On-Line Prediction of Nonstationary Variable-Bit-Rate Video Traffic

Sungjoo Kang, Seongjin Lee, Youjip Won, and Byeongchan Seong

Abstract—In this paper, we propose a model-based bandwidth prediction scheme for variable-bit-rate (VBR) video traffic with regular group of pictures (GOP) pattern. Multiplicative ARIMA process called GOP ARIMA (ARIMA for GOP) is used as a base stochastic model, which consists of two key ingredients: prediction and model validity check. For traffic prediction, we deploy a Kalman filter over GOP ARIMA model, and confidence interval analysis for validity determination. The GOP ARIMA model explicitly models inter and intra-GOP frame size correlations and the Kalman filter-based prediction maintains “state” across the prediction rounds. Synergy of the two successfully addresses a number of challenging issues, such as a unifile framework for frame-type dependent prediction, accurate prediction, and robustness against noise. With few exceptions, a single video session consists of several scenes whose bandwidth process may exhibit different stochastic nature, which hinders recursive adjustment of parameters in Kalman filter, because its stochastic model structure is fixed at its deployment. To effectively address this issue, the proposed prediction scheme harbors a statistical hypothesis test in the prediction framework. By formulating the confidence interval of a prediction in terms of Kalman filter components, it not only predicts the frame size but also determines validity of the stochastic model. Based upon the results of the model validity check, the proposed prediction scheme updates the structures of the underlying GOP ARIMA model. We perform a comprehensive performance study using publicly available MPEG-2 and MPEG-4 traces. We compare the prediction accuracy of four different prediction schemes. In all traces, the proposed model yields superior prediction accuracy than the other prediction schemes. We show that confidence interval analysis effectively detects the structural changes in the sample sequence and that properly updating the model results in more accurate prediction. However, model update requires a certain length of observation period, e.g., 60 frames (2 s). Due to this learning overhead, the advantage of model update becomes less significant when scene length is short. Through queueing simulation, we examine the effect of prediction accuracy over user perceivable QoS. The proposed bandwidth prediction scheme allocates less 50% of the queue (buffer) compared to the other bandwidth prediction schemes, but still yields better packet loss behavior.

Index Terms—Confidence interval analysis, GOP ARIMA, Kalman filter, MPEG, multimedia, nonstationary, scene change detection, traffic prediction, variable-bit-rate (VBR).

I. INTRODUCTION

A. Motivation

RAPID advances in the performance of hardware, communication networks, storage and video compression technologies enable the user to access video contents in ubiquitous fashion. The typical application includes various types of video streaming services, e.g., video-on-demand, video conference, real-time streaming of news and sports events, each of which requires widely different bandwidths and QoS requirements, e.g., HDTV (45 Mbits/s), video conferencing (4.5 Mbits/s), VoIP (0.3 Mbits/s), data (10 Mbits/s) and on-line games (2 Mbits/s) [1]. For variable-bit-rate video traffic which usually has slowly decaying sample autocorrelations, static bandwidth allocation results in an overprovisioning of bandwidth and causes bandwidth wastage [2]–[4]. Accurate real-time prediction of the future bandwidth process is very important in many aspects: fair bandwidth utilization, dynamic bandwidth allocation, end-to-end QoS control of real-time multimedia streams, etc. All these issues are critical in cell/packet-based B-ISDN (e.g., ATM), best effort Internet [5], [6] or in circumstances where per-flow QoS management is enabled [7]. Furthermore, recent deployment of wireless networks calls for more efficient use of network bandwidth. For example, IEEE 802.15.2 wireless network (piconet) requires careful modeling of VBR video for dynamic bandwidth allocation [8], [9].

There are a number of issues which a good bandwidth prediction scheme should address. First, a prediction scheme should be able to exploit the sample correlation structures in predicting the future frame size. The linear prediction method and neural network method do not properly incorporate the correlation structures of the underlying frame size sequence. Particularly, the linear prediction models use separate models for each frame type sequence and therefore is incapable of representing correlation structures among different types of frames. Second, the prediction scheme should be robust against noise and should converge fast. There are three, I, B, and P, frame types in MPEG coding scheme. I frame, which is also known as intracoded frames, is coded as a single frame, without references to any other frames. Predictive coded frame or P frame contains the difference of earlier I or P frames in the GOP. And B frame or bidirectional coded frame contains the difference from earlier and later I or P frames in the sequence. The first frame in the new scene is compressed via intracoding regardless of the frame type. Among the three frames, intracoded B type frames
are typical noisy input. This may result in an exceptionally large B-type frame. A good prediction model should be able to properly filter out noisy samples in predicting the future frame size. Simple linear prediction is vulnerable to noisy input. A neural network-based learner cannot quickly adapt to short term structure changes in the underlying sequence. Third, the prediction scheme should be able to detect the structural changes in the underlying sequence, e.g., scene change, and should be able to update the prediction model accordingly.

In this paper, we propose a model-based bandwidth prediction scheme for VBR video traffic with a regular group of picture (GOP) pattern. We use the GOP ARIMA (ARIMA for Group of Pictures) model as a base stochastic model for the underlying sequence [10]. Our prediction framework consists of two major components: frame size prediction and model update. For accurate and robust prediction, we deploy a Kalman filter over the base stochastic model, GOP ARIMA. An advantage in using GOP ARIMA as the base stochastic model over other models is that GOP ARIMA is designed to preserve correlations among different types of frames. With GOP ARIMA, we can make frame type dependent prediction with a single model. Many of the studies on VBR frame size prediction use separate predictors for each frame type sequence, which results in the loss of important correlation information. With GOP ARIMA, it is not necessary to use separate prediction model for each frame type sequence. Kalman filter-based recursive error adjustment maintains “state” across prediction rounds. Frame type aware prediction becomes more accurate and robust against unusual input, e.g., intracoded B frame. The stochastic nature of the underlying frame size sequence is highly subject to the nature of a scene, e.g., motion dynamics. We argue that when consecutive scenes have different stochastic structure, the prediction model needs to be updated properly to reflect the stochastic nature of underlying frame size sequence. Recursive error adjustment in the Kalman filter may not be sufficient to properly incorporate the fundamental changes in the underlying frame size sequence.

In this study, we develop an efficient statistical hypothesis test technique to determine the validity of prediction model. We model the confidence interval of the frame size estimation in terms of the Kalman filter components, i.e., process matrix, measurement matrix, state and error covariance. Our confidence interval-based analysis not only effectively detects scene change but also provides rigorous ground on prediction accuracy. The proposed model-based prediction method significantly improves the prediction accuracy and prediction responsiveness compared to existing linear prediction-based methods and neural network-based methods.

The proposed prediction algorithm predicts the future frame size solely based upon underlying frame size sequence and GOP structure. It does not require any knowledge on source coding algorithm and/or rate control algorithm at the source end. We analyze the effectiveness of the proposed prediction algorithm via examining the prediction accuracy of three different prediction schemes on total of six video traces (three MPEG-2 video traces and three MPEG-4 video traces). These video traces are chosen to represent different degrees of motion dynamics in underlying scenes and different source coding standards. Unfortunately, however, only frame size sequences were available and details of source coding algorithms were not available to public [11]–[13]. In all cases, the proposed algorithm (GOP ARIMA-based prediction) outperforms the existing prediction algorithms proposed by Yoo [14], Adas [15].

B. Related Works

A fair amount of research has been dedicated to developing a model for MPEG VBR traffic. Garret et al. [16] analyzed the long range dependent property of VBR traffic in the context of self-similarity. Lucantoni et al. [17] modeled the VBR source using the Markov renewal process. Krunz et al. [18] examined the structure of the empirical VBR process in a relatively smaller time scale and proposed a simple model which can simulate the autocorrelation structure of interframe coded video traffic.

Many studies also analyzed the marginal distribution of the underlying sequence. Doulamis et al. [19] used the AR(1) process to represent the relationship between each type of frame in consecutive GOP’s and added another layer to represent inter-GOP behavior. Turaga et al. [20] enhanced this model by using the doubly Markov process to model irregular GOP patterns. A common limitation of these models is that AR(1) process has a geometrically bounded autocorrelation function and thus cannot properly reflect the slowly decaying autocorrelation structure. Lombardo et al. [21] modeled the frame size correlations within GOP and proposed an algorithm to generate MPEG traces which have long range dependent (LRD) properties at the GOP level.

Several works explicitly model the time dependent behavior, i.e., regular GOP pattern of VBR traffic: ARIMA [22], gamma-best autoregression (GBAR) model [23], GOP GBAR model [24], and GOP ARIMA model [10]. However, the GBAR and GOP GBAR models yield exponentially decreasing autocorrelations while empirical VBR traffic has much more slowly decreasing autocorrelations. The GOP ARIMA model successfully represents the slowly decaying sample autocorrelations of the VBR sequence. The ARIMA model has also been used to model the nonstationary aspects of traffic in networked games [25] and I/O workload [26].

Traffic prediction requires in depth understanding of the fundamental stochastic structure, e.g., sample autocorrelations, of the underlying sequence. In predicting the future bandwidth process, Manzoni et al. [27] used the aggregate characteristics, i.e., the first- and the second-order statistics of the I frame size sequence. They did not consider the autocorrelation structure of the underlying sequence. Grossglauser et al. [28] proposed the use of Renegotiated Constant Bit Rate (RCBR) service to support VBR video. They based their approach on the AR(1) model and used heuristics to predict the future bandwidth. Adaptive linear prediction (ALP) has been popular for real-time prediction of the VBR bandwidth process [15], [29]. The recursive least square (RLS) predictor is sometimes used as an alternative to the least mean square (LMS) method for better convergence although the computational complexity of RLS is much higher. While linear prediction is very simple and fast, it does not properly capture the inter- and intra-GOP correlations. In addition, it cannot quickly react to structural changes in the underlying sequence, e.g., scene change, and it is vulnerable to noisy input, e.g., intracoded B frame. Yoo [14] extended the work of Adas et al. [15] by incorporating a threshold-based scene detection scheme. Some studies predict
the VBR traffic from frequency domain analysis. Chong et al. [30] analyzed traffic in frequency domain and used RLS and time-delayed neural network (TDNN) methods to predict the future bandwidth process. Wang et al. [31] analyzed the VBR process using wavelets. They effectively resolved the slow convergence problem in least-mean-square (LMS)-based prediction.

Neural network-based prediction can be useful for long term prediction including scene length prediction and scene change detection [30], [32]–[34]. The learning overhead of a neural network-based approach may prohibit the quick adaptation to structural changes in the underlying sequence. Bhattacharya et al. [35] used Recurrent Neural Networks (RNN) to predict MPEG-coded video traffic. They designed a single-step and multistep predictor via multiresolution learning. Their results are quite accurate, but were only conducted on at most a four step-ahead prediction, which is considered very short-term when allocating dynamic bandwidth. According to [35], it takes more than an hour to train the neurons, which produces accurate single step prediction than Yoo [14] and Adas [15]. However, with 30 frames/s playback, one step prediction is to predict the frame size which will arrive in next 33 ms. It is practically infeasible to adjust bandwidth allocation in every 33 ms. Gupta et al. [36] studied the performance of multistep prediction of neuro-predictors and linear predictors. He showed that autoregressive Exogenous model performed 23% better than Recurrent Multilayer Perceptron [35]. Kalman filters have been effectively used to predict the burstiness of network traffic [37] and to detect the spread of worms in networks [38].

Detection of scene change has been under intense research for more than a decade [39]–[42]. Fair amount of works have been dedicated on scene change detection from content analysis point of view. Content-based analysis is mainly for off-line computation and indexing, e.g., video annotation, finding key frames, automatic generation of video summary, etc. These techniques use color histogram, motion vector, brightness and etc. which is automatic generation of video summary, etc. These techniques use color histogram, motion vector, brightness and etc. which is obtained after uncompressing the video [43]–[46]. Due to its computational complexity and processing overhead, content-based scene change detection methods do not fit for real-time bandwidth prediction and resource allocation.

The rest of this paper is organized as follows. Section II introduces the GOP ARIMA model. Section III introduces on-line prediction algorithms. Section IV discusses the state system model of GOP ARIMA and the respective Kalman filter equations. Section V analyzes the characteristics of the proposed prediction technique. In Section VI, we present confidence interval analysis to detect the structural changes in underlying sequence. Section VII presents the notion of a sampling window for the GOP ARIMA model. Section VIII presents the results of the experiments and Section IX concludes the paper.

II. SYNOPSIS: GOP ARIMA MODEL

A. Statistical Characteristics of VBR Video Traffic

There are three types of frames in an MPEG coding scheme: I, P and B. Frame type I is self-contained. Frame type P carries the information difference from the preceding I or P type frame. Frame type B contains the interpolated information between consecutive I or P frame type pairs [47]. The GOP structure specifies the number and temporal order of P and B frames between two successive I frames. GOP structure is represented by GOP(S, s), where S is the distance between successive I frames and s is the distance between consecutive P frames or the distance between I frame and following P frame. For example, GOP(15,3) denotes the frame sequence "I B B P B B P B B P B B B B B B B B". A fixed GOP pattern is used primarily to achieve the random access granularity requirement with minimum decoder complexity and with maximum compression ratio. An MPEG-2 coding scheme with GOP structure, GOP(15,3), is used as the source coding standard for digital broadcasting service [48], [49]. There are a few important characteristics which have been commonly observed in most empirical VBR processes. The first characteristic is the slowly decaying sample autocorrelations. The second characteristic is the periodicity in the frame size sequence. There is not much difficulty in finding the clear cause for these characteristics: regular GOP structure. Fig. 1 illustrates the sample frame size sequence and the sample autocorrelations for MPEG-2 GOP(15,3) compressed video.

We use the term nonstationary to denote the VBR frame size process which has strict time dependent behavior. In our case, strict time dependent behavior corresponds to a regular GOP pattern. A process is said to be covariance stationary if covariance of the time series. $X_t, t = 1, 2, \ldots, E[(X_t-\mu)(X_{t+k}-\mu)]$, is independent of $k$.

B. GOP ARIMA Model

GOP ARIMA is designed to model the frame size sequence with regular GOP pattern. It very well captures the correlation structures between the frame sizes within a GOP as across GOP’s, i.e., inter-GOP and intra-GOP correlations. The GOP ARIMA model is a special form of the multiplicative seasonal ARIMA model [50]. The ARIMA model is widely used in modeling, analysis and prediction of nonstationary time series, e.g., economic data or weather data. There are two ways to handle time dependent components in a time series. The first is to seasonally adjust the data, to construct the suitable forecast model and then to add the seasonal effect back to the forecast function. The second is to directly embed the seasonal component in the forecast model. Many of the existing works take the first approach. They remove the seasonality by aggregation, addition, or subtraction on the original time series. However, removal of seasonality prior to establishing model risks losing important correlation information. GOP ARIMA addresses this problem by building the seasonal components directly into the
model. The GOP ARIMA model explicitly models deterministic time dependent behavior of regular GOP pattern. The GOP ARIMA elaborately incorporates the intra- and inter-GOP correlation structures of the empirical VBR process and provides rigorous explanation of the slowly decaying sample autocorrelations of the empirical process. It delivers what Norros terms casual concrete understanding as well as abstract statistical understanding [51].

A fair amount of studies has been dedicated to developing a good model for compressed video traffic. Several stationary time series models, e.g., AR(1) [52], DAR(1) [53], ARMA [54], Markov Renewal process [55] were proposed. The notion of self-similarity was also introduced to explain the slowly decaying sample autocorrelations in MPEG video dynamics [56], [57]. These models require that underlying time series is covariance stationary. Frame size sequence with regular GOP pattern has strict time dependent behavior and therefore is not stationary.

A number of works proposed to use separate models, one for GOP and one for intra-GOP frame size sequence to represent nonstationarity in frame size sequence [58]–[60]. The drawback of this approach is that separating the GOP level (aggregation at GOP scale) modeling and intra-GOP frame level modeling may not represent the sample correlations at lags of GOP size (e.g., 15 frames). To preserve correlations, Frey and Nguyen-Quang [61] proposed a nonstationary process called GOP GBAR model, which is based on Heyman’s gamma-beta autoregression (GBAR) model [62]. The GOP GBAR model requires that the variance of B frame sizes should be less than the variance of P frame sizes and that the variance of P frame sizes is less than the size of I frame sizes. This assumption does not always hold [10]. GOP ARIMA model well addresses the problem of existing frame size sequence model.

The basic idea of the GOP ARIMA model is to decompose a sequence into subcomponents and to find a proper model for each subsequence and to identify the correlation structures within as well as between the subsequences. The GOP ARIMA model is represented as GOP ARIMA \((p,d,q)_{s}\times(F,D,Q)_{S}\). Where \(p\) and \(q\) are the order of autoregressive and moving average, respectively, \(d\) denotes the difference order, \(s\) and \(S\) denote the length of seasonal lags, i.e., P-to-(P or I) frame distance and I-to-I frame distance. In case of GOP\((15,3)\), \(s\) and \(S\) corresponds to 3 and 15, respectively, \(p\) and \(q\) denote the autoregressive orders, \(d\) and \(D\) denotes the difference orders, and \(P\) and \(Q\) denote the moving average orders for modeling the intra- and inter-GOP correlations. Let \(X_t\) be a frame size sequence of VBR compressed video with GOP\((S,s)\). Since this time series consists of the sizes of I, P, and B frames, we decompose the sample process \(X_t\) as follows:

\[
X_t = x_t^S + x_t^P + \epsilon_t
\]

where \(x_t^S\) and \(x_t^P\) denote the seasonal components which appear in every \(s^{th}\) and \(S^{th}\) samples, respectively, and \(\epsilon_t\) is stochastic components of the sample sequence.

In this paper, we develop frame size prediction method for fixed GOP pattern and do not address the situation where different GOP patterns coexist. In practice, GOP pattern is explicitly specified at the header of the video content. Therefore, it is possible to build new model on-line based upon the underlying sequence whenever GOP pattern of the underlying sequence changes. Further, it is also possible to identify GOP structure automatically. The model in (1) can also be modified in order to handle situations where different structures coexist. For example, we can use an extended additive model such as \(X_t = \sum_{i=1}^{\#s} x_t^{i} + \epsilon_t\), where \(\#s\) denotes the number of seasonal components. In some applications, the components may be combined multiplicatively, for more detail see Durbin and Koopman [63]. See the work of [10], which provides thorough analysis and examples on the structure of GOP ARIMA in nonstationary VBR process, e.g., GOP(6,3), GOP(9,3), GOP(30,3), etc.

Determining the GOP ARIMA model for a given empirical frame size process consists of two steps: (i) removing the nonstationary, i.e., time dependent, components and (ii) ARMA fitting. In time series analysis, it is common practice to first remove the time dependent components from the underlying sequence to make it more analyzable. First, we remove the seasonal components by taking the difference at the lags of 3rd and 15th. The resulting time series, say \(Y_t\), then becomes more analyzable one. We use backward operator \(B\) which is widely used in statistics to make the time series expression more compact. \(B^k X_t\) denotes \(X_{t-k}\). \((1-B)X_t\) thus denotes the differenced time series, \(X_t - X_{t-1}\). We can perform this differencing operation multiple times for each lag. The number of differencing operations are called difference orders and are denoted as \(D\) and \(d\) for a lag of 15 and a lag of 3, respectively. The differenced process at the lag of 3 and 15 can be obtained by applying the \((1-B^3)^D(1-B^{15})^D\) operator. Differenced process \(Y_t\) can be formulated as

\[
Y_t = (1-B^3)^d(1-B^{15})^D X_t.
\]

The second step is ARMA fitting. Since we take the difference from the original process, the resulting process \(Y_t\) now becomes a multiplicative ARMA process. We denote it by ARMA\((p,q)_3\times(F,Q)_{15}\). ARMA\((p,q)\) and ARMA\((P,Q)\) are used to model the intra- and inter-GOP sample auto correlations of the underlying sequence. For the GOP\((15,3)\) process, they correspond to \(ARMA(p,q)_{3}\) and \(ARMA(P,Q)_{15}\). There are a number of ways to determine the difference orders \((d\) and \(D)\), autoregressive orders \((p\) and \(P)\) and moving average width \((q\) and \(Q)\) of the GOP ARIMA models. Some of them require human interaction, e.g., using least square or maximum-likelihood estimator, and others do not, e.g., Schwartz Bayesian Criterion (SBC) and Akaike Information Criterion (AIC) [50], [64]. GOP ARIMA model for VBR frame size sequence in Fig. 1 is formulated as in (3), and parameters are fitted using SBC

\[
(1-B^{15})(X_t-2.757) = (1-0.7303B^{15})\epsilon_t, \quad \hat{\sigma}^2 = (6680.3)^2.
\]

Consult [10] for thorough analysis on GOP ARIMA for nonstationary series.

If underlying time series is known to have multiplicative seasonality, we can easily determine the length of a period or multiplicative periods. We can determine the cycle length by taking the difference of samples which are D,
(D = 2, 3, 4, · · · , 30), lags apart and checking if the resulting time series is stationary. Generally, the determination of differencing order D is performed by various types of unit roots test such as augmented Dickey-Fuller test [65]. The unit root test can be also done by on-line or automatically, for example, see [66]. Once the periods of the underlying time series is obtained, we can apply standard method to build a times series for an underlying sequence [10].

III. SYNOPSIS: ON-LINE PREDICTION OF VBR VIDEO TRAFFIC

A. On-Line Prediction Algorithms

Exponential smoothing is widely used to predict the future value of time series samples by taking the linear combination of the past values: \( \hat{X}_{t+1} = \alpha \hat{X}_t + (1 - \alpha) \hat{X}_t \), where \( \hat{X}_t \) and \( \hat{X}_t \) correspond to estimate and observation at time \( t \) and \( 0 < \alpha < 1 \). Exponential smoothing does not excel when there is a trend in the data. Double exponential smoothing (DES) is used when the data shows a trend. The smoothing with a trend works much like simple smoothing except that two components must be updated each period-level (DES1, •) and trend (DES2, •). The level is a smoothed estimate of the value of the data at the end of each period. The trend is a smoothed estimate of average growth at the end of each period. The specific formula for DES is: \( \text{DES}_{1,t} = \alpha \hat{X}_t + (1 - \alpha)(\text{DES}_{1,t-1} + \text{DES}_{2,t-1}) \) and \( \text{DES}_{2,t} = \beta(\text{DES}_{1,t} - \text{DES}_{1,t-1}) + (1 - \beta) \text{DES}_{2,t-1} \), for \( 0 < \alpha < 1 \) and \( 0 < \beta < 1 \). Note that the current value of the series is used to calculate its smoothed value replacement in double exponential smoothing. Therefore, 1-step prediction in DES can be calculated as \( \hat{X}_{t+1} = \text{DES}_{1,t} + \text{DES}_{2,t} \), because the series in DES are assumed to be composed with two components, level and trend. Naturally, an N-step prediction is as

\[
\hat{X}_{t+N} = \text{DES}_{1,t} + N \cdot \text{DES}_{2,t}.
\]

For more details on DES, see [67]. Laviola examines the effectiveness of Double Exponential Smoothing and Kalman filter in predictive motion tracking [68].

ALP is widely used because of its simplicity and relatively good performance. It does not require any prior knowledge of data statistics, nor does it assume stationarity. By [14], a \( p^{th} \)-order linear predictor has the form

\[
\hat{X}_{t+N} = \sum_{l=0}^{p-1} w_l(t)X_{t-l} = w^T_n X_t
\]

where \( w_l = (w_l(0), \ldots , w_l(p-1))^T \) denotes a prediction filter coefficient vector which minimizes the mean square error, and \( X_t \) is a set of current and previous values of \( X_t \). Parameter \( p \) indicates the number of past values used for estimation. Let \( e_t = X_{t+N} - \hat{X}_{t+N} \). Starting with an initial estimate of the filter coefficient \( w \), and for each new data point, the ALP method updates \( w \) using the following recursive equation by [14]:

\[
w_{t+1} = w_t + \frac{\mu e_t X_t}{||X_t||^2}.
\]

The step size \( \mu \) is fixed during the entire prediction. If \( 0 \leq \mu < 2 \), then the least mean square error will converge to the mean. The use of a large \( \mu \) results in faster convergence and quicker response to traffic change. In contrast, a small \( \mu \) results in slower convergence with less fluctuation after convergence [15].

B. Kalman Filter and On-Line Prediction

A Kalman filter is a kind of an adaptive filter that provides a recursive solution to the linear optimal filtering problem. A Kalman filter is essentially a set of mathematical equations and state space models that implements a predictor-corrector type estimator that is optimal in the sense that it minimizes the estimated error covariance. It applies to nonstationary as well as stationary processes. For more details on Kalman filter, see [67]. A Kalman filter has two important vectors: state and measurement.

The state vector \( F_t(M \times 1) \) is the minimal set of data to describe the dynamic behavior of the system. In other words, the state is the least amount of data about the past behavior of the system that is needed to predict its future behavior. The measurement vector \( X_t(N \times 1) \) is a measurement at time \( t \). The Kalman filter uses two equations: the Process Equation and the Measurement Equation. The Process Equation is used to predict the state of the system at \( t + 1 \) for a given \( F_t \) and is defined as

\[
F_{t+1} = A_t F_t + W_t.
\]

The \( N \times M \) matrix \( A_t \) in (7) is called the process matrix. The process noise, \( W_t \) is assumed to be a zero-mean, additive, white Gaussian process with the process noise covariance matrix \( Q_t \) defined by \( E[W_{n}W_{n}^T] = \begin{cases} Q_t & \text{for } n = t, \\ 0 & \text{for } n \neq t. \end{cases} \)

The Measurement Equation derives the measurement from the state. Equation (8) is the definition of the measurement equation

\[
X_t = H_t F_t + V_t.
\]

The \( N \times M \) matrix \( H_t \) is called the measurement matrix. The measurement noise \( V_t \) is assumed to be a zero-mean, additive, white Gaussian process with measurement noise covariance matrix \( R_t \) defined by \( E[V_{n}V_{n}^T] = \begin{cases} R_t & \text{for } n = t, \\ 0 & \text{for } n \neq t. \end{cases} \)

The Process noise \( W_t \) and the measurement noise \( V_t \) are uncorrelated with each other. Let the state-error vector \( E_t \) be the difference between the state \( F_t \) and the estimated state \( \hat{F}_t \), i.e., \( E_t = F_t - \hat{F}_t \). We define the error covariance matrix \( P_t \) as \( E[(F_t - \hat{F}_t)(F_t - \hat{F}_t)^T] \), for simplicity we will put it as \( E[EE_t^T] \). Kalman filtering operates by predicting and correcting recursively. In the Kalman filtering algorithm we use a pair of time points priori- and posteriori-. At time \( t \), we already have an estimate of state \( \hat{F}_t \) predicted at time \( t-1 \). We call this priori estimate of the state, \( \hat{F}_t^- \) (a ‘-’ symbol over \( F_t \) means a priori estimate). We call this process prediction. In a similar manner, priori error covariance matrix \( P_t^- \) at time \( t \) is defined as \( P_t^- = E[(F_t - \hat{F}_t^-)(F_t - \hat{F}_t^-)^T] \). Via linear combinations of the priori estimate and a measurement \( X_t \) made at time \( t \), we generate a posteriori estimate of the state, \( \hat{F}_t \). This process is called correcting. Given \( P_t^- \), we can compute the posteriori covariance matrix \( P_t \) at time \( t \) as \( P_t = E[(F_t - \hat{F}_t)(F_t - \hat{F}_t)^T] \).
model of the Kalman filter. This is normal practice in time series-based forecasting [50]. The term state space is mostly used by control engineers to model systems which vary over time. One of the advantages of using the Kalman filter is that it enables us to predict not only immediate future frame size but also to make long term prediction if future frame size. In a state space model, measurement at \( t \) is taken to be the linear combination of state variables, which as a whole constitutes the state of a model.

To establish a state space model for Kalman filter, we first define state and measurement of the model in the context of the GOP ARIMA. Measurement at \( t \) corresponds to frame size at \( t \) denoted by \( X_t \). Generally, the measurement (or observation, equivalently) is assumed to be contaminated by error and most control model embodies the error in its measurement model. In our context, however, frame size we observe is actual size of the frame and does not contain any error. In our model, we do not have measurement error term in measurement equation (8). State of the system is minimal set of current and past values of variables upon which the future state can be determined. State vector is collection of these variables.

In deriving a state space model from GOP ARIMA, we need to represent \( X_t \) as a linear combination of its past values and some stochastic components. To help the understanding, we proceed with an example model, GOPARIMA\((1,1,1)_3 \times (1,1,1)_{15}\). Let \( X_0 \) and \( Y_t \) be a frame size sequence and a differenced process from \( X_t \), i.e.,

\[
Y_t = (1 - B^3)(1 - B^{15})X_t,
\]

respectively. Then, \( X_t \), can be represented as

\[
X_t = X_{t-3} + X_{t-15} - X_{t-18} + Y_t. \tag{12}
\]

The differenced process \( Y_t \) is a multiplicative ARMA process, ARMA\((1,1)_3 \times (1,1)_{15}\) and can be represented as in

\[
(1 - \phi B^3)(1 - \Phi B^{15})Y_t = (1 + \theta B^3)(1 + \Theta B^{15})\epsilon_t
\]

where \( \phi, \Phi, \theta, \) and \( \Theta \) denote coefficients in moving average and autoregressive expression. To make (13) more manageable, we introduce AR process \( Z_t \),

\[
Z_t = (1 - \phi B^3)(1 - \Phi B^{15})Z_t = \epsilon_t.
\]

With \( Z_t \), we rewrite \( Y_t \),

\[
Y_t = Z_t + \theta Z_{t-3} + \Theta Z_{t-15} + \theta \Theta Z_{t-18}. \tag{14}
\]

From (12), (13), and (14), \( X_t \) can be represented by the linear combination of its past values and the autoregressive process \( Z_t \)

\[
X_t = X_{t-3} + X_{t-15} - X_{t-18} + Z_t + \theta Z_{t-3} + \Theta Z_{t-15} + \theta \Theta Z_{t-18}.
\]

(15)

\[
X_{t-3}, X_{t-15}, X_{t-18}, Z_t, Z_{t-3}, Z_{t-15}, \text{ and } Z_{t-18}
\]

are sufficient to represent \( X_t \). Since these terms are a certain lag apart, there need \( X_{t-3}, X_{t-15}, X_{t-18} \) and \( Z_t, Z_{t-3}, Z_{t-15}, \) and \( Z_{t-18} \) to properly represent the \( X_t \). Now, we define the state of GOP ARIMA model. Let \( X_t \) and \( Z_t \) be the set of the most recent eighteen values of \( X_t \)’s and the most recent nineteenth values of \( Z_t \)’s, i.e., \( X_{t-1} = [X_{t-18}, \ldots, X_{t-1}]^T \) and \( Z_{t-1} = [Z_{t-18}, Z_{t-17}, \ldots, Z_{t}]^T \). From (15), frame size \( X_t \) can be obtained using the linear combination of the components in \( X_{t-1} \) and \( Z_t \). We define a state \( \mathbf{F}_t \) of GOPARIMA\((1,1,1)_3 \times (1,1,1)_{15}\) as the concatenation of

\[
\begin{align*}
\text{Fig. 2. Kalman filtering algorithm.}
\end{align*}
\]
**B. Prediction of State**

The prediction scheme determines the future state of a system based upon some predefined evolution rule and the current state information. In Kalman filter, process equation defines the rule of evolution. Particularly in model-based prediction, process equation (or process matrix) embodies the time series model upon which the Kalman filter is deployed. We develop a process equation for our prediction model.

The state of our system consists of two vectors $Z_t$ and $X_{t-1}$. As time proceeds, individual elements in these two vectors are shifted to its left. The right most elements in $Z_t$ and $X_{t-1}$ are shifted out. The left most positions of these vectors are filled with new values. The new values in $Z_t$ and $X_{t-1}$ are obtained according to the autoregressive model (17) and GOP ARIMA model (15), respectively. In this light, we can partition the process matrix into two components: those for updating $Z_t$ and those for updating $X_t$.

We first discuss how to update the autoregressive process $Z_t$, $(1-\phi B^3)(1-\Phi B^{15})Z_t = \epsilon_t$ and the respective process matrix.

In GOP ARIMA $(1,1,1)^3 \times (1,1,1)^{15}$, $Z_t$ is represented by the linear combination of its previous values and some stochastic component as

$$Z_t = \phi Z_{t-3} + \Phi Z_{t-15} - \phi \Phi Z_{t-18} + \epsilon_t$$

$$\epsilon_t \sim WGN(0, \sigma^2)$$  \hspace{1cm} (17)

Let $\Gamma$ be process matrix for $Z_t$. Basically, $\Gamma$ is responsible for shifting the elements in the current state vector and for taking the linear combination of the state variables according to (17) to obtain the new one. We can obtain the process equation for one-step state prediction as $Z_{t+1} = \Gamma Z_t + W_t$, $W_t = [0 \cdots 0 \epsilon_{t+1}]$.

Equation (18) illustrates this equation:

$$
\begin{bmatrix}
Z_{t-17} \\
Z_{t-16} \\
\vdots \\
Z_t \\
Z_{t+1}
\end{bmatrix} =
\begin{bmatrix}
0 & 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 0 & 1 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 1 & 0 \\
0 & \phi \Phi & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
\theta \Theta & 0 \cdots & \Theta & 0 \cdots & \Theta & 0 \cdots & 1 - 1 & 0 \cdots & 1 & 0 \cdots & 1 & 0 \cdots & 0
\end{bmatrix}
\begin{bmatrix}
Z_{t-17} \\
Z_{t-16} \\
\vdots \\
Z_{t} \\
Z_{t+1}
\end{bmatrix} +
\begin{bmatrix}
0 \\
0 \\
\vdots \\
0 \\
\epsilon_{t+1}
\end{bmatrix} \hspace{1cm} (18)
$$

Evolution of $X_t$ follows precisely the same fashion with the evolution of $Z_t$. The elements in the current state vector get shifted and the new value is derived based upon some stochastic model. The only difference between the updates (or evolution equivalently) of $Z_t$ and $X_t$ is the way how the new value is computed. For $X_t$, the new value is computed based upon GOP ARIMA model in (15) while for $Z_t$, the new value is computed based upon autoregressive model in (17). Combining the process matrix for $Z_t$ and $X_t$, we finally establish the process matrix $A$ and error $W_t$ for GOPARIMA $(1,1,1)^3 \times (1,1,1)^{15}$ as in (19) and (20), respectively, shown at the bottom of the page.

Fig. 3 schematically represents the organization of the process matrix $A$ in (19) and error $W_t$. In practice, process matrix has a sparse and very simple structure. The process matrix $A$ consists of process matrix for $Z_t$ and $X_{t-1}$. The fast
row of the process matrix $A$ is used to derive $X_t$, which is shown in (15).

As a final step, we develop a measurement equation which obtains measurement from the state. Recall that the last element of $F_{t+1}$ is $X_{t+1}$. Therefore, prediction of the measurement can be directly obtained from predicted state. We can define the measurement matrix $H$ as the last row of the process matrix $A$, i.e., $H = [A_{37,1}, A_{37,2}, \ldots, A_{37,37}]$. We can obtain frame size $X_t$ as $X_t = HF_t + V_t$. Note that there is no measurement error in frame size prediction and $V_t = 0$.

V. ANALYSIS OF KALMAN FILTER-BASED PREDICTION

A. Robustness and Convergence

The Kalman filter has robust properties in that it is insensitive to the distributions of the process noise $W_t$ (7) and measurement noise $V_t$ (8). For example, when the distributions are non-Gaussian, the Kalman filter still works well (see, e.g., [64]). The Kalman filter generally produces a good prediction in spite of various and uncertain kinds of noisy input. Furthermore, compared with prediction using a neural network algorithm, prediction by the Kalman filter is very fast in its convergence. Watson [69] showed that the convergence rate of the prediction is the squared root of the sample size. Accordingly, prediction by the Kalman filter is very fast in its convergence. Watson [69] showed that the convergence rate of the prediction is the squared root of the sample size. Accordingly, prediction by the Kalman filter is very fast in its convergence.

Kalman filter-based prediction consists of matrix to vector multiplications and computation of vector products. The dimensions of the matrix and vectors involved in the prediction step majorly govern the complexity of the computation. In our prediction scheme, the computational complexity can improve significantly exploiting the structure of the state model. There are three major components in Kalman filter-based prediction: process matrix, state vector, and measurement matrix. In our prediction scheme, the dimension of the state vector is governed by the difference orders ($d$ and $D$) and the lengths of the seasonal legs ($s$ and $S$). The state vector is the minimum amount of information required for prediction, and in our prediction scheme, the dimension of state vector corresponds to $O(2(sd + SD) + 1)$. For $GOPARIMA(1, 1, 1)_3 \times (1, 1, 1)_{15}$, dimension of state vector corresponds to 37. Theoretically, the dimension of the process matrix corresponds to $(2(sd + SD) + 1)^2$, which is a practically not feasible. In our scheme, process matrix is responsible for shifting the elements in $Z_t$ and $X_t$ and computing the new values. Therefore, the process matrix can be virtually represented with coefficients of autoregressive process for $Z_t$ and GOP ARIMA process for $X_t$. Further, coefficients for $Z_t$ are a subset of coefficients for $X_t$. Therefore, we require only $O(p + P + q + Q)$ space for the process matrix. In practice, each of $p$, $P$, $q$, and $Q$ is rarely greater than 1.

Let us examine the computational aspect of the prediction. The most complex computation in our prediction scheme is the computation of the process equation, $F_{t+1} = A_t F_t + W_t$. This computation requires $O((sd + SD)^2)$ computational steps, theoretically. As described in Section IV, our process equation is responsible for shifting the position of the elements in $X_t$ and $Z_t$ and computing the new values. Shifting the position of the elements does not require any computation if we use proper data.

<table>
<thead>
<tr>
<th>Video</th>
<th>Scene Changes</th>
<th>Average (min : sec)</th>
<th>Variance (sec²)</th>
<th>Max (min : sec)</th>
<th>Min (min : sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drama</td>
<td>7</td>
<td>0.49</td>
<td>1458.4</td>
<td>1.50</td>
<td>0.09</td>
</tr>
<tr>
<td>News</td>
<td>43</td>
<td>0.07</td>
<td>24.6</td>
<td>0.21</td>
<td>0.01</td>
</tr>
<tr>
<td>Sports</td>
<td>32</td>
<td>0.09</td>
<td>71</td>
<td>0.41</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Fig. 4. Scene length distribution of empirical video traces (5 min each). (a) Drama. (b) News. (c) Sports.
structure for the state vector, e.g., circular array. Therefore, the computational complexity of the process equation is bounded by the complexity of computing a new value in $X_t$ according to (15). In GOP ARIMA modeling, $(q + 1)(Q + 1)$ terms are required to represent the moving average and the $(p + 1)(P + 1) - 1$ nonzero terms to express the autoregressive part of the time series. The computation of the new value, i.e., predicting the new value in the next state can be done in $O(pP + qQ)$. The rest of the Kalman filter procedure, e.g., computing Kalman gain, error adjustment, are less computationally intensive.

We close this discussion by briefly presenting the results from the physical experiment. We modeled 22 video sequences with different GOP patterns, different frame rates and different motion dynamics. In 18 of 22 cases, $p = 0$ and $P = 0$. The moving average order never goes beyond 1. Refer to Table II and Table IV for the actual the GOP ARIMA models. In 21 of the 22 GOP ARIMA models, the prediction of the new frame size involves less than four nonzero terms. Despite the dimensions of the process matrix, the actual computational overhead of the Kalman filter-based prediction is very small and does not detract from its practical usage.

VI. UPDATING PREDICTION MODEL

A. Traffic Model and Structural Change

One of the fundamental assumptions in Kalman filter-based prediction is that the system state evolves according to the process equation. Because the process equation embodies the stochastic model for the underlying sequence, Kalman filter-based prediction (or any model-based prediction) may not be able to handle the structural changes in underlying sequence even though it has an effective recursive error adjustment mechanism. We argue that a good prediction scheme should be able to determine the validity of the stochastic model upon which the prediction is made. Further, we argue that such determination should be fully integrated into the prediction framework. We adopt confidence interval analysis to achieve this objective.

Due to the characteristics of the state of the art video compression schemes, the stochastic characteristics of the frame size sequence are closely dependent upon the nature of the scene [10]. Scenes with highly dynamic motions, e.g., sports video, and scene with static nature, e.g., news video, may exhibit very different correlation structures in their respective frame size sequences. We first examine the primitive statistics of “scenes” in video clips. The length of a scene ranges from several seconds to several minutes [70]. We visually inspect three video clips, Drama, News, and Sports, each of which is approximately 6 min. long. Table I summarizes the statistics of each clip, and Fig. 4 shows the scene length distribution of each empirical video trace.

News video clips have the most frequent scene changes with an average scene length of 7 s. Drama video clip has the least frequent scene changes with an average scene length of 49 s. News usually has frequent scene changes. In drama sequence, scene changes occurs less frequently. If we properly exploit the categorical information of video contents, we can make scene change detection much more accurate.

B. Confidence Interval Analysis

We perform a confidence interval analysis to determine the validity of our prediction model. When the actual frame size lies within a certain distance from the predicted value, we regard that the prediction model is valid. Let us briefly re-visit the notion of confidence interval analysis. When it is not possible to examine all the elements in the set, which is actually true in most cases, we examine a fraction of the original set and estimate the statistical characteristics of the original set. A typical example is estimating the mean and/or variance of the original set. The confidence interval is denoted with probability $p$. Let $\bar{X}_n$ and $n$ be the sample mean and sample size, respectively.

Let us assume that the standard deviation of the original population, $\sigma$, is known. Then, we can estimate the mean, $\mu$, of the original population

$$\bar{X}_n - z \frac{\sigma}{\sqrt{n}} \leq \mu \leq \bar{X}_n + z \frac{\sigma}{\sqrt{n}}.$$  

(21)

The $z$ in (21) determines the width of the confidence interval relative to probability $p$. In practice, the values of $z$, 1.645, 1.96 and 2.58 are often used for the 90%, 95% and 99% confidence levels, respectively. Although populations do not follow Normal distribution, $P(-z \leq \xi_n \leq z)$ corresponds to 0.90, 0.95 and 0.99 when $\xi_n = \sqrt{\frac{n}{2}}(\bar{X}_n - \mu)/\sigma$ and $z$ is 1.645, 1.96 and 2.58, respectively, according to the central limit theorem. For more details on Confidence Interval Analysis, see [71]. One of the important requirements of our confidence interval analysis is on-line detection capability. The detection method should be rigorously and efficiently integrated into the prediction scheme. To address this issue, we use confidence interval analysis in our prediction scheme. The key ingredient for integration is to represent the confidence interval of the prediction in terms of Kalman filter components. The confidence interval analysis not only detects scene change but also provides a rigorous basis on detection accuracy. A confidence interval-based approach manifests itself when we do not have any prior knowledge on the future frame size sequence, e.g., live video feed.

If we have a scene change at time $t + 1$ with its statistical characteristics being very different from those of the scene.
up to time $t$, there is a good possibility that the frame size observed at $t + 1$ does not lie within the confidence interval obtained at $t$. Strictly speaking, we are interested in a conditional confidence interval because our confidence interval is strongly based on the frame sizes of the past. To establish a confidence interval for the $(t + 1)^{th}$ frame prediction, we establish the sample mean and sample variance for the $(t + 1)^{th}$ frame prediction. Recall that frame size in our context corresponds to measurement, $X_t$. Let the sample mean and sample variance be $\bar{E}(X_{t+1} | X_t, X_{t-1}, \cdots)$ and $\text{var}(X_{t+1} | X_t, X_{t-1}, \cdots)$, respectively. We represent these values using the components in our Kalman filter model. From the definition of a Kalman filter, $\bar{E}(X_{t+1} | X_t, X_{t-1}, \cdots)$ is equivalent to $\hat{X}_{t+1}^\sim$. $\hat{X}_{t+1}^\sim$ is obtained by applying measurement matrix $H_t$ to the prior estimate of state $\hat{F}_{t+1}$.

$$\hat{X}_{t+1}^\sim \equiv \bar{E}(X_{t+1} | X_t, X_{t-1}, \cdots) = H_t \hat{F}_{t+1}.$$  \hspace{1cm} (22)

From Theorem 1, variance of one step prediction $\text{var}(X_{t+1} | X_t, X_{t-1}, \cdots)$ is defined as

$$\text{var}(X_{t+1} | X_t, X_{t-1}, \cdots) = H_t (A_t P_t A_t^T + Q_t) H_t^T + R_t.$$  \hspace{1cm} (23)

**Theorem 1:** Let $A_t$ and $H_t$ be the process and measurement matrix in the Kalman filter for the GOP ARIMA model. Let $P_t$, $Q_t$, and $R_t$ be the error covariance matrix of the state-error vector $E_{\theta}$, error covariance in the process equation, and error covariance in the measurement equation, respectively. Then, the variance of one step prediction of $(t + 1)^{th}$ frame size corresponds to $H_t (A_t P_t A_t^T + Q_t) H_t^T + R_t$.

**Proof:** From the definition of variance and given the fact that $X_{t+1}$ and $\hat{X}_{t+1}$ are scalar values, we can establish the following relationship:

$$\text{var}(X_{t+1} | X_t, X_{t-1}, \cdots) = \bar{E} \left[ (X_{t+1} - \hat{X}_{t+1}^\sim) (X_{t+1} - \hat{X}_{t+1}^\sim)^T \right].$$  \hspace{1cm} (24)

In the Kalman filter model, we assume that $V_{t+1}$ is uncorrelated with $F_{t+1}$ and therefore $\bar{E}[W_{t+1}(F_{t+1} - \hat{F}_{t+1})^T H_t^T] = 0$. Furthermore, since $\hat{F}_{t+1}^\sim$ is a linear function of the set of the states and the measurements observed through time $t$, $\hat{F}_{t+1}^\sim$ must be uncorrelated with $V_{t+1}$ by the fact that $\bar{E}(V_{t+1} F_{t+1}^T) = E[V_{t+1} (H_t F_t + V_t)^T] = 0$. For more details on confidence interval analysis, see Hamilton [65]. From this property and measurement equation of the Kalman filter, we can establish the following relationship:

$$\bar{E} \left[ (X_{t+1} - \hat{X}_{t+1}^\sim) (X_{t+1} - \hat{X}_{t+1}^\sim)^T \right] = \bar{E} \left[ (H_t F_{t+1} + V_{t+1} - H_t \hat{F}_{t+1}^\sim) \times (H_t F_{t+1} + V_{t+1} - H_t \hat{F}_{t+1}^\sim)^T \right]$$

$$= \bar{E} \left[ H_t \left( F_{t+1} - \hat{F}_{t+1}^\sim \right) \left( F_{t+1} - \hat{F}_{t+1}^\sim \right)^T H_t^T \right]$$

$$+ \bar{E} \left[ V_{t+1} V_{t+1}^T \right].$$

Then, from the definitions of $P_{t+1}$ and $R_t$, and the covariance matrix of the Kalman filter, we can obtain the following:

$$\bar{E} \left[ H_t \left( F_{t+1} - \hat{F}_{t+1}^\sim \right) \left( F_{t+1} - \hat{F}_{t+1}^\sim \right)^T H_t^T \right] + \bar{E} \left[ V_{t+1} V_{t+1}^T \right] = H_t P_{t+1} H_t^T + R_t$$

$$= H_t (A_t P_t A_t^T + Q_t) H_t^T + R_t.$$  \hspace{1cm} (25)

Hence the claim. Q.E.D.
for more than or equal to $\psi$ times. If the scene changes, we believe that the changes in the stochastic characteristics should be immediately visible or within a few numbers of GOP’s at the latest. Thus, we set the detection window size $\tau$ to the length of the GOP. There are a number of practical issues in determining $\tau$ and $\psi$. We will address these issues in the experiment section (Section VIII-C).

VII. SAMPLING WINDOW FOR GOP ARIMA MODELING

We can build a more accurate model as we examine a larger number of samples. However, in practice, we need to balance model accuracy and the overhead of building the prediction model. More importantly in our context, scenes change dynamically and therefore we cannot wait long to collect frame size samples. We define the number of samples used to build the model as the sampling window. We examine the sampling window size and the accuracy of the model. This experiment helps us to determine the proper sampling window size. We use frame size sequences from a Drama video clip [Fig. 13(a)]. Table II summarizes the results. Fig. 5 shows the prediction error based on each GOP ARIMA model in Table II. Each graph corresponds to 30, 60, and 90 step prediction. The $x$ axis corresponds to the sampling window size, and the $y$ axis denotes normalized mean square error (NMSE). We found that sampling window sizes of 90 and 120 frames yield the lowest prediction error.

VIII. EXPERIMENTS

A. Environment

We perform comprehensive experiments to examine various aspect of the proposed prediction scheme. In this experiment, we use total of twelve frame size sequences. We use two different compression methods: MPEG-2 and MPEG-4. We use three different GOP structures: GOP(15,3), GOP(12,3), and GOP(9,3). This comprehensive test enables us to verify the effectiveness of the proposed prediction scheme under various settings. There are three MPEG2-GOP(15,3) sequences, three MPEG4-GOP(15,3), three MPEG2-GOP(12,3), and three MPEG2-GOP(9,3). MPEG-2 traces are in-house generated, with 4 Mbits/s playback rate (DVD quality). Bandwidths of MPEG-4 traces ranges from 250 to 600 Kbits/s. Under these two compression scheme, we can evaluate the effectiveness of prediction methods under HD quality video streaming as well as video streaming in a mobile wireless environment.

For MPEG-2 video traces, we carefully select video clips with different motion dynamics and scene change characteristics: News, Drama, and Sports (Basketball). These traces are publicly available at [12]. They have GOP(15,3) structure with 30 frames/sec frame rate, i.e., total GOP size is 15 frames and P frame appears in every 3 frames. MPEG-4 video traces used in the study are obtained from public site [11]. We use three well-known and widely used video traces: Bean (580 Kbits/s), Jurassic Park (770 Kbits/s), and Star Wars (280 Kbits/s). Summaries of the frame size sequences are presented in Table III and Table IV, and Fig. 6 shows a snapshot of drama, news, and sports video traces.

Our experiments examine five aspects of the bandwidth prediction: (i) accuracy of prediction schemes; (ii) accuracy of confidence interval-based scene change detection; (iii) effect of scene change detection and prediction error; (iv) effect of Model update on prediction accuracy; and (v) effect of bandwidth prediction over application level QoS. We consider three prediction schemes to compare the test results against: Double Exponential Smoothing [68], ALP [14], and Prediction Scheme by Adas et al. [15]. Our predictor uses the first 90 samples (frame sizes) to construct the GOP ARIMA model for prediction. Using Kalman filter, our prediction model continuously predicts and recursively updates the model parameters. If the difference between the predicted and actual frame size satisfies our model update condition, it builds new model.
<table>
<thead>
<tr>
<th>Name of the Movie</th>
<th>Mr. Bean</th>
<th>Jurassic Park I</th>
<th>Star Wars IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stream Type</td>
<td>Mpeg-r</td>
<td>Mpeg-1</td>
<td>Mpeg-1</td>
</tr>
<tr>
<td>Number of Frames</td>
<td>89113</td>
<td>89998</td>
<td>89998</td>
</tr>
<tr>
<td>Frame Rate</td>
<td>25 frames/sec</td>
<td>25 frames/sec</td>
<td>25 frames/sec</td>
</tr>
<tr>
<td>Mean Frame Size (byte)</td>
<td>2909</td>
<td>3831</td>
<td>1376</td>
</tr>
<tr>
<td>Min/Max Frame Size (byte)</td>
<td>93/15251</td>
<td>72/16745</td>
<td>26/9370</td>
</tr>
<tr>
<td>Number of GOP</td>
<td>7426</td>
<td>7499</td>
<td>7499</td>
</tr>
<tr>
<td>Mean GOP Size (byte)</td>
<td>35000</td>
<td>46300</td>
<td>17000</td>
</tr>
<tr>
<td>Min/Max GOP Size (byte)</td>
<td>2294/130332</td>
<td>2105/144732</td>
<td>900/70850</td>
</tr>
</tbody>
</table>

Table IV: Parameters for MPEG-4 Video Traces

Fig. 7. Prediction step versus prediction error for GOP ARIMA, double exponential smoothing and ALP, GOP(15,3), 30 frames/s, 4 Mbits/s. (a) Drama. (b) News. (c) Sports.

Fig. 8. Cumulative distribution of video traces.

There are a number of metrics for quantifying prediction accuracy. We use both signal-to-noise ratio (SNR)\(^1\) and normalized mean square error (NMSE).\(^2\) These two metrics have slightly different characteristics. SNR metric is used in many areas, especially in signal processing. It captures the ratio of the total error to total frame size, which is a good quick method for determining predictor’s performance. References [14], [15], and [35] used SNR\(^{-1}\) as a performance metric for prediction accuracy. NMSE consists of two parts, mean square error (MSE) and normalize part. The MSE quantifies error in time domain and the deviation measures performance in space domain.

### B. Prediction Accuracy: MPEG-2

We compare the prediction accuracy of the three prediction schemes: GOP ARIMA-based prediction, ALP [14] and Double Exponential Smoothing-based Prediction (DESP) [68]. In this experiment, we consider prediction within a scene having 60, 90, and 120 prediction steps. We use MPEG-2 traces with GOP(15,5) in Table III. Results of only one GOP structure can obscure view on other perspectives on different GOP structures. In order to overcome the limitation, GOP(9,3) and GOP(12,3) are also used in our experiment to generalize the performance of GOP ARIMA.

In ALP, we use three linear predictors for each frame type, I, B, and P. The step size \(\mu\) in (6) is selected to be 0.01. Using a small \(\mu\) results in slow convergence and less fluctuation after convergence. Using this value, the least-mean square (LMS) will converge on the mean [15]. We select the order \(p\) with \(6 \leq p \leq 12\) with Akaike Information Criterion (AIC) [64]. For GOP ARIMA-based prediction, we use (26) as GOP ARIMA models for the three video clips with a sampling window size of 90.

\[
\text{[Drama]} \quad (1 - B^{15})X_t = (1 - 0.63B^{15})\varepsilon_t, \quad \sigma^2 = (2030.6)^2 \\
\text{[News]} \quad (1 - B^{15})X_t = (1 - 0.41B^{15})\varepsilon_t, \quad \sigma^2 = (2314.3)^2 \\
\text{[Sports]} \quad (1 - 0.514B^3 - 0.405B^9)(1 - B^3)(1 - B^{15})X_t = (1 - 0.98B^{15})\varepsilon_t, \quad \sigma^2 = (1795.5)^2.
\]

(26)

Fig. 7 quantifies the prediction error for the three prediction schemes, and illustrates NMSE under varying prediction steps. In all three prediction schemes, prediction error tends to increase with the prediction step. Recall that we are using a 30 frames/s frame sequence. 90 step prediction, for example, estimates the frame size in 3 s interval. As can be seen in Fig. 7, GOP ARIMA-based prediction exhibits superior accuracy compared to DESP and ALP. The relative difference in prediction accuracy becomes larger as the number of prediction steps increases.

In addition, prediction error is much larger in the Sports clip than in the other two video clips. We suspect that this is partly
due to the scene length distribution of the Sports clip and the dynamic nature of the video scene. Table V illustrates the scene length statistics. Drama has the longest scene length with an average of 49 s and median of 34 s. For News, the average and median value of scene length are 7 and 5 s, respectively. For the Sports video clip, the average and median scene length are 9 and 7 s, respectively. Fig. 8 illustrates the CDF of scene length distributions of three video clips. We compare the prediction accuracy of GOP ARIMA, Adas, and Yoo’s methods for GOP(12,3) and GOP(9,3) video traces. Table VI illustrates GOP ARIMA model of three video traces for 60 step prediction. Fig. 9 illustrates the results of our experiment. We examine NMSE of 30, 60, and 90 steps prediction under two different GOP structures. Fig. 9(a) and (b) illustrates the experiment results for GOP(9,3) and GOP(12,3), respectively. In both of GOP structures, prediction with GOP ARIMA with Kalman filter and Yoo yields lower NMSE score than the prediction based upon Adas’s scheme. In
News and Sports video clips, GOP ARIMA-based prediction yields similar accuracy to Yoo’s scheme.

We examine the prediction accuracy of the proposed model under publicly available MPEG-4 traces (Jurassic Park, Mr. Bean, Star Wars [11]) with varying prediction steps. In this prediction, Kalman filter dynamically updates the model based upon its scene change detection mechanism. We use predictors developed by Adas [15] and Yoo [14] for comparison. Fig. 10 illustrates the results in normalized mean square error (NMSE) and SNR. The proposed scheme yields more accurate prediction results to the other two models for both metrics.

C. Detection of Structural Change: Confidence Interval Analysis

We propose to use confidence interval analysis to determine the validity of the prediction model. We illustrate how the confidence interval analysis is used to determine the model validity and present the actual test result. Fig. 11 illustrates the frame
Fig. 13. Scene changes and prediction error: GOP(15,3); 4 Mbits/s; 30 frames/s. (a) Drama: Actual Scene Changes. (b) News: Actual Scene Changes. (c) Sports: Actual Scene Changes. (d) Drama: Prediction Error. (e) News: Prediction Error. (f) Sports: Prediction Error.

size sequence of the original traffic and the result of one-step prediction. The solid line represents the frame size sequence of the original traffic. The predicted values are augmented with the confidence intervals at confidence levels of 90%, 95%, and 99%, respectively. The confidence level represents the probability that the “real” value lies within a given predicted range. The range becomes larger as confidence level is increased. We call the range as confidence interval. Refer to (26) for more details of confidence interval.

Fig. 11 illustrates the original frame size and predicted frame size with different levels of confidence: 90%, 95%, and 99%, respectively. We can see that some frame sizes lie within the prediction range and some frame sizes are completely outside of all three prediction ranges. For example, the size of 7th and the 10th frames lie outside all confidence intervals. Note that these are P type frames. The size of 13th frame (I type) also lies outside all three confidence intervals. Let us consider 19th frame. Its size lies outside the 90% confidence level but within the 99% confidence level. In this case, validity of the prediction is subject to the choice of confidence level.

We visually examine the video clip and check whether a proposed method accurately detects the scene change. We use a News video clip (30 frames/s, GOP(15,3), and 4 Mbits/s) in this test. We vary the confidence level (90%, 95%, and 99%) and the detection threshold values, $\psi = 3, 5,$ and 7. The detection threshold is the number of mispredictions within a given time interval. If the number of mispredictions is greater than or equal to $\psi$ during a certain time window, we determine that scene has changed and that the current prediction model is invalid. Fig. 12 illustrates the results of scene change detection under various settings.

The notion of “scene change” is a rather context sensitive term. In content-based video analysis, objectives of scene change detection are annotation, indexing, summarization and etc. On the other hand, our objective of scene change detection is to more accurate frame size prediction. Therefore, we are not concerned about scene change from the video content’s point of view if the stochastic characteristics of the underlying sequence does not change. In this section, we perform sensitivity analysis of scene change detection parameters on detection accuracy. To effectively address this issue, we define two types of scene change: technical and semantic scene change. A technical scene change includes all types of minor changes, e.g., change in background picture, change of camera angle and etc. A semantic scene change is a change to a different plot. There are approximately 70 technical scene changes and approximately 40 semantic scene changes. We study the accuracy of scene change detection under varying scene change detection parameters from a technical and a semantic change’s point of view. The x axis in the graphs of Fig. 12 has three columns: True (Hit), False Negative (Mis) and False Positive (Mistake).

Since the B type frame is encoded with bidirectional interpolation, its size not only is small but also it does not vary much. The size of the I-frame does not change much within a scene. The size of the P type frame exhibits rather different characteristics. It contains the difference with its nearest preceding P or I frame. The size of the P frame varies more compared to the other frame types. In our experiment, one or two P type frames in a GOP lie outside the confidence interval (on the average) even though scene did not change. Therefore, we recommend making the detection threshold greater than two. As we increase $\psi$, the case of false negative increases.

For the detection of technical scene change, a 90% confidence level with $\psi = 3$ yields the best performance. To detect the semantic scene change, a 99% confidence level with $\psi = 3$ yields...
TABLE VII
GOP ARIMA MODEL FOR EACH SCENE IN DRAMA, NEWS AND SPORTS VIDEO CLIPS

<table>
<thead>
<tr>
<th>Video</th>
<th>Scene No.</th>
<th>Mean(Byte)</th>
<th>Variance(Byte^2)</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drama</td>
<td>1</td>
<td>15942</td>
<td>73789957</td>
<td>(0,1,1)_3 x (0,1,1)_5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>20013</td>
<td>101219155</td>
<td>(2,1,0)_3 x (0,1,1)_5</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>19938</td>
<td>102360999</td>
<td>(0,1,1)_3 x (0,1,1)_5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>15082</td>
<td>72029727</td>
<td>(0,1,1)_3 x (0,1,0)_5</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>16910</td>
<td>115799842</td>
<td>(0,1,1)_3 x (0,1,0)_5</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>15593</td>
<td>83766902</td>
<td>(0,1,1)_3 x (0,1,0)_5</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>18189</td>
<td>126855131</td>
<td>(0,1,0)_3 x (0,1,0)_5</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>16490</td>
<td>82708781</td>
<td>(0,0,0)_3 x (0,1,1)_5</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>15173</td>
<td>90282004</td>
<td>(0,1,1)_3 x (0,1,0)_5</td>
</tr>
<tr>
<td>News</td>
<td>1</td>
<td>16895</td>
<td>32120504</td>
<td>(1,1,0)_3 x (0,1,1)_5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>16411</td>
<td>62367342</td>
<td>(2,1,0)_3 x (0,1,1)_5</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>16137</td>
<td>35629539</td>
<td>(1,1,0)_3 x (0,1,1)_5</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>16934</td>
<td>58407209</td>
<td>(0,1,1)_3 x (0,1,1)_5</td>
</tr>
<tr>
<td>Sports</td>
<td>1</td>
<td>16895</td>
<td>32120504</td>
<td>(1,1,0)_3 x (0,1,1)_5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>16411</td>
<td>62367342</td>
<td>(2,1,0)_3 x (0,1,1)_5</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>16137</td>
<td>35629539</td>
<td>(1,1,0)_3 x (0,1,1)_5</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>16934</td>
<td>58407209</td>
<td>(0,1,1)_3 x (0,1,1)_5</td>
</tr>
</tbody>
</table>

Fig. 14. Model update and prediction accuracy. (a) Jurassic Park. (b) Mr. Bean. (c) Star Wars.

TABLE VIII
SCENE CHosen IN MR. BEAN, JURASSIC PARK AND STAR WARS VIDEO CLIPS

<table>
<thead>
<tr>
<th>Video</th>
<th>Scene No.</th>
<th>Mean(Byte)</th>
<th>Variance(Byte^2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mr. Bean</td>
<td>2</td>
<td>2815623</td>
<td>(1587.23)^2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>2815623</td>
<td>(1587.23)^2</td>
</tr>
<tr>
<td>Jurassic Park</td>
<td>8</td>
<td>3947442</td>
<td>(1637.03)^2</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>1442799</td>
<td>(1729.55)^2</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1131208</td>
<td>(1232.85)^2</td>
</tr>
<tr>
<td>Star Wars</td>
<td>6</td>
<td>3240955</td>
<td>(1070.73)^2</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>926457</td>
<td>(1011.37)^2</td>
</tr>
</tbody>
</table>

D. Scene Change and Prediction Error

In this section, we investigate the effect of scene change over prediction error. Fig. 13(a), (b), and (c) illustrates the scene change points and scene snapshots at the start of each scene. The bold vertical line is used to separate the different scenes. Table VII illustrates the GOP ARIMA model for each scene in Fig. 13(a), (b), and (c), respectively. We use Akaike Information Criterion (AIC) [64] in determining the orders and parameters of the GOP ARIMA model. According to our study, each of the scenes has different stochastic structure, i.e., not only in parameters but also in terms of orders, i.e., autoregressive order, moving average order and difference order.

There are 2, 7, and 4 scenes in Fig. 13(a), (b), and (c), respectively. Table VII shows that the structure of GOP ARIMA model for each scene has different structure, i.e., different orders and different parameters. We investigate the prediction error behavior when we do not update the prediction model throughout the entire playback. The images in Fig. 13 [(d), (e), and (f)] show the mean square error for frame size estimation with a 30-step prediction. Each figure is annotated by the scene change point. As can be seen in Fig. 13, prediction error sharply increases when the scene changes. The results of this study illustrate the importance of scene change detection and model update in order to obtain more accurate frame size prediction. Note that before a new model is obtained, the old model is used to predict the frame size, while the prediction algorithm accumulates frame sizes for observation window. These samples are used to build a new prediction model.

E. Effect of Model Update

We examine the effectiveness of the model update. We compare three prediction methods: (i) GOP ARIMA without model update; (ii) GOP ARIMA with model update; and (iii) ALP [14]. We choose to use ALP [14] in our comparison study because it adopts threshold-based scene change detection mechanism.

Fig. 14 illustrates effectiveness of model update for each of the three prediction methods. x axis and y axis denotes the prediction step and prediction error (NMSE), respectively. In this study, the objective is not only to determine which model best captures the scene change, but also to determine which of the three methods can quickly adapt its prediction to a new scene in order to provide accurate prediction. To effectively address this issue, we define scene change as simply an abrupt
change in the frame size variance. Scene 2 and scene 3 from Mr. Bean are frames from 1068 to 3756. As for Jurassic Park, frames in scene 8, 9, and 10 are from 7453 to 9122. For Star Wars (scene 6 and scene 7), frames from 3409 to 5760 are chosen. Table VIII illustrates the primitive statistics of the scenes. Scenes are chosen such that they allow comprehensive understanding on behavior of the models in change of variance. We observed that in GOP ARIMA-based prediction, model update can improve the prediction accuracy by more than a factor of two especially when there exists sharp changes in the underlying frame size sequence. When the structure of the time series changes, past samples do not give sufficient information to forecast the behavior of the time series. Therefore, updating the model with respect to the characteristics of the underlying time series yields more accurate prediction results.

F. Effect on User Perceivable QoS

The ultimate objective of bandwidth prediction is to improve resource utilization or to deliver better QoS video streaming service to end user. We perform simulation study to examine the effect of prediction scheme over user perceivable QoS. Quantifying the user perceivable QoS is by itself profound subject and we do not delve into details on user perceivable QoS modeling. As a resort to quantify the effectiveness of various prediction schemes, we measure packet loss probability and buffer utilization when queue (or buffer) is allocated based upon a given prediction scheme. For each prediction scheme, queue length is dynamically adjusted, and recalculated every 30 frame interval based upon predicted bandwidth.

Fig. 15 shows packet loss probability and buffer utilization and, boxplot for queue size. y axis on the left in Fig. 15(a) shows packet loss probability and y axis on the right shows the buffer utilization. Table IX shows the result from the experiment. Total of 23339 packets are used as input trace to the queue and number of loss count of packets are 307, 498, and 472 for GOP ARIMA, Adas, and Yoo, respectively. When buffer is allocated based upon GOP ARIMA prediction, packet loss was the smallest. Packet loss can be minimized simply by overprovisioning. GOP ARIMA has about 10% higher utilization rate compared to other schemes. We examine Packet Loss Rate/Buffer Utilization to show the packet loss probability over buffer utilization. This value means that how well a given buffer is exploited. If packet loss improves due to overprovisioning, buffer utilization will become worse. Therefore, Packet Loss Rate to Buffer utilization ratio becomes larger, and GOP ARIMA has least score among three. Fig. 15(b) and Table X show result on queue size. According to our experiment, GOP ARIMA uses smaller queue compared to Adas and Yoo. However, GOP ARIMA-based prediction scheme delivers better QoS behavior.

IX. CONCLUSION

In this paper, we develop a novel bandwidth prediction scheme for VBR compressed video with regular GOP pattern. We use GOP ARIMA as the base stochastic model for the underlying time series. We deploy a Kalman filter in GOP ARIMA and for more accurate prediction we update the prediction model based upon a statistical hypothesis test. Our prediction scheme successfully addresses a number of challenging issues. The prediction scheme preserves the correlation structure of the frame size sequence. Our prediction scheme does not require a separate prediction model for individual type frames and therefore makes more accurate predictions. Since Kalman filter-based recursive error adjustment maintains “state” across the prediction rounds, the proposed prediction scheme becomes more robust against noisy input than stateless
prediction schemes. Our prediction model effectively copes with structural changes in the underlying sequence. It performs statistical hypothesis testing and determines the need for model update. Since we represent the confidence interval of a given prediction with Kalman filter components, the hypothesis test can be seamlessly embedded into the prediction model. Confidence interval analysis provides rigorous measure on its detection accuracy. The results of the performance study show that our prediction scheme significantly improves the prediction accuracy and prediction responsiveness compared to existing linear prediction-based methods and neural network-based methods. We also examined the performance of the prediction algorithm from the bandwidth’s perspective. We compare the bandwidth prediction accuracy of three prediction schemes: {GOP} {ARIMA} with Kalman filter, Adas [15], and Yoo [14], using number of publicly available MPEG-2 and MPEG-4-based video traces [11], [12]. We quantify the prediction accuracy using normalized mean square error (NMSSE) and $S N R^{-1}$. According to our experiment, GOP ARIMA-based prediction algorithm makes more accurate prediction. This can significantly improve the QoS to bandwidth ratio. By properly updating the model based upon the confidence interval analysis, we can significantly improve the accuracy of prediction. The Kalman filter–based prediction scheme proposed in this work makes significant contributions to various aspects of network traffic engineering and resource allocation.

REFERENCES

casts/mpeg4trace/trace.html


Sungjoo Kang received the B.S. degree in electronics and electrical engineering from Hanyang University, Korea, in 2003, the M.S. degree in electrical and computer engineering from Hanyang University, Korea, in 2005. Since 2005, he has been with the Electronics and Telecommunications Research Institute. His research interests are in the areas of Multimedia Streaming, Web 2.0, and Software as a Service.

Seongjin Lee received the B.S. and M.S. degrees in electronics and computer engineering from Hanyang University, Korea, in 2006, and 2008, respectively. He is a Ph.D. degree candidate in Electronics and Computer Engineering at Hanyang University, Seoul, Korea. His research interests are in the areas of network traffic modeling and analysis, traffic engineering, and performance measurement and analysis.

Youjip Won received the B.S. and M.S. degrees in computer science from Seoul National University, Seoul, Korea, in 1990 and 1992, respectively. He received the Ph.D. degree in computer science from the University of Minnesota, Minneapolis, in 1997. After receiving the Ph.D. degree, he joined Intel as Server Performance Analyst. Since 1999, he has been with the Department of Electronics and Computer Engineering, Hanyang University, Korea, as an Associate Professor. His research interests are operating system, storage subsystem, multimedia networking, and network traffic modeling and analysis.

Byeongchul Seong received the B.S. and M.S. degrees in computer science and statistics, and the Ph.D. degree in statistics, all from Seoul National University, Seoul, Korea, in 1995, 1997, and 2004, respectively. He was a Postdoctoral Researcher at Washington State University, Pullman, WA, in 2004 and 2005, and a Visiting Assistant Professor with the Department of Mathematics, POSTECH, Korea, in 2006. Since 2007, he has been an Assistant Professor with the Department of Statistics, Chung-Ang University, Korea. His research interest is in the area of time series analysis.