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 multi-modal 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 Bodensteiner & Arens (2012) in LIDAR-video for instance, build separate vision and lidar feature extraction methods and 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

Kishore here

A great amount has been published on various multimodal fusion methods. The most common approach taken is to generate features of interest in each modality such as lines and create a decision support mechanism that aggregates the outputs across modalities. If spatial alignment is required across modalities, as it is for LiDAR/ video such filter methods Thrun (2011) are required to ensure the registration across modalities. These filter methods for leveraging 3D LiDAR and 2D images are geometric in nature and make use of projections between the different data spaces.

In recent years, a number of papers have been written on the topic of using deep neural networks to analyze multimodal sensor inputs including audio/video Ngiam et al. (2011) Kim et al. (2013), image/text Srivastava & Salakhutdinov (2012), image/depth Lenz et al. (2013) and Lidar/video. To the best of our knowledge the use of multimodal deep neural networks for dynamic real time LIDAR/video registration has not been presented.

A common question often arises in data fusion methods, which is "at what level should features from the differing sensor streams be brought together?" Most similar to the more traditional data fusion methods is to train DNN's independently on sensor modalities and then use the high-level outputs of those networks as inputs to a subsequent DNN. This is analogous to the earlier example of learning 3D/2D features and subsequently identifying common geometric features.

It is possible however to apply DNN's in a more agnostic view enabling a unified set of features to be learned across multimodal data. In these cases the input channels aren't differentiated. Unsupervised methods in particular applying DBM's for learning such joint representations have been successful.

DCNN's enable a similar agnostic approach to input channels. A significant difference of course is that target data is required to train them as classifiers. *find examples in the literature*. This is the approach chosen by us for automating the registration of LiDAR/video/optical-flow, in which we are combining 1D/3D/2D data representations to learn a unified model across all 6D.

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 DCNN's were implemented for the detection of small spatial misalignments in Lidar and Video frames. The data was collected from a driverless car was chosen as the multi-modal fusion test case. LV is a common combination for providing perception capabilities to many types of ground and airborne platforms including driverless cars [google, ford].

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:

- Velodyne HDL-64E LIDAR with two blocks of lasers spinning at 10 Hz and a maximum range of 120m.

3.1 Ford LIDAR-Video Dataset and Experimental Setup

Kishore -Detailed description of the ford data set [], our test and training and the justifications for it. framerate. Vivek - a brief description of the hardware used.

Ford Campus Vision and Lidar dataset (Pandey et al., 2011) (Ford-VL) is a publicly available dataset collected by an autonomous ground vehicle mounted with multiple sensors shown in Figure 1. In this work we used time-registered Velodyne 3D-lidar scanner (Lidar) and Point Grey Ladybug3 omnidirectional camera(Video) data collected while driving in a loop in the downtown Dearborn Michigan. Velodyne sensor has a range of 120 m with the lidar spinning at 10 Hz. Video is captured at 8 fps (1600x600 resolution) using array of six 2-Megapixel cameras to collect video from 80% of sphere.

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 lighting is source is never the same in the testing and training sets. If our methods are generalizable, they should be able to overcome this bias in the data.

Figure 1: Left: The modified Ford F-250 pickup truck Pandey et al. (2011). Right: Relative position of the sensors with respect to the body frame Pandey et al. (2011).

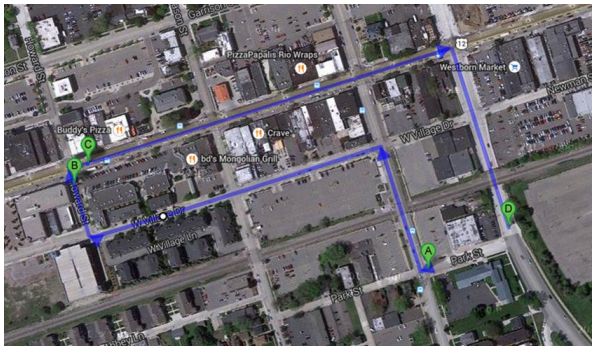

Figure 2: Training (A to B) and testing (C to D) tracks in the downtown Dearborn Michigan.

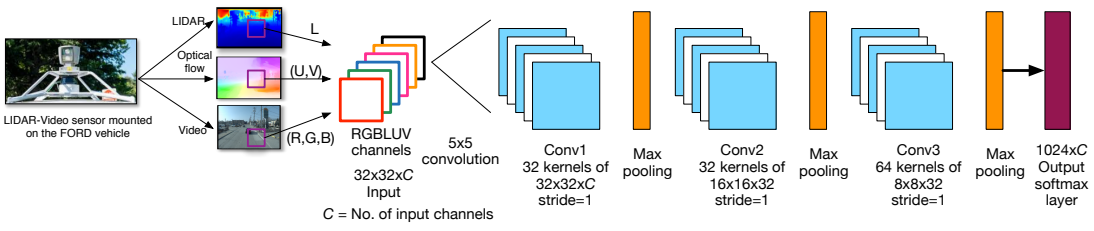

Figure 3: Experimental setup of the LIDAR-Video DCNN

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 components of optical flow computed from two consecutive frames and Lidar depth information. Each channel was cropped to a uniform 800x??? pixels. Each time step has an 800 x ??? 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.**

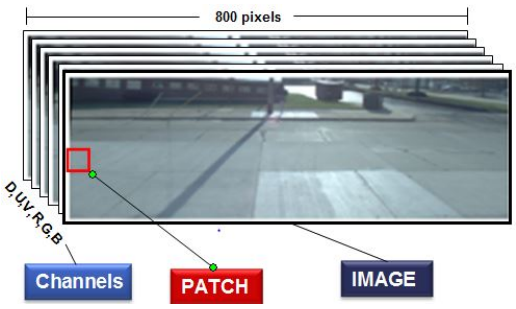

Figure 4: At each time step the channels were sampled at the noted patch size at a fixed stride

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

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

Our models for auto-registration are DCNN's trained to classify the current misalignment of the LiDAR/video data streams into one of a predefined set of offsets. DCNN's are probably the most successful deep learning model to date on fielded applications. The fact that the algorithm shares weights in the training phase, results in fewer model parameters and more efficient training. DCNN's are particularly useful for problems in which local structure is important, such as object recognition in pictures. The alternating steps of convolution and pooling **(seen in figure X)** generates features at multiple scales which in turn imbues DCNN's with scale invariant characteristics.

The basic CNN structure we use has the following three characteristics: 1) input locality: We learn a set of filters, each of which receives the input from a local range of frequencies; 2) weight sharing: Each filter shifts along the frequency axis while computing the output with tied filter weights (this is mathematically equivalent to the ubiquitous convolution operation in DSP); and 3) max pooling or sub-sampling: High-level features with lower resolution are produced by the CNN. A combination of these characteristics endows the CNN with invariant properties for the input acoustic patterns that shift along the frequency axis.

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. **?** demonstrated that optical flow can be performed in real-time on FPGA and GPU architectures. In our work, we used the algorithm described in **?** 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

Bodensteiner, Christoph and Arens, Michael. Real-time 2D Video 3D LiDAR Registration. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 2206–2209. IEEE, 2012.

Kim, Yelin, Lee, Honglak, and Provost, Emily Mower. Deep Learning for Robust Feature Generation in Audiovisual Emotion Recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 3687–3691. IEEE, 2013.

Lenz, Ian, Lee, Honglak, and Saxena, Ashutosh. Deep Learning for Detecting Robotic Grasps. *arXiv preprint arXiv:1301.3592*, 2013.

Ngiam, Jiquan, Khosla, Aditya, Kim, Mingyu, Nam, Juhan, Lee, Honglak, and Ng, Andrew Y. Multimodal Deep Learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 689–696, 2011.

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.

Srivastava, Nitish and Salakhutdinov, Ruslan. Multimodal Learning With Deep Boltzmann Machines. In *Advances in neural information processing systems*, pp. 2222–2230, 2012.

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