https://doi.org/10.2352/ISSN.2470-1173.2018.17.AVM-258
© 2018, Society for Imaging Science and Technology

Camera Radar Fusion for Increased Reliability in ADAS Applications

Ziguo Zhong*, Stanley Liu†, Manu Mathew†, and Aish Dubey†

* Perception & Analytics Lab, Silicon Development, Embedded Processing
z-zhong@ti.com
†Automotive Processor, Embedded Processing
{stanley_liu, manu.mathew, aish}@ti.com

***Abstract* – Driven by the mandated adoption of advanced safety features enforced by governments around the global as well as strong demands for upgraded safety and convenience experience from the consumer side, the automotive industry is going through an intensified arms race of equipping vehicles with more sensors and boosted computation capacity. Among various sensors, camera and radar stand out as a popular combination offering complementary capabilities. As a result, camera radar fusion (or CRF in short) has been regarded as one of the key technology trends for future advanced driving assistant system (ADAS). This paper reports a camera radar fusion system developed at TI, which is powered by a broad set of TI silicon products, including CMOS radar, TDA SoC processor, FPD-Link II/III SerDes, PMIC, and so forth. The system is developed to not only showcase algorithmic benefits of fusion, but also the competitiveness of TI solutions as a whole in terms of coverage of capabilities, balance between performance and energy efficiency, and rich supports from the associated HW and SW ecosystem.**

***Keywords*: ADAS, radar, camera, sensor fusion, reliability.**

I. Introduction

Advanced driving assistant system (ADAS) and autonomous driving have been buzz words in recent years because of strong driven forces from the policy makers [1, 2, 3] as well as booming consumer demands for safer and smarter vehicles. In many applications (e.g., collision warning and avoidance, adaptive cruise control, lane keeping, autonomous parking, etc.), accurately perceiving (including sensing, understanding, and modeling) the surrounding world in a real-time manner serves as the precondition for follow-up decision making and actuation. However, such a task remains largely open and challenging due to its extreme complexity in highly dynamic environments, but very limited resources (for sensing and processing) on-board each individual vehicle.

Motivated by the need of high reliability, multisensorial data fusion has been considered as one of the pivotal means towards real life safety assurance, because no single technology is able to cover all requirements [4, 5]. In fact, a recent study reports that only systems applying sensor fusion achieved 100% best Euro NCAP Rating for safety [6]. In this paper, we report recent progress made and initial results obtained at TI on the topic of camera radar fusion (CRF in short), and demonstrate a prototype system built with a portfolio of TI chips centered with TDA SoC processor and CMOS radar.

II. Fusion Overview

Radar and camera feature complimentary characteristics that naturally lead to the desire for sensor fusion. As illustrated in Figure 1, the fusion of camera and radar can address many inherent limitations of each sensor if used alone. Benefits of integration come from the fact that data from separate sensors that excel on different tasks can be matched, verified, and fused to achieve greatly improved accuracy and reliability.

Perception	Camera	Radar	Fusion
Distance	Fair	Excellent	Excellent
Angle	Excellent	Fair	Excellent
Velocity (Radial)	Limited	Excellent	Excellent
Velocity (Lateral)	Fair	Limited	Excellent
Boundary	Excellent	Limited	Excellent
Obstacle	Fair	Excellent	Excellent
Classification	Excellent	Fair	Excellent
Weather/Lighting/Dirt	Limited	Excellent	Excellent

Excellent
Fair
Limited

Figure 1: Advantage of Fusion in Different Tasks

II. Prototype Fusion System

Before unfolding the algorithm details, this section provides an overview about the fusion system established at TI in order to cover necessary backgrounds on this project

The first prototype system was built to evaluate object-level fusion for key ADAS functions including object detection, recognition, and tracking. The hardware platform established was named "xCAM" that is equipped with a TDA2x SoC, a COMS radar module (using TC2, a test chip containing the full signal chain from RF to ADC), and a third-party camera sensor. On the software side, device drivers and signal processing algorithms were implemented on Vision SDK v.02.06.

Figure 2 provides an overview of system by annotating on a snapshot of its real-time HDMI output. On the left side, radar and camera results obtained from radar processing pipeline and computer vision pipeline are given. Specifically, the top two views are radar detection points and tracking results; and the bottom two views are vision outputs including sparse optical flow and a neural network for vision-based object detection.

In the right part of Figure 2, fusion results obtained by integrating outputs from all left views are presented. Fusion offers a rich list of information regarding sensed objects,

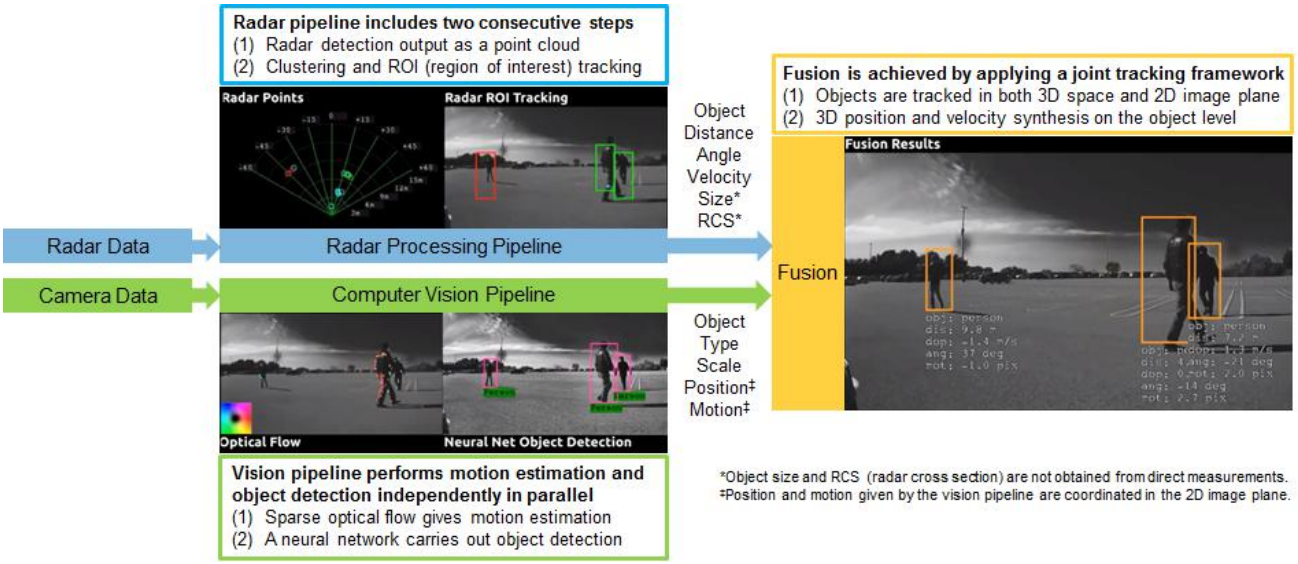

Figure 2: Camera Radar Fusion Concept and Prototype System Outputs

including its type, distance, velocity, angle, and angular velocity. Fusion results feature improved robustness since it tolerates miss detection of any one sensor to a certain degree.

IV. ALGORITHMS

This section details design of camera radar fusion system. Figure 3 (on next page) illustrates the data flow and algorithms developed to achieve the goal of object-level fusion, where main processing blocks include:

- "Radar Signal Processing" block, which works on radar data and outputs clustered points and target ROIs (i.e., region of interest) with associated information such as radial distance, angle, radial velocity, and so forth;
- "Vision Motion Estimation" block, which outputs sparse optical flow vectors indicating motion on the image plane;
- "Vision Recognition" (Image Processing + Recognition), which conducts object classification with reasoning trees and outputs typed (e.g., person) bounding boxes;
- "*CRF Fusion*" block, which performs velocity synthesis and target tracking both in the 3D physical space and on the 2D image plane (more details are provided later).

In the following, key algorithms of each building block are explained with an emphasis on radar and fusion blocks.

Radar Signal Processing

Figure 4 provides a more detailed list on algorithms and methods used in radar signal process. For each frame of radar ADC data, it is processed in four steps sequentially.

- "Chirp Processing" carries out necessary pre-processing on the ADC data (e.g., zero padding and bit extension) and then range FFT with transposed result write back.
- "Frame & Cross-Antenna Processing" performs two asks: (i) Doppler FFT and peak detection on the range-Doppler plane using CFAR (i.e., constant false alarm rate) [7], and then (ii) angle estimation (i.e., DOA in Figure 4 and 5) for selected peaks using the direct beam forming method. Besides, range-gating was applied as "Filter A" to remove noisy range bins closer than 50 cm.
- "Clustering and ROI Generation" clusters the detection point cloud into a dynamic number of ROIs. The algorithm "NCIMD" was developed based on normalized cut theory [8] and served as the initial clustering method.
- Finally, "ROI-based Object Tracking & Classification" applies temporal and spatial filters on accumulated ROI records to reduce noise (due to both false alarm and miss detection) and alleviate "multi-path ghosts".

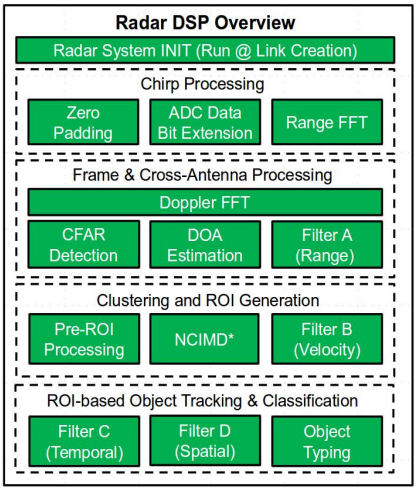

Figure 4: Radar Processing Stages and Algorithms

Vision Motion Estimation

Compared with radar, vision (or camera) in general features much higher resolution and better accuracy on lateral motion sensing for objects located in near to middle range, given sufficient lighting and surface texture. Thus, sparse optical

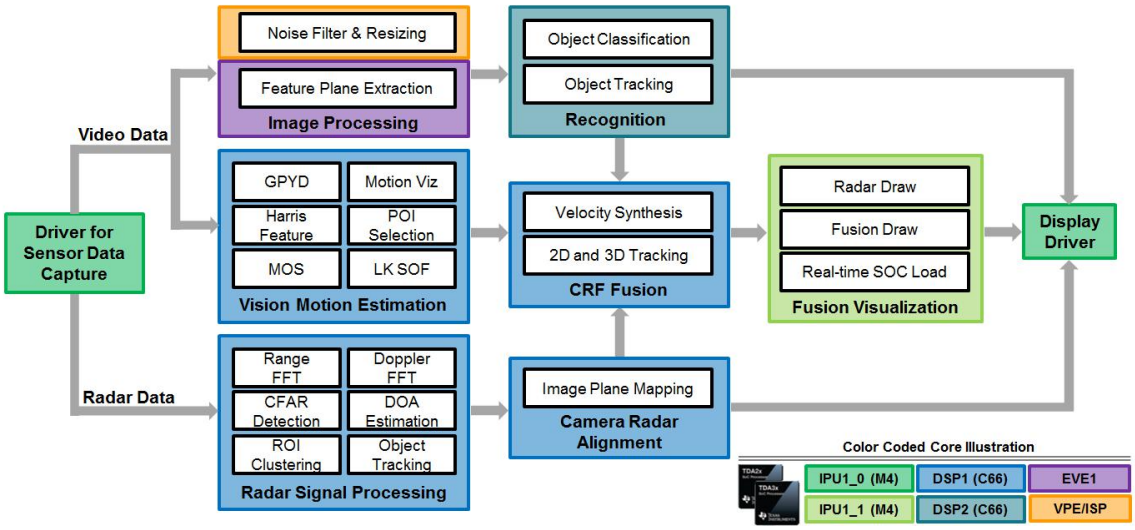

Figure 3: Data Flow and Algorithm Map (notice that blocks are color coded according to their running cores)

flow was implemented and partially optimized on C66 DSP to fulfill the task of vision motion estimation on the image plane. As depicted in Figure 3, a classical processing chain, which is composed of Gaussian image pyramid generation (i.e., GPYD), Harris corner feature extraction, non-maximum suppression (i.e., POI selection), and Lucas-Kanade motion estimation (translation-only model) was deployed. Notice that vision motion estimation is computationally heavy and could be the bottleneck of the system. For example, a 30-FPS motion estimation engine outputting around 500 flow vectors would consume more than 50% cycles of a 500MHz C66 DSP.

Vision Recognition

Another clear advantage of vision compared with radar is its much richer information embedded in the data, which is widely considered more suitable for the task of object classification.

In the reported fusion system, a classical vision recognition pipeline, which applies the method of patch scanning with manually designed features and cascaded reasoning trees, was adopted for an initial evaluation. It was implemented as two processing blocks as show in Figure 3.

- "Image Processing" handles pixel-level noise filtering, image resizing (to construct a 17-layer image pyramid), and oriented gradient feature extraction similar to HoG [9]. Its output is a 10-channel descriptor composed of Y, Cb, Cr, gradient magnitude, as well as six oriented gradient magnitudes at 0, 30, 60, 90, 120, and 150 degrees.
- "Recognition" performs classification by feeding features to a reasoning forest made of 1280 decision trees, and evaluates if the accumulated response is greater than a threshold. In addition, Kalman tracking is carried out on recognized objects across consecutive frames.

Further details on vision recognition are omitted here due to the space constraint as well as our decision of completely moving to CNN-based solutions in future systems.

CRF Fusion

Object-level fusion expects to deliver improved position and velocity estimations on in-field targets by integrating outputs from radar and vision (motion and recognition independently). The major difficulty comes from the fact that sensing results obtained from radar and camera lie in different spaces: radar ROIs are defined in 3D physical space while both motion vectors and object bounding boxes live on 2D image plane. To accommodate such a practical constraint, we proposed a fusion framework that tracks sensed objects simultaneously in the 3D space and on the 2D image plane as illustrated by Figure 5.

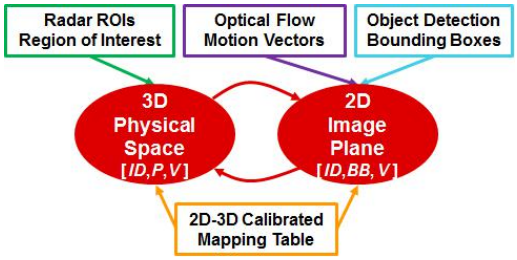

Figure 5 Proposed Object-level Fusion Framework

For each object, it is modeled in the 3D space using the classical six-parameter kinetic model (i.e., 3D position and velocity) and on the 2D image plane as a bounding box with 2D velocities. The fusion framework makes sure that these two models corresponding to each individual target match each other all the time. This is achieved by coupled model updating. For example, the coming of new radar ROIs (depicted as the green arrow in Figure 5) triggers updating on object models in the 3D space; while 2D models of these objects get updated immediately according to their latest states in the 3D space based on a geometric mapping table calibrated beforehand. Similar procedures are carried out upon inputs from the vision side as illustrated by directional red arrows in Figure 5.

For model propagation and updating, an uncertainty-driven mechanism similar to Kalman filter was applied to make a balanced use of sensing results obtained with diverse qualities. Furthermore, a carefully designed model aging scheme as well as time-stamped source data ensure that expired sensing results or out-of-order messages arriving at the fusion block would not lead to erroneous model propagation. Despite their importance, details are omitted here due to space constraint.

3D Velocity Synthesis

A major challenge for the reported system is to recover the true 3D velocity associated with each target in the field. It is difficult because each sensor alone only provides partial information and thus fusion becomes a must. Figure 6 below is used to model the problem, where $P(t) = [X, Y, Z]$ and $V(t) = [V_x, V_y, V_z]$ are 3D position and velocity vectors to estimate.

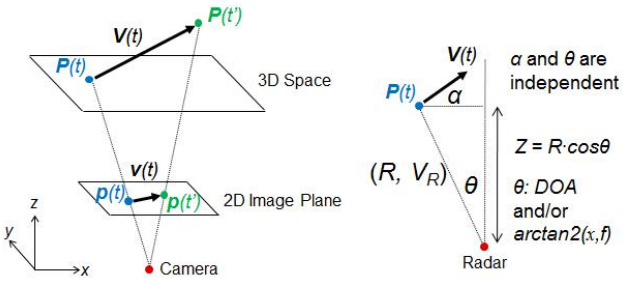

Figure 6 Velocity Synthesis with Camera Radar Fusion

Camera outputs $p(t)$ and $v(t)$ in the form of

$$p(t) = [x, y, f], v(t) = [v_x, v_y, 0]$$

where f is the focal length and $[v_x, v_y]$ is the flow vector. On the other hand, basic geometric rule tells that

$$p = P \cdot f \cdot Z^{-1} \Rightarrow v = \frac{\partial p}{\partial t} = (f \cdot Z^{-2})(Z \cdot V - V_z \cdot P)$$

This equation can be unfolded and transformed into

$$V_x = (v_x \cdot Z + V_z \cdot x) \cdot f^{-1}$$
$$V_y = (v_y \cdot Z + V_z \cdot y) \cdot f^{-1}$$

where x, y, f, v_x, v_y are known from camera and Z is estimated from radar outputs as shown in the right of Figure 6. Finally, by applying radar sensed velocity with equations below

$$V_z = |V(t)| \cdot \sin(\alpha)$$
$$V_R = |V(t)| \cdot \cos(\pi/2 + \theta - \alpha) = |V(t)| \cdot \sin(\alpha - \theta)$$
$$|V(t)|^2 = V_x^2 + V_y^2 + V_z^2$$

$V(t) = [V_x, V_y, V_z]$ can be solved to give 3D velocity estimation.

V. Computation Profile

Figure 7 shows typical loads on different cores of a TDA2x SoC running the reported fusion system. Please be advised that code optimization was limited due to time constraints. On the SoC, IPU1 (ARM Cortex-M4) handles drivers, IPC among cores, and visualization. Radar algorithms were implemented on DSP1. TC2 ran at 6 FPS, a low frame rate due to hardware constraints of the test chip. At this rate, it consumed about 10% of DSP1 (500 MHz C66). For vision, 30-FPS object detection used EVE1 (500 MHz) for feature extraction and DSP2 (500 MHz C66) for recognition. 30-FPS optical flow was calculated by DSP1 (45~50% DSP load); fusion algorithms were also deployed on DSP1 and consumed about 5% DSP load.

Figure 7 Real-time Loads of Cores on TI TDA SoCs

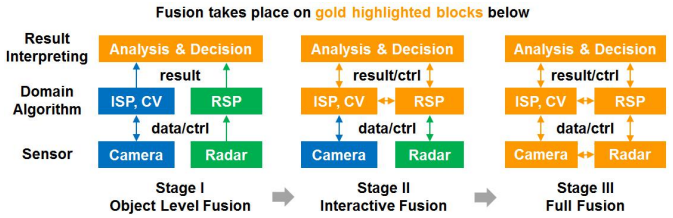

Figure 8 Long-Term Fusion Goals

VI. Limitations and Future Work

As our first attempt on sensor fusion for ADAS, the reported design exposed a number of limitations, among which the most important is the lack of data-driven quantitative evaluation. We are currently working towards the solving of this problem by establishing a many-sensor data acquisition system.

Moving forward, we expect more capable fusion algorithms as well as integrating more sensors of different types. Figure 8 shows potential direction of future work, in which fusion gets pushed from the object-level on the top (i.e., Stage I) to domain algorithms with raw data fusion (i.e., Stage II), and finally throughout the stack (i.e., Stage III) with optimized sensor operation. An example can be that radar and camera operation profiles vary dynamically according to real-time needs (e.g., see further) and constraints (e.g., low light).

References

[1] NHTSA, "U.S. Department of Transportation Releases Policy on Automated Vehicle Development", NHTSA.org, May 2013.
[2] ERTRAC, "Automated Driving Roadmap", v. 5.0, July, 2015.
[3] Ministry of Transport of P. R. China, "National Standard of PRC GB7258-2016: Safety Specifications for Power-Driven Vehicles Operating on Roads", draft version, MOC.gov.cn, April 2016.
[4] J. Laneurit, et al., "Multisensorial Data Fusion for Global Vehicle and Obstacles Absolute Positioning", *IEEE IVS, 2003*.
[5] N. Shima, et al., "Fusion Sensor for Driving Assistance System", FUJITSU TECH. Journal. No.17, 2001.
[6] F. Melzer, "Towards Autonomous Driving", Autoliv CMD'15.
[7] L. Scharf, "*Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*", Addison Wesley, NY.
[8] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation", IEEE Transactions on PAMI, 22(8), August 2000.
[9] N. Dala and B. Triggs, "Histograms of Oriented Gradients for Human Detection", CVPR 2015.

Free access to this paper is brought to you with the generous support of ON Semiconductor.
All research funding for this paper is referenced in the text; unless noted therein, no research funding was provided by ON.