Advances in Engineering Research (AER), volume 143
6th International Conference on Energy and Environmental Protection (ICEEP 2017)

Implementation Of Harris Corner Matching Based On FPGA

Xu Chengda[a], Bai Yunshan[b]

Transportion Service Department, Bengbu Automation NCO Academy, Bengbu 233000, China

[a]756761643@qq.com, [b]504122278@qq.com

Keywords: FPGA, Pre-filtering, Harris corner extraction, Harris corner matching

Abstract. A method for applying FPGA to realize the Harris corner detection and matching, aiming to overcome the problem that it is slow to fetch feature information from the image which show the planar motion of robot. Considering the theory of Harris corner detection and the structure of FPGA, the algorithm is improved in order to simplify calculation and simulated. The simulation result show that the algorithm which guarantee of corner extraction could decrease the number of redundancy corners and raise the real-time performance of system. Otherwise it can also lower the difficulty of design and reduce the hardware overhead, in favor of analyzing the movement of robot.

Introduction

The main way to analyze robot motion is using machine vision to acquire valid information from the image and dispose. Corner matching mainly includes corner extraction and corresponding corner matching. At present, the commonly used corner extraction algorithms are Harris algorithm, Susan algorithm and Moravec algorithm [1] etc. Harris corner algorithm is widely used in computer vision because of its accurate location, strong anti-interference and simple operation. However, for the large amount of calculation, it is difficult to realize real-time requirement by PC. Through the analysis of Harris algorithm, it is found that although the algorithm is large in computation but calculation operation is simple. So it is suitable for using FPGA(Field Programmable Gate Array) to implement, which can increase processing speed. Harris corner matching [2] is primarily implemented by frame difference method, which the corresponding corner from front and back frame are matched. Based on pipeline processing of FPGA, it can effectively shorten the time required for matching. Considering the hardware structure of FPGA and the principle of Harris corner points, some appropriate improvements are made to the algorithm.

Harris corner detection principle and improvement

Harris algorithm is a point feature extraction algorithm based on gray value proposed by C.Harris and J.Stephens [3] in 1988. The algorithm is inspired by the auto-correlation function of signal processing. The main principle is to detect the edge which can be accurately measured and positioned in the image, then the auto correlation matrix M is obtained.The eigenvalue of matrix M is the first-order derivative of the correlation function. If the derivative values are both high, the point is the desired corner point. Main formula of algorithm, which is given by

$$M(x, y) = \begin{bmatrix} w(x, y) * (I_x(x, y))^2 & w(x, y) * (I_x(x, y)I_y(x, y)) \\ w(x, y) * (I_x(x, y)I_y(x, y)) & w(x, y) * (I_y(x, y))^2 \end{bmatrix} \quad (1)$$

$$R = \det(M) - a * trace^2(M) \quad (2)$$

$w(x, y)$ denotes the gauss operator; $I_x(x, y)$ denotes the gradient of the X direction; $I_y(x, y)$ denotes the gradient of the Y direction; the value of experience value a is in range of 0~0.25. From the Eq. (1) and Eq. (2), the solution process needs to manipulate at least differential operation one times and the Gaussion filter three times. In addition, Harris algorithm is sensitive to image noises, so the smoothing filter needs to be added before the Harris algorithm. Gaussion filter is usually used to reduce

the interference of image noises and the computation of corner matching. Using Harris algorithm to extract corner needs to manipulate the Gaussion filter four times. In order to improve the real-time and operability of the algorithm, it is very necessary to simplify the Gaussion filter.

The definition of Gaussian function is as follows [4]:

$$w(x,y) = e^{-\frac{x^2+y^2}{2s^2}} \tag{3}$$

σ is the standard deviation which determines the smoothness of the Gauss filter. In order to simplify the computation and locate the target accurately, 5 dimensional Gaussion matrix ($s = 1$) is generally used. The Gaussion kernel matrix is computed by Matlab, which can be shown as 1/84*[1 2 3 2 1; 2 5 6 5 2 ; 3 6 8 6 3 ; 2 5 6 5 2 ; 1 2 3 2 1] after discretization. According to the kernel matrix, Gaussion filter needs a total of 25 multiplier, 24 adders and 1 divider. If use FPGA to structure the associated components directly, it increases the difficulty of calculation, not conducive to the operation of real-time. In addition, it also increases the workload of design, occupying large amount of resources.

According to the form of the Susan operator in the article [5], in combination with FPGA signal flow made, it is necessary to improve the window matrix appropriately. Aiming to reduce the resource overhead of hardware, the sum of matrix values must be 2 integers. So the specific form is as follows: (1/16)*[0 0 1 0 0;0 1 1 1 0;1 1 4 1 1;0 1 1 1 0;0 0 1 0 0]. The improved template can basically realize the filtering function of Gaussion filter. Then it also satisfy the precision requirement of Harris corner extraction. The hardware design is simplified from the following two points:

a. Normalize the coefficient of the matrix: all the coefficients of the matrix are all integers of 2, which can be achieved by the logical shift easily instead of using multiplier and divider, to reduce the complexity of design.

b. Improve real-time performance of the system: it greatly reduces the sysytem computing operations, because the hardware design only need addition and shift operations.

Corner matching is mainly based on the ***R*** of the detected corner points. According to Eq. (1) and Eq. (2), each pixel point of the input image corresponds to an ***R***, although it may exist the same ***R***, the probability is small so that it can be ignored. Fig.1 is the chart of the whole process.

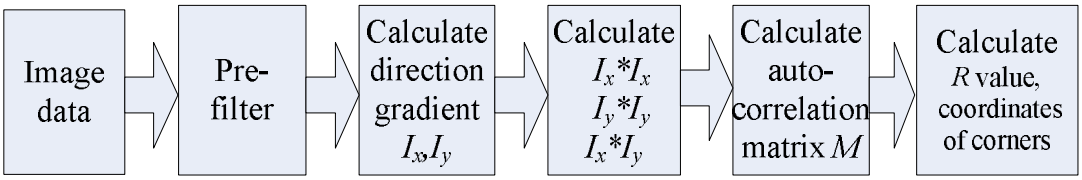


Fig.1. Harris algorithm process

Hardware implementation of corner matching

Based on Fig.1, the algorithm can be divided into linear buffer module, pre-filtering module, First-order derivative module, low-pass filter module, non-maximal suppression module and corner matching module. As the pre-filtering module, low-pass filter module and derivation module belong to matrix operation, the design of derivation module is only explained here. The design of specific modules is described in this section.

Linear buffer module

Since the 3 dimensional matrix is used in the first-order derivative module, it need to construct 2 row buffers to implement the normal operation of the matrix, Fig. 2 is a diagram of line caching.

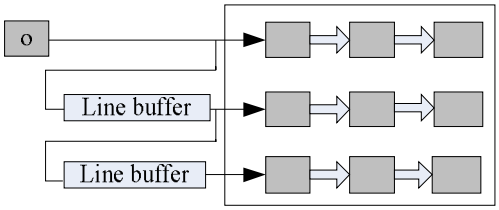


Fig.2. Line buffer diagram

linear buffer is constructed using the Ram Register IPCore of the ISE platform, which data width is defined as 8 bits and each row buffer depth is defined as 256, corresponding to the pixel values of the gray image and the width of image, respectively. This allows the data in the cache to be reused in subsequent modules to achieve computation accelerating.

First-order derivative module

In order to calibrate the pixel coordinate position which used be calculated on the X and Y axis, use the X, Y direction matrix of Sobel operator instead of first-order dimensional derivative vector. S_x,S_y can be acquired by the following formula:

$$s_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad s_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

As the one-dimensional vector ,the Sobel operator just includes addition, subtraction, and shift operations, which can be finished in one operation cycle. For example, the S_x can be designed as follows:

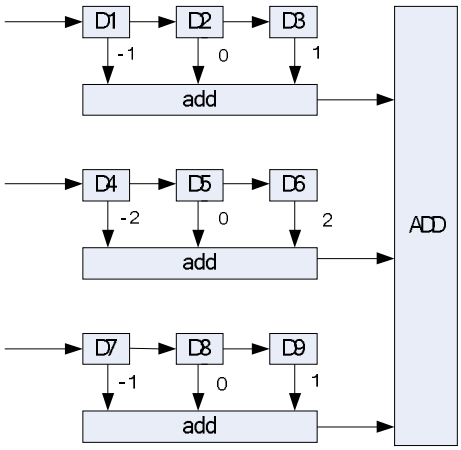


Fig.3. diagram of first-order derivation module

Non-maximal suppression module

According to Eq.2, calculating Harris response *R* of pixels need to use subtraction and multiplication. In the case of satisfying the performance of the algorithm, assign the value of □ with negative integer powers of 2 in order to simplify operations. Non-maximal suppression need to compare the R of pixel with the *R* of adjacent eight pixels; in addition to, the *R* of pixel need compare with R_{max}. If the value of pixel is greater than the *R* of adjacent eight pixels and $\frac{1}{64}R_{\max}$, the pixel point is corner point of image. Then the coordinate values of corner should be recorded. The main operation of the module is logical comparison, which use 3 dimensional matrix to traverse all image pixels to confirm all the corner of image.

Overall realization

The overall FPGA program framework is shown in Fig.4. The Block Ram1 records the ***R*** and coordinate values of corner in the current frame; the Block Ram2 records the ***R*** and coordinate values of corner in the last frame. The FIFO1 records the coordinate values of the matching corner in the current frame and the FIFO2 records the coordinate values of the matching corner in the last frame. When the program starts to work, the data stored in the Block Ram2 is set to zero. While FPGA reads an untreated image, the data stored in the Block Ram1 copy to the Block Ram2.

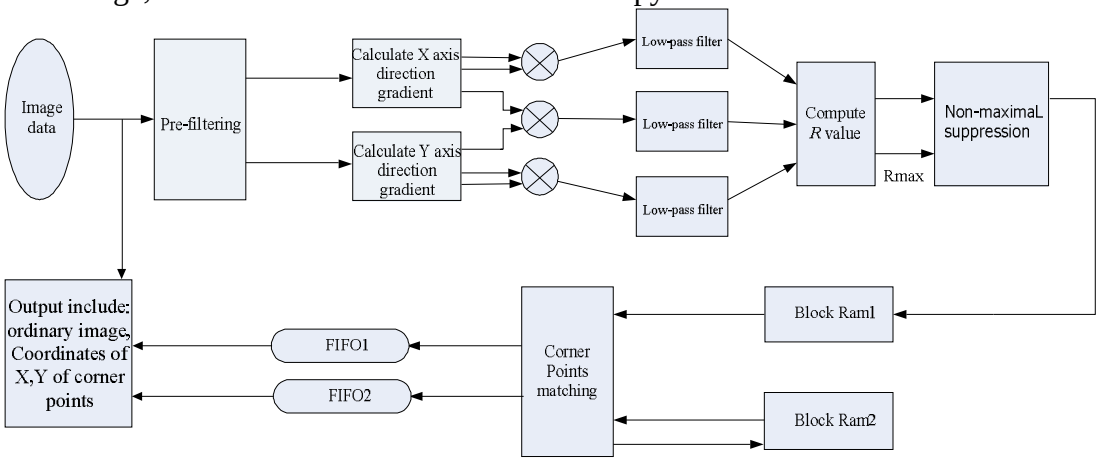
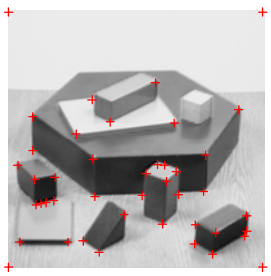


Fig.4 overall block diagram of FPGA program

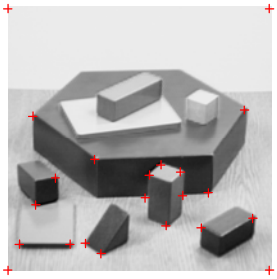
As shown in the Fig.4, each module is connected , constituting a parallel pipeline working mode. Simultaneously, this mode is working inside of each module.This mode can be used to acquire the image at the same time to complete the corner detection, effectively reducing the time cost of the algorithm. So the processing speed depends mainly on the image acquisition speed.

Simulation results and analysis

The hardware used in the experiment is xc3sd3400a-4fg676 chip of Xilinx, which clock frequency is 50MHz. The software platform is ISE 14.5. The size of the acquire image is 256*256. The processing cycle of system is about 2.3ms. In order to observe the results of experiment, output data are showed through matlab 2012a.



a) Harris algorithm



b) add pre-filtering Harris algorithm

Fig.5 result of Corner detection

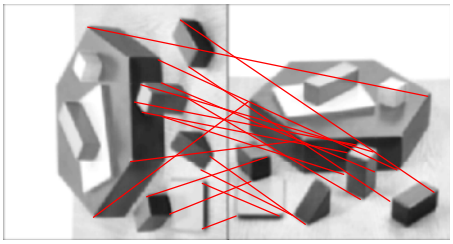


Fig.6 result of Corner matching

According to compare Fig.5(a) and Fig.5(b), Adding a pre-filtering can reduce the amount of corner and remove those fake corner points. The algorithm used in this paper can basically achieved the exact match of the corner by Fig.6.

Conclusion

In this paper, the powerful parallel computing power of FPGA is used to improve the calculation speed of the corner matching algorithm, and the result of matching is accurate .This is beneficial to the calculation of the motion of the robot and improve the real-time performance of the system. In addition, by adding the pre-filter and using the simplified matrix instead of the original Gaussian kernel matrix, the design complexity of the FPGA is reduced and the hardware resources are saved, which has strong application value. On the basis of ensuring the detection of feature points, the scale invariance of the algorithm is the direction of the next step.

References

[1] Cristina Cabani, Implementation of an affine-invariant feature detector in field-programmable gate arrays. University of Toronto, 2006:5-13.

[2] Yong Zhang, Jianping Yu, Junwei Sun, Tie Jin. Corner Matching Research based on Harris Algorithm [J]. Computer and Modernization, 2011(11).

[3] HARRIS C, STEPHENS M. A Combined Corner and Edge Detector[C]. UK: Manchester University Press, 1988:147-151.

[4] Lihua Lu. Research on Image Matching Algorithm and High - speed Parallel Implementation Method. Nanjing University of Aeronautics and Astronautics, 2010(12)：40-46.

[5] Jinxiao Yang, Qiang Zhao, Baiping Yang, et al. FPGA Implementation of an Improved SUSAN Algorithm [J]. Information Security and Communication, 2010(03): 57-59.