Development of Adaptive Image Processing based on FPGA for Real-time Robot Vision System

HAYATO HAGIWARA* Non-member, KENICHI ASAMI* Member
MOCHIMITSU KOMORI* Non-member

(Received May 20, 2013, revised August 10, 2013)

Abstract: This paper presents a real-time image processing system for a mobile service robot using evolutionary computation on FPGA. The digital image processing acquired by a CMOS image sensor is performed on the embedded FPGA board and Linux-based real-time video communication module. The image processing and evolutionary computation are implemented into a reconfigurable device as original logic. The image processing circuit captures digital images from a CMOS camera, and extracts the edge and feature points in the images for self-localization to adjust the current course on the traveling map data. Further, this image processing application enables the robot vision system to turn the visual field to the target direction where an object is recognized by a specific color. The genetic algorithm optimizes threshold values for the filtering operations according to various lighting environments. The integrated robot vision system aims to achieve a suitable platform available from both hardware and software.

Keywords: Robot vision system, Mobile robots, FPGA, Evolutionary robotics, Genetic algorithm, Real-time Linux

1. Introduction

The robot industry is one of the most promising markets for Japan’s future. There are predictions which indicate expansion of domestic production by 2035 to about ten times the current scale. Autonomous mobile robots for quality of life improvement need the spatial recognition capability and various detective functions to carry out their work [1]–[3]. In Japan’s aging society, with fewer children in the future, autonomous mobile robots are expected to be useful for security applications such as surveillance [4] [5]. Autonomous mobile robots need to recognize objects and their surroundings using image processing combined with sensors and transducers [6]. Robot vision systems require real-time response [7] and low electric power consumption for accurate and sustained activity to accomplish their tasks. Therefore, common image processing for application software such as outline and edge detection should be included on the hardware [8]–[10]. Effective recognition and use of visual information processing would increase the autonomy and ability of mobile service robots. The adequate integration of hardware and software for image processing is designed for the robot vision system.

In this paper, a real-time robot vision system using a networked FPGA board for image processing and evolutionary computation for the visual navigation system and real-time extension of Linux OS is introduced. The image capture circuit is made using hardware description language on the FPGA. The stepper motor control circuit is integrated with the image recognition. A CMOS image sensor is connected to I/O ports of the FPGA device in order to directly manipulate the image data from the image processing circuits. Traditional image processing detectors such as Sobel gradient operator and Moravec interest point operator [11]–[12] are implemented as concurrent circuits on the FPGA device. Evolutionary computation as an optimization method is used to decide suitable thresholds for image processing operators, where object detection in varying environments using genetic algorithms improves the adaptive obstacle avoidance and course recognition for autonomous mobile robots. In addition, dedicated Linux OS with hard real-time extension manages the peripheral devices on the FPGA. This offers integration of hardware and software for image processing that can be utilized for robot vision applications.

2. System Configuration

The real-time robot vision system is composed of the head, with an FPGA board, a CMOS camera module and a wireless LAN module (Fig. 1), and the base, with four DC motors with omni-directional wheels and motor drivers. The robot has a height of 660 mm and a horizontal diameter of 300 mm (Fig. 2).

The FPGA board SUZAKU-V [13] as an embedded Linux running environment is installed with a Xilinx Virtex-4 FX [14] device that can have user logic circuits together with a PowerPC 405 processor at 350 MHz as a hard core in
Development of Adaptive Image Processing based on FPGA for Real-time Robot Vision System

3. Image Processing System

Image processing systems for robot vision based on PC and USB interface usually use multithread software applications, so hard real-time response can be difficult for the integration of image recognition and behavioral usage. Therefore, this robot vision system uses image processing by FPGA directly connected to the CMOS image sensor.

We used a Xilinx ISE development environment which supports designing processes for embedded devices and enables the developers to implement specific logic circuits using abstract hardware description language such as VHDL and Verilog HDL. Xilinx EDK integrates the development for embedded systems including processors with original IP cores that enable user-specific function.

The image capture circuit for this robot vision system supplies a MCLK (master clock) signal to the CMOS camera module as input, which is generated from a DCM (digital clock manager) module. The CMOS camera module responds with a PCLK (pixel clock) signal to the output terminal, and serves PDATA (pixel data) in 8 bits as sequential image data together with HSYNC (horizontal synchronization) and VSYNC (vertical synchronization) signals. When the maximum speed of the MCLK is set at 27 MHz, the CMOS camera module outputs image data at 30 fps. The image capture circuit also reduces image size for moving object recognition, where a BRAM (block RAM) plays the role of buffer. The default pixel format of YUV422 at 640 x 480 VGA resolutions is produced in the PDATA signals. The YUV pixel format reduces image data size compared to RGB format and is useful for recognition of objects by color without being influenced by brightness. The YUV model defines a visual representation for one luma (Y, brightness) and two chrominance (U, V, color difference of blue and red) components. The YUV422 format gives luma (Y) for each pixel in 1 byte, and shares chromaticity (U, V) for adjacent two pixels in 2 bytes respectively. Therefore, the total amount of data size per pixel is 2 bytes. The visual receptors for human eyes are sensitive to changes in brightness more than in chromaticity, so that the image processing for the robot vision system can achieve efficient object recognition with suitable compressed data.

The block diagram for the image processing circuit shown in Fig. 3 has been developed for the robot vision system. The image processing circuit includes a state machine circuit and a FIFO memory buffer that controls the timing for pixel data transmission to the embedded processor. The image capture circuit of this robot vision system receives one pixel data (1 byte) at 27 MHz from the CMOS camera module.
era module and collects four pixel data (4 bytes) to deliver to the state machine circuit in compliance with the width size of 32 bits in the FIFO memory buffer. The state machine circuit adjusts timing between the image capturing and the buffering to insert the 4 bytes data for 2 pixels to the FIFO. Therefore, as shown in Fig. 4, there are three states of ‘wait’, ‘write’ and ‘write acknowledge’ for pixel data transmission. The device driver software on the PowerPC processor can read pixel data at a suitable speed. Simultaneously, this image processing circuit detects the moving object in the view by differencing two sequential frames.

4. Image Recognition Application

In this robot vision system, the application software for tracking moving objects using color information obtained from the CMOS camera module has been newly developed. This image processing application enables the robot vision system to turn the visual field to the target direction where the object is recognized by a specific color. The simple and primitive image processing library can be utilized in many ways for the robot vision system.

The method of object recognition by color sets thresholds for pixel components, and all pixels are judged to be the target color. Fig. 5 shows the results for color judgment using RGB and YUV formats respectively. Here the target color is set as orange. The white pixels indicate the target color and the black, non-target. While the method using RGB format cannot recognize the whole colored ball, the thresholds for YUV components are independent of brightness from the lighting. Therefore, the object recognition by color uses YUV format in order to ascertain accurately the region of the target object.
Development of Adaptive Image Processing based on FPGA for Real-time Robot Vision System

(a) Grouping region by deciding RGB.

(b) Grouping region by deciding YUV.

Figure 6: Results of region detection for target object recognition.

In order to eliminate noise around the target object in color detection, region detection methods by segmentation and grouping are applied to the binary image by color detection. The method of region segmentation divides the whole image data into certain small segments and judges whether the segment indicates an effective region where many pixels with the target color exist. The method of region grouping finds clusters of effective regions which are contiguous with each other and determines the biggest cluster with effective regions. Finally, the central coordinates are decided for tracking the target object position. As shown in Fig. 6, more accurate central coordinates can be decided by using YUV format than with RGB format.

5. Visual Navigation

Autonomous mobile robots need spatial recognition capability to correct their position and orientation. Specifically, it is important to recognize the outline of obstacles or the perimeter of floors and walls in order to confirm their proper traveling route. Moreover, it is useful to calculate the deviation from the position and orientation which have already been learned during traveling. In this robot vision system, the application software for image recognition has been developed for detecting passage boundaries between floors and walls in a typical indoor environment (Fig. 7). The methods of edge and line detection are applied to the image data from the CMOS camera module.

The edge detection uses the Sobel gradient operator, and the feature point detection uses the Moravec interest point operator, where the choice of threshold is essential to decide the point extraction in a time-varying environment for a mobile robot. Genetic algorithm (GA) is an effective way to solve an optimization problem [15]. In the visual navigation system, GA is adopted to find proper parameters in the Sobel and Moravec operators. In this method, genetic operations of crossover, mutation, and reproduction are introduced for the model of chromosome, which is coded as a binary number. Fitness function is evaluated to measure the goodness of chromosome. The chromosome with the maximum fitness is kept as a solution in the current generation and copied to the next generation. A new population is obtained by the genetic operators of crossover, mutation, and reproduction. The evolvable hardware circuit is able to achieve image filter automatically and rapidly. FPGA would be a suitable platform to implement image processing and evolutionary computation with high speed, low cost, and low power consumption for visual processing of autonomous mobile robots.

The edge detection extracts border location with intense contrasts preserving important structural properties in an image. The Sobel operator uses a discrete differentiation to compute an approximation of gradient intensity among adjacent pixels. The two $3 \times 3$ kernels are used to calculate the derivatives of $G_x$ for horizontal and $G_y$ for vertical changes by convolving the original image. Therefore, the gradient magnitude can be combined as the following equation:

$$ G = \sqrt{G_x^2 + G_y^2} $$

This filtering procedure is achieved on the FPGA hardware circuit where the result of the gradient magnitude can be calculated at the same time as the image capture. The filter circuit uses BRAM (block RAM) on the Xilinx device for conserving the previous pixel data as line buffers. The FPGA device effectively executes simple and large data processing by using filter circuits for robot vision purposes.

While computing gradient magnitude for all pixels in an image, judgment that the pixel lies on an edge is achieved by thresholding. The adaptive threshold can be determined by GA in a varying light environment. The GA technique is designed for searching an appropriate global maximum in a problem space. The parameter of edge detection threshold is represented by chromosome, and initialization of a population for candidate solutions to the edge detection problem is created at the beginning. For the current population, generative procedures of reproduction, recombination and evaluation are repeated by a certain generation. For the edge detection problem, genetic operators known as crossover and mutation are adopted. Here the number of population is set to 10, the crossover rate is set to 0.8, and the mutation rate is set to 0.1. The method of evaluating individuals uses the continuity of edge at a certain direction in the image. The measurement of edge length with a single point width repre-
sents the fitness of individual in the environment. In reproducing a population, individuals with higher fitness are preferentially selected for genetic operations. Consequently, the optimal individual remains in the final generation. Fig. 8 shows the result image of edge detection by using GA for the original image of a corridor shown in Fig. 7.

In this case, the generation of individuals proceeded to 100, and the fitness increased from 0.57 to 0.81 as shown in Fig. 9. GA can be applied to a wide range of problems in varying real environments. In order to decide a central position and a straight direction of passage on the traveling route, boundaries between the floor and both walls must be detected. The boundary detection algorithm uses Hough transformation to identify the lines as set of continuous edge points in the edge image [16]. The Hough idea is that all lines through coordinate \((x, y)\) can be represented by \((\rho, \theta)\) where \(\rho\) shows the distance between the line and the origin, and \(\theta\) shows the angle of the vector from the origin to this closest point in the equation \(\rho = x \cdot \cos \theta + y \cdot \sin \theta\). All the edge points are transformed to the form of prospective lines \((\rho, \theta)\) and the pair of the parameters accumulates the number of appearances. The pair with the maximum frequency of appearance is found to be a line indicating the strongest feature on straight continuous points. In a typical indoor passage, there must be left and right boundaries as lines with the strong features, when the Robot Vision System moves in the straight direction of passage. Fig. 10 shows the result of boundary detection where the respective ten lines are found on the left and right sides from the edge image. The visual navigation system selects the closest line to this side as the boundary. The boundary detection was verified to be robust for some passing people and obstacles.

The purpose of the visual navigation system consists of local and global guidance in the traveling course for the mobile robot. The local guidance includes collision avoidance and selection of suitable location on the course, where the simultaneous edge information is utilized in dynamic motion. Recognition of objects and surroundings requires real-time processing on FPGA for the robot vision system. On the other hand, the global guidance confirms the important positions on the traveling course, where the similarity of scenery feature is evaluated by interest feature points.
Therefore, the map of this Robot Vision System includes the record of motion and the correspondence to feature values for visual scene. This method of combined local and global guidance enables the mobile robot to reduce the time and effort for making an elaborate measured map, where the visual navigation system fits a simple patrol objective in a repetitious travel route.

The feature point detection defines finding corner points as the end of an edge in this visual navigation system. The Moravec interest point operator uses four directional patches (horizontal, vertical, and on the two diagonals) for calculating the self-similarity for adjacent pixels. A corner point has the lower similarity in the local area. The similarity is calculated by the sum of squared differences (SSD) from pixel brightness I as in the following equation:

$$S = \sum_{i,j} (I(x + i, y + j) - I(x, y))^2$$  \hspace{1cm} (2)

The smallest SSD for four patches represents the strength of interest point. If the value is locally maximal, then a corner point is present. However the Moravec operator has a drawback that it is not possible to detect corner points at the crossing edges. Furthermore, the complicated procedure of finding the local maximum value for the similarity is not suitable for robot vision application. Therefore, the directional restriction of edges and simplified calculation of the similarity for binary corner detection (BCD) are added to the feature point detector. Here BCD judges a terminal point of edge and reduces the calculation of self-similarity to binary edge data. This improved feature point detector requires an adequate edge image as the input, so that the adaptive edge detection by using GA is suitable for the BCD. Fig. 11 shows the result for feature point detection using BCD. Feature point detection is necessary for object tracking and background identification.

The visual navigation system records the distribution of feature points at main places on the traveling route in teaching mode. When the Robot Vision System repeats the movement on the traveling route in patrol mode, the confirmation of the current position and the modification of the deviation from the route are achieved by the navigation system. Recognition of the visual scene at an important place can be done by using the set of feature points, where all the feature points are compared with the coordinates in range and the corresponding ratio indicates the previous visual scene.

6. Real-Time Module

This robot vision system needs to achieve hard real-time response to effective physical movement by image processing and to integrate high-level services such as image communication among user terminals and robots. Therefore, this robot vision system requires a real-time operating system on the embedded system. For this requirement, the hard real-time extension for general GNU/Linux operating system has a suitable capability, because of the real-time multithread scheduler, networking protocol stack, various device drivers, and useful development tools on the open platform for almost all embedded systems [17]. A Xenomai [18] real-time subsystem was ported to the embedded environment for this robot vision system. The Xenomai project develops micro kernel (co-kernel) with interrupt controller and preemptive scheduler tightly integrated with the GNU/Linux operating system. The target kernel and device driver with real-time APIs were built and downloaded to the on-board flash ROM. The Xenomai operating system and the device driver for image processing run suitably on the Virtex-4 FX device environment. The robot vision software integrates real-time image capturing, functions for image recognition, pan-tilt motion control, and TCP/IP communication for remote manipulation.

7. Conclusions

This paper has described the development of a robot vision system with real-time image processing and evolutionary computation on FPGA. This robot vision system detects and tracks a moving object smoothly. Moreover this robot vision system detects the corridor boundaries between floors and walls, and judges its position and orientation in the passage environment. The system would be useful for applications such as supervising aged people or detecting a suspicious person as an indoor patrol camera robot. The application of evolvable hardware to the low-level image processing operators enables the visual navigation system to be adapted to various lighting environments.

As the next step, stereo measurement capability and motion tracking should be implemented in this system. Stereo measurement by trigonometry would enable expansion of object recognition incidental to the distance with two or more cameras on FPGA. GA could be used to determine the corresponding point in searching stereo images. Motion tracking by optical flow would enable perception of dynamic objects in the view, where tracking predictions for interest points could be found with flexibility using GA on FPGA in order to reduce the huge search computation on account of the real-time reaction of mobile robots.

References

[1] E. Bayramoğlu, N.A. Andersen, N.K. Poulsen, J.C. Andersen, and O. Ravn, “Mobile Robot Navigation in a Corridor Using Visual Odometry”, Proc. Int. Conf. on Advanced Robotics, id.58, 2009.

[2] G.N. DeSouza and A.C. Kak, “Vision for Mobile Robot Navigation: A Survey”, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.24, No.2, pp.237–267, 2002.

[3] S. Se, D.G. Lowe, and J.J. Little, “Vision-Based Global Localization and Mapping for Mobile Robots”, IEEE Trans. on Robotics, Vol.21, No.3, pp.364–375, 2005.

[4] K. Kudo, Y. Myokan, W.C. Than, S. Akimoto, T. Kanamaru, and M. Sekine, “Hardware Object Model and Its Application to the Image Processing”, IEICE Trans. Fundamentals, Vol.E87-A, No.3, pp.547-558, 2004.
[5] K. Okada, M. Inaba and H. Inoue, “Walking Navigation System of Humanoid Robot using Stereo Vision based Floor Recognition and Path Planning with Multi-Layered Body Image”, Proc. IEEE Int. Conf. on Intelligent Robots and Systems, pp.2155-2160, 2003.

[6] T. Morita, “Tracking Vision System for Real-time Motion Analysis”, Advanced Robotics, Vol.12, No.6, pp.609-617, 1999.

[7] J. Kiszka and B. Wagner, “Domain and Type Enforcement for Real-Time Operating Systems”, Proc. IEEE Int. Conf. on Emerging Technologies and Factory Automation, Vol.2 pp.439-446, 2003.

[8] S. Hadjitheophanous, C. Ttofis, A.S. Georgiades and T. Theocarides, “Towards Hardware Stereoscopic 3D Reconstruction: A Real-time FPGA Computation of the Disparity Map”, Proc. Conf. on Design, Automation and Test in Europe, 2010.

[9] N. Sawasaki, M. Nakao, Y. Yamamoto and K. Okabayashi, “Embedded Vision System for Mobile Robot Navigation”, Proc. IEEE Int. Conf. on Robotics and Automation, pp.2693-2698, 2006.

[10] K. Shimizu and S. Hirai, “CMOS+FPGA Vision System for Visual Feedback of Mechanical Systems”, Proc. IEEE Int. Conf. on Robotics and Automation, pp.2060-2065, 2006.

[11] H.P. Moravec, “Towards Automatic Visual Obstacle Avoidance”, Proc. Int. Joint Conf. on Artificial Intelligence, p.584, 1977.

[12] H.P. Moravec, “Sensor Fusion in Certainty Grids for Mobile Robots”, AI Magazine, Vol 9, No 2, pp.61-74, 1988.

[13] Atmark Techno, Inc., “SUZAKU-V Hardware Manual: SZ410-U00”, 2011.

[14] VXilinx Inc., “irtex-4 FPGA User Guide”, 2008.

[15] G. Hollingworth, A. Tyrrell, and S. Smith, “Simulation of Evolvable Hardware to Solve Low Level Image Processing Tasks”, Proc. Evolutionary Image Analysis, Signal Processing and Telecommunications Workshop, Vol.1596 of LNCS, Springer-Verlag, pp.46-58, 1999.

[16] R.O. Duda and P.E. Hart, “Use of the Hough Transformation to Detect Lines and Curves in Pictures”, Communications of the ACM, Vol.15, pp.11-15, 1972.

[17] K. Yaghmour, J. Masters, G. Ben-Yossef, and P. Gerum, “Building Embedded Linux Systems, 2nd Edition”; O’Reilly Media, Inc., pp.351-386, 2008.

[18] Xenomai Project, http://www.xenomai.org, access date: 2013.05.20

Hayato HAGIWARA (Non-member) He received M.E. degree from Kyushu Institute of Technology in 2013. He is a doctoral student at Graduate School of Engineering, Kyushu Institute of Technology from 2013. His current research interests include embedded systems, evolutionary robotics, and real-time robot vision system.

Kenichi ASAMI (Member) He received B.E. and M.E. degrees of information engineering in 1992 and 1994 respectively, and Dr.Eng. of information science in 1997, all from Kyushu Institute of Technology. He is an associate professor at Faculty of Engineering, Kyushu Institute of Technology from 2004. His current research interests include embedded systems, evolutionary robotics, and reconfigurable computing.

Mochimitsu KOMORI (Non-member) He received B.S. and M.S. degrees from Osaka University in 1982 and 1984, respectively. He received Ph.D. degree from Kyushu University in 1995. Now he is a professor in the Department of Applied Science for Integrated System Engineering, Kyushu Institute of Technology, Japan. His current research interests include superconducting bearings, micro bearings and their systems applications.