EVALUATING A 16-BIT YCBCR (6:5:5) COLOR REPRESENTATION FOR LOW MEMORY, EMBEDDED VIDEO PROCESSING

Jongmyon Kim and D. Scott Wills
Microelectronics Research Center
Electrical and Computer Engineering, Georgia Institute of Technology
Atlanta, Georgia 30332-0250

ABSTRACT

This paper examines a 16-bit YCbCr (6:5:5) color representation for limited-memory, embedded video-processing applications. The 16-bit format, combined with YCbCr 4:2:0 video streams, reduces storage requirements by 29% over a 4:2:0 chroma format and 65% over a baseline 24-bit YCbCr format while providing satisfactory image quality and sufficient information for motion estimation.

1. INTRODUCTION

Future embedded image/video-processing products must achieve greater computational performance while maintaining a low cost [1]. However, the trend towards larger imaging resolutions results in higher data rates and increasing storage requirements of processors. Since this storage (buffers, registers, and caches) consumes a large percentage of silicon area, the ability to reduce data format size prior to compression can provide a reduction in system cost. The reduction in data bandwidth can also simplify system design and packaging. The MPEG-2 [2] video compression standard employs a 4:2:0 chroma format to cut storage requirements in half. This paper presents a representation that reduces storage requirements even further.

This paper examines a 16-bit YCbCr (6:5:5) color representation for reduced-memory, embedded video processing. This pixel-truncation technique differs from similar techniques (e.g., 4:2:2 and 4:2:0 chroma formats) used in video-compression applications in that it reduces the data size of individual pixels rather than in each dimension through subsampling. The 16-bit YCbCr (6:5:5) color representation, combined with the 4:2:0 format, reduces the average per pixel word storage requirements by 65% while maintaining acceptable PSNR performance.

2. IMPLEMENTATION COSTS

The 16-bit YCbCr (6:5:5) representation can be computed from 24-bit RGB pixel data using the color conversion hardware. The conversion is defined in (1) below, where Y can assume values between [0,63], and Cb and Cr assume values between [0,31].

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.0748 & 0.1468 & 0.0285 \\ -0.0211 & -0.0414 & 0.0625 \\ 0.0625 & -0.0524 & -0.0101 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 16 \\ 16 \end{bmatrix} \quad (1)$$

This color transformation matrix can be computed with a nine cycle latency and a three cycle per pixel throughput using the pipelined datapath, shown in Fig. 1 (from [3]).

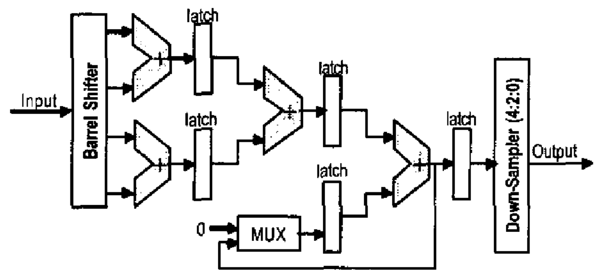

Fig. 1. A block diagram of a color converter.

For example, 0.0748 in the upper left-hand corner in (1) can be approximated by the sum $2^{-4}+2^{-7}+2^{-8}+2^{-11}$, and 0.0748R is represented by the sum R(4)+R(7)+R(8)+R(11), in which R(n) denotes a right shift of R by n bits. Following the same procedure, the 6-bit Y data can be obtained from

$$\begin{aligned} Y = & R(4)+R(7)+R(8)+R(11)+ \\ & G(3)+G(6)+G(8)+G(9)+ \\ & B(6)+B(7)+B(8)+B(10). \end{aligned} \quad (2)$$

The barrel shifter loads four data values at a time. For example, [R(4), R(7), R(8), R(11)] are loaded in the first cycle, [G(3), G(6), G(8), G(9)] are loaded in the second cycle, and so on. Using pipelining, a color pixel transformation can be completed every three cycles. To obtain the RGB values from a set of YCbCr values, the same hardware can also be used for the inverse matrix operation. Thus, 24-bit RGB to 16-bit YCbCr (6:5:5) conversion can be computed in a simple datapath without the need for area intense multiplication hardware.

YCbCr color representations exploit the properties of human visual perception by reducing the number of bits used to represent high-frequency chrominance components (Cb and Cr). The 16-bit YCbCr (6:5:5) representation, combined with the 4:2:0 chroma format,

0-7803-8838-0/05/$20.00 ©2005 IEEE.

reduces per pixel storage by 15 bits (out of 24 bits) with minimal perceptual distortion, shown in Fig. 2 (available in color at [4]).

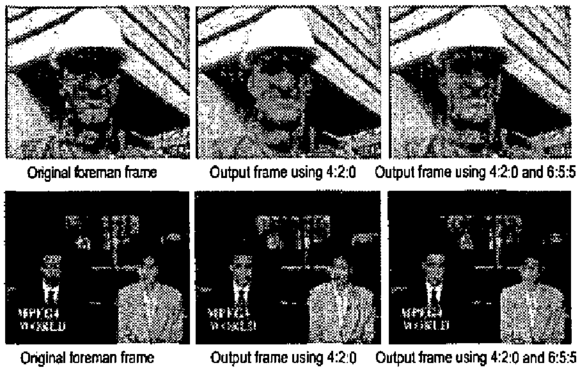

Fig. 2. Original images with converted output images [4] using the 16-bit YCbCr (6:5:5) and the 4:2:0 format.

3. MOTION ESTIMATION USING THE 16-BIT YCBCR (6:5:5) REPRESENTATION

The effectiveness of the 16-bit YCbCr (6:5:5) color representation is evaluated using motion estimation. In this experiment, the two implementations of motion estimation are executed in MATLAB using a test suite of two color videos, each containing forty frames of three-band CIF resolution (352×288) pixels. One implementation uses 8-bit luminance components, while the other uses 6-bit luminance components. In this experiment, the macroblock size is 16×16 pixels, and the search range is ±8. Both implementations incorporate 4:2:0 chroma encoding. Video quality is measured by the peak signal-to-noise ratio (PSNR), defined as:

$$PSNR = 10\log\left[\frac{255^2}{\frac{1}{m\cdot n}\sum_{i=1}^{m}\sum_{j=1}^{n}[x(i,j)-y(i,j)]^2}\right][dB] \qquad (3)$$

where x, y are the original and estimated frames, respectively, and m, n denote the dimensions of each frame.

Figures 3 and 4 show the PSNR values versus frame number for the 6- and 8-bit implementations of motion estimation. The reported PSNR is the average PSNR of the three channels.

Results indicate that the overall quality of motion estimation using the 16-bit YCbCr (6:5:5) representation plus the 4:2:0 format is comparable to the 24-bit YCbCr plus the 4:2:0 format, showing an average PSNR of 30.9 dB for the foreman video and 32.2 dB for the news video (an average loss of 0.7 dB (2%) for the foreman video and 0.4 dB (1%) for the news video). Another widely used noise reduction algorithm for color image and video, vector median filtering (VMF) [5], has been examined with similar results, shown in Table 1.

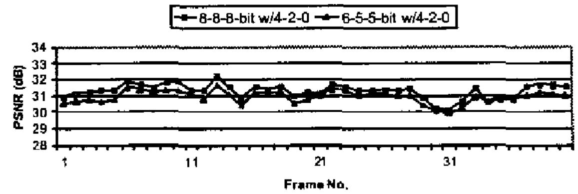

Fig. 3. PSNR versus frame number of the foreman video for motion estimation.

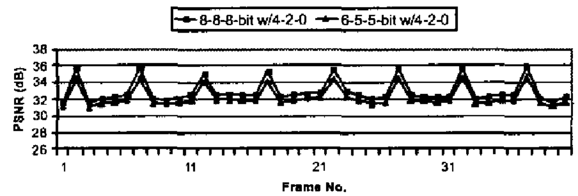

Fig. 4. PSNR versus frame number of the news video for motion estimation.

Table 1. An average PSNR of the foreman and news videos using the VMF. Note that original video frames are corrupted with 2% "salt and pepper" noise.

	Foreman	News
Using a YCbCr (6:5:5) format	29.5 dB	31.4 dB
Using a YCbCr (8:8:8) format	29.7 dB	31.8 dB

4. CONCLUSIONS

We have examined a 16-bit YCbCr (6:5:5) color representation to reduce storage requirements for video-processing applications. Results show that the YCbCr (6:5:5) representation, combined with the 4:2:0 chroma format, reduces the pixel word storage by 29% over the 24-bit 4:2:0 YCbCr representation. Overall video quality remains high, and motion estimation continues to perform well using the reduced storage format.

REFERENCES

[1] K. Diefendorff and R. Dubey, "How multimedia workloads will change processor design," *IEEE Computer,* vol. 30, no. 9, pp. 43-45, Sept. 1997.

[2] *Coding of Moving Pictures and Audio,* ISO/IEC JTC1/SC29/WG11 N3312, 2000.

[3] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards,* Kluwer Academic Publishers, 1997.

[4] Supporting materials for this paper: http:// www.ece.gatech.edu/research/pica/pubs/ICCE05/

[5] J. Kim and D. S. Wills, "Fast vector median filter implementation using the color pack instruction set," in *Proc. of IEEE Intl. Workshop on Digital Signal Processing*, pp. 339-343, Oct. 2002.