Neuro-Fuzzy Modeling and Prediction of VBR MPEG Video Sources

The work presented in this paper intends to apply neuro-fuzzy methods for the modeling and prediction on traffic intensity of digital video sources which are coded with hybrid Motion Compensation/Differential Pulse Code Modulation/Discrete Cosine Transform (MC/DPCM/DCT) algorithm. Although current coding standards recommend constant bit rate (CBR) output by means of a smoothing buffer, the hybrid algorithm inherently produces variable bit rate (VBR) output. This paper describes the novel application of a fuzzy predictor for the purposes of modeling and prediction on video sources. The computation requirement of the fuzzy predictor and its neural network implementation are also discussed. The proposed fuzzy prediction method and its neural network version can be applied to the development of connection admission control, usage parameter control and congestion control algorithms in ATM networks.

© 1999 Academic Press

B. Qiu¹, S. Suthaharan² and H.R. Wu¹

¹School of Computer Science and Software Engineering, Monash University, Wellington Road, Clayton, Vic. 3168, Australia
²Department of Computer Science, Tennessee State University, Nashville, TN 37209-1561, USA

Introduction

ATM networks have the potential to carry variable bit rate traffics more efficiently in their characteristics are known. A lot of work has been carried out in the area of VBR video characterization and modeling. The application of Markov processes and other statistical methods has been extensively investigated and can be found in most of the relevant publications so far [1–4]. These methods can satisfactorily describe some VBR coded video with low ratio of peak-to-average bit rates and quite gradual variation of bit rates.

Hybrid MC/DPCM/DCT is a popular algorithm which is adopted in all current image/video coding standards such as JPEG, MPEG-1, 2, 4 and H. 261, 263. It is capable of providing high efficiency and high quality encoding of visual information. In the processes of visual information compression, the hybrid coding algorithm produces bit streams with high peak-to-average ratio and pseudo-periodical type of output pattern which are not statistically stationary processes. Current codec implementations recommended by the standards are normally Constant Bit Rate (CBR) [5, 6], with the introduction of smoothing buffers at output. The CBR implementation facilitates the transmission through communication networks. However, the smoothing to constant bit rate output results in degradation of picture quality. It is important that the output bit-rate is made variable to achieve high and constant picture quality.

This paper proposes a fuzzy logic model which can be used for single-step and multistep prediction of output bit streams. Results show that the mean and the standard deviation of the error for the fuzzy predictor
are significantly smaller than the normal autoregressive models. The experiments were carried out with a software MPEG-2 encoder. VBR was achieved by ignoring the output smoothing buffer. The implementation of the fuzzy predictor using a feed forward neural network is also presented. The neural network implementation can reduce computation for multistep prediction. The proposed fuzzy prediction method and its neural network version can be applied to the development of connection admission control, usage parameter control and congestion control algorithms in ATM networks [7–9].

**Video Source Characteristics**

A number of digitized standard test video sequences are coded with the MPEG-2 encoder. The output streams are saved as disk files and subsequently analysed. The sequences are the most typical ones used by other researchers in this area which include short ones such as “flower garden”, “mobile and calendar” and longer ones such as “star wars”. They are quite representative for scenes from documentary and some feature films with medium level of scene changes. The hybrid algorithm produces an output bit stream which consists of I, P and B pictures periodically. The intra pictures (I pictures) are coded without any reference to other pictures. They result in the highest number of bits per picture since only moderate compression is involved. Predicted pictures (P pictures) use a previous I or P picture as the reference for motion prediction compensation. They are more heavily compressed than the I pictures. Bidirectionally predicted pictures (B pictures) may require both previous and future I or P pictures with interpolation if necessary for the best compression. The complete coded stream consists of sharp and somewhat periodical variations in bit rates as illustrated in Figure 1. The separated I, P and B streams present much more mild bit-rate changes Figure 2. The prediction can be carried out with mixed streams or separated individual I/P/B streams, in which case the precision is higher. The proposed research work has been carried out on separated I/P/B streams.

**Fuzzy Modeling and Prediction**

Autoregressive estimation for prediction has been thoroughly described in [10,11]. Analysis on prediction errors is also available if the time sequence is a stationary Guassian process. For non-Gaussian or non-stationary processes which represent general traffic patterns, the prediction error can not be easily analysed. For AR prediction, there is no intuitive relationship between the parameters and the estimation error. It is therefore not possible to build a fuzzy logic predictor based on the AR models. Khedkar and Keshav [12] used a slightly different model whose parameters can be adjusted according to estimation errors with a fuzzy feedback system as illustrated in Figure 3. The model is a variant of the Kalman predictor called the “exponential average predictor”. It was originally designed to make one step ahead prediction in an environment where both observation and system noise exist. We do not consider system noise in this work. The basic
predictor is adapted here with some modification [13]. Figure 3 only shows the case of one step prediction. It can also be expressed as:

\[ \hat{X}_{t,1} = \hat{X}_{t-1,1} + (1 - x)X_t \]

Where, \( \hat{X}_{t,1} \) is the one-step ahead predicted value of \( X_{t+1} \), made at moment \( t \), and \( \hat{X}_t \) is the true value at that moment. If this notation is followed, \( \hat{X}_t = X_t \) is true by definition.

Multi-step prediction can be achieved by applying the prediction recursively, which can be shown in the following equation:

\[ \hat{X}_{t,L} = \alpha \hat{X}_{t-1,L} + (1 - \alpha) \hat{X}_{t,L-1} \]

Where, \( \hat{X}_{t,L} \) is the \( L \)-step ahead predicted value of \( X_{t+L} \), made at moment \( t \). The \( \alpha \) in the above equation can to intuitively considered as the weight given to the past history. The larger the value \( \alpha \) is, the more weight is given to the past history. Putting it simply, if the prediction error is small, then \( \alpha \) should be large, and vice versa. Since the distribution of \( X_t \) is unknown, a fuzzy tuning system with the prediction error as input and \( \alpha \) as output provides the simplest solution.

Fuzzy linguistic rules for the inference of \( \alpha \) are easily established as:

- If prediction error is large then \( \alpha \) is small
- If prediction error is medium then \( \alpha \) is medium
- If prediction error is small then \( \alpha \) is large

Figure 3 shows typical input/output membership functions used in this work for simulation. Triangular shape membership function with only one parameter is used for both membership functions to reduce computation. In order to smooth \( \alpha \) further, a two stage fuzzy exponential averager system was also proposed but little advantage accrued from it.

Tables 1 and 2 provide sample comparisons of the proposed fuzzy predictor with the linear autoregression predictors of up to 4th order. The prediction was made on a two hours traffic trace consisting mainly video. Table 1 shows one-step ahead predictions while Table 2 shows ten-step predictions. The original trace has a mean \( (m) = 2.2363 \times 10^4 \) and standard deviation \( (s) = 1.7369 \times 10^3 \). These results are therefore reasonably accurate. It should be noted that the number of parameters required for AR predictions are the same as the order of the AR. For the fuzzy prediction, only one parameter \( \alpha \) is needed for the prediction. It can be seen that higher order AR processes do not have a clear advantage over lower order ones, but fuzzy prediction is generally advantageous compared with AR prediction. The error distribution closely resembles the distribution of the original series.

### Neural Network Implementation

The computation required for the proposed fuzzy predictor increases linearly with the number of prediction steps, due to the recursive application for multi-step prediction. In order to achieve fixed computation prediction for real-time operation, a 3-layer feed-forward neural network (FFNN) [14] with neurons in 10:10:1 structure is trained to perform the prediction. The input sequence is shifted through layer one and the output is provided at the single neuron of layer three, as illustrated in Figure 6.

<table>
<thead>
<tr>
<th>Table 1. Error statistics of one step predictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean (m)</td>
</tr>
<tr>
<td>AR(1)</td>
</tr>
<tr>
<td>AR (2)</td>
</tr>
<tr>
<td>AR(3)</td>
</tr>
<tr>
<td>AR(4)</td>
</tr>
<tr>
<td>Fuzzy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2. Error statistics of ten step predictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean (m)</td>
</tr>
<tr>
<td>AR(1)</td>
</tr>
<tr>
<td>AR (2)</td>
</tr>
<tr>
<td>AR(3)</td>
</tr>
<tr>
<td>AR(4)</td>
</tr>
<tr>
<td>Fuzzy</td>
</tr>
</tbody>
</table>

![Figure 3. Block diagram of Khedkar prediction system.](image-url)
Layers one and three of the neural network are used for input and output respectively. These two layers have linear transfer functions. Layer two is hidden, and neurons in this layer use hypertangent sigmoid transfer function, as it is usually the case for FFNN. The input and output data sets from the original fuzzy predictor are used for training. Training sequences are selected to have similar mean and variance values to those encountered in production mode. Table 3 shows a performance example of the neural fuzzy predictor. It can be seen that the prediction results are not as good as, but are close to the original fuzzy predictor. This trend lasts beyond the ten steps illustrated in the table. The input traffic trace has a mean of $2.5044 \times 10^4$ and standard deviation of $1.687 \times 10^3$, and it is not a training sequence.

### Table 3. Error comparison of the fuzzy predictor and its neural network implementation

<table>
<thead>
<tr>
<th>Steps</th>
<th>Original fuzzy predictor</th>
<th>Neural network implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean (m)</td>
<td>Deviation (σ)</td>
</tr>
<tr>
<td>1</td>
<td>-88.46</td>
<td>810.5</td>
</tr>
<tr>
<td>2</td>
<td>-1.953</td>
<td>840.5</td>
</tr>
<tr>
<td>3</td>
<td>-119.4</td>
<td>765.3</td>
</tr>
<tr>
<td>4</td>
<td>20.41</td>
<td>827.8</td>
</tr>
<tr>
<td>5</td>
<td>-180.3</td>
<td>858.6</td>
</tr>
<tr>
<td>6</td>
<td>100.2</td>
<td>763.4</td>
</tr>
<tr>
<td>7</td>
<td>-197.2</td>
<td>798.4</td>
</tr>
<tr>
<td>8</td>
<td>160.8</td>
<td>770.6</td>
</tr>
<tr>
<td>9</td>
<td>-183.2</td>
<td>769.5</td>
</tr>
<tr>
<td>10</td>
<td>193.9</td>
<td>753.7</td>
</tr>
</tbody>
</table>

Figure 5 shows the computation requirements to operate the fuzzy predictor and its neural network counterpart. It can be seen that the cross over point is just below 20 steps, which means FFNN can be applied for less computation when the prediction is 20 steps or more. When 80-step prediction is necessary as discussed in [8], the amount of computation required for the FFNN is about 25% of the original fuzzy predictor.

### Conclusions

Fuzzy systems offer viable alternatives to analytical methods that are difficult to establish or inaccurate in description. For this specific application, it can be clearly seen from Tables 1, 2, and 3 that the fuzzy logic prediction has smaller error mean and error standard deviation than the autoregressive prediction. This indicates the fuzzy predictor is more accurate. From Figure 7, it can be seen that the neural network implementation requires about 250 flops of computation for each prediction, which is independent of prediction steps. The amount of computation is considered quite
small by today’s standards. It is therefore possible to achieve multi-step ahead prediction on-line and in real-time without the use of dedicated hardware.

References