Vehicle's Lane-changing Behavior Detection and Tracking — Mid Progress Report

Mengwen He
CyLab-ECE
4720 Forbes Ave. CIC 2223G
mengwenh@andrew.cmu.edu

Iljoo Baek
CyLab-ECE
4720 Forbes Ave. CIC 2224G
ibaek@andrew.cmu.edu

Abstract

The lane-level localization accuracy is very important for autonomous vehicles. The Global Navigation Satellite System (GNSS), e.g. GPS, is a generic localization method for vehicles, but is vulnerable to the multi-path interference in the urban environment. Integrating the vision-based relative localization result and a digital map with the GNSS is a common and cheap way to increase the global localization accuracy and thus to realize the lane-level localization. This project is to develop a mono-camera based lane-changing behavior detection and tracking algorithm module for the correction of lateral GPS localization. We implemented a Support Vector Machine (SVM) based framework to directly classify the driving behavior, including the lane keeping, left and right lane changing, from a sampled data of the raw image captured by the mono-camera installed behind the window shield. The training data was collected from the driving around Carnegie Mellon University, and we compared the trained SVM models w/ and w/o the Principle Component Analysis (PCA) dimension reduction technique. Next, we intend to compare the SVM-based classification method with the CNN method.

1. Introduction

The autonomous vehicle highly relies on the accurate localization technique because it enables reliable planning and control operations for the safe autonomous driving. The Global Navigation Satellite System (GNSS), e.g. GPS, GLONASS, Beidou (Compass), and Galileo, provides commercial localization devices with affordable price but low accuracy to vehicles. It works well while driving on the highway; however, it is vulnerable to the multi-path interference caused by trees, buildings, or overhead bridges in the urban area.

The prevalent method to enhance the global localization accuracy is to use 3D Light Detection and Ranging (LiDAR) sensors, e.g. Velodyne, to conduct registration with 3D point-cloud map. This can guarantee centimeter level localization accuracy [3] so that this method can be found on various commercial or experimental autonomous vehicles like Google, Baidu, Uber, and Toyota. However, this method is very expensive because of the 3D LiDAR and dense point-cloud map.

Therefore, the common and cheaper camera-based methods are preferred for affordable autonomous vehicles. The visual odometry method can achieve decimeter level relative localization [2]. By implementing filter-based, e.g. Kalman filter and its variants, or graph-based, i.e. non-linear least squares, methods, we can fuse the GNSS global localization with the visual odometry relative localization to get an enhanced global result.

In this project, we want to realize a vehicle's lane-changing behavior detection and tracking algorithm based on a mono-camera. Following the integration of GNSS and camera, we want to use the road lane information from a mono-camera to enhance the global localization result; therefore, the detection and tracking of the lane-changing behavior is required to tell which lane the vehicle is on. Coupled with a digital map with road lane information, we can laterally reduce the global localization error from the GNSS as Fig.1.

We employed a Support Vector Machine (SVM) based framework to detect the lane-changing behavior, and the training and test data/feature is directly sampled from the raw image of the mono-camera [Iljoo, please add a figure to show the raw image and sample data]. Therefore, we performed a Principle Component Analysis (PCA) dimension reduction technique to compress the feature's dimensionality as well as to keep more than 90% of its original energy.

Morevoer, because of the limitation of data size, we choose to use SVM instead of the Convolution Neural Network (CNN) [1] to classify the lane-changing behavior as lane-keeping, right lane-changing, or left lane-changing.

[1] 18-794 Pattern Recognition Theory homework assignment 3

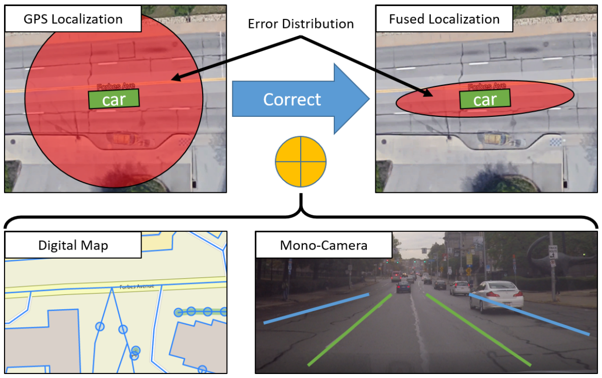

Figure 1. The illustration of our project's objective. We get a global localization result from GPS with a Gaussian error model. By integrating the digital map and the lane information from our lane-changing behavior detection and tracking algorithm, we can laterally correct the global localization error.

But, we plan to compare the training results between the SVM and CNN using the same training data.

We conducted the experiment around Carnegie Mellon University with a Go-Pro HD Camera (Hero 4, 1920×1080) mounted after the window shield of Iljoo's vehicle. We used the off-the-shelf LibSVM [1] to train the lane-changing behavior classfier w/ or w/o PCA dimension reduction.

2. Related Work

Iljoo, please add this section using the SVM-based lane-behavior detection paper and the LibSVM tutorial.

3. Approach and Algorithm

3.1. Training Data Collection and Labeling

Iljoo, please add how and why we sample the training data from the raw image (you can also include your lane detection method and result and explain why we don't use it currently for this project). Then show explain how to label the data (the two-level two-class classification problem)

3.2. PCA Dimension Reduction

The training data/feature's dimensionality is 16,000; therefore, we want to use the PCA to shrink its dimensionality and focus on the main components. Moreover, because the dimensionality is greatly larger than the data size, we will use the Gram Matrix Trick to accelerate the eigen decomposition operation.

3.2.1 Gram Matrix Trick

Given a centralized training data matrix X with dimension $d \times N$, we derive its covariance matrix as $\Sigma = E(XX^T)$. The PCA needs to solve the problem as below:

$$\begin{aligned} \Sigma\vec{v} &= \lambda\vec{v} \\ XX^T\vec{v} &= \lambda'\vec{v} \\ X^TXX^T\vec{v} &= \lambda' X^T\vec{v} \\ X^TX\vec{v}' &= \lambda''\vec{v}' \quad (\vec{v}' = \eta X^T\vec{v}) \end{aligned} \tag{1}$$

Therefore, we only need to solve the eigen decomposition of the Gram matrix X^TX with dimension $N \times N$. To get the final eigenvectors $\{\vec{v}_i\}$, we need to solve:

$$\left\{ \begin{aligned} XX^T\vec{v} &= \lambda'\vec{v} \\ \vec{v}' &= \eta X^T\vec{v} \end{aligned} \right. \Rightarrow \vec{v} = \eta' X\vec{v}' \tag{2}$$

Then we can use $\Sigma\vec{v} = \lambda\vec{v}$ to derive all the corresponding eigenvalues $\{\lambda_i\}$.

3.2.2 Dimension Reduction

Firstly, We sort the derived eigenvalues (associated with the corresponding eigenvectors) in a descending order, and we calculate the total energy as below:

$$\Lambda = \sum_{i=1}^{d} \lambda_i \tag{3}$$

Then, we choose the first M biggest eigenvalues whose summation is just larger than a threshold ratio r of the total energy Λ.

$$\begin{aligned} \textstyle\sum_{i=1}^{M} \lambda_i &\geq r\Lambda \\ \textstyle\sum_{i=1}^{M-1} \lambda_i &< r\Lambda \end{aligned} \tag{4}$$

Finally, we use the first M biggest eigenvalues' corresponding eigenvectors to form a PCA dimension reduction matrix P as below:

$$P = [\vec{v}_1, \ldots, \vec{v}_M] \tag{5}$$

Therefore, the dimension reduced new centralized trainidng data matrix is $X' = P^TX$. If we use the PCA dimension reduction on the training data as well as the SVM, we also need to apply the dimension reduction on the test data Y following these two steps:

1. Centralize the test data with the mean (μ_X) of the training data.
2. Reduce its dimension to get $Y' = P^T(Y - \mu_X)$

3.3. LibSVM Training and Test with RBF Kernel

Iljoo, we can finish this section together.

3.4. Lane Changing Behavior Tracking

Not done yet, will be finished after the mid-progress report.

4. Experiments and Progress

Iljoo, please add your current experiment results. Because this is a mid progress report, we only need to show we have passed all the main steps with imperfect results.

4.1. Data Collection and Labeling

Iljoo, please add your UI software and some sample data

4.2. SVM Training Result

4.2.1 Without PCA Dimension Reduction

4.2.2 With PCA Dimension Reduction

4.3. CNN Training Result

Not done yet, will be finished after the mid-progress report.

5. Conclusion

References

[1] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[2] C. Merfels and C. Stachniss. Pose fusion with chain pose graphs for automated driving. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[3] E. Takeuchi and T. Tsubouchi. A 3-d scan matching using improved 3-d normal distributions transform for mobile robotic mapping. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3068–3073. IEEE, 2006.