Motion Detection and Its Localization Using CMU cam4

Shraddha Moon[#1]**, Trushna Deotale** [#2]

Shraddha Moon, M-tech student Department of Electronics Engineering
G. H. Raisoni College of Engineering, Nagpur

Trushna Deotale, Assistant Professor Department of Electronics Engineering
G. H. Raisoni College of Engineering, Nagpur

ABSTRACT

Generally fast computers are needed to capture and process camera images. It is also necessary to write the software to perform this processing. Because of this it is difficult to use vision as a sensor in simple systems. The CMUcam4 vision system uses a fast low cost microcontroller to handle all of the high speed processing of the camera data and contains software to perform simple vision tasks. Because the user can choose to output only low bandwidth high level information from the vision system, like the red object is at position X-Y, it is possible for a simple processor like an Arduino microcontroller to read this data and direct a small robot in tasks like chasing a colored ball. The CMUcam4 vision system makes it possible to ignore the complexity of camera interfacing and use vision just like any other sensor (i.e. sonar) often used in robotic systems. Additionally, with new TV output functionality, you can see what the CMUcam4 sees on the TV to verify that everything is working! Hence proposed system is designed to detect a motion on the basis of difference between current and previous frame. While finding the difference system will indicate the location or the position of the motions. System will also capable to detect motion in dark by reducing the infrared filtering from CMOS sensor. The system will then give alert to user through light operating or buzzing the sound.

Key words: CMOS sensors, CMUcam4, Arduino

.

Corresponding Author: S P.Moon

I. INTRODUCTION

In market-driven applications such as surveillance, automotive, and machine vision, there is an increased demand for imaging systems with real-time processing capabilities. In some cases, these specific requirements are quite hard to be fulfilled through a conventional approach, consisting of a standard charge-coupled device or complementary metal–oxide–semiconductor (CMOS) camera linked to a digital signal-processing platform. These systems are typically based on general purpose architectures, performing real-time image processing. Although their high computational power and high flexibility are satisfactory for many applications, there are some low-level images processing tasks that can be efficiently executed using ad-hoc image processing capabilities embedded directly in the imager. Thanks to the great advantages of CMOS sub micrometer technology, allowing even smaller device feature size, some recent CMOS image sensors with integrated signal processing have been developed, following two main approaches: pixel-level and array-level processing.

Problem definition**:**

System is useful for security purpose as the system is performing motion detection and its quadrant wise analysis. System work in dark area and if there is detection of motion then the alert system will give the user information. Hence proposed system is design to detect a motion on the basis of difference between current and previous frame. Also the system will then give alert to user through light operating or buzzing the sound.

II. RESEARCH METHODOLOGY TO BE EMPLOYED

Proposed system is mainly divided in two following modules. First is video capturing to get the video frames, next is image processing to get the images from frame, and next is to get the pixel information from the image, the detection of color from pixel and at last controlling the hardware.

A. Image acquisition

A digital image is produced by one or several image sensors, which, besides various types of light-sensitive cameras, include range sensors, tomography devices, radar, ultra-sonic cameras, etc. Depending on the type of sensor, the resulting image data is an ordinary 2D image, a 3D volume, or an image sequence. The pixel values typically correspond to light intensity in one or several spectral bands (gray images or color images), but can also be related to various physical measures, such as depth, absorption or reflectance of sonic or electromagnetic waves, or nuclear magnetic resonance.

B. Pre-processing

Before a computer vision method can be applied to image data in order to extract some specific piece of information, it is usually necessary to process the data in order to assure that it satisfies certain assumptions implied by the method. Examples are Re-sampling in order to assure that the image coordinate system is correct. Noise reduction in order to assure that sensor noise does not introduce false information. Contrast enhancement to assure that relevant information can be detected. Scale-space representation to enhance image structures at locally appropriate scales.

C. Feature extraction

Image features at various levels of complexity are extracted from the image data. Typical examples of such features are Lines, edges and ridges. Localized interest points such as corners, blobs or points.

D. Detection/segmentation

At some point in the processing a decision is made about which image points or regions of the image are relevant for further processing. Examples are Selection of a specific set of interest points Segmentation of one or multiple image regions which contain a specific object of interest.

E.High-level processing

At this step the input is typically a small set of data, for example a set of points or an image region which is assumed to contain a specific object. The remaining processing deals with, for example: Verification that the data satisfy model-based and application specific assumptions. Estimation of application specific parameters, such as object poses or objects size. Classifying a detected object into different categories.

F. Hardware controlling

It is a microcontroller based device control system where connected appliances can be operated using digital logic 1 or 0. In proposed system emergency alarm or the lighting system can be activated.

III. PROPOSED MODEL

Figure 1 describes the proposed system architecture where the CMOS sensor will be used to capture the frame at maximum possible speed. These frames are then transferred to central processor as array of data which will always get store at buffer of exact size of captured frame. This buffer will be utilized by the processing unit to find out any changes in current captured view with respect to buffer image. If there are any dissimilarity between buffered image and current captured image more that the defined threshold values then the system will generate the alert signals [5]. Defining the threshold value is compulsory CMOS sensor result are environmental changes dependent so system will never get exact result even after the camera and the scene is steady. Design a CMOS module interfacing with controller board system. Develop a software program to capture a camera view and detect motion in frames. Build microcontroller based hardware in order to ON the emergency alert system.

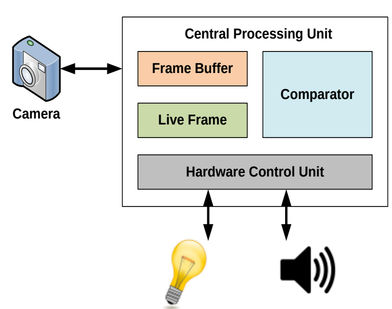

Fig 1 Block diagram of proposed system

Proposed system is mainly divided in two following modules. First is video capturing to get the video frames, next is image processing to get the images from frame, and next is to get the pixel information from the image, the detection of color from pixel and at last controlling the hardware.

IV. Requirement analysis

Fig 2 CMUcam4 v1.02 Firmware

The CMUcam4 is a fully programmable embedded computer vision sensor. The main processor is the Parallax P8X32A (Propeller Chip) connected to an Omni Vision 9665 CMOS camera sensor module.

A. Programming advantages

The object-based high-level Spin language is easy to learn, with special commands that allow developers to quickly exploit the Propeller chip’s unique and powerful features. Propeller Assembly instructions provide conditional execution and optional flag and result writing for each individual instruction. This makes critical, multi-decision blocks of code more consistently timed; event handlers are less prone to jitter and developers spend less time padding, or squeezing, cycles.

B. Applications

The Propeller chip is particularly useful in projects that can be vastly simplified with simultaneous processing, including: Industrial control systems, Sensor integration, signal processing, and data acquisition, Handheld portable human-interface terminals, Motor and actuator control, User interfaces requiring NTSC, PAL, or VGA output, with PS/2 keyboard and mouse input, Low-cost video game systems, Industrial, educational or personal-use robotics, Wireless video transmission (NTSC or PAL).

C. Programming platform support

The Propeller Demo Board convenient means to test-drive the Propeller chip's varied capabilities through a host of device Interfaces on one compact board. Main features are P8X32A-Q44 Propeller Chip, 24LC256-I/ST EEPROM for program storage, Replaceable 5.000 MHz crystal, 3.3 V and 5 V regulators with on/off switch, USB-to-serial interface for programming Communication, VGA and TV output, Stereo output with 16 Ω headphone amplifier, Electret microphone input, Two PS/2 mouse and keyboard I/O connectors, 8 LEDs (share VGA pins), Pushbutton for reset, Big ground post for scope hookup, I/O pins P0-P7 are free and brought out to header, Breadboard for custom circuits.

D. Propeller tool software

The Propeller Tool Software is the primary development environment for Propeller programming in Spin and Assembly Language. It includes many features to facilitate organized development of object-based applications: multi-file editing, code and document comments, color-coded blocks, keyword highlighting, and multiple window and monitor support aid in rapid code development. Optional view modes allow you to quickly drill down to the information you need—by hiding comment lines, method bodies, or by showing the object's compiled documentation only. Example objects, such as keyboard, mouse, and graphics drivers, come standard with the free Propeller Tool software.

D.1 Propellent library and executable

The Parallax Propellent software is a Windows-based tool for compiling and downloading to the Parallax Propeller chip—without using the Propeller Tool development software. The Propellent Executable provides the ability to do things like compile Spin source, save it as a binary or EEPROM image, identify a connected Propeller chip, and download to the Propeller chip, all via simple command-line switches or drag-and-drop operations.

D.2 Propeller GCC

The Propeller GCC Compiler tool-chain is an open-source, multi-OS, and multi-lingual compiler that targets the Parallax Propeller's unique multi core architecture. Parallax has collaborated with industry experts to develop all aspects of the tool chain, including the creation of a new development environment that simplifies writing code, compilation, and downloading to a Propeller board. Using the Large Memory Model (LMM) and Extended Memory Model (XMM) gives the developer the ability to write C or C++ programs that run faster than Spin or exceed Spin's 32 KB program size limit, respectively. Example objects, including C objects, are available through the Propeller Object Exchange.

D.3 Introduction of QT creator

The motion detection on CMU cam4 kit is display on the graphical user interface. This GUI is coded in Qt designer._Qt Creator provides a cross-platform, complete integrated development environment (IDE) for application developers to create applications for multiple desktop and mobile device platforms. It is available for Linux, Mac OS X and Windows operating systems.Qt Creator is an intuitive, modern cross platform Integrated Development Environment that enables developers to create graphically appealing applications for desktop, embedded, and mobile devices. It also provides many unique features that make Qt Creator easier.

D.4 Features QT creator

C++ and JavaScript code editor. Integrated UI designer. Project and build management tools. Support for version control. Simulator for mobile UIs.Support for desktop and mobile targets

V. RELATED WORK

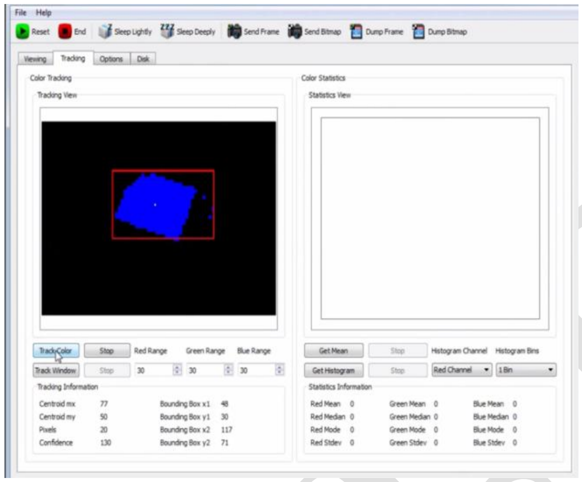

Fig 3 Tacking moving objects in GUI

By using CMU cam4 GUI it is possible to no, off the device and also perform various filtering automatically, also using CMU cam4 GUI it is possible to control the servo motor at list two one in horizontally and another in vertically so that the CMU cam 4 kit can rotated in all direction thus help for tracking purpose.

VI. CONCLUSION

It is possible to make a low-power camera based motion detection system with the help of CMU cam4 kit. With enough time and testing, an advanced motion detection algorithm could be designed that minimizes power consumption while detecting motion easily. More advanced cameras with higher resolutions and better processors in the future will further this effort to create a truly smart sensor that knows when a room is occupied and when a room is not occupied

VII. REFERENCES

[1] Bo Zhao, Student Member, IEEE, Xiangyu Zhang, Student Member, IEEE, Shoushun Chen, Member, IEEE,Kay-Soon Low, Senior Member, IEEE, and Hualiang Zhuang “A 64 × 64 CMOS Image Sensor With On-Chip” IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 22, NO. 4, APRIL 2012
Moving Object Detection and Localization
[2] A. Blumberg, L. Keeler, and A. Shelat, “Automated traffic enforcement which respects ‘driver privacy’,” in Proc. IEEE ITSC, Sep. 2005, pp.941–946.
[3] Y. Cho, S. O. Lim, and H. S. Yang, “Collaborative occupancy reasoningin visual sensor network for scalable smart video surveillance,” IEEETrans. Consumer Electron., vol. 56, no. 3, pp. 1997–2003, Aug. 2010.
[4] Y.-J. Wu, F.-L. Lian, and T.-H. Chang, “Traffic monitoring and vehicle tracking using roadside cameras,” in Proc. IEEE Int. Conf. SMC, vol. 6.Oct. 2006, pp. 4631–4636.

[5] H. Zhuang, K.-S. Low, and W.-Y. Yau, “On-chip pulse based parallel neural circuits for object-tracking system,” IET Electron. Lett., vol. 46, no. 9, pp. 614–616, Apr. 2010.
[6] M. Gottardi, N. Massari, and S. Jawed, “A 100μW 128 × 64 pixels contrast-based asynchronous binary vision sensor for sensor networks applications,” IEEE J. Solid-State Circuits, vol. 44, no. 5, pp. 1582–1592, May 2009.
[7] E. Artyomov and O. Yadid-Pecht, “Adaptive multiple-resolution CMOS active pixel sensor,” IEEE Trans. Circuits Syst. I: Regular Papers, vol. 53, no. 10, pp. 2178–2186, Oct. 2006.
[8] C. Stauffer, W.E.L. Grimson, “Adaptive background mixture models for real-time tracking”, published.
[9] M. Piccardi, “Background subtraction techniques: a review”, unpublished.
[10] A. Verdant, A. Dupret, H. Mathias, P. Villar, L. Lacassagne, “Low Power Motion Detection with Low Spatial and Temporal Resolution for CMOS Image Sensor”, published.
[11] K. Agyeman, “CMUcam4: Open Source Programmable Embedded Color Vision Sensor”, unpublished.
[12] A. Teman, S. Fisher, L. Sudakov, A. Fish, and O. Yadid-Pecht, “Autonomous CMOS image sensor for real time target detection and tracking,” in Proc. IEEE ISCAS, May 2008, pp. 2138–2141.
[13] A. Fish, L. Sudakov-Boresha, and O. Yadid-Pecht, “Low-power tracking image sensor based on biological models of attention,” Int. J. Inform.Theory Applicat., vol. 14, no. 2, pp. 103–114, 2006.
[14] S. Chen, F. Boussaid, and A. Bermak, “Robust intermediate read-out for deep submicron technology CMOS image sensors,” IEEE SensorsJ., vol. 8, no. 3, pp. 286–294, Mar. 2008.
[15] S. Chen, A. Bermak, and Y. Wang, “A CMOS image sensor with onchip image compression based on predictive boundary adaptation and memoryless QTD algorithm,” IEEE Trans. Very Large Scale Integr. Syst., vol. 19, no. 4, pp. 538–547, Apr. 2011.
[16] Y. Chi, U. Mallik, M. Clapp, E. Choi, G. Cauwenberghs, and R. Etienne-Cummings, “CMOS camera with in-pixel temporal change detection and ADC,” IEEE J. Solid-State Circuits, vol. 42, no. 10, pp. 2187–2196, Oct.2007.
[17] S. Mizuno, K. Fujita, H. Yamamoto, N. Mukozaka, and H. Toyoda, “A 256 × 256 compact CMOS image sensor with on-chip motion detection function,” IEEE J. Solid-State Circuits, vol. 38, no. 6, pp. 1072–1075,Jun. 2003.
[18] P. Lichtsteiner, C. Posch, and T. Delbruck, “A 128 × 128 120 dB 15μs latency asynchronous temporal contrast vision sensor,” IEEE J. Solid-State Circuits, vol. 43, no. 2, pp. 566–576, Feb. 2008.
[19] J. Choi, S.-W. Han, S.-J. Kim, S.-I. Chang, and E. Yoon, “A spatialtemporal multiresolution CMOS image sensor with adaptive frame rates for tracking the moving objects in region-of-interest and suppressing motion blur,” IEEE J. Solid-State Circuits, vol. 42, no. 12, pp. 2978–2989, Dec. 2007.
[20] M. Litzenberger, C. Posch, D. Bauer, A. Belbachir, P. Schon, B. Kohn, and H. Garn, “Embedded vision system for real-time object tracking using an asynchronous transient vision sensor,” in Proc. Digit. Signal Process. Workshop, Sep. 2006, pp. 173–178.
[21] M. Litzenberger, A. Belbachir, N. Donath, G. Gritsch, H. Garn, B. Kohn,C. Posch, and S. Schraml, “Estimation of vehicle speed based on asynchronous data from a silicon retina optical sensor,” in Proc. IEEE ITSC, Sep. 2006, pp. 653–658.

[22] S. Chen, W. Tang, and E. Culurciello, “A 64 × 64 pixels UWB wireless temporal-difference digital image sensor,” in Proc. IEEE ISCAS, Jun. 2010, pp. 1404–1407.
[23] H. Su and F.-G. Huang, “Human gait recognition based on motion analysis,” in Proc. Int. Conf. Mach. Learning Cybern., vol. 7. Aug. 2005, pp. 4464–4468.
[24] J. Triesch and C. von der Malsburg, “A system for person-independent hand posture recognition against complex backgrounds,” IEEE Trans.Pattern Anal. Mach. Intell., vol. 23, no. 12, pp. 1449–1453, Dec. 2001.