Proceedings of The 2016 IEEE International
Conference on Real-time Computing and Robotics
June 6-9, 2016, Angkor Wat, Cambodia

High-speed Target Tracking base on FPGA

Congyi Lyu, Yunhui Liu, *Fellow, IEEE*, Weiguo zhou, Jianqing Peng, ShanshanYang, Huijun Zhang, and Linsen Yang

Abstract— **Imaging based target tracking by a robotics is an interesting and popular topic. While the high computational power requirement of imaging processing algorithm is still a big limitation, some flying robot like micro aerial vehicles also has a payload limitation, so the high weight computer and GPU is no longer a good choice. This paper present a high speed ball target tracking system based on FPGA. With a CMOS sensor, the original frame can be transfer to the FPGA port. After the video streams go through the parallel image processing core, the color and corner features can be extracted. By comparing with the predefined target's pattern, the target can be detected. AXI bus can transfer the detected target's feature to the ARM core in the ZYNQ device. The PID controller embedded in ARM core can be used to control the Pan-tilt header to move the target to the center of frame.**

Keywords – FPGA, Target tracking, PID controller, Sobel operator.

I. INTRODUCTION

Tracking a target is one of the fundamental topics in robotic and computer vision system. Most of the computer vision application still relies on the PC or even GPU device to process the image data. Detect the target means separate the object from the background. The common ways to detect the target are background subtraction [1] and frame difference [2]. The background subtraction can just adopted by the static camera which can provide the almost unchanged background. Even via the background updating algorithm [3] the computation cost is very high. Frame difference method can remove the unchanged part by the frame subtraction. The computation cost is very low. Meanwhile the frame difference has its own problems. The frame difference will lose a lot of feature information for example color [4], corner [5] and optical flow [6]. Unfortunately these features are important for most robotics and computer vision applications.

Real-time detecting or tracking target by image processing is always widely adopted by robotics vision, industrial machine vision and other application area. The more and more high resolution of digital image sensor required a more powerful processing ability of image processing system. Different from the PC based image processing system, more and more image processing algorithm were implemented on FPGA or DSP device, for example Xiaofeng Lu et al. [7] using parallel Hough transforms to detect straight line feature by FPGA. N Isakova et al. [8] achieve stereo vision base on FPGA platform. Kazuhiro Shimizu and Shinichi Hirai [9] using CMOS and FPGA to build a real-time and robust target tracking system. Girish ChapIe et al. [10] design a sobel operator based on FPGA to detect edge information. Tomasz Kryjak et al. [11] realized a real-time background generation and target detection system. J.D. Anderson et al. [12] proposed a FPGA based vision system for an autonomous mobile robot. J Batlle et al. [13] proposed a FPGA/DSP hybrid image processing systemcan be used as a real-time application. In [16] Dillinger, P et al. using a FPGA and DSP based image processing system to detect falling objects, the areas calculation algorithm they proposed is efficiency. Liu et al. [17] proposed a FPGA based Zernike Moments calculation method and using this method to detect target in laser image. The Moments method is useful for the object recognition in binary image, but the processing speed of Moments by PC can't fulfill the real-time application requirement. Santos et al. [18] proposed a FPGA and fuzzy logic based position tracking system. They prove the FPGA's possibility of running a high level logic calculation algorithm.

In this paper we proposed a Xilinx Zynq (FPGA+ ARM) device based image processing system. Different from the FPGA/DSP or FPGA/ARM hybrid system, in our system FPGA and ARM core are putting together inside the Zynq device. Between the two processing cores, a AXI-bus can transfer data with a high band width and high frequency. Base on this AXI-bus we implement the image processing algorithm in FPGA core and robot control algorithm in ARM core. In section I we will describe our proposed target tracking system structure. In section II we will detail the target tracking algorithm implemented on FPGA device. In section III we will introduce the PID based Pan-Tilt head control algorithm. In last section, the experiment and result will appear and show the efficiency of our tracking system.

II. OVERVIEW OF THE SYSTEM

The proposed real-time target tracking system include a digital CMOS module, Xilinx ZYNQ-7000 FPGA development board, two degree of freedom Pan-Tilt servo platform, and a HDMI display screen. The CMOS module transfer the image sequence to the ZYNQ FPGA board, after the parallel processing by image processing cores in FPGA, the features can be extracted. Through the AXI-bus, the features information can be transfer to the ARM core in ZYNQ device. The PID controller based visual feedback control algorithm will be carried out in the ARM device, and control the motion of the Pan-Tilt servo platform to tracking the predefined target.

Congyi Lyu is with School of Mechatronical Engineering, Beijing Institute of Technology, Beijing, China. The Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, Hong Kong (e-mail: lylvcongyi@ 126.com).

Yunhui Liu is with The Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, Hong Kong

Shanshan Yang, Huijun Zhang, Linsen Yang, Weiguo zhou and Jianqing Peng are with School of Mechatronical Engineering, Harbin Institute of Technology Shenzhen graduate school, Shenzhen, China.

978-1-4673-8959-4/16/$31.00 © 2016 IEEE

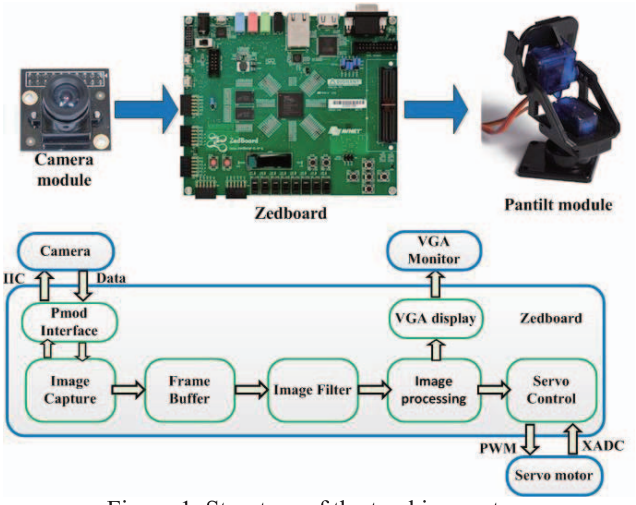

Figure 1. Structure of the tracking system

III. IMAGE PROCESSING IN FPGA

After the images grabbed via Pmod port in FPGA, the original image can be duplicated into three copies and the copies will be split in three processing tunnels. The first tunnel will calculate the frame difference between two consecutive frames. This tunnel will bring us the motion feature to us. The second tunnel will convert the original RGB image to HSV(Hue, Saturation, Value) [15] color space. The last tunnel will extract the canny [14] corner. All the three processing tunnels can parallel run with a frame synchronizer which can synchronize all the tunnels' processing rate.

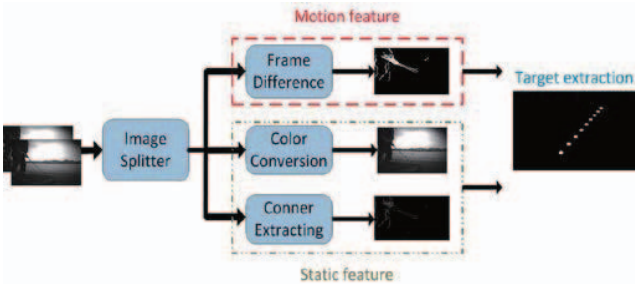

Figure 2. The structure of image processing

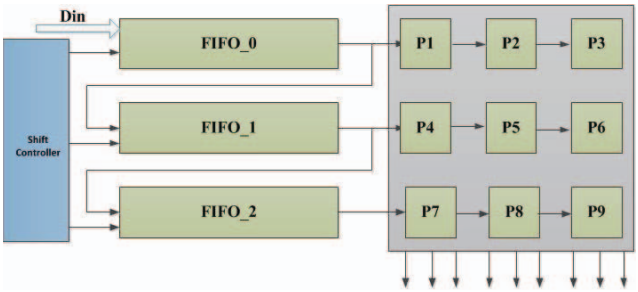

Figure 3. 3x3 Line buffer

In this paper, we use three FIFO (First In First Out) devices to build a line buffer. By this line buffer device the 3x3 window based filter algorithm can be established. In figure 3, the three lines of a window were store in the three FIFO. By the shift controller, the data flow can sequentially shift in the line buffer. The data input is the image pixel data, the output is the filtered data by 3x3 filter.

a.FRAME DIFFERENCE METHOD

Frame difference is commonly used algorithm to remove the unchanged part in frame and extract the changed target. The difference image can be easily converted to binary image by a preset threshold value.

$$D_n(x,y)=\begin{cases}1, & |P_n(x,y)-P_{n-1}(x,y)|>T_t\\ 0, & other\end{cases} \quad (1)$$

In equation (1) T means the threshold value, P means the grey level in (x,y) location of the pixel in frame. D means the result by frame difference algorithm.

b. Color space conversion

HSV color space was first introduced by [15], HSV is actually a nonlinear transform of RGB(red, green, blue) color space. In HSV color space the image can be represent by Hue, Saturation and Value. The conversion between RGB to HSV can be found in equation (2).

$$H=\begin{cases}0, R=G=B\\ \dfrac{(G-B)60^\circ}{\max(R,G,B)-\min(R,G,B)}+360^\circ, R?G\ B\\ \dfrac{(B-R)60^\circ}{\max(R,G,B)-\min(R,G,B)}+120^\circ, G?R\ B\\ \dfrac{(R-G)60^\circ}{\max(R,G,B)-\min(R,G,B)}+240^\circ, B?R\ G\end{cases} \quad (2)$$

$$S=\begin{cases}0, \max(R,G,B)=0\\ \dfrac{\max(R,G,B)-\min(R,G,B)}{\max(R,G,B)}, others\end{cases} \quad (3)$$

$$V=\max(R,G,B)$$

In FPGA system, we first split the image into red, green and blue channel. With the logic processing, the HSV value can be obtain. The structure of the logic schematic can be found in figure 4.

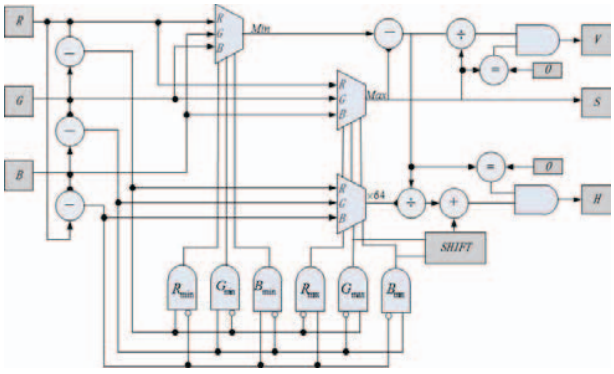

Figure 4. The logic of color convertor

c. Conner extraction

The edge information of the image can represent the contour of the target. Here we select the Sobel operator [14] algorithm to collect the edge information. Before calculation of gradient in x and y direction, we use the median filter to remove some noise pixels in frame. Median filter using the median value in a 3x3 window to replace the original value of the pixel. The equation (a) is the median filter mathematical representation.

$$p'(i,j)=\frac{1}{n^2}\sum_{k=-\frac{n}{2}}^{\frac{n}{2}}\sum_{l=-\frac{n}{2}}^{\frac{n}{2}}p(i+k,j+l) \qquad (4)$$

The median algorithm is easy to carry out on FPGA device. The figure 5 shows the simulation of median filter circuit.

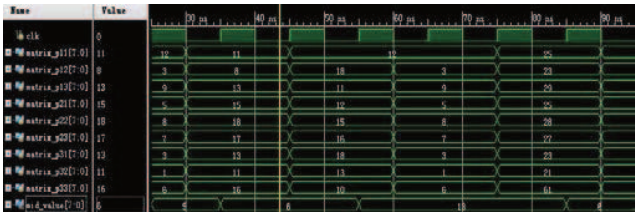

Figure 5. the simulation result of corner extraction

After the median filter, the pixel will be send to the Sobel calculation core. The 3x3 window in figure 6 (a) can express the neighborhood of the target pixel; we can complete Sobel edge detection by calculating the 3x3 window with Sobel convolution factors like figure 6 (b) and (c).

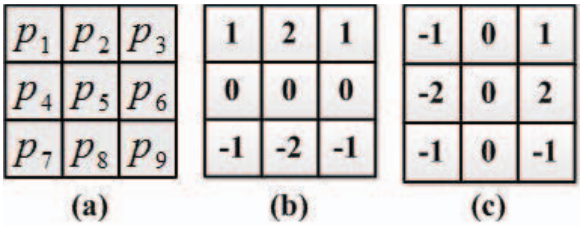

Figure 6. (a)Original image 3x3 pixels (b)Vertical Sobel operator (c) Horizontal Sobel operator

$$G_x=-p_1-2p_2-p_3+p_7+2p_8+p_9$$
$$G_y=p_1-p_3+2p_4-2p_6+p_7-p_9 \qquad (5)$$
$$G=\sqrt{G_x^{\,2}+G_y^{\,2}}$$

With the predefined vertical or horizontal Sobel operators and original image, the convolution calculation can be carried out. The x direction and y direction gradient can be obtain. The calculation in equation (5) can be converted to comparator, subtractor and bits shifter, the logic schematic of Sobel operator can be found in figure 7. After the gradient calculation, the defined threshold value has been used to remove the small gradient pixels. So after the Sobel operator the gradient image can be found in figure 8.

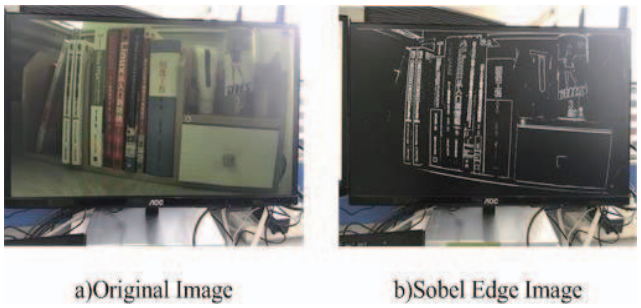

Figure 8. The Sobel edge extraction result

d. Multiple feature based target recognition

As the figure 1 showed, the original image can be copied and transfer to three processing tunnels. After the image go through three tunnels the difference image, HSV image and corner image can be obtained. We define the HSV color and corner features as the static features which means the features on the surface of target. The feature can still be detected when the target is static. On the contrary, the frame difference feature can give us more motion information, but when the target is static the frame difference is useless. So we combine both the motion and static feature to detect the target we want find. The detail process can be found as follow steps:

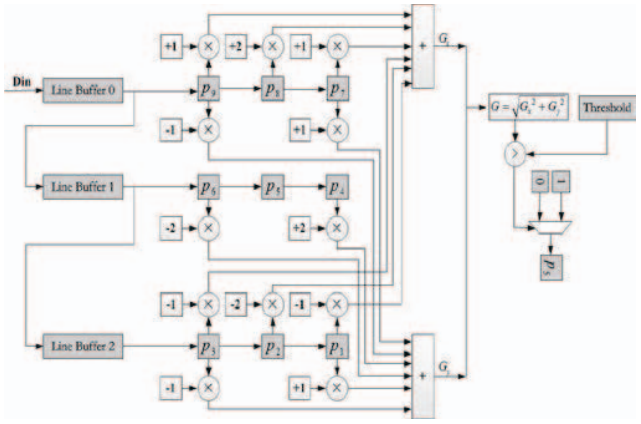

Figure 7. The logic schematic of Sobel extraction

a) Keep the camera at static status and active the frame difference core to detect the motion feature.

b) Find the ROI (region of interesting) from the difference image.

c) If the size of the ROI is bigger than threshold value, jump to the static feature step.

d) Carry out the static feature detection on ROI detected by first step.

e) Repeat the target comparing processing until the detection failed.

IV. TARGET TRACKING

After feature extraction by FPGA part, the target's features can be obtained. Through the high-speed AXI bus design by Xilinx, the data and even the image can be transfer from FPGA core to ARM core. In the ARM core, a PID based control algorithm was carried out. Through the Pmod

extension port, the servo motor control signal can be sent to the Pan-Tilt header.

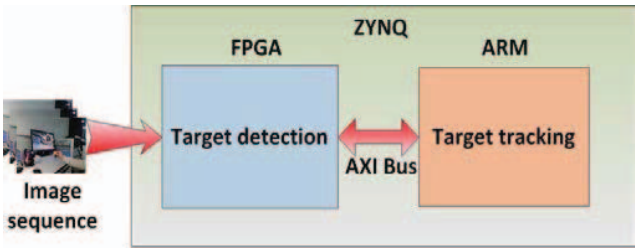

Figure 9. AXI bus between FPGA and ARM cores

We define difference between center of frame and the current target's location as the error or input value to the PID controller. The output of the controller is PWM wave to control the servo motor. The pan-tilt header with two servo motor can move the camera in both x and y direction. To make sure the convergence of the controller, we use a tunable PID controller base on the error feeding back. We set up a LUT (look up table) to store PID parameter. The PID result relies on the error information and parameters.

$$u = k_i \int_0^t ed\tau + k_p e + k_d \frac{de}{dt} \qquad (6)$$

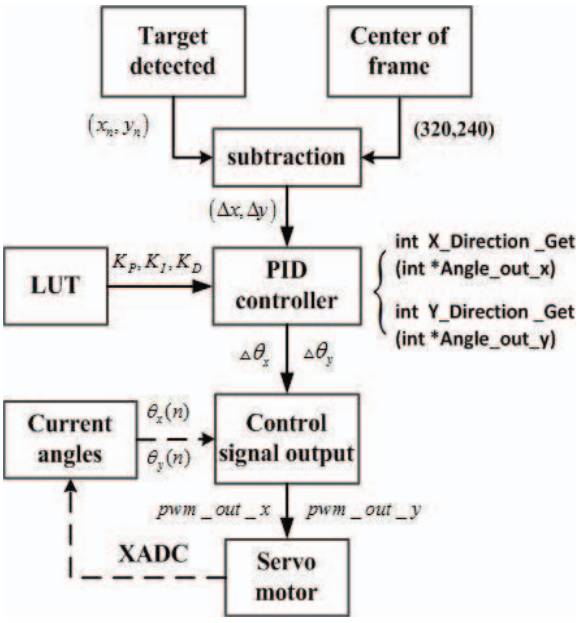

Figure 10. The flowchart of PID controller

V. Experiment and Result

The figure 11 showed the experiment set up, the tracking system include a ov7755 digital CMOS module, a Xilinx Zynq board, Pan-tilt platform with two servo motors and a VGA display screen. The CMOS transfer the image data with parallel port, the output data is 8 bits digital signal which is easy to capture by FPGA device. The PID controller parameter can be decided by pre-loaded parameter LUT in the experiments.

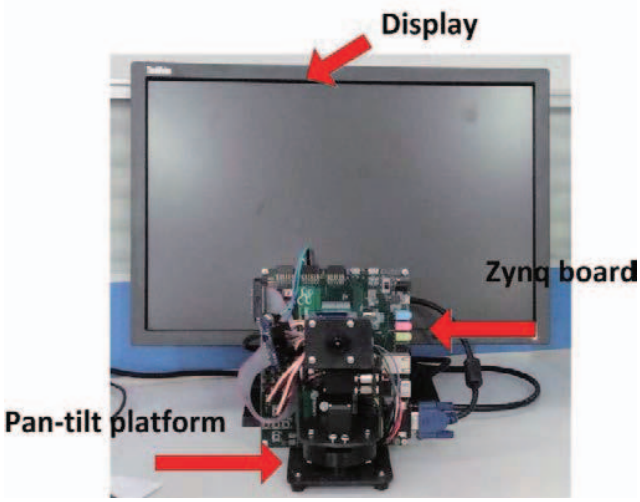

Figure 11. Experiment set up

The figure 12 showed the real time tracking result, in the figures 12 the target (ping pang ball) on screen is marked by green marker, and we can find the green target can be robustly keeping in the center of screen even the scale of the target has been changed. The CMOS sensor can provide video at 60 fps (frame per-second). Our proposed tracking system can do the target tracking at around 58fps which is enough for the real-time application.

Figure 12. Sample frames of tracking result

VI. Conclusion

This paper present a high speed ball target tracking system based on FPGA. Hardware based color and corner features processing cores has been adopted to process the image grabbed by CMOS module. By our proposed static and motion feature based two-step tracking algorithm, the static or moving target can be robustly detected. By comparing with the predefined target's pattern, the user defined target can be detected. With the PID controller embed in the ARM for the Pan-Tilt control can keep the target in the center of frame.

The experiments result showed the efficiency and robustness of our proposed tracking system.

VII. ACKNOWLEDGMENT

The work supported by Shenzhen Peacock Plan Team grant (KQTD20140630150243062). Shenzhen Fundamental Research grant (JCYJ20140417172417120). Shenzhen Fundamental Research grant (JCYJ20140417172417145). Shenzhen Key Laboratory grant (ZDSYS201405081618250 65).

REFERENCES

[1] Piccardi, M.: Background subtraction techniques: a review. In: IEEE International Conference on Systems, Man and Cybernetics. Volume 4. (2004).

[2] Zhan C, Duan X, Xu S, et al. An Improved Moving Object Detection Algorithm Based on Frame Difference and Edge Detection. IEEE Computer Society, 2007:519-523.

[3] Kim K, Harwood D, Davis L S. Background updating for visual surveillance[C]// Proceedings of the First international conference on Advances in Visual Computing. Springer-Verlag, 2005:337-346.

[4] Pérez P, Hue C, Vermaak J, et al. Color-Based Probabilistic Tracking[C]// Proceedings of the 7th European Conference on Computer Vision-Part I. Springer-Verlag, 2002:661-675.

[5] Yaron O, Sidi M. A Combined Corner and Edge Detector[C]// In Proc of Fourth Alvey Vision Conference. 1988:147-151.

[6] Barron J L, Fleet D J, Beauchemin S S. Performance of optical flow techniques[C]// IEEE Computer Society Conference on Computer Vision & Pattern Recognition, Cvpr. 1994:236 - 242.

[7] Lu X, Song L, Shen S, et al. Parallel Hough Transform-Based Straight Line Detection and Its FPGA Implementation in Embedded Vision[J]. Sensors, 2013, 13(7):9223-9247.

[8] Isakova N, Basak S, Sonmez A C. FPGA design and implementation of a real-time stereo vision system[C]// Innovations in Intelligent Systems and Applications (INISTA), 2012 International Symposium on. IEEE, 2012:1-5.

[9] Shimizu K, Hirai S. Implementing planar motion tracking algorithms on CMOS+FPGA vision system[C]// Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on. IEEE, 2006:1366-1371.

[10] Chaple G, Daruwala R D. Design of Sobel operator based image edge detection algorithm on FPGA[C]// Communications and Signal Processing (ICCSP), 2014 International Conference on. IEEE, 2014.

[11] Kryjak T, Komorkiewicz M, Gorgon M. Real-time background generation and foreground object segmentation for high-definition colour video stream in FPGA device[J]. Journal of Real-Time Image Processing, 2014, 9(1):61-77.

[12] Anderson A J D, Lee D J, Archibald J K. FPGA Implementation of Vision Algorithms for Small Autonomous Robots[J]. Proceedings of SPIE - The International Society for Optical Engineering, 2005, 6006:600612-600612-11.

[13] Batlle, J., et al. "A New FPGA/DSP-Based Parallel Architecture for Real-Time Image Processing." Real-Time Imaging 8.5(2002):345-356.

[14] Kanopoulos N, Vasanthavada N, Baker R L. Design of an image edge detection filter using the Sobel operator[J]. Solid-State Circuits, IEEE Journal of, 1988, 23(2):358-367.

[15] Cucchiara R, Grana C, Piccardi M, et al. Improving Shadow Suppression in Moving Object Detection with HSV Color Information[C]// 2001 IEEE INTELLIGENT TRANSPORTATION SYSTEMS PROCEEDINGS. 2001:334-339.

[16] Dillinger, P.; Vogelbruch, J.F.; Leinen, J.; Suslov, S.; Patzak, R.; Winkler, H.; Schwan, K., "FPGA-Based Real-Time Image Segmentation for Medical Systems and Data Processing," in Nuclear Science, IEEE Transactions on , vol.53, no.4, pp.2097-2101, Aug. 2006

[17] Liu Li-feng; Ma Hui-min; Lu Ming-quan, "A FPGA and Zernike Moments Based Near-Field Laser Imaging Detector Multi-scale Real-Time Target Recognition Algorithm," in Information Science and Engineering (ISISE), 2010 International Symposium on , vol., no., pp.370-374, 24-26 Dec. 2010

[18] Santos MPSD, Ferreira J A F. Novel intelligent real-time position tracking system using FPGA and fuzzy logic[J]. Isa Transactions, 2013, 53(2):402-414.