Application of temporal error diffusion to Motion-JPEG

Jeffrey B. Mulligan

NASA Ames Research Center

ABSTRACT

The term *error diffusion* has been used in the halftoning literature to describe processes in which pixels' quantization errors are spread in space to their unquantized neighbors, causing neighboring errors to be negatively correlated and relatively invisible. The general principle may be extended to the time dimension as well, which we will refer to as *temporal error diffusion.* In this paper we will consider the use of temporal error diffusion to ameliorate the errors introduced by JPEG image compression of a stream of images.

Keywords: motion, motion JPEG, M-JPEG, error diffusion, temporal contrast sensitivity, visual quality

1. INTRODUCTION

The term *motion JPEG* (or M-JPEG) refers to a video coding scheme in which a sequence of images is represented as a series of independent still images each of which is coded using JPEG [1]. A number of commercial products are available which use this scheme to encode and decode standard video, usually by compressing the two interlaced fields (half frames) separately. Because this scheme does not exploit the sequential correlations between the images, it typically does not achieve compression factors as large as those obtained using MPEG [2], but the absence of interdependencies between the frames simplifies video editing. Also, for capturing video streams, M-JPEG codecs are currently less expensive than real-time MPEG encoders. A number of M-JPEG systems are currently in use in the NASA Ames Vision Laboratory, and this paper describes a technique developed to improve their fidelity in displaying video sequences.

JPEG is a "lossy" compression process: the decompressed images are not identical to the encoded input images. The compression rate and output quality are controlled by a *quantization matrix;* the simplest form of rate control is obtained by scaling a particular form of the matrix which has been determined to produce good visual quality [3]. The scaling factor is sometimes referred to as the *Q factor;* on our Silicon Graphics system (Cosmo Compress) it takes on a value from 1 (low quality) to 99 (highest quality).

In using this system to display stimuli for psychophysical stimuli at the NASA Ames Vision Laboratory, our primary concern is to attain the highest degree of fidelity; we are not concerned with the amount of compression, as long as the data rate is low enough to permit real-time transfer from disk. This paper is concerned with a method designed to improve the final visual quality of M-JPEG sequences, without affecting the bit rate or quantization parameters. It is based on a procedure we refer to as *temporal error diffusion.* The term *error diffusion* is used to describe a halftoning procedure introduced by Floyd and Steinberg [4]. In this procedure, a halftone image is created by sequentially scanning the input image; the first pixel is quantized, and the difference or "error" is computed between the quantized value and the input. This error is then spread to the neighboring pixels which have yet to be quantized; when these pixels are subsequently quantized, the resulting errors tend to cancel those of their neighbors.

Further information for correspondence -
Mailing address: MS 262-2, NASA Ames Research Center, Moffett Field, CA, 94035-1000
email: jbm@vision.arc.nasa.gov
www: http://vision.arc.nasa.gov/personnel/mulligan/mulligan.html
tel: 415 604-3745 (voice) 415 604-3323 (FAX)

This principle can be extended to the time dimension as well, and has been applied to halftoning by Mulligan [5]. In temporal error diffusion of a static image, the image is first quantized (by halftoning or some other process), and the corresponding error image is computed by subtracting the input from the quantized version. This error is then subtracted from the original data to form the input for the second frame. The successive errors at a given location will be negatively correlated in time, and temporal integration by the human visual system will help to produce the effect of an interpolated gray level. The process is analogous to the way in which gray levels are perceived in static halftone images, following the visual system's spatial integration of neighboring pixels. It should be emphasized that while the process of temporal error diffusion was first investigated in the context of halftoning, it is not specific to halftoning, and *any* other process which introduces errors can fill this role; in this paper, errors will result from JPEG still image compression. The next section outlines the procedure in detail.

2. TEMPORAL ERROR DIFFUSION

2.1 Basic principles

A few symbols and equations will allow the ideas to be stated more explicitly: let $\mathbf{I}(t)$ represent the input image at time t, with $\mathbf{D}(t)$ the final decompressed image which is displayed. We define the net process error $\mathbf{E}(t)$ to be the difference of these:

$$\mathbf{E}(t) = \mathbf{D}(t) - \mathbf{I}(t) . \tag{1}$$

We use the symbol $\mathbf{C}$ to represent the net transformation resulting from compression and subsequent decompression. Thus, for the case of frame-independent motion JPEG, $\mathbf{D}(t) = \mathbf{C}(\mathbf{I}(t))$. Because JPEG is in general a lossy compression process, $\mathbf{D}(t)$ will not be identical to $\mathbf{I}(t)$, and the difference is typically referred to as the *quantization error*, due to the fact that most of the error arises from quantization of the coefficients of the discrete cosine transform. We will use the symbol $\mathbf{Q}(t)$ to refer to the quantization error at time t.

The basic error diffusion algorithm is shown in diagrammatic form in figure 1, and is described by the following equations:

$$\mathbf{D}(t) = \mathbf{C}(\mathbf{I}(t) - \mathbf{Q}(t-1)) . \tag{2a}$$

$$\mathbf{Q}(t) = \mathbf{D}(t) - (\mathbf{I}(t) - \mathbf{Q}(t-1)) . \tag{2b}$$

$$\mathbf{Q}(0) = 0 . \tag{2c}$$

Note that the process error $\mathbf{E}(t)$ is not the same as the quantization error $\mathbf{Q}(t)$, because (after the first frame) the input to the compressor is not the input $\mathbf{I}(t)$, but rather the input after subtraction of the quantization error from the preceding frame. By regrouping the terms of equation 2b, we can express the quantization error $\mathbf{Q}(t)$ in terms of the process error $\mathbf{E}(t)$:

$$\mathbf{Q}(t) = (\mathbf{D}(t) - \mathbf{I}(t)) + \mathbf{Q}(t-1) , \tag{3a}$$

$$= \mathbf{E}(t) + \mathbf{Q}(t-1) . \tag{3b}$$

From equations 2c and 3b, it follows that

$$\mathbf{Q}(1) = \mathbf{E}(1) , \tag{4a}$$

$$\mathbf{Q}(2) = \mathbf{E}(2) + \mathbf{E}(1) , \tag{4b}$$

and, by induction,

$$\mathbf{Q}(t) = \sum_{i=1}^{t} \mathbf{E}(i) . \tag{4c}$$

Figure 1: Block diagram of basic error diffusion process (section 2.1). The block labeled $\mathbf{z}^{-1}$ represents a delay of one sample.

Figure 2: Alternative view of error diffusion process with arbitrary error filtering (section 2.2).

2.2 Generalized error filtering

Equation 4c suggests an alternative way of viewing the error diffusion process which is diagramed in figure 2. Instead of "diffusing" the quantization error $\mathbf{Q}(t)$, we introduce a new symbol $\boldsymbol{\delta}$ for the diffused error, which is obtained by applying a filter $\mathbf{F}$ to the process errors $\mathbf{E}$. (The symbol $\mathbf{E}$ without an argument represents the set of errors $\mathbf{E}(i)$ for $1 \le i < t$.) The error diffusion scheme outlined above can be seen in this general framework when the filter $\mathbf{F}$ is an ideal integrator.

We can formalize this view by replacing equations 2a-c with the following set of equations,

$$\mathbf{D}(t) = \mathbf{C}(\, \mathbf{I}(t) - \boldsymbol{\delta}(t)\,) \, , \tag{5a}$$

$$\boldsymbol{\delta}(t) = \mathbf{F}(\mathbf{E}) \, , \tag{5b}$$

where, in general, $\mathbf{F}$ is an arbitrary filter that must be specified.

Equation 4c makes clear that, in the original scheme, the correction applied at time t is attempting to correct *all* of the preceding errors since the beginning of the sequence. Because the human visual system is not an ideal integrator, it is unlikely that this is the optimal thing to do if visual quality is the goal; intuitively, it seems more sensible to only try to correct errors which occurred within the integration time of the visual system (about 100 ms; see, for example, Sperling and Jolliffe [6]). A simple and efficient way to do this is to make $\mathbf{F}$ an exponential low-pass filter, which can be implemented as a one-tap recursive filter:

$$\boldsymbol{\delta}(t) = \mathbf{E}(t-1) + \mathbf{w_E}\, \boldsymbol{\delta}(t-1) \, . \tag{6a}$$

$$\boldsymbol{\delta}(t) = \sum_{i=1}^{t} \mathbf{w_E}^{t-i}\, \mathbf{E}(i) \, . \tag{6b}$$

This method has been implemented, and produced results similar to straight error diffusion, as judged by informal visual comparisons. It is an open problem how the parameters of the human visual system such as temporal contrast sensitivity might dictate the design of the visually optimal filter $\mathbf{F}$.

2.3 Output Smoothing

The results of the temporal error diffusion process can be described as increasing the amplitude of the artifacts, while moving them into a higher temporal frequency band. To the extent that the meaningful content of the images (the "signal") has its energy predominantly in lower temporal bands, this allows the visual system to easily separate the signal from the noise, in spite of the fact that this noise can be of increased visibility. This separation of signal and noise based on temporal frequency also allows automatic noise reduction by post-decompression filtering. Although this is not a feature offered by current M-JPEG products, it is one which could be added with little additional complexity.

We have investigated this by smoothing the output with an exponential low-pass filter. Let $\mathbf{D}(t)$ represent the output of the decompressor, as above. We introduce the symbol $\mathbf{S}(t)$ to represent the smoothed output at time t, which we obtain as follows:

$$\mathbf{S}(t) = (\mathbf{D}(t) + \mathbf{w_S}\,\mathbf{S}(t-1))\,(1 - \mathbf{w_S})\,. \tag{7a}$$

$$\mathbf{S}(0) = \mathbf{D}(1)\,. \tag{7b}$$

This procedure has been simulated for $\mathbf{w_S} = 0.5$, and dramatically reduces the amount of visible noise. Naturally, this does not occur without a cost: motion blur is introduced as exponentially decaying trails behind moving objects in the scene, and brief events (e.g., flashes) will be attenuated and prolonged. Nevertheless, for many classes of imagery, this may be an acceptable tradeoff.

The question then arises how this procedure compares to a frame-independent scheme at a lower frame rate, where the per-image bit-rate is increased to maintain the overall bit-rate. While a definitive answer cannot be offered at this time, it is likely that in many situations a certain amount of motion blur would be preferable to the artifacts resulting from inadequate temporal sampling (usually referred to as "judder").

An additional feature of the proposed scheme is that, because the temporal smoothing would be implemented in the decoder and would not be part of the coded sequence, the amount of smoothing could be left up to the user, who could control it with a knob. In this way, individual users could make their own adjustment of the amount of superimposed noise versus the amount of motion blur.

2.4 Gamma Correction

Up to this point, we have ignored the fact that encoded images typically represent the *voltages* of the analog video signal which is used to transmit the image to the final display device, such as a cathode ray tube (CRT). Because of the physics of CRT's, the displayed luminance is not proportional to the applied voltage, but rather varies according to a power function. This relation is referred to as the monitor's *gamma function;* the exponent of the power function is usually referred to as the "gamma," and typically has a value of around 2.5. Camera circuits usually incorporate a compensating nonlinearity, so that camera images are displayed more-or-less correctly; for synthetic imagery and when high precision is desired (e.g., psychophysical stimuli), *gamma correction* is often employed. Gamma correction refers to the use of a monitor's gamma function data to determine what voltage should be applied to produce a desired output luminance; this is often accomplished by table look-up.

It is often possible to conceal the issue of gamma correction 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 data can then be displayed correctly on different monitors (each having a different gamma), simply by loading calibration data appropriate for a given monitor into the hardware colormap.

The relevance of this to error diffusion has to do with the fact that the error diffusion process seeks to produce errors which mutually cancel one another. But errors which cancel in the voltage domain will *not* exactly cancel in the luminance domain, particularly when these errors occur on different target levels. This is illustrated in figure 3.

Figure 3: Graph of CRT gamma nonlinearity illustrating how equal corrections in the voltage domain do not produce equal corrections in the luminance domain when the corrections are made on different base voltages. See section 2.4 for details.

Unfortunately, there are no M-JPEG products of which the author is aware which incorporate output lookup tables; therefore the "transparent" approach to gamma correction described above is not feasible. In other words, the hardware dictates that the compressed image values must represent CRT voltages, and not output luminances. In order to insure that the errors cancel in the luminance domain, it is therefore necessary to modify the basic error diffusion scheme by simulating the gamma nonlinearity before computing the errors, and the inverse gamma nonlinearity to compute the appropriate voltage correction.

We assume the input images $\mathbf{I}(t)$ represent the voltages of the input images, and introduce the symbols $\mathbf{L}(t)$ and $\mathbf{G}$ to represent the corresponding desired output luminances and the gamma nonlinearity, respectively. Thus,

$$\mathbf{L}(t) = \mathbf{G}(\,\mathbf{I}(t)\,)\ . \tag{8}$$

The decompressed image $\mathbf{D}(t)$ is a voltage image, and the resulting output luminance is $\mathbf{G}(\,\mathbf{D}(t)\,)$. The luminance error is therefore

$$\mathbf{E}_{\mathrm{L}}(t) = \mathbf{G}(\,\mathbf{D}(t)\,) - \mathbf{G}(\,\mathbf{I}(t)\,)\ . \tag{9}$$

Recall that in equation 5b above, we computed a correction $\boldsymbol{\delta}(t)$ by applying a filter $\mathbf{F}$ to the preceding errors $\mathbf{E}$. Now we wish to compute a voltage correction which will shift the output luminance by a given amount; this requires knowledge of both the desired luminance correction *and* the base voltage. The voltage correction $\boldsymbol{\delta}(t)$ is given by

$$\boldsymbol{\delta}(t) = \mathbf{I}(t) - \mathbf{G}^{-1}(\,\mathbf{G}(\,\mathbf{I}(t)\,) - \mathbf{F}(\,\mathbf{E}_{\mathrm{L}}\,)\,)\ . \tag{10}$$

In this equation, $\mathbf{G}(\,\mathbf{I}(t)\,)$ is the luminance corresponding to the input frame $\mathbf{I}(t)$, $\mathbf{F}(\,\mathbf{E}_{\mathrm{L}}\,)$ is the luminance correction, and the gamma-corrected value $\mathbf{G}^{-1}(\,\ldots\,)$ is the voltage required to produce the corrected luminance. A similar formulation can be made when the input images $\mathbf{I}(t)$ represent luminances instead of voltages.

3. SIMULATIONS

The processes described above have been implemented for use in the creation of psychophysical stimuli [7], and have also been tested with more natural stimuli. This section describes results obtained using a test sequence from the MPEG-4 test set, which depicts a woman communicating in sign language in front of a brightly colored background. This sequence was obtained at half resolution (360x288), and cropped to a size of 320x248 to facilitate subsequent display.

Simulated output sequences were computed using both the standard frame-independent M-JPEG, and the basic error diffusion process described in section 2.1. Error images were computed by subtracting the original inputs from the simulated outputs, which were transformed from RGB to YUV [8]. From the transformed error images, radially averaged spectra were computed following Ulichney [9], using a 128x128 subregion taken from the center of the image, aligned with the 8x8 DCT block boundaries.

Error spectra for selected still images are shown in figure 4, for two different quality levels. In each case, the error diffused image has more noise power, but roughly the same spectral shape. (The spectral shape is mainly determined by the form of the quantization matrix.) Spatial frequency is expressed in units of cycles per 8x8 DCT block.

To appreciate the benefit imparted by the error diffusion process, we must look not at the error in single frames (where it is increased by error diffusion), but in the cumulative error summed over several frames; here we have chosen to sum over 6 frames, corresponding to a visual system integration time of 100 ms at a frame rate of 60 Hz. Spectra for such cumulative error images are shown in figure 5. In figure 5 we see that the cumulative error has been reduced by error diffusion, relative to frame-independent M-JPEG.

4. CONCLUSIONS

Temporal error diffusion, inspired by the problem of digital halftoning, can be extended to any process in which image data are corrupted, such as motion JPEG. This results in a more accurate rendition of the signal, at the expense of somewhat more visible noise. This additional noise, however, has markedly different spatial and temporal properties from the signal, and is thus easily segmented by the visual system. This is in contrast to the case of frame-independent M-JPEG, where steady artifacts can persist in stationary parts of a scene. Because of the different temporal characteristics of the noise and signal, the visual quality may be improved by temporal smoothing at the decoder. The technique is general and might be applied to MPEG encoders.

ACKNOWLEDGMENTS

This work was supported by NASA RTOP's 505-64-53 and 199-06-12-31. The author would like to thank Brent Beutter, Al Ahumada, and Ken Gaskins for comments on the manuscript.

REFERENCES

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

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

3. Lohscheller, H., "A subjectively adapted image communication system." *IEEE Transactions on Communications,* **COM-32**, 1316-1322, (1984).

4. Floyd, R. W., and Steinberg, L., "Adaptive algorithm for spatial grey scale." *SID International Symposium Digest of Technical Papers,* 36-37, (1975).

5. Mulligan, J. B., "Methods for spatiotemporal dithering." *Society for Information Display International Symposium Digest of Technical Papers,* **24**, 155-158, (1993).

6. Sperling, H. G., and Jolliffe, C. L., "Intensity-time relationships at threshold for spectral stimuli in human vision." *J. Opt. Soc. Am.,* **55**, 191-199, (1965).

7. Mulligan, J. B., "Application of M-JPEG compression hardware to dynamic stimulus production." *Spatial Vision,* in press.

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

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

Figure 4: Radially averaged error spectra for a single image frame, with and without error diffusion. Plots in the left hand column are for a quality factor Q=10, while the right hand column is for a Q=90. Upper plots are for luminance (Y) error, while the middle and lower plots are for chrominance (U and V, respectively). In each plot, the curve for error diffusion lies above the curve for frame-independent coding.

Figure 5: Radially averaged spectra for cumulative error computed over 6 successive frames, with and without error diffusion. In each plot, the curve for error diffusion lies below the curve for frame-independent coding. Parameters for each subplot are the same as in figure 4, and are given in the legend above each plot frame.