2D Object Recognition for Industrial Machine Vision with Processor SDK on Sitara™ AM57x Processors

Hongmei Gou, Djordje Senicic

ABSTRACT

2D object recognition is widely used in industrial machine vision applications. Equipped with Arm Cortex-A15 CPUs, DSPs, and multiple subsystems such as IVAHD, GPU, and PRU-ICSS, TI's Sitara AM57x processors are power efficient SoCs that meet the need of 2D object recognition and many other industrial machine vision applications. With open programming and open source technologies such as OpenCV, the Processor Software Development Kit for Linux running on a Sitara AM57x enables fast development and deployment of 2D objection recognition applications for industrial machine vision.

Contents

List of Figures

List of Tables

Trademarks

All trademarks are the property of their respective owners.

1 Overview

2D object recognition is widely used in industrial machine vision applications such as automated inspection, robotic vision, and high-speed identification systems. One specific example is industrial machine vision on conveyor belts. 2D object recognition can be used to identify objects and make a quality control decision to accept or reject the objects. It can also facilitate robotic manipulator control to pick a target object and place it to a specified location.

Industrial applications often have well controlled and illuminated environments. This is critical to ensure reduced complexity and increased accuracy of the system. By controlling factors such as known distance and known target object, classic methods based on hand crafted features and using standard classification methods like SVM (Support Vector Machine) are suitable. Within these constrained systems, computationally expensive algorithms, such as DNN (Distributed Neural Network) and CNN (Convolutional Neural Network), are generally not required.

TI’s Sitara AM57x processors are power efficient SoCs (System-on-Chips) that meet the need of machine vision in industrial applications, including 2D object recognition and many others.

The Processor Software Development Kit (Processor-SDK) for Linux provides a fundamental software platform for development, deployment and execution of Linux based applications. It leverages open programming and bundles various open source technologies, including standard Linux driver, OpenCV/OpenCL/OpenGL, Qt, and others. Processor-SDK Linux running on a Sitara AM57x enables fast development and deployment of 2D objection recognition applications for industrial machine vision.

2 TI’s Sitara AM57x

TI’s Sitara AM57x family is a scalable line of processors that offers maximum flexibility of a highly integrated mixed processor solution. Devices in the family combine programmable video processing with a broad and highly integrated peripheral set which is well suitable for industrial applications. The AM57x is equipped with single or dual-core Arm® Cortex®-A15 RISC CPUs with Arm® NEON™ technology, and one or two TI C66x VLIW floating-point DSP cores. The Arm Cortex lets developers keep separate control functions from other algorithms that are programmed on the DSPs and coprocessors. The separated control functions reduce the complexity of the system software. The Arm Cortex-A15 CPU supports multiple operating frequencies at a range of up to 1.5GHz. The C66x DSP core is more efficient than the Arm A15 at running the computational pieces of machine vision algorithms and is an ideal offload engine to provide a power-efficient embedded solution.

The AM57x is composed of multiple main subsystems to support video, image, and graphics processing, including image and video accelerator high-definition (IVA-HD) subsystem, video input capture (VIP), video processing engine (VPE), graphics processing unit (GPU), and display subsystem (DSS). The AM57x processor family is highly scalable with devices that have a subset of the above subsystems so that an optimal device for a system’s needs can be chosen. AM57x devices support multiple camera interfaces like VIP, USB and, on some devices, Camera Serial Interface (CSI).

The AM57x processors are configured with two dual-core Programmable Real-Time Unit and Industrial Communication Subsystems (PRU-ICSS). The PRU-ICSS can be used for communication protocols such as EtherCAT Master and Slave, PROFINET, Ethernet/IP, SERCOS, and so forth. This enables real-time industrial communication and automation control.

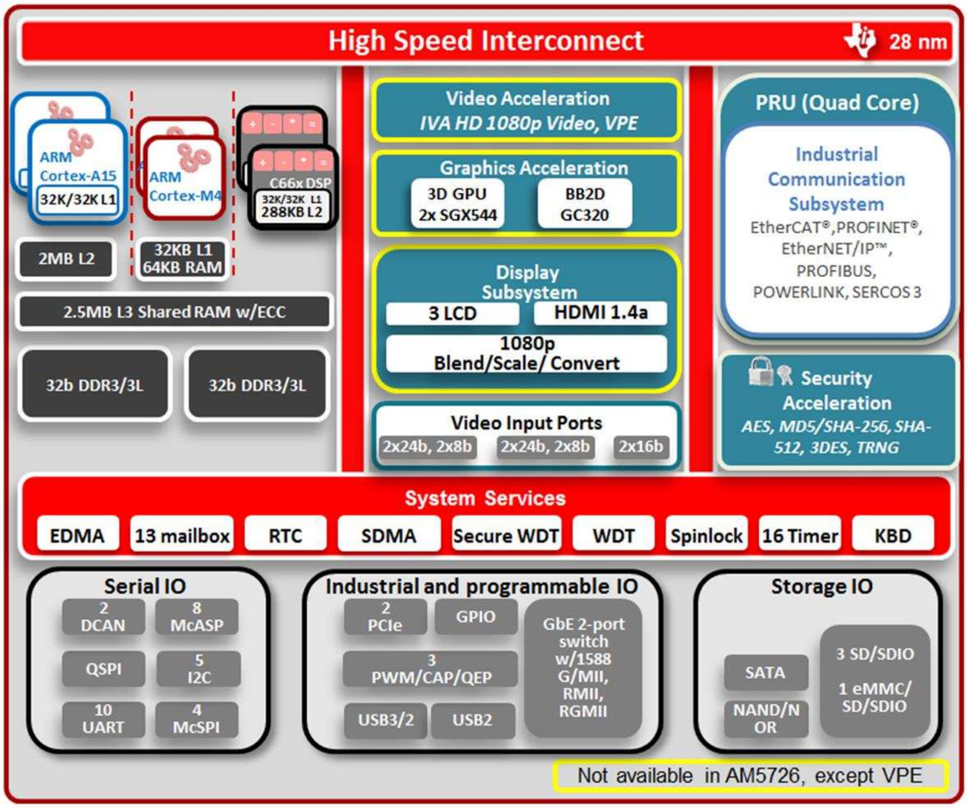

Figure 1. Block Diagram of TI's Sitara AM57x

3 Processor SDK Linux

With open programming, Processor SDK Linux provides a good starting point to develop and deploy Linux embedded systems on TI Processors such as AM57x. Besides the typical board support packages such as bootloader, Linux kernel, and filesystem, it also contains tools for developing on TI Processors, pre-built libraries that are usable without the need of rebuilding, and out of box feature applications that can run as is on TI Processors. The software stack below illustrates at a high level the various components provided with the Processor SDK for Linux.

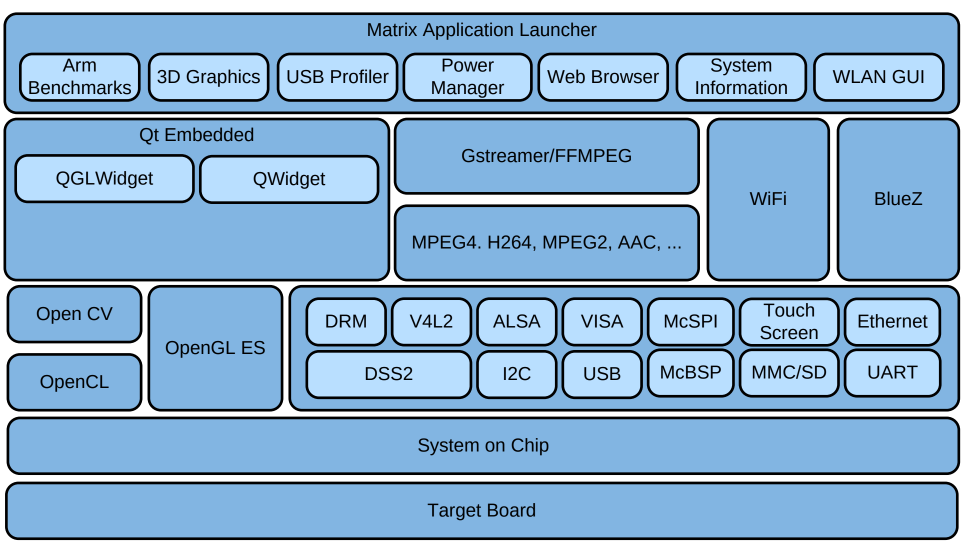

Figure 2. Software Stack of Processor SDK Linux

4 2D Object Recognition with Processor SDK Linux on Sitara AM57x

Processor SDK Linux on AM57x enables a ready-to-show, all-in-one demo for 2D object recognition in industrial machine vision, including image acquisition, pre-processing, object detection, counting, classification and position/dimension/orientation extraction, visualization and reporting. It provides a completely embedded solution on a single SoC of AM57x, and can be configurable with simple operator HMI (Human Machine Interface). With standard Processor SDK Linux component, various reporting options can be supported, including UART, Ethernet, and USB.

4.1 Building Blocks of 2D Object Recognition Systems

As shown in the diagram below, the major building blocks of the 2D object detection systems include:

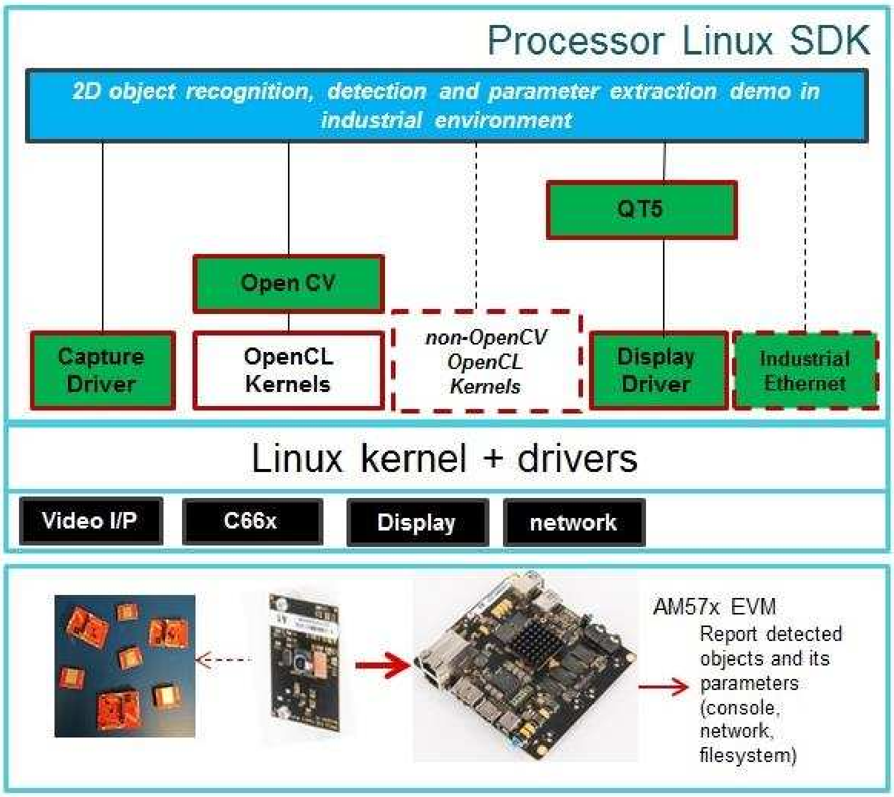

Figure 3. 2D Object Recognition with Processor SDK Linux on Sitara AM57x

- Capture driver: the camera module on AM57x acquires color frames with the standard V4L2 Linux driver, feeding to the OpenCV framework.
- OpenCV: provides open-source libraries with various computer vision algorithms, with computational efficiency and strong focus on real time applications. The feature2d module provides multiple ready-to-use kernels to facilitate 2D object detection.
- OpenCL: TI's OpenCV implementation for hybrid Arm-DSP devices such as AM57x offloads signal-processing-rich algorithms to the DSP via OpenCL to achieve high efficiency.
- Qt5: provides a user friendly HMI to set the configuration as well as report data from the detection
- Display driver: renders and displays the camera input as well as the detection results with graphics accelerator and display subsystem.
- Industrial Ethernet: transmits data from 2D object recognition (e.g., object type, orientation, size, location), and further sends events/information for PLC control via industrial communication protocols, such as EtherCAT, Profinet, and CCLink IEF Basic.

4.2 OpenCV Kernels of Interest

OpenCV has been integrated into Processor SDK Linux. It provides various modules for computer vision, including calibed, core, feature2d, flann, imgcodecs, imgproc, ml, objdetect, photo, shape, stitching, superres, video, and videoio.

OpenCV available with Processor SDK also allows OpenCV kernels to be offloaded to the C66x DSPs of Arm-DSP devices such as AM57x. The specialized architecture of the C66x DSP provides higher performance and better power-efficiency for many machine vision algorithms. By offloading OpenCV kernels to the C66x DSP(s), the overall system performance can be greatly increased as the Arm A15(s) are freed up to do other processing. The DSP optimized OpenCV OpenCL kernels currently supported by Processor SDK Linux include erode, dilate, SobelX/SobelY, threshold, GaussBlur(3x3), convertScaleAbs, and MOG2 (mixture of Gaussians).

The 2D features framework (feature2d) from OpenCV provides various kernels such as the feature point detectors, descriptors and matching framework, which can facilitate 2D object detection in machine vision applications. As listed in [1], there are multiple corner detectors such as Harris corner detector, Shi-Tomasi corner detector, and corner detection in subpixels. Another category is feature detection and description, and further for feature matching. Furthermore, kernels are provided for finding a known object, detecting planar objects, and planar tracking. Details of the feature2d kernels including the goal, the theory, and the code snippets can be found from [1].

The tutorial demos for some of the feature2d kernels have been provided by Processor SDK Linux and are ready-to-use on an AM572x EVM (TMDSEVM572X). For example, the demos below for corner detections can be found from the filesystem of AM57x under /usr/share/OpenCV/samples/bin directory:

- Harris corner detector: cpp-tutorial-cornerHarris_Demo
- Shi-Tomasi corner detector: cpp-tutorial-goodFeaturesToTrack_Demo
- Detecting corners location in subpixels: cpp-tutorial-cornerSubPix_Demo

For tutorial demos which are not installed on the filesystem of AM57x, such as feature matching with FLANN [2], and Features2D + Homography to find a known object [3], Processor SDK Linux has provided the OpenCV libs and headers as well as the build infrastructure for natively compiling OpenCV sample applications on the target. The steps are simple and allow quick turnaround from the source code to a binary executable on AM57x target.

1. Identify the OpenCV sample applications of interest, e.g., the tutorial demo for feature matching with FLANN [2].
2. Copy the source code of the sample application as a .c file on target, e.g., copying the Code from [2] as tutorial_feature_flann_matcher.c.
3. Run command below to link the opencv libs as needed and build the OpenCV sample application *g++ -I/usr/local/include/opencv -I/usr/local/include/opencv2 -L/usr/local/lib/ -g -o tutorial_feature_flann_matcher tutorial_feature_flann_matcher.c -lrt* ***-lopencv_core -lopencv_highgui -lopencv_features2d -lopencv_flann -lopencv_imgcodecs -lopencv_xfeatures2d***
4. After the command in step 3 is completed, it generates a binary executable at the same place as the source .c file, with the name specified in the gcc command above, e.g., *tutorial_feature_flann_matcher* for this example.

4.3 Run OpenCV Kernels on AM57x for 2D Object Detection

This section shows the 2D object detection results running OpenCV feature2d kernels on an AM572x EVM. The natively re-built tutorial_feature_flann_matcher demo from the previous section is used as the example.

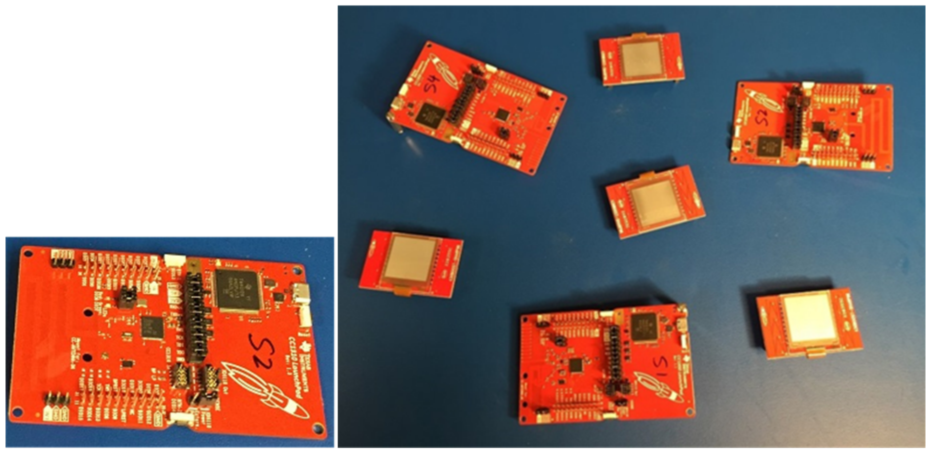

Figure 4. Test Vector for 2D Object Recognition Results on AM572x

First, the test vector is created as above, with a single EVM (EVM.jpg on the left) as the object to be identified, and multiple EVM objects along with several smaller LCD boards (MultiBoards.jpg on the right) as the scene from which the EVM objects shall be identified.

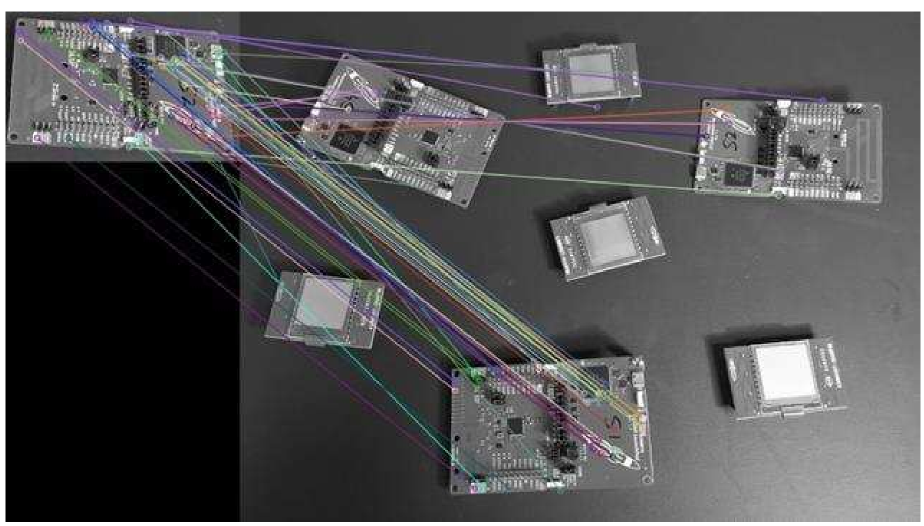

Figure 5. 2D Object Recognition Results on AM572x

Then, run the re-built tutorial_feature_flann_matcher demo binary on an AM572x EVM, and the detection results are shown as above. We can see that all the three EVM objects are correctly identified. The console log from running the demo binary is attached below. It lists the good match points between the single EVM object and the scene with multiple objects.

```
root@am57xx-
evm:/usr/share/OpenCV/samples/source# ./tutorial_feature_flann_matcher EVM.jpg MultiBoards.jpg
-- Max dist : 0.530873
-- Min dist : 0.095940
init done
Using Wayland-EGL
wlpvr: PVR Services Initialised
-- Good Match [0] Keypoint 1: 2  -- Keypoint 2: 6
-- Good Match [1] Keypoint 1: 4  -- Keypoint 2: 3
-- Good Match [2] Keypoint 1: 5  -- Keypoint 2: 14
-- Good Match [3] Keypoint 1: 33  -- Keypoint 2: 60
-- Good Match [4] Keypoint 1: 41  -- Keypoint 2: 3
-- Good Match [5] Keypoint 1: 57  -- Keypoint 2: 289
-- Good Match [6] Keypoint 1: 66  -- Keypoint 2: 242
-- Good Match [7] Keypoint 1: 85  -- Keypoint 2: 366
-- Good Match [8] Keypoint 1: 90  -- Keypoint 2: 254
-- Good Match [9] Keypoint 1: 93  -- Keypoint 2: 411
-- Good Match [10] Keypoint 1: 95  -- Keypoint 2: 289
......
```

5 Conclusions

This paper provides a completely embedded solution for 2D object recognition in industrial machine vision applications. It is enabled by Processor SDK Linux with TI's Sitara AM57x Processors. It presents the building blocks for the 2D object recognition system, leveraging open programming of Processor SDK Linux. It also discusses how to utilize the 2D features framework (feature2d) kernels from OpenCV, and illustrates with an example on how to build and run the feature2d kernels for 2D object recognition on TI's AM572x EVM.

6 References

[1] OpenCV Tutorials, 2D Features framework (feature2d module): http://docs.opencv.org/3.1.0/d9/d97/tutorial_table_of_content_features2d.html

[2] OpenCV Tutorials, 2D Features framework (feature2d module), Feature Matching with FLANN, http://docs.opencv.org/3.1.0/d5/d6f/tutorial_feature_flann_matcher.html

[3] OpenCV Tutorials, 2D Features framework (feature2d module), Feature 2D + Homography to find a known object, http://docs.opencv.org/3.1.0/d7/dff/tutorial_feature_homography.html