2005

A New Digital Image Compression Algorithm Based on Nonlinear Dynamical Systems

Chance Glenn
Rochester Institute of Technology

Michael Eastman
Rochester Institute of Technology

Gaurav Paliwal
Rochester Institute of Technology

A NEW DIGITAL IMAGE COMPRESSION ALGORITHM BASED ON NONLINEAR DYNAMICAL SYSTEMS

Chance M. Glenn, Sr., Michael Eastman, and Gaurav Paliwal
The Laboratory for Advanced Communications Technology
The Center for Advancing the Study of CyberInfrastructure
Rochester Institute of Technology
Rochester, New York (USA)
cmgiee@rit.edu

Abstract

In this paper we discuss the formulation of, and show the results for, a new compression/decompression algorithm called DYNAMAC, that has its basis in nonlinear systems theory. We show that we are able to achieve significant compression of RGB image data while maintaining good image quality. We discuss the implementation of this algorithm in hardware, show that the same process is applicable to other digital forms of data, demonstrate that the decompression process is ideal for streaming applications, and show that the algorithm has an exploitable aspect of encryption useful for digital rights management and secure transmission. We discuss our methodology for the improvement of the performance of this codec.

Keywords
Compression, digital, security, digital rights management, chaos, nonlinear, dynamics.

1. BACKGROUND

DYNAMAC (dy-NAM-ac) stands for dynamics-based algorithmic compression. The basic foundation of the process lies in the realizations that (a) chaotic oscillators are dynamical systems that can be governed by mathematical expressions, and (b) chaotic oscillators are capable of producing diverse waveform shapes. The premise is this: a segment of a digital sequence, such as that derived from image data, can be replaced by the initial conditions of a chaotic oscillation that matches it within an acceptable error tolerance. If the size of the data needed to specify the initial conditions needed to reproduce the chaotic oscillation are smaller than the size of the digital sequence, compression is achieved. Further, if we improve the chaotic oscillator's ability to produce diverse waveform shapes, we increase the probability of matching arbitrary digital sequence segments. There are a number of compression algorithms for digital images [1]. We introduce this new nonlinear dynamics-based algorithm and attempt to show the potential it has for comparative improvements given a deeper study of its mechanisms.

1.1 Chaotic Dynamics

Chaotic systems have been studied for years in physics and engineering. Chaotic processes have been shown to provide efficient operation [2] and have applications in various engineering disciplines [3],[4]. A typical chaotic system is the Colpitts oscillator. These sets of equations are actually derived from a standard electrical circuit architecture that is used widely in engineering [5]. The equations are a three-dimensional set of nonlinear ordinary differential equations that have the form:

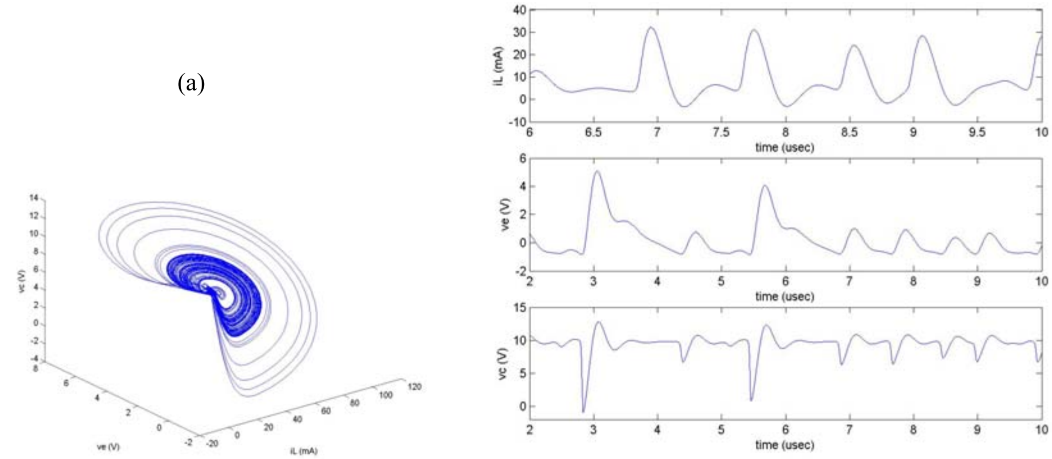
$$L\frac{di_L}{dt} = V_{CC} - v_c - (R + R_L)i_L$$

$$C_e\frac{dv_e}{dt} = i_L - \frac{v_e - V_{EE}}{R_e}$$

$$C\frac{dv_c}{dt} = C\frac{dv_e}{dt} + i_L - i_c$$

where i_c is the forward transistor collector current defined by $i_c = \gamma\left(e^{-\alpha v_e} - 1\right)$, γ and α are empirically derived factors for the transistor and R_L is the series resistance of the inductor. If we integrate these equations forward in time from a set of initial conditions $[i_{L_0}, v_{e_0}, v_{c_0}]$ we get a set of time dependent waveforms $i_L(t), v_e(t), v_c(t)$ that can be plotted versus time or as a state-space plot as seen in figure 1(a), and sections of the corresponding time-dependent waveforms in figure 1(b).

(a)



(b)

.
Figure 1. (a) Three-dimensional state-space plot of the solutions of the Colpitts equations and (b) the time-dependent waveforms

1.2 Algorithm Description

The DYNAMAC process takes advantage of the time diversity inherent in chaotic processes [6]. Figure 2 shows 64-point segments of the red, green, and blue components the image given. It is visually apparent that there is similarity in the shapes of the time-dependent waveforms from the chaotic oscillators and the RGB component sequences. The premise is this: a segments of a digital sequence, such as that derived from audio, video, and image data, can be replaced by the initial conditions of a chaotic oscillation that matches it within an acceptable error tolerance. Symbolically, we can describe a DYNAMAC operator as $\bar{d} = \mathcal{D}(\bar{x}, \mathbf{C}, \bar{k})$, where $\bar{x}$ is the original digital sequence, $\mathbf{C}$ is the combined chaotic oscillation matrix, and $\bar{k}$ is the matrix ordering sequence. If we call $l(.)$ a length function, then if $l(\bar{d}) < l(\bar{x})$ then compression occurs. We reproduce the digital sequence by $\bar{x}' = \mathcal{D}^{-1}(\bar{d}, \mathbf{C}, \bar{k})$. The error is defined as $\bar{\varepsilon} = \bar{x} - \bar{x}'$, and total error over the sequence is $\mathrm{E} = \sum_{Ns} |\bar{\varepsilon}|$, where Ns is the length of the digital sequence. If E = 0, then the compression is lossless.

The key to high compression ratios and high image quality is to the chaotic oscillator's ability to produce diverse waveform shapes. By doing so, we increase the probability of matching arbitrary digital sequence segments [7]. The images that we examine are in bitmap format, where each R,G, and B component is specified by 8-bit integers. So then, each segment of the image is 8x3xNs bits long. For this particular image each 64-point segment represents 1,536 bits.

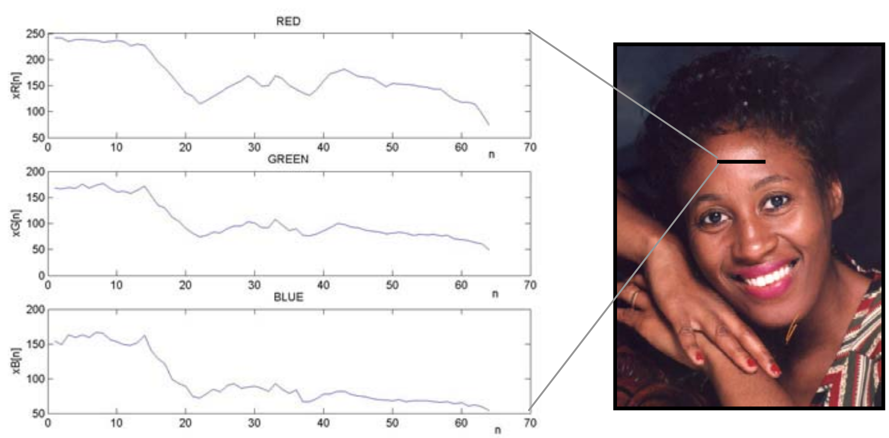


Figure 2. 64-point digital sequences representing red, green, and blue components of the image shown.

The key to the algorithm is in finding a proper match to the input digital sequence $xp[n]$ using a combined chaotic oscillation represented by $c[n]$. Figure 3 is a block diagram of the algorithm. The combined chaotic oscillation matrix, or CCO matrix, stores 32 oscillation types that can be accessed by submitting a type number, *Nt*, an oscillation starting point, *Ni*, and an oscillation length, *Nc*. The resulting chaotic oscillation will then be decimated down to the length of xp[n]. This new oscillation will be called cn[n]. The rms error between xp[n] and cn[n] is calculated. Oscillations having the smallest error value will be chosen as replacements for xp[n]. The information needed to reproduce the chosen chaotic oscillation can be distilled down to two 16-bit integers and one 4-bit integer we call *digital bites*, or *D-bites*. Specifically, **D** = [D1,D2,D3]. Since there are three digital sequences connected with an RGB image, and each component requires its own D-bite, $\mathbf{D}_R$, $\mathbf{D}_G$, $\mathbf{D}_B$, then each image segment requires 108 bits. In this example, the compression ratio is 1536:108 or 14.2:1. The new file, which we've called .dyn files, are stored as [HEADER],$\mathbf{D}_{R1}$,$\mathbf{D}_{G1}$,$\mathbf{D}_{B1}$,$\mathbf{D}_{R2}$,$\mathbf{D}_{G2}$,$\mathbf{D}_{B2}$,$\mathbf{D}_{R3}$,$\mathbf{D}_{G3}$,$\mathbf{D}_{B3}$,…,$\mathbf{D}_{RN}$,$\mathbf{D}_{GN}$,$\mathbf{D}_{BN}$, where $N = L\text{x}W/Ns$.

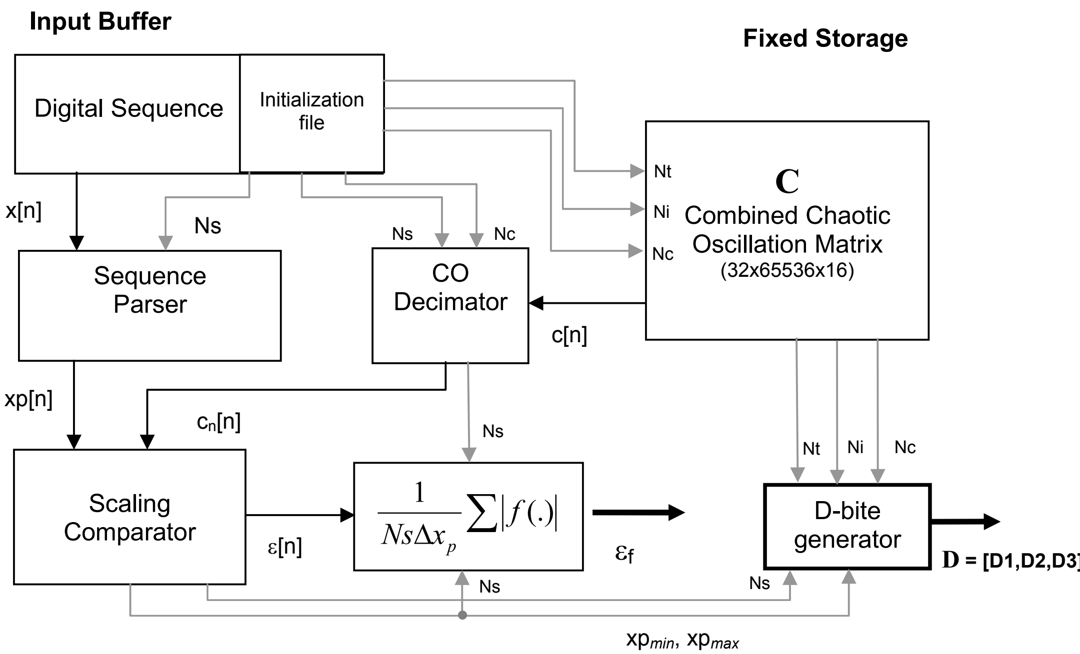


Figure 3. DYNAMAC™ compression engine block diagram.

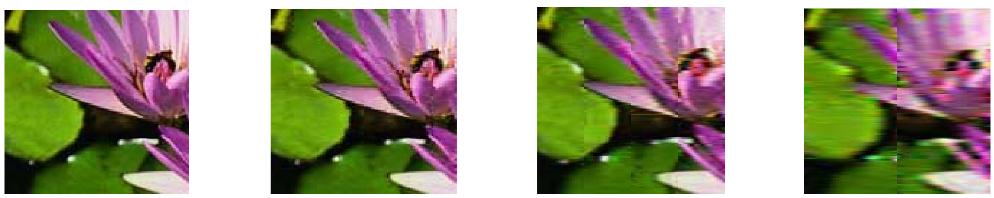
2. RESULTS

In the following section we will show examples of two different test images being operated on by the algorithm and show some of the salient measures.

2.1 Full Image Example

Our first example, in figure 4, shows the comparisons of two different images with their subsequent decompressed versions for different compression ratios. We chose two images that were sufficiently different than each other in order to demonstrate that the algorithm is viable given image variation. For each case we ran the algorithm for Ns = 16, Ns = 32, and Ns = 64. We use the definition of the mean square error, that is, $MSE = \frac{1}{LW}\sum_{l=1}^{L}\sum_{w=1}^{W}\left[I(l,w) - I'(l,w)\right]^2$, where I is the original image and I' is the new decompressed image, and the peak signal to noise ratio, $PSNR = 20\log_{10}\left(\frac{255}{\sqrt{MSE}}\right)$ [8]. We show the average values for these measures for R, G, and B.

Compression Ratio	3.6:1	7.1:1	14.2:1
MSE	17.74	41.62	112.62
PSNR	35.70 dB	32.02 dB	27.75 dB



Compression Ratio	3.6:1	7.1:1	14.2:1
MSE	115.35	277.59	658.54
PSNR	27.52 dB	23.70 dB	19.96 dB

Figure 4. Two images compressed using the DYNAMAC algorithm at three different compression ratios showing the mean-square error and the peak signal-to-noise ratio.

2.2 Other Data Types

It is clear that given digital sequences with smooth relationships, the DYNAMAC algorithm will work. We have shown in previous work that the algorithm works well for digital audio and we have developed a formal approach to the implementation of the algorithm on digital video. The compression metrics are different for each format. For example, for 16-bit, stereo audio files there four components to the D-bite D = [D1,D2,D3,D4], with there being three 16-bit integers and one 4-bit integer, for a total of 52 bits. A 64-point sequence for one channel would be 1024 bits, so the compression ratio would be 19.7:1.

3. Further Implications

In this section we discuss three additional aspects of the DYNAMAC algorithm that add to its features.

3.1 Streaming

The DYNAMAC algorithm generates self-contained digital packets that are decompressed and are completely independent of the packets preceding or following it. In this case, it is ideal for streaming applications. The decompression algorithm, $\mathcal{D}^{1}$, is significantly less calculation intensive than the compression algorithm, $\mathcal{D}$. A potential application lies in the providing of streamed, secure content for users on a network. In this case the image can be reconstructed sequentially or according to a diffusive reconstructive process [9].

3.2 Hardware

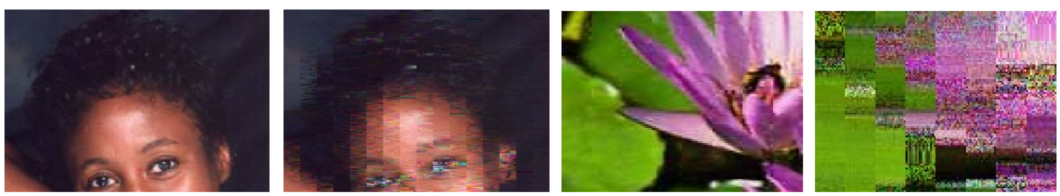
The architecture of the DYNAMAC Hardware Compression Engine (DHCE) is designed to perform rapid compression of a variety of file types using the proprietary DYNAMAC algorithm. The primary purpose of the hardware implementation is to improve the processing time. Currently, a single image may take hours to compress, however, by taking advantage of the inherent parallel nature of the algorithm it is possible to improve the processing speed over a hundred-fold. This type of drastic improvement in performance is expected to greatly enhance our ability to analyze and improve both image quality and compression ratios achieved.

Files to be compressed are delivered to the DHCE over a Universal Serial Bus (USB) connection and are buffered within an input FIFO for processing. By buffering the data within the FIFO, the input waveform can be processed rapidly without placing any unreasonable delivery time constraints on the host system. This also means that the size of the incoming file is effectively unbounded.

At the heart of the DHCE is the Sequence Processing Module (SPM). By instantiating several SPMs inside a Xilinx Virtex II Field Programmable Gate Array (FPGA) a homogeneous architecture has been created that exploits the parallel nature of the algorithm. We are currently designing the hardware system and will report on the results in a later publication.

3.3 Security and Digital Rights Management

The root of the codec is the formulation and organization of the CCO matrix. Presently it is made up of 32 pre-calculated, pre-arranged chaotic oscillation combinations described by 16-bit integers. If we let $\mathbf{k}_n$ be a vector of indices for the order of the combined chaotic oscillations then a typical ordering would be $\mathbf{k}_1 = [1,2,3,...,32]$. All of the previous images that have been compressed and decompressed have conformed to this ordering. It is clearly shown that there are 32! orderings possible available. We can achieve an image scrambling if the ordering used for compression does not match the ordering for decompression. Figure 5 shows the test image decompressed under the k1 ordering then decompressed under an alternate ordering. The vector k can act as a decryption key having up to 32! combinations. It is apparent that all orderings are not sufficiently different to cause significant scrambling, however, there are large sets that exist that will be sufficient. It is beyond the scope of this paper to explore this.



(a) (b) (c) (d)

Figure 5. DYN files decompressed under the k_1 ordering (a) & (c) and decompressed under an alternate ordering (b) & (d), showing image degradation.

4. CONCLUSIONS

The goal of this paper was to introduce a new method of digital signal compression using image data as an example. We have shown that this new algorithm, based on chaotic dynamical systems, is capable of compressing images while being ideal for streaming applications, digital rights management, and parallel processing architectures. Since the work in chaotic dynamics by Ott, Grebogi, and Yorke in 1990 [10], [11], there has been many applications postulated for the use of chaotic processes in the applied sciences. Few of these applications have shown promise in useful technological applications. We believe that if we are able to push the performance boundaries then many applications in digital data transfer are available.

There remains great opportunity to improve the algorithm. We are exploring several areas of focus in order to improve the (a) compression ratio, (b) the image quality, and (c) the compression processing time. We are exploring improved chaotic oscillation matrices and *application specific combined chaotic matrices* that are dependent on digital data type. We are also exploring deeper relationships of this process with notions like regarding digital redundancy and advanced memory modeling.

References

[1] *http://www.ph.tn.tudelft.nl/Courses/FIP/frames/fip.html,* "Image Processing Fundamentals".
[2] Gleick, J., *CHAOS: Making a New Science*, Penquin Books, New York, NY, 1987.
[3] Glenn, C. M., "High-Gain, High-Efficiency Power Amplification for PCS", International Symposium on Advanced Radio Technology Symposium Digest, March 2003.

[4] Glenn, C. M., "Weak Signal Detection by Small-Perturbation Control of Chaotic Orbits", 1996 IEEE-MTT Symposium Digest (Invited Paper), March 1996.
[5] Moon, F, *Chaotic Vibrations*, Wiley & Sons, New York, 1987.
[6] Hasler, M. J., *Electrical Circuits with Chaotic Behavior*, Proceeding sof the IEEE, vol. 75, no. 8, August 1987.
[7] Blahut, R. E., *Principles and Practice of Information Theory*, Addison-Wesley Publishing Company, New York, 1987.
[8] *http://www.debugmode.com/imagecmp/,*"An Introduction to Image Compression".
[9] Anderson, P.G., "Linear Pixel Shuffling for Image Processing, an Introduction", *The Journal of Electronic Imaging*. April 1993, pp. 147-154.
[10] Ott, E.,Gregobi, C., Yorke, J.A., Physical Review Letters 64, 1196 (1990).
[11] Ott, E., *Chaos in Dynamical Systems*, Cambridge Univ. Press, Canada, 1993.