1 Introduction

In this paper we present our experience with the implementation of a color facsimile machine according to the recent ITU standard. We believe our main contributions are in the imaging processing portion of the system and this will be the main focus of this paper.

The coming together of three breakthroughs is about to cause as rapid a shift from binary black and white facsimile machines to a color version, as has recently happened for ink jet printers and multifunction peripherals. These breakthroughs are as follows:

Digital imaging: New compression algorithms such as JPEG [7] allow a considerable reduction of image data size, making possible the transmission over conventional telephone lines.

Hardware cost/performance: The costs of CPUs, memory, scanners, and printers have plummeted, making it possible to build an affordable color facsimile machine for the small office/home office (SOHO) market.

International standard: In November 1994 the International Telecommunications Union – Telecommunications Sector (ITU-T) approved recommendation T.42 with annexes to the Group-3 and Group-4 facsimile standards that allow for the exchange of continuous-tone color and gray-scale images [8–11, 14, 16].

At the time this standard was approved, it was believed that facsimile machines with new capabilities would appear on the market immediately, because T.42 is based on well-proven technologies. This did not happen for two reasons, one technical and one related to marketing.

Although the underlying technologies are well known per se, building a facsimile machine is more than assembling components. Seamlessly integrating the components into a system is a task that requires skills in many areas, such as real time system architecture design, color science, data compression, data communications, and halftoning. It turns out that such broad skills are rarely found.

In terms of marketing, facsimile is mostly a user interface and owners of color facsimile machines will expect the receivers to accept documents in color and not fall back to the binary black and white mode. This creates the classical problem of disruptive technologies, where a substantial marketing investment is necessary to stimulate a critical market size.

1.1 Project goal

The user's main concern is the transmission time per page, because it determines the cost of owning the equipment. We set ourselves the goal of transmitting the Four-Color-Printing Facsimile Test Chart 4CP01 on a color facsimile machine in the same time as on a conventional binary machine.

This is easier to solve than it may appear at first because the binary Group-3 compression fails on images. With some skill, JPEG allows the compression of a 24 bit per-pixel image to a file of the same size as the image thresholded to 1 bit per-pixel and compressed with the conventional vertical encoding, thus enabling a similar transmission time, yet at an incomparably better image quality.

Fig. 1 shows the various stages in the processing of a color facsimile page. An ordinary colorimetric bitmap from a scanner is compressed using the JPEG method and the resulting data structure is encapsulated for transmission in the general facsimile protocol. The receiving machine performs the steps in reverse order.

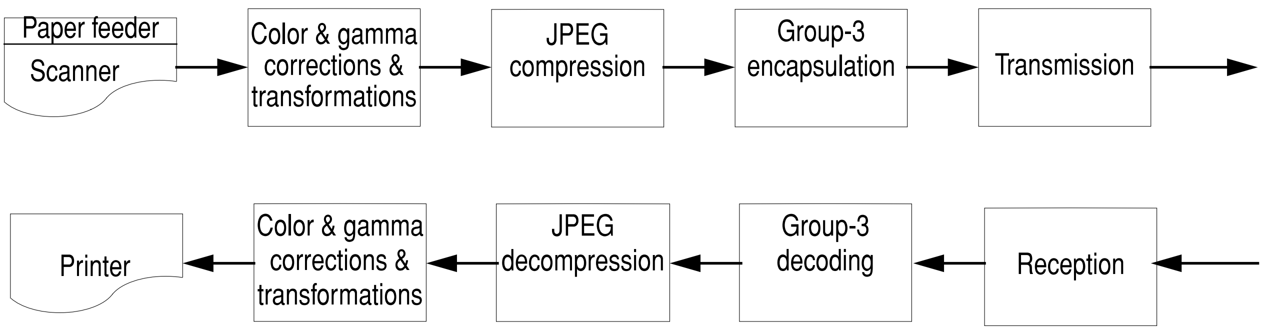

Figure 1. Color facsimile pipeline. The RGB bitmap is converted to the CIELAB color space and the chrominance channels are spatially subsampled. The image is then compressed using the JPEG method encapsulated for the conventional Group-3 facsimile protocol, and transmitted over an ordinary phone line. At the receiving end, the inverse operations are applied in reverse order.

2 Color processing

A study proposal by the Nippon Telegraph and Telephone Company (NTT) considers seventeen color spaces. Given the constraints of robustness with respect to quantization error, compatibility with compression algorithms, and color stability with white point change, the CIE 1976 ($L^*a^*b^*$) color space [6] (CIELAB) has been selected as the mandatory color space for color facsimile. Salient advantages of CIELAB are as follows:

Uniformity: CIELAB allows the for best compressibility because quantization errors are at the same detectability threshold regardless of the color.

Colorimetric: Because it is based on the CIE colorimetry, it is very easy to compute CIELAB coordinates. Software algorithms are widely described in the literature and several vendors offer hardware solutions.

Von Kries-type appearance model: CIELAB is based on the von Kries hypothesis for chromatic adaptation and on E.Q. Adams' zone theory. In this hypothesis the output from the visual receptors is weighted to take into account the illuminant, providing the required stability with white point change.

In a departure from the CIE recommendation for the illuminant, the color facsimile standard specifies CIE Standard Illuminant D_{50}, which is the ANSI recommendation for the graphic arts industry. This is a somewhat unfortunate choice because it poses some challenges for soft copy.

The CIELAB color space is based on the continuous set of real numbers and the chromatic coordinates are not bounded, requiring the specification of a specific color gamut to make possible a discrete representation in octets. A large gamut will accommodate many color devices but introduce gross quantization errors, while a small gamut will reduce quantization errors at the expense of a restricted gamut. This problem has to be viewed under the simultaneous presence of compression quantization artifacts, and a study [15] has shown that a gamut range of a* = [–80, 80] and b* = [–80, 120] results in relatively unobjectionable artifacts. Accordingly, the color facsimile standard specifies a gamut of a* = [–85, 85], b* = [–75, 125], which allows for a "safety margin."

3 Data compression

3.1 The JPEG pipeline

The color facsimile standard specifies the baseline JPEG compression method. As shown in Fig. 2, this method consists of four stages. First, the raster image is subdivided into blocks. Second, a sequential discrete cosine transform (DCT) is applied, which is an orthogonal and separable transform that allows near-optimum energy compaction and for which a number of fast algorithms with low computational complexity have been developed.

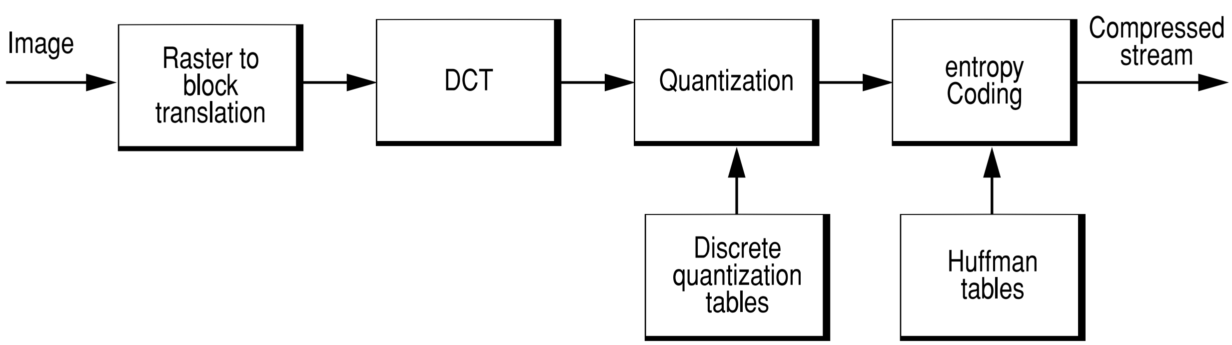

Figure 2. Simplified diagram of a discrete cosine transform (DCT) encoder. The processing steps are specified in the JPEG standard, while the tables are specified in each image.

In a third stage the data is quantized based on the discrete quantization table (DQT); each element of the DQT matrix can be any integer value between 1 and 255 and defines the quantization step for the corresponding DCT coefficients. This stage is lossy and—as we will see later—implementors design the DQT so that no visible artifacts are introduced in the image. The compression ratio of a file can be increased by setting a so-called *q-factor* or *scaling factor*, which is essentially a uniform multiplicative parameter that is applied to the quantization tables. Even when the tables are carefully designed to be perceptually lossless, a large q-factor will introduce artifacts, such as blockiness in areas of constant color or ringing on text characters. Thus, the q-factor is only a very crude tool to control the compressed file size.

The final step in the JPEG method is a lossless Huffman encoder, which eliminates the entropy in the image file. The Huffman table (HT) controls the effectiveness of the lossless compression. At the receiving end, the inverse transforms are applied in reverse order, creating a rendition of the original image.

Unfortunately the simple method of controlling the file size with the q-factor is not useful for the color facsimile application. While in conventional image processing applications images are mostly pictorial, images typical for color facsimile communication contain a fair amount of text. Blockiness artifacts in text make it difficult to read and convert using an OCR engine.

We are forced to revisit the JPEG algorithm and optimize each step for color facsimile. These steps consist in designing new quantization tables for the quantizer of the DCT-encoded image and new Huffman coding tables for the entropy coder. Both table types are part of the JPEG data structure, so it is possible to change the tables at will. Before discussing the table parameters, we briefly discuss the JPEG markers for color facsimile.

3.2 JPEG markers

Special application markers are used to identify the data structure as it relates to color facsimile (to assure the application of the proper color model operator) and to specify the parameters necessary to decompress the image if the data structure is used outside the context of a facsimile transmission, thus making it fully self-contained. In a way, the BFT (Binary File Transfer) mode of T.30 could be used to transmit a JPEG file. However, the new color extensions allow the use of the simple and powerful facsimile user interface, as well as the capabilities negotiation protocol between receiver and sender; the special markers allow for full flexibility when a computer facsimile application is created.

The *APPn* markers are undefined markers provided in JPEG to facilitate the adaptation of the standard to particular applications. In establishing the standard for color facsimile, the *APP1* marker—which follows directly the *SOI* marker— is reserved and defined for the exclusive use of color facsimile applications. In particular, the *APP1* marker initiates identification of the image as a G3FAX or G4FAX application, and defines the spatial resolution and subsampling. The resolution chosen as basic for color facsimile is 200×200 pels/25.4 mm, which is familiar in most facsimile machines as the “fine mode.” At 24 bits/pel this resolution is sufficient for quality printing on a 600 dpi CMYK printer.

3.3 JPEG DQT matrices

The file format for the JPEG image compression method is specified in the standard [7] and guarantees full portability of the files. The only parameters that can be chosen by implementors are the discrete quantization tables (DQT) and the Huffman tables (HT) shown in Fig. 2. These tables are stored in each JPEG-encoded file and can thus be customized for each image or class of images.

Let us first clarify the terms *lossy* and *lossless*. *Lossless* compression means that no data is lost when an image is compressed and then decompressed, i.e., the original and the processed image are physically identical. *Perceptually lossless* compression means that no visible data is lost when an image is compressed and the decompressed, i.e., the original and the processed image are perceptually identical. In *lossy* compression, information is lost. In *perceptually lossy* compression the loss of information is visible. The perceptual alterations to the image are called *compression artifacts* or, in the case of JPEG, *quantization errors* because they are introduced in the quantization step.

In Lohscheller's psychophysical method based on the human visual system (HVS), the elements in the DQT are chosen so that the quantization errors are just below the visibility threshold, i.e., the compression is lossy but perceptually lossless. In most practical JPEG applications, the Lohscheller's example tables given in the standard [7] are used. These example tables have been designed for pictorial images viewed on CRT displays. In color facsimile typical images contain a fair amount of text and the output device is a printer. Consequently, the example tables are not appropriate for color facsimile applications because the spatial information in text is different from pictorial images, and printers have a higher spatial resolution than CRTs.

The usual method to increase the compression ratio of the JPEG method is to increase the q-factor until the quantization errors can no longer be accepted by the user. This simple method can produce acceptable results in applications such as teleconferencing, where the images are sequences of talking heads. It does not work for color facsimile, where the images are static, observed for a relatively long time, and the preservation of detail in fine structures such as characters is important.

In a sequence of increasing sophistication, we design the DQT matrices in four steps that for reasons that will become clear later, we call the physical world, two perceptual worlds, and the semantic world. As Fig. 3 suggests, the four design iterations are carried out in a certain order; since a completely different framework is used in each step, we call them "worlds."

Physical world: In the physical world we examine the statistical properties of typical images used in color facsimile communication. We use a technique called *bit-rate control*, where the key idea is to allocate more bits to the coefficients in which the images contain more energy. A common correlate for the energy is the statistical variance. For our prototype color facsimile machine we followed the example discussed in detail in reference [1].

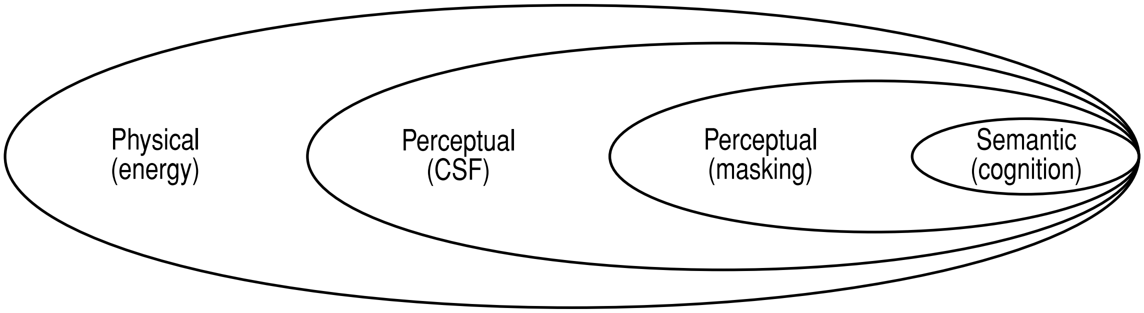

Figure 3. We can study the DQT's effect in different domains or "worlds." These worlds are hierarchical, in the sense that the DQTs are designed from the outside towards the inside in a sequence of successive refinement.

Perceptual world (contrast sensitivity): In this step we note that image quality depends on the contrast visible at a given spatial resolution. We weight the DQT elements by the contrast sensitivity function (CSF) of the HVS [5]. The result is much better than just increasing the q-factor, which is equivalent to a constant CSF.

Up to this point the resulting compression method is lossy, but we cannot make any assertion regarding the presence and magnitude of artifacts because we have not applied the visibility thresholds to the image or the DCT kernels.

Perceptual world (visual masking): The CSF is determined with a sine-wave grating. When a new structure (texture) is added to an existing structure, the new structure may mask the old structure or vice-versa. Quantization noise is a structure that is added to the original image; noise that is masked by the image is perfectly acceptable, allowing for higher compression ratios than the CFS method alone. This method is used mostly for image-dependent (adaptive) compression, but it turns out that there is little variation in the DQT elements for images of similar contents, which is the case in color facsimile communication.

One of the main researchers active in this field is Andrew B. Watson [19] of NASA Ames, and in the past few years some excellent work has been performed by Stefan J.P. Westen [20] at TU Delft in the Netherlands. Watson developed an iterative method by which the DQT matrix elements are changed until the quantization errors in each DCT kernel element are at a predetermined multiple of the visual threshold (jnd—just noticeable difference). Watson's method is based on a model of the HVS, which is much more convenient than having to perform psychophysical experiments.

Although Watson's method also works by starting with a random DQT matrix, by performing the bit-rate control step and weighting it by the CSF, Watson's method converges much more rapidly. After this step we have a compressed image where for each DCT kernel element we know the quantization error in terms of jnds. Unfortunately, to achieve a viable transmission time for color facsimile communication, we have to compress the image even more.

Semantic world: The critical image element with respect to data compression for color facsimile communication is text, which consists mostly of high-frequency information. For the quality of color facsimile, the reading performance is a critical factor. The reading performance of text is the speed at which text can be read without errors. When compressing, we can discard information that does not impact reading performance. We identify the parts of characters in fonts that affect reading performance (e.g., serifs) and discard prevalently spatial information not related to these character parts.

So far, these considerations have applied only to the luminance channel because the chrominance channels are usually considered to have the same CSF as the luminance channel, just shifted toward the lower frequencies, which is where the subsampling of the chrominance channels for a^* and b^* comes from. In the case of color facsimile, the chrominance channels are easier to handle than in the general case and we can quantize more aggressively. As Vivienne Smith [18] of the University of Chicago has noted recently, the DQTs for the red/green channel can be weighted reciprocally to the luminance channel DQT because in practice little or no text information is in this channel alone (document creators tend to avoid designs that are difficult to read and there is substantial evidence of masking interactions among the chrominance and the luminance channels in the HVS). As for the blue/yellow channel, it is sufficient to consider predominantly the DC element, since all spatial information relating to text is already in the luminance channel versus the blue/yellow channel.

In summary, we start with a DQT matrix optimized based on bit-rate control techniques, which takes into account the energy of typical images used in color facsimile communication. Then, instead of applying a global q-factor to achieve a higher compression rate, we weigh the DQT elements according to the contrast sensitivity of the HVS. After considering the visual masking occurring in average images to produce a DQT matrix with uniform and well-defined quantization error visibility, a second weighting preserving the character parts that impact the reading performance of text is applied to drastically increase the compression ratio.

3.4 JPEG Huffman tables

The example Huffman tables (HT) in the JPEG standard [7] have been determined for the perceptually lossless compression of YC_bC_r images. In color facsimile, when DQT matrices designed according to the method above are used, compression is visually lossy. Furthermore, the perceptually uniform CIELAB color space is used, which is different from the YC_bC_r color space. In the following we briefly summarize the results presented in detail at a previous meeting [12].

Our experimental results show that we can design good constrained-length Huffman codes using the simple ad hoc technique of setting to 2^{-16} all symbol probabilities less than 2^{-16} and then proceeding as if there were no constraints on the code word lengths. For each of our test images, first we compute the probability distribution of all possible symbols for which Huffman code words are needed, and then we design image-dependent custom tables. We have found that using custom Huffman tables we can improve the compression ratio by a factor between 8% and 14% compared with using the example tables from the JPEG standard. We have also found that if we use image-independent tables obtained by averaging the statistics of all test images, we only lose about 0.5% compression ratio. Hence we are presently using image-independent custom Huffman tables, achieving an average improvement of 11% in the compression ratio of typical images used in color facsimile communication.

3.5 JPEG image sharpening

Reducing the cost of the scanner part in a color facsimile machine is a difficult problem. In binary black and white machines very low cost and quality scanner assemblies can be used. Ideally, to have a similar cost for color machines, the scanner assembly should be relatively low cost. Unfortunately, most CCD (charge coupled device) sensors for color scanners are designed for high-resolution/high-speed applications in which cost is a smaller issue. At least in the beginning, until dedicated sensors are available in volume, we expect the scanner part to be a weak component in terms of the cost/performance of color scanners.

When text is scanned on a color scanner it looks somewhat fuzzy due to misregistration in the sensor and to the limited modulation transfer function (MTF). When the resulting bitmap is printed on the receiving color facsimile machine, fuzzy black text is rendered in halftone color instead of solid black, further impairing text quality and readability.

Since the JPEG method already requires transforming the image data in the frequency domain, we can boost the high frequencies to sharpen the images. Fig. 4 shows the general principle for computing the scaling factors. We compute the energy in each DQT element as described above, once for a synthetic reference image and once for the scanned image.

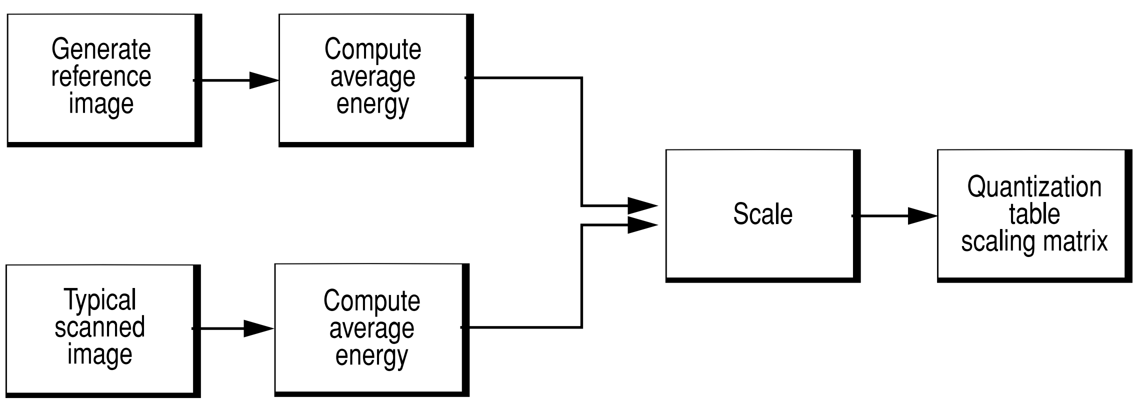

Figure 4. Sharpening an image in the JPEG domain during the encoding.

The element-wise ratio yields a scaled DQT matrix. Edge sharpening is achieved by using the original DQT for the encoding, while at the receiving facsimile machine the decoder uses the scaled DQT matrix. This method of using a different DQT matrix for the encoder and for the decoder is fully compliant with the JPEG standard, because the scaled DQT matrix is the one included in the JPEG file. Further advantages of this method are that it does not require any computation, and—since only the scaled matrix is transmitted—the original DQT matrix and the actual scaling factors can remain a trade secret of the color facsimile machine manufacturer.

4 Methodology and system architecture

We have developed our color facsimile prototype machine in two phases. In the first phase we have used the principle of stepwise refinement [21] to decide how to partition the system and which algorithms to use; we have not been concerned about the data structures, as we decided early in the project to simply pass descriptors to scan lines or frames. In the second phase we have coded the system in a way that requires only minimal operating system support (for portability) and runs optimally on 16 bit processor architectures (for cost). The language is ANSI C.

In the first phase we have collected algorithms for each processing step of a color facsimile machine and staged them on an HP725 workstation running the Unix operating system. Each algorithm is implemented as a stand-alone Unix program, processing one scan line that is read from standard input and written to standard output after processing. We have taken advantage of the Unix pipe mechanism to dynamically assemble a system at execution time. This allowed us to quickly refine and tune the system.

The second phase has consisted of replacing the Unix pipe mechanism with a simple pipeline implementation. We have then ported this new code to an HP Vectra 486/66XM personal computer running the MS-DOS operating system. For three reasons, this has been the most tedious step. First we have examined each arithmetic operation and then converted it carefully from 32 bit arithmetic to 16 bit arithmetic. Second, we have examined each pointer, cleaned up the pointer arithmetic, and selected the appropriate pointer type with the correct handling of arrays longer than 64K bytes. Third, we have examined the heap management for each object in dynamic memory to make sure it is disposed at the correct execution time.

The last step of the second phase has been the port to a custom board. This board is an image processor based on a TMS320C50 DSP from Texas Instruments. For the prototype, the board has been hosted in an HP Vectra PC, so it can easily be debugged. In addition, this method has made it possible to quickly integrate the peripheral devices, namely a scanner engine, a data pump for the communication part, and a print engine for the output.

4.1 Software

The stages in the software pipeline reflect the facsimile flow diagram in Fig. 1. The first stage is a scanner driver that allocates a buffer for a scan line and a descriptor of this buffer. The scanning mechanism is instructed to scan one line and to place it in the buffer. The descriptor is then passed to stage two, which performs in place the exposure (shading) control and the tone correction. The third stage performs the conversion to CIELAB and the fourth stage consumes the buffer to perform the JPEG compression. The fifth stage assembles the Group-3 frames and passes them to the modem. In the receiver part, the inverse stages are applied symmetrically.

Because the JPEG method operates on image blocks and the chrominance channels are subsampled by a factor of two, the compression stage operates on bursts of 16 scan lines. The communication stage operates on frames and manages the retransmission for the required ECM (error correction mode). The prototype would run faster if we also operate the scanner and the printer on bursts of scan lines. However, for our prototype this has not been necessary.

A facsimile machine is a real-time system because the facsimile standard has strict timing rules. To fulfill this requirement we have implemented an event manager. Fig. 5 shows the high-level block diagram of the software system in our prototype.

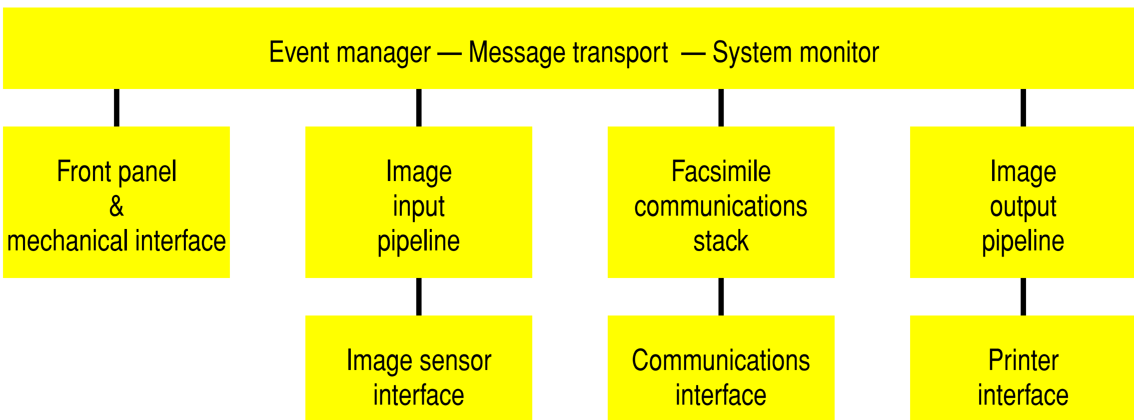

Figure 5. Block diagram of the software architecture.

4.2 Hardware

Fig. 6 shows a block diagram of our custom board, a DSP-based programmable color facsimile controller. The core is a commercially available TMS320C50 DSP from Texas Instruments. This is a far more expensive processor than the processor that would be used in an actual product. However, support tools such as an ANSI C compiler and a debugger provide an environment in which even people not experienced in DSP-related designs can test and evaluate color processing and image processing algorithms.

The programmable DSP controls the scanner, color processing, compression and decompression. In our implementation, the modem can be controlled either by the PC processor or the DSP. Instead of a modem we have designed a simple circuit based on the Rockwell R144EFX facsimile data pump and a line interface; in retrospect this has not been a good decision because we have had trouble operating it at the rated speed.

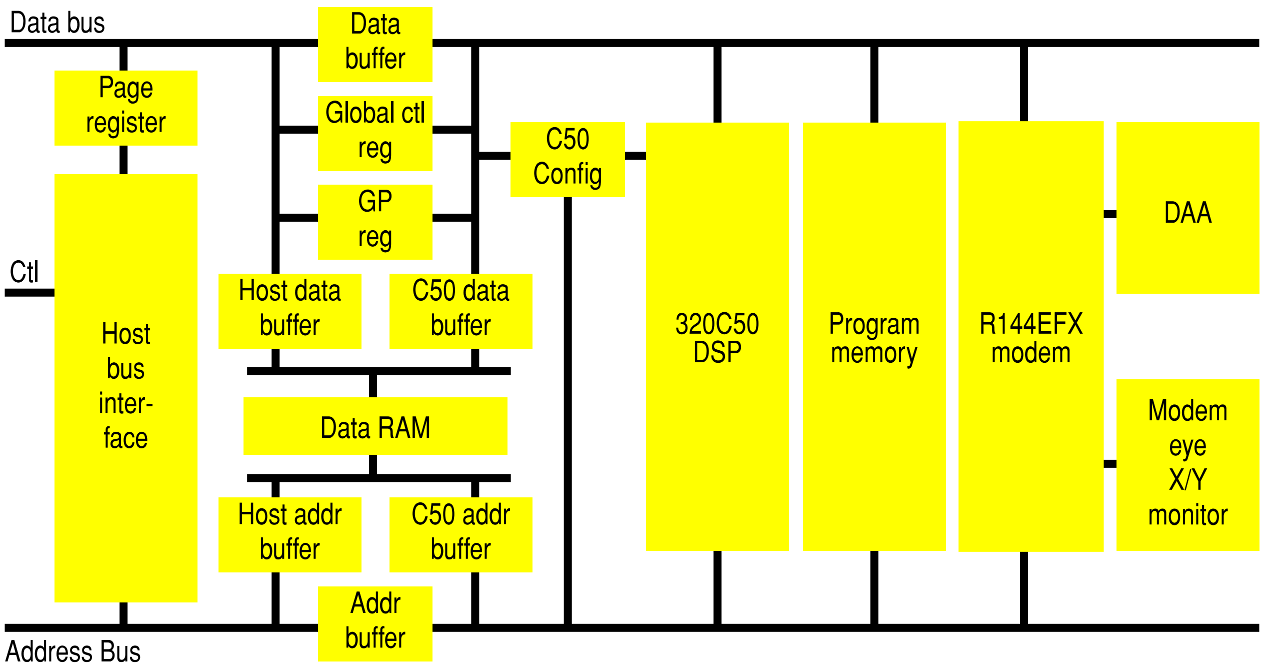

Figure 6. Block diagram of the DSP-based board for a color facsimile prototype.

5 Experimental results

We have achieved error-free communication at speeds up to 9600 bits/s, the limiting factor being the speed at which we can execute the modem functions in software. With our custom JPEG table we have been able to achieve 60:1 or better compression ratios, achieving transmission times of less than three minutes [4]. With a V.34 facsimile modem it should be possible to transmit a full-color facsimile page in under a minute, which is an acceptable transmission time.

5.1 Protocol issues

As noted in an earlier paper [3], two technical points should be noted during the message phase. They relate to inter-line fills and the color space.

Inter-line fills: The entire page must be encoded and transmitted strictly according to the JPEG standard and the error correction mode. In particular, it is not correct to introduce null octets into an ECM frame. This will cause a transmission failure due to corruption of the JPEG data structure. Introduction of null frames may occur because the conventional implementation practice is to require an inter-line fill across error-corrected block boundaries so that individual scan lines are not split between blocks.

Color space: The second point regards the JPEG encoder/decoder. Once the image is decoded from DCT coefficients to spatial coordinates, a color space transformation is required before printing or display. Most color JPEG implementations have provisions for images represented in the 3-dimensional RGB and YUV color spaces. The YUV and the CIELAB color spaces are both luminance-chrominance spaces, yet YUV is quite different from the CIELAB space used for the color facsimile standard.

In addition, the CIELAB color space always requires the specification of a white achromatic stimulus. As noted earlier, the color facsimile standard assumes CIE standard illuminant D_{50}. If the colors are bluish or yellowish, the transformation from device coordinates to CIELAB or vice versa might have been carried out for an incorrect reference white or illuminant.

6 Conclusions

New extensions to the standard ITU facsimile protocol have become available that allow the transmission of gray-scale and full-color images. We have verified the standard's viability by achieving a full implementation on a personal computer. For documents containing color images, the transmission time is approximately the same as on a conventional binary facsimile machine, although at an incomparably superior image quality. Following the standardization of hard-copy facsimile, the standardization work continues on the soft-copy and palette-color extensions (using JBIG coding) to the color facsimile standard. Furthermore, the problems and prospects for color management in the context of color facsimile need to be explored; work in this direction has already appeared [17].

Acknowledgments

The authors would like to thank Makoto Matsuki of NTT Printek Corporation for his leadership in the ITU Color Facsimile Standardization efforts.

References

[1] Bhaskaran V. and K. Konstantinides, *Image and Video Compression Standards — Algorithms and Architectures*, Kluwer Academic Publishers, Boston, USA, 1995.

[2] Beretta G.B., "Experience with the New Color Facsimile Standard," *Proc. ISCC 64th Annual Meeting*, Greensboro, USA, 28–36, April 23–25, 1995.

[3] Beretta G.B., E.F. Chan, K. Konstantinides, D.T. Lee, H.J. Lee, and A.H. Mutz, "New Group-3 Color-Facsimile Standard: Protocol Testing and Performance," *SID Digest of Technical Papers*, Orlando, USA, **XXVI**, 909–912, May 23–25, 1995.

[4] Beretta G.B., K. Konstantinides, D.T. Lee, H.J. Lee, and A.H. Mutz, "Implementing a color facsimile machine," *IS&T's 11th Int. Congress on Advances in Non-Impact Printing Technologies*, 475–478, Hilton Head, USA, November 2–7, 1995.

[5] Chen W.-H., C.H. Smith, "Adaptive Coding of Monochrome and Color Images," *IEEE Transactions on Communications*, **COM-25**, 1285-1292, 1977.

[6] CIE, *Colorimetry*, 2^{nd} Edition, 15.2, Vienna, Austria, 1986.

[7] ITU-T Rec. T.81 | ISO/IEC 10918-1 (JPEG), *Information technology —- Digital compression and coding of continuous-tone images*, 1992.

[8] ITU-T T.42, *Continuous-tone colour representation method for facsimile*, 1995.

[9] ITU-T T.30, *Procedures for document facsimile transmission in the general switched telephone network*, 1992.

[10] ITU (Q.5/8 Rapporteur), *Amendments to ITU-T Rec. T.30 for enabling continuous-tone colour and grayscale mode for Group 3*, COM 8-43-E, June 1994.

[11] ITU (Q.5/8 Rapporteur), *Amendments to ITU-T Rec. T.4 to enable continuous-tone colour and grayscale mode for Group 3*, COM 8-44-E, June 1994.

[12] Konstantinides K., "Effects of Custom Huffman Tables on the Compression of CIELAB Images Using JPEG, *SID Digest of Technical Papers*, San Diego, USA, **XXVII**, 951–954, May 12–17, 1996.

[13] Konstantinides K., H.J. Lee, G.B. Beretta, and A.H. Mutz, "A Testbed for Color Imaging and Compression," Proc. *Intern. Conf. on Digital Signal Processing*, **1**, 29–34, Limassol, Cyprus, June 26–28, 1995.

[14] Matsuki M. and D.T. Lee, "Developments in Color-Facsimile Standards: An Overview," Proc. *SID 94 Digest*, San José, USA, **XXV**, 883–886, June 1994.

[15] Mutz A.H., "Effect of Gamut Range Upon Color Quantization During JPEG Coding," *Proc. SID'94 Digest*, San José, USA, **XXV**, 891–893, June 1994.

[16] Mutz A.H. and D.T. Lee, "An International Standard for Color Facsimile," *2^{nd} IS&T/SID Color Imaging Conference*, Scottsdale, USA, 52–54, November 15–18, 1994.

[17] Schmitt F., J.-P. Crettez, Y. Wu, and G. Boulay, "Color Calibration for Color Facsimile," *SID Digest of Technical Papers*, Orlando, USA, **XXVI**, 499–502, May 23–25, 1995.

[18] Smith V., personal communication, April 1995.

[19] Watson A.B., "DCT Quantization Matrices Visually Optimized for Individual Images," *Conference on Human Vision, Visual Processing, and Digital Display IV*, San José, USA, **1913**, 202–216, February 1993.

[20] Westen S.J.P., R.L. Lagendijk, and J. Biemond, "Optimization of JPEG Color Image Coding Using a Human Visual System Model," *Conference on Human Vision, Visual Processing, and Digital Display VII*, San José, USA, **2657**, paper 43, February 1996.

[21] Wirth N., "Program Development by Stepwise Refinement," *Communications of the ACM*, **14**, 4, 221–227, April 1971.