Course Project Report on

Real-Time Target Tracking

Digital Signal Processing -
System Design and Implementation [EE750]

Aswin Ajayan – 183079032

Jinit Patil – 18307R031

Shubham Singhania – 183079025

Krishna Khairnar – 18307R027

TA – Pulkit Jain

Under the Guidance of

Prof. V. Rajbabu

Department of Electrical Engineering

India Institute of Technology, Bombay

Abstract

In this project, we have implemented a real-time object tracker. We have used the TMDSEVM6678LE (C6678L) Evaluation module from Texas Instruments (TI) as the platform for the implementation of object detection. TIVA C LaunchPad (TIVA) Evaluation Kit from TI is used to implement the real-time tracking of the object with the help of a laser, mounted on a custom 3D printed assembly. The servo motors provide movement in horizontal as well as vertical direction.

Introduction

Object detection and motion estimation, investigated for a long time, is still an emerging field of particular interest due to a variety of applications such as navigation, surveillance, self-driving cars and robotics.

Real-time object tracking is the process of estimating or tracking the path of an object using a camera in real-time. The objective is to segment the object of interest from the video scene frame by frame and keeping track of its movement by associating video frames consecutively. At its core, the tracking algorithm uses tracking-by-detection paradigm.

To implement real-time object tracking, we have used C6678 digital signal processor from Texas Instruments which is a high-performance, low-power DSP with floating-point capabilities. It has eight C66x VLIW cores which runs at 1 GHz and dissipates a maximum power of 10 W.

Also, the software support from TI is also available for C6678L as well as TIVA. We have used Code Composer Studio (CCS) CCSv5.2 and Multi-Core Software Development Kit (MCSDK) version 2.

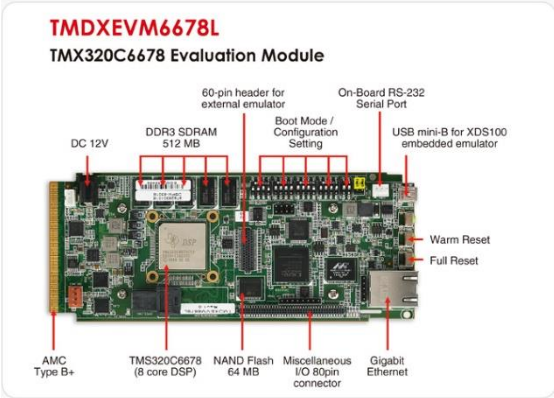

Figure 1 TMDSEVM6678 Lite Evaluation module

Brief overview of the block diagram:

- A USB camera is used to acquire the video signal. This acquired signal is passed frame by frame in the form of UCP packets to C6678 using UDP connection.
- C6678 perform object detection and tracking using thresholding and generate (x,y) coordinates of the moving object in real-time.
- These (x,y) coordinates are transmitted back to PC, which serially transmits it to TIVA using UART communication.

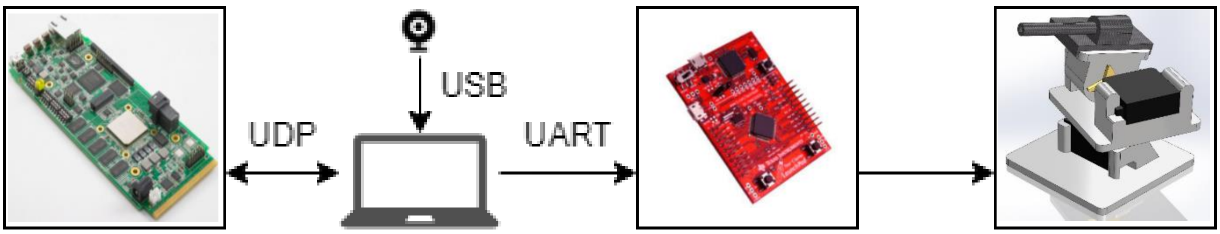

Figure 2 Block Diagram

- TIVA generates a control signal according to the coordinates to control the two servo motors mounted on a custom 3D printed assembly.
- This assembly carries a laser pointer which tracks the motion of the object in real-time.

Implementation

1. *Image Acquisition*

Video signal is acquired using a generic USB Camera. The 640X480 resolution of the acquired signal is converted to 320X240 resolution in computer. The scaled-down version of the acquired signal is transmitted to C6678L using User Datagram Protocol (UDP) connection from computer.

Multi-Core Software Development Kit (MCSDK) from TI provides Network Design Kit (NDK) which can be used to implement UDP connection. The video data is processed frame by frame, where each frame is of 320X240 bytes. These frames are converted to packets to be transmitted using UDP. Each packet is of 1024 bytes. Therefore, a complete frame can be transmitted using 75 such packets.

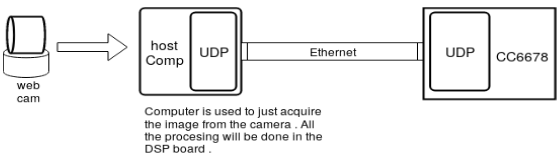

Figure 3 Image Acquisition model

The latency in image acquisition from USB Camera is more compared to the latency of transmission using UDP in our case. Therefore, UDP suits our requirement.

2. *Object Detection and Tracking*

The received frames are in RGB image space. In C6678L, this RGB image space is transformed into Grey image space. The mask for object detection can be obtained using thresholding and contrast of the background plane.

The object has a specific range of grey pixel values. On the other hand, the background has a different set of grey pixel values. This contrast in background and object is used to determine threshold which is used on the object pixel values to obtain a mask. This mask is ANDed with the grey image space to obtain a marker. This marker shows the presence of an object on the background.

The implementation of image acquisition, and object detection and tracking on C6678L are done using threads. Two threads run in parallel. One thread is assigned for UDP (T1), and another is dedicated for object detection and thresholding (T2). These two threads run on two different cores. The priority of T1 is higher than T2.

Image transfer follows concept similar to the one used in Ping-Pong buffer implementation. Acquisition is done on one buffer and processing is done on the other buffer. At any point of time there will be two different frames in the memory.

Once and entire frame is captured, a global variable is set to signal the process task. Process task pickup the signal, acts on the frame and returns the x,y coordinates

Optimization:

The optimization of both these steps, that is, image acquisition, and object detection and tracking is achieved by parallelizing the two threads. We have used memcpy() which reduce latency through pointer manipulation . The difference in the latency was quite apparent with and without optimization.

3. *Use of Laser to track the object in real-time*

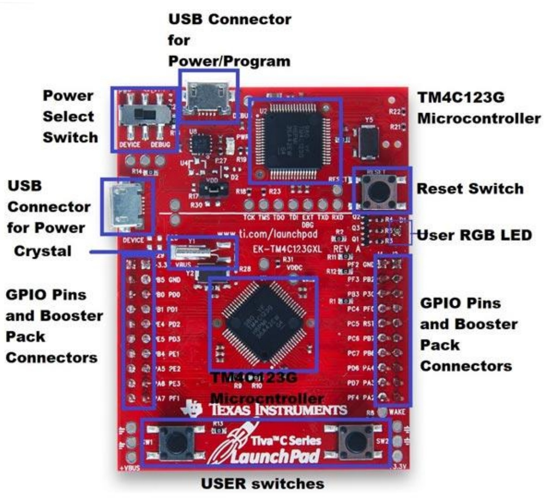

Figure 4 TIVA C LaunchPad Evaluation Kit

The TIVA Series TM4C123G LaunchPad Evaluation Board (EK-TM4C123GXL) is a low-cost evaluation platform for ARM® Cortex™-M4F-based microcontrollers. The Tiva C Series LaunchPad design highlights the TM4C123GH6PMI microcontroller USB 2.0 device interface, hibernation module, and motion control pulse-width modulator (MC PWM) module. The TIVA Series LaunchPad also features programmable user buttons and an RGB LED for custom applications.

The calculated (x,y) coordinates from the C6678L is transmitted to the PC using UDP communication. Termios API was used to achieve this.

TIVA receives (x,y) coordinates from PC using UART protocol. All (x,y) coordinates are transmitted as characters.

The transmitted data format is as follows:

X(X-coordinates)Y(Y-coordinates)\r

for example, x = 1234 and y = 5678 is received as "X1234Y5678\r"(11 characters).

After receiving these coordinates as a character stream, integer conversion happens for further processing. With prior knowledge of the distance between the background plane and the assembly, the required angles can be calculated. These angles are provided to the assembly in the form of pulse width modulation (PWM). The assembly has two degrees of freedom (altitude and azimuth). Two servo motors control the altitude and azimuth of LASER mounted on the assembly. Servo motor accepts PWM signals and converts it in the form of rotation of mechanical arm with the desired angle. PWM signal is in the form of a high pulse of 1-2 µs with a frequency of 50 Hz. 0˚ corresponds to 1 µs pulse width, and 180˚ corresponds to 2 µs pulse width. After passing the desired pulse width signal, servo motors rotate at the desired angle to point LASER on target.

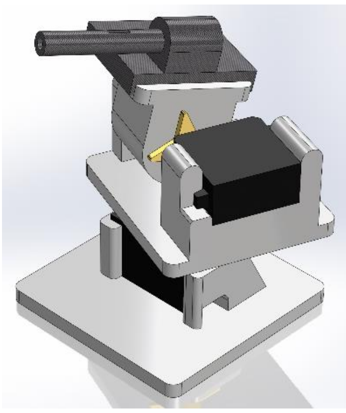

Figure 5 Laser mounted custom 3D printed assembly with two servo motors

Conclusion

In this project, we were able to implement object detection and tracking, and UDP connection for movement of data to and fro.

We have also implemented a better algorithm for object detection and tracking in python called Histogram Back Projection.

In our case, the resources required for computations were less. We have used two cores out of eight cores available. The constraint is due to unavailability of compiler support for NDK to be implemented on more than two cores. However, C6678L is capable of supporting much more compute-intensive digital signal processing, image processing and machine learning based applications.

To parallelize the implementation on all the eight cores, OpenMP has to be used which was not possible in our case because of the NDK usage for UDP connection.

Furthermore, due to the overwhelming amount of documentation available from TI, the initial set up of the C6678 EVM is very time-consuming for a time-constrained project which uses older software suites for development. The documentation available is generalized for the available range of products called Processor Software Design Kit (SDK). Therefore, to migrate from the older versions to newer versions is in itself a considerable task.

Future Work

Given enough amount of time, C6678 can support very sophisticated applications from elementary image processing to deep learning algorithms.

To further improve the real-time target tracking, all the eight cores of the C6678 can be used with the help of OpenMP. Next implementation can altogether remove PC, and the camera can be directly interfaced with the C6678. In this case, processor SDK can be used.

An alternative is to use Sitara type of boards, which have an ARM core as well. It is easier to interface common camera modules through the OS loaded on ARM core and offload all of the processing to the DSP cores.

Histogram Back Projection can be implemented on C6678 which provides better accuracy and real time performance.

References

1. http://software-dl.ti.com/processor-sdk-linux/esd/docs/latest/linux/Overview_Getting_Started_Guide.html
2. http://processors.wiki.ti.com/index.php/BIOS_MCSDK_2.0_Getting_Started_Guide
3. https://opencv.org/
4. http://www.ti.com/lit/ug/spmu296/spmu296.pdf
5. https://github.com/xanthium-enterprises/Serial-Port-Programming-on-Linux
6. http://www.codebind.com/cpp-tutorial/install-opencv-ubuntu-cpp/
7. https://docs.opencv.org/3.4/dc/df6/tutorial_py_histogram_backprojection.html

Project Code

The entire code and detailed instructions on how to run the code is available at:

https://github.com/aswinpajayan/C6678-objectTracking.git

Please, refer to the README.md and follow the steps if you want to recreate the project. Please feel free to contribute as well.