Bayesian On-line Learning of Driving Behaviors

Jérôme Maye*, Rudolph Triebel*, Luciano Spinello†, and Roland Siegwart*
* Autonomous Systems Lab, ETH Zurich, Switzerland
email: {jerome.maye, rudolph.triebel, roland.siegwart}@mavt.ethz.ch
† Social Robotics Lab, University of Freiburg, Germany
email: spinello@informatik.uni-freiburg.de

***Abstract*— This paper presents a novel self-supervised on-line learning method to discover driving behaviors from data acquired with an inertial measurement unit (IMU) and a camera. Both sensors where mounted in a car that was driven by a human through a typical city environment with intersections, pedestrian crossings and traffic lights. The presented system extracts motion segments from the IMU data and relates them to visual cues obtained from camera data. It employs a Bayesian on-line estimation method to discover the motion segments based on change-point detection and uses a Dirichlet Compound Multinomial (DCM) model to represent the visual features extracted from the camera images. By incorporating these visual cues into the on-line estimation process, labels are computed that are equal for similar motion segments. As a result, typical traffic situations such as braking maneuvers in front of a red light can be identified automatically. Furthermore, appropriate actions in form of observed motion changes are associated to the discovered traffic situations. The approach is evaluated on a real data set acquired in the center of Zurich.**

I. Introduction

The development of intelligent driver assistant systems has become a very active research field in the last years. The large spectrum of potential applications for such systems ranges from automatic warning systems that detect obstacles and dynamic objects over automated parking systems to fully autonomous cars that are able to navigate in busy city environments. One aspect that is of major importance in all these systems is the perception part of the vehicle, i.e., the data acquisition and semantic interpretation of the environment. The major challenges here include the required accuracy of the detection system, the time constraints given by the speed of the vehicle and its implied temporal restrictions on the decision process, as well as the large variability in which potential objects and the environment itself may appear. Especially this latter point poses a significant challenge on the perception task, because standard learning techniques that most often rely on supervised off-line classification algorithms tend to give poor results when the test environment largely differs from the acquired training data. Furthermore, such systems are not capable of adapting to new, unseen situations, which reduces their applicability for long-term use cases.

In this paper, we present a self-supervised on-line learning algorithm that recognizes driving behaviors and predicts appropriate actions accordingly. A driving behavior in our context is defined as a short sequence of actuation commands to the vehicle that typically occur in certain traffic situations.

Fig. 1. Example of a traffic light scenario (label 10) detected by our algorithm. The suggested action is a braking maneuver.

An example is the braking maneuver in front of a red traffic light. In our system, the driving behaviors are observed using an inertial measurement unit (IMU) and a camera while a human is driving the vehicle. Using our approach, the system is able to detect and classify new traffic scenarios and predict appropriate actions based on the driving behaviors learned in earlier stages of the data acquisition process. The principle idea is to first segment the data stream from the IMU into consistent sequences using *change-point detection*, and then relate these motion sequences to visual features observed in the camera data during the corresponding motion. To find the change-points in the motion data, we use an efficient Bayesian approach based on a Rao-Blackwellized particle filter. The visual features are represented in a bag-of-words approach using a Dirichlet Compound Multinomial (DCM) model. The detected motion segments are grouped on-line and without human intervention, according to their similarities in their corresponding visual features. This enables the system to predict new motion commands according to the traffic situation it detects from new camera data. Thus, it predicts a braking maneuver when it encounters enough evidence for a red light in the camera data. Fig. 1 shows a typical output of our algorithm.

The paper is structured as follows. Section II summarizes the previous works related to ours. Section III introduces our Bayesian framework. Section IV describes our motion segmentation method. Section V shows how we model a traffic situation. Section VI demonstrates our action model. Section VII presents experimental results. Section VIII outlines our conclusions and provides some insights for future work.

II. Related Work

Existing driving behavior models in psychology are largely subjective and based on self-report scales [1]. They are difficult to quantify, because they include many psychological aspects like motivation, or risk assessment. Many works in the intelligent vehicle literature [2], [3], [4], [5] focus on modeling the driver behavior via their steering behavior or road tracking information or desired driver's path as source of behavior's information. Other works recognize driver's intentions via Bayesian reasoning on a complex input including the driver's current control actions and the traffic environment surrounding them [6], [7]. In a previous work [8], we were able to infer an action from a direction sign in an indoor environment with a semi-supervised approach using vision and prerecorded robot actions. We extend this idea to outdoor, remove any supervision, and predict vehicle actions in an on-line fashion. Meyer *et al.* [9] predicted traffic situations using Hidden Markov Models (HMM). They however restricted their situations space by modeling states with respect to surrounding vehicles (distance, speed, bearing) and manually segmented image sequences for initial estimates. In this paper, we exclude any manual intervention in the process and use a more complete set of variables for predicting states. Other works [10], [11] make use of supervised off-line classification methods for learning the relation between driving actions and visual features. The actions are manually annotated and discretized in the training phase. To our knowledge, there has been few research works that combine traffic scenario recognition and action prediction in an on-line and unsupervised fashion.

III. Problem Formulation

Given a vehicle equipped with an Inertial Measurement Unit (IMU) and a monocular camera, we seek to learn the relation between motion and visual data in an on-line and unsupervised manner. We shall follow an entirely probabilistic approach and formulate the problem as the estimation of the joint filtering distribution

$$p(r_t, l_t, \mathbf{a}_t \mid \mathbf{z}_{1:t}, \mathbf{c}_{1:t}), \tag{1}$$

where r_t represents the motion segment length at time t, l_t the image label at time t, $\mathbf{a}_t$ the predicted action at time t, $\mathbf{z}_{1:t}$ the IMU measurements up to time t, and $\mathbf{c}_{1:t}$ the camera measurements up to time t.

Assuming r_t is conditionally independent of $\mathbf{c}_{1:t}$ given $\mathbf{z}_{1:t}$, l_t of $\mathbf{z}_{1:t}$ given $\mathbf{c}_{1:t}$, and $\mathbf{a}_t$ of $\mathbf{c}_{1:t}$ given $\mathbf{z}_{1:t}$, we can decompose (1) into

$$p(r_t, l_t, \mathbf{a}_t \mid \mathbf{z}_{1:t}, \mathbf{c}_{1:t}) = p(r_t \mid \mathbf{z}_{1:t}) p(l_t \mid r_t, \mathbf{c}_{1:t}) p(\mathbf{a}_t \mid r_t, l_t, \mathbf{z}_{1:t}). \tag{2}$$

$p(r_t \mid \mathbf{z}_{1:t})$ corresponds to the motion segmentation of Section IV, $p(l_t \mid r_t, \mathbf{c}_{1:t})$ to the traffic situation modeling of Section V, and $p(\mathbf{a}_t \mid r_t, l_t, \mathbf{z}_{1:t})$ to the action prediction model of Section VI.

IV. Bayesian On-line Segmentation of Motion Data

Our motion segmentation algorithm is based on *change-point detection*. A change-point is an abrupt variation in the generative parameters of sequential data. An efficient Bayesian on-line method for detecting change-points has been independently proposed by Adams and MacKay [12] and by Fearnhead and Liu [13]. In the following, we first present this method in general and then show how we apply it to the problem of segmenting motion data.

A. Change-Point Detection

Suppose we are given a time-dependent sequence of observations $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T$, where the $\mathbf{z}_t$ can be scalars or vectors. Our goal is to find segments $s_1, s_2, \ldots, s_N$ with $s_n = [\mathbf{z}_{b_n}, \ldots, \mathbf{z}_{e_n}]$, where $e_n > b_n$ and $b_n = e_{n-1} + 1$ for $n = 1, \ldots, N$. We assume that all data points $\mathbf{z}_{b_n}, \ldots, \mathbf{z}_{e_n}$ of a segment s_n are independently and identically distributed (i.i.d.) according to a parameterized statistical model $p(\mathbf{z} \mid \boldsymbol{\eta}_n)$. The parameter vectors $\boldsymbol{\eta}_1, \ldots \boldsymbol{\eta}_N$ are also assumed to be i.i.d. The computation of the segments is done on-line, i.e., at each time step t a decision is made whether $\mathbf{z}_t$ is added to the current segment $s_n = [\mathbf{z}_{b_n}, \ldots, \mathbf{z}_{t-1}]$ or a new segment is started. As shown above, we denote the length of the current segment as r_t. Thus, after deciding on $\mathbf{z}_t$, we have either $r_t = r_{t-1} + 1$ or $r_t = 0$ in case we start a new segment.

To determine whether time step t is a change-point, we analyze the posterior distribution of the segment length conditioned on the data observed so far, i.e. $p(r_t \mid \mathbf{z}_{1:t})$. Using the product rule, this filtering distribution can be written as

$$p(r_t \mid \mathbf{z}_{1:t}) \propto p(r_t, \mathbf{z}_{1:t}). \tag{3}$$

The joint distribution in (3) can be further expressed as

$$\begin{aligned} p(r_t, \mathbf{z}_{1:t}) &= \sum_{r_{t-1}} p(r_t, r_{t-1}, \mathbf{z}_{1:t}) \\ &= \sum_{r_{t-1}} p(r_t, \mathbf{z}_t \mid r_{t-1}, \mathbf{z}_{1:t-1}) p(r_{t-1}, \mathbf{z}_{1:t-1}) \\ &= \sum_{r_{t-1}} p(r_t \mid r_{t-1}) p(\mathbf{z}_t \mid r_{t-1}, \mathbf{z}_{1:t-1}) p(r_{t-1}, \mathbf{z}_{1:t-1}). \end{aligned} \tag{4}$$

The right-hand side of (4) consists of three terms: the *transition probability* $p(r_t \mid r_{t-1})$ of the Markov chain formed by $r_1, r_2, \ldots, r_t$, the *predictive distribution* $p(\mathbf{z}_t \mid r_{t-1}, \mathbf{z}_{1:t-1})$, and the posterior $p(r_{t-1}, \mathbf{z}_{1:t-1})$ from the previous time step. We have exploited Markov assumption for the simplifications in (4).

As there are only two possible successor states for r_t, namely $r_{t-1} + 1$ or 0, we can model the transition probability using statistical survival analysis, i.e., the segment length can either "survive" or "die". To do this, we define a *survival function* $S(t)$ as the probability that the current segment is still alive after time step t. The complement of S is usually named the *lifetime distribution function* $F(t) = 1 - S(t)$ and its temporal derivative $f(t)$ is denoted the *event rate*. Finally, the *hazard function* $h(t)$ is defined as the event rate conditioned on the survival of the segment at time t, i.e.

$$h(t) = \frac{f(t)}{S(t)} = \frac{f(t)}{1 - F(t)}. \quad (5)$$

Intuitively, $h(t)$ represents the probability that the segment dies exactly at the current time instant t. We can use $h(t)$ to model the transition probability as

$$p(r_t \mid r_{t-1}) = \begin{cases} h(r_{t-1}+1) & \text{if } r_t = 0 \\ 1 - h(r_{t-1}+1) & \text{if } r_t = r_{t-1}+1 \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

A common approach is to model $S(t)$ as an exponential function $S(t) = \exp(-\lambda t)$ with some given rate parameter λ. Then, the hazard function turns into

$$h(t) = \frac{\lambda \exp(-\lambda t)}{\exp(-\lambda t)} = \lambda. \quad (7)$$

Thus, the hazard rate is constant and the process is "memoryless".

For the computation of the predictive distribution, we can make it dependent only on the last data point $\mathbf{z}_{t-1}$ since we are doing a sequential update of the parameters. Thus, it can be expressed as $p(\mathbf{z}_t \mid r_{t-1}, \mathbf{z}_{t-1})$. We finally introduce the model parameters $\boldsymbol{\eta}^{r_{t-1}}$ that are learned on the current segment and compute the predictive distribution by marginalizing them out, i.e.

$$p(\mathbf{z}_t \mid r_{t-1}, \mathbf{z}_{t-1}, \boldsymbol{\psi}^{r_{t-1}}) = \int_{\boldsymbol{\eta}^{r_{t-1}}} p(\mathbf{z}_t \mid \boldsymbol{\eta}^{r_{t-1}}) p(\boldsymbol{\eta}^{r_{t-1}} \mid r_{t-1}, \mathbf{z}_{t-1}, \boldsymbol{\psi}^{r_{t-1}}) d\boldsymbol{\eta}^{r_{t-1}}. \quad (8)$$

Here, we have added the prior hyperparameters $\boldsymbol{\psi}^{r_{t-1}}$ for completeness. The integral in (8) can be solved analytically if we model the prior of the parameter vector $\boldsymbol{\eta}^{r_{t-1}}$ as a conjugate to the probability density function $p(\mathbf{z}_t \mid \boldsymbol{\eta}^{r_{t-1}})$. Otherwise, this leads to expensive numerical computations. When the terms inside the integral are conjugate models, the marginal distribution is usually a function of the hyperparameters $\boldsymbol{\psi}^{r_{t-1}}$ which can be updated iteratively as data arrives.

B. Complexity and Approximate Inference

In order to exactly infer the positions of all change-points until time t, we need to compute and store $p(r_t \mid \mathbf{z}_{1:t})$ for t and all previous time steps. We can then get the Maximum A Posteriori (MAP) estimate of the sequence of segment lengths using the on-line Viterbi algorithm of [13].

Regarding complexity, if we have processed n data points, the storage of the full posterior distribution has a memory cost of $O(n^2)$ and $O(n)$ computational cost. This might be prohibitive for huge datasets. For this reason, the distribution has to be approximated. A simple way sketched in [12] is to discard values where the distribution is significantly low, i.e., lower than a given threshold. However, as we want to accurately estimate our distribution and control the computational costs, we use a *particle filter*. The state-space of r_t being discrete and the number of successor states being small, we can evaluate all the possible descendants of r_t. Indeed, if r_t takes k possible values, r_{t+1} will take $k+1$ possible values. At each time step t, we approximate the posterior distribution with a set $\{r_t^{(i)}, \boldsymbol{\psi}^{r_{t-1},(i)}\}_{i=1}^{M}$ of M particles weighted by $\{w_t^{(i)}\}_{i=1}^{M}$ with

$$w_t^{(i)} \propto p(\mathbf{z}_t \mid r_{t-1}^{(i)}, \mathbf{z}_{t-1}, \boldsymbol{\psi}^{r_{t-1},(i)}). \quad (9)$$

In order to limit the number of particles at each time step, we use the Stratified Optimal Re-sampling (SOR) presented in [13], whenever M gets bigger than our particles number limit P.

Using this method reduces the memory costs to $O(n)$ and the computational costs to $O(1)$, i.e. constant run-time. We also notice that this particle filter is Rao-Blackwellized [14] and has thus a lower variance since the sampling space of the state is reduced to r_t and the rest is marginalized out.

C. Application to Motion Data Segmentation

In our particular case, data comes from an IMU and we consider accelerations in the x, y axes and the *yaw* rate, with x pointing forward, y on the left, and z upward. We can safely assume that an IMU measurement $\mathbf{z}_t$ arises from a multivariate normal distribution with mean $\boldsymbol{\mu}_n$ and covariance matrix $\boldsymbol{\Sigma}_n$ for segment s_n. The parameter vector for segment r_t is thus $\boldsymbol{\eta}^{r_t} = \{\boldsymbol{\mu}^{r_t}, \boldsymbol{\Sigma}^{r_t}\}$. In order to solve the integral in (8) analytically, we model the parameter prior as a normal-Wishart distribution which is conjugate to the multivariate Gaussian. This distribution has four hyperparameters $\boldsymbol{\psi}^{r_t} = \{\kappa^{r_t}, \boldsymbol{\rho}^{r_t}, \nu^{r_t}, \boldsymbol{\Lambda}^{r_t}\}$ that can be updated iteratively as a new data point $\mathbf{z}_t$ arrives with:

$$\begin{aligned} \kappa^{r_t} &= \kappa^{r_{t-1}} + 1 \\ \boldsymbol{\rho}^{r_t} &= \frac{\kappa^{r_{t-1}} \boldsymbol{\rho}^{r_{t-1}} + \mathbf{z}_t}{\kappa^{r_{t-1}} + 1} \\ \nu^{r_t} &= \nu^{r_{t-1}} + 1 \\ \boldsymbol{\Lambda}^{r_t} &= \boldsymbol{\Lambda}^{r_{t-1}} + \frac{\kappa^{r_{t-1}}}{\kappa^{r_{t-1}} + 1} (\mathbf{z}_t - \boldsymbol{\rho}^{r_{t-1}})(\mathbf{z}_t - \boldsymbol{\rho}^{r_{t-1}})^\intercal. \end{aligned} \quad (10)$$

In case we start a new segment and $r_t = 0$, the hyperparameters are fixed to some prior values $\boldsymbol{\psi}_0 = \{\kappa_0, \boldsymbol{\rho}_0, \nu_0, \boldsymbol{\Lambda}_0\}$.

From (10), we can express the parameters of the resulting multivariate normal distribution in (8) as

$$\begin{aligned} \boldsymbol{\mu}^{r_t} &= \boldsymbol{\rho}^{r_t} \\ \boldsymbol{\Sigma}^{r_t} &= (\boldsymbol{\Lambda}^{r_t})^{-1} / \kappa^{r_t}. \end{aligned} \quad (11)$$

Finally, for the computation of the predictive distribution in (8), we approximate the multivariate normal distribution with a Student's t-distribution which is known to be more robust to outliers in case of few data points. This distribution converges to the Gaussian when its degrees of freedom go to infinity. We use the number of processed points as the degrees of freedom for the distribution, so as to have a bigger variance at the beginning.

V. Labeling of Traffic Situations

Our aim is to find a label for each segmented motion pattern. This label represents a traffic situation, e.g., a stop or turn condition. Moreover, we are interested in associating two different motion segments to the same label whenever they depict the same traffic situation. In the following, we show how we can integrate this labeling into the on-line framework of Section IV.

A. Traffic Situation Model

As shown above, we denote the label of a segment r_t as l_t. This label can take values in $\{1, 2, \ldots, N\}$ corresponding to N parametric models $M_1, M_2, \ldots, M_N$. Each of the M_i is a generative model $p(\mathbf{c}_t \mid \boldsymbol{\eta}_i)$ for a particular traffic situation with parameter vector $\boldsymbol{\eta}_i$. At time t, we estimate the distribution over the known models conditioned on the data seen so far and the segment we are in with Bayes law as

$$\begin{aligned} p(l_t \mid r_t, \mathbf{c}_{1:t}) &\propto p(\mathbf{c}_t \mid l_t, r_t, \mathbf{c}_{1:t-1}) p(l_t \mid r_t, \mathbf{c}_{1:t-1}) \\ &= p(\mathbf{c}_t \mid l_t, r_t, \mathbf{c}_{t-1}) p(l_t \mid r_t, \mathbf{c}_{1:t-1}). \end{aligned} \quad (12)$$

For the prior part in (12), we use the posterior of the previous time step, that is $p(l_t \mid r_t, \mathbf{c}_{1:t-1}) = p(l_{t-1} \mid r_{t-1}, \mathbf{c}_{1:t-1})$. If we are in a new segment with $r_t = 0$, we set the prior to a uniform distribution over the known models, i.e., $p(l_t = 1 : N \mid r_t, \mathbf{c}_{1:t-1}) = \frac{1}{N}$. The likelihood part in (12) is computed with the model probability density function $p(\mathbf{c}_t \mid \boldsymbol{\eta}_i)$ in the same fashion as in (8), i.e., using a conjugate prior with hyperparameters $\boldsymbol{\psi}_i$ as will be detailed below. Furthermore, we have only kept the dependency on the last data point $\mathbf{c}_{t-1}$ since we update the parameters iteratively.

As we want to be able to discover new traffic situations on-line, we have to state if the current data $\mathbf{c}_t$ is unlikely to come from any of the N known models so far. We use Bayesian hypothesis testing and compute the *Bayes factor* [15] for all the models M_i against an alternative model:

$$B = \frac{p(\mathbf{c}_t \mid l_t = i, \boldsymbol{\psi}_i)}{p(\mathbf{c}_t \mid l_t, r_t, \mathbf{c}_{t-1}, \boldsymbol{\psi}^{r_{t-1}})}, \quad (13)$$

where $\boldsymbol{\psi}^{r_{t-1}}$ are the hyperparameters learned over the current segment r_{t-1}.

The value B in (13) indicates our confidence in the model M_i and we compare it to a threshold ξ for the decision. In case all models are rejected, we create a new instance M_{N+1} with hyperparameter vector $\boldsymbol{\psi}^{r_{t-1}}$, set $p(l_t = N + 1 \mid r_t, \mathbf{c}_{1:t}) = p_{new}$, and $p(l_t = 1 : N \mid r_t, \mathbf{c}_{1:t}) = (1 - p_{new})/N$. We finally update the hyperparameters $\boldsymbol{\psi}_i$ of model M_i, such that $i = \arg\max_{j=1:N} p(l_t = j \mid r_t, \mathbf{c}_{1:t})$, with $\mathbf{c}_t$.

From an implementation point of view, we attach the distribution (12), the hyperparameters $\boldsymbol{\psi}^{r_{t-1}}$, and the incremental set of known models M_i to each particle. Thus, our system is able to learn new traffic situations on-line and refine its knowledge over previously visited scenes.

B. Measurements Representation

We represent images using the widely adopted *bag-of-words* model [16]. In the document modeling formulation, text documents are represented as histograms of word counts from a given dictionary. This model can be easily applied to computer vision tasks, words being replaced by features and text documents by images.

We use Scale-Invariant Feature Transform (SIFT) [17] descriptors computed at Difference of Gaussians (DoG) keypoints. SIFT descriptors have been shown to be highly discriminative for object recognition. Although some authors claim that they obtain significantly better results with dense grid representations [18], DoG interest points are more suitable for our purpose. Indeed, we are not interested in capturing uniform regions such as sky, but rather focused on objects. N images are randomly selected from the entire dataset to build a *codebook* or dictionary of features using K-means clustering. Each feature of an image is then assigned to the nearest *codeword* of the dictionary and we can therefore build a convenient histogram representation.

The link between bag-of-*features* models in computer vision and bag-of-words models in text document modeling is intuitive. We can therefore use the generative model of [19] to represent an image in a probabilistic manner as was already proposed in [20]. Image histograms are modeled with a Dirichlet Compound Multinomial (DCM), also known as multivariate Polya distribution. The DCM combines a multinomial model and a Dirichlet prior, and provides an analytical solution to the marginalization of the multinomial parameters [21]. The multinomial distribution $p(\mathbf{c}_t \mid \boldsymbol{\theta})$ has parameters $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_K]$, corresponding to $\boldsymbol{\eta}$ above. The Dirichlet prior $p(\boldsymbol{\theta} \mid \boldsymbol{\alpha})$ has hyperparameters $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_K]$, corresponding to $\boldsymbol{\psi}$ above. The likelihood part of (12) can now be formulated as

$$\begin{aligned} p(\mathbf{c}_t \mid l_t, r_t, \mathbf{c}_{t-1}, \boldsymbol{\alpha}^{r_{t-1}}) &= \\ \int_{\boldsymbol{\theta}^{r_{t-1}}} p(\mathbf{c}_t \mid \boldsymbol{\theta}^{r_{t-1}}) p(\boldsymbol{\theta}^{r_{t-1}} \mid l_t, r_t, \mathbf{c}_{t-1}, \boldsymbol{\alpha}^{r_{t-1}}) d\boldsymbol{\theta}^{r_{t-1}} &= \\ \frac{n!}{\prod_{k=1}^{K} n_k!} \frac{\Gamma(\alpha^{r_{t-1}})}{\Gamma(n + \alpha^{r_{t-1}})} \prod_{k=1}^{K} \frac{\Gamma(n_k + \alpha_k^{r_{t-1}})}{\Gamma(\alpha_k^{r_{t-1}})}, & \end{aligned} \quad (14)$$

where $\Gamma(.)$ is the Gamma function, $n_k = \mathbf{c}_t(k)$, $n = \sum_{k=1}^{K} n_k$, $\alpha^{r_{t-1}} = \sum_{k=1}^{K} \alpha_k^{r_{t-1}}$, and we have added the hyperparameters $\boldsymbol{\alpha}^{r_{t-1}}$ in the conditional.

For the iterative update of the hyperparameters $\boldsymbol{\alpha}^{r_t}$, we can use the simple rule

$$\boldsymbol{\alpha}^{r_t} = \boldsymbol{\alpha}^{r_{t-1}} + \mathbf{c}_t. \quad (15)$$

In case we start a new segment and $r_t = 0$, the hyperparameters are fixed to some prior values $\boldsymbol{\psi}_0 = \{\boldsymbol{\alpha}_0\}$.

VI. Action Model

We want to estimate the posterior probability distribution over actions conditioned on the current traffic situation and segment. To this end, we closely follow the strategy of

Section V. To each of the traffic situation model M_i is associated an action model A_i, which we fit with a Gaussian Mixture Model (GMM). For the same traffic situation M_i, we are able to model several possible behaviors corresponding to the different Gaussian components. For instance, when we reach a traffic light, we might brake when the light is red and continue when it is green. Moreover, a driver does not always brake or accelerate exactly the same manner every time. Finally, our system can adapt to new drivers. We can thus formulate the following model that we estimate and update at each time step:

$$p(\mathbf{a_t} \mid r_t, l_t, \mathbf{z}_{1:t}, \boldsymbol{\psi}^{\mathbf{x}_t}) = \sum_{\mathbf{x}_t} p(\mathbf{x}_t) p(\mathbf{a}_t \mid \mathbf{x}_t, r_t, l_t, \mathbf{z}_{1:t}, \boldsymbol{\psi}^{\mathbf{x}_t}), \quad (16)$$

where $\mathbf{x}_t$ is a K-dimensional vector with a single one at the position k encoding for the k-th Gaussian and zeros elsewhere, $p(\mathbf{x}_t)$ is the prior for selecting a particular Gaussian component $\mathbf{x}_t$, and $p(\mathbf{a}_t \mid \mathbf{x}_t, r_t, l_t, \mathbf{z}_{1:t}, \boldsymbol{\psi}^{\mathbf{x}_t})$ is a Gaussian distribution with hyperparameters $\boldsymbol{\psi}^{\mathbf{x}_t}$. In a similar fashion as in Section IV, we have marginalized out the parameters of the Gaussian and are thus able to iteratively update the hyperparameters. The prior distribution $p(\mathbf{x}_t)$ is defined as

$$p(\mathbf{x}_t(i) = 1) \propto n_t^i, \quad (17)$$

where n_t^i is the sum of the points assigned to Gaussian component i.

Upon reception of a new data point $\mathbf{z}_t$, we compute the Bayes factor for all the Gaussian components $\mathbf{x}_{t-1}$ of the model l_t and compare it to ε. If all the components are rejected, a new Gaussian is created with hyperparameters $\boldsymbol{\psi}_0$. We update the hyperparameters of the most likely Gaussian component with the rule from (10) and increment the corresponding n_t^i.

From an implementation point of view, the distribution (16) and the learned GMM A_i are attached to the particle filter of Section IV.

VII. Experiments

In order to evaluate the approach proposed in this paper, we have collected a dataset with a car in an urban setting. Our car is equipped with a Sony XCD-SX910 camera recording 1280x960 images at 3.75 frames per second and an XSens MTi IMU running at 100 Hz with x pointing forward, y to the right, and z upward. The sequence contains 8218 images and lasts around 40 minutes. We have encountered different scenes comprising of traffic lights, crosswalks, or changes of speed limit. We have driven in a loop so as to come several times in the same situation and thus have an estimation of the quality of our solution.

A. Simulation

Since it is easier to have a ground truth on simulated data and hence validate our approach, we first display an experiment of the whole algorithm on synthetic data. For visualization purposes, we have simulated IMU data with an univariate normal distribution and have introduced change-points every 50 data points. We have randomly generated 3 different $\boldsymbol{\alpha}_i$ with $K = 256$ coding for the traffic situations. Although the algorithm starts with no prior knowledge, we could also start with previously learned models M_i and A_i.

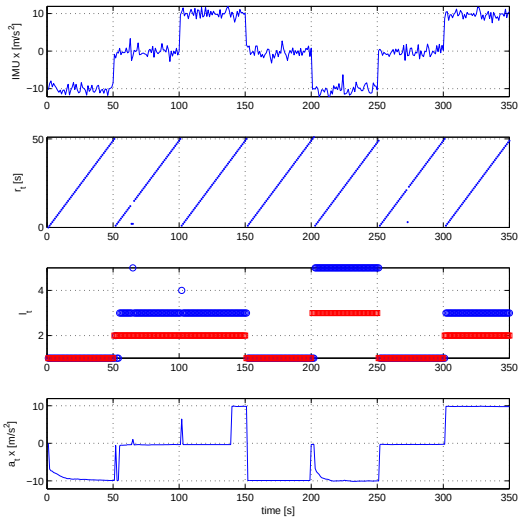

Fig. 2. Simulation results of the algorithm. From top to bottom, the plots display the simulated IMU data $\mathbf{z}_t$, the inferred segment lengths r_t, the inferred labels l_t (blue circle) with ground truth (red square), and the MAP estimate for the action $\mathbf{a}_t$.

Fig. 2 depicts the output of the simulation and demonstrates the pertinence of our method. We display the MAP solution for (16) on the bottom plot and thus the prediction reflects the Gaussian with the maximum number of data points. At time step 100, a new Gaussian with mean 10 is created for label 2. It becomes the MAP only at time step 300 after accumulating enough evidence. Even though the label numbers l_t differ from the ground truth, they are actually correctly estimated since the induced partition is equivalent.

B. Motion Segmentation

We have estimated the quality of our motion segmentation algorithm from Section IV on real-world data and performed inference on the final posterior distribution (4) to get the optimal sequence of segment lengths which represents our motion segments. We set the hazard rate to $\lambda = 1/10$, the number of particles to $P = 100$, and the prior hyperparameters of the normal-Wishart to $\kappa_0 = 1, \boldsymbol{\rho}_0 = \mathbf{0}, \nu_0 = 3, \boldsymbol{\Lambda}_0 = \mathbf{I}$. We only considered IMU data at 10 Hz.

Fig. 3 shows the extracted motion segments along with the corresponding IMU data. Our algorithm identified 165 segments which are validated by visual inspection of the IMU data. Furthermore, the segmentation has been compared to a manual annotation of our image sequence and exhibited an accuracy of approximatively 92%. For the labeling of the change-points, we have watched the video and noted down where we would expect a change of driving behavior. The parameter λ controls the false positives/negatives rates.

C. Traffic Situation Labeling and Recognition

We have evaluated the technique presented in Section V and performed inference on (12) to obtain the most likely

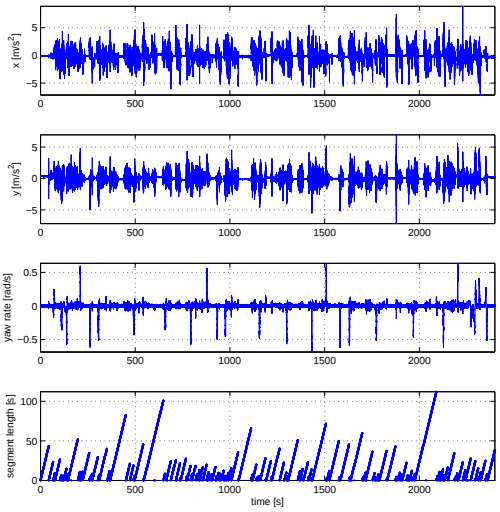

Fig. 3. Optimal motion segmentation from IMU data. The three top plots are the IMU raw values over time. The bottom plot depicts the motion segments discovered by our algorithm.

label for a scene. In a first phase, we have collected a subset of images from traffic lights, yield signs, and pedestrian crossings. Models M_i were learned on these images using (15) and frozen during the evaluation. In a second phase, we have started the algorithm with no prior models. The dictionary was created from a set of $N = 400$ randomly picked images and the SIFT features quantized into $K = 256$ visual words. The prior hyperparameters of the Dirichlet distribution were set to $\boldsymbol{\alpha}_0 = \mathbf{1}$.

In the supervised case, we have manually annotated the image sequence and compared the resulting labeling to the ground truth. We obtained an accuracy of 93% for traffic light scenes, 99% for yield scenes, and 91% for pedestrian crossings scenes. We except these results to drop slightly in a previously unseen environment. In the unsupervised case, ξ acts as a concentration parameter, i.e., it controls the tendency to create new classes. The final labeling is challenging to evaluate. Two traffic lights scenes might for instance get different labels without interfering into the final action prediction. With $\xi = 200$, our algorithm discovered 15 different traffic situations and was able to re-associate correctly to the same labels in the different runs of our driving loop.

D. Action Prediction

The strategy presented in Section VI is relatively straightforward to evaluate, since predictions can be compared to incoming IMU data. We set the threshold for creating a new Gaussian to $\epsilon = 5$ and inferred on (16). Our algorithm performed accurately in predicting the driving actions.

VIII. Conclusion

In this paper, we have presented a novel approach for on-line learning of driving behaviors in an unsupervised fashion. To this end, we have developed an entire Bayesian framework that is able to learn and adapt to new traffic situations and drivers. Visual traffic situations models have been modeled probabilistically from image streams and associated to motion segments from IMU data. Potential actions related to a particular traffic scene are jointly learned, providing predictions in unseen environments. Our system is suitable for lifelong learning since it is able to continuously update its models. We quantified the usefulness and the performance of this approach on a challenging urban dataset.

In a further work, we aim at improving our image representation with a more sophisticated model in order to determine which object in the scene induces an action. Action modeling at a higher level could be represented with a Hidden Markov Model (HMM).

Acknowledgment

This work has partly been supported by the EC under FP7-231888-EUROPA and by the DFG under SFB/TR-8.

References

[1] T. A. Ranney, "Models of driving behavior: A review of their evolution," *Accident Analysis & Prevention*, vol. 26, no. 6, pp. 733–750, Dec. 1994.
[2] E. Donges, "A two-level model of driver steering behavior," *J. Human Factors and Ergonomics Soc.*, vol. 20, no. 6, pp. 691–707, Dec. 1978.
[3] D. T. McRuer, "Human dynamics in man-machine systems," *Automatica*, vol. 16, no. 3, pp. 237–253, May 1980.
[4] R. A. Hess and A. Modjtahedzadeh, "A control theoretic model of driver steering behavior," *IEEE Control. Syst. Mag.*, vol. 10, no. 5, pp. 3–8, 1990.
[5] C. C. MacAdam, "Application of an optimal preview control for simulation of closed-loop automobile driving," *IEEE Trans. Syst. Man Cybern.*, vol. 11, no. 6, pp. 393–399, Jun. 1981.
[6] N. Oliver and A. P. Pentland, "Graphical models for driver behavior recognition in a smart car," in *Proc. IEEE Intel. Veh. Symp.*, 2000.
[7] A. Liu and D. Salvucci, "Modeling and prediction of human driver behavior," in *Proc. 9th Int. Conf. Human-Comput. Interaction*, 2001.
[8] J. Maye, L. Spinello, R. Triebel, and R. Siegwart, "Inferring the semantics of direction signs in public places," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2010.
[9] D. Meyer-Delius, C. Plagemann, and W. Burgard, "Probabilistic situation recognition for vehicular traffic scenarios," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2009.
[10] M. Heracles, F. Martinelli, and J. Fritsch, "Vision-based behavior prediction in urban traffic environments by scene categorization," in *Proc. Brit. Mach. Vis. Conf.*, 2010.
[11] N. Pugeault and R. Bowden, "Learning pre-attentive driving behaviour from holistic visual features," in *Proc. Europ. Conf. Comput. Vis.*, 2010.
[12] R. P. Adams and D. J. MacKay, "Bayesian online changepoint detection," University of Cambridge, Cambridge, UK, Tech. Rep., 2007.
[13] P. Fearnhead and Z. Liu, "On-line inference for multiple changepoint problems," *J. Roy. Stat. Soc. Series B*, vol. 69, no. 4, pp. 589–605, Apr. 2007.
[14] G. Casella and C. P. Robert, "Rao-Blackwellisation of sampling schemes," *Biometrika*, vol. 83, no. 1, pp. 81–94, Jan. 1996.
[15] R. E. Kass and A. E. Raftery, "Bayes factors," *J. Americ. Stat. Assoc.*, 1995.
[16] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003.
[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[18] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pat. Recog.*, 2005.
[19] R. E. Madsen, D. Kauchak, and C. Elkan, "Modeling word burstiness using the Dirichlet distribution," in *Proc. Int. Conf. Mach. Learn.*, 2005.
[20] A. Ranganathan and F. Dellaert, "Bayesian surprise and landmark detection," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2009.
[21] T. P. Minka, "Estimating a Dirichlet distribution," Microsoft Research, Tech. Rep., 2003.