

A Study of Digital Color Imaging System Design for Embedded Systems

Kenny Kuan Yew Khoo and Yong Haur Tay

Computer Vision and Intelligent Systems (CVIS) Group
Universiti Tunku Abdul Rahman, MALAYSIA.

kennykhooky@gmail.com, tayyh@mail.utar.edu.my

Abstract

In this paper, we examine design issues in integrating digital color imaging onto embedded systems. We aim to design a suitable digital color imaging system that is low cost, transmit lossless image and only requires minimum interfacing bandwidth. We inspect from the aspects of optical format, type of image sensor, video capturing format, real-time video codec and system interface bandwidth. This paper gives an overview and practical guide in designing and developing a color imaging systems for embedded systems. We found that given currently available embedded systems chipset and video codec, we are not able to design a lossless color imaging system beyond 7.6 Mpixel resolutions. This is due to the limitation of bandwidth and video compression technology.

Key words: imaging sensor, embedded system, digital color, systems design

1. Introduction

In the past, color imaging systems interface with core processing systems using analog means, e.g. RF connector, composite video, s-video, and component video. One disadvantage of analog interface is that single video input port can only connect to an analog video input. Furthermore, unrecoverable video signal loss could happen if the interface medium lengthens. Currently, embedded systems which support analog video input must have additional video decoder chipset, e.g. BT261 or BT868 from Conexant Inc [1].

Today most of the color imaging system communicates with the core processing systems using digital interface. The most common digital interface is universal serial bus (USB), Ethernet, and Firewire (or IEEE1394). The benefit of digital interface is that many of them support multi-drop, where a single digital host interface can be connected

to many devices. It is also much easy to recover and reconstruct the loss during transmission.

The problems of digital interface is that embedded systems need extra processing power to handle the data controlling, data transferring and video encoding. Some of the digital handling might use up all the processor power. Hence, analog interface is still the best choice for slow embedded system.

At the architect level, the criteria of an ideal digital color imaging system for embedded systems are:

- Lossless image data.
- Minimum interface bandwidth.
- Real time video capture
- On time image data transfer

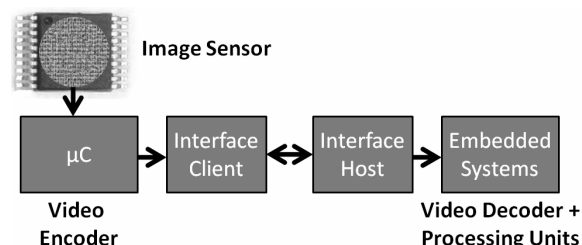


Figure 1: Imaging systems design block

Figure 1 shows the design block for the whole imaging system that consists of imaging sensor, video codec, and systems interface.

Section 2 explains the basic idea of optical format. Section 3 describes all types of color imaging. Section 4 reports the test result of all the lossless video codec comparison. Section 5 shows the finding of available systems interface chipset for embedded system.

2. Optical format

Designing or choosing digital image sensors requires understanding of optical format. Optical format is simple in principle but always misunderstood and misreported in practice [2]. Today standardization of lens systems came from traditional image sensor, Vidicon tubes. Vidicon tubes produces in standard sizes like 1 inch, 1/2 inch. It had a limited useable image area, a 1 inch Vidicon tubes did not have 1 inch of good visible image. In fact, the good of visible image is about 16 mm only. Therefore, today lens manufacturer, making 1 inch optical lens is only provide about 16 mm of useable diameter. A standard was set: 1 inch optics became 16 mm; 1/2 inch optics became 8 mm.

Today some manufacturers may bend the truth on their optical format to fit within the lower-cost lens system. For example, a VGA imager (640 x 480 pixel resolution) with a 6-micron square pixel calculates to 0.30 inch optical format. While this fits between the 1/2 inch and 1/3 inch optical format, the optimal solution is probably to use 1/3 inch optics. Using the smaller optical format will decrease lens cost but it may result in "vignetting," in which the corners of the image are dark since it is outside the lens capture area.

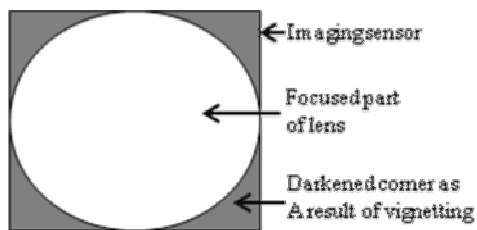


Figure 2: Vignetting effect in imaging sensor

The final decision between using a 1/2 in. and 1/3 in. system will be a cost and performance trade-off for the engineer, meaning that either system designers should calculate the exact optical format or imager companies should start listing both the actual optical format in decimals as well as the "recommended" lens system to use with the imager.

Since most imager data sheets don't list actual optical format, but do list the size of their pixels in microns, a more helpful equation is to convert the pixel size, and array size, directly to optical format. The equation for this is:

$$\text{Optical Format} = \sqrt{\frac{(wp)^2 + (hp)^2}{16}} \times 1000$$

Where,

- w = width of array (number of pixels)
- h = height of array (number of pixels)
- p = pixel size (microns)

3. Image sensor

Two most popular color image sensor technologies today are charge-coupled device (CCD) and complimentary metal-oxide semiconductor (CMOS). There have been some noticeable differences between CCD and CMOS sensors [3].

- CCD sensors create high-quality, low-noise images.
- CCDs use a process that consumes more power.
- CCDs consume as much as 100 times more power than an equivalent CMOS sensor.
- CMOS chips can be fabricated on just about any standard silicon production line, so they tend to be extremely inexpensive compared to CCD sensors.

Table 1 shows that there are several main types of color image sensors, differing by the means of the color separation mechanism.

- Bayer sensor
It is low-cost and is the most commonly used image sensor. By using a Bayer filter it passes red, green, or blue light to selected pixels, forming interlaced grids sensitive to red, green, and blue. The image is then interpolated using a demosaicing algorithm [4].
- Foveon X3 sensor
It uses an array of layered sensors where every pixel contains three stacked sensors sensitive to the individual colors.
- 3CCD sensor
It uses three discrete image sensors, with the color separation done by a Dichroic prism. It produces the best quality, and is generally more expensive than single-CCD sensors.

Table 1: color imaging sensor separation mechanism

Sensor Type	Cost	Lossless
Bayer	Low	Yes
Foveon X3	Average	No
3CCD	High	No

Color imaging sensor specification is [5]:

- Resolution (e.g. 352x288, 640x480, 1280x1024, 1600x1200, 2560x2048, 3264x2448)
- Resolution ratio (e.g. 4:3, or 5:4 or 11:9)
- Optical format (e.g. 1 inch, 1/2 inch, 1/2.5 inch, 1/3 inch, 1/4 inch, 1/4.5 inch, 1/5 inch)
- Color format (e.g. RGB-32bit, YUV-8bit)
- Output pin (e.g. 22, 24, 28, 36, 40, 42, 48, 52)
- Pixel size (e.g. 2, 2.2, 3.18, 3.6, 4.2, 5.6, 6, 9)
- Frame per second (e.g. 15, 30, 60, 78, 90, 120)

Color imaging sensor preferable for embedded systems will be CMOS technology with Foveon X3

color separation mechanism. This is because it is less expensive than CCD sensors and it consumes less power. Current available CMOS color imaging sensor is up to 8 Mega pixels from OVT Inc. and Micron Inc.

4. Video Capture format and Video codec

Nowadays all color imaging sensor is using Truecolor format, where each pixels is represented in 24 bit of color or 16.78 million of color. The human eye is popularly believed to be capable of discriminating between as many as ten million colors only. Therefore Truecolor imaging sensor is capable enough to display high quality photographic images or complex graphic for human eye.

Imaging systems is generally used to capture a sequence of image. Capturing speed within 15 frames per second (fps) to 30 fps is good for human eye motion response. If the rotating blade's angular velocity exactly matches that response time each revolution would superimpose the previous one and the blade would appear to stand still.

To achieve virtually real time in color imaging systems for human eye, 24bit RGB color with 15 fps specification must be fulfill.

The table 2 analyzes the transfer rate according to the image resolution for raw video data at 15 fps capturing speed to achieve virtually real time for human eye.

Table 2: Color imaging systems resolution and the transfer rate at 15fps capturing speed

Resolution		Mpixel	MB /Sec
X	Y		
352	288	0.1	4.56
640	480	0.31	13.82
1280	1024	1.31	58.98
1600	1200	1.92	86.40
2048	1536	3.15	141.56
2560	2048	5.24	235.93
3264	2448	7.99	359.56

Refer to table 5 in Section 5, there is no way to directly transfer a 3 Megapixel raw image data to embedded systems in real time, because there is no digital interface in embedded systems to support such large transfer rate. Hence color imaging systems must have video capture format or video codec locally to compress the sequence of image before transmit it to embedded systems

Video capture format must be lossless. Ideally imaging device should deliver the actual image data to the core systems before any image preprocessing

Below are the available lossless video codec:

- CorePNG (Version 0.8.2, date 18-Jul-2005)
- x264 (Revision 664, date 7-Jul-2007)

- HuffYUV (Version 2.1.1, date 23-Aug-2000)
- Lagarith (Version 1.3.13, date 12-Nov-2006)
- LCL Libraries (version 2.2.3, date 20-Sep-2000)
- MSU (Version 0.6.0, date: 19-Sep-2005)
- TSCC Codec (Version 2.0.6, date: 27-Jun-2005)

All above video codec had compared to find out the best video codec as video capture format and the best compression ratio relatively to the compression speed. Only absolutely lossless video codec were studied in this comparison.

All video codec were tested in RGB24 color spaces. All necessary conversion was done using StaxRip and VirtualDub video convertor software. There are 4 video file to test. Each video file is 30 second length at 25 fps speed or 750 frames.

All the conversion tests were run on PC of the following configuration:

- Processor : Intel Duo T2300, 1.66 GHz
- Memory : 1.25 GB of RAM
- Storage : 5400rpm hard disk
- Operating System : Windows XP Pro SP2

Table 3: Comparisons of compression results based on file size

Codec	File size after conversion (MB)			
	DieHard 4.0	Rata touille	Shrek 3	Trans former
Raw data	184	181	138	184
x264	42	31	25	40
HuffYUV	116	119	90	121
Lagarith	72	84	61	85
LCL	159	169	130	182
MSU	38	31	18	40
CorePNG	112	116	91	132
TSCC	140	114	74	152

The table 3 shows the video file size after the necessary conversion. Compression ratios are calculated by divide raw data file size with the file size after conversion. The compression ratios are use as a part of the video codec comparison matrix graph.

Table 4: Comparisons of time to convert 30 second raw video into particular video format

Codec	Time to convert from raw data (Second)			
	DieHard 4.0	Rata touille	Shrek 3	Trans former
x264	24	19	17	20
HuffYUV	13	16	8	10
Lagarith	15	15	11	15
LCL	25	23	19	24
MSU	50	46	25	40
CorePNG	93	125	76	118
TSCC	32	35	24	41

Meanwhile all the conversion time is recorded in Table 4. Conversion speed ratios are calculated by

divided total time (30 second) with the particular conversion time. The compression speed ratios are also used as a part of the video codec comparison matrix graph.

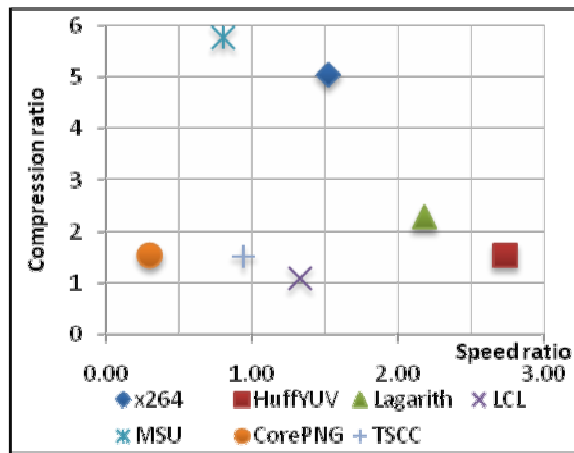


Figure 3: Video codec comparison matrix graph-average compression ratio vs. average compression speed ratio

Figure 3 indicates that CorePNG, TSCC, and MSU codec are not able to compress video on time; their speed ratio is less than 1. Hence, they are not suitable for use in digital color imaging system.

Assuming maximum transfer rate for transferring the data to embedded systems is 6 megabytes per second (MBps) and that at less 2.5 Mpixel video data to transfer, then the required video codec compression ratio must be more than 2. Therefore LCL and HuffYUV are not suitable to use in digital color imaging systems too.

The following conclusions were drawn from the figure 3:

- Video codec as video capture format, the overall clear winner is Lagarith. It is relatively good speed and high compression for various presets.
- Maximum compression ratio the overall winner is x264 relatively to real time speed.

5. Systems Interface

Table 5 lists out all available peripheral interface buses for PC platform. Notice that some of the very high speed peripheral interface is currently not designed for embedded systems platform.

Table 5: Peripheral interface bus

Peripheral interface bus Note: * not design for embedded systems	Transfer Rate	
	Mbps	MBps
USB Low Speed	1.54	0.19
USB Full Speed	12	1.50
USB Hi-Speed	480	60.00
FireWire 100	98	12.29

FireWire 200	196	24.58
FireWire 400	393	49.15
FireWire 800 *	786	98.30
Ethernet (10base)	10	1.25
Fast Ethernet (100base)	100	12.50
Gigabit Ethernet (1000base) *	1000	125.00

Presently, USB is the most popular peripheral interface for consumer electronics. However, most of the industrial cameras are using Ethernet as their peripheral interface. The reason behind it is that Ethernet allow transfer rate up to 1000 megabits per second (Mbps) or 125MBps and it can virtually support up to 16.7 Mpixel video camera data at 15 fps capturing speed.

Firewire is not widely used. Currently Windows XP SP2 lacks the supports for FireWire; every FireWire device that is attached to this OS will only run at S100 (100 Mbit/second) speed [6]. Apple Inc., who is one of the early adopters of FireWire technology, has replaced Firewire with USB for its iPod device, due to space constraints and to support wider device compatibility [7].

Table 6: MCU with peripheral interface bus controller intergraded

Com-pany	Type	Micro-controller	MCU speed	Price	Trans-fer Rate
Atmel	USB	AT83 C5134	8 MIPs	USD 1.30	12 Mbps
Free scale	USB	MC68HC 908JW32	8 MIPs	USD 3.15	12 Mbps
Infi neon	USB	SAFC165 UTAHLF	18 MIPs	N/A	12 Mbps
Micro chip	USB	PIC18 F4550	12 MIPs	USD 6.06	12 Mbps
TI	USB	TUS B3410	4 MIPs	USD 4.61	12 Mbps

Table 6 lists out the current available MCU which had intergraded USB client controller. To design color imaging systems, USB systems interface is preferred, because most MCU manufacturers target consumer peripheral market and these MCU only support the wider compatibility interface, i.e. USB.

Assuming an imaging system uses MSU as the video codec and uses one of the USB controllers from Table 6 as system interface, at 15 fps capture speed, this system can only deliver 0.2 megapixel lossless image.

Table 7: Peripheral interface bus controller for add-on in MCU

Com-pany	Type	Controller	Host inter-face	Price (USD)	Trans-fer Rate
Cy-press	USB	CY7 C68000	UTMI	2.55	480 Mbps

Maxim	USB	MAX 3420E	SPI	2.65	12 Mbps
Micro chip	Ether-net	ENC 28J60	SPI	2.96	10 Mbps
TI	Fire-wire	TSB43 AA82A	IO	13.89	400 Mbps

Table 7 lists out all available interface-controller chips. It is an add-on co-processor at MCU to enable peripheral interfacing. It is an alternative solution to achieve high resolution for the color imaging systems.

Again, if the imaging systems uses MSU video codec, and uses Cypress chip (Table 7) as the add-on co-processor, at 15 fps capture speed, this system can delivery up to 8.0 megapixel lossless image.

Table 8: Peripheral interface bus controller for CPU/miniPC platform

Com-pany	Type	Controllor	Host inter-face	Price	Trans-fer Rate
Broad-com	Ether-net	BCM5703	PCI	N/A	1000 Mbps
Davi-com	Ether-net	DM9702	PCI	N/A	1000 Mbps
Intel	Ether-net	82540EM	PCI	N/A	1000 Mbps
Mar-vell	Ether-net	88E8062	PCI	N/A	1000 Mbps

Table 8 lists out all available interface controller chip for CPU/miniPC platform. It is not designed for MCU. Even though it can be the solution to achieve very high resolution for the color imaging systems for CPU/miniPC platform, but few engineers are doing so for consumer product due to high production cost.

If the imaging system uses MSU video codec, and uses one of the peripheral interface bus controller from Table 8, at 15 fps capture speed, this system can delivery up to 16.7 megapixel lossless image.

6. Conclusion

To design a low cost color imaging systems for embedded systems, we need to carefully choose imaging sensor alongside optical format, because it will affect the cost of lens. It is almost impossible to design color imaging systems for embedded systems with real time capability and lossless video design idea, because current available lossless video codec and peripheral interface bus had the limitation on compression rate (limit at 5.7 ratio with MSU codec) and transfer rate (limit at 60MBps with high speed USB).

To design the system at the edge of the limitation, theoretically this color imaging systems is able to

deliver 7.6 Mpixel video data at 15 fps capturing speed.

Beside the theoretical limitation, there are few implementation issues to solve. Currently all the video codec runs on PC platform with MMX ability, none of them is ready for embedded systems platform. For systems interface to achieve maximum transfer rate, overall hardware design must have perfect interconnection and intercommunication. To achieve maximum transfer rate for systems interface, the hardware design must have enough buffer memory and effective interrupt routine to eliminate any waiting state.

We found that given current available embedded systems chipset and video codec, we are not able to design a lossless color imaging system beyond 7.6 Mpixel resolutions. This is due to the limitation of bandwidth and video compression technology.

7. Acknowledgment

This research is partly funded by Malaysian MOSTI ScienceFund 01-02-11-SF0019.

8. Reference

- [1] Gordon Wyeth. Implementing Active Vision in Embedded Systems, Proc. of Mechatronics and Machine Vision in Practice 4, IEEE Computer Society Press, 1997, pp. 240-245.
- [2] Y-Media, Inc "Digital Imaging Optical Format" <http://archive.chipcenter.com/analog/tn051.htm>
- [3] Nicolas Blanc, Zurich "CCD versus CMOS – has CCD imaging come to an end?" Photogrammetric Week 01, 2001, pp. 131-137.
- [4] Alexey Lukin, Denis Kubasov, "An Improved Demosaicing Algorithm", Graphicon 2004 conference proceedings.
- [5] K. Parulsi and K. Spaulding (2003). Color image processing for digital cameras. In Digital Color Imaging, G. Sharma, ed. Boca Raton, Florida: CRC Press. 727–757.
- [6] Microsoft Corp "Performance of 1394 devices may decrease after you install Windows XP Service Pack 2" <http://support.microsoft.com/kb/885222>
- [7] Apple Inc "Fifth Generation iPod (Late 2006)- Technical Specifications" http://support.apple.com/specs/ipod/iPod_Fifth_Generation_Late_2006.html