A Diverse Low Cost High Performance Platform for Advanced Driver Assistance System (ADAS) Applications

Prashanth Viswanath
Texas Instruments India Pvt Ltd
Bangalore, India
p-viswanath@ti.com

Kedar Chitnis kedarc@ti.com
Pramod Swami pramods@ti.com
Mihir Mody mihir@ti.com
Sujith Shivalingappa sujith.s@ti.com
Soyeb Nagori soyeb@ti.com
Manu Mathew mathew.manu@ti.com
Kumar Desappan kumar.desappan@ti.com
Shyam Jagannathan shyam.jagannathan@ti.com
Deepak Poddar deepak.poddar@ti.com
Anshu Jain anshu.jain@ti.com
Hrushikesh Garud hrushikesh.garud@ti.com
Vikram Appia vikram.appia@ti.com
Mayank Mangla mayank@ti.com
Shashank Dabral shashank@ti.com

Abstract

Advanced driver assistance systems (ADAS) are becoming more and more popular. Lot of the ADAS applications such as Lane departure warning (LDW), Forward Collision Warning (FCW), Automatic Cruise Control (ACC), Auto Emergency Braking (AEB), Surround View (SV) that were present only in high-end cars in the past have trickled down to the low and mid end vehicles. Lot of these applications are also mandated by safety authorities such as EUNCAP and NHTSA. In order to make these applications affordable in the low and mid end vehicles, it is important to have a cost effective, yet high performance and low power solution. Texas Instruments (TI's) TDA3x is an ideal platform which addresses these needs. This paper illustrates mapping of different algorithms such as SV, LDW, Object detection (OD), Structure From Motion (SFM) and Camera-Monitor Systems (CMS) to the TDA3x device, thereby demonstrating its compute capabilities. We also share the performance for these embedded vision applications, showing that TDA3x is an excellent high performance device for ADAS applications.

1. Introduction

Advanced driver assistance systems (ADAS) is becoming the need of the hour, as mobility has come to be a basic need in today's life. Approximately 1.24 million people died in road accidents around the globe in 2010 [2]. Pedestrians, cyclists and motorcyclists comprise half of the road traffic deaths and motor vehicle crashes are ranked number nine among top ten leading causes of death in the world [5]. These statistics are mandating the car manufacturers to ensure higher safety standards in their cars. The European New Car Assessment Program (EUNCAP) and National Highway Traffic Safety Administration (NHTSA) provide safety ratings to new cars based on the safety systems that are in place. EUNCAP [6] provides better star rating for cars equipped with Auto Emergency Braking (AEB), Forward Collision Avoidance (FCA), Lane Keep Assist (LKA) etc. which ensures higher safety for on-road vehicles, pedestrians, cyclists and motorcyclists.

ADAS can be based upon various sensor systems such as radar, camera, lidar and ultrasound [7]. Additionally, they can integrate and use external information sources such as global positioning systems, car data networks and vehicle-to-vehicle or vehicle-to-infrastructure communication systems to efficiently and accurately achieve desired goals. While different sensor modalities have varying performance based on different environmental conditions and applications, camera sensors are emerging as a key differentiator by car manufacturers. Camera based ADAS use various computer vision (CV) technologies to perform real-time driving situation analysis and provide warning to the driver. The advantages of camera based ADAS include reliability and robustness under difficult real life scenarios, and ability to support multiple varied applications such as traffic sign recognition (TSR), traffic light detection, lane and obstacle detection.

To enable different safety aspects of ADAS, camera-based systems are deployed in front and back viewing and surround viewing [8]. The front camera systems are used for applications such as AEB and FCW. The rear view and surround view systems are used for park assist and cross traffic alert applications. Front camera systems can use mono or stereo camera setup. Stereo camera is useful to obtain 3D information by generating disparity. However, stereo camera systems are more expensive compared to mono camera systems. Structure from Motion (SFM) technology [17] [11], which enables obtaining depth from a single camera that is moving, is being widely researched for its applicability in ADAS. Surround view systems use multiple cameras (4 to 6) placed around the car. The feed from multiple cameras are re-mapped and stitched to provide a 360° view to the driver. Also, analytics are performed on these images to alert the driver. Recently, Camera Mirror Systems (CMS) are increasingly replacing mirrors in mid/high end cars. In CMS systems the side and rear view mirrors are replaced by cameras and the camera feed is displayed to the driver via display panels (typically OLED display panels). Cameras with wide angle field of view avoids blind spots for the driver. Sophisticated features like wide dynamic range (WDR) [15] and noise filter allow the system to be used in variety of lighting condition including low light, high glare scenarios. Due to the low surface area of the camera lens vs a conventional mirror, a CMS system is less susceptible to the effects of dust, rain. CMS have added advantage of reduced wind drag and thus aiding fuel efficiency. Also, the CMS opens the possibility of running vision analytics on them [12]. Figure 1 shows the flow for different ADAS applications.

In order to fully utilize the capability of camera based systems for multiple applications, it is therefore an absolute necessity to have a high performance, low power and low cost embedded processor, which is capable of analyzing the data from multiple cameras in real time. In order to solve this problem, Texas Instruments has developed a family of System-on-Chip (SOC) solutions that integrate heterogeneous compute architectures like General Purpose Processor (GPP), Digital Signal Processor (DSP), Single Instruction Multiple Data (SIMD) processor and Hardware Accelerators (HWA) to satisfy the compute requirements while meeting the area and power specifications. The rest of the paper is organized as follows: Section 2 provides an introduction to a high performance, low area and power, third generation of SoC solution from Texas Instruments called Texas Instruments Driver Assist 3x (TDA3x), Section 3 illustrates different applications such as LDW, OD, SFM, SV, CMS and their mapping to the TDA3x platform, Section

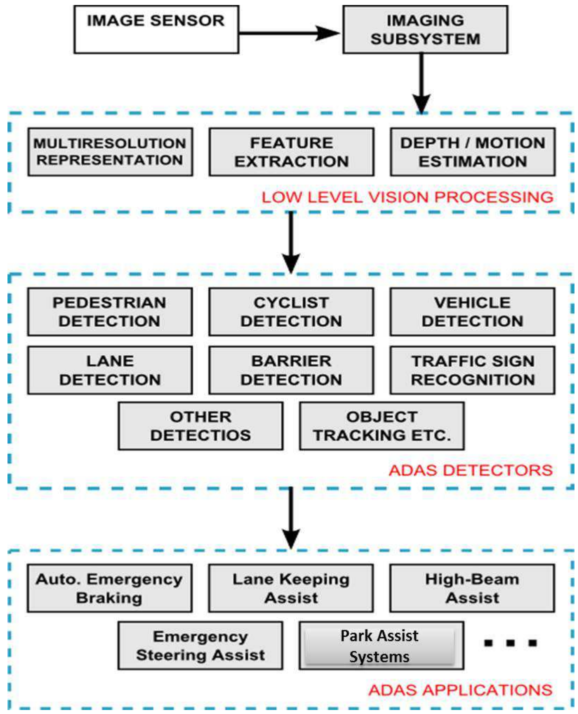

Figure 1. Flow chart of ADAS applications.

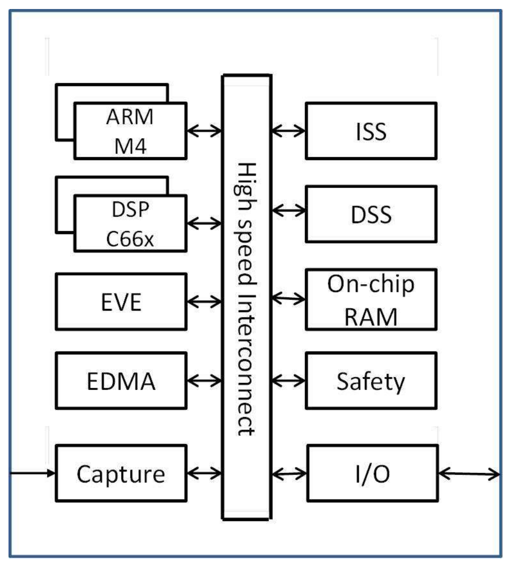

Figure 2. TDA3x SoC Block Diagram.

4 shows the results of our implementation and the performance data and Section 5 provides conclusion.

2. TDA3x Introduction

The TDA3x SoC [4] has a heterogeneous and scalable architecture that includes a dual core of ARM Cortex-M4, dual core of C66x DSP and single core of Embedded Vision Engine (EVE) for vector processing as shown in Fig-

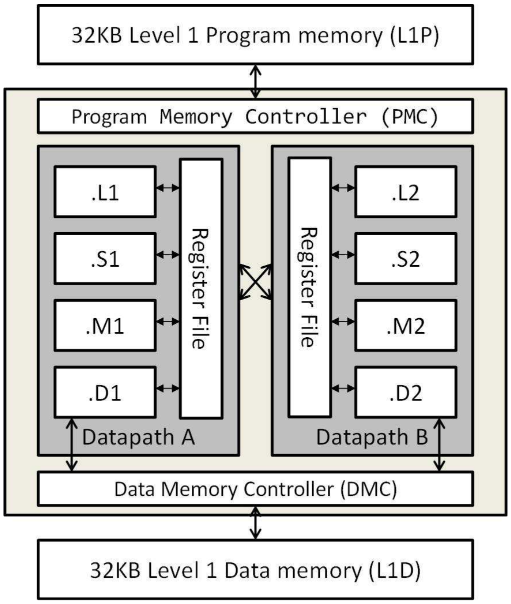

Figure 3. C66x Processor Block Diagram.

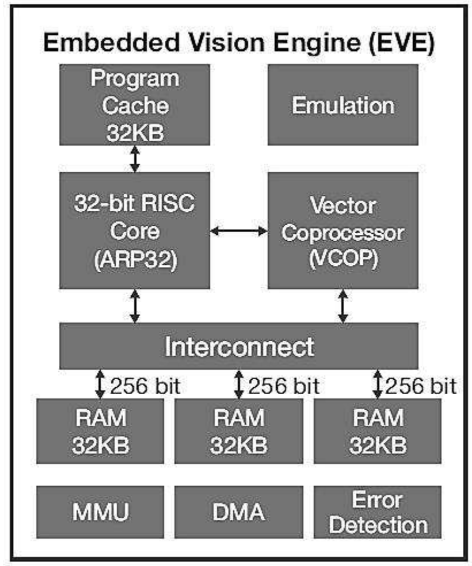

Figure 4. EVE Processor Block Diagram.

ure 2. It integrates hardware for camera capture, Image signal processor (ISP) and Display sub-system resulting in better video quality at lower power. It also contains large on-chip RAM, rich set of input/output peripherals for connectivity, and safety mechanism for automotive market and offers lower system cost. There are three types of programmable cores in the TDA3x SoC.

2.1. General Purpose Processor (GPP)

The dual core or ARM Cortex-M4 CPU running at 212.8 MHz serves as the general purpose processor in the TDA3x [1]. The M4 cores deliver efficient control and processing camera stream.

2.2. Digital Signal Processor (DSP)

TDA3x contains a dual core of C66x DSP. The C66x DSP [3] is a floating point VLIW architecture with 8 functional units (2 multipliers and 6 arithmetic units) that operate in parallel as shown in Figure 3. It comprises of 64 general purpose 32-bit registers shared by all eight functional units. There are four arithmetic units .L1/.L2, .S1/.S2, two multiplier units for .M1/.M2 and two data load and store units .D1/.D2. Each C66x DSP core has configurable 32KB of L1 data cache, 32KB of L1 instruction cache and 288KB of unified L2 data/instruction memory.

2.3. Embedded Vision Engine (EVE)

TDA3x contains a single core of Embedded Vision Engine (EVE). EVE is a fully programmable accelerator specifically to enable the processing, latency and reliability needs found in computer vision applications. The EVE includes one 32-bit Application-Specific RISC Processor (ARP32) and one 512-bit Vector Coprocessor (VCOP) with built-in mechanims and unique vision-specialized instructions for concurrent, low overhead processing. The VCOP is a dual 8-way SIMD engine with built-in loop control and address generation. It has certain special properties such as transpose store, de-interleave load, interleaving store and so on. The VCOP also has specialized pipelines for accelerating table look-up and histograms [13]. Figure 4 shows the block diagram of EVE processor.

3. Applications and System Partitioning

3.1. System Partitioning

A computer vision application can be roughly categorized into three types of processing: Low-level, Mid-level and High-level processing. The low-level processing functions include pixel processing operations, where the main focus is to extract key properties such as edges, corners and forming robust features. The mid-level processing include feature detection, analysis, matching and tracking. The high-level processing is the stage where heuristics are applied to make meaningful decisions by using data generated by low and mid-level processing. The EVE architecture is an excellent match for low-level and mid-level vision processing functions due its number crunching capability. C66x DSP with program and data caches enables mix of control as well data processing capabilities and suits well for mid and high-level vision processing functions. High level OS (or RTOS) runs on ARM as main controller and does I/O with real world.

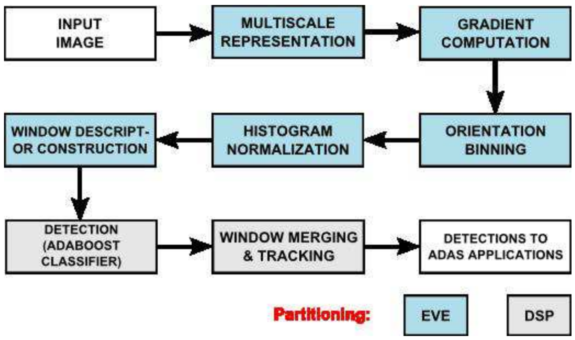

Figure 5. Object Detection algorithm Partitioning.

```
Program: Gradient Magnitude

Z = 0;
for (I1 = 0; I1 < height; I1++) {
 for (I2 = 0; I2 < (width/16); I2++) {
   // Separate Address generation hardware
   Addr1 = I1*pitch*ELEMSZ + I2*VECTORSZ*2;
   Addr2 = I1*width*ELEMSZ*2 + I2*VECTORSZ*2*2;
   Addr3 = I1*width*ELEMSZ*2 + I2*VECTORSZ*2*2;
   // Data Load for dual SIMD in VCOP
   (VinT1,VinT2) = (pIn+1)[Addr1].deinterleave();
   (VinL1,VinL2) = (pIn+pitch)[Addr1].deinterleave();
   (VinR1,VinR2) = (pIn+pitch+2)[Addr1].deinterleave();
   (VinB1,VinB2) = (pIn+2*pitch+1)[Addr1].deinterleave();
   // Vector Computation
   VgX_1 = VinR1 - VinL1;
   VgY_1 = VinB1 - VinT1;
   VgX_2 = VinR2 - VinL2;
   VgY_2 = VinB2 - VinT2;
   Vmag1 = abs(VgX_1);
   Vmag2 = abs(VgX_2);
   Vmag1 += abs(VgY_1-Z);
   Vmag2 += abs(VgY_2-Z);
   // Data Store from dual SMID in VCOP
   pGradX[Addr2].interleave()   = (VgX_1,VgX_2);
   pGradY[Addr2].interleave()   = (VgY_1,VgY_2);
   pMag[Addr3].interleave()    = (Vmag1,Vmag2);
  }
 }
```

Figure 6. Gradient Magnitude Computation on EVE.

3.2. Object Detection and Traffic Sign Recognition

The object detection algorithm consists of low, mid and high level processing functions and are mapped across the EVE and DSP cores as shown in Figure 5. As EVE is suitable for low and mid-level processing, stages such as Gradient Computation, Orientation Binning, Histogram Equalization etc are mapped to EVE while the classification stage is mapped to the C66x DSP.

3.2.1 Gradient Computation on EVE

Gradient computation is one of the most commonly used operation in feature detection stage of various algorithms such as Histogram of Gradients (HoG) [9] and ORB [20]. The gradient is calculated by finding absolute difference of pixel in horizontal and vertical direction and adding both providing magnitude of gradient. Figure 6 shows optimized code written in kernel-C (C-like language for EVE) for gradient magnitude computation. Each VCOP computation instruction/line in Figure 6 operates on 8 elements. VCOP has two 256 bit functional units each and can operate on 8 data elements in parallel. Two instruction/lines can be executed in a cycle. Address computations are performed by dedicated units so it can happen in parallel with core computation. Loop counters are managed by nested loop controller of VCOP and does not add any overheads. Data load and store instruction can be hidden by compute cycles. The loop in Figure 6 takes just 4 cycles per iteration (generating output for 16 pixel locations in parallel), resulting in 64 times faster performance.

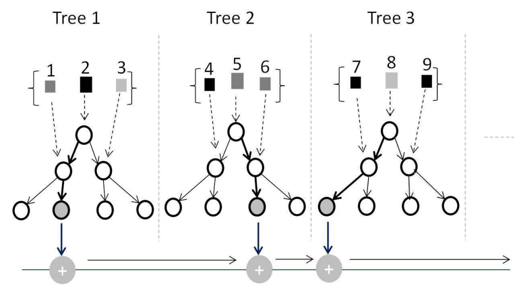

Figure 7. Adaboost Classifier Diagram.

3.2.2 Adaboost Classification on C66x DSP

Adaboost classifier uses a set of simple decision trees whose individual classification accuracy is slightly more than 50% [18]. By combining the response of several such simple decision trees, a strong classifier can be constructed without the need for sophisticated classifier engine as shown in Figure 7. Each individual tree comprises of 3 nodes and 4 leaves. Nodes are the locations where an input value is compared against a predetermined threshold. Depending on the comparison result, the tree is traversed left or right till it reaches one of the four possible leaf values. The tree structure, the threshold values, the leaf values and even the offsets from which input has to be read is predetermined during the training stage. The final leaf value or responses of each tree is accumulated. The accumulated response is compared against a cascade threshold which finally classifies the object. This algorithm is data bound with 3 thresholds, 3 offsets, 3 inputs and 4 leaf values read for each tree. Assuming that all data is 16 bit, accessing the inputs, offsets, thresholds, leaf values one at a time will be inefficient on C66x DSP with 64 bit data paths. As the C66x DSP supports SIMD processing for fixed point data, the first step is to load and process four 16 bit data in a single cycle. TI provides intrisic instructions which the can be used to perform SIMD operations. As input data tested at each node is fetched from a sparse offset in the memory, software can perform SIMD loads of the predetermined offsets, thresholds and leaf values stored contiguously in the memory [16].

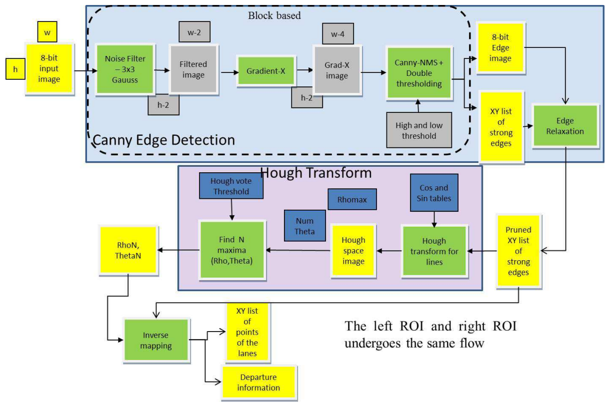

Figure 8. LDW Algorithm Block Diagram.

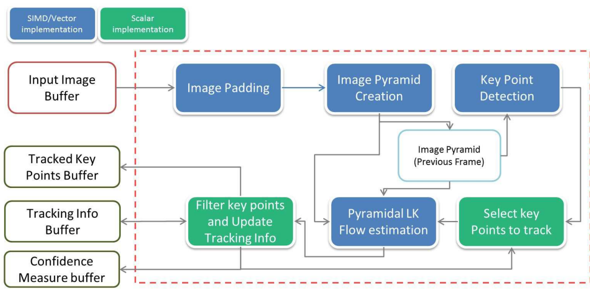

Figure 9. Sparse Optical Flow Block Diagram.

3.3. Lane Departure Warning

Lane Departure Warning (LDW) algorithm consists of low and mid level processing functions. Due to the presence of a single EVE on the TDA3x, the LDW is mapped to C66x DSP. Also, since the LDW algorithm uses Canny edge detection which includes edge relaxation stage which cannot be made block based and due to the limited internal memory of EVE, this algorithm is mapped to the C66x DSP for simplicity of design. The block diagram of LDW is as shown in Figure 8. The LDW algorithm is purely image based and uses simple processing functions such as Canny Edge Detection and Hough Transform for Lines to detect the lanes. Algorithmic enhancements and simplifications such as Intelligent ROI definition, Computation of Horizontal Gradient Only, Detection of Inner/Outer Edge and Curve Detection using Hough Transform for Lines are performed. More details of the algorithm implementation can be found in [19]. Dataflow optimization techniques such as use of Direct Memory Access (DMA) to transfer smaller blocks of data in to L1D, ping-pong data processing are employed to reduce the DDR bandwidth.

3.4. Structure From Motion (SFM)

Structure from Motion (SFM) is a key algorithm which enables computation of depth using a single camera which is moving [17] [11]. The key components of SFM are

Figure 10. 2D Surround View System Output.

Figure 11. 3D Surround View System Output.

Sparse Optical Flow (SOF) and Triangulation. Optical flow estimates pixel between two temporally ordered images. Lucas Kanade (LK) [14] based SOF is widely used for these purposes. Figure 9 shows the various modules involved in SOF. The SOF algorithm is implemented on EVE engine of TDA3x. Although SOF operates on sparse points which is not typically suitable for EVE, the algorithm is designed to operate optimally by operating on multiple sparse points together and also utilizing the DMA engine to organize data suitably, thereby utilizing the SIMD capability of EVE. Also, special instructions of EVE such as collated-store and scatter help greatly to save computation. Collate-store is used to collate all the converged points and further computation is performed only for those points. Scatter is used to later revert the results back to its original location. Once the SOF algorithm provides reliable optical flow tracks for various key points, triangulation is performed on the C66x

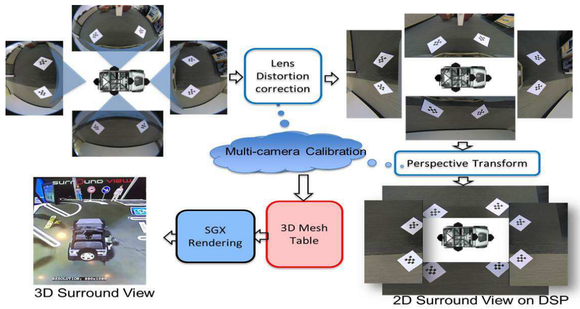

Figure 12. Surround View Algorithm Flow.

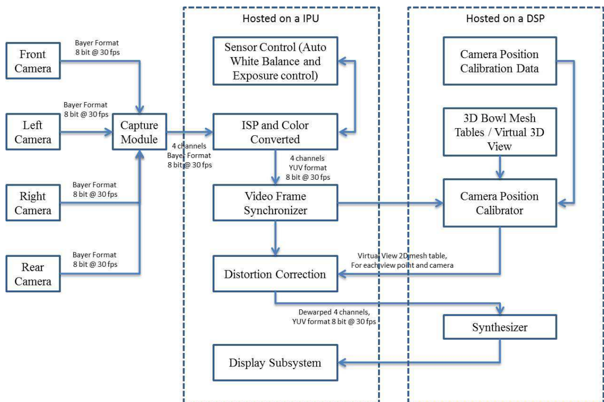

Figure 13. Surround View Data Flow.

DSP to obtain the 3D point cloud.

3.5. 3D Surround View (SV) System

Surround View (SV) systems are becoming popular in low and mid end cars [21]. 2D SV systems provides stationary top-down view of the surroundings from above the vehicle, whereas 3D SV systems provide rendering capability of surroundings of vehicle from any virtual view point and transitions between view points. Example of 2D SV and 3D SV are shown in Figure 10 and Figure 11 respectively. A SV system is constructed by stitching multiple video cameras placed around the periphery of the vehicle as shown in Figure 12. Typically, a dedicated Graphics Processing Unit (GPU) processor would be employed for composition of 3D SV. Since TDA3x does not have a GPU, the 3D SV is implemented using the combination of Lens Distortion Correction (LDC) accelerator and C66x DSP. The GPU typically stores entire representation of the 3D world and hence allowing the user to change view points. However, this can be optimized by projecting only those 3D points that are in the visible region. Distortion correction is required to correct the fish eye distortion present in the image sensors used in SV systems. The ISP of TDA3x SoC has a robust LDC accelerator which performs the distortion correction. In order to create multiple view points, a minimized 3D world map for all the cameras and view points are generated and

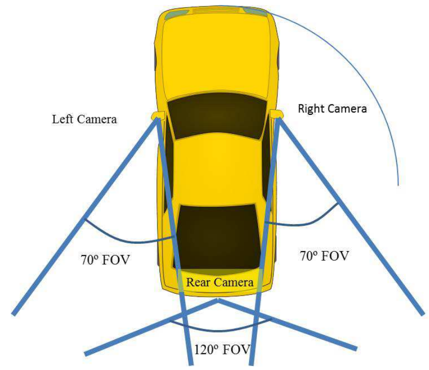

Figure 14. CMS Camera Placement Illustration.

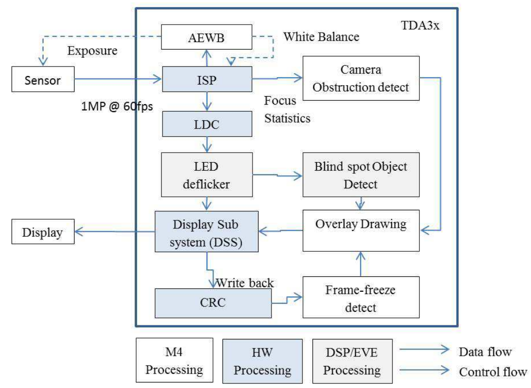

Figure 15. Algorithm and Data Flow in CMS.

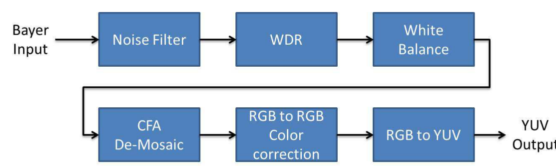

Figure 16. ISP Data Flow in CMS.

these view points are stored in non volatile memory. This can be done offline and once during the camera setup. When the system boots up, for all the valid view points, 3D world map is read and associated LDC mesh table is generated. Then, these outputs from the LDC are stitched together to obtain the 360° view for any view point. The data flow for the same is shown in Figure 13.

3.6. Camera Monitor System (CMS)

Figure 14 shows the placement of the cameras in a typical CMS. The algorithm processing involved in a CMS system and its partitioning in TDA3x SoC is shown in Figure 15. CMOS sensors are used to capture the scene. Frame-rate of 60fps is typically employed to reduce latency of the scene as viewed by the car driver. The data format of CMOS sensors is typically bayer raw data and it is passed through many stages of the ISP before it is converted to viewable video data as shown in Figure 16. Few key steps in the ISP are: Spatial Noise Filter, which helps to improve the image quality in low light conditions and Wide dynamic range, which increases the dynamic range of the scene so that bright areas are not clipped and at the same time details in shadow (or darker) regions are visible. This allows the CMS system to operate in variety of lighting conditions. ISP in TDA3x also outputs Auto White Balance (AWB) [10] and Auto Exposure (AE) statistics which is used by the AEWB algorithm to dynamically adjust the sensor exposure and scene white balance to adapt the cameras settings, to dynamically changing lighting conditions. Additionally, Focus statistics are output by the ISP, which indicates the degree to which the camera is in focus. When camera is obstructed by dirt or water on the lens, the scene will not be in focus. Due to safety critical nature of the CMS application, it is important to detect such scenarios. Focus statistics can be used in algorithms to detect such events, thereby warning the user of suboptimal scene capture. The harwdare LDC module is used to adjust for lens distortion correction due to wide angle field of view.

A common problem associated with camera for visual systems like CMS is the problem of LED flicker. LEDs are commonly used in car headlights, traffic signals and traffic signs. LEDs are typically pulsed light sources. However, due to persistence of vision our eyes cannot see the LED flicker. However camera sensor's, especially when they operate at low exposure time due to bright lighting conditions, could capture LED pulse in one frame and miss the LED pulse in next frame causing an unpleasing and unnatural flicker like effect. Worst case, it could happen that the LED pulse is not captured at all, say a red light or car headlight, thus giving a dangerous false scene representation to the user. De-flicker algorithm is typically employed to eliminate the flicker due to LED lights. This is a pixel processing intensive algorithm and is typically run on DSP/EVE. After LED de-flicker algorithm, the scene is displayed on the display via the display sub-system (DSS). A key safety measure in CMS system is informing the user in case of a frame freeze scenario. Since the user is not constantly looking at the mirror, it could happen that due to HW or SW failure, the data that is displayed on the screen is frozen with the same frame repeating. This can cause a hazardous situation to the road users. In TDA3x, this can be detected by using the DSS to write back the pixel data that is being displayed and then computing a CRC signature for the frame using a HW CRC module. If the CRC signature matches for a sequence of consecutive frames, then it implies that there is frame freeze somewhere in the system and a warning is given to the user or the display is blanked out. Additional analytics algorithms like object detect can be run in blind spot to notify driver.

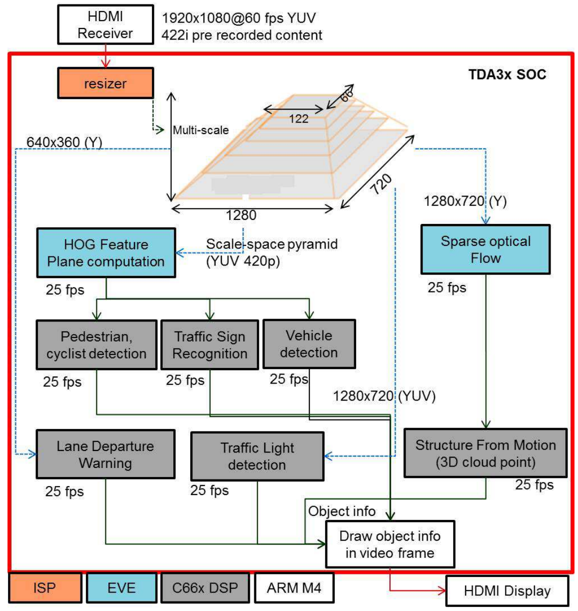

Figure 17. Algorithm partitioning for Front Camera Applications on TDA3x.

Table 1. Algorithm Configurations.

Algorithms	Frame Rate	Configuration Details on TDA3x SoC
Vehicle, Pedestrian Cycle Detect	25	Resolution = 1280x720, multi-scale Minimum object size = 32x64
Traffic Sign Recognition	25	Resolution = 1280x720, multi-scale Minimum object size = 32x32
Traffic Light Recognition	25	Resolution = 1280x720 Radii range = 8
Lane Depart Warning	25	Resolution = 640x360 Number of lanes detected = 2
Structure From Motion	25	Resolution = 1280x720 Number of SOF tracks = 1K points 3D cloud points generated = 800
Surround View System	30	Input Resolution = 4 channels of 1280x800 Output Resolution = 752x1008
Camera Mirror System	60	Number of video channels = 1 Input Resolution = 1280x800

4. Results and Analysis

In this section, we provide details of system partitioning and performance of multiple applications executing on the TDA3x. TI's TDA3x EVM is used as the experimental platform. In order to show case our algorithms, we captured multiple scenes with various camera sensors placed around the car. The video sequence contained urban roads with pedestrians, vehicles, traffic signs, lane marking, traffic lights and parking lot scenarios. This video sequence is then decoded via HDMI player and fed to the TDA3x EVM as shown in Figure 17. The algorithms then utilizes all the available compute blocks such as ISP, EVE, DSP and ARM Cortex M4 to perform various functions such as OD, LDW, TSR, SFM, SV and CMS. The output of these algorithms are provided to the ARM Cortex M4 to draw these markings in original video and sends out the annotated video to HDMI display. A LCD display is used to watch the video along with object markings to confirm the expected behavior of algorithms. The configuration parameters of these algorithms are listed in Table 1.

In case of front camera applications, the capture is configured for 25 frames per second (fps). As a first step, multiple scales of the input frame are created by using resizer functionality in ISP. The scale space pyramid is useful to detect objects of different sizes with a fixed size template. For every relevant pixel in these scales, histogram of oriented gradients (HOG) [9] signatures are formed. This module involves intensive calculations at pixel level and hence executed on EVE. After formation of the HOG feature plane, EVE runs SOF algorithm and C66x DSP1 runs the adaboost classifier stage of object detection algorithm. The classifier is executed separately for each object category such as pedestrians, vehicles, cyclists and traffic signs. From the scale space pyramid, 640x360 and 1280x720 scale is fed to DSP2 on which lane detection and traffic light recognition algorithms are run. After completion of SOF, EVE sends the optical flow tacks to DSP2 to perform triangulation to obtain 3D location of key points in the frame, thereby helping to identify distance of various objects in the scene. In this setup, the ARM Cortex-M4 manages the capture and display device, feeds the data to ISP and collects information from DSPs before annotating the objects and displaying them.

In case of Surround View application, 4 channels of video of resolution 1280x800 at 30 fps are captured from RAW video sensors (Bayer format) which is supported by the ISP. The ISP then converts the Bayer format data to YUV format for further processing. Auto white balance and exposure control algorithms ensure each video source is photometrically aligned. Then, the camera calibrator will generate the required mesh table for distortion correction based on the view point and distortion correction is performed. Synthesizer will then receive the corrected images and sticth to form the SV output with a resolution of 752x1008 at 30 fps.

In case of Camera Mirror System, each camera input operates on one TDA3x SoC. Each channel of video of resolution 1280x800 at 60 fps are captured from RAW video sensors (Bayer format) which is supported by the ISP. The ISP then converts the Bayer format data to YUV format as shown in Figure 16. Algorithms such as OD is run for Blind spot detection. Also, De-flicker algorithm is run to remove any LED flicker related issues, before displaying it to the driver. Table 2 shows the loading of the various processors of the TDA3x SoC while running these algorithms. For front camera applications, 53% of DSP1, 66% of DSP2, 79% of EVE and 33% of one ARM Cortex-M4 are utilized.

Table 2. Performance Estimates for different applications on TDA3x.

Algorithms	DSP1 Utilization	DSP2 Utilization	EVE Utilization	ARM Cortex-M4 Utilization	Frame Rate (fps)
Front Camera Analytics	53%	66%	79%	33%	25
Surround View System	45%	0	0	33%	30
Camera Mirror System	68%	20%	40%	30%	60

For SV application, 45% of DSP1 is utilized and 33% of one ARM Cortex-M4 is utilized. The unused C66x DSP and EVE can be used to run analytics on the SV output if needed. For CMS application, 68% of DSP1 is consumed to run Deflicker algorithm and 20% of DSP2 and 40% of EVE is used for blind spot detection algorithm.

5. Conclusion

ADAS applications require high-performance, low power and low area solutions. In this paper, we have presented one such solution based on Texas Instruments TDA3x device. We have provided insight into key algorithms of ADAS such as front camera, surround view and camera monitor systems. We have also presented the system partitioning of these algorithms across multiple cores present in TDA3x and their performance have been provided. We have shown that TDA3x platform is able to generate 3D SV efficiently, without a GPU. We have also shown that TDA3x platform is able to map various ADAS algorithms and still have headroom for customer's differentiation.

References

[1] Cortex-m4: Technical reference manual, arm inc., 2010.

[2] Global health observatory (gho) data, 2010.

[3] Tms320c66x dsp: User guide, sprugw0c, texas instruments inc., 2013.

[4] Tda3x soc processors for advanced driver assist systems (adas) technical brief, texas instruments inc., 2014.

[5] The top 10 causes of death, 2014.

[6] 2020 roadmap, rev. 1, european new car assessment programme, 2015.

[7] Advanced driver assistance system (adas) guide 2015, texas instruments inc., 2015. Supplied as material `SLYY044A`.

[8] S. Dabral, S. Kamath, V. Appia, M. Mody, B. Zhang, and U. Batur. Trends in camera based automotive driver assistance systems (adas). *IEEE 57th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1110–1115, 2014.

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:886–893, 2005.

[10] H. Garud, U. K. Pudipeddi, K. Desappan, and S. Nagori. A fast color constancy scheme for automobile video cameras. *International Conference on Signal Processing and Communications (SPCOM)*, pages 1–6, 2014.

[11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[12] B. Kisaanin and M. Gelautz. *Advances in Embedded Computer Vision*. Advances in Computer Vision and Pattern Recognition. Springer, 2014.

[13] Z. Lin, J. Sankaran, and T. Flanagan. Empowering automotive with ti's vision accelerationpac, 2013. http://www.ti.com/lit/wp/spry251/spry251.pdf.

[14] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *International Joint Conference on Artificial Intelligence*, 81:674–679, 1981.

[15] M. Mody, N. Nandan, H. Sanghvi, R. Allu, and R. Sagar. Felixble wide dynamic range (wdr) processing support in image signal processor (isp). *IEEE International Conference on Consumer Electronics (ICCE)*, pages 467–479, 2015.

[16] M. Mody, P. Swami, K. Chitnis, S. Jagannathan, K. Desappan, A. Jain, D. Poddar, Z. Nikolic, P. Viswanath, M. Mathew, S. Nagori, and H. Garud. High performance front camera adas applications on ti's tda3x platform. *IEEE 22nd International Conference on High Performance Computing (HiPC)*, pages 456–463, 2015.

[17] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. *European Conference on Computer Vision (ECCV)*, 2:709–720, 1996.

[18] P. Viola and M. Jones. Fast and robust classification using asymmetric adaboost and a detector cascade. *Advances in Neural Information Processing System*, 2001.

[19] P. Viswanath and P. Swami. A robust and real-time image based lane departure warning system. *In Proceedings. IEEE Conference on Consumer Electronics (ICCE)*, 2016.

[20] P. Viswanath, P. Swami, K. Desappan, A. Jain, and A. Pathayapurakkal. Orb in 5 ms: An efficient simd friendly implementation. *Computer Vision-ACCV 2014 Workshops*, pages 675–686, 2014.

[21] B. Zhang, V. Appia, I. Pekkucuksen, A. U. Batur, P. Shastry, S. Liu, S. Sivasankaran, K. Chitnis, and Y. Liu. A surround video camera solution for embedded systems. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 676–681, 2014.