{t,n}^{14}+\\
&+2\mathbf{Q}_{t,n}^{23}-2\mathbf{Q}_{t,n}^{34}+2\mathbf{Q}_{t,n}^{24}-2\mathbf{Q}_{t,n}^{44}+\mathbf{Q}_{t,n}^{44}+\mathbf{Q}_{t,n}^{22}-2\mathbf{Q}_{t,n}^{24}=\\
=&\mathbf{Q}_{t,n}^{11}+\mathbf{Q}_{t,n}^{33}+\mathbf{Q}_{t,n}^{22}-2\mathbf{Q}_{t,n}^{13}-2\mathbf{Q}_{t,n}^{12}+2\mathbf{Q}_{t,n}^{23}=\\
=&\left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{11}+\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_1\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_1^\top+\left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{22}+\\
&+\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_2\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_2^\top+\left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{33}+\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_3\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_3^\top-\\
&-\left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{12}-\left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{21}-\left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{13}-\left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{31}-\\
&-\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_1\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_2^\top-\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_2\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_1^\top-\\
&-\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_3\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_1^\top-\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_1\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_3^\top+\\
&+\left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{23}+\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_2\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_3^\top+\left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{32}+\\
&+\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_3\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_2^\top
\end{aligned}$$

$$\begin{aligned}
\mathbf{E}_{t,n}^{V^{(r+1)}} =&\left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{22}+\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_2\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_2^\top+\left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{44}+\\
&+\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_4\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_4^\top-\left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{14}-\left(\mathbf{\Psi}_{t,n}^{\zeta(r+1)}\right)_{41}-\\
&-\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_1\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_4^\top-\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_4\left(\boldsymbol{\mu}_{t,n}^{\zeta(r+1)}\right)_1^\top
\end{aligned}$$

where the subscribes here denote the quarter in the correspondant dimension. Therefore, we can rewrite (C.25) as:

$$\begin{aligned}
\mathcal{Q}_{\mathbf{\Lambda}_n^P,\mathbf{\Lambda}_n^V} &= \frac{1}{2}\left((T-1)\ln\left|\left(\mathbf{\Lambda}_n^P\right)^{-1}\right|-\sum_{t=2}^{T}\text{tr}\left(\left(\mathbf{\Lambda}_n^P\right)^{-1}\mathbf{E}_{t,n}^{S^{(r+1)}}\right)+\right.\\
&\left.+(T-1)\ln\left|\left(\mathbf{\Lambda}_n^V\right)^{-1}\right|-\sum_{t=2}^{T}\text{tr}\left(\left(\mathbf{\Lambda}_n^V\right)^{-1}\mathbf{E}_{t,n}^{V^{(r+1)}}\right)\right) \quad \text{(C.26)}
\end{aligned}$$

Taking now the derivative with respect to $\mathbf{\Lambda}_n^P$ and $\mathbf{\Lambda}_n^V$ respectively and cancelling it:

$$\boxed{\mathbf{\Lambda}_n^{P^{(r+1)}} = \frac{1}{T-1}\sum_{t=2}^{T}\mathbf{E}_{t,n}^{P^{(r+1)}}} \tag{C.27}$$

$$\boxed{\mathbf{\Lambda}_n^{V^{(r+1)}} = \frac{1}{T-1}\sum_{t=2}^{T}\mathbf{E}_{t,n}^{V^{(r+1)}}} \tag{C.28}$$

The parameters $\left(\boldsymbol{\nu}_n^P, \mathbf{\Omega}_n^P, \boldsymbol{\nu}_n^V, \mathbf{\Omega}_n^V\right)$ Finally, we evaluate the 4^{th} term of the developed Q-function C.24.

From C.17, we already know:

$$q\left(\boldsymbol{S}_{1,n}\right) = \mathcal{N}\left(\boldsymbol{S}_{1,n}; \boldsymbol{\mu}_{1,n}^{(r+1)}, \mathbf{\Psi}_{1,n}^{(r+1)}\right)$$

and we define and develop

$$\begin{aligned}
\mathcal{Q}_{\boldsymbol{\nu}_n^P, \mathbf{\Omega}_n^P, \boldsymbol{\nu}_n^V, \mathbf{\Omega}_n^V} &= \sum_{n=1}^{N} \mathrm{E}_{q(\boldsymbol{S}_{1,n})}\left\{\ln p\left(\boldsymbol{S}_{1,n}|\boldsymbol{\nu}_n, \mathbf{\Omega}_n\right)\right\} \\
&= \sum_{n=1}^{N} \mathrm{E}_{q(\boldsymbol{S}_{1,n})}\left\{\ln \mathcal{N}\left(\boldsymbol{S}_{1,n}; \boldsymbol{\nu}_n, \mathbf{\Omega}_n\right)\right\} \\
&= \sum_{n=1}^{N} \mathrm{E}_{q(\boldsymbol{S}_{1,n})}\left\{-\ln 2\pi^{D/2} + \frac{1}{2}\ln\left|\left(\mathbf{\Omega}_n\right)^{-1}\right| - \right. \\
&\quad \left. -\frac{1}{2}\left[\left(\boldsymbol{S}_{1,n} - \boldsymbol{\nu}_n\right)^{\top}\left(\mathbf{\Omega}_n\right)^{-1}\left(\boldsymbol{S}_{1,n} - \boldsymbol{\nu}_n\right)\right]\right\} \\
&= -\ln 2\pi^{D/2} + \sum_{n=1}^{N}\left[\frac{1}{2}\ln\left|\left(\mathbf{\Omega}_n\right)^{-1}\right| - \right. \\
&\quad \left. -\frac{1}{2}\mathrm{E}_{q(\boldsymbol{S}_{1,n})}\left\{\left(\boldsymbol{S}_{1,n} - \boldsymbol{\nu}_n\right)^{\top}\left(\mathbf{\Omega}_n\right)^{-1}\left(\boldsymbol{S}_{1,n} - \boldsymbol{\nu}_n\right)\right\}\right] \\
&= -\ln 2\pi^{D/2} + \sum_{n=1}^{N}\left[\frac{1}{2}\ln\left|\left(\mathbf{\Omega}_n\right)^{-1}\right| - \right. \\
&\quad -\frac{1}{2}\left[\mathrm{tr}\left(\mathbf{\Omega}_n^{-1}\left(\mathbf{\Psi}_{1,n}^{(r+1)} + \boldsymbol{\mu}_{1,n}^{(r+1)}\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)^{\top}\right)\right) - \right. \\
&\quad \left.\left. -2\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)^{\top}\mathbf{\Omega}_n^{-1}\boldsymbol{\nu}_n + \boldsymbol{\nu}_n^{\top}\mathbf{\Omega}_n^{-1}\boldsymbol{\nu}_n\right]\right]
\end{aligned}$$

First of all, recalling that $\mathbf{\Omega}_n$ is a block diagonal matrix:

$$\mathbf{\Omega}_n = \left(\begin{array}{c|c} \mathbf{\Omega}_n^P & 0 \\ \hline 0 & \mathbf{\Omega}_n^V \end{array}\right) \quad \Rightarrow \quad (\mathbf{\Omega}_n)^{-1} = \left(\begin{array}{c|c} \left(\mathbf{\Omega}_n^P\right)^{-1} & 0 \\ \hline 0 & \left(\mathbf{\Omega}_n^V\right)^{-1} \end{array}\right)$$

and we now analyze the two terms inside the summation,

$$\ln\left|(\mathbf{\Omega}_n)^{-1}\right| = \ln\left(\left|\left(\mathbf{\Omega}_n^P\right)^{-1}\right|\left|\left(\mathbf{\Omega}_n^V\right)^{-1}\right|\right) = \ln\left|\left(\mathbf{\Omega}_n^P\right)^{-1}\right| + \ln\left|\left(\mathbf{\Omega}_n^V\right)^{-1}\right|$$

and

$$\begin{aligned}
&\operatorname{tr}\left(\mathbf{\Omega}_n^{-1}\left(\mathbf{\Psi}_{1,n}^{(r+1)} + \boldsymbol{\mu}_{1,n}^{(r+1)}\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)^\top\right)\right) - 2\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)^\top \mathbf{\Omega}_n^{-1}\boldsymbol{\nu}_n + \boldsymbol{\nu}_n^\top\mathbf{\Omega}_n^{-1}\boldsymbol{\nu}_n = \\
=&\operatorname{tr}\left(\left(\mathbf{\Omega}_n^P\right)^{-1}\left(\left(\mathbf{\Psi}_{1,n}^{(r+1)}\right)_{11} + \left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_1\left(\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_1\right)^\top\right)\right) - \\
&- 2\left(\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_1\right)^\top\left(\mathbf{\Omega}_n^P\right)^{-1}\boldsymbol{\nu}_n^P + \left(\boldsymbol{\nu}_n^P\right)^\top\left(\mathbf{\Omega}_n^P\right)^{-1}\boldsymbol{\nu}_n^P + \\
&+ \operatorname{tr}\left(\left(\mathbf{\Omega}_n^V\right)^{-1}\left(\left(\mathbf{\Psi}_{1,n}^{(r+1)}\right)_{22} + \left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_2\left(\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_2\right)^\top\right)\right) - \\
&- 2\left(\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_2\right)^\top\left(\mathbf{\Omega}_n^V\right)^{-1}\boldsymbol{\nu}_n^V + \left(\boldsymbol{\nu}_n^V\right)^\top\left(\mathbf{\Omega}_n^V\right)^{-1}\boldsymbol{\nu}_n^V
\end{aligned}$$

Finally,

$$\mathcal{Q}_{\boldsymbol{\nu}_n^P,\mathbf{\Omega}_n^P,\boldsymbol{\nu}_n^V,\mathbf{\Omega}_n^V} =$$

$$\begin{aligned}
&-\ln 2\pi^{D/2} + \sum_{n=1}^{N}\left[\frac{1}{2}\ln\left|\left(\mathbf{\Omega}_n^P\right)^{-1}\right| + \frac{1}{2}\left[2\left(\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_1\right)^\top\left(\mathbf{\Omega}_n^P\right)^{-1}\boldsymbol{\nu}_n^P + \right.\right. \\
&+ \left(\boldsymbol{\nu}_n^P\right)^\top\left(\mathbf{\Omega}_n^P\right)^{-1}\boldsymbol{\nu}_n^P - \operatorname{tr}\left(\left(\mathbf{\Omega}_n^P\right)^{-1}\left(\left(\mathbf{\Psi}_{1,n}^{(r+1)}\right)_{11} + \left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_1\left(\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_1\right)^\top\right)\right)\left.\right] + \\
&+\frac{1}{2}\ln\left|\left(\mathbf{\Omega}_n^V\right)^{-1}\right| - \frac{1}{2}\left[\operatorname{tr}\left(\left(\mathbf{\Omega}_n^V\right)^{-1}\left(\left(\mathbf{\Psi}_{1,n}^{(r+1)}\right)_{22} + \left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_2\left(\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_2\right)^\top\right)\right) - \right. \\
&\left.\left.-2\left(\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_2\right)^\top\left(\mathbf{\Omega}_n^V\right)^{-1}\boldsymbol{\nu}_n^V + \left(\boldsymbol{\nu}_n^V\right)^\top\left(\mathbf{\Omega}_n^V\right)^{-1}\boldsymbol{\nu}_n^V\right]\right] \qquad \text{(C.29)}
\end{aligned}$$

To compute $\left(\boldsymbol{\nu}_n^P, \mathbf{\Omega}_n^P, \boldsymbol{\nu}_n^V, \mathbf{\Omega}_n^V\right)$, we set the derivative of C.29 with respect to each of the parameters to zero and after rearranging, we obtain the following formulas:

$$\frac{\partial\mathcal{Q}_{\boldsymbol{\nu}_n^P,\mathbf{\Omega}_n^P,\boldsymbol{\nu}_n^V,\mathbf{\Omega}_n^V}}{\partial\boldsymbol{\nu}_n^P} = -2\left(\mathbf{\Omega}_n^P\right)^{-1}\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_1 + 2\left(\mathbf{\Omega}_n^P\right)^{-1}\boldsymbol{\nu}_n^P = 0$$

$$\boxed{\boldsymbol{\nu}_n^P = \left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_1} \tag{C.30}$$

$$\begin{aligned}\frac{\partial \mathcal{Q}_{\boldsymbol{\nu}_n^P,\boldsymbol{\Omega}_n^P,\boldsymbol{\nu}_n^V,\boldsymbol{\Omega}_n^V}}{\partial (\boldsymbol{\Omega}_n^P)^{-1}} =& \frac{\boldsymbol{\Omega}_n^P}{2} - \frac{1}{2}\left[\left(\boldsymbol{\Psi}_{1,n}^{(r+1)}\right)_{11} + \left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_1 \left(\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_1\right)^\top - \right.\\ & \left. -2\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_1 \left(\boldsymbol{\nu}_n^P\right)\top + \boldsymbol{\nu}_n^P \left(\boldsymbol{\nu}_n^P\right)\top\right]\\ =& 0\end{aligned}$$

$$\begin{aligned}\boldsymbol{\Omega}_n^P \quad =& \quad \left(\boldsymbol{\Psi}_{1,n}^{(r+1)}\right)_{11} + \left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_1 \left(\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_1\right)^\top - \\ & -2\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_1 \left(\boldsymbol{\nu}_n^P\right)^\top + \boldsymbol{\nu}_n^P \left(\boldsymbol{\nu}_n^P\right)^\top\\ \overset{(C.30)}{=}& \quad \left(\boldsymbol{\Psi}_{1,n}^{(r+1)}\right)_{11}\end{aligned} \tag{C.31}$$

$$\frac{\partial \mathcal{Q}_{\boldsymbol{\nu}_n^P,\boldsymbol{\Omega}_n^P,\boldsymbol{\nu}_n^V,\boldsymbol{\Omega}_n^V}}{\partial \boldsymbol{\nu}_n^V} = -2\left(\boldsymbol{\Omega}_n^V\right)^{-1}\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_2 + 2\left(\boldsymbol{\Omega}_n^V\right)^{-1}\boldsymbol{\nu}_n^V = 0$$

$$\boxed{\boldsymbol{\nu}_n^V = \left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_2} \tag{C.32}$$

$$\begin{aligned}\frac{\partial \mathcal{Q}_{\boldsymbol{\nu}_n^P,\boldsymbol{\Omega}_n^P,\boldsymbol{\nu}_n^V,\boldsymbol{\Omega}_n^V}}{\partial (\boldsymbol{\Omega}_n^V)^{-1}} =& \frac{\boldsymbol{\Omega}_n^V}{2} - \frac{1}{2}\left[\left(\boldsymbol{\Psi}_{1,n}^{(r+1)}\right)_{22} + \left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_2 \left(\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_2\right)^\top - \right.\\ & \left. -2\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_2 \left(\boldsymbol{\nu}_n^V\right)\top + \boldsymbol{\nu}_n^V \left(\boldsymbol{\nu}_n^V\right)\top\right]\\ =& 0\end{aligned}$$

$$\begin{aligned}\boldsymbol{\Omega}_n^V \quad =& \quad \left(\boldsymbol{\Psi}_{1,n}^{(r+1)}\right)_{22} + \left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_2 \left(\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_2\right)^\top - \\ & -2\left(\boldsymbol{\mu}_{1,n}^{(r+1)}\right)_2 \left(\boldsymbol{\nu}_n^V\right)\top + \boldsymbol{\nu}_n^V \left(\boldsymbol{\nu}_n^V\right)\top\\ \overset{(C.32)}{=}& \quad \left(\boldsymbol{\Psi}_{1,n}^{(r+1)}\right)_{22}\end{aligned} \tag{C.33}$$

Appendix D

Inverting the model's parameter $\mathbf{\Lambda}$

In chapter 3, we defined the source positions and velocities by modelling them as Gaussian distributions. Besides, we combined the two normals into only one defining the covariance matrix $\mathbf{\Lambda}$.

Since $\mathbf{\Lambda}^{-1}$ is a block matrix, we use the following formulas from [Petersen and Pedersen, 2012]

$$\left[\begin{array}{c|c} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \hline \mathbf{A}_{21} & \mathbf{A}_{22} \end{array}\right]^{-1} = \left[\begin{array}{c|c} \mathbf{C}_1^{-1} & -\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{C}_2^{-1} \\ \hline -\mathbf{C}_2^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & \mathbf{C}_2^{-1} \end{array}\right]$$

with

$$\begin{aligned} \mathbf{C}_1 &= \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21} \\ \mathbf{C}_2 &= \mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12} \end{aligned}$$

So, as far as concerned our covariance:

$$\mathbf{\Lambda}^{-1} = \begin{pmatrix} \mathbf{\Sigma}_S^{-1} & -\mathbf{\Sigma}_S^{-1} \\ -\mathbf{\Sigma}_S^{-1} & \mathbf{\Sigma}_S^{-1} + \mathbf{\Sigma}_V^{-1} \end{pmatrix}$$

$$\Downarrow$$

$$\mathbf{\Lambda} = \left(\begin{array}{c|c} \mathbf{C}_1^{-1} & \mathbf{\Sigma}_S\mathbf{\Sigma}_S^{-1}\mathbf{C}_2^{-1} \\ \hline \mathbf{C}_2^{-1}\mathbf{\Sigma}_S^{-1}\mathbf{\Sigma}_S & \mathbf{C}_2^{-1} \end{array}\right) = \left(\begin{array}{c|c} \mathbf{C}_1^{-1} & \mathbf{C}_2^{-1} \\ \hline \mathbf{C}_2^{-1} & \mathbf{C}_2^{-1} \end{array}\right)$$

with

$$\begin{aligned} \mathbf{C}_1 &= \mathbf{\Sigma}_S^{-1} - \mathbf{\Sigma}_S^{-1}\left(\mathbf{\Sigma}_S^{-1} + \mathbf{\Sigma}_V^{-1}\right)^{-1}\mathbf{\Sigma}_S^{-1} = (\mathbf{\Sigma}_S + \mathbf{\Sigma}_V)^{-1} \\ \mathbf{C}_2 &= \left(\mathbf{\Sigma}_S^{-1} + \mathbf{\Sigma}_V^{-1}\right)^{-1} - \mathbf{\Sigma}_S^{-1}\mathbf{\Sigma}_S\mathbf{\Sigma}_S^{-1} = \mathbf{\Sigma}_V^{-1} \end{aligned}$$

Finally,

$$\Rightarrow \mathbf{\Lambda} = \begin{pmatrix} \mathbf{\Sigma}_S + \mathbf{\Sigma}_V & \mathbf{\Sigma}_V \\ \mathbf{\Sigma}_V & \mathbf{\Sigma}_V \end{pmatrix} \tag{D.1}$$

Now, we check the inverse:

$$\begin{aligned} \mathbf{\Lambda}^{-1}\mathbf{\Lambda} &= \begin{pmatrix} \mathbf{\Sigma}_S^{-1} & -\mathbf{\Sigma}_S^{-1} \\ -\mathbf{\Sigma}_S^{-1} & \mathbf{\Sigma}_S^{-1} + \mathbf{\Sigma}_V^{-1} \end{pmatrix}\begin{pmatrix} \mathbf{\Sigma}_S + \mathbf{\Sigma}_V & \mathbf{\Sigma}_V \\ \mathbf{\Sigma}_V & \mathbf{\Sigma}_V \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{I} + \mathbf{\Sigma}_S^{-1}\mathbf{\Sigma}_V - \mathbf{\Sigma}_S^{-1}\mathbf{\Sigma}_V & \mathbf{\Sigma}_S^{-1}\mathbf{\Sigma}_V - \mathbf{\Sigma}_S^{-1}\mathbf{\Sigma}_V \\ -\mathbf{I} - \mathbf{\Sigma}_S^{-1}\mathbf{\Sigma}_V + \mathbf{\Sigma}_S^{-1}\mathbf{\Sigma}_V + \mathbf{I} & \mathbf{I} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{I} \end{pmatrix} \\ &= \mathbf{I} \end{aligned} \tag{D.2}$$

Appendix E

Kalman equations

Here, we relate equations for Linear Dynamical Systems in [Bishop, 2006] to our equations defined in the EM algorithm.
In general, LDS provides equations for parameters that are well-known as Kalman filter and Kalman smoother. In other words, in the E-step we derived a formulation that includes a forward and backward recursion. Each of them is related to Kalman filter and Kalman smoother respectively.

Kalman Filter

Starting from the forward recursion, we have reported from Bishop's book:

$$\boldsymbol{P}_{n-1} = \boldsymbol{A}\boldsymbol{V}_{n-1}\boldsymbol{A}^{\top} + \boldsymbol{\Gamma}$$

$$\boldsymbol{\mu}_n = \boldsymbol{A}\boldsymbol{\mu}_{n-1} + \boldsymbol{K}\left(x_n - \boldsymbol{C}\boldsymbol{A}\boldsymbol{\mu}_{n-1}\right)$$

$$\boldsymbol{V}_n = \left(\mathbf{I} - \boldsymbol{K}_n\boldsymbol{C}\right)\boldsymbol{P}_{n-1}$$

with the *Kalman gain matrix* defined as following:

$$\boldsymbol{K}_n = \boldsymbol{P}_{n-1}\boldsymbol{C}^{\top}\left(\boldsymbol{C}\boldsymbol{P}_{n-1}\boldsymbol{C}^{\top} + \boldsymbol{\Sigma}\right)^{-1}$$

From $\boldsymbol{V}_n$ to $\boldsymbol{\Psi}^{\phi}_{tn}$ In our case, we assumed

$$\boldsymbol{A} = \boldsymbol{C} = \mathbf{I}, \qquad \boldsymbol{\Gamma} = \boldsymbol{\Lambda}_n, \qquad \boldsymbol{\Sigma} = \boldsymbol{\Psi}^{\iota}_{tn}, \qquad \boldsymbol{P}_{n-1} = \boldsymbol{\Psi}^{\phi}_{t-1n} + \boldsymbol{\Lambda}_n$$

Substituting,

$$\boldsymbol{K}_n = \boldsymbol{P}_{n-1} \left(\boldsymbol{P}_{n-1} + \boldsymbol{\Psi}^{\iota}_{tn} \right)^{-1}$$

and

$$\begin{aligned}
\boldsymbol{V}_n &= \left(\mathbf{I} - \boldsymbol{K}_n \boldsymbol{C} \right) \boldsymbol{P}_{n-1} \\
&= \left[\mathbf{I} - \boldsymbol{P}_{n-1} \left(\boldsymbol{P}_{n-1} + \boldsymbol{\Psi}^{\iota}_{tn} \right)^{-1} \right] \boldsymbol{P}_{n-1} \\
&= \boldsymbol{P}_{n-1} - \boldsymbol{P}_{n-1} \left(\boldsymbol{P}_{n-1} + \boldsymbol{\Psi}^{\iota}_{tn} \right)^{-1} \boldsymbol{P}_{n-1} \\
&= \left(\boldsymbol{P}^{-1}_{n-1} + \left(\boldsymbol{\Psi}^{\iota}_{tn} \right)^{-1} \right)^{-1} \\
&= \boxed{\left(\left(\boldsymbol{\Psi}^{\iota}_{tn} \right)^{-1} + \left(\boldsymbol{\Psi}^{\phi}_{t-1n} + \boldsymbol{\Lambda}_n \right)^{-1} \right)^{-1} = \boldsymbol{\Psi}^{\phi}_{tn}} \\
&\Rightarrow \left(\boldsymbol{\Psi}^{\phi}_{tn} \right)^{-1} = \left(\boldsymbol{\Psi}^{\iota}_{tn} \right)^{-1} + \left(\boldsymbol{\Psi}^{\phi}_{t-1n} + \boldsymbol{\Lambda}_n \right)^{-1}
\end{aligned}$$

where we used the Woodbury identity from 3^{rd} to 4^{th} steps

$$\left(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C} \right)^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{B} \left(\boldsymbol{D} + \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B} \right)^{-1} \boldsymbol{C}\boldsymbol{A}^{-1}$$

From $\boldsymbol{\mu}_n$ to $\boldsymbol{\mu}^{\phi}_{tn}$ Besides the previous assumption, as well we now add

$$x_n = \boldsymbol{\mu}^{\iota}_{tn}$$

and so

$$
\begin{aligned}
\boldsymbol{\mu}_n &= \boldsymbol{A}\boldsymbol{\mu}_{n-1} + \boldsymbol{K}\left(x_n - \boldsymbol{C}\boldsymbol{A}\boldsymbol{\mu}_{n-1}\right) \\
&= \boldsymbol{\mu}_{n-1} + \boldsymbol{P}_{n-1}\left(\boldsymbol{P}_{n-1} + \boldsymbol{\Psi}_{tn}^{\iota}\right)^{-1}\left(\boldsymbol{\mu}_{tn}^{\iota} - \boldsymbol{\mu}_{n-1}\right) \\
&= \boldsymbol{\mu}_{n-1} + \boldsymbol{P}_{n-1}\left(\boldsymbol{P}_{n-1} + \boldsymbol{\Psi}_{tn}^{\iota}\right)^{-1}\boldsymbol{\mu}_{tn}^{\iota} - \boldsymbol{P}_{n-1}\left(\boldsymbol{P}_{n-1} + \boldsymbol{\Psi}_{tn}^{\iota}\right)^{-1}\boldsymbol{\mu}_{n-1} \\
&= \left[\mathbf{I} - \boldsymbol{P}_{n-1}\left(\boldsymbol{P}_{n-1} + \boldsymbol{\Psi}_{tn}^{\iota}\right)^{-1}\right]\boldsymbol{\mu}_{n-1} + \boldsymbol{P}_{n-1}\left(\boldsymbol{P}_{n-1} + \boldsymbol{\Psi}_{tn}^{\iota}\right)^{-1}\boldsymbol{\mu}_{tn}^{\iota} \\
&= \left[\left(\boldsymbol{P}_{n-1}^{-1} + \left(\boldsymbol{\Psi}_{tn}^{\iota}\right)^{-1}\right)^{-1}\boldsymbol{P}_{n-1}^{-1}\right]\boldsymbol{\mu}_{n-1} + \boldsymbol{P}_{n-1}\left(\boldsymbol{P}_{n-1} + \boldsymbol{\Psi}_{tn}^{\iota}\right)^{-1}\boldsymbol{\mu}_{tn}^{\iota} \\
&= \left(\left(\boldsymbol{\Psi}_{tn}^{\iota}\right)^{-1} + \boldsymbol{P}_{n-1}^{-1}\right)^{-1}\boldsymbol{P}_{n-1}^{-1}\boldsymbol{\mu}_{n-1} + \left[\mathbf{I} - \left(\left(\boldsymbol{\Psi}_{tn}^{\iota}\right)^{-1} + \boldsymbol{P}_{n-1}^{-1}\right)^{-1}\boldsymbol{P}_{n-1}^{-1}\right]\boldsymbol{\mu}_{tn}^{\iota} \\
&= \left(\left(\boldsymbol{\Psi}_{tn}^{\iota}\right)^{-1} + \boldsymbol{P}_{n-1}^{-1}\right)^{-1}\boldsymbol{P}_{n-1}^{-1}\boldsymbol{\mu}_{n-1} + \left(\left(\boldsymbol{\Psi}_{tn}^{\iota}\right)^{-1} + \boldsymbol{P}_{n-1}^{-1}\right)^{-1} \times \\
&\qquad \times \left[\left(\boldsymbol{\Psi}_{tn}^{\iota}\right)^{-1} + \boldsymbol{P}_{n-1}^{-1} - \boldsymbol{P}_{n-1}^{-1}\right]\boldsymbol{\mu}_{tn}^{\iota} \\
&= \left(\left(\boldsymbol{\Psi}_{tn}^{\iota}\right)^{-1} + \left(\boldsymbol{\Psi}_{t-1n}^{\phi} + \boldsymbol{\Lambda}_n\right)^{-1}\right)^{-1}\left[\left(\boldsymbol{\Psi}_{t-1n}^{\phi} + \boldsymbol{\Lambda}_n\right)^{-1}\boldsymbol{\mu}_{n-1} + \left(\boldsymbol{\Psi}_{tn}^{\iota}\right)^{-1}\boldsymbol{\mu}_{tn}^{\iota}\right] \\
&= \boxed{\boldsymbol{\Psi}_{tn}^{\phi}\left[\left(\boldsymbol{\Psi}_{t-1n}^{\phi} + \boldsymbol{\Lambda}_n\right)^{-1}\boldsymbol{\mu}_{n-1} + \left(\boldsymbol{\Psi}_{tn}^{\iota}\right)^{-1}\boldsymbol{\mu}_{tn}^{\iota}\right] = \boldsymbol{\mu}_{tn}^{\phi}}
\end{aligned}
$$

and the Woodbury identity is used again.

Kalman Smoother

If

$$\boldsymbol{A} = \boldsymbol{C} = \mathbf{I}, \qquad \boldsymbol{\Gamma} = \boldsymbol{\Lambda}_n, \qquad \boldsymbol{V}_n = \boldsymbol{\Psi}_{tn}^{\phi}$$

$$\hat{\boldsymbol{V}}_t = \left(\left(\boldsymbol{\Psi}_{tn}^{\phi}\right)^{-1} + \left(\boldsymbol{\Psi}_{tn}^{\beta}\right)^{-1}\right)^{-1}$$

we the smoother relationship:

$$
\begin{aligned}
\hat{\boldsymbol{V}}_t &= \boldsymbol{V}_t + \boldsymbol{V}_t\boldsymbol{P}_t^{-1}\left(\hat{\boldsymbol{V}}_{t+1} - \boldsymbol{P}_t\right)\boldsymbol{P}_t^{-1}\boldsymbol{V}_t \\
&= \boldsymbol{V}_t - \boldsymbol{V}_t\left(\boldsymbol{V}_t + \boldsymbol{\Omega}_t\right)^{-1}\boldsymbol{V}_t \\
&= \left(\boldsymbol{V}_t^{-1} + \boldsymbol{\Omega}_t^{-1}\right)^{-1}
\end{aligned}
$$

We went to know under which condition $\hat{\boldsymbol{V}}_{t+1} = \left((\boldsymbol{\Psi}^{\phi}_{t+1n})^{-1} + (\boldsymbol{\Psi}^{\beta}_{t+1n})^{-1}\right)^{-1}$ implies.

$$\begin{aligned}
&\hat{\boldsymbol{V}}_t = \left((\boldsymbol{\Psi}^{\phi}_{tn})^{-1} + (\boldsymbol{\Psi}^{\beta}_{tn})^{-1}\right)^{-1} \textit{: (equivalently)} \\
&\hat{\boldsymbol{V}}_{t+1} = \left((\boldsymbol{\Psi}^{\phi}_{t+1n})^{-1} + (\boldsymbol{\Psi}^{\beta}_{t+1n})^{-1}\right)^{-1} \\
&\Rightarrow \qquad \boldsymbol{\Omega}_t = \boldsymbol{\Psi}^{\beta}_{tn}
\end{aligned}$$

Let us impose:

$$\begin{aligned}
(\boldsymbol{V}_t + \boldsymbol{\Omega}_t)^{-1} &= -\boldsymbol{P}_t^{-1}\left(\hat{\boldsymbol{V}}_{t+1} - \boldsymbol{P}_t\right)\boldsymbol{P}_t^{-1} \qquad \Leftrightarrow \\
\Rightarrow \boldsymbol{V}_t + \boldsymbol{\Omega}_t &= -\boldsymbol{P}_t\left(\hat{\boldsymbol{V}}_{t+1} - \boldsymbol{P}_t\right)^{-1}\boldsymbol{P}_t^{-1} \\
&= \boldsymbol{P}_t\left(\boldsymbol{P}_t - \hat{\boldsymbol{V}}_{t+1}\right)^{-1}\boldsymbol{P}_t \qquad \textit{(Woodboury Identity)} \\
&= \boldsymbol{P}_t\left(\boldsymbol{P}_t^{-1} - \boldsymbol{P}_t^{-1}\left(\boldsymbol{P}_t^{-1} - \hat{\boldsymbol{V}}_{t+1}^{-1}\right)^{-1}\boldsymbol{P}_t^{-1}\right)\boldsymbol{P}_t \\
&= \boldsymbol{P}_t - \left(\boldsymbol{P}_t^{-1} - \hat{\boldsymbol{V}}_{t+1}^{-1}\right)
\end{aligned}$$

Notice that:

$$\boldsymbol{P}_t = \boldsymbol{V}_t + \boldsymbol{\Gamma} = \boldsymbol{\Psi}^{\phi}_{tn} + \boldsymbol{\Lambda}_n$$

and

$$\begin{aligned}\hat{\boldsymbol{V}}_{t+1}^{-1} &= \left(\boldsymbol{\Psi}_{t+1n}^{\phi}\right)^{-1} + \left(\boldsymbol{\Psi}_{t+1n}^{\beta}\right)^{-1} \qquad \textit{(Forward relation)} \\ &= \left(\boldsymbol{\Psi}_{t+1n}^{\iota}\right)^{-1} + \left(\boldsymbol{\Psi}_{tn}^{\phi} + \boldsymbol{\Lambda}_n\right)^{-1} + \left(\boldsymbol{\Psi}_{t+1n}^{\beta}\right)^{-1}\end{aligned}$$

$$\begin{aligned}\Rightarrow \qquad \boldsymbol{P}_t^{-1} - \hat{\boldsymbol{V}}_{t+1}^{-1} &= \left(\boldsymbol{\Psi}_{tn}^{\phi} + \boldsymbol{\Lambda}_n\right)^{-1} - \left[\left(\boldsymbol{\Psi}_{t+1n}^{\iota}\right)^{-1} + \left(\boldsymbol{\Psi}_{tn}^{\phi} + \boldsymbol{\Lambda}_n\right)^{-1} \right. \\ &\quad \left. + \left(\boldsymbol{\Psi}_{t+1n}^{\beta}\right)^{-1}\right] \\ &= -\left[\left(\boldsymbol{\Psi}_{t+1n}^{\iota}\right)^{-1} + \left(\boldsymbol{\Psi}_{t+1n}^{\beta}\right)^{-1}\right]\end{aligned}$$

$$\begin{aligned}\Leftrightarrow \qquad \boldsymbol{V}_t + \boldsymbol{\Omega}_t &= \boldsymbol{P}_t - \left(\boldsymbol{P}_t^{-1} - \hat{\boldsymbol{V}}_{t+1}^{-1}\right) \\ &= \boldsymbol{\Psi}_{tn}^{\phi} + \boldsymbol{\Lambda}_n - \left(-\left[\left(\boldsymbol{\Psi}_{t+1n}^{\iota}\right)^{-1} + \left(\boldsymbol{\Psi}_{t+1n}^{\beta}\right)^{-1}\right]\right)^{-1}\end{aligned}$$

$$\Leftrightarrow \qquad \boldsymbol{\Omega}_t = \boldsymbol{\Lambda}_n + \left[\left(\boldsymbol{\Psi}_{t+1n}^{\iota}\right)^{-1} + \left(\boldsymbol{\Psi}_{t+1n}^{\beta}\right)^{-1}\right]^{-1}$$

which is our exact backward recursion.

Relationship between learning LDS and M-step

In our model, we defined and derived formulas for model parameters in the M-step of EM algorithm. The same happens in linear dynamical systems, especially in the learning part, as Bishop shows in his book.

Again, we report here formulas from the book

$$\begin{aligned}\mathrm{E}\left\{\boldsymbol{z}_n\right\} &= \hat{\boldsymbol{\mu}}_n \\ \mathrm{E}\left\{\boldsymbol{z}_n \boldsymbol{z}_{n-1}^{\top}\right\} &= \boldsymbol{J}_{n-1}\hat{\boldsymbol{V}}_n + \hat{\boldsymbol{\mu}}_n \hat{\boldsymbol{\mu}}_{n-1}^{\top} \\ \mathrm{E}\left\{\boldsymbol{z}_{n-1} \boldsymbol{z}_n^{\top}\right\} = \mathrm{E}\left\{\boldsymbol{z}_n \boldsymbol{z}_{n-1}^{\top}\right\}^{\top} &= \hat{\boldsymbol{V}}_n \boldsymbol{J}_{n-1} + \hat{\boldsymbol{\mu}}_{n-1} \hat{\boldsymbol{\mu}}_n^{\top} \\ \mathrm{E}\left\{\boldsymbol{z}_n \boldsymbol{z}_n^{\top}\right\} &= \hat{\boldsymbol{V}}_n + \hat{\boldsymbol{\mu}}_n \hat{\boldsymbol{\mu}}_n^{\top}\end{aligned}$$

and

$$\begin{aligned}
\boldsymbol{\mu}_0^{new} &= \mathrm{E}\left\{\boldsymbol{z}_1\right\} \\
\boldsymbol{V}_0^{new} &= \mathrm{E}\left\{\boldsymbol{z}_1\boldsymbol{z}_1^\top\right\} - \mathrm{E}\left\{\boldsymbol{z}_1\right\}\mathrm{E}\left\{\boldsymbol{z}_1^\top\right\} \\
\boldsymbol{\Sigma}^{new} &= \frac{1}{N}\sum_{n=1}^{N}\left[\boldsymbol{x}_n\boldsymbol{x}_n^\top - \mathrm{E}\left\{\boldsymbol{z}_n\right\}\boldsymbol{x}_n^\top - \boldsymbol{x}_n\mathrm{E}\left\{\boldsymbol{z}_n^\top\right\} + \mathrm{E}\left\{\boldsymbol{z}_n\boldsymbol{z}_n^\top\right\}\right] \\
\boldsymbol{\Gamma}^{new} &= \frac{1}{N-1}\sum_{n+2}^{N}\Big[\mathrm{E}\left\{\boldsymbol{z}_n\boldsymbol{z}_n^\top\right\} - \mathrm{E}\left\{\boldsymbol{z}_{n-1}\boldsymbol{z}_n^\top\right\} - \mathrm{E}\left\{\boldsymbol{z}_n\boldsymbol{z}_{n-1}^\top\right\} \\
&\quad + \mathrm{E}\left\{\boldsymbol{z}_{n-1}\boldsymbol{z}_{n-1}^\top\right\}\Big]
\end{aligned}$$

where we skipped the parameters $\boldsymbol{A}^{new}$ and $\boldsymbol{C}^{new}$ because they are not necessary for our model.

From $(\boldsymbol{\mu}_0^{new}, \boldsymbol{V}_0^{new})$ to $(\boldsymbol{\nu}_n, \boldsymbol{\Omega}_n)$ These parameters refer to initial time of the temporal order model, which is identified by the set of latent variables $\mathbf{S}$ in our model. So, it is quite straightforward to relate the notations.

$$\boxed{\boldsymbol{\mu}_0^{new} = \mathrm{E}\left\{\boldsymbol{z}_1\right\} = \hat{\boldsymbol{\mu}}_1 = \boldsymbol{\mu}_{1n}^{(r+1)} = \boldsymbol{\nu}_n}$$

$$\begin{aligned}
\boldsymbol{V}_0^{new} &= \mathrm{E}\left\{\boldsymbol{z}_1\boldsymbol{z}_1^\top\right\} - \mathrm{E}\left\{\boldsymbol{z}_1\right\}\mathrm{E}\left\{\boldsymbol{z}_1^\top\right\} \\
&= \hat{\boldsymbol{V}}_1 + \hat{\boldsymbol{\mu}}_1\hat{\boldsymbol{\mu}}_1^\top - \hat{\boldsymbol{\mu}}_1\hat{\boldsymbol{\mu}}_1^\top \\
&= \boxed{\boldsymbol{\Psi}_{1n} + \boldsymbol{\nu}_n\boldsymbol{\nu}_n^\top - \boldsymbol{\nu}_n\boldsymbol{\nu}_n^\top = \boldsymbol{\Psi}_{1n} = \boldsymbol{\Omega}_n}
\end{aligned}$$

From $\boldsymbol{\Sigma}^{new}$ to $\boldsymbol{\Sigma}_n$ As our observations depend on more than one latent variable, precisely on two variables, the parameter $\boldsymbol{\Sigma}_n$ is a bit more complex with a variable depending on the second hidden variable. However, all other part of the formula correspond with the one provided from Bishop's book.

We assumed

$$\boldsymbol{x}_n = \boldsymbol{f}_{tk}, \qquad \hat{\boldsymbol{\mu}}_n = \boldsymbol{\mu}_{tn}, \qquad \hat{\boldsymbol{V}}_n = \boldsymbol{\Psi}_{tn}$$

So,

$$
\begin{aligned}
\boldsymbol{\Sigma}^{new} &= \frac{1}{N}\sum_{n=1}^{N}\left[\boldsymbol{x}_n\boldsymbol{x}_n^\top - \mathrm{E}\left\{\boldsymbol{z}_n\right\}\boldsymbol{x}_n^\top - \boldsymbol{x}_n\mathrm{E}\left\{\boldsymbol{z}_n^\top\right\} + \mathrm{E}\left\{\boldsymbol{z}_n\boldsymbol{z}_n^\top\right\}\right] \\
&= \frac{1}{N}\sum_{n=1}^{N}\left[\boldsymbol{x}_n\boldsymbol{x}_n^\top - \hat{\boldsymbol{\mu}}_n\boldsymbol{x}_n^\top - \boldsymbol{x}_n\hat{\boldsymbol{\mu}}_n^\top + \hat{\boldsymbol{V}}_n + \hat{\boldsymbol{\mu}}_n\hat{\boldsymbol{\mu}}_n^\top\right] \\
&= \frac{1}{N}\sum_{n=1}^{N}\left[\hat{\boldsymbol{V}}_n + \boldsymbol{x}_n\boldsymbol{x}_n^\top - \hat{\boldsymbol{\mu}}_n\boldsymbol{x}_n^\top - \boldsymbol{x}_n\hat{\boldsymbol{\mu}}_n^\top + \hat{\boldsymbol{\mu}}_n\hat{\boldsymbol{\mu}}_n^\top\right] \\
&\Rightarrow \frac{\sum_{t,k=1}^{T,K_t}\eta_{ktn}\left(\boldsymbol{\Psi}_{tn} + \boldsymbol{f}_{tk}\boldsymbol{f}_{tk}^\top - \boldsymbol{\mu}_{tn}\boldsymbol{f}_{tk}^\top - \boldsymbol{f}_{tk}\boldsymbol{\mu}_{tn}^\top + \boldsymbol{\mu}_{tn}\boldsymbol{\mu}_{tn}^\top\right)}{\sum_{t,k=1}^{T,K_t}\eta_{ktn}} \\
&= \boxed{\frac{\sum_{t,k=1}^{T,K_t}\eta_{ktn}\left[\boldsymbol{\Psi}_{tn} + \left(\boldsymbol{\mu}_{tn} - \boldsymbol{f}_{tk}\right)\left(\boldsymbol{\mu}_{tn} - \boldsymbol{f}_{tk}\right)^\top\right]}{\sum_{t,k=1}^{T,K_t}\eta_{ktn}} = \boldsymbol{\Sigma}_n}
\end{aligned}
$$

Bibliography

[Alameda-Pineda et al., 2011] Alameda-Pineda, X., Khalidov, V., Horaud, R., and Forbes, F. (2011). Finding audio-visual events in informal social gatherings. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 247–254.

[Arnaud et al., 2008] Arnaud, E., Christensen, H., Lu, Y.-C., Barker, J., Khalidov, V., Hansard, M., Holveck, B., Mathieu, H., Narasimha, R., Taillant, E., Forbes, F., and Horaud, R. (2008). The CAVA corpus: synchronised stereoscopic and binaural datasets with head movements. In *ICMI 2008 - ACM/IEEE International Conference on Multimodal Interfaces*, pages 109–116.

[Arulampalam et al., 2002] Arulampalam, M. S., Maskell, S., and Gordon, N. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, 50:174–188.

[Bar-Shalom and Fortmann, 1988] Bar-Shalom, Y. and Fortmann, T. (1988). *Tracking and Data Association*. Academic Press.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

[Blackman, 2004] Blackman, S. (2004). Multiple hypothesis tracking for multiple target tracking. *Aerospace and Electronic Systems Magazine, IEEE*, 19:5–18.

[Clark and Godsill, 2007] Clark, D. and Godsill, S. (2007). Group target tracking with the gaussian mixture probability hypothesis density filter. In *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*, pages 149–154.

[Comaniciu and Meer, 1999] Comaniciu, D. and Meer, P. (1999). Mean shift analysis and applications. pages 1197–1203.

[Cox, 1993] Cox, I. J. (1993). A Review of Statistical Data Association Techniques for Motion Correspondence. *International Journal of Computer Vision*, 10:53–66.

[Feldmann et al., 2011] Feldmann, M., Franken, D., and Koch, W. (2011). Tracking of extended objects and group targets using random matrices. *Signal Processing, IEEE Transactions on*, 59:1409–1420.

[Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Alvey vision conference*, page 50.

[Ishiguro et al., 2008] Ishiguro, K., Yamada, T., and Ueda, N. (2008). Simultaneous clustering and tracking unknown number of objects. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

[Julier and Uhlmann, 1997] Julier, S. J. and Uhlmann, J. K. (1997). A new extension of the kalman filter to nonlinear systems. In *Int. symp. aerospace/defense sensing, simul. and controls*, pages 3–2.

[Kalman, 1960] Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82:35–45.

[Lowe, 2003] Lowe, D. G. (2003). Distinctive image features from scale-invariant keypoints.

[Lucas and Kanade, 1981] Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. pages 674–679.

[Papageorgiou et al., 1998] Papageorgiou, C. P., Oren, M., and Poggio, T. (1998). A general framework for object detection. In *Sixth International Conference on*, pages 555–562.

[Pece, 2002] Pece, A. E. (2002). Generative-model-based tracking by cluster analysis of image differences. *Robotics and Autonomous Systems*, 39:181 – 194.

[Petersen and Pedersen, 2012] Petersen, K. B. and Pedersen, M. S. (2012). The Matrix Cookbook.

[Reid, 1978] Reid, D. (1978). An algorithm for tracking multiple targets. In *Decision and Control including the 17th Symposium on Adaptive Processes, 1978 IEEE Conference on*, pages 1202–1211.

[Shi and Tomasi, 1994] Shi, J. and Tomasi, C. (1994). Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 593–600.

[Smith et al., 2005] Smith, K., Gatica-Perez, D., Odobez, J., and Ba, S. (2005). Evaluating multi-object tracking. In *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, pages 36–36.

[Stauffer and Grimson, 2000] Stauffer, C. and Grimson, W. E. L. (2000). Learning patterns of activity using real-time tracking. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 22:747–757.

[Streit and Luginbuhl, 1995] Streit, R. L. and Luginbuhl, T. E. (1995). Probabilistic Multi-Hypothesis Tracking. Technical report.

[Viola et al., 2003] Viola, P., Jones, M., and Snow, D. (2003). Detecting pedestrians using patterns of motion and appearance. In *In ICCV*, pages 734–741.

[Wan and Van Der Merwe, 2000] Wan, E. A. and Van Der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Sig-*

nal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000, pages 153–158.

[Welch and Bishop, 1995] Welch, G. and Bishop, G. (1995). An introduction to the kalman filter. Technical report.

[Willett et al., 2002] Willett, P., Ruan, Y., and Streit, R. (2002). PMHT: problems and some solutions. *Aerospace and Electronic Systems, IEEE Transactions on*, 38:738–754.

[Willett et al., 1995] Willett, P., Streit, R., and Rago, C. (1995). A comparison of the JPDAF and PMHT tracking algorithms. In *International Conference on Acoustics, Speech, and Signal Processing, ICASSP-95.*, pages 3571–3574.

[Yilmaz et al., 2006] Yilmaz, A., Javed, O., and Shah, M. (2006). Object tracking: A survey. *ACM computing surveys (CSUR)*, 38:13.