Manifold-based Motion Prediction

C. Hermes, F. Kummert — Applied Informatics Group, Bielefeld University, 33615 Bielefeld, Germany
{chermes, franz}@techfak.uni-bielefeld.de

J. Wiest, U. Kreßel — Daimler AG, Group Research and Advanced Engineering, 89081 Ulm, Germany
{Juergen.Wiest, Ulrich.Kressel}@daimler.com

C. Wöhler — Image Analysis Group, TU Dortmund, 44227 Dortmund, Germany
christian.woehler@tu-dortmund.de

1 Overview

Advanced driver assistance systems (ADAS) have to cope with complex traffic situations, especially in the road crossing scenario. To detect potentially hazardous situations as early as possible, it is therefore desirable to know the position and motion of the ego-vehicle and vehicles around it for several seconds in advance. The standard motion prediction approach is the so-called *kinematic prediction*, i.e. constant yaw rate and constant acceleration, but it systematically fails at road intersections. The proposed approach uses previously observed driving manoeuvres to find a low-dimensional representation of common motion patterns. A probabilistic filter with mode detection tracks the vehicle's driving path and simultaneously predicts its motion several seconds ahead. Evaluated on a Differential GPS trajectory dataset, the proposed system shows significantly better results than the standard prediction approach for different prediction horizons.

2 Motion Prediction Method

The proposed prediction system is a functional mapping

$$f : H_{t-\Delta T:t} \rightarrow F_{t:t+\Delta t} \tag{1}$$

which takes the vehicle motion history for ΔT time steps as an input and predicts the vehicle motion Δt time steps ahead (time horizon).

2.1 Trajectory Concept

We represent the motion patterns of vehicles by trajectories, which are defined as ordered tuples

$$X = ((x_1, t_1), \ldots, (x_N, t_N)), \tag{2}$$

combining states x_i with a time stamp t_i. Therefore, each trajectory element describes the current state of the tracked object over time. This representation does not depend on a specific sensor type, and parts of the state may originate from different types of sensors. When applied to vehicle motion, the basic information is the object position in the 2D plane and the orientation angle relative to the ego-vehicle, here termed yaw angle. Furthermore, we found it useful to additionally encode the temporal derivatives, i.e. the velocity and the yaw rate, in the trajectory.
Various trajectories may differ in their length N, which makes this representation unsuitable for most regression techniques. Therefore, we apply the Chebyshev decomposition [1] to the trajectory representation in yaw angle and velocity components, which covers 99% of the trajectory information. This results in a trajectory representation as a vector of Chebyshev coefficients.

2.2 Manifold Representation

When a vehicle approaches an intersection, only a minor subset of possible yaw angle and velocity trajectory configurations are likely to occur. We thus assume that the set of common motion patterns is embedded in a low-dimensional sub-space of the Chebyshev coefficient space.

The Unsupervised Kernel Regression (UKR) [2] spans a low-dimensional manifold in the original high-dimensional space by the use of support vectors:

$$f(x) = \sum_i y_i \frac{K(x - x_i)}{\sum_j K(x - x_j)}, \tag{3}$$

where y_i denotes a support vector in the original coefficient space and x_i is the corresponding support vector on the manifold, also known as *latent variables*. The kernel $K(x)$ indicates the influence of each manifold support x_i vector to its neighbours and is normally chosen to be in exponential or quadratic form. One should notice that the UKR method only maps from the embedded space to the original coefficient space but not vice versa.

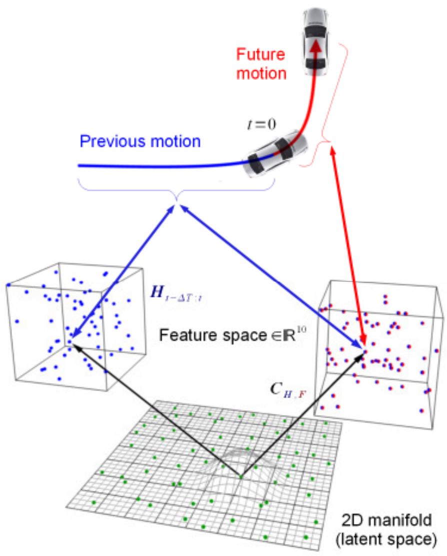

Figure 1 Motion prediction system with manifold and coefficient feature space.

Figure 1 visualises the UKR mapping. In the motion prediction system we make the support vector set $\{y_i\}$ interchangeable by two sets: The first set represents only the motion history $H_{t\text{-}\Delta T:t}$ for comparison with an observed history, whereas the second set integrates both motion history $H_{t\text{-}\Delta T:t}$ and future motion $F_{t:t+\Delta t}$ into one vector $C_{H,F}$ for motion prediction.

The latent variables x_i are determined during a learning stage. As suggested by [2] these support vectors are initialised by applying the Local Linear Embedding (LLE) approach and are then further refined in an iterative gradient descend process on the re-projection error using the RPROP algorithm. The coefficient support vector set $\{y_i\}=\{C_{H,F}{}^{(i)}\}$ is used during training.

2.3 Motion Prediction

As mentioned earlier, the UKR method is not able to provide a direct mapping from the coefficient space to latent space. Therefore, we utilise a probabilistic filter algorithm - namely a particle filter - as a temporal optimisation technique. The particle filter can handle multiple modes at the same time, i.e. if a vehicle approaches an intersection, the turning manoeuvres and straight driving are kept simultaneously until further information decides the direction.

At the first iteration step, the particle set is equally distributed over the manifold. During the weight update step, each particle is projected by the UKR mapping into the coefficient space using the $H_{t\text{-}\Delta T:t}$ support vectors, and each particle is then compared with the observed vehicle motion history. This results in a new weight value for each particle, by which the particles are drawn via importance sampling to a new particle set.

A mean-shift algorithm [3] detects the current probability density modes. The centre of each mode describes a possible driving manoeuvre and is projected first into the coefficient space using the $C_{H,F}$ support vectors and then into the trajectory space to predict the vehicle motion (cf. Figure 1).

4 Experimental Evaluation

The proposed vehicle motion prediction is tested in a real-world scenario with a Differential GPS (DGPS) dataset of approximately 23.6 km length (see Figure 2). Three intersections have been traversed by every possible configuration of driving manoeuvres.

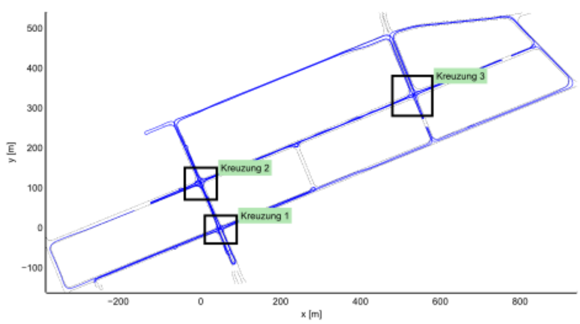

Figure 2 Bird's eye view on the DGPS dataset.

The motion prediction system is evaluated in a leave-one-out manner, i.e. one intersection is taken for testing and the remaining ones to train the UKR method.

Figure 3 shows the system's predictive power (blue) for prediction horizons from 1.0 s to 4.0 s compared to the kinematic prediction (green), i.e. constant acceleration and constant curve radius. The positional error shows the Euclidean distance between the estimated position by the prediction method and the ground truth, i.e. the path actually driven in the future. The errors in yaw angle and velocity indicate the deviation of the method's estimate in each component.

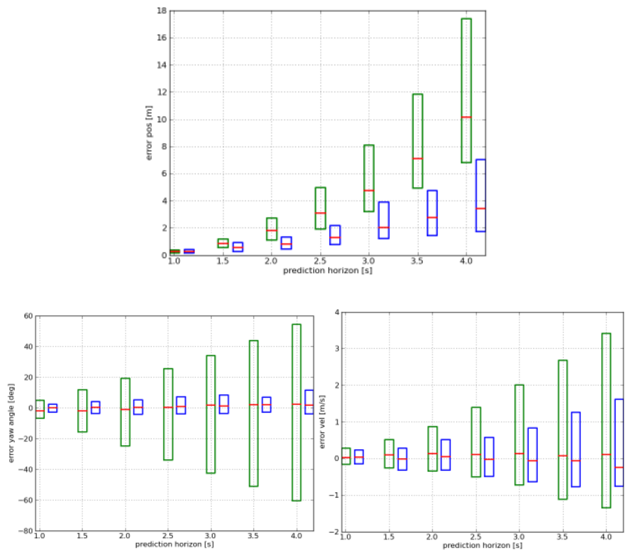

Figure 3 Prediction errors in position, yaw angle and velocity (vel) for various prediction horizons. Blue: output of the proposed system; green: kinematic prediction. The lower, middle and upper bound depict the 25%, 50% and 75% quantiles, respectively.

5 Conclusion

The results show a clear advantage by using learned motion patterns for prediction horizons of 2.0 s and above. However, a prediction horizon of 4.0 s usually does not yield an appropriate prediction because of a high error rate. Interestingly, the yaw angle error is quite stable for all prediction horizon values, while the velocity error causes the main distraction in the overall positional error for the proposed system.

6 References

[1] Press, W.; Teukolsky, S.; Vetterling, W. & Flannery, B.: *Numerical Recipes in C*, Cambridge University Press (1992).

[2] Meinicke, P.; Klanke, S.; Memisevic, R. & Ritter, H.: *Principal surfaces from unsupervised kernel regression*, IEEE Trans. Pattern Analysis and Machine Intelligence (2005), vol. 27, pp. 1379-1391.

[3] Comaniciu, D. & Meer, P.: *Mean Shift: A Robust Approach Toward Feature Space Analysis*, IEEE Trans. Pattern Anal. Mach. Intell. (2002), vol. 24, pp. 603-619.