PROGRAMMABLE DSP PLATFORM FOR DIGITAL STILL CAMERAS

Klaus Illgner[1], Hans-Georg Gruber[2], Pedro Gelabert[2], Jie Liang[1], Youngjun Yoo[1], Wissam Rabadi[2], Raj Talluri[1]

[1] Texas Instruments, Incorporated
8330 LBJ Freeway, Dallas, TX 75243
[2] Texas Instruments, Incorporated
12203 SW Freeway, Stafford, TX 77477

ABSTRACT

This paper presents a programmable DSP platform for Digital Still Cameras based on the Texas Instruments TMS320C54x family. One major advantage of this platform is that, after capturing an image from a CCD sensor, processing the raw image, and compressing the image for storage is performed on the Digital Signal Processor (DSP). This provides a short shot-to-shot delay and a high degree of flexibility. The system realized also allows instant viewing and selective storing of captured images. This paper outlines the various processing stages necessary to take the raw CCD data and produce a JPEG compressed bit stream and highlighting the advantages of DSPs for this application. The programmable nature of this platform allows for the exploration of different image processing and compression techniques. The low power nature of the digital signal processor provides long battery life.

1. INTRODUCTION

Recently, Digital Still Cameras (DSCs) have become a very popular consumer appliance appealing to a wide variety of users ranging from photo hobbyists, web developers, real estate agents, insurance adjusters, photo-journalists to everyday photography enthusiast [1][2]. Recent advances in large resolution CCD arrays coupled with the availability of low power DSPs has led to the development of DSCs that come quite close to the resolution offered by traditional film cameras [3]. These DSC offer several additional advantages to traditional film cameras in terms of data storage, manipulation, and transmission. The digital representation of captured images enables the user to easily incorporate and transmit the images into any number of electronic media. The ability to instantly view and selectively store captured images provides the flexibility to minimize film waste and instantly determine if the image needs to be captured again. With its digital representation the image can be corrected, altered, or modified after its capture.

2. DSC SYSTEM

Figure 1 shows the various functional blocks in a typical DSC system. Most DSCs use a CCD imager to sense the images. The driver electronics and the Timing Generator circuitry generate the necessary signal to clock the CCD. Correlated Double Sampling and Automatic Gain Control electronics are used to get a good quality image signal from the CCD sensor. This CCD data is then digitized and fed into the DSC Engine. All the image processing and image compression operations are performed in the DSC engine. On most DSCs the user has the ability to view the image to be captured on the LCD display. The captured images are stored on the Flash memory for later use. Most DSC systems also provide an NTSC/PAL video signal to view the captured images (also the preview images) on a TV monitor. The current DSCs also provide a number of ways to connect to the external PC or printer through an RS 232 or a USB port.

Future DSC systems are expected to be even more versatile with the ability to annotate images with text/speech. Including a modem and TCP/IP interface allows for directly connecting to the Internet. Future DSCs will also run more complex multi-tasking operating systems to schedule the various real-time tasks [4].

3. IMAGE ACQUISITION

A typical DSC has to perform multiple processing steps before a high quality image can be stored. The first step is the image acquisition. The intensity distribution reflected from the scene is mapped by an optical system onto the imager. Nowadays, most

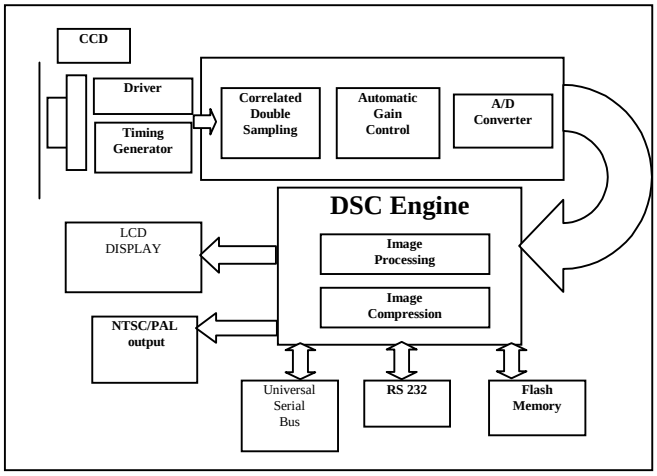

Figure 1 A DSC System Block Diagram

cameras use CCDs although CMOS imagers are also used in some [5]. The image captured by the CCD sensor has each pixel masked by a color filter to provide a color image. This raw CCD image is normally referred as a Color Filtered Array (CFA) [6]. The masking pattern of the CCD array as well as the filter color primaries vary between different manufactures. In DSC applications, the CFA pattern that is most commonly used is an RGB Bayer pattern that consists of 2x2 cell elements which are tiled across the entire CCD-array [7]. Figure 2 depicts a subset of this Bayer pattern in the matrix block following the CCD camera. The output signal of the CCD is digitized with a 10 or 12-bit A/D converter.

4. IMAGE PIPELINE

The CFA data needs to undergo significant amount of image processing before the image can be finally presented in a usable format for compression. All these processing stages are collectively called as the “image pipeline”. A typical image pipeline in a DSC is shown in Figure 2. As can be seen, a typical DSC has to perform multiple processing steps before a high quality image can be stored. Most of these tasks are multiply-accumulate (MAC) intensive operations [8]. The TMS320C54x DSP is well suited to perform these tasks efficiently and generate a high quality image that is close to image quality offered by traditional film from the raw CCD data. We outline the various image pipeline processing stages below.

Black Clamp

To optimize the dynamic range of the pixel values represented by the CCD imager, the pixels representing black need to be corrected since the CCD-cell still records some non-zero current at these pixel locations. The black clamp function adjusts for this difference by subtracting an offset from each pixel value, but clamping/clipping to zero to avoid a negative result.

Lens Distortion Compensation

Imperfections in the lens introduce non-linearities in the brightness of the image. These non-linearities reduce the brightness between the center of the image to the border of the image. The image pipeline compensates for the lens by adjusting the brightness of each pixel depending on its spatial location.

Fault Pixel Interpolation

Large pixel CCD-arrays may have defective pixels. The missing pixels are interpolated with an interpolation scheme to provide the rest of the image pipeline data values at each pixel location. The manufacturer of the CCD sensor typically provides the locations of the missing pixels. The faulty pixel locations can also be computed by the DSC engine offline by first capturing an image with the lens cap closed. The faulty pixels are imaged as "white spots" and the rest of the image is dark. The faulty pixel locations can then be identified with a simple threshold detector and stored in memory. During the normal operation of the DSC the image values at the faulty pixel locations are computed by an interpolation technique.

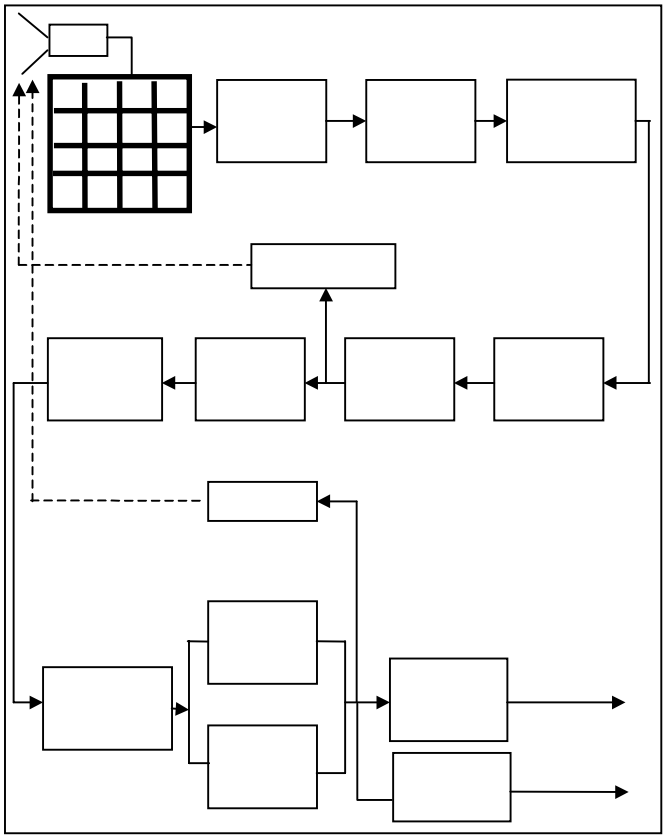

Figure 2 DSC Image Pipeline

the illumination when viewing a picture. This results in different color appearance that is typically seen as bluish appearance of a face or reddish appearance of the sky. Also, the sensitivity of each color channel varies such that grey or neutral colors are not represented correctly. To compensate for these unbalances in colors, the gain of the red, green and blue channels is equalized. This is accomplished by computing the average brightness of each color component and to determining a scaling factor for each color component. Since the illuminants are unknown, a frequently used technique just balances the energy of the three colors. This equal energy approach requires an estimate of the unbalance between the color components.

CFA Interpolation

Due to the nature of a color filtered array, at any given pixel location we only have one color pixel information (R, G or B in the case of a Bayer pattern). However, the image pipeline needs full color resolution (R, G, and B) at each pixel in the image. Therefore, the two missing pixel colors are reconstructed by interpolating the neighboring pixels [9].

Gamma Correction

Display devices used for image viewing and printers used for image hardcopy have a non-linear mapping between the image gray value and the actual displayed pixel intensities. To correct for these differences, a Gamma correction stage compensates for the differences between the images generated by the CCD sensor and that the image displayed on a monitor or printed into a page.

Color Space Conversion

Typical image compression algorithms such as JPEG operate on the YCbCr color space. Therefore a color space conversion is performed to transform the image from an RGB color space to a YCbCr color space. This conversion is a linear transformation of each Y, Cb, and Cr value as a weighted sum of the R, G, B values at that pixel location.

Edge Enhancement

The nature of CFA interpolation filters introduces a low pass filter that smoothes the edges in the image. To sharpen the images, the image pipeline uses an edge detector to compute the edge magnitude in the Y channel at each pixel. The edge magnitude is then scaled and added to the original luminance (Y) image to enhance the sharpness of the image.

False Color Suppression

Note that the edge enhancement is only performed in the Y channel of the image. This leads to misalignment in the color channels at the edges resulting in rainbow-like artifacts. Suppressing the color components, Cb and Cr, at the edges reduces these artifacts.

Auto Focus

It is also possible to automatically adjust the lens focus in a DSC through image processing. These auto focus mechanisms operate in a feed back loop. They perform image processing to detect the quality of lens focus and move the lens motor iteratively till the image comes sharply into focus.

Auto Exposure

Due to the varying scene brightness, to get a good overall image quality, it is necessary to control the exposure of the CCD. This is also accomplished in the DSC by sensing the average scene brightness and appropriately adjusting the CCD exposure time and/or gain. Similar to auto focus, the DSP performs this operation also in a closed loop feed back fashion.

Image Compression

Most DSCs are limited in the amount of memory available on the camera. Hence image compression is employed to reduce the memory requirements of the captured images. Typically compression ratios of about 10:1 to 15:1 are used. Most of the existing DSCs use JPEG compression [10]. The DCT and Huffman encoding stages dominate the computations in JPEG. Our current implementation is capable of performing an 8×8 DCT in 1220 cycles or 12.20 μsec.

Future DSC will likely migrate to JPEG2000 standard - which employs a wavelet-coding scheme.

5. TMS320C54X DSC Implementation

This image pipeline was implemented on a TMS320C549 DSP [11]. Figure 3 shows a picture of the prototype C54X DSC system. This system interfaces to the CCD/CMOS module directly. The DSP reads the data from the sensor, processes the raw sensor data and writes it to the SDRAM. The built-in NTSC/PAL encoder chip on the board allows direct display of the processed picture on the TV monitor. In implementing the image pipeline on the DSP, we designed an SDRAM controller that efficiently fetches the image data from the external memory to the on chip memory in 16 x 16 blocks. All the image pipeline operations including JPEG compression can be performed on the C54x on this 16 x 16 block of pixels and then the compressed bit stream is written out to the SDRAM.

Figure 3 A TMS320C54x DSC System

On this system, all image pipeline operations can be executed on-chip since only a small 16 x 16 block of the image is used. Therefore, the TMS320C549 is well suited due to its large on-chip memory [32K x 16-bit RAM and 16K x 16-bit ROM]. In this way the processing time is kept short, because there is no need for external high-speed memory. Furthermore, this device offers high performance (100 MIPS) at low power consumption (0.45mA/MIPS) [11].

Table 1 illustrates a detailed cycle count for the different stages of the image pipeline software. Due to the efficiency of the C54x instruction set and architecture, the entire image pipeline including JPEG takes about 150 cycles/pixel. Hence, a 1 Mega-pixel CCD image can be processed in 1.5 seconds on a 100MHz C54x. This offers about 2 second shot-to-shot delay including data movement from external memory to on-chip memory.

Current DSCs also provide the ability for the user to view the captured images on LCD screen on the camera or on an external TV monitor. Since the captured images are stored on a Compact Flash Card as JPEG bitstreams, a playback mode software is also provided on the DSP. This playback mode software decodes the JPEG bit stream, scales the decoded image to the appropriate spatial resolution, and displays it on the LCD screen and/or the external TV monitor. Our C54x playback mode software executes in 100 cycles/pixel. This offers a 1 second playback of a Mega-pixel image.

Table 1: C54x Performance

Task	Cycles/Pixel
Preprocessing (Black Clamping, Gain, White Balance, Gamma Correction	22
Color Space Conversion	10
Interpolation	41
Edge Enhancement & False Color Suppression	27
Total	90
4:1:1 Decimation & JPEG Encoding	62
Total	152

Table 2 shows the program and data memory requirements to process the image pipeline and compress the image with JPEG standard. This code density allows the complete image pipeline software to reside on-chip. Having the complete software on-chip reduces external memory accesses and allows the use of slower external memory. This reduces overall system cost and lowers system power consumption.

Table 2: C54x Memory Requirements

Memory	K Bytes
Program	1.7
Data	4.6

Currently, most of the image pipeline operations are non-standardized. Having a programmable DSC engine offers the ability to upgrade the software to conform to new standards or improve image pipeline quality. Unused performance can be dedicated to other tasks, such as human interface, voice annotation, audio recording/compression, modem, wireless communication, etc.

6. CONCLUSIONS

In this paper, we have demonstrated an innovative approach to implementing a digital still camera image pipeline software on the TMS320C5000 platform. Our results show the feasibility of implementing the image pipeline on a TMS320C549. Also, the programmable nature of this approach offers the flexibility needed in the DSC engine to allow the incorporation of innovative image processing algorithms. Finally, the proposed approach offers the performance and low power that is crucial in the development of photo quality images on a portable, battery-operated digital camera.

7. REFERENCES

[1] James Martin, “Snap Judgments”, PC World, Feb. 1998.

[2] “Digital Still Cameras Develop into A New Semiconductor Market”, Dataquest Focus Report, Feb., 1997.

[3] J. Eyre and J. Bier, “DSP processors hit the mainstream,” *Computer*, Aug. 1998, pp. 51-59.

[4] S. Venkataraman, K. Peters, and R. Hecht, “Next Generation Digital Camera Integration and Software Development Issues”, Digital Solid State Cameras: Design and Applications, 28-29 January 1998, San Jose, CA, Proceedings of SPIE, Vol. 3302.

[5] Don W. Lake, “CMOS Image Capture for Digital Stills Cameras”, IS & T’s 1998 PICS Conference, May ’98, Portland, Oregon.

[6] Gerald C. Holst, “CCD arrays, Cameras and Displays”, SPIE Optical Engineering Press, 1996.

[7] B. E. Bayer, “Color Imaging Array”, U. S. Patent No. 3971065, Eastman Kodak Inc.

[8] Rajeev Raman, “Image Processing Data Flow in Digital Cameras”, Digital Solid State Cameras: Design and Applications, 28-29 January 1998, San Jose, CA, Proceedings of SPIE, Vol. 3302.

[9] James E. Adams, “Interactions between color plane interpolation and other image processing functions in electronic photography”, Cameras and Systems for Electronic Photography and Scientific Imaging, Mar 1995, San Jose, CA, Proceedings of SPIE, Vol. 2416.

[10] W. B. Pennebaker and J. L. Mitchell, “JPEG Still Image Data Compression Standard”, Van Nostrand Reinhold, 1993.

[11] Texas Instruments official TMS320C54x www site:

http://www.ti.com/sc/docs/dsps/products/c5000/c54x/index.htm