IMAGE PROCESSING ALGORITHMS FOR ROBOTICS ON EMBEDDED SYSTEM

[1]SUDHIR D. ZAWARE, [2]PRAJWAL G. AWADE, [3]CHINMAY A. JOSHI, [4]R. V. TORNEKAR

[1,2,3,4]Department of Electronic, Vishwakarma Institute of Technology (Affiliated to Pune University), Pune, Maharashtra, India
E-mail: sudhirdzaware@gmail.com, prajwal.awade@rediffmail.com, chinmay1009@gmail.com

Abstract- Vision is the most versatile and efficient sensory system. So, it is not surprising that images contribute an important role in human perception. This is analogous to machine vision such as shape recognition application which is an important field nowadays. This paper describes implementation of image processing on embedded platform and an embedded application, a robot capable of tracking an object in 3-dimensional environment. It is a real time operating system (RTOS) based embedded system which will run the Digital Image Processing Algorithms to extract the information from the images. The camera connected on USB bus is used to capture images on the ARM9 core running RTOS. Depending upon the information extracted, the locomotion is carried out. The camera is a simple CMOS USB-camera module which has a resolution about 0.3MP. Video4Linux API's provided by kernel are used to capture the image, and then it is decoded, and the required object location is detected using image processing algorithms. The actuations are made so as to track the object. The embedded Linux kernel provides support for multitasking and ensures that the task is performed within the real time constraints. The OS makes system flexible for changes such as interfacing new devices, handling the file system and memory management for storage of data.

Keywords- Embedded Linux, ARM, Video4Linux, YUYV, Embedded C, Object detection, CMOS, USB, SOC, Kerne

I. INTRODUCTION

Object recognition is an important part of the Machine Vision field. The problem of object detection can be seen as a classification problem, where we need to distinguish between the object of interest and any other object. With the advent of System On-Chip (SOC) technology it is now possible to integrate complex hardware functionality on a single chip. For such applications, the software part is usually the most performance critical factor. For this reason it is important that such components are highly optimized in terms of speed while using a relatively small section of the available embedded memory. The object recognition and manipulation algorithms are characterized by their computational complexity due to the size of both the image and the source system added to the large number of complex arithmetic operations. It is extremely desirable that such applications are performed on a standard SOC embedded processor without the need for large and expensive memories and co−processors. The main aim of this paper is therefore to show the feasibility of implementing object recognition algorithms on a standard SOC target which provides the advantage of flexibility in addition to that of real time speed.

II. LITERATURE REVIEW

"Implementation of Image Processing Algorithms & Applications On Embedded Platform" [1], by Debaleena Chattopadhyay of West Bengal University of Technology Kolkata, India, gives overview of experience of implementing the basic imaging algorithms on the embedded platform. The project uses DSP processor TMS320 and its tool-chain as embedded platform. DSP processor enables faster processing of the signal processing algorithms. TMS320's software tool-chain xDM and xDAIS makes the interface between input, output, processor and the observation PC easy. The project works on

- Noise Removal
- Edge Detection
- Morphology
- Region Growing
- Connected Component Analysis
- Histogram Equalization

One of the challenges of object detection is detects an object with a low false detection probability and at sufficiently high rate. Object detection using ARM9 is very fast in detection process, however, the performance on conventional general purpose processors is only 4 frames for second.

Hence using ARM9 it meets hard real time constraints that are imposed in embedded environments, with low power and performance trade-offs. Object detection applications are associated with real-time performance constraints that originate from the embedded system that they are often deployed in.[2] Embedded system using the ARM 32-bit micro-controller has the feature of image/video processing by using the various features and classification algorithms have been proposed for object detection. "Enhanced image detection on an ARM based embedded system."[3], by Evans, Jonathan R., and Tughrul Arslan, presents a new technique for the detection of ICs within images of PCBs which uses region growing. The application of region growing to the detection of ICs is a novel approach. This limitation is the fact that only ICs within a restricted band of grey scale intensities are detected. The Hough transform is of order O(n6) compared to region growing and find IC algorithms which have complexity O(n). It has presented the porting of the techniques to an ARM7 System on-chip target. The porting of an object recognition operator to this target system is novel and the feasibility of this approach has been shown.

In a paper on real time image processing algorithms we found a suite of algorithms for determining the possible trajectories of an autonomous robot while navigating through obstacles. A paper on Estimating and Reducing the Memory Requirements of Signal Processing Codes for Embedded Systems is helpful in reducing memory footprint of the project. We also referred few books on ARM Processor and Image processing

III. IMPLEMENTATION

A. Hardware Approach

The system proposed in this paper uses the EP9302 SOC which is built around the ARM920T core. The EP9302 is a high-performance, low-power, RISC based single-chip computer built around an ARM920T microprocessor core with a maximum operating clock rate of 200 MHz. Its low power, simple, elegant and fully static design is suitable for cost- and power-sensitive applications. The ARM core operates from a 1.8 V supply, while the I/O operates at 3.3V with power usage between 100 mW and 750 mW. It adopts a new bus architecture known as Advanced Micro controller Bus Architecture (AMBA). The ARM core adopts Harvard architecture with separate instruction and data buses (and caches), significantly increasing its potential speed. The ARM920T processor implements an enhanced ARM architecture v4 MMU to provide translation and access permission checks for instruction and data addresses. The camera used here is the CMOS camera with the resolution of 640x320. The camera is interfaced to the system via USB interface. The USB camera continuously captures the image from live video stream. The captured images are processed by using different image processing algorithms. The images obtained from camera are in the raw format. The L293 based motor driver circuit is used to drive the 200 RPM motors working on 12V supply obtained from the battery. The ARM board as well as motor driver circuit works on 5V regulated supply derived from battery.

B. Working Principle

The main purpose of this system is to detect the object in particular image frame and direction in which the robot should move so as to reach near the object. To detect the object we will use camera which is interfaced to microprocessor. This camera continuously captures frames in the form of stream and it will check for object in that frame. The V4L(Video4Linux) APIs provides with the functionality to control the capturing device. The frame obtained in the main memory (SDRAM) which is then processed using different image processing algorithms. The image is captured using USB webcam. The image is stored in RAM as there is no need to store image permanently. The pseudo-algorithm of image processing is as follows- i. Capture the image frame ii. Decode the image by extracting Y-component from YUYV format.

iii. Convert the extracted Y-component into proper image frame

iv. Negate the image

v. Filter the image to remove noise

vi. Extract the object from frame

vii. Calculate the object location

viii. Measure the angular distance between the object and the orientation of the robot.

After the detection of object, the direction in which robot should move is calculated depending upon the location of object in the frame. Then depending upon the direction of movement, the motors are actuated to get the movement in the desired direction.

C. Software Approach

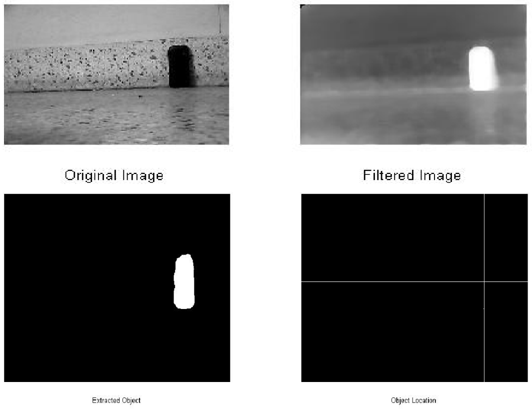
The application code is written completely in Embedded C language which contains code for initializing devices, capturing image, object detection and locomotion related code. The ARM is running embedded Linux kernel v2.6.20 hence the cross compiler arm-linux-gcc is used to compile the code into binary file. On boot the board provides console over serial port using the Minicom package on Desktop System. The code is transferred to the board using TFTP protocol over the Ethernet to the SDRAM. The code is then run from the local memory by proving command over serial console.

IV. COST ESTIMATE

The estimated cost of the project is around 10,000INR

V. RESULTS

The left side figure shows the gray scale image and the right side figure shows the filtered image.



Original Image

Filtered Image

Extracted Object

Object Location

The left side figure above shows the object extracted from the image and in the right side figure the intersection of two lines shows the location of required object in the image frame.

CONCLUSION

The use of system on chip (SOC) makes the application portable. A simple image processing algorithms on embedded platform, i.e. an ARM9, is implemented to detect and track desired object. There is trade-off between processing time and quality precision of the output image. Further analysis on performance improvement related to factors, like execution time and accuracy is under study. The comparative study regarding performance parameters in MATLAB and Embedded C environments is also observed. The efficiency of system can be improved by using 64-bit or 128-bit processor. The speed of operation can be improved by use of higher system clock or parallel processing hardware. The project led us to the conclusion that in order to get desired output with optimum time and memory complexity, we have to compromise with the quality and precision of output. Image processing often deals with mathematical processing of integer values. The floating point processing is not much required. Extensive mathematical calculations such as additions, subtractions, multiplications, etc. on image pixel values increase the time complexity. Embedded platform support must also be checked while selecting camera so as have hardware compatibility. Thus, camera device selection is crucial.

REFERENCES

[1] Debaleena Chattopadhyay, “ Implementation of Image Processing Algorithms & Applications On Embedded Platform”, West Bengal University of Technology Kolkata, India.

[2] B. Heisele, T. Serre, S. Prentice and T. Poggio, ”Hierarchical Classification and Feature Reduction for Fast Face Detection with Support Vector Machines pattern Recognition”, vol. 36,no.9, pp. 2007-2017, 2003.

[3] J.R.Evans and T.Arslan,, “Enhanced Image Detection on an ARM based Embedded System”, University of Edinburgh , Department of Electronics and Electrical EngineeringKing’s Buildings , Mayfield Road , Edinburgh EH9 3JL , Scotland, UK

[4] Ilas, C., S. Paturca, and M. Ilas. "Real-time image processing algorithms for object and distances identification in mobile robot trajectory planning." Journal of Control Engineering and Applied Informatics 13.2 (2011): 32-37.

[5] Sloss, Andrew, Dominic Symes, and Chris Wright. ARM System Developer's Guide: Designing and Optimizing System Software. Morgan Kaufmann, 2004.

[6] Gonzalez, Rafael C., Richard E. Woods, and Steven L. Eddins. Digital image processing using MATLAB. Vol. 2. Knoxville: Gatesmark Publishing, 2009.

[7] Ramanujam, J., et al. "Estimating and reducing the memory requirements of signal processing codes for embedded systems." Signal Processing, IEEE Transactions on 54.1 (2006): 286-294.

[8] Hallinan, Christopher. Embedded Linux primer: a practical, real-world approach. Pearson Education India, 2007.

[9] LTD, ARM. "ARM920T technical reference manual." ARM Ltd.

[10] EP9302 Reference manual.

★★★