A DNN-LSTM based Target Tracking Approach using mmWave Radar and Camera Sensor Fusion

Arindam Sengupta, Feng Jin and Siyang Cao
Department of Electrical and Computer Engineering
University of Arizona
Tucson, AZ USA
Email: {sengupta, fengjin, caos}@email.arizona.edu

***Abstract*—A new sensor fusion study for monocular camera and mmWave radar using deep neural network and LSTMs is presented. The proposed study includes a decision framework to produce reliable output when either sensor fails. Experiment results to demonstrate single sensor uncertainty and the proposed method's advantages are also presented.**

***Index Terms*—Sensor Fusion, DNN, LSTM, Target Tracking, mmWave Radar, Monocular Camera**

I. Introduction

Target tracking is a topic of immense interest, primarily in the computer vision (CV) community, as it provides artificial intelligence (AI) driven applications access to temporal real world data. Some emerging applications include, but are not limited to, autonomous vehicles, automated security surveillance, and traffic management [1]–[4]. Furthermore, sensor fusion, i.e. using data from multiple sensors to solve a particular task, is extremely desired, especially for the problem of effective target localization and tracking. Monocular camera provides high-resolution images that allows for localization using segmentation and morphological operations with good resolution in the cross-range. However extracting highly accurate depth information using monocular cameras is non-trivial and challenging. Moreover, cameras fail to operate in adverse weather conditions such as rain/fog or in situations with poor illumination such as at night. Radars on the other hand use its own radio-waves to illuminate targets which allows them to operate in any conditions. Radars also localize targets with a highly accurate depth information but a low cross-range resolution. As camera and radars play to each others' advantages and disadvantages, there has been interest to fuse data from these two sources to obtain an accurate position of a target, as shown in Fig. 1, and subsequently track its trajectory.

In an early 2002 work [5], FADE: a vehicle detection and tracking system, is proposed, that uses monocular camera and radars, and fuse the 12 resulting features to predict/propose position of a vehicle. In [6], a vehicle detection system using a high-level camera-radar fusion is presented. First the radar detections are mapped onto the image plane to identify a region of interest (ROI), followed by a bounding box generation. Vertical symmetry of vehicles within the ROI was exploited to validate the presence of a vehicle. In [7], another sensor-fusion scheme is presented to detect obstacles, for collision avoidance, and track using an Unscented Kalman Filter (KF).

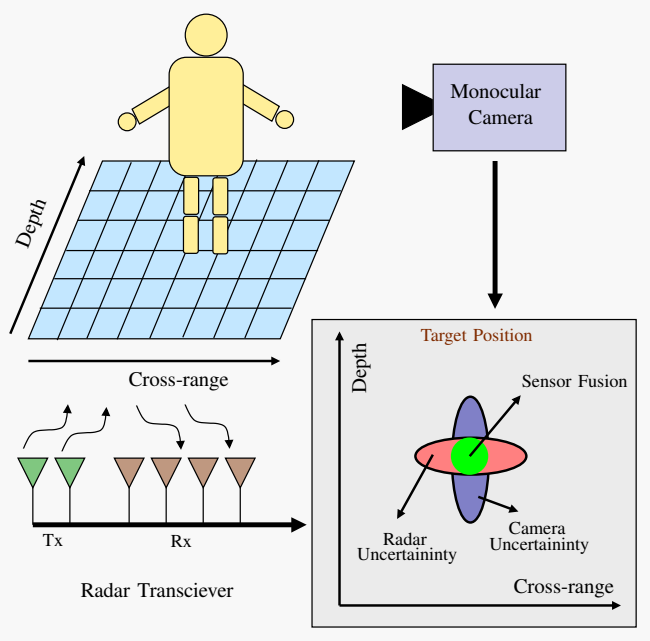

Fig. 1. Radar and camera sensors provide uncertainity in the cross-range and depth respectively, when used independently. A sensor fusion approach using both radar and camera data provides a more accurate position of the target.

In [8], data from LIDAR, Radar and Camera was used to detect and track moving objects. An on-road obstacle detection scheme is proposed in [9]. The radar data was first projected on the camera coordinate system by using a transformation matrix computed from the intrinsic and extrinsic camera parameters, and was used to validate detections from the camera. In [10], the transformed radar-to-camera data was used to identify an ROI and draw a square bounding box, followed by an active-contour method to detect and validate the presence of a target and use KF for tracking.

However, these methods do not suggest the implications when either of the sensors fail, as data from both the sources are equally and heavily relied on. The improvements in the tracking accuracy from the sensor-fusion approach, as opposed to using a single sensor system, both in the depth and cross-range, have also not been addressed adequately. Furthermore, several pre-processing schemes require prior knowledge of the camera parameters to formulate a transformation matrix, and

978-1-7281-1416-3/19/$31.00 ©2019 IEEE

KF-based tracking algorithms rely on the assumption of the system being linear with a Gaussian distribution. For a highly accurate target tracking application, the system could be non-linear, and the sensor uncertainity could take up a non-gaussian distribution, which would make higher-complexity tracking algorithms such as a particle filter or extended KF a more appropriate choice.

To overcome and address some of the aforementioned challenges, a target tracking approach using a monocular camera and millimeter-wave (mmWave) radar sensor fusion is proposed. The target is first localized in the image frame and a bounding box is drawn around it. A trained deep neural network (DNN) is then used to generate the position of the target using the bounding box dimensions. In the fusion stage, the DNN generated position from the camera is associated with the radar returned position of the target. A long short-term memory (LSTM) module is used to generate a continuous target trajectory using the fused data. The proposed system is therefore camera parameters and system distribution independent. The fusion decision framework also has the ability to track an object with reasonable accuracy as long as it is detected by at least one sensor, and with a high accuracy when detected by both the sensors.

The paper is organized as follows. Section II presents a quick overview of the radar signal processing chain and LSTMs. The proposed architecture is presented in detail in Section III, followed by the results and discussion in Section IV. Finally, the study is summarized and the potential future work is presented in Section V.

II. Background Theory

A. Radar Signal Processing Chain

The frequency modulated continuous wave (FMCW) mmWave radar sensor, primarily aimed at target localization, has emerged as a lucrative sensor for various applications in recent years, such as automotive advanced driver-assistance systems (ADAS), unmanned aerial vehicle (UAV) altimeter, robotic navigation, etc. This radar sensor transmits a linear frequency modulated (LFM) signal, also referred to as a 'chirp', with up to 4 GHz bandwidth at the carrier frequency of 79 GHz [11]. By adopting stretch processing, which essentially mixes the transmitted chirp signal and the received echo from the target, the beat frequency of the signal is determined. The mixed signal is then fed into an analog-to-digital converter (ADC), that samples the signal into the digital domain for further signal processing. In general, in one coherent processing interval (CPI), multiple consecutive chirps through an antenna array are sent to solve for range, velocity and angle of the target at the same time. As a result, a 3-D radar data-cube is generated, where the fast time dimension is from the sampling of one chirp, the slow time dimension is from the sampling across one CPI, and the phase center dimension is the spatial sampling over multiple antenna channels [12].

First, by performing the Fast Fourier Transform (FFT) along the fast time dimension, the range of the target is obtained. The beat frequency is proportional to the time delay between the transmitted chirp and received echo, which in turn is representative of the range between the radar sensor and the target. Second, the Doppler shift in the frequency, resulting from target motion during one CPI, is obtained by by performing an FFT along the slow time dimension. This process yields the velocity information of the target. Third, the constant phase difference between two consecutive antenna channels is exploited by performing an FFT along the phase center dimension to solve for the angle of the target. The multiple-input and multiple-output (MIMO) technique can also be applied to improve the angle resolution further.

Furthermore, the desired target signal is isolated from clutter due to static objects, such as road curbs and trees using moving target indication (MTI). The constant false alarm rate (CFAR) detection estimates the noise floor around the target to ensure a more reliable target detection. A density-based spatial clustering of applications with noise (DBSCAN) clusters and distinguishes multiple targets [13]. A Kalman filtering based tracking follows the target's trajectory and makes a smoother estimation of target's position. The signal processing chain for the mmWave radar sensor is shown in the overview of proposed architecture in Section III. It has to be noted that, due to its low-cost, the mmWave radar sensor typically has only a few antenna channels, thus the angle resolution is poor. Although the range resolution is very promising, the cross-range resolution is comparatively poor, on account of its direct dependency on angular resolution.

B. Neural Networks and LSTM

With the advent of graphical processing units (GPUs), neural networks have ascended as one of the primary machine learning methods for object classification [14], [15]. Neural networks are computational models that are inspired from biological neural networks [16]. An artifical neuron or node accepts a weighted input and provides an output following a projection via activation functions. A typical feed-forward DNN can be constructed by introducing multiple fully connected hidden layers between the input and output layers, where each layer consists of several neurons. The training process involves feeding the DNN with labeled inputs with the objective to minimize the cost function, or in this case the mean square error (MSE) of the predicted output with respect to the desired output. This is achieved by vanilla/variation of gradient descent using back-propagation [17].

Recurrent Neural Networks (RNNs) are a class of neural networks with a feedback loop. This implies, that at any given instant, the previous state of the node is also taken into account while determining the output. This temporal dependency, or in an essence memory, find several applications in natural language processing, speech recognition and time series prediction. RNNs can also be viewed as DNNs with shared weights, and are also trained using back-propagation. Therefore, for long sequences, RNNs lose track of long term dependencies due to vanishing and exploding gradients. LSTMs are a variant of vanilla RNNs that overcome

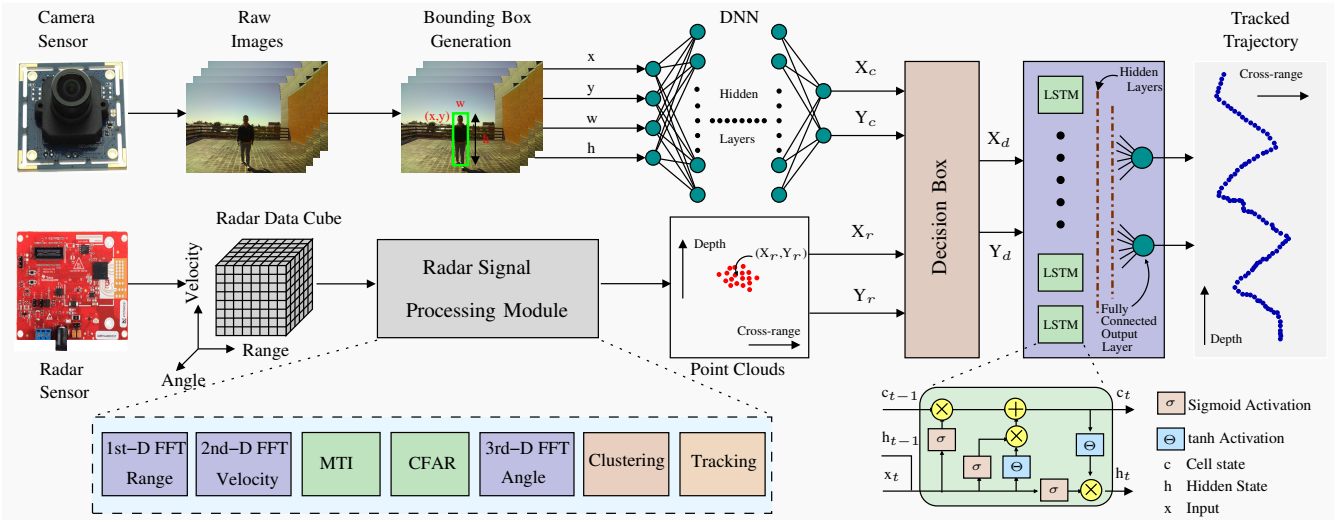

Fig. 2. Camera captures images of the scene where the target is first localized using a bounding box, whose coordinates are then fed to a trained DNN to obtain the position of the target (X_c, Y_c). Parallely, the radar sensor receives reflected signals from the target in the form of a radar datacube, and following signal processing and clustering, the position of the centroid is obtained (X_r, Y_r). The radar and camera data is fed to a decision box, where sensor fusion occurs and the fused data (X_d, Y_d) is passed on to the LSTM network for the tracking trajectory update.

these challenges. LSTMs include a common long-short-term memory line, also known as the cell state, that keeps updating based on a non-linear combination of its previous state, the current input and the hidden state of the module. These special units allow LSTMs to 'remember' or 'forget' previous states which makes them more robust in preserving useful memory [18].

III. Proposed Architecture

As introduced in the previous sections, sensor fusion makes it the preferred approach to detect and track targets accurately. Adverse weather conditions and poor illumination lead to camera failing to make detection, while radar performance in such conditions remain unaffected. On the other hand, strong clutter level may cause radar detection failures while camera may still clearly show the target. In general, radars provide us with highly accurate depth information, while camera provides an accurate lateral position of the target. Therefore, a mmWave radar and camera sensor fusion system is proposed, which provides us with high-accuracy target trajectory as opposed to using just a single sensor.

Moreover, it is also desired to be able to continuously make detections even if a single sensor fails. It is possible for the radar sensor to continuously make detections even if the target cannot be detected by the camera. However for the scenario where the radar fails, a monocular camera would still be able to provide us with the lateral position information about the target, but not the depth information directly. To overcome this challenge, an approach to extract depth information of targets using bounding box coordinates from camera images, and a DNN, is presented. This makes our proposed system also robust to single sensor failures.

The overview of the system architecture is shown in Fig. 2 On the camera side, the target position is predicted by a DNN with the input of the bounding box parameters from the images. On the radar side, the radar parallely detects the target centroid position, following the radar signal processing chain. A decision is made to output a better depth and cross-range depends on which scenario it is, i.e. both sensors work or either one fails. Finally, a robust and high accuracy target trajectory using the optimized depth and cross-range of the target output from the decision box is generated by the LSTM network. The system architecture is explained in details in the following subsections.

A. Camera Bounding-Box DNN

The camera provides us with high resolution images of the scene, that also includes the target of interest. The target can be localized from the background using image segmentation, contouring followed by bounding box generation. The bounding box coordinates $\{x, y, w, h\}$ represent the top-left pixel coordinates, width (in pixels) and height (in pixels) respectively. During target's lateral movement at a fixed depth, $\{y, w, h\}$ would continue to remain the same, with x changing to x' linearly with respect to the displacement. Therefore, it becomes easier to empirically determine the cross-range information of the target from images, and the high resolution of the images provide this with high accuracy.

However, when the target moves in the direction of depth, all four bounding box coordinates $\{x, y, w, h\}$ change to $\{x', y', w', h'\}$. Moreover, closed-form expressions of the non-linear relationship of $\{x', y', w', h'\}$ with respect to the depth and cross-range positions of the target, is difficult to model. Therefore, a DNN to model this relationship is proposed with the objective to minimize a cost function. The input to the DNN would be a 4×1 vector of bounding box coordinates $\{x, y, w, h\}$ and the output would provide us with the depth and cross-range coordinates of the target. During the training process, DNN is provided with labelled bounding box data, that is collected from the experimental setup. The weights

of the DNN nodes are optimized by trying to minimize the MSE of the predicted output and the ground truth. The DNN predictions ensure that we have information about the target position, in both depth and cross-range directions, from the camera data alone - even if the radar fails to make a detection.

B. Decision Box

The high-level sensor fusion in our architecture is achieved in the decision box. The radar-camera system captures and returns data parallelly, albeit in an asynchronous fashion with different frame rates. Every camera frame checks for the target, and if detected, returns us with the position of the target (X_c, Y_c) via bounding-box driven DNN prediction. Similarly every radar detection also generates a radar frame that provides us with a target position (X_r, Y_r), following the radar signal processing chain described in Section II(A). Here $X_{c/r}$ and $Y_{c/r}$ represents the 2-D position of the target in depth and cross-range axis, respectively. In order to be able to correlate a specific camera frame to a radar frame, we use the additional meta-data accompanying the measurements, which in our case is the coordinated universal time (UTC) timestamp. As radar and camera provides a better detection for depth and cross-range respectively, in the event of simultaneous detection, the fused position of the target is given by $(X_d = X_r, Y_d = Y_c)$. The UTC of a single camera frame is compared to all the radar frames and the index of the radar frame with the least absolute UTC difference is considered. Simultaneous detection is said to occur if the UTC timestamp difference between the radar and camera detection frames is $\leq$ 25ms. Otherwise, at least one of the sensors is said to have missed the detection, and the position of the target from the other available sensor is output to ensure continuous detection. The algorithm for the decision box is shown in Algorithm 1. $[X_c, Y_c]$, $[X_r, Y_r]$, C_{UTC}, R_{UTC} are the Camera & Radar detected positions and timestamps, respectively. The index $[i]$ corresponds to the radar frame that yeilds the minimum $|C_{UTC}\text{-}R_{UTC}[i]|$. The output of the decision box is fed to the LSTM network to continuously track the target's trajectory.

Algorithm 1 Algorithm for Decision Box

Input: $[X_c, Y_c]$, $[X_r, Y_r]$, C_{UTC}, R_{UTC}
Output: $[X_d, Y_d]$

```
while Camera frames recieved do
  if |C_UTC-R_UTC[i]| ≤25ms then
    if Target in camera frame then
      X_d ← X_r[i]
      Y_d ← Y_c
    else
      X_d ← X_r[i]
      Y_d ← Y_r[i]
    end if
  else if Target in camera frame then
    X_d ← X_c
    Y_d ← Y_c
  end if
end while
return [X_d, Y_d]
```

C. Target Trajectory using LSTM

As introduced in Section II-B, LSTMs belong to the class of RNNs with special units to overcome the vanishing gradient problem. LSTMs have been widely used for NLP applications, especially for text prediction. However, lately, LSTMs are also being used for time-series predictions and target tracking. In our proposed approach, a 3 layer LSTM network is used to continuously make updates to the target trajectory, based on the last 5 detected positions of the target. In order to achieve this, the LSTM network is first trained on radar data, collected experimentally, for several target trajectories. 5 frames of data is fed to the LSTM network and the MSE is computed by comparing the predicted output to the 6^{th} frame of data.

IV. Experimental Results and Discussion

In order to evaluate our proposed model, we primarily performed 2 sets of experiments. The first was to validate the fused output accuracy in both depth and cross-range directions, and it's improvement over a single sensor detection. Second was to validate the robustness of the proposed model in case of a single sensor failure.

A. Setup

The experiment environment was setup in an open area in the Electrical and Computer Engineering (ECE) department at the University of Arizona. Texas Instruments AWR1642BOOST [19] mmWave radar sensor and a USB RGB camera were used to collect the data. A 3-D printed framework was used to mount the RGB camera and the mmWave radar sensor together, and both these sensors were connected to a laptop via USB cables for data collection, as shown in Fig. IV-A. The camera was set to an 800×600 resolution operating at 30 frames per second(fps). The mmWave radar sensor was configured to transmit 256 chirps (3GHz bandwidth under 51.2 us duration) in one CPI through 2 transmitting antennas and 4 receiving antennas, which led to a range resolution of 5 centimeters, velocity resolution of 0.08 m/s, and azimuth angle resolution of 14.5 degrees. Robot operating system (ROS) was used on the laptop to collect the camera and radar data simultaneously. The obtained camera data were images of the front scene while the radar returned the target's centroid position at 20 fps. Every radar and camera frame also accompanied the frame UTC timestamp, which was used in the decision box to associate and fuse the data as described in Section III-B.

B. Training DNN and LSTM modules

First, labelled data was collected to train the DNN, in order to be able to predict the target position (depth and cross-range) using only camera images. The camera field of view was cut to a maximum distance of 5 meters into small grids of 0.1m×0.1m, each. Images of a person in every cell were collected, along with the ground truth of the actual position

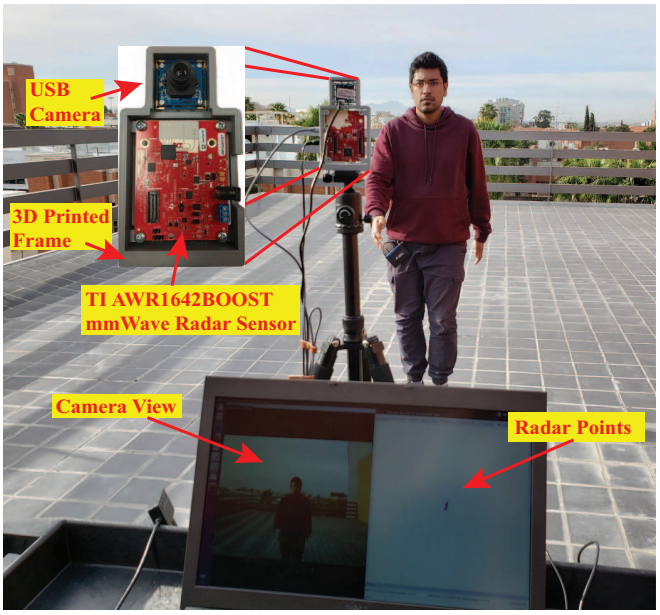

Fig. 3. The experiment setup included an USB RGB camera module and TI AWR1642 boost mmWave radar, mounted together in a 3-D printed frame. The data was collected via USB cables on a laptop running ROS on Linux. A snapshot from the experiment with the camera and radar point clouds on the monitoring screen is also shown.

as training labels. The bounding box dimensions of the target of interest, which would be the input to the DNN, were extracted manually. A 3-hidden layer DNN, with 256, 128 and 64 nodes respectively, was then trained on 70% of the shuffled labelled data, while the remaining 30% was used for validation. The model was trained on Google's Collaboratory Platform, powered by a Tesla K80 GPU [20], using Adam optimizer [21], with a batch-size of 20, for 25 epochs, yielding a training loss of 0.0028 and a validation loss of 0.0030. This trained model was used to make prediction of the target's position, based on the bounding box parameters, for the rest of our experiments.

The predicted position of the target from camera, obtained using the aforementioned DNN, and the position detected by the radar was fed into the decision box to obtain a better range and cross-range position of the target using a high-level sensor fusion. The output of the decision box was fed to a 2 hidden layered LSTM network, with 64 and 32 LSTM nodes, respectively. The LSTM takes in 5 frames of data as input, and predicts the position of the target in the subsequent frame. In order for the LSTM to model the time-position relationship, the LSTM weights were optimized using an Adam-optimizer driven mini-batch gradient descent with the objective to minimize the MSE. The training process was carried out for 20 epochs over 2 sets of 2500 frames of continuous radar data, with an additional 500 frames for validation. The training process yielded a training and validation loss of 1.6×10^{-4}.

C. Fusion Results

1) Improved Localization Accuracy: In order to demonstrate the proposed model's localization accuracy, radar and camera data were collected for two scenarios. First, the target was made to move laterally at a fixed depth of 3 m. The aim of this experiment was to demonstrate radar's smaller variance in detected depth from the ground truth (3 m), when compared to the depth prediction from the DNN, which used camera data. Second, the target was made to move in the direction of depth straight along the line-of-sight (cross-range = 0 m). The aim of this experiment was to show that the camera data would yeild a lower variance from the ground truth, in the cross-range direction, when compared to the radar. The results of the mean, variance and the MSE with respect to the ground truth, for both the scenarios is consolidated in Table I. The corresponding plots from these experiments is presented for visual representation in Fig. 4. We see that the camera provides a considerable amount of improvement in MSE in the cross-range direction, while radar outperforms camera in the detected depth accuracy. Therefore, with these two experiments we validate our methodology of using the radar's depth and camera's cross-range in the fusion stage of the decision box.

TABLE I
VARIATION OF RADAR AND CAMERA DETECTION COMPARED TO GROUND TRUTH

	Scenario-1 (Cross-range fixed at 0 m)			Scenario-2 (Depth fixed at 3 m)		
Sensor	Mean (m)	Variance (m^2)	MSE (m^2)	Mean (m)	Variance (m^2)	MSE (m^2)
Radar	-0.0286	0.0024	0.0032	2.9983	0.0001	0.0001
Camera	-0.0099	0.0004	0.0005	2.9240	0.0029	0.0087

2) Robustness: In order to demonstrate our model's robustness to single sensor failures, we performed an experiment where the target was made to move diagonally with respect to the dual-sensor setup. The target, for the first set of frames, was in the camera FOV, but out of radar's FOV, thereby emulating radar sensor's failure to detect the target. The target then moved into the camera and radar's common FOV, which would

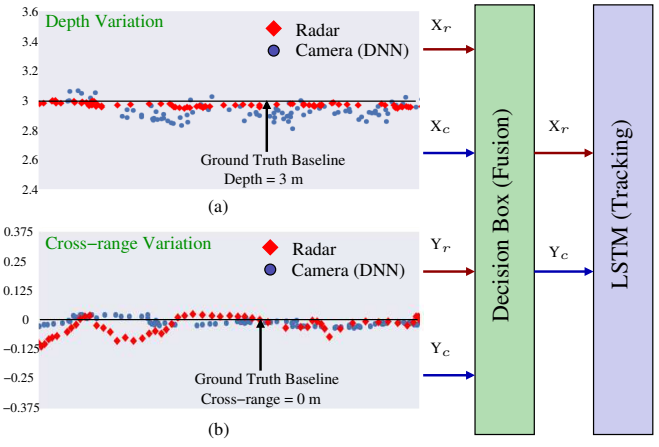

Fig. 4. The depth and cross-range variation from the radar $[X_r, Y_r]$, and DNN prediction using camera data $[X_c, Y_c]$ is shown in (a) and (b), respectively. The decision box provides a fused position $[X_r, Y_c]$ on account of their lower MSE and variance in their respective domains

be the sensor-fusion region before the target finally moving out of the camera FOV to emulate camera failing to make a detection. The data from both the sensors was subjected to the decision box followed by the LSTM network for track update and generation. From Fig. 5, we see that our system is able to continuously track the target, even when one of the sensors fail to make a detection.

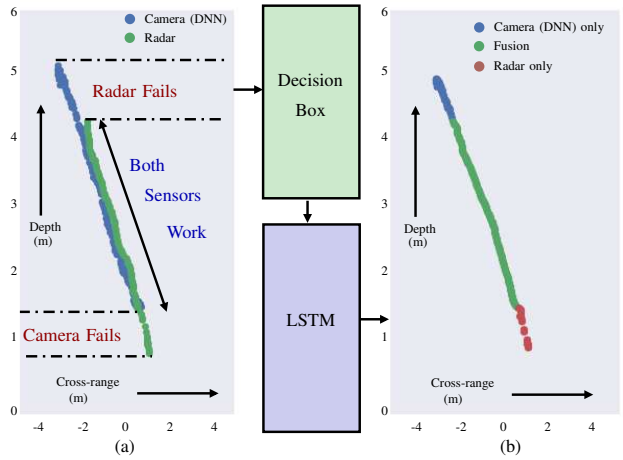

Fig. 5. (a) The target trajectory mapped individually from radar and camera data is shown. We notice the discontinuity in the detected positions from either sensors, as mentioned in the experiment description. (b) The final trajectory of the target as output by the LSTM. The decision box is able to detect missed readings from the failing sensor and passes on the available position from the other sensor to the LSTM. For the region where both radar and camera data are available, the fused position of the target is sent to LSTM.

V. Conclusions and Future Work

In this paper, a novel DNN-LSTM based approach to track a target using a high-level radar and camera data fusion is presented. The target detected by camera is first identified and a bounding box is generated. A trained DNN is then used to predict the target's position using the bounding box parameters. The radar data is collected parallely following the radar signal processing chain. The asynchronous camera and radar detected positions are then associated and fused in the decision box using the frame UTC timestamps. An LSTM module is used to continuously generate the target trajectory based on the previous 5 frames of detection. From our experiments, robustness of the proposed design was demonstrated in single sensor failure mode. Our proposed fusion approach was further validated experimentally by measuring the MSE and variance with respect to ground truth. The fusion approach outperforms the detection results from individual sensors. Future work include extending this approach to multiple target tracking and parallel object classification and behavior detection.

References

[1] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Autonomous Robots*, vol. 26, no. 2-3, pp. 123–139, 2009.

[2] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[3] V. N. Dobrokhodov, I. I. Kaminer, K. D. Jones, and R. Ghabcheloo, "Vision-based tracking and motion estimation for moving targets using small uavs," in *2006 American Control Conference*, pp. 6–pp, IEEE, 2006.

[4] R. Reulke, S. Bauer, T. Doring, and F. Meysel, "Traffic surveillance using multi-camera detection and multi-target tracking," in *Image and Vision Computing New Zealand*, pp. 175–180, 2007.

[5] B. Steux, C. Laurgeau, L. Salesse, and D. Wautier, "Fade: A vehicle detection and tracking system featuring monocular color vision and radar data fusion," in *Intelligent Vehicle Symposium, 2002. IEEE*, vol. 2, pp. 632–639, IEEE, 2002.

[6] G. Alessandretti, A. Broggi, and P. Cerri, "Vehicle and guard rail detection using radar and vision data fusion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 95–105, 2007.

[7] E. Richter, R. Schubert, and G. Wanielik, "Radar and vision based data fusion - advanced filtering techniques for a multi object vehicle tracking system," in *2008 IEEE Intelligent Vehicles Symposium*, pp. 120–125, June 2008.

[8] R. O. Chavez-Garcia and O. Aycard, "Multiple sensor fusion and classification for moving object detection and tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 525–534, 2016.

[9] F. A. Alencar, L. A. Rosero, C. Massera Filho, F. S. Osório, and D. F. Wolf, "Fast metric tracking by detection system: radar blob and camera fusion," in *2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR)*, pp. 120–125, IEEE, 2015.

[10] X. Wang, L. Xu, H. Sun, J. Xin, and N. Zheng, "On-road vehicle detection and tracking using mmw radar and monovision fusion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 2075–2084, 2016.

[11] K. Ramasubramanian and K. Ramaiah, "Moving from legacy 24 ghz to state-of-the-art 77-ghz radar," *ATZelektronik worldwide*, vol. 13, no. 3, pp. 46–49, 2018.

[12] M. A. Richards, *Fundamentals of radar signal processing*. Tata McGraw-Hill Education, 2005.

[13] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: The algorithm gdbscan and its applications," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 169–194, 1998.

[14] K.-S. Oh and K. Jung, "Gpu implementation of neural networks," *Pattern Recognition*, vol. 37, no. 6, pp. 1311–1314, 2004.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[16] S. Haykin, *Neural networks*, vol. 2. Prentice hall New York, 1994.

[17] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: perceptron, madaline, and backpropagation," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1415–1442, 1990.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] T. Instruments, *AWR1642 Evaluation Module (AWR1642BOOST) Single-Chip mmWave Sensing Solution*, Apr. 2018. http://www.ti.com/lit/ug/swru508b/swru508b.pdf.

[20] Google, *Google Colaboratory: Tensorflow with GPU*, 2018. https://colab.research.google.com/notebooks/gpu.ipynb.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.