AUTOMATIC INSPECTION SYSTEM FOR CMOS CAMERA DEFECT

Byoung-Wook Choi*, Kuk Won Ko, Kyoung-Chul Koh*** , Bok Shin Ahn******

*Dept. of Electrical Engineering, Seoul Nat'l Univ. of Technology, Seoul, Korea
**Dept. of Control and Measurement Engineering, Sun Moon Uni., Asanshi, Korea
***Dept. of Mechanical Engineering, Sun Moon Univ., Asanshi, Korea
****CEO of P&C Tech, Anyangshi, Korea

Abstract: This paper presents a development of automatic complementary metal-oxide-semiconductor (CMOS) camera inspection system to examine defects. The image capture board based on embedded linux using system-on-a-chip (SoC) and a complex programmable logic device (CPLD) is developed to capture CMOS sensor data. The captured sensor data is transferred to the host computer through TCP/IP socket communication to perform fast inspection. The liquid crystal display (LCD) monitor is used to load the inspection charts electrically so that it can reduce their changing time. The various algorithms for performing error inspection were implemented such as the line dot defect inspection and dim defect inspection. Experimental results reveal that the proposed system can focus the lens of CMOS within 5 seconds and recognize various types of defect of CMOS modules with precision and high speed.

Keywords: embedded systems, hardware, image sensors, image processing, error detection

1. INTRODUCTION

The SoC and digital technology development recently enabled the emergence of high-performance and low-power image devices, which has been primarily adopted by various handheld devices like a mobile phone and personal digital assistant (PDA). A CMOS image sensor is mainly used for such image devices. It provides improved image sensing speed, smaller form-factor and lower power consumption than a Charge Coupled Device (CCD) image sensor. In the past, CCD was preferred to CMOS because of its better image quality. However, CMOS has been increasingly adopted more recently because the improved semiconductor technology enables CMOS image quality to be much improved (Filippov, 2003).

With the increasing demand for Internet and mobile applications, there has been a considerable demand for high-speed production of a compact camera module (CCM). The major burdens of production of CCM are both assembly of lens module onto CMOS package and inspection of captured image quality of assembled CMOS. In most assembly processes, CCM is automatically assembled via surface-mount technology. However, lens focus adjustment and the inspection of image of assembled CCM are executed by human operators.

The inspection procedures by human operators are as follows. 1) Lens focus adjustment, 2) six more imaging tests: black defect, dark defect, white balance test, color test, color temp test, dim test and so on. These several human operations take more than 30 seconds for a highly-skilled human operator to do those tests. In order to reduce overall production time for CCM, we developed an automatic inspection method for finding defects of CMOS.

The developed inspection system consists of image capture board, inspection chart with illumination system and imaging processing algorithm. The development of a network based image capture board is introduced by using embedded Linux and fast Ethernet. The electrical chart is also developed to reduce handling time to change various inspection charts.

The performance of the developed inspection system and its algorithm are tested on samples of 10,000 CMOS sensors. The experimental results reveal that the proposed system can focus the lens of CMOS within 5 seconds as well as recognize various types of defects for CMOS modules with precision and high speed.

2. IMAGE CAPTURE BOARD

The target CMOS sensor is PO1030 CMOS image sensor made by Pixel Plus. The size of a sensor is 1/4.5 inches and has a color filter and micro lens. The output format supports 8 bit YCbCr/YUV, 9 bit Bayer Data, 5:6:5 RGB and 12 bit 8:8:8 RGB. This supports two modes such as video mode and still image capture mode. The video mode displays the images continuously. The still image capture mode supports the mechanical and electrical shutter functions. The maximum frame rate is 30 fps at 27 MHz. By changing the internal register values through I^2C, we could modify the image gains, balance, output formats and so on (Pixelplus, 2003). In order to acquire CMOS data, a network-based image capture board is required.

The board consists of a Micro Processing Unit (MPU), a Complex Programmable Logic Device (CPLD) and CMOS image sensor. In addition, an Operating System (OS) is used in order to control and develop the capture board more efficiently. Embedded Linux is used instead of Real-time OS and commercial embedded OS. Embedded Linux is open source and provides powerful networking and supports various hardware architectures. Additionally, the various application programs, development tools and libraries, which are provided free, enable the reduced development time and cost (Lennon, 2001; Hong, 2003).

CPLD is implemented by using very high speed integrated circuit hardware description language (VHDL), in order to store the image data from CMOS data into the RAM. The Linux-based application program to transfer the stored data to PC by using TCP/IP is also developed. This program consists of I^2C communication, GPIO (General Purpose I/O) handling and network socket programs. The Fig.1 shows the system block diagram.

Fig. 1. Block diagram of the system

The ARM7TDMI based S3C4530A, SoC, is used in the board in order to transmit the image data from the CMOS image sensor through TCP/IP. This integrates the Ethernet controller within an MPU for a network-based system. And it includes an I^2C interface (Samsung, 2001). CPLD is used to store image data from CMOS image sensor into RAM. In this CPLD, counter, buffer and comparator are programmed with VHDL, corresponding to the output format of CMOS image sensor. The captured board includes peripherals like RJ-45 port for networking, Ethernet driver, serial port for debugging, connectors between CMOS image sensor and capture board and so on. The system memory is the flash memory storing kernel and file system including user application, SDRAM running Linux and SRAM storing the image data from CMOS image sensor. Fig 2 shows a layout of capture board. The image capture program stores image data into SRAM and transmits the stored image data to PC through TCP/IP while controlling the CMOS image sensor. This program runs on uClinux and is developed on Linux environment by using a cross complier and library [linuxdevices, 2004; Choi, *et.al*.,2004].

As a TCP/IP socket server, the capture board includes the function of controlling I^2C and GPIO in order to control the CMOS image sensor. This program is activated while Linux is being booted. Once it is activated, it waits for connection with the client. When the connection is made, it initializes the CMOS, sets up the register value and captures data and sends an image to the client according to commanders from the client. Viewer program is the one to display the image data sent from capture board. As a network socket client, the viewer program sends the commands of controlling CMOS image sensor to a capture board and displays the bitmap, after changing YCbCr format data from the board to RGB format data.

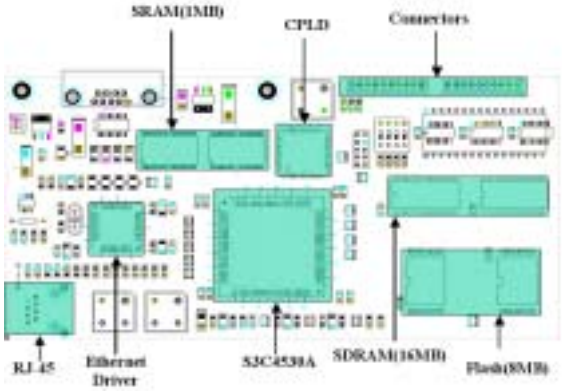


Fig. 2. System layout of the capture board

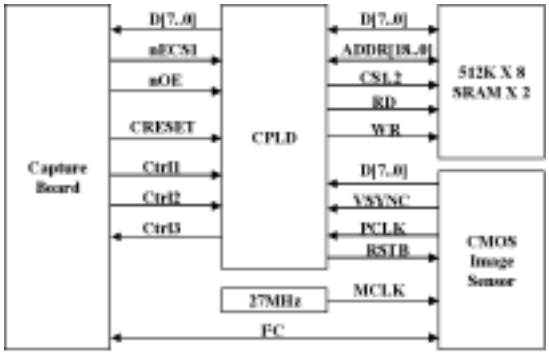


Fig. 2. Interface between the capture board and CMOS image sensor

3 IMAGE CAPTURE PROCESSING

3.1 CPLD Programming

CPLD is used to store the image data displayed from CMOS image into RAM. As shown in Fig 3, the internal register is set up for output format of PO1030 CMOS image sensor. VSYNC indicates the start of a frame in image data. HSYNC is the signal indicating a line included in frames. PCLK is the signal indicating a pixel of image data. As a PCLK signal is set up to be displayed in case HSYNC signal is high, HSYNC signal is not used as an input signal for CPLD.

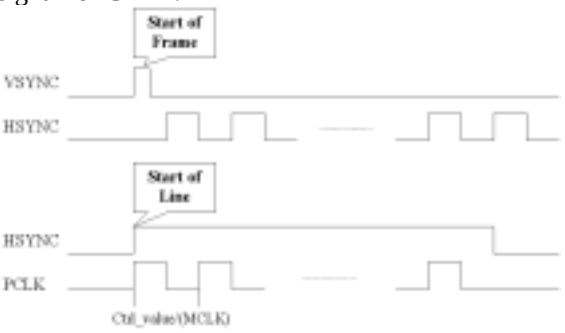


Fig. 3. Output format of CMOS image sensor

Fig 4 shows the block diagram of implemented logic. According to MPU's control signal, CPLD switches the control bus, data bus and address bus of SRAM. As to say, the input and output bus of SRAM is connected to MPU or CMOS image sensor.

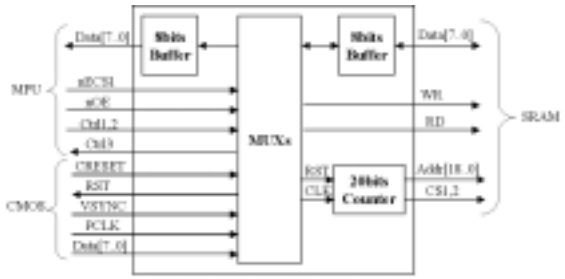


Fig. 4. Block diagram within CPLD

Ctrl1, Ctrl2 and CRESET are output signals from MPU and Ctrl3 is input signal into MPU. CRESET is the reset signal used to initialize CMOS image sensor, connected with RSTB. Because the operating voltage levels for MPU (3.3V) and CMOS image sensor (2.5V) are different, these are connected through CPLD. Ctrl1 is the signal to switch the input/output buses of SRAM with buses of MPU and CMOS image sensor. In high, MPU and SRAM would be connected. In low, CMOS image sensor and SRAM would be connected. Ctrl2 is the signal used to initialize the counter implemented in the CPLD. Ctrl3 is the signal transformed from high to low when a frame of image data is stored into SRAM.

CPLD consists of buffer, comparator and counter. The buffer protects the collision of data when sharing the data buses. The comparator makes the internal control signals by translating the control signals from MPU. The counter is connected to the address bus of SRAM, which is enabled by reading and storing the images in sequence. In this thesis, the size of a frame of image data is 640*480*2 Bytes = 600KB because output of CMOS is set up YCbCr format. Therefore, the counter is designed as 20bit. The inputs of the counter are reset and the clock and its output are the values counted. In accordance to signal of Ctrl1, the reset is connected with Ctrl2 of MPU or VSYNC of CMOS image sensor. In addition, the clock is connected with nECS1 of MPU or PCLK. The output of the counter is connected to the address bus and Chip Select (CS) of SRAM. Two SRAMs are used to cause the size of SRAM to be 512KB in this paper, and the most significant bit (MSB) of counter output is used as CS.

The operating steps are as follows.

1) In the order of High->Low->High, signal is made to CRESET. PO1030 would be reset while RST is in Low.
2) Transform the Ctrl1 signal to Low. Namely, the control bus and data bus of SRAM would be connected with CMOS image sensor.
3) In Low status of Ctrl1 signal, the image data from CMOS image sensor would be stored into RAM since VSYNC signal turns Active Low like the fig 5.

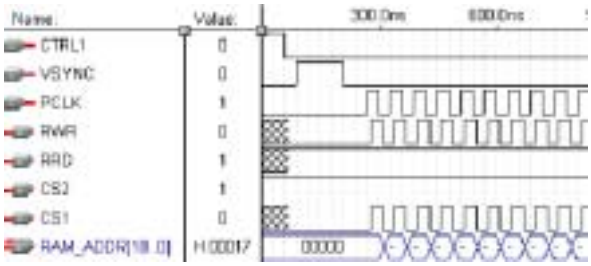


Fig. 5. Control signals when image data is stored into SRAM

4) As shown in Fig 6, when the VSYNC signal becomes Active Low again after storing the image data, the Ctrl3 signal would be transformed from High to Low.

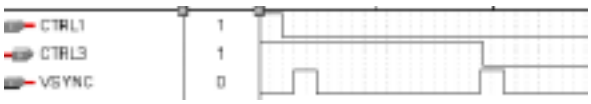


Fig. 6. Control signals when a frame of image data is stored into SRAM

5) S3C4530A would change the Ctrl1 signal as High when Ctrl3 turn Low, observing Ctrl3 through polling after Ctrl1 is transformed to Low.
6) As shown in Fig 7, the address would be initialized by making the Ctrl2 signal as Low->High->Low in order, resetting the counter.

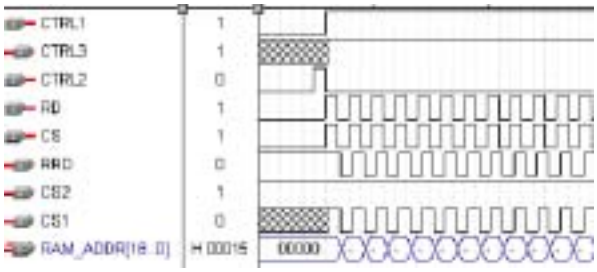


Fig. 7. Control Signal when reading the image data stored in SRAM

7) The image data stored in SRAM would be read.
8) The image data would be stored again into SRAM by transforming the Ctrl1 signal as Low.

3.2 Image Viewer Program

The image capture program stores image data into SRAM and transmits the stored image data to PC through TCP/IP, controlling the CMOS image sensor. This program runs on uClinux and is developed on Linux environment by using a cross complier and library (uClinux, 2004).

As a TCP/IP socket server, this includes the function of controlling I^2C and GPIO, in order to control the CMOS image sensor. This program is activated while Linux is being booted. Once it is activated, it waits for connection with the client. When it is connected, the CMOS image sensor is initialized, the register value of CMOS image sensor are set up, and the image is captured and sent to the client according to commanders from the client.

Fig. 8. The applied viewer program

Viewer program is the one to display the image data sent from capture board. As a network socket client, viewer program sends the command of controlling CMOS image sensor to a capture board and display the bitmap after changing YCbCr format data to RGB format data.

4. IMAGE PROCESSING ALGORITHM

For inspections, several charts should be used. LCD monitor is used to load the inspect chart electrically and it can reduce the changing time of each chart greatly. Fig. 9 shows the overall imaging inspection procedures for CCM. Each defect and its inspection algorithm are explained.

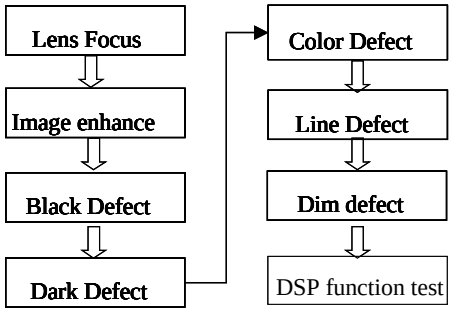
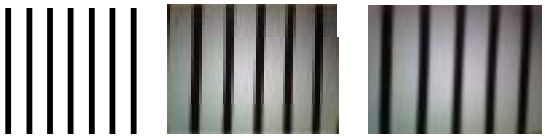


Fig. 9. Flowchart of overall CCM inspection

4.1. Focus and resolution inspection algorithm

The lens of CCM should be well focused to acquire a quality image. A chart image for focus inspection is shown in Fig 10(a). The image processing algorithm for focus inspection is based on the measurement of the sharpness of edge. The well-focused and defocused images are shown in Fig. 10.



(a) Chart (b) Well-focused (c) Defocused

Fig.10. Focus inspection algorithm

The sharpness of the edge is calculated by the variance of the data set, which can be extracted by the level of difference between the adjacent pixel through the horizontal line (Ballard, 1985). After assembling lens unit onto CMOS cell, the lens is unfocused and mostly is fully loosened. The motorized focusing unit rotates lens body while focusing. At every rotation step, the edge sharpness is measured and lens should be rotated until image has sharp line. The motorized focusing unit has 3 steps of rotation angle depending on the sharpness of edge. At small value of sharpness edge the image is not well focused, the motorized focusing unit rotates lens in large increments. Otherwise, a small rotation increment can produces a fine focus adjustment (Juyang, 1992).

4.2. Image enhancement algorithm

After the procedure of inspection for lens focus of CCM is well finished, the acquired image is shown in Fig. 11(a). Although the imager is well focused, the image is slightly distorted in brightness. The pixels in corner image are slightly darker than those in center because of lens characteristics. This distortion of brightness in image produces difficulties in image process when extracting the defect from acquired image. The multi-layered neural network is used to calibrate the brightness in image. It has a 2-hidden layered structure and 3 input nodes (horizontal pixel coordinate, vertical pixel coordinate, pixel value), 3 output nodes (horizontal pixel coordinate, vertical pixel coordinate, pixel value), and 30-30 nodes in two hidden layers. The neural networks make inverse

model of bright distortion via learning procedure (Burel, 1995). By using the inverse model for bright calibrated image, fully calibrated image of Fig. 11(b) is obtained.

(a) Distorted brightness (b) Bright calibrated image
Fig. 11. Image enhancement algorithm

4.3 Dark/bright dot defect inspection algorithm

Dark dot defect is caused by the defect of CMOS pixel which is bright or visible and can always be seen on an all black background. Bright dot defect is caused by the defect of CMOS pixel which is dark and can be always seen on an all white background. These pixels on CMOS are damaged or do not work correctly.

To detect dark dot defect, the background should be black or dark, the inspection chart for the dark dot defect should be black. Otherwise, to detect bright detect, the background should be white, the inspection chart for the bright dot defect should be white or bright. Fig (c) and (d) show the each defect image. The defect can be easily separated from background by simple threshold method.

(a) Defect of dark dotted (b) Defect of black dotted
Fig. 12. Dark dot defect inspection

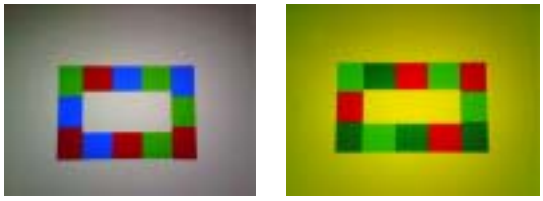
4.4 Line Dot defect inspection algorithm

Line dot defect is caused by the defect of two adjacent CMOS pixel which is always the same color vertically or always of different color dots horizontally as shown in Fig 13(a) and (b). Each R,G and B pixel in image are projected vertically and horizontally. The defect line has different value in projected pixel data and can be easily detected by simple threshold method.

(a) Line detect(pink) (b) Line detect(blue)
Fig. 13. Line dot defect inspection

4.5 Color inspection algorithm

Fig. 14(a) shows a color inspection chart which consists of red, green and blue. The each color R,G and B pixel value of acquired image is compared to known R,G and B pixel value. The color defect has pixels which can not be read color correctly and the color defected pixel has greatly different value. Fig. 14(b) shows the image of color defect.



(a) Color inspection chart (b) Color defected image
Fig. 14. Color detect inspection

4.5 Dim Defect inspection algorithm

Dim defect is caused by dust on the lens. The dusts or particles on lens make blurred dark area on images and the size of particle is unknown. This is called dim and it is very difficult to detect because the gray level difference between adjacent pixels is very small. Fig. 15 shows the gray level profile on dim defect. The self-organizing neural network is used to memorize this defected profile. The neural network memorizes profiles of this defect and distinguishes this defect from image after training procedure.

(a) Small dim defected (b) Large dim defected
Fig. 15. Dim defect inspection

4.6 White Balance and Auto iris test inspection algorithm

CCM has DSP to enhance the quality of captured images. The important image processing algorithms in DSP are white balance and auto-iris algorithm. White Balance Settings are an important function on most modern digital color cameras. They can mean the difference between a beautifully exposed and balanced digital image and an over or under exposed shot. To test white balance function in DSP, two color temperature filters are used. Blue filter (Kodak 80B 3400K->5500L) is used to increase color temperature and red filter (Kodak 85B 5500K-> 3400K) is used to reduce color temperature. The two filters are located in front of lens under white background, then, image is captured.

When the DSP do white balance correctly, Fig 16(a) image is acquired. Fig. 16(b) is the defect image when DSP do not work white balance. The color value in acquired image is checked pixel by pixel and the defect is detected. Auto-iris is also important DSP function to control brightness of images via changing exposed time. Fig. 16 shows the two images which are good auto-iris functioned image and defect image. Under given bright condition, two images are captured. One is auto-iris-on-image and the other is auto-iris-off-image. These two images are compared and the average brightness of image is compared. The difference in average brightness is an important feature to detect the defect of auto-iris function in DSP.

(a) Good balanced image (b) White balance error
Fig. 16. White balance inspection

(a) Good Auto-Iris image (b) Auto-Iris defect image
Fig. 17. Auto-Iris inspection

5. EXPERIMENTS AND DISCUSSIONS

To evaluate the performance of the proposed method, a series of experiments was performed for 2500 samples. Table 1 shows the inspection results. For 2500 samples, black/dark/line/color defects are perfectly detected, and there are only 2 error in focus inspection /adjustment because lens bodies do not work correctly. However, the success rate of dim inspection is 98.2% success rate. The large particle on the lens can be detected with a good accuracy, but the particle within 2-3 pixel size can not be easily detected because the gray level difference between adjacent pixel is nearly 3-4 gray level value. To detect this small dim defect, we should develop new method to detect this small dim defect. The total inspection time is 29 seconds to execute all inspection procedures include loading and unloading CCM and the processing time is only 8 seconds.

Table 1. Inspection results

	Focus	Black /Dark	Color	Line	Dim	DSP function
Success	2498	2500	2500	2500	2455	2500
Fail	2	0	0	0	45	0
Success rate	99.92	100	100	100	98.2	100

6. CONCLUSIONS

In this paper, an automated vision system and vision algorithms for compact camera module has been developed. The developed electric chart and image enhancement algorithm reduce the subsequent image processing algorithm to calibrate bright distortion and to reduce the time to change each chart for several inspections. As a result, simple image processing algorithms such as a binary threshold method, a simple edge extracting method, and pixel valve comparison method are easily implemented. The experimental results show the developed vision system can detect the defect of CCM with a precision and fast processing time less than 30 seconds of human operators in real CCM production line.

7. REFERENCES

Ballard, D. H. (1985*), Computer Vision*, Prentice Hall.

Burel,,G., F. Bernard and W. J. Venema (1995), Vision feedback for SMD placement using neural network, *IEEE international conference*, pp 1491-1496.

Choi, B. W , E. C. Shin, and S. Yi (2004), Web-based Building Automation System using Embedded Linux, *J. of ICASE*, **10**, 335-341

Filippov, A. (2003) Reconfigurable High Resolution Network Camera, *Proc,11th Annual IEEE Symp, on Field-Programmable Custom Computing Machines*

Juyang Weng, Paul Cohen and Marc Herniou (1992), Camera Calibration with Distortion Models and Accuracy Evalution, *IEEE Trans, on Pattern Analysis and Machine Intelligence*, **14**, No. 10.

Hong, S. (2003) Embedded Linux Outlock in the Post Industry, *Proc. 6th Int. Symp. On Object-Oriented Real-Time Distributed Computing*

LinuxDevices (2003), http://www.linuxdevices.com

Lennon, A. (2001) Embedded Linux, *IEE Review*, pp 33-37

Pixelplus (2003), *PO1030 Data Sheet*

uClinux (2004), http://uclinux.org/description

Samsung (2001), *S3C4530A User's guide*