

Minimizing Uplink Data in Wireless Free-Viewpoint Video Transmission Applications

Carl J. Debono, *Senior Member, IEEE* and Clifford De Raffaele, *Student Member, IEEE*

Department of Communications and Computer Engineering,
University of Malta,

Msida, Malta

c.debono@ieeee.org, cderaffaele@ieeee.org

Abstract—The increase in processing power of mobile multimedia terminals, the improvements in camera technology and efficient video compression algorithms have made the implementation of mobile Free-Viewpoint Video (FVV) technology possible. However, the use of this technology in wireless environments poses a number of challenges. Amongst these there are issues in bandwidth utilisation and the limited supply power of the mobile device. To this effect, this paper compares the implementation of two prediction algorithms to forecast the next viewpoint in an attempt to minimise the amount of viewpoint requests uploaded to the server. Simulation results demonstrate that a reduction of up to 72% in uplink transmission traffic is achieved when emulating a conventional usage scenario. This means that the application of prediction schemes can drastically reduce the power consumption and resource requirements of mobile multimedia terminals.

Keywords—Free-viewpoint, Kalman filter, least-mean-square algorithm, wireless transmission.

I. INTRODUCTION

Driven by the technological developments, multiview broadcasting has recently augmented interest from academic and industrial perspectives alike [1-2]. The expectations from the consumer's side have also increased and interactive broadcasting services are gaining popularity [3]. Multiview technology provides the potential of expanding the user's experience far beyond the current conventional media [4]. In Free-Viewpoint Video (FVV) the user can interactively choose a desired viewpoint and observe a three-dimensional panoramic scene from arbitrary perspectives [5].

A unique scene captured from multiple views requires the implementation of a number of cameras which are distributed around the site. In order to adhere to feasibility issues, the number of cameras and the spacing between them must be restricted [6]. This implies that the viewer can only observe a continuous video stream if standpoints, which are not quantized and limited to the actual camera locations, are available through techniques which apply Intermediate View Reconstruction (IVR). Through these solutions, virtual views between cameras are synthesized to provide a gradual change in viewpoint from one camera to another as shown in Fig.1. The intensity of the FVV experience that is directed to the user is directly related to the technique employed to generate these virtual views.

A number of Video-based Rendering (VBR) techniques have been proposed in literature to generate the required IVR. The authors in [4] interpolate the images between views through 3D meshes which require prior knowledge of the scene geometry. A similar approach was adopted in [6], where human actors were rendered after making use of predefined models. A depth-based rendering approach is found in [7], where linear interpolation of the view and colour is applied together with post-warping of the image.

Unfortunately, most of the highlighted techniques are computationally intensive and require a number of sequential steps. Even though these methods produce an adequate visual quality, the sheer amount of data and the sophisticated algorithms make them unsuitable for real-time rendering. The method in [8] addresses this issue, and presents an algorithm which renders the virtual view after analyzing the colour variance from each camera to detect the foreground and the background in the video in a plane-sweep methodology. Computation is however still done on a pixel-by-pixel basis, thus requiring a custom computing architecture to execute the algorithm.

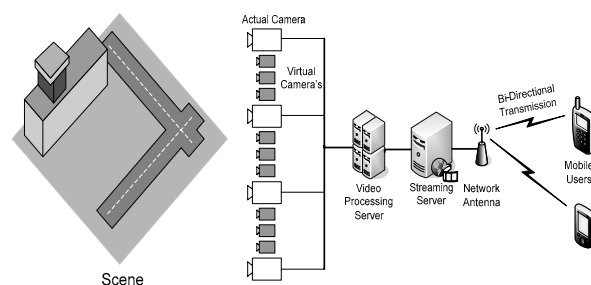


Figure 1. Block diagram of the entire system representing the scene capturing by multiple cameras, video-based rendering servers, the streaming server and the wireless network.

Simultaneous to this progress, emerging mobile terminals are all equipped with colour displays, and enough processing power to allow presentation and decoding of video sequences which can be employed to view FVV streaming. Additionally, emerging and future wireless systems will have a sufficient infrastructure to support the necessary bandwidth for video communication applications [9]. However, bit rates will always be scarce in wireless transmission environments due to the physical

bandwidth spectrum allocation and the transmission power limitations [10-11]. Another main bottleneck for this technology is the processing capabilities of the mobile terminals and their dependence on battery power availability [12].

These implications make the transmission of several camera views for rendering at the mobile terminal an impractical solution in real-time environments. Thus, a sensible implementation of such FVV systems requires the implementation of the VBR on the server, such that only the required view is sent to the mobile terminal. Inherently, this strategy involves the requirement of a feedback channel from the mobile terminal towards the service station to request the particular view required by the user.

This paper presents a novel solution which reduces the amount of feedback transmitted by the mobile terminal during free-viewpoint operation. Prediction strategies are adopted to estimate the next view request, with feedback being sent only when the received view does not match the user's indicated view. The server interprets the lack of feedback as a confirmation that the last estimate was correct. This results in preservation of the mobile terminal's battery power, reduction in bandwidth utilisation and minimisation of round-trip delays incurred by the system due to the otherwise continuous requests on the uplink channel.

This paper is organised as follows; the proposed prediction algorithms are described in Section 2 with discussions on the parameters employed. Section 3 presents the simulation results which highlight the reduction in feedback transmission achieved by each solution. Finally, comments and conclusions are presented in Section 4.

II. VIEW PREDICTION ALGORITHMS

The implementation of the view prediction algorithms involves the utilisation of a set of readings to construct the original view set within some pre-defined accuracy. Prediction is performed by combining knowledge of the system's operation together with either the previously received data or the last predicted data. The latter is used when the error is smaller than the threshold.

A. Least-Mean Square Algorithm

The least-mean-square (LMS) algorithm, proposed in [13], employs an iterative process that updates a weight vector by performing successive corrections which ultimately lead to the minimum square error. The LMS algorithm provides several advantages for implementation in real-time systems particularly due to its low computational complexity, stable and robust performance, as well as an inherent suitability in slow time-varying environments, because of its iterative nature.

The function of the LMS algorithm is defined as [14]:

$$\hat{x}(n) = \sum_{i=1}^N w_i(n) \times x(n-i) \quad (1)$$

where \hat{x} denotes the estimated input signal n derived by the algorithm, w_i represents the current system weight vector, and $x(n-i)$ corresponds to the delayed inputs.

In order to calculate the prediction error, $e(n)$, the estimated value is subtracted from the real input value $x(n)$:

$$e(n) = x(n) - \hat{x}(n) \quad (2)$$

The system weight vector is then modified using:

$$w(n+1) = w(n) + \mu \alpha(n) e(n) \quad (3)$$

The adaptation of the algorithm depends on the step size μ , which influences the speed of convergence, and the order of the algorithm N . The computational load of each iteration can be summarized as $2N+1$ multiplications and $2N$ additions. A filter length N of 5 was found to provide a good compromise, keeping the execution time and memory footprint low whilst still preserving accuracy. The values for μ were determined through heuristic techniques. This parameter was adapted during simulation to speed up convergence at the start, and then changed to reduce overshoot and reach the correct weight values.

The operation of the proposed system involves the implementation of the LMS algorithm at the mobile terminal, where the user's desired view is estimated from the preceding readings and the adapted LMS coefficients. The error between the views is then calculated using (2). Transmission from the mobile terminal, requesting an update to the server, is only performed if this error exceeds a pre-defined threshold. The LMS algorithm on the mobile terminal automatically interprets this as an alteration to the input pattern demanded by the user and triggers updating of the LMS weights. Although the system will quickly readapt its weights to the new pattern via the learning rate parameter μ , the inconsistencies of the weight vector during the conversion allows for a suitable error signal to be detected by a comparator.

On the server node, an identical LMS algorithm is employed to generate equivalent estimates. Thus, the server is capable of tracking the predictions done at the mobile terminal's side without the requirement of constant transmission requests to update the current state. The server assumes that the estimate it has calculated is correct if no transmission is received from the mobile terminal. In case feedback is received, the reported value is analysed and the algorithm's weight vector is retrained to the new pattern through twenty training reference values obtained from the mobile terminal. Through this behavior, synchronisation of the dual LMS architecture is ensured with respect to the outlook being viewed by the user.

B. Kalman Filter Algorithm

The Kalman Filter proposed in [15] employs a recursive algorithm, with a low computational cost, to address the estimation problem for linearly evolving systems. Apart from its practical demands being suitable for real-time applications, a convergence to a stable steady state is guaranteed by the filter [16].

The Kalman Filter predicts the forward state by multiplying the current system state x_k with the state transition matrix A :

$$\hat{x}_{k+1} = A \times x_k \quad (4)$$

The filter also forecasts the predicted error covariance and uses this value together with the observation matrix H to compute the Kalman gain K . A measurement of the system output z_k is then acquired and is used to update the estimate value x_{k+1} using:

$$x_{k+1} = \hat{x}_{k+1} + K(z_k - H\hat{x}_{k+1}) \quad (5)$$

The obtained positional measurement of the scene is compared with the previous value, and since the measurements are done within constant time intervals, the view change request rate can be calculated. This new value is used to update the input vector x_{k+1} . Finally, the error covariance is updated for the next iteration.

The Kalman Filter is implemented on the server to predict the desired view sequences requested by the user. A training set composed of fifteen view samples, provided by the mobile terminal, is used by the filter to converge to a linear system. Once the system is switched offline, the server will transmit the VBR view predicted by the Kalman algorithm. A simple comparator on the mobile terminal checks whether the received view is actually the one requested by the user. The server node assumes that the view transmitted was correct if no feedback is received from the mobile terminal, hence prediction continues along the predicted response. When the mobile system notes a discrepancy between the received view and user's demanded one, a feedback packet containing the requested view is transmitted. On reception of this packet, the Kalman filter is turned online again and starts a retraining routine to adapt itself to the new linear system, where the entire process iterates again.

III. SIMULATION AND RESULTS

To simulate and analyse the employment of a FVV scenario objectively, a number of different usage scenarios were modeled in Matlab[®]. The simulations represent typical usage profiles for free-viewpoint video utilisation, where users actively seek to change their viewpoint throughout the simulated 350 step timeframe. The system considered involves scenes captured from a 12 camera setup, whilst allowing the additional video rendering of nine distinct virtual views between the fixed cameras' perspective. The variable time FVV patterns simulated allow for different view changing speeds to be implemented, thus emulating a more flexible and realistic scenario.

To evaluate the performance of the proposed algorithms, each simulation is compared to the current operational system specified by the authors in [12]. This reference methodology, displayed in part (a) of Figs. 2 - 4, involves the mobile terminal sending feedback to the server in every time step, requesting the latter to render and transmit the view selected by the user. Therefore, the mobile terminal needs to transmit at a frequency equal to the frame rate of the video sequence. Assuming that there are no transmission errors, this implies that the reference profile will precisely follow the requests.

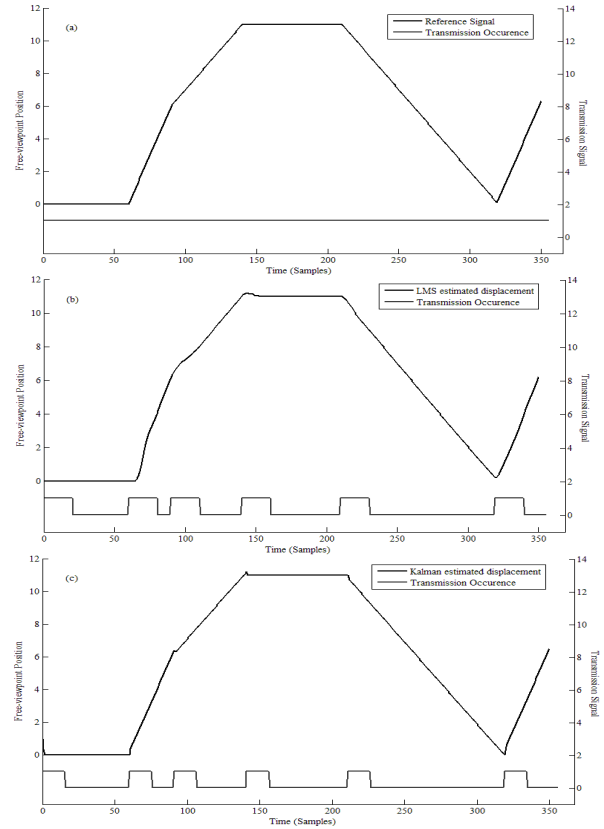


Figure 2. Simulation results comparing the reference system to the proposed algorithms in a typical low-usage FVV scenario; (a) reference, (b) Least-Mean Squares, and (c) the Kalman Filter.

The simulation of the LMS algorithm is presented in part (b) of Figs. 2 - 4. The inherent slow convergence speed of the algorithm is noted in the simulation results. Even though the learning rate parameter is adaptively increased during the initial stage of training, to speedup convergence, a certain amount of time still has to elapse before the algorithm's internal memory buffers are occupied and the weight vectors updated accordingly. Hence, the LMS algorithm needs the delay of an initial settling time before it can be used successfully in FVV applications.

The transmission occurrence signal indicates that when the input demand of the view pattern is changed by the user, the converged weights are no longer valid, hence yielding a maximum error signal at the mobile station which re-initiates training. The amplitude of this error is proportional to the learning rate parameter's value. Unfortunately, the value of this parameter also effects directly the time for convergence, hence a balance has to be found such that a noticeable error signal is produced while still stabilising the system in a short time. Nonetheless, the compromise made on μ hinders the maximum frequency at which the system can alter the viewing pattern and the maximum change in gradient the system is able to adhere to.

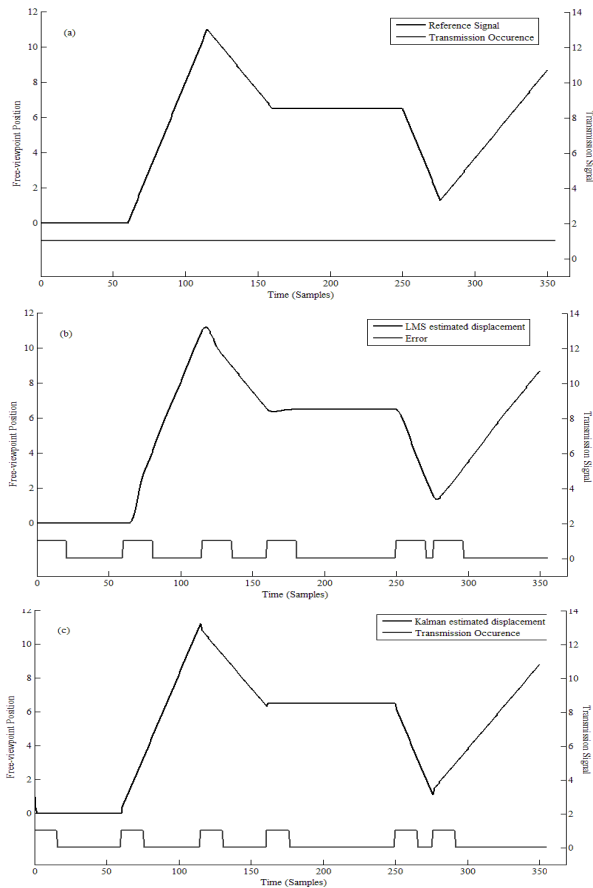


Figure 3. Simulation results comparing the reference system to the proposed algorithms in a typical medium-usage FVV scenario; (a) reference, (b) Least-Mean Squares, and (c) the Kalman Filter.

The LMS algorithm provides a considerable improvement on the standard reference system. With a minimal increase in computation time and resources on both the mobile terminal and the server, an average reduction of 63% in feedback transmissions is obtained. This reduces the traffic on the uplink channel from 350 transmissions to approximately 130 transmissions in the simulated scenarios.

The Kalman Filter results, illustrated in part (c) of Figs. 2 - 4, implicitly expose the benefits derived from the powerful adaptation of this algorithm. Compared to the LMS method, the initial convergence speed is greatly enhanced, drastically reducing the necessary setup time. The computational complexity of the system is rewarded with a much sharper error signal and the requisite of fewer input samples to stably converge to the new linear system once training commences. The frequency of the system is also significantly improved, and therefore it can tolerate a much higher rate of change of view pattern in a stable manner.

The Kalman filter, although requiring a more complex computational cost at the server, relieves the execution cost from the mobile terminal, which is now only required to perform a comparison between the view number being received to the view number demanded by the viewer. The prediction properties of the latter methodology are also

more robust requiring only fifteen view training samples to adapt to the new pattern with a significantly reduced error such that offline operation is sustained. This enables the system to be more flexible to dynamic view changes, making it a more suitable solution.

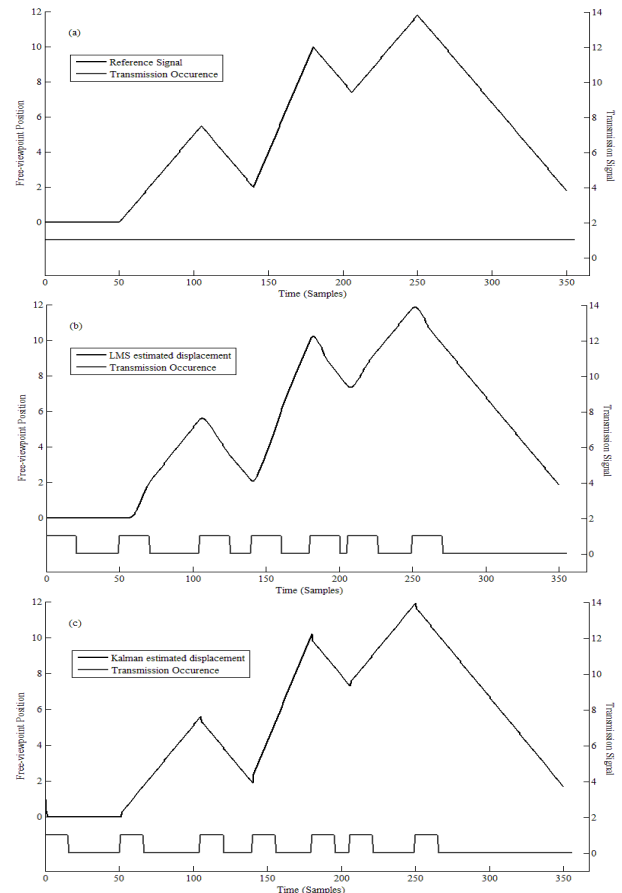


Figure 4. Simulation results comparing the reference system to the proposed algorithms in a typical high-usage FVV scenario; (a) reference, (b) Least-Mean Squares, and (c) the Kalman Filter.

The Kalman filter solution results in an average reduction of 72% of the necessary feedback transmissions on the uplink channel. This significantly reduces the battery power requirements for the implementation of the FVV technology, as the transmitter is the most power hungry component of a mobile terminal. A simulation of FVV implementation with CIF resolution on a PDA with a 1500mAh battery was performed on an HSPA network adopting 1.4Mbps and 7.2Mbps on the uplink and downlink respectively. Results showed that under these conditions, the battery availability increases from 72 hours to 91 hours if FVV transmission only is considered.

Both prediction algorithms achieve a reduction in the system bandwidth utilisation, posing an attractive prospect to service providers and justifying the small added computational cost at the server. An enhancement is also registered in the Quality of Service (QoS) of the FVV technology, since the pro-active system is capable of providing the required view instantaneously to the user, thus reducing the round-trip delay incurred in the reference method.

IV. CONCLUSION

This paper has presented a novel application of prediction algorithms to free-viewpoint video technology. The small computational costs incurred at the server and at the mobile terminal are justified by a drastic reduction, 63% in the case of the LMS and 72 % for the Kalman filter, in the amount of feedback packets transmitted on the wireless uplink channel to request a different perspective of the scene in typical FVV usage profiles. The benefits derived from the proposed system enable a considerable gain in terms of power conservation of the battery within the mobile multimedia terminal and improve bandwidth utilisation.

REFERENCES

- [1] S.U. Kum, and K. Mayer-Patel, "Intra-Stream Encoding for Multiple Depth Streams", in Proc. of the 16th Int. Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), pp. 62–67, Newport, RI, USA, May 2006.
- [2] S. Tan, A. Aksay, C. Bilen, G.B. Akar, and E. Arikan, "Error resilient layered stereoscopic video streaming," in Proc. of the Int. Conf. on True Vision Capture, Transmission and Display of 3D Video (3DTV '07), Kos Island, Greece, May 2007.
- [3] C. Fehn, P. Kauff, M. Op de Beeck, F. Ernst, W.I. Jsselsteijn, M. Pollefeys, L. Van Gool, E. Ofek, and I. Sexton, "An evolutionary and optimised approach on 3D-TV," in Proc. of the International Broadcast Conference (IBC '02), pp. 357–365, Amsterdam, The Netherlands, Sept. 2002.
- [4] A. Smolic, H. Kimata, and A. Vetro, "Development of MPEG Standards for 3D and Free Viewpoint Video", in Proc. of the SPIE Conf. Optics East 2005: Communications, Multimedia & Display Technologies, Vol. 6014, pp. 262–273, Nov. 2005.
- [5] X. Guo, Y. Lu, F. Wu, W. Gao, and S. Li, "Free viewpoint switching in multi-view video streaming using Wyner-Ziv video coding", in Proc. of SPIE Visual Communications and Image Processing 2006, San Jose, California, USA, Vol. 6077, pp. 298–305, Jan. 2006.
- [6] J. Carranza, C.M. Theobalt, M. Magnor, and H.P. Seidel, "Free-viewpoint video of human actors," in ACM Trans. on Graphics (ACM SIGGRAPH 2003), Vol. 22, No. 3, pp. 569–577, Jul. 2003.
- [7] C. Zhang, L. Huo, C. Xia, W. Zeng, and W. Gao, "A Virtual View Generation Method for Free-Viewpoint Video System," in Proc. of 2007 Int. Symp. on Intelligent Signal Processing and Communication Systems (ISPACS 2007), Xiamen, China, pp. 361–364, Nov. 2007.
- [8] V. Nozick, and H. Saito, "On-line Free-Viewpoint Video: From Single to Multiple View Rendering," in Int. Journal of Automation and Computing (IJAC), Vol. 5, No. 3, pp. 257–267, Jul. 2008
- [9] T. Stockhammer, M.M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," in IEEE Trans. on Circuits and Systems for Video Technology, Vol.13, No.7, pp.657–673, Jul. 2003.
- [10] H.K. Arachchi, and W.A.C. Fernando, "H.264/AVC in Mobile Environment: A Review," in SPIE Optical Engineering Journal, Vol. 44, Issue 9, Sep. 2005.
- [11] T. Stockhammer, and T. Wiegand, "H.264/AVC for wireless applications," in IEEE international workshop on mobile multimedia communications, Munich, 2003
- [12] J. Kwon, M. Kim, and C. Choi, "Multiview Video Service Framework for 3D Mobile Devices," in Proc. of the Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing, pp. 1231–1234, Aug. 2008.
- [13] B. Widrow, and M.E. Hoff, "Adaptive switch circuits," in IRE WESCON Convention Record, Vol. 55, part 4, pp. 96–104, New York, NY, USA, Aug. 1960.
- [14] A. Zhang, and N. Gong, "A Novel Variable Step Size LMS Algorithm Based on Neural Network," in Proc. of the Int. Conf. on Intelligent Systems and Knowledge Engineering (ISKE 2007), Chengdu, China, Oct. 2007.
- [15] R.E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", in Trans. of the ASME- Journal of Basic Engineering, Vol. 82, pp. 35–45, Mar. 1960.
- [16] X. Huan, and S. Mannor, "A Kalman Filter Design Based on the Performance/Robustness Tradeoff", in Proc. of the 44th Annual Allerton Conf. on Communication, Control, and Computing, pp. 59–63, Sep. 2007. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.