Video Compression with Random Neural Networks

Christopher Cramer, Erol Gelenbe and Hakan Bakircioglu
Department of Electrical and Computer Engineering
Duke University
Durham, NC 27708-0291
E-mail: {cec,erol,hakan}@ee.duke.edu
Tel: (919) 660-5442, Fax: (919) 660-5293

Abstract

We summarize a novel neural network technique for video compression, using a "point-process" type neural network model we have developed [1, 2, 3, 4] which is closer to biophysical reality and is mathematically much more tractable than standard models. Our algorithm uses an adaptive approach based upon the users' desired video quality Q, and achieves compression ratios of up to 500 : 1 for moving gray-scale images, based on a combination of motion detection, compression and temporal subsampling of frames. This leads to a compression ratio of over 1000 : 1 for full-color video sequences with the addition of the standard 4:1:1 spatial subsampling ratios in the chrominance images. The Signal-to-Noise-Ratio obtained varies with the compression level and ranges from 29dB to over 34dB. Our method is computationally fast so that compression and decompression could possibly be performed in real-time software.

1 Introduction

Complex and often animated digital images are found in a wide variety of applications including remote sensing, earth observation, HDTV, entertainment and film, medical imaging and video conferencing. As the amount of data from these sources increases, the need for compression becomes ever more important. In this paper we will describe a new method for compressing and decompressing still and moving images using a learning algorithm for the "random neural network" model (Gelenbe 1989, 1990, 1991, 1993 [1, 2, 3, 4]. A schematic representation of the complete method we propose is shown in Figure 1. It uses a simple motion detection scheme to compress every $S-th$ frame, together with the set of learning neural networks for compression and decompression. At the decompression end, missing frames are then numerically interpolated using spline approximations as discussed later in this paper. In our approach ([38]), both the input, intermediate and output image is subdivided into equal-sized blocks and still image compression is carried out block-by-block. We use a feedforward encoder/decoder random neural network with one intermediate layer as shown in Figure 2. The weights between the input layer and the intermediate layer correspond to the encoding or *compression* process, while the weights from the intermediate to the output layer correspond to the decoding or *decompression* process. Currently we are using 8×8 blocks for compression ratios of 8:1, 16:1 and 32:1 and blocks of size 4×4 for compression ratios of 4:1. For each block in the picture, we encode the 8-bit gray scale value as a real number in the interval [0,1]. This value is interpreted as the external excitation of

0-8186-7456-3/96 $5.00

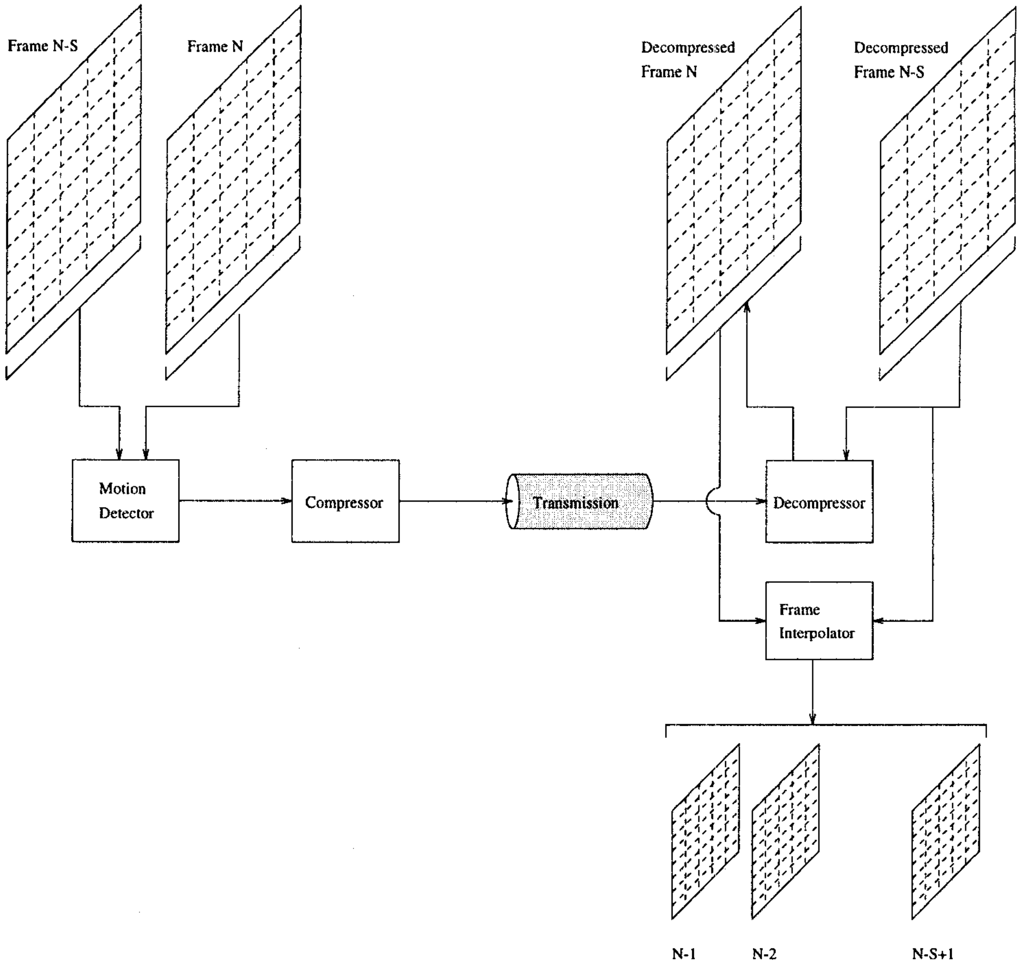


Figure 1: Block diagram of the complete compression scheme.

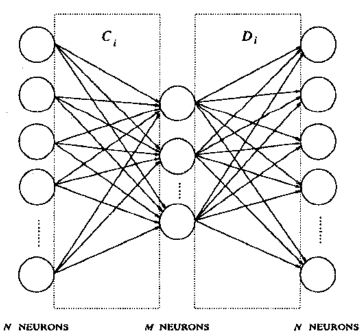
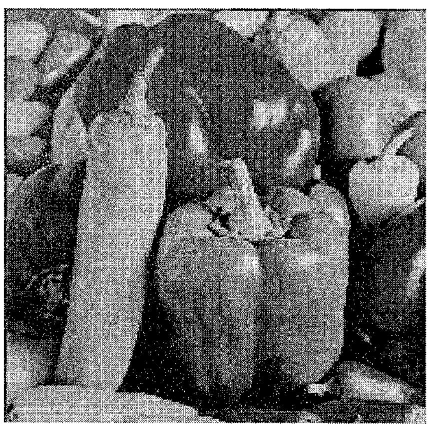


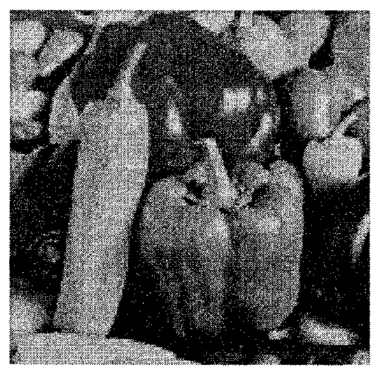
Figure 2: A Neural Network Compression/Decompression Pair

a neuron. The network is then trained to minimize the squared error between output and input values (maximizing the PSNR). Note that when demonstrating the effectiveness of this technique, we quantize the intermediate neuron values (those corresponding to the hidden layer) to 8 bits, thus obtaining the image that truly corresponds to a given compression ratio.

Figure 3: Original LENA image used to train the neural networks.



PEPPERS original



SNR = 28.236

Figure 4: Test results for 16 : 1 compression (0.5 bits/pixel) with the random neural network

2 Compression for Moving Images

In this section we will describe and evaluate the original compression scheme for video sequences of natural images, using a combination of the motion detection scheme described earlier and our adaptive still block-by-block (Figure ??) random neural network compression/decompression. The approach is based on a quality metric Q specified by the user. This metric Q maps into a block–by–block PSNR in db, between the original block and the corresponding decompressed block. Thus the sytem carries out compression based on this user specified measure Q. Our compression scheme is composed of <u>three</u> main parts:

- The first part scans successive boxes (fixed size portions of the image) in sequence, and identifies those boxes where motion has taken place, as described above. If a box is considered to be identical to the same box in the previous frame, it is neither compressed nor transmitted.

  It is also possible that the block detected by the motion detector as having moved would still meet the quality threshold Q when compared to the same block in the previous frame; this can occur when d is small. If so that block will not be transmitted and the block from the previous frame will be used at the receiver, since the resulting compression ratio is "infinite" (no data is transmitted for that block).

- The second one carries out compression of the box which is identified by the first part using the still image compression described in the previous section. It contains a set of distinct neural compression networks $C_1, \ldots, C_L$ which are designed to achieve different compression levels. Each of these networks compresses the box in parallel. The choice of the compression level to be selected is carried out by the third part.

- The third part performs the decompression, and provides a measure of the "quality" Q of the compression-decompression. In fact it is composed of L distinct decompression networks $D_1, \ldots, D_L$, where D_i decompresses the output of C_i.

The network pair C_i, D_i which yields the highest compression ratio while achieving a desired quality threshold Q is then deemed acceptable for the given box, and the compressed box is transmitted with information indicating its position in the image. Q is given as an external parameter to the video compression codec and is a $PSNR$ value generally around 30 dB.

Figure 5 shows the block diagram of the adaptive still image compression network. Note that with the exception of the initial learning phase which is carried out once and for all with the *Lena* image, all the operations which have been outlined above can be carried out "on-the-fly", i.e. in <u>real-time</u> as each box goes through the transmitter, and as each compressed box goes through the receiver.

At the *"receiving or decompression"* end, data is received which states where the following block should be placed and what the compression level the block was compressed at. The position data is given as a 16 bit (x,y) pair which indicates which is measured in *blocks*, not pixels. This allows for video sequences of size 2048×2048, which is substantially larger than even 16CIF video frames. The next information received is the number of the compressor used to compress the data. This information will then be used to calculate how many bytes long the compressed data is. Since four compression levels is generally adequate, this data field is generally two bits long. We then use the network D_i to decompress the box, which is subsequently placed in appropriate sequence into the output image. The relationship between any two compression/decompression networks C_i, D_i is shown in Figure 2.

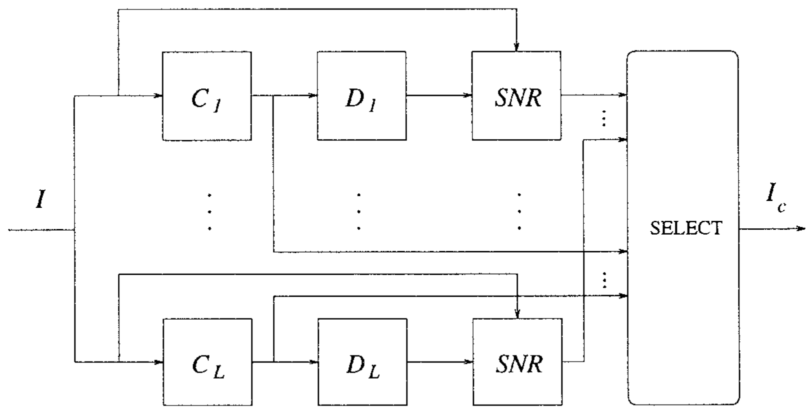


Figure 5: Block diagram of the adaptive still image compression network

2.1 Cubic Spline Interpolation

While the results given in the previous section indicate a very powerful video compression technique, like all other video compression techniques it is still not able to compress real-time full-color video traffic to rates that can be transmitted and received over phone lines or cellular modems. In order to increase the compression ratio we use temporal subsampling of the input video sequence as follows. Every S-th frame in the original input video sequence is retained, the intermediate $S - 1$ frames are dropped. The remaining frames are then compressed in some appropriate manner (we use our Neural Video Compression scheme, however, H.261, MPEG or even still image compression could be used). The total compression ratio from this scheme is therefore the compression achieved by our neural compression technique multiplied by the sampling rate S: $CR_{total} = CR_{standard} \times S$. To reconstruct the video sequence one first performs the inverse of the video compression scheme that was used. This results in a set of frames containing $\frac{1}{S}$-th of the original video sequence. We then reconstruct the missing frames on a pixel by pixel basis. To reconstruct pixel (x,y) in the missing frames, we construct a cubic spline using as control points the pixels in position (x,y) from the portion of the video sequence that was transmitted and decompressed. We then evaluate the cubic spline at the points in time that the missing frames represent, filling in the missing frames with the result of the evaluation.

To illustrate and evaluate the effect of the addition of temporal subsampling to ANVC, we use the "Salesman" sequence once again. This is a 449 frame, 360×288 pixel, 8 bit gray-scale sequence. Figure 6 shows the results of these tests where the solid line is the baseline of no subsampling (S=1), the dashed line is a subsampling level of 1 in 3 frames (S=3), the dot-dash line is 1 in 4 frames (S=4), the dotted line is 1 in 5 frames (S=5), the solid line with the stars represents results from 1 frame in 6 (S=6), and the solid line with the o's represents H.261. This graph demonstrates the superiority of our method over the existing H.261 standard. The quality of the H.261 compression method deteriorates at a much faster rate than does the neural technique with or without temporal subsampling. It can be seen that H.261 is no longer usable for compression levels higher than 150:1, whereas, we can achieve the same quality of reconstructed video sequence at a compression ratio of over 500:1. One surprising feature of this technique appearing in the graph is that for a given compression level, subsampling gives better quality than no subsampling. This shows that in addition to being the only current method of achieving very high compression ratios, subsampling of frames gives better quality even within the original compression range of the non-augmented ANVC.

3 Conclusions

In this paper we have demonstrated the ability to compress gray scale video sequences by a factor of 500:1. This implies the compression of color sequences by a factor of 1000:1 using 4:1:1 spatial subsampling in the chrominance components. This amount of compression should allow for the transmission of full motion, true color video sequences across standard telephone lines. The quality of the reconstructed sequences has been shown to be relatively high, with the temporal subsampling having little adverse effect on the quality of the reconstructed sequence.

References

[1] Erol Gelenbe, "Random neural networks with negative and positive signals and product form solution", *Neural Computation*, vol. 1, no. 4, pp. 502–511, 1989.

[2] Erol Gelenbe, "Stability of the random neural network model", *Neural Computation*, vol. 2, no. 2, pp. 239–247, 1990.

[3] Erol Gelenbe and A. Stafylopatis, "Global behaviour of homogeneous random neural systems", *Applied Math. Modelling*, vol. 15, pp. 534–541, 1991.

[4] Erol Gelenbe, "Learning in the recurrent random neural network", *Neural Computation*, vol. 5, no. 1, pp. 154–164, 1993.

[5] D. A. Huffman, "A method for the construction of minimum-redundancy codes", *Proceedings of the Institute of Electrical Radio Engineers*, vol. 40, no. 9, pp. 1098–1101, September 1952.

[6] J. J. Rissanen and G. G. Langdon Jr., "Arithmetic coding", *IBM Journal of Research and Development*, vol. 23, no. 2, pp. 149–162, December 1979.

[7] T. A. Welch, "A technique for high performance data compression", *IEEE Computer*, vol. 17, no. 6, pp. 8–19, June 1984.

[8] Gregory K. Wallace, "The JPEG still image compression standard", *Communications of the ACM*, vol. 34, no. 4, pp. 30–43, April 1991.

[9] "Digital Compression and Coding of Continuous-Tone Still Images, Part 1, Requirements and Guidelines", ISO/IEC JTC1 Committee Draft 10918-1, February 1991.

[10] "Digital Compression and Coding of Continuous-Tone Still Images, Part 2, Compliance Testing", ISO/IEC JTC1 Committee Draft 10918-2, Summer 1991.

[11] N. Ahmed, T. Natarajan, and K.R. Rao, "Discrete Cosine Transform", *IEEE Transactions on Computers*, vol. C-23, no. 1, pp. 90–93, January 1974.

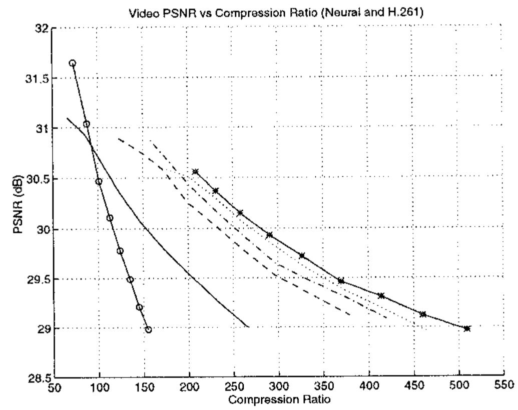


Figure 6: Graph of video quality versus compression ratios for subsampling of 1 (solid), 3 (dashed), 4 (dot-dashed), 5 (dotted) and 6 (solid with stars), and H.261 (solid with o's)

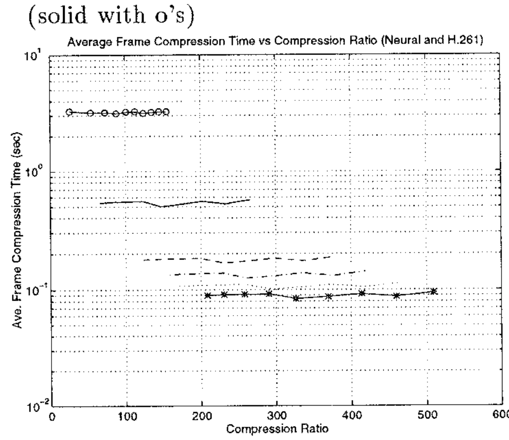


Figure 7: Graph of compression time versus compression ratios for subsampling of 1 (solid), 3 (dashed), 4 (dot-dashed), 5 (dotted) and 6 (solid with stars), and H.261 (solid with o's)

[12] R.C. Reininger and J.D. Gibson, "Distributions of the two-dimensional DCT coefficients for images", *IEEE Transactions on Computers*, vol. C-32, no. 7, June 1983.

[13] J. Woods and S.D. O'Neil, "Subband coding of images", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 5, pp. 1278–1288, October 1986.

[14] R.M. Gray, "Vector quantization", *IEEE ASSP Magazine*, vol. 1, no. 2, pp. 4–29, April 1984.

[15] B. Ramamurthi and A. Gersho, "Classified vector quantization of images", *IEEE Transactions on Communications*, vol. COM–34, no. 11, pp. 1105–1115, November 1986.

[16] E.H. Adelson and E. Simoncelli, "Orthogonal pyramid transforms for image coding", *Proceedings of the SPIE: Visual Communications and Image Processing II*, vol. 845, pp. 50–58, 1987.

[17] W. Zettler, J. Huffman, and D.C.P. Linden, "Application of compactly supported wavelets to image compression", *Proc. SPIE: Image Processing Algorithms and Techniques*, vol. 1244, pp. 150–160, 1990.

[18] A.E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations", *IEEE Transactions of Image Processing*, vol. 1, no. 1, pp. 18–30, January 1992.

[19] M. Kunt, M. Benard, and R. Leonardi, "Recent results in high-compression image coding", *IEEE Transactions on Circuits and Systems*, vol. CAS–34, no. 1, pp. 1306–1336, 1987.

[20] T. Kohonen, *Self Organization and Associative Memory*, Springer-Verlag:Berlin, 1988.

[21] Martine Naillon, "Neural approach for television image compression using a Hopfield type network", in *Advances in Neural Information Processing Systems*, pp. 264–271. Morgan-Kaufmann, 1989.

[22] Nasser M. Nasrabadi and Yushu Feng, "Vector quantization of images based upon Kohonen self organizing feature maps", in *Proceedings, International Conference on Neural Networks*. 1988, pp. 101–108, IEEE Press.

[23] G.W. Cottrell, P. Munro, and D. Zipser, "Image compression by backpropagation: an example of extensional programming", in *Models of cognition: a review of cognition science*, N.E. Sharkey, Ed. NJ:Norwood, 1989.

[24] S. Carrato, "Neural networks for image compression", in *Neural Networks: Advances and Applications 2*, Erol Gelenbe, Ed., pp. 177–198. Elsevier North-Holland, 1992.

[25] John G. Daugman, "Relaxation neural network for non-orthogonal image transformations", in *Proceedings, International Conference on Neural Networks*. 1988, pp. 547–560, IEEE Press.

[26] N. Sonehara, M. Kawato, S. Miyake, and K. Nakane, "Image data compression using a neural network model", in *Proceedings, International Joint Conference on Neural Networks*. 1989, pp. 35–41, IEEE Press.

[27] A. Namphol, "Higher order data compression with neural networks", in *Proceedings, The International Joint Conference on Neural Networks.* 1991, pp. 55–59, IEEE Press.

[28] Didier Le Gall, "MPEG: A video compression standard for multimedia applications", *Communications of the ACM*, vol. 34, no. 4, pp. 46–58, April 1991.

[29] "MPEG proposal package description", Document ISO/WG8/MPEG/89-128, July 1989.

[30] "Coding of moving pictures and associated audio", Committee Draft of Standard ISO11172: ISO/MPEG 90/176, December 1990.

[31] "Video codec for audio visual services at px64 kbits/s", CCITT Recommendation H.261, 1993.

[32] "Video coding for low bitrate communication", ITU-T Recommendation H.263, 1996.

[33] D.E. Rumelhart, J.L. McClelland, and the PDP Research Group, *Parallel Distributed Processing: Volumes 1 & 2*, MIT Press, 1986.

[34] S. H. Courellis, "An artifical neural network for motion detection and speed estimation", in *Proceedings, International Joint Conference on Neural Networks.* 1990, pp. 407–421, IEEE Press.

[35] Yi Wu Chiang, "Motion estimation using a neural network", in *Proceedings, IEEE International Symposium on Circuits and Systems.* 1990, pp. 2516–2519, IEEE Press.

[36] Erol Gelenbe, Mert Sungur, and Christopher Cramer, "Learning random networks for compression of still and moving images", in *A Decade of Neural Networks - a Workshop at the Jet Propulsion Laboratory.* May 1994, pp. 171–189, JPL.

[37] Erol Gelenbe, Mert Sungur, Christopher Cramer, and Pamir Gelenbe, "Traffic and video quality with adaptive neural compression", *accepted for publication in Multimedia Systems*, Special Issue on Traffic Control in Multimedia Networks 1996.

[38] Erol Gelenbe and Mert Sungur, "Random network learning and image compression", in *IEEE International Conference on Neural Networks.* 1994, pp. 3996–3999, IEEE Press.

[39] Michael Kass, Andrew Witkin, and Demetri Terzopoulos, "Snakes: Active contour models", *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[40] Amir A. Amini, Saeid Tehrani, and Terry E. Weymouth, "Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints", in *Proceedings of the 2nd International Conference on Computer Vision.* 1988, pp. 95–99, IEEE Press.