IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 4, Issue 2, Ver. III (Mar-Apr. 2014), PP 01-06
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Real Time Implementation of Computer Vision Algorithms on Beagleboard

Pradeep Kumar M^1 and Lokesha H^2
1Project Trainee, Digital signal processing and systems Group Aerospace Electronics & Systems Division, National Aerospace Laboratories, Bangalore, India
2Scientist.,Digital signal processing and systems Group Aerospace Electronics & Systems Division, National Aerospace Laboratories, Bangalore,(Council of Scientific & Industrial Research (CSIR), New Delhi),India

Abstract: *The exact reconstruction and restoration of perceived image and its properties such as shape, illumination are the severe problem in an image processing field. This problem can be successfully overcome through efficient computer vision algorithm. This paper presents an attempt to implement image processing algorithm on embedded hardware provided by low cost processor from Texas Instruments (TI). In this approach, it is planned to study the feasibility of using a low cost microcontroller board in Computer Vision application. In this approach the computer vision algorithm is often implemented by using a low cost Microcontroller like Beagle Board. Beagle Board is an Open Multimedia Application Platform (OMAP), System on Chip Processor which includes ARM Cortex-A8 core of Texas Instruments designed C64X+ Digital Signal Processor (DSP) with On-board Graphics engine of Dual Data Rate Random Access Memory (DDRAM).*

Keywords: *Real-time image processing, Computer Vision, Sobel edge detection, Median filter.*

I. Introduction

Computer vision is the emulation of the visual ability of human beings using computer. This is a field that includes methods for acquiring, processing, analysing, and understanding images and video streams [multiple frames].A theme in the development of this field has been to duplicate the abilities of human vision by electronically perceiving and understanding an image. Real time computer vision applications like video streaming on cell phones, remote surveillance and virtual reality have stringent performance requirements but can be severely restrained by limited resources.

The use of optimized algorithms is vital to meet real-time requirements. This paper presents a work on performance optimization of basic computer vision algorithms such as sobel edge detection for preserving the important structural properties of an image and noise reduction by median filter algorithm. The algorithms are benchmarked on the Beagleboard-xM, which is a revised new low-cost, low power platform based on the Texas Instruments (TI) OMAP DM3730CBP processor consists of an asymmetric dual -core combined architecture, including an ARM and a DSP supported by a shared memory. OpenCV, which is computer vision library developed by Intel corporation was utilized for algorithms.

The computer vision algorithms are generally based on Digital Signal Processors which are efficient in DSP algorithms execution, but they are expensive. Many applications of computer vision algorithm such as mobile applications, video streaming, smart cameras, vehicle navigation, smart traffic light systems and virtual reality require low cost controllers to reduce the overall system cost. In this direction, the microcontrollers will be of right choice as the price of micro controller is about one-seventh of that of a DSP processor.

The implementation of computer vision algorithm is computationally intensive and resource exhaustive. But in our approach in order to speed up the computation process we are arriving at the available resource with set of OpenCV libraries [3] for generation of efficient computer vision algorithm. OpenCV is an open source computer vision library developed by intel corporation.

One common factor that lies with most of the signal and image processing algorithm is that it is highly computationally memory intensive. Specifically for image processing, the memory is the key requirement. Embedded hardware always faces the memory limitations. To overcome these criteria the Beagleboard-xM[2] is associated with special unique feature is that the Beagle Board comes with a Micron POP (Package on Package) memory. This means that the NAND and SDRAM are mounted on top of the OMAP3530 processor. This Micron POP memory provides (i) 2 GB NAND * 16 (256MB) (ii) 2GB MDDR SDRAM *32 (256MB).Additional memory elements can be added by using a SD or MMC in the SD/MMC slot or by using a USB thumb drive/hard drive. In our setup we have used the SD card.

Hence in this thesis an efficient low cost microcontroller will be used as a controlling unit for the implementation and demonstration of few real-time computer vision algorithms

II. Background

Sobel edge detection and Median Filter algorithms are find intensive application in preserving the important structural properties of an image [7]. Trade-off made between noise removal and preservation of fine, low-contrast detail that may have characteristics similar to noise. These are most common classes of algorithms used in computer vision.

2.1 Sobel Edge Detection

Sobel algorithm generally finds edges where the gray scale intensity of the image changes the most. These areas are found by determining gradients of the image. The operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives - one for horizontal changes, and one for vertical. If we define **A** as the source image, and $\mathbf{G}_x$ and $\mathbf{G}_y$ are two images which at each point contain the horizontal and vertical derivative approximations, the computations are as follows:

$$Gx=\begin{bmatrix}+1 & 0 & -1\\ +2 & 0 & -2\\ +1 & 0 & -1\end{bmatrix}*A \quad \text{and} Gy=\begin{bmatrix}+1 + 2 + 1\\ 0 \quad 0 \quad 0\\ -1 - 2 - 1\end{bmatrix}*A$$

Where $*$ denotes the 2-dimensional convolution operation.

Since the Sobel kernels can be decomposed as the products of an averaging and a differentiation kernel, they compute the gradient with smoothing. For example,

Where Gx can be written as

$$Gx \quad = \quad \begin{bmatrix}+1 & 0 & -1\\ +2 & 0 & -2\\ +1 & 0 & -1\end{bmatrix} = \begin{bmatrix}1\\ 2\\ 1\end{bmatrix}\begin{bmatrix}+1 & 0 & -1\end{bmatrix}$$

The *x*-coordinate is defined here as increasing in the "right"-direction, and the *y*-coordinate is defined as increasing in the "down"-direction.

At each point in the image, the resulting gradient approximations can be combined to give the gradient magnitude, using:

$$G = \sqrt{Gx^2 + Gy^2}$$

Using this information, we can also calculate the gradient's direction:

$$\theta = \operatorname{atan}\left(\frac{G_y}{G_x}\right)$$

Where, for example, **Θ** is 0 for a vertical edge which is darker on the right side.

2.2 Noise reduction by Median Filter

In signal processing, it is often desirable to be able to perform some kind of noise reduction on an image or signal. The medianfilter is nonlinear digital filtering technique, often used to remove noise. Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise.

The median filter considers each pixel in the image in turn and looks at its nearby neighbours to decide whether or not it is reprehensive of its surroundings. Instead ofsimply replacing the pixel value with themean of neighbouring pixel values; it replaces it with the median of those values. The median is first calculated by first sorting all the pixel values from the surrounding neighbourhood into numerical order and then replacing the pixel being considered with the middle pixel value.(If the neighbourhood under consideration contains an even number of pixels, the average of the two middle pixel values are used . Figure 2.2 illustrate an example calculation.

	123	**125**	**126**	**130**	**140**	
	122	**124**	**126**	**127**	**135**	
	118	**120**	**150**	**125**	**134**	
	119	**115**	**119**	**123**	**133**	
	111	**116**	**110**	**120**	**130**	

Figure 2.2 **Neighbourhood values:**

115,119,120,123,124,125,126,127,150.

Median value: 124

As it can be seen, the central pixel value of 150 is rather unrepresentive of the surrounding pixel and is replaced with the median value: 124. A 3×3 square neighbourhood is used here ---- larger neighbourhoods will produce more severe smoothing.

The median value is determined by placing the brightness's in ascending order and selecting the centre value. The obtained median value will be the value for that pixel in the output image. Figure 2.3 shows an example of the median filter application, as in this case habitually a 3×3 median filter is used.

Brightness Values

	-1	0	1	← i
-1	10	30	5	
0	20	200	20	
1	15	10	30	
↑				
j				

Brightness Values in Order

5 10 10 15 **20** 30 30 200

⇧

Median

Figure 2.3 3×3 median filter

The main idea of the median filter is to run through the signal entry by entry, replacing each entry with the median of neighbouring entries. The pattern of neighbours is called the "window", which slides, entry by entry, over the entire signal. For 1D signal, the most obvious window is just the first few preceding and following entries, whereas for 2D (or higher-dimensional) signals such as images, more complex window patterns are possible (such as "box" or "cross" patterns). Note that if the window has an odd number of entries, then the median is simple to define: it is just the middle value after all the entries in the window are sorted numerically. For an even number of entries, there is more than one possible median.

III. Hardware Implementation

The hardware implementation of the projects will involve image acquisition using camera, Beagleboard-xM to implement edge detection and median filter algorithms and a display device such as an LCD to view the processed image.

3.1 Image Acquisition Device

The image is acquired through a PS3 camera with digital outputs (Not PAL/NTSC based output used in conventional cameras).Camera which supports following resolutions:

Features

- 60 hertz at a 640×480 pixel resolution, and 120 hertz at 320×240 pixels.
- Two adjustable fixed focus zoom lens.
- It can be set to a 56° field of view (red dot).
- 75° field of view (blue dot) for long shot framing.

Another important aspect of any live image capturing device is its "Frame rate" or sampling rate. In this camera normally 120 frames per second are processed to view a good motion picture.

3.2 Beagleboard-xM

A modified version of the Beagle Board called the BeagleBoard-xM[2] started shipping on August 27, 2010. The BeagleBoard-xM measures in at 82.55 by 82.55 mm and has a faster CPU core (clocked at 1 GHz compared to the 720 MHz of the BeagleBoard), more RAM (512 MB compared to 256 MB), on-board Ethernet jack, and 4 port USB hub. The BeagleBoard-xM lacks the on-board NAND and therefore requires the OS and other data to be stored on a micro SD card. The addition of the Camera port to the -xM provides a simple way of importing video via Leopard Board cameras [1].

Since BeagleBoard-Xm is associated with more advanced features in recent existing version, it has been selected to work for the proposed project.

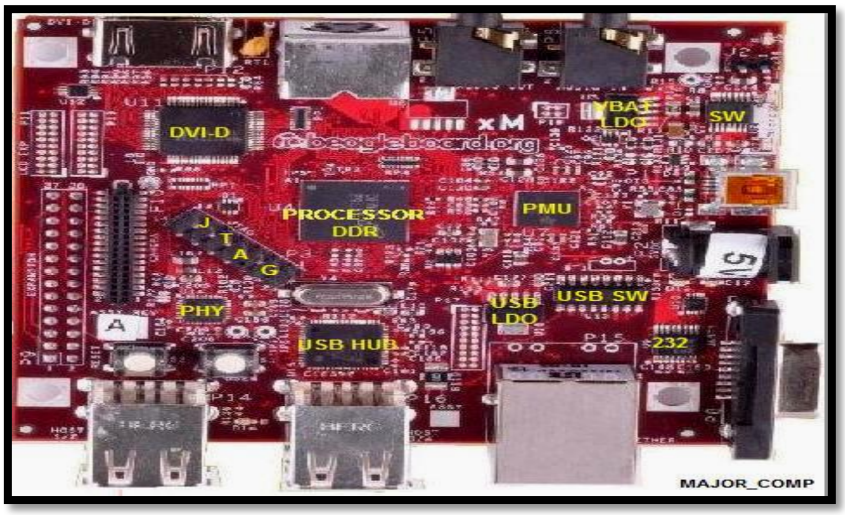

Figure 3.2 Beagle Board-xM

Features:

• DM3730CBP Microprocessor – 1200 DMIPS, based on ARM Cortex-A8 running at 1GHz
• TMS320C64x+ DSP for versatile signal processing at up to 800MHz
• Memories: 512MB of RAM and lacks the onboard NAND.
• SD/MMC Card slot, can be used as OS file system, file storage and more
• USB Host port which gives the option to connect a full set of peripherals using a hub (ethernet adaptor, keyboard, mouse...)
• DVI-D digital output and S-VIDEO (NTSC/PAL) output
• Stereo audio output and microphone input
• Multiplexable expansion header: I2C, SPI, UART, GPIO, SD/MMC
• Typical power consumption: 2W (at 5V)

3.3 Processor

The BeagleBoard-xM[2] processor is the DM3730CBP 1GHz version and comes in a 0.4mm pitch POP package. POP (Package on Package) is a technique where the memory is mounted on top of the processor.

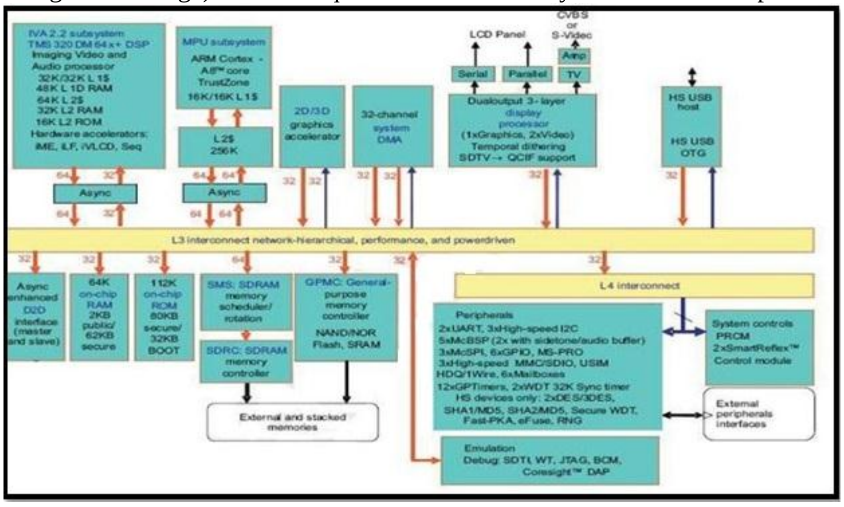
IVA 2.2 subsystem TMS 320 DM 64x+ DSP Imaging Video and Audio processor 32K/32K L1$ 48K L1D RAM 64K L2$ 32K L2 RAM 16K L2 ROM Hardware accelerators: iME, iLF, iVLCD, Seq
MPU subsystem ARM Cortex - A8™ core TrustZone 16K/16K L1$
L2$ 256K
2D/3D graphics accelerator
32-channel system DMA
LCD Panel
CVBS or S-Video
Amp
Serial
Parallel
TV
Dualoutput 3-layer display processor (1xGraphics, 2xVideo) Temporal dithering SDTV→ QCIF support
HS USB host
HS USB OTG
Async
Async
L3 interconnect network-hierarchical, performance, and powerdriven
Async enhanced D2D interface (master and slave)
64K on-chip RAM 2KB public/ 62KB secure
112K on-chip ROM 80KB secure/ 32KB BOOT
SMS: SDRAM memory scheduler/ rotation
SDRC: SDRAM memory controller
GPMC: General-purpose memory controller NAND/NOR Flash, SRAM
External and stacked memories
L4 interconnect
Peripherals 2xUART, 3xHigh-speed I2C 5xMcBSP (2x with sidetone/audio buffer) 3xMcSPI, 6xGPIO, MS-PRO 3xHigh-speed MMC/SDIO, USIM HDQ/1Wire, 6xMailboxes 12xGPTimers, 2xWDT 32K Sync timer HS devices only: 2xDES/3DES, SHA1/MD5, SHA2/MD5, Secure WDT, Fast-PKA, eFuse, RNG
System controls PRCM 2xSmartReflex™ Control module
External peripherals interfaces
Emulation Debug: SDTI, WT, JTAG, BCM, Coresight™ DAP
64
32
64
32
64
64
64
64
32
32
32
32
32
32
32
32
32
32
64
32
32
32

Figure 3.3 DM37x Block diagram

3.4 Overview

The DM3730 is a high-performance, multimedia application device and is integrated onto TI's advanced 45-nm process technology. The processor architecture is configured with different sets of features in different tier devices. Some features are not available in the lower-tier devices. The architecture is designed to provide best-in-class video, image, and graphics processing sufficient to various applications.
The processor supports high-level operating systems (OSs), such as:

- Windows CE
- Linux
- QNX

- Symbian
- Others

IV. Software Implementation

All features have to be managed by an operating system in order to use them in an efficient way. The best option for BB is Linux, mainly because there is an active project supporting BB which includes working drivers for the features listed above, a good tool chain (GCC based) and many common packages ready to build and install into BB.
Linux was originally developed as a free operating system for Intel x86-based personal computers. It has since been ported to more computer hardware platforms than any other operating system. It is a leading operating system on servers and other big iron systems such as mainframe computers and supercomputers: more than 90% of today's 500 supercomputer run some variant of Linux, including the 10 fastest. Linux also runs on embedded systems (devices where the operating system is typically built into the firmware and highly television and video game consoles: the Android system in wide use on mobile devices is built on the Linux tailored to the system) such as mobile phones, tablet computers, network routers, building automation controls, kernel.

4.1 Ubuntu

It is a most popular Linux desktop system and also it is a well-understood with extremely large community .Ubuntu is a Linux operating system. This is based on Debian Linux Distribution. The operating system is an open source and is freely available. It uses its own desktop environment.

4.2 OpenCV

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision, developed by Intel, and now supported by Willow Garage and Itseez. This software mainly aims towards real time implementation of an algorithm in computer vision.
It is free for use under the open source BSD license. The library is cross platform. The library is written in C and C++ and runs under Linux, Windows and Mac OS X. There is active development on interfaces for Python, Ruby, Matlab, and other languages. OpenCV was designed for computational efficiency and with a strong focus on real-time applications [3].

V. Results

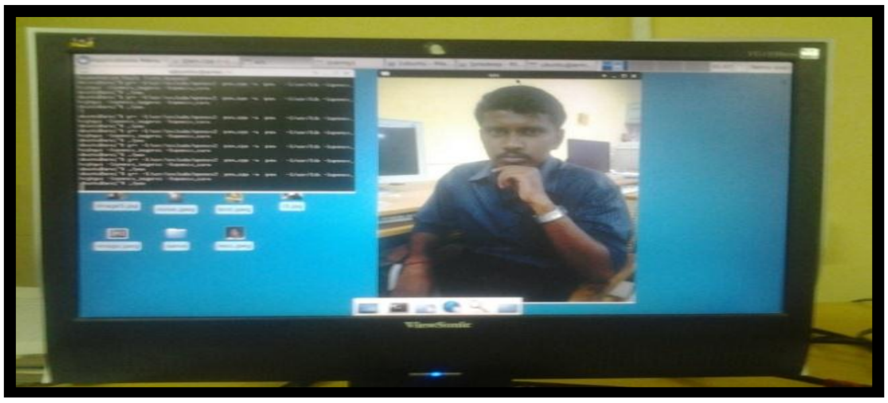
Figure 5.1 Ubuntu Window showing the real time captured colour Image

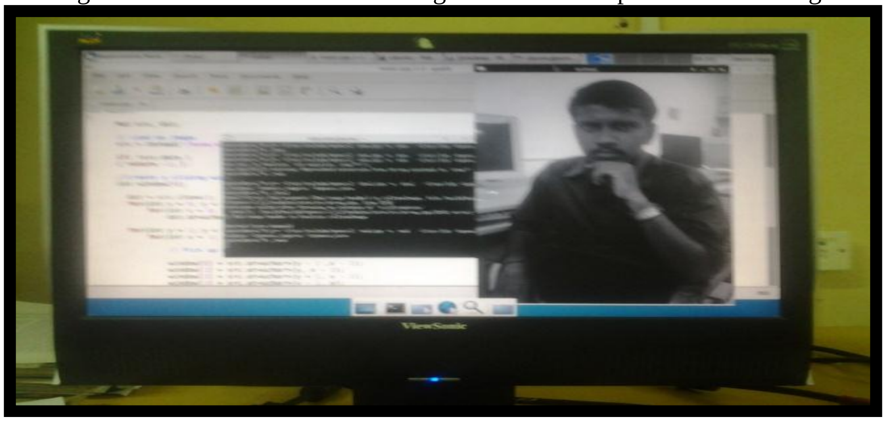
Figure 5.2 Ubuntu window showing the conversion of colour image into gray scale Image.

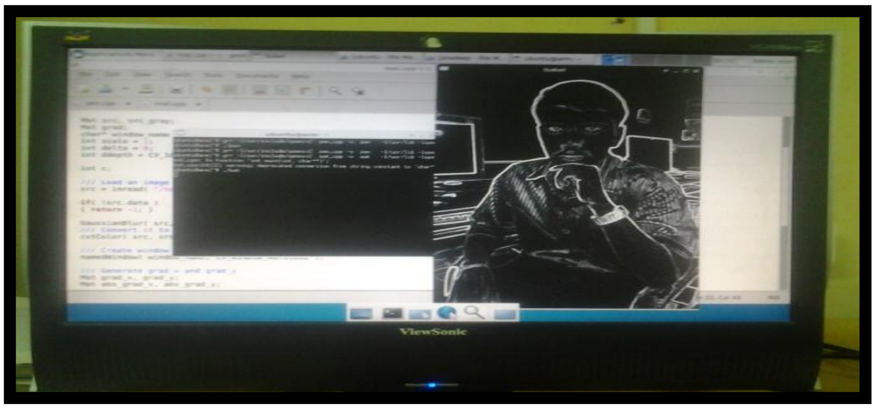

Figure 5.3 Ubuntu window showing the resultant sobel edge Detected image.

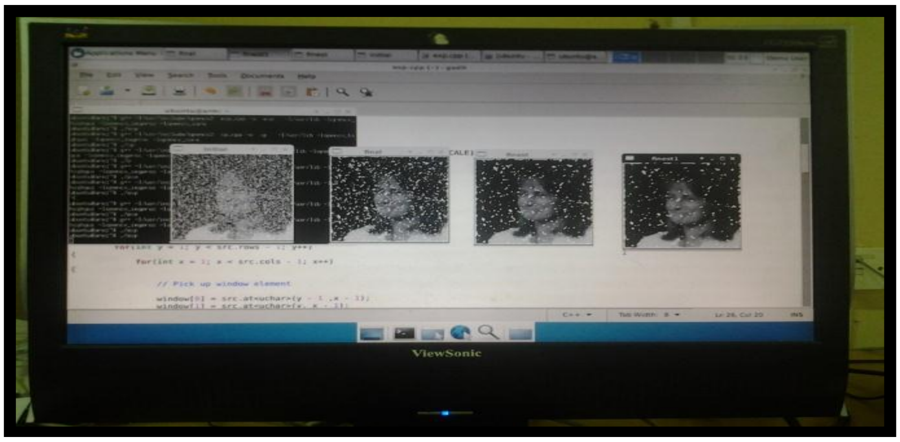

Figure 5.4 Ubuntu windows showing the resultant images after performing the median filter operation repeatedly.

VI. Conclusions

The goal of this research paper is to study the feasibility of using a low cost microcontroller board in Computer Vision application. The low cost microcontrollers Beagleboard-xM will be used as a controlling unit for the implementation and demonstration of few real-time computer vision algorithms like Edge detection and Noise reduction by Median Filter. This attempt can be extended to applications such as reduce distortion in image of Micro Air Vehicles (MAV) and object tracking system.

References

[1] BeagleBoard system Reference Manual, http://beagleboard.org, Oct.2009.

[2] BeagleBoard-xM Rev C system Reference Manual,Revision 1.0,April 4, 2010 Oct.

[3] G. Bradski and A. Kaehler*, "Learning openCV", O'Reilly Publications,*2008.

[4] R Radke, *"A survey of distributed computer vision algorithms",in Handbook of Ambient Intelligence and Smart Environments. (New York:* Springer-Verlag 2009, pp. 35-55).

[5] PramodPoudel and MukulShivaikar,*"Optimization of Computer Vision Algorithm for Real Time Platform", South Eastern Symposium on System Theory,* Tyler, TX, USA, March 7-9,2010.

[6] George Bebis, Member, IEEE, Dwight Egbert, Senior Member, IEEE, and Mubarak Shah, Fellow, IEEE*"Review of Computer Vision Education" IEEE transactions on education,* VOL. 46, NO. 1, February 2003.

[7] Robert, Mario &Maltoni, Md. S. Bhuiyan, Y. Iwahori,and A. Iwata.*"Optimal Edge detection under difficult imaging conditions".* Lecture Notes in computer Science, 1997,vol.1352/1997,pp.25-32.

[8] Raymond H.Chan, Chung-WaHo, and Mila Nikolova(2005*).'Salt- and –Pepper Noise Removal by Median-Type Noise Detectors and Detail-Preserving Regularization'. IEEE Trans.on Image Processing,* Vol.14,No.10, pp.1479-1485.

[9] P. Kahn*, "Building Blocks for Computer Vision Systems"*, IEEE Expert, vol. 8, no. 6, pp. 40–50, Dec. 1993.