Multi-modal Sensor Registration for Vehicle Perception via Deep Neural Networks

Michael Giering, Kishore Reddy, Vivek Venugopalan
Decision Support & Machine Intelligence Group
United Technologies Research Center
E. Hartford, CT 06060, USA
Email: gierinmj, kkreddy, venugov@utrc.utc.com

Abstract

When performing multi-modal fusion to perform an analytic task, spatio-temporal registration of the incoming signals is often a prerequisit to analyzing the fused data and critical to the stability of the analysis. Lidar-Video systems like on those many driverless cars are a common example of where keeping the Lidar and video channels registered to common physical features is important. We develop a deep learning method that takes multiple channels of heterogeneous data to detect the misalignment of the Lidar-video inputs. A number of variations were tested on the Ford LV driving test data set with minimal tuning of the deep conv nets parameters.

1 Motivation

Navigation and situational awareness of optionally manned vehicles requires the integration of multiple sensing modalities such as LIDAR and video, but could just as easily be extended to other modalities including Radar, SWIR and GPS. Spatio-temporal registration of information from multimodal sensors is technically challenging in its own right. For many tasks such as pedestrian and object detection tasks that make use of multiple sensors, decision support methods rest on the assumption of proper registration. Most approaches [][] in LIDAR-video for instance, build separate vision and lidar feature extraction methods and then try to identify common anchor points in both. Generating a single feature set on Lidar, Video and optical flow, enables the system to to capture mutual information among modalities more efficiently. The ability to dynamically register information from the available data channels for perception related tasks can alleviate the need for anchor points *between* sensor modalities. We see auto-registration as a prerequisit need for operating on multi-modal information with confidence.

Deep neural networks lend themselves in a seamless manner for data fusion on time series data. It has been shown [Ng multimodal] for some problems that features generated on the fused information [] can provide insight that neither input alone can. In effect the ML version of, ”the whole is greater than the sum of it’s parts”.

Speed constraints of real time navigation also constrain model selection. The trained nnets easily run within the real-time constraints of common frame rates and lidar data collection.

From an applied research perspective, it is possible to create such systems with far less overhead. The need for domain experts and hand-crafted feature design are lessened, thereby allowing more rapid prototyping and testing.

The generalization of autoregistration across multiple assets is clearly a path to be explored.

By including optical flow as input channels, we imbue the nnet with information on the dynamics observed across time steps.

2 Previous Work

Need some references to define the state of the art

3 Problem Statement

Being able to detect and correct the misalignment (registration, calibration) among sensors of the same or different kinds is critical when operating on the fused information emanating from them. For this work Deep Convolutional Neural Networks (DCNN) were implemented for the detection of small spatial misalignments in LIDAR and Video frames. The data collected from a driverless car was chosen as the multi-modal fusion test case. LIDAR-Video is a common combination for providing perception capabilities to many types of ground and airborne platforms including driverless cars Thrun (2011).

3.1 Ford LIDAR-Video Dataset and Experimental Setup

Figure 1: Left: The modified Ford F-250 pickup truck. Right: Relative position of the sensors with respect to the body frame.

The FORD LIDAR-Video dataset Pandey et al. (2011) is collected by an autonomous Ford F-250 vehicle integrated with the following perception and navigation sensors as shown in Figure 1:

- Velodyne HDL-64E LIDAR with two blocks of lasers spinning at 10 Hz and a maximum range of 120m.
- Point Grey Ladybug3 omnidirectional camera system with six 2-Megapixel cameras collecting video data at 8fps with 1600×1600 resolution.
- Two Riegl LMS-Q120 LIDAR sensors installed in the front of the vehicle generating range and intensity data when the laser sweeps its 80°field of view (FOV).
- Applanix POS-LV420 INS with Trimble GPS system providing the 6 degrees of freedom (DOF) estimates at 100 Hz.
- Xsens MTi-G sensor consisting of accelerometer, gyroscope, magnetometer, integrated GPS receiver, static pressure sensor and temperature sensor. It measures the GPS coordinates of the vehicle and also provides the 3D velocity and 3D rate of turn.

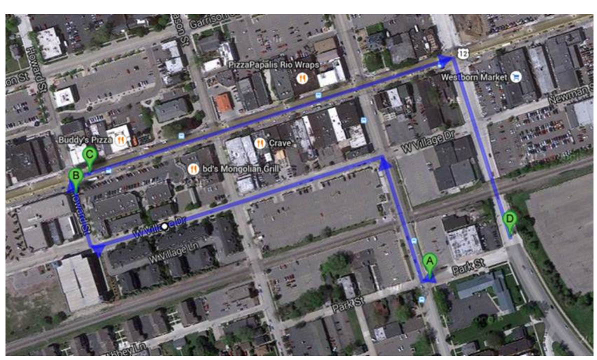

Figure 2: Training (A to B) and testing (C to D) tracks in the downtown Dearborn Michigan.

This dataset is generated by the vehicle while driving in and around the Ford research campus and downtown Michigan. The data includes feature rich downtown areas as well as featureless empty parking lots. As shown in Figure 2, we divided the data set into training and testing sections A to B and C to D respectively. They were chosen in a manner that minimizes the likelihood of contamination between training and testing. Because of this, the direction of the light source is never the same in the testing and training sets.

3.2 PREPROCESSING

At each video frame timestep, the inputs to our model consisted of C-channels of data with C ranging from 3-6 channels. Channels consisted of inputs that included greyscale and (R,G,B)-video channels, horizontal and vertical optical flow and Lidar depth information. Each channel was cropped to a uniform 800x250 pixels. Each time step has an 800 x 250 x C array of integer values.

These arrays were subdivided into p x p x C patches at a prescribed stride. For any experiment we can denote the preprocessing parameters

- R,G,B — Frame color channels.
- U,V — optical flow channels.
- L — lidar depth channel.
- C — number of input channels.
- p — patch size.
- s — stride.

For a given frame of size 800 x h there are approximately n= (800 x h)/s patches (exact number?). The training and test sets had X and Y frames respectively, therefore the entire data set consists of N = n x X inputs of the patch-size dimension.

Preprocessing is repeated O times, where O is the number of offset classes. For this work we used two setups. A 5 class, linearly distributed set of offsets and a 9 class elliptically distributed set of offsets. (see figure x) For each offset class, **Kishore explain how you generated the data.**

In order to accurately detect misalignment in the LV sensor data, we've assumed there needs to be a lower bound on the amount of information present in each channel. For this data set, L was the only channel with regions of low information. A preprocess step was to eliminate all patches corresponding to L data with variance ¡x. This leads to the elimination of the majority of foreground patches in the data set, reducing the size of the training set by **z pct KISHORE**

4 MODEL DESCRIPTION

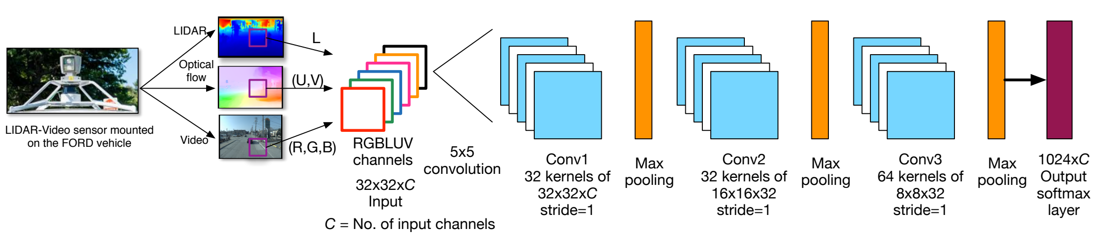

Figure 3: Experimental setup of the LIDAR-Video DCNN

need to describe the parameters post-processing,classification metric for each patch,a table with common params for the experiments would help,voting scheme

The model consists of a 4-layer **?** CNN classifier *see image of network* that estimates the offset between the LV inputs at each time step. For each patch within a timestep, there are O variants with the LVF inputs offset by the predetermined amounts. The CNN outputs to a softmax layer, thereby providing an offset classification value for each patch of the frame. figure x: In the 5 class example we color each patch of the frame with a color corresponding to the predicted class.

For each frame a simple voting scheme is used to aggregate the patch level offset predictions to frame level predictions. A sample histogram of the patch level predictions is show in figure x.

4.1 Optical Flow

kishore, please discuss the motivation to include dynamics, how we performed it and how we'd need to do it if running in real time. this is where we can point ot proof that it improves prediction.

Optical flow gives a rough estimate of velocity at each pixel given two consecutive frames. Given a stationary scene, the optical flow between two consecutive frames with small motion depends on the distance of the pixel from the camera. For example, closer the pixel, higher the displacement. We intent to use optical flow obtained from video as a proxy for depth. Chase et al. (2008) demonstrated that optical flow can be performed in real-time on FPGA and GPU architectures. In our work, we used the algorithm described in Liu (2009) for computing optical flow.

5 Experiments and Post-processing

Need a complete list of the experiments run images to visualize the frame level results please place any confusion matrices and your comments on what you think the results say. feel free to suggest any tables or other visuals to include.

5.1 5 class tests

In our initial tests, the linearly distributed set of 5 offsets of the LV data were performed. Table 1 lists the inputs and CNN parameters explored ranked in the order of increasing accuracy **(define accuracy and other cm metrics), include training vs test error and conf mats if room allows**.

As can be seen ...

5.2 9 class tests

The subsequent tests were designed to understand whether the simple linear displacement model of the 5-class test could be generalized to a model capable of discriminating multiple directions and displacement magnitude. To achieve this 8 positions were chosen on an ellipse along with it's center **describe the parabola**. LV was offset in a manner similar to the 5 class test. Nine training and test sets were generated and an identical patch level CNN was constructed differing only in the 9 class softmax output layer.

Table 2 lists the inputs and CNN parameters explored ranked in the order of increasing accuracy **(define accuracy and other cm metrics), include training vs test error and conf mats if room allows**.

Discussion: what results confirmed expectations or surprised us (grey scale). Can we confidently say optical flow improves prediction.

6 Conclusions and Future Work

We did it. We're great.

future: implement a method that doesn't require ground truth and also generalizes easily to a wide array of sensors. Test it on data collected from airborne platforms that are noisier and have more degrees of freedom.

7 References

populate the papers to be cited in the folder and if possible the bib file

REFERENCES

Chase, J., Nelson, B., Bodily, J., Wei, Zhaoyi, and Lee, Dah-Jye. Real-Time Optical Flow Calculations on FPGA and GPU Architectures: A Comparison Study. In *Field-Programmable Custom Computing Machines, 2008. FCCM '08. 16th International Symposium on*, pp. 173–182, April 2008. doi: 10.1109/FCCM.2008.24.

Liu, Ce. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2009. AAI0822221.

Pandey, Gaurav, McBride, James R, and Eustice, Ryan M. Ford Campus Vision And Lidar Data Set. *The International Journal of Robotics Research*, 30(13):1543–1552, 2011.

Thrun, Sebastian. Google's driverless car. *Ted Talk, Ed*, 2011.