A-Remote-Hand-Gesture-Recognition-System

Team PR085

Description

本项目拟采用 FPGA 为控制器和运算器，搭建一套远程手势识别系统。本系统使用摄像头作为输入设备，对特定手势的含义进行识别，并通过 UART 通讯串口给电脑发送数据实现控制电脑进行 PPT 上下翻页等功能。意在摆脱空间的限制，使使用者可以远距离更自由的控制电脑。

Project Proposal

1. High-level Project Description

1.1 Purpose of the design

In recent years, the motion capturing technology is widely used in computer games, video special effects and motion analysis. The system to implement human-body motion capturing is a complex platform which consists of sensors and computer aided analysis. This project aims to use Intel FPGA as controller and operator, to implement a **Remote Hand Gesture Recognition System (RHGRS)**. We aim to use ***Terasic DE10-Nano Kit*** as our developing platform.

1.2 Application Scope

The system to implement human-body motion capturing usually has two basic components: Frontend and Backend. Frontend is generally used as the collector of data which relates to human movement, while Backend has functions of data processing and implementing judgement algorithms. Our application (system) intends to use a camera as the input and a PC as the terminal. By replacing the operation of clicking, rolling and moving with different gestures, out system collects hand movements and transfers these gestures to computer instructions, such as volume controlling, slides displaying and software running.

1.3 Targeted Users

The **Remote Hand Gesture Recognition System (RHGRS)** can be widely used to users with various careers. For people who dedicate to education enterprise, the system can liber their hands for contacting with interfacing facilities (mouse or keyboard) which have to be placed near PCs. Teachers, for example, can walk about in classrooms without carrying a wireless mouse anywhere. For disabled people who cannot do the motion of clicking because of the atrophic muscles (aka. *amyosthenia*), the system can aid them to operate the computer precisely as normal people can do.

1.4 The use of ***Terasic DE10-Nano Kit***

The intention of using ***DE10-Nano Kit*** as our developing platform is mainly based on following considerations:

1. Comparing to general CPUs, the use of FPGAs can accelerate calculation speed and reduce programming effort.
2. Compared to other FPGAs, the developing kit has 80 IO pins, which absolutely meet the need of multiple datapaths transmission.
3. There are several 32-bit fast Fourier transform (FFT) engines, which could be used to transform video signal into spectrums to analyze the feature of videos.
4. The kit has 5,570Kb block RAM in it, which can store more than 2 frames of images.
5. Comparing to Terasic **OpenVINO Starter Kit**, DE10-Nano Kit integrates an ARM core, which brings more freedom in programming.
6. Most importantly, HDMI interface is supported on this kit, which is helpful when testing camera and verificating design algorithms.

Here is a table show the FPGA resource comparation between DE10-Nano Kit, OpenVINO Starter Kit and Xilinx EGO1 Kit.

FPGA Developing Kit	DE10-Nano Kit	OpenVINO Starter Kit	Xilinx EGO1 Kit
Logic Elements (LE)	110,000	301,000	33,280
Block RAM (KB)	5,570	13,917	1,800
UART to USB	√	√	√
DSP blocks	112	N/A	90
VGA output	×	×	√

HDMI output	√	×	×
ARM core	√	×	×
Arduino Pins	√	√	×
SPI	√	√	√

2. Block Diagrams

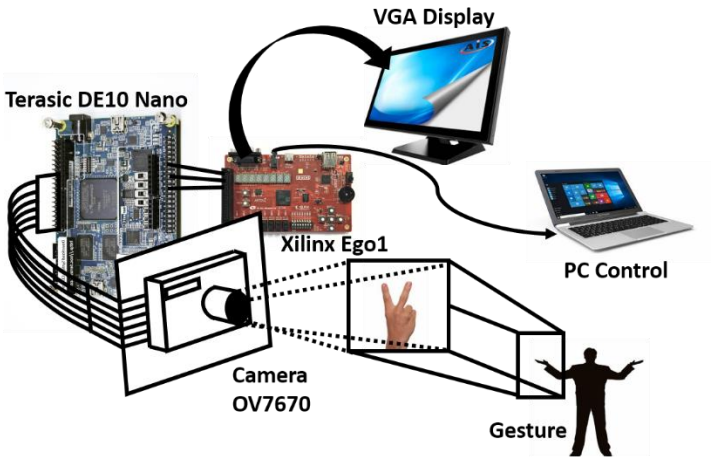


Fig 1. System Overview

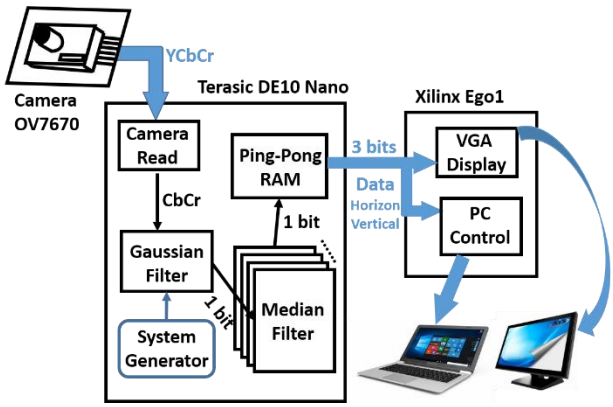


Fig 2. Block Diagram

2.1 System Overview

As shown in Fig 1, this whole system contains 4 parts. A camera is used to get the gesture of user as input part. DE10-Nano Kit and Ego1

play roles of controller and calculator. Then the output signal is transported by USB cable from Ego1 to the laptop which will be controlled. It is essential to use a screen to display video signals when debugging the system.

2.2 Block Dataflow

The RHGRS consists of 7 core blocks. VGA Driver, Gauss Filter, Median Filter, System Generator, Ping-Pong RAM, Finger Detection and Hand-Segment. Block diagram is shown in Fig 2. The Structure is mainly based on the ICSP2014 Proceedings Article ***Real-time visual static hand gesture recognition system and its FPGA-based hardware implementation.*** The dataflow can be briefly described as followed:

1. When a frame of image is detected, the camera OV7670 can automatically transfer it into YUV format.
2. The VGA Driver collects this frame, sample the U and V dimensions, and sends them to Gauss Filter.
3. In Gauss Filter, the U and V dimensions are compressed and compared with a filter (the System Generator block) which contains basic messages of human-skin color. The comparator in Gauss Filter will binarize the frame in accordance to the comparation results. The output of Gauss Filter is a frame of distinguished binarized data.
4. Median Filter consists of 10 sub-filters which can enhance the messages outputted by Gauss Filter. With 10 sub-filters, the system can gain a promoted accuracy and transmission speed.
5. The output of Median Filter will be bifurcated into two paths: the path to Ping-Pong RAM and the path to Hand-Segment. Ping-Pong RAM is an advanced technology of sampling acceleration, which can guarantee the speed of transmitting, while Hand-Segment is a kind of algorithm which detect the hand in the whole frame.
6. The output of Ping-Pong RAM will be sent to Display terminals through HDMI, and will also be used as the supervising messages in finger detection. The Finger Detection Block collects outputs of Hand-Segment and Ping-Pong RAM and processes the data into detailed instructors through integrated algorithms. The instructors will finally be sent to PC through UART of USB, which can finally achieve the function of PC control.

Reference:

R. Wang, Z. Yu, M. Liu, Y. Wang and Y. Chang, "Real-time visual static hand gesture recognition system and its FPGA-based hardware implementation," ***2014 12th International Conference on Signal Processing (ICSP),*** **Hangzhou, 2014, pp. 434-439.**

3. Intel FPGA Virtues

There are 3 main performance parameters in the design: Accuracy, Delay and Stability. Our design aims to reach the goal of ~80% recognition accuracy and <150 ms reaction delay, respectively. The quantification of stability will be verified according to the maximum hanging up time, which describes the maximum time the system can stay hanging up (do not send any signals to terminals) when there's no effective inputs.

Among parameters above, accuracy will always be the first concerned one. It mainly depends on the performance of camera and the recognition algorithm. With higher resolution of video input, the accuracy can reach a higher percentage, while the space consumption will also increase. A trade-off should be considered between these two facts. Similarly, if we want to design a high-performance algorithm, we must do minimal simplification of the data which also need larger storage. Based on analysis above, the main factor limiting accuracy of the device is the storage of FPGA.

Next, limitations of latency are the number of Logic Elements in FPGA and the algorithm. FPGA with larger number of Logic Elements has a high parallelism, which means logical calculation can be performed in a higher speed. The design of algorithm also has effect on latency by involving execution order of the program.

It is important for the device to work with high stabilities. It mainly depends on the recognition algorithm, and the performance of the camera. To achieve design goal, we temporarily choose camera OV7670 with a proper resolution which means it meets requirements of both minimizing resolution and maximizing the utility of storages on the FPGA.

Above all, there are 2 main factors of hardware which can affect performance we need ---- memory size and number of logic cells. We have realized the design using Xilinx EGO1 but it cannot meet three parameter requirements at the same time. For example, when we use Camera OV7670 as the input, accuracy can reach 80% but latency will beyond 300 milliseconds. When we change a camera with lower resolution as input signal, the latency can meet requirement but the accuracy is low and

the system stability is less than 1 minute. Compared with Xilinx EGO1, DE10-Nano has 3 times of Block RAMs capacity and 3 times of Logic Elements quantity, which will bring about a reduced latency.

According to discussions above, it's feasible to use DE10-Nano to meet our overall design requirements.

4. Design Introduction

The system uses OV7670 camera as the gesture detector, and send the video signals to processing and recognition. The system uses Gauss filter to binarize each frame in accordance to skin color, then uses medium filter for smoothing. Finally, by using positioning and compartmentalizing methods, the pre-processed images are achieved for further recognition algorithm.

We want to thank the paper **R. Wang, Z. Yu, M. Liu, Y. Wang and Y. Chang, "Real-time visual static hand gesture recognition system and its FPGA-based hardware implementation,"** ***2014 12th International Conference on Signal Processing (ICSP),*** **Hangzhou, 2014, pp. 434-439.** for providing us with a good hardware solution, and by making some revisions based on this architecture, we make the implementation of the function that we need more easier.

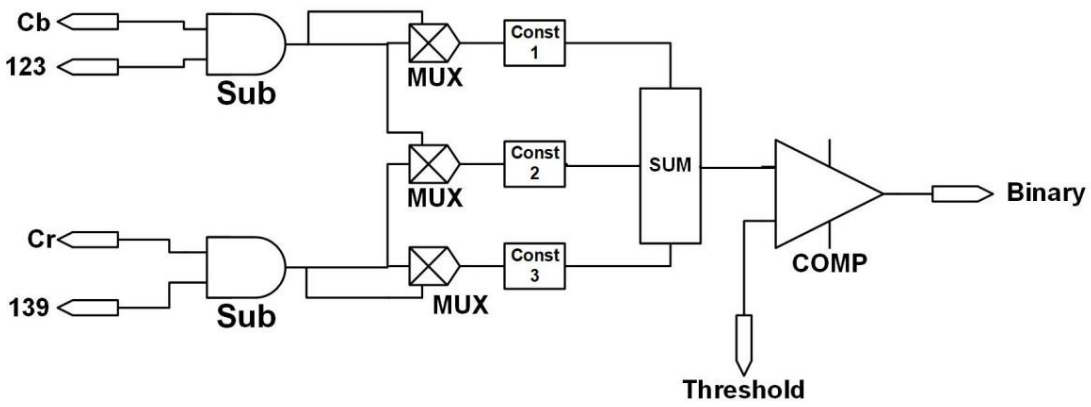
4.1. Pre-Processing

The system uses the design simultaneous clock, in which pclk (30 MHz) is the system clock. By configuring the registers in OV7670, the format of YUV/RGB images (We choose YUV) can be obtained. YCbCr space is a dispersion of YUV space, in which U equals to Cr and V equals to Cb. The collected image data are formed into YUV4:2:2, and later we filter out Y signal, which we don’t care much, and get only U and V signals. The following images shows what U and V channels are like when displaying.

4.2. Gauss Processing

Through clustering analysis of skin on different color spaces, skin color is compressed in YCbCr space, so we use this space to process skin color. It is possible to use Gauss distribution describe skin color, and later detect it from back colors.

2D Gauss Possibility Model is: P(Cb，Cr)=exp[-0．5(x—M)TC-1(x—M)], x(x=[Cb，Cr]T)is the value of pixels in YCbCr space, M(M=E(x)) is the sample means of skin color in YCbCr space, and C (C=((x—M)(x—M)T))is the Covariance matrix of skin color similarity. Through dataset we can know that M=[123.1015,139.2258]T, C=[0.0108,-0.0019;-0.0019,0.0074]. By importing IP cores, binarization is implementable.

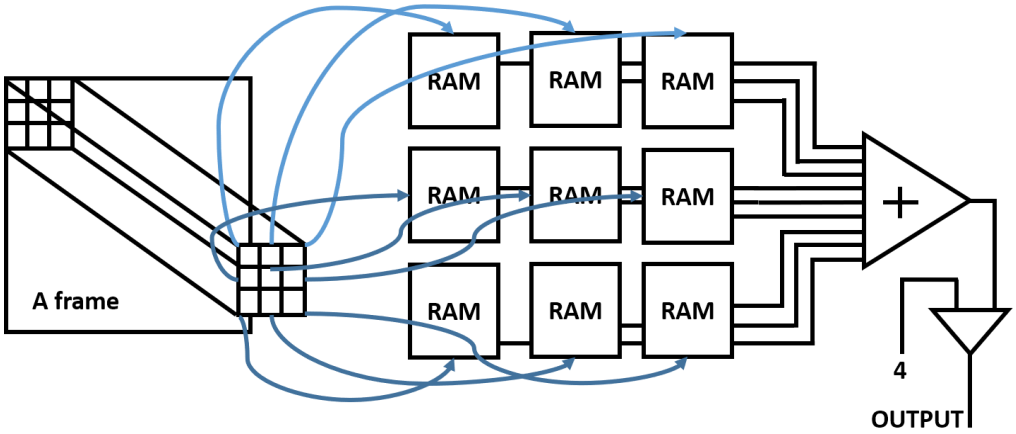


4.3. Medium Filter

The model is used for enhance relative information in images while obviate irrelative information to promote the ability of recognition of certain information.

Medium Filter replaces each random pixel with the means of its neighbored 8 pixels. In the system we apply this to do the filtration and by calculating, we finally get the medium filtered images.

Since the speed is critical to overall system performance, 30 frames per second is required, which need simultaneous processing of all models. Take this into consideration, we have to implement the fliting before next frame comes, so we assign 8 RAM to read and write simultaneously. By applying gauss_clk as system clock, we compart signal into 8 identical parts and do medium filter at the same time. After that, we output the processed result while read in the next frame, and compare the previous frame with number 4. The model uses first read last write to prevent data volatile. This design optimizes the clock distribution and reduces competing, makes it possible for the whole system to do real-time processing.

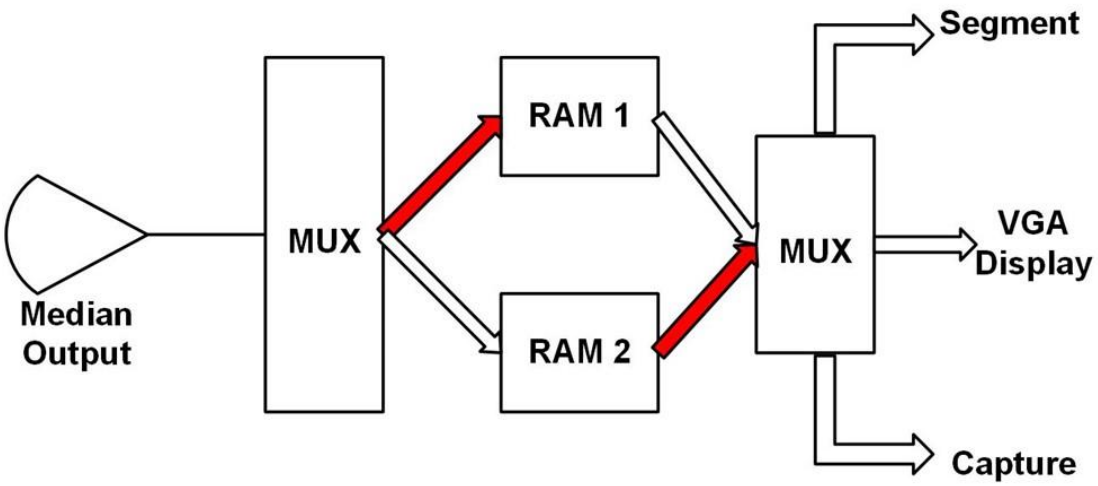


4.4. Ping-Pong RAM

The block RAMs in FPGA is not enough for multi-frame display, so we need a Ping-Pong cache to record camera data flow. The procedure can be summarized as: though 2 specific data buffers A & B, the received data flow will be allocated into two parts with the same time scale.

In the first clock period, input image pixels will be stored into RAM A, then in the second period, with the switch of MUX the input image are switched into RAM B while data in RAM A are read out at the same time for vga display. Reciprocal procedure will be repeated in the 3rd and 4th time periods and by rotating this, PP RAM is implemented.

Ping-Pong RAM achieves the matching between high-speed dataflow and low-speed display by complementing them into one model. The RAM makes the storage and processing of data smooth and fluent, which promotes the efficiency.



4.5. Gesture Compartmentalize

With this model, we need to clarify the centroid of hand to determine the position and do further compartmentalize through this. The formulae of calculation centroid are:

$$M_{00} = \sum x \sum y f(x,y)$$

$$M_{10} = \sum x \sum y x \times f(x,y)$$

$$M_{01} = \sum x \sum y y \times f(x,y)$$

$$X_c = M_{10} / M_{00}$$
$$Y_c = M_{10} / M_{00}$$

f(x,y) is the binarized pixel values, and Xc is the x-coord, Yc is the y-coord. The Bottom Line can be obtained by scanning the maximum row that "1" pixels appear. After getting the centroid of hand, we can compart the screen into 9 parts in accordance to the image shown below.

4.6. Page Turning:

The implementation of page turning is implemented my DE-10 NANO, in which we use UART to transfer data from FPGA to PC and control the PPT Turning. To implement the function of turning page, we design a system to analyze the signal and communicate with computers using DE-10 Nano. Generally speaking, there are 3 main steps in the procedure.

First, we use the position of frame lines to judge whether there is a turning command. Without change of the environment the in the scene of camera, these 4 lines will stabilize in small range. While if we swing our hand front of the camera, the positions of these lines will have an obvious change following the position of hand. Therefore, a large position transform is the foundation of judgement.

Second, during experiments and debugging we found that just one-time swinging will output many active signals. So, we have built a delay function to make sure this system can't give more than one command to computers continuously. Once we detect an active signal, the output will be turn down in about 1.5 second so that this problem was solved.

Thirdly, the final module is about the communication between FPGA board and computer. Using UART-USB port on the developing board we can realize the function. As we know, When we want

to switch page of PPT, “enter”, “space”, “down”, “right” and “N” on the qwerty all can be used to realize. So it is available to send the ASCII number of letter “N” (hex:E4) to turn down the page by UART. After our test, this device can work correctly.

5. Function Summary

1. Recognize Hand Region and Compart the Screen into 9 Parts,

2. Using Hand Gesture to do Page Turning,

3. Recognize Gesture of Number 1~5.

6. Parameters

Important Parameters:

6.1. Page Turning Rate:

If you wave hand in N times and the slides turn M pages, then Page Turning Rate can be described as

$$\mathbf{PTG = N/M}$$

Note that PTG can be both larger than 1 and smaller than 1. The more PTG is close to 1, the better performance the system has.

Test Times	Turned Pages	Calculated PTG
100	87	0.87

6.2. Recognition Correction:

If your gesture is identical to the displayed numerial number, then it means the recognition is correct. When doing K tests and there are T corrections, then Recognition Correction is:

$$\mathbf{RC = Y/K}$$

RC must be smaller than 1.

Numerial	Test Times	Corrections	Calculated RC
1	50	36	72%
2	50	40	80%
3	50	44	88%
4	50	42	84%
5	50	27	54%