Application of M-JPEG compression hardware to dynamic stimulus production

Jeffrey B. Mulligan

NASA Ames Research Center

mailing address:

MS 262-2
NASA Ames Research Center
Moffett Field, CA 94035-1000

tel: (415) 604-3745
fax: (415) 604-3323
email: jbm@vision.arc.nasa.gov

running head: M-JPEG compression

ABSTRACT

Inexpensive circuit boards have appeared on the market which transform a normal microcomputer's disk drive into a video disk capable of playing extended video sequences in real time. This technology enables the performance of experiments which were previously impossible, or at least prohibitively expensive. The new technology achieves this capability using special purpose hardware to compress and decompress individual video frames, which enables a video stream to be transferred over relatively low bandwidth disk interfaces. This paper will describe the use of such devices for visual psychophysics and present the technical issues that must be considered when evaluating individual products.

INTRODUCTION

The generation of extended stimulus 'movies' has been a constant challenge to researchers interested in spatiotemporal phenomena. In the past, many ingenious solutions have been found. An early approach was to use a motion picture camera to do single-frame animation (e.g. Braunstein, 1976). Nakayama and Tyler (1981) produced continuous motion fields with amplitudes of less than one pixel-width by mixing signals from a sine-wave generator with the deflection signals applied to an oscilloscope. The introduction of digital frame buffer displays permitted many advances to be made; summaries can be found in Mulligan and Stone (1989) and Cox (1997); here we mention a few techniques which are representative of the diversity of solutions. When the amount of video memory is larger than the display area, short sequences can be produced by preloading the frames into memory, and then quickly changing which part is displayed. Frame buffer displays usually include a hardware color lookup table, which maps 8 bit pixel values to red, green and blue (RGB) output levels; this table can be rewritten on a frame-by-frame basis to achieve a number of animation effects. Mulligan and Stone (1989) combined lookup table animation with digital halftoning techniques to produce a method for generating sine-wave plaids drifting with arbitrary speed and direction within a stationary spatial contrast window.

While each of these techniques is well-suited for a particular class of stimuli, all suffer from limitations of one sort or another. Stimuli played back from a movie projector are of high quality, but cannot be presented in arbitrary order. Perturbing deflection signals works well for producing tiny motions, but can generate only a limited class of motion fields. Similarly, lookup table animation techniques are limited in the class of stimuli which they can produce. Frame sequencing from video memory is limited by the amount of video memory in the system; few systems can display more than a few seconds of high-resolution imagery from frame-buffer memory, although today's increased operating speeds may allow some systems to transfer images from the system's main memory into the display memory in real time, particularly if the image size is small. Regardless of which *type* of memory is used to hold the images, the image data must first be loaded from disk or some other storage medium before it can be displayed. When the amount of data is large (i.e., high resolution and/or long duration) the inter-trial delays introduced by loading stimuli on demand may be unacceptable; the experimenter is then forced to pre-load all of the stimuli before the start of the experiment, in which case the number of stimuli is limited by the size of the memory. One approach that has been used in the author's laboratory is to maintain a database of stimuli in memory, and to fetch new stimuli from disk when needed, making room by deleting the least recently used stimulus. This works fairly well when using a staircase procedure in which a small number of stimuli are used repeatedly; but, because it cannot be known ahead of time *which* of the possible stimuli will actually be needed, the occasional reloading of stimuli can introduce variable inter-trial delays, which can provide subtle cues and may be unacceptably long.

What many researchers desire is a system that allows real-time playback of movies stored on a random-access mass-storage device (such as magnetic disk). Unfortunately, the disk transfer rate on typical personal computers and workstations is too low to support uncompressed broadcast-quality video. Video editing systems made up from large parallel disk arrays have existed for some time, but their expense has kept them out of the laboratory. Within the last few years, however, the development of image and video compression algorithms and hardware has brought video editing capabilities to the personal computer. This paper will describe the use of such devices for visual psychophysics and present the technical issues that must be considered when evaluating individual products.

A brief overview of M-JPEG video compression

The area of video compression standards is a confusing jungle of acronyms and algorithms. The situation is further confused by the fact that hardware and software vendors are not compelled to make their products conform to the standards. Another common situation is that a vendor will make arbitrary choices for algorithm parameters not specified by the standard. The discussion in this paper will focus on typical current implementations.

Products can be sorted by coding scheme into two main groups: JPEG and MPEG. JPEG refers to a class of still image compression algorithms recommended by the Joint Photographic Experts Group (JPEG); detailed information may be found in Wallace (1991), and Pennebaker and Mitchell (1993). Most ‘JPEG’ boards implement a particular algorithm based on the discrete cosine transform (DCT). The term ‘motion JPEG’ (or M-JPEG) is used to refer to systems which compress and decompress video sequences by image in the sequence as an independent still image. For interlaced formats (such as that defined by the National Television System Committee, or NTSC), each ‘image’ in the sequence is actually a ‘field’ or half-frame. Compression ratios of 10 to 1 can typically be achieved without noticeable degradation of visual quality. Most M-JPEG boards perform both compression and decompression.

For the compression of video sequences, a group of standards has been defined by the Motion Picture Experts Group (MPEG) which exploit the redundancy between successive frames to attain higher compression ratios than are possible with M-JPEG. Boards that use MPEG compression are currently on the market; unlike the case of M-JPEG, more computation is usually required to compress than to decompress, and therefore the more expensive compressors are sold separately from the decompressors. For psychophysical experimentation, if the stimuli can be compressed ahead of time using software, a hardware MPEG decompressor may be practical. The author’s entry into this technological arena was motivated by an application that required real-time image acquisition (Mulligan and Beutter, 1995; Mulligan, 1997), and M-JPEG was chosen due to the high cost of MPEG compressors. An additional consideration which may be relevant for psychophysics is that, in M-JPEG, all of the frames are treated separately, and therefore identically. This is not the case for MPEG, where a frame may be encoded as an array of differences with respect to some other frame. A detailed description of MPEG is beyond the scope of this paper, and can be found elsewhere (Mitchell *et al.*, 1997). This paper will concentrate on motion M-JPEG, although many of the considerations will be relevant for MPEG or other schemes.

Typical M-JPEG implementations

As stated above, most M-JPEG products perform both compression and decompression. For generating psychophysical stimuli, individual frames will typically be computed by a program and then passed to the compressor as an RGB image. The JPEG algorithm, as typically implemented, consists of the following steps: color transformation, chroma subsampling, block DCT, quantization, and lossless coding. First, color images are decomposed into a luminance component (Y) and two chrominance components (U and V, or C_r and C_b). The standard does not specify a particular color transformation, but decomposition into YUV components seems to be ubiquitous. Many systems then subsample the chrominance signals in the horizontal dimension. This process is analogous to the reduced chroma bandwidth in the NTSC format (Loughren, 1953; Hunt, 1987), and its use is justified by the fact that human spatial sensitivity is poorer for chromatic variation than for luminance variation (Hartridge, 1944, 1945; McIlwain, 1952; van der Horst and Bouman, 1969;

Losada and Mullen, 1994). Chroma subsampling by a factor of two is designated as 4:2:2, meaning that for every four luminance samples, there are two samples of C_r and two of C_b. Similarly, 4:1:1 subsampling would indicate that only one in four chroma samples are retained, while 4:4:4 indicates no subsampling. (There is no compelling reason why these ratios should always begin with 4, but that is how the jargon has evolved.) In addition to chroma subsampling, many inexpensive products subsample the entire image by a factor of 2 both horizontally and vertically, resulting in the so-called Standard Image Format (SIF) resolution of 320x240.

After subsampling, each component image is divided into 8x8 blocks, and each block is transformed using the DCT. The transform coefficients are then quantized by using a ‘quantization matrix’ (also called the ‘Q-table’ or ‘Q-matrix’), which controls the relative distribution of error between the different DCT channels. This matrix controls the degree of compression and the resulting visual quality. Typically, there is one matrix for the luminance component and a second matrix for both chrominance channels. The same matrix is used for all of the blocks of a given color component. The JPEG standards group includes a sample set of quantization matrices which empirically has produced the best visual quality for a particular set of viewing conditions (Lohscheller, 1984). For better or for worse, this sample is often interpreted as being recommended, and has become the de-facto standard in most commercial products. Some implementations allow user-defined Q matrices to be loaded, while others only allow the user to vary a single ‘quality factor’ which is used to uniformly scale the default matrix. Theory-based methods have been proposed to develop visually optimal quantization matrices for particular viewing conditions (Watson 1993, 1994; Peterson *et al.,* 1994; Solomon *et al.,* 1994; Watson *et al.,* 1994), and particular applications (e.g. machine vision tasks) may benefit from the use of matrices that are optimized with respect to other criteria. The availability of software which supports user-defined quantization matrices may be an important consideration for vision researchers considering this type of technology for stimulus production.

A compressed image file consists of the list of quantized coefficients coded using lossless methods. The decompression process is the inverse of the compression process: the coded data is first decompressed to obtain the DCT coefficients; the (quantized) coefficients are then transformed by an inverse DCT, and possibly up-sampled to produce a final image of the correct dimensions.

COMPRESSION HARDWARE AND PSYCHOPHYSICS

The question of whether or not compressed images are suitable for use as psychophysical stimuli ultimately rests on the nature of the experiment, and whether there are significant artifacts resulting from compressing a particular stimulus of interest. Because the quantization errors are generated in the transform domain, artifacts in the resulting image appear as little patches of gratings and checkerboards (the DCT basis functions). Because the transforms are computed independently on 8x8 blocks, highly quantized images look ‘blocky.’

It should be emphasized that distortions arising from compression artifacts are quite different than those stemming from random noise. The compression artifacts are part of the stimulus, and can therefore be known exactly. Their contrast can be computed, at which point it can be decided on reasonably objective grounds whether they will be significant or not. If the software allows the quantization matrix to be changed, then the frequency content of the artifacts can be controlled to some extent. Temporal error diffusion (Mulligan, 1993; Mulligan 1997) is a technique which may prove useful in reducing the visibility of compression artifacts in M-JPEG sequences.

Interlaced displays

Currently available devices are designed around standard video formats such as NTSC and Phase Alternating Line (PAL), and generate video output in an interlaced format. In interlaced formats, a complete image or *frame* is transmitted sequentially as two half-frames or *fields,* an *odd field* consisting of the odd-numbered lines (starting with the first, top line), followed by an *even field* consisting of the even-numbered lines. Interlaced displays pose special problems for use as psychophysical stimulators, and while not related to compression per se, interlacing will briefly be considered here as a consequence of using current M-JPEG boards. The discussion will center on the US RS-170 format, with its 30 Hz frame rate and 60 Hz field (half-frame) rate; similar considerations apply to the European formats based on a 50 Hz field rate.

The most obvious artifact resulting from interlaced displays is the 30 Hz flicker of isolated bright pixels and thin horizontal lines. Small text also becomes illegible. Large uniform regions, on the other hand, are not seen to flicker, reflecting the visual system's insensitivity to high spatio-temporal frequencies.

Flickering can be reduced, while sacrificing some resolution, by blurring the image data in the vertical dimension such that small features are represented equally in both fields. Presumably something like this is done by Silicon Graphics' 'flicker filter'; using an undisclosed algorithm, this filter achieves effective reduction of interlace artifacts in the display of high resolution text and graphics.

Because the two fields are displayed sequentially, particular care must be taken when rendering moving stimuli. A simple, if somewhat computationally expensive, approach is to render every field at the full resolution of the frame, and then select the odd or even rows to be compressed for the next field. The full-frame version of the stimulus must not contain frequencies higher than the Nyquist frequency of the field sampling array; for example, dots must be blurred to cover at least 3 or 4 lines before sampling. Because the number of lines rendered is twice the number needed, a more efficient method is to render each field directly into a field-sized buffer prior to compression. In this case, the stimuli for the even fields must be shifted vertically by half a pixel (at the field resolution) to produce a stable display.

Gamma correction

For many stimuli, it is important to precisely control the displayed gray levels. Because of the physics of cathode ray tubes, the displayed luminance is a power function of the applied voltage (see Bach *et al,* 1997, for a more complete discussion). This relation is referred to as the monitor's *gamma function;* the exponent of the power function is referred to as the 'gamma,' and typically has a value of around 2.5. *Gamma correction* refers to the use of calibration data to determine what voltage should be applied to produce a desired output luminance.

From the point of view of software modularity, it is desirable to conceal the issue of gamma correction within a low-level subroutine library, so that application programs can be written as if a perfectly linear display were available. This is especially easy when using graphics displays incorporating color lookup tables, or colormaps ('pseudocolor' displays). Images can be represented by values representing linear luminances, and the gamma correction data can be loaded into the hardware colormap. The same image can be displayed correctly on different monitors by loading calibration data appropriate to each monitor.

Unfortunately, none of the M-JPEG products surveyed here have output lookup tables, so the deferred approach to gamma correction described in the preceding paragraph is not feasible. Instead, gamma correction for the target monitor must be applied to the input imagery before compression. This means that the compressed data file contains a combination of the desired stimulus and the monitor calibration, and so must necessarily be recomputed when the experimental monitor is changed or there is a calibration shift.

Example: Band-pass noise drifting in a stationary contrast window

To illustrate the use of M-JPEG in psychophysical experiments, we now consider a set of stimuli which would be difficult to produce by other means: namely, drifting noise in a stationary contrast window (Figure 1a). While we could easily produce one or two such stimuli using a frame buffer with enough video memory to hold a complete sequence of frames, this approach is less satisfactory when the number of potential stimuli is large, for example if many different speeds and directions of motion are required for the experiment.

----

figure 1 about here

Stimuli were created using locally developed software called ‘QuIP’ (Quick Image Processing). QuIP is an interpretive environment, similar to Matlab, with additional commands for assembling and playing M-JPEG movies on several hardware platforms; the platform used was Silicon Graphics' Cosmo Compress.

Field images were computed at a resolution of 640 (H) by 248 (V). (Cosmo Compress displays only 243 lines per field, but the number of lines passed to the compressor is constrained to be a multiple of 8.) Each stimulus consisted of two patches of drifting texture; each patch was computed in a 256 x 256 staging area, and the top 248 lines were copied into the field buffer. To allow sub-pixel displacements of the base texture, a Fourier transform-based method was used. The Fourier transforms of two base textures were computed and stored. For each field in the stimulus sequence, a translated version of the base texture was computed by multiplying the Fourier transform by appropriate phase shift factors, and computing the inverse transform. The vertical displacements of alternate fields were adjusted by half a pixel to compensate for the vertical offset in the field positions. Each translated texture was then multiplied by a Gaussian contrast window having a standard deviation of 48 pixels horizontally and vertically. (Due to interlacing, the displayed window had an apparent vertical standard deviation of 96 pixels.) No interlace correction was applied to the position of the window, i.e., the same window position relative to the field buffer was used for both odd and even fields. A DC offset of gray value 127 was added to the image, and a fixation spot (a 16 x 32 rectangle in the field images which appeared square when displayed) was drawn in the center of the field buffer with a gray value of 220.

The textures were 64 x 64 binary noise patterns which were enlarged (by pixel replication) to 256 x 256, scaled to the gray level range ±100, and band-pass filtered by a difference-of-Gaussians filter having center and surround standard deviations of 8 and 4 cycles per image, respectively. (The vertical standard deviations were twice these values to compensate for the vertical stretching from interlacing.) The center and surround filters were normalized to have a gain of 1 at DC; therefore the gain at the center of the passband was significantly less than 1, and after filtering the textures had values in the gray level range ± 58. When superposed on the mean gray value of 127, this resulted in a peak contrast slightly less than 0.5.

To further improve the final quality, a temporal error diffusion process was also applied (Mulligan, 1993; Mulligan, 1997). The term *error diffusion* was introduced in the digital halftoning literature (Floyd and Steinberg, 1975) to describe a process in which serially computed local quantization errors are ‘diffused’ to their yet unquantized neighbors; when these pixels are subsequently quantized, the resulting errors tend to be negatively correlated with those of their neighbors. Here we diffuse the JPEG errors not in space, but forward in time: in each field, the negative JPEG error from the preceding field is introduced, relying on the visual system’s temporal integration to produce perceptual cancellation.

----

figure 2 about here

This process is illustrated by the diagram in Figure 2, and described explicitly by the following equations. Let $I(t)$ represent the input luminance image at time t, with $D(t)$ the corresponding final displayed luminance image. We define the *process error* $E(t)$ as the difference of these:

$$E(t) = D(t) - I(t)\ . \qquad (1)$$

The quantization error for the first stimulus frame is easily computed: first, the desired voltage necessary to produce $I(1)$ is obtained by application of a gamma correction table G^{-1}. This voltage image is compressed and decompressed; the symbol C represents the net transformation. The resulting voltage image is then transformed by the monitor's gamma nonlinearity G to produce the final displayed image, $D(1)$:

$$D(1) = G(\ C(\ G^{-1}(I(1)\ )\ )\ )\ . \qquad (3)$$

The mappings gamma function G and G^{-1} are tabulated from calibration data; the transformation C is provided by the display system software (Silicon Graphics' compression library). For the results presented here, compression was performed using the default Q matrix, which was scaled with a Q factor of 90.

The compressed image at time t, for $t > 1$, is computed as follows: let the symbol $\mathrm{Q}(t)$ represent the error to be diffused into the image at time t. Let $\mathrm{Q}(1) = 0$, and $\mathrm{Q}(2) = \mathrm{E}(1)$. The error $\mathrm{Q}(t)$ is subtracted from the desired image $\mathrm{I}(t)$ prior to compression, yielding the following general equation for the displayed image:

$$\mathrm{D}(t) = \mathrm{G}(\,\mathrm{C}(\,\mathrm{G}(\,\mathrm{I}(t) - \mathrm{Q}(t-1)\,)\,)\,)\,. \qquad (4)$$

The error $\mathrm{Q}(t)$ to be 'diffused' to the next frame is the difference between the displayed image and the intended image:

$$\mathrm{Q}(t) = \mathrm{D}(t) - (\,\mathrm{I}(t) - \mathrm{Q}(t-1)\,)\,.$$

Note that the 'intended image' is not the input image $\mathrm{I}(t)$, but rather $\mathrm{I}(t)$ *with* the error diffusion correction included, but *without* compression artifacts. Therefore, in general $\mathrm{E}(t) \neq \mathrm{Q}(t)$. No correction for positional offsets due to interlacing was applied to these error images.

To illustrate the error diffusion process, a sequence was computed in which error diffusion was applied only to the right half of each field image. Figure 1(a) shows a typical field image obtained by this procedure. (The field images shown in Figure 1 correspond to their memory representation; when displayed, they are vertically expanded by a factor of 2 due to interlacing.) Figure 1(b) shows a contrast-enhanced version of the error $\mathrm{E}(t)$ for one field, obtained by multiplying the errors by 8 and adding a constant mean luminance.

Qualitative differences in the two error signals are easily seen in Figure 1;

these are quantified in Figure 3, which shows radially averaged power spectra (after Ulichney, 1987) of the left and right subimages. These spectra were computed by zero-padding each of the 256x248 subregions to a size of 256x256, and computing the two-dimensional Fourier transform. The transform was normalized so that a unit-amplitude, integer-frequency sinusoid generated a pair of unit transform coefficients. The squared magnitude of the complex transform coefficients was computed, and the results were averaged across orientation in 180 bins having a width of 1 cycle per image.

----

figure 3 about here

The curves in Figure 3 confirm what we can see in Figure 1(b), namely that the error diffusion process has increased the noise power over most of the image. The benefit of error diffusion is apparent in Figure 1(c), which shows the cumulative error over 6 fields, contrast enhanced by a factor of 4. (The term *cumulative error* refers to a summation of $E(t)$.) On the left, where each field was compressed independently, errors are positively correlated, and the cumulative error has increased. Radially averaged power spectra for these images are shown in Figure 4, where we can see that the spectrum for the case without error diffusion has retained the same overall shape, but has risen by about 0.8 log units, corresponding to nearly linear error summation over the 6 fields. With error diffusion, however, successive errors tend to be negatively correlated; below a radial frequency of 64 cycles per image, the cumulative error power is even less than that for a single field. The frequency dependence of this effect and the shape of the basic error spectrum are controlled primarily by the quantization matrix.

----

figure 4 about here

The preceding analysis is insufficient to fully describe the costs and benefits of error diffusion, but may illustrate the usefulness of this approach. Additional material concerning this technique has been reported by Mulligan (1997).

CONCLUSIONS

Hardware image compression devices now permit presentation of a large number of complex image sequences using ordinary microcomputer equipment. For many psychophysical applications, the degradations introduced by by the compression process may be tolerable. Because the artifacts are generated by a deterministic process, they can be simulated and evaluated, and can be controlled to some extent by the values of the quantization matrix, as well as by techniques such as temporal error diffusion. In many situations this type of hardware may offer an affordable method to perform experiments that would otherwise be impractical.

Acknowledgments

The author is grateful to Hans Strasburger, Beau Watson, Al Ahumada, and Brian Wandell for suggestions concerning the manuscript. This work was supported by NASA RTOP's 505-64-53 and 199-06-12-31. Requests for information can be sent by electronic mail to the author at jbm@vision.arc.nasa.gov.

REFERENCES

Bach, M., Meigen, T. and Strasburger, H. (1997). Raster-scan cathode ray tubes for vision research - limits of resolution in space, time and intensity, and some solutions. *Spatial Vision,* **XXX**, xxx-yyy.

Braunstein, M. L., (1976). *Depth Perception Through Motion.* Academic Press, New York, NY.

Cox, M. J. (1997). Making things move - the options for computer-based displays. *Spatial Vision,* **XXX**, xxx-zzz.

Floyd, R.W. and Steinberg, L. (1975). Adaptive algorithm for spatial grey scale. *SID Intl. Symp. Dig. Tech. Papers* **6**, 36-37.

Hartridge, H. (1944). Visibility of blue and yellow. *Nature (London)* **153**, 775-776.

Hartridge, H. (1945). The change from trichromatic to dichromatic vision in the human retina. *Nature (London)* **155**, 657-662.

Hunt, R. W. G. (1987). *The reproduction of colour in photography, printing and television.* Fountain Press, Tolworth, England.

Lohscheller, H. (1984). A subjectively adapted image communication system. *IEEE Trans. Comm.* **COM-32**, 1316-1322.

Losada, M. A. and Mullen, K. T. (1994). The spatial tuning of chromatic mechanisms identified by simultaneous masking. *Vision Res.* **34**, 331-341.

Loughren, A. V. (1953). Recommendations of the National Television System Committee for a color television signal. *J. Soc. Motion Picture and Television Eng.* **60**, 321-336,596.

McIlwain, K. (1952). Requisite color bandwidth for simultaneous color-television systems. *Proc. IRE* **40**, 909-912.

Mitchell, J.L. Pennebaker, W.B. Fogg, C.E. and LeGall, D.J. (eds.) (1997). *MPEG video compression standard.* Chapman & Hall, New York, NY.

Mulligan, J.B. (1997). Application of temporal error diffusion to motion JPEG. In Rogowitz, B. and Pappas, T. (eds.): *Human Vision, Visual Processing, and Digital Display VIII,* SPIE Proc. **3016**, SPIE, Bellingham, WA, in press.

Mulligan, J.B. (1993). Methods for spatiotemporal dithering. *SID Intl. Symp. Dig. Tech. Papers* **24**, 155-158.

Mulligan, J.B. (1997). Image processing for improved eye tracking accuracy. *Behav. Res. Methods Instrum. Comput.,* in press.

Mulligan, J.B. and Beutter, B.R. (1995). Eye-movement tracking using compressed video images. in *Vision Science and its Applications* **1**, 1995 OSA Tech. Dig. Series, 163-166.

Mulligan, J.B. and Stone, L.S. (1989). Halftoning method for the generation of motion stimuli. *J. Opt. Soc. Am. A* **6**, 1217-1227.

Nakayama, K. and Tyler, C. W. (1981). Psychophysical isolation of movement sensitivity by removal of familiar position cues. *Vision Res.* **21**, 427-433.

Pennebaker, W. B. and Mitchell, J. L. (1993). *JPEG Still Image Data Compression Standard.* Van Nostrand Reinhold, New York, NY.

Peterson, H. A., Ahumada, A. J. Jr. and Watson, A. B. (1994). Visibility of DCT Quantization Noise: Spatial Frequency Summation. *SID Intl. Symp. Dig. Tech. Papers* **25**, 704-707.

Solomon, J. A., Watson, A. B. and Ahumada, A. J., Jr. (1994). Visibility of DCT quantization noise: contrast masking. *Society for Information Display International Symposium Digest of Technical Papers,* Morreale, J. (ed.), **25**, 701-703.

Ulichney, R. (1987). *Digital Halftoning*, MIT Press, Cambridge, Massachusetts.

van der Horst, G. J. C. and Bouman, M. A. (1969). Spatiotemporal chromaticity discrimination. *J. Opt. Soc. Am.* **59**, 1482-1488.

Wallace, G. K. (1991). The JPEG still picture compression standard. *Comm. ACM* **34**, 30-44.

Watson, A. B. (1993). DCT quantization matrices visually optimized for individual images. In: Rogowitz, B. and Allebach, J. (eds.), *Human Vision, Visual Processing, and Digital Display IV,* SPIE Proc. **1913**, SPIE, Bellingham, WA, 202-216.

Watson, A. B. (1994). Perceptual optimization of DCT color quantization matrices. *Proc. IEEE Intl. Conf. Image Proc.*, IEEE Computer Society Press, Austin, TX, 100-104.

Watson, A. B., Solomon, J. A., Ahumada, A. J. Jr. and Gale, A. (1994). DCT basis function visibility: effects of viewing distance and contrast masking. In: Rogowitz, B. and Allebach, J. (eds.), *Human Vision, Visual Processing, and Digital Display V*, SPIE Proc. **2179**, SPIE, Bellingham, WA, 99-108.

Appendix: A non-uniform survey of vendors and products

For those with Internet access, a useful source of information on video compression products is the list of ‘frequently asked questions’ (FAQ) from the Usenet newsgroup comp.compression. Part 3 deals with image compression hardware. That document is posted monthly to the newsgroup; the web page at http://www.cis.ohio-state.edu/hypertext/faq/usenet-faqs/bygroup/comp/compression/top.html contains links to the three parts of the FAQ as well as to the MPEG FAQ. The field of image compression is changing rapidly, and the list of vendors on the FAQ cannot be exhaustive; the short list presented here certainly is not. Most of the M-JPEG products on the market are based upon the CL550 chip manufactured by C-Cube corporation. C-Cube distributes a list of of their OEM customers which is useful in locating vendors of board-level products. Their web site has good tutorial material on JPEG and MPEG.

A good source for free JPEG software is the Independent JPEG Group (IJG). The group can be reached by email at jpeg-info@uunet.uu.net. They also maintain a JPEG FAQ, which can be accessed at http://www.cis.ohio-state.edu/hypertext/faq/usenet/jpeg-faq/faq.html. This software provides a good tool for experimenting with image compression and observing the associated artifacts. Unfortunately, many of the hardware products use proprietary file formats, making it difficult to import images and sequences created with other software systems such as that from IJG.

The following table lists a small sample of M-JPEG products, followed by vendor contact information. The heart of all of these boards is C-Cube's CL550 chip. Unless stated otherwise, each board supports full-resolution video (640x486). The column labeled 'Source?' refers to the availability of source code for the vendors software drivers and application libraries, which may be useful in the case that the software as supplied does not meet the needs of a particular application (e.g., if the software does not support user-defined Q-tables, the default Q-tables could be changed or the feature could be added).

Vendor/Product	Format(s)	Sampling	Bus	Source?	Notes
Active Imaging/Crunch	RGB	4:4:4	ISA,PCI	Yes	1
Optibase/JPEG-2000	NTSC,SVHS	?	ISA	No?	1
Orchid/Vidiola Premium	NTSC/SVHS	SIF	ISA	?	1
Parallax Graphics/XVideo	NTSC,RGB	4:2:2	SUN S-bus	No	2
Silicon Graphics/Cosmo	NTSC,SVHS	4:2:2	Indigo ISA slot	No	3
Truevision/Targa	NTSC	4:2:2	EISA,PCI,NuBus	No	1
VIC Hi-Tech/Video Packer	RGB	4:4:4	ISA,EISA	?	1

Notes:

1. Author has little or no direct experience with these products.

2. Requires optional daughter card for RGB input/output. Requires additional

software package for real-time I/O to and from disk. Real-time performance requires SUN Sparc10 host or better. Software does not detect dropped frames. Proprietary file format. Q-tables are user programmable only for still images. (Remarks reflect author’s experience and may not apply to latest software release.)

3. Must be used in conjunction with one of two video options. An additional component (luminance/chrominance) input is available with the high-end option. Good software with user-programmable Q-tables, proprietary file format.

Vendors

Active Imaging (Omnix)
PO Box 7879
Incline Village, NV 89452
(800) 898-1500
(702) 832-0792
WWW: http://ns1.win.net/~omnix/
email: djohnson@omnix.win.net

C-Cube Microsystems
1778 McCarthy Blvd.
Milpitas, CA 95035

(408) 944-6300

WWW: http://www.c-cube.com/

Optibase Inc.

5000 Quorum Drive, Suite 700

Dallas, TX 75240

(214) 774-3800

email: info@optibase.com

WWW: http://www.optibase.com/

Orchid Technology

45365 Northport Loop W.

Fremont, CA 94538

(510) 683-0300

Parallax Graphics, Inc.

2500 Condensa Street

Santa Clara, CA 95051

(408) 727-2220

email: info@parallax.com

WWW: http://www.parallax.com/

Silicon Graphics Computer Systems

2171 Landings Drive

Mountain View, CA 94043

(415) 390-3900

WWW: http://www.sgi.com/

Truevision

2500 Walsh Avenue

Santa Clara, CA 95051

(408) 562-4200

email: info@truvision.com

WWW: http://ftpserver.truevision.com/Truevision.html

VIC Hi-Tech Corp.

2221 Rosecrans Avenue, Suite 237

El Segundo, CA 90245

(310) 643-5193

FIGURE LEGENDS

Figure 1: (a) Example stimulus image after JPEG compression. This 640 (H) by 248 (V) field image will be expanded vertically by a factor of two due to interlacing. The right hand side of the image was pre-processed using a temporal error diffusion process (see text). (b) Error image for the image in Figure 1(a). Contrast has been increased by a factor of 8 to increase visibility. (c) Cumulative error, computed over 6 fields. Contrast has been increased by a factor of 4.

Figure 2: Block diagram illustrating the temporal error diffusion procedure. $\mathrm{I}(t)$ is the desired luminance image at time t, $\mathrm{D}(t)$ is the final displayed image. G and G^{-1} represent the forward and inverse gamma transformations, respectively, and C represents the net process of compression and decompression. $\mathrm{E}(t)$ is the error at time t, while $\mathrm{Q}(t)$ is the luminance error resulting from the compression process C. The error $\mathrm{Q}(t)$ is subtracted from (or ‘diffused to’) the next frame. The box labeled $\mathbf{z}^{-1}$ represents a delay of one temporal sample.

Figure 3: Radially averaged power spectra for the single-image errors shown in Figure 1(b). Lower curve (’without error diffusion’) corresponds to left-hand side of Figure 1(b); upper curve (’with error diffusion’) corresponds to right-hand side.

Figure 4: Radially averaged power spectra for the six-field cumulative errors shown in Figure 1(c). Upper curve ('without error diffusion') corresponds to left-hand side of Figure 1(b); lower curve ('with error diffusion') corresponds to right-hand side. Note that the relative vertical positions of the curves is reversed from that in Figure 2.