

# Applying Prediction Techniques to Reduce Uplink Transmission and Energy Requirements in Mobile Free-Viewpoint Video Applications

Clifford De Raffaele, *Graduate Student Member, IEEE* and Carl James Debono, *Senior Member, IEEE*

Department of Communications & Computer Engineering  
University of Malta  
Msida, MSD 2080, Malta  
cderaffaele@iee.org and c.debono@iee.org

**Abstract** - The increased efficiency of video compression algorithms, the improvements registered in reliability, data rates and quality of service of wireless transmission technologies, and the development of mobile multimedia terminals have made possible the implementation of Free-Viewpoint Video (FVV) technology on mobile platforms. The mobile environment however presents several restrictions. Two of these limiting factors being bandwidth constraints and energy availability in battery-operated mobile terminals. This paper looks at the possibility of employing prediction algorithms at the FVV server to anticipate the next viewpoint expected by the user. In doing so, the number of uplink requests is reduced to situations where the estimated view defers from the requested one and the associated transmissions required in retraining the algorithm once this occurs. Simulation results on two different prediction techniques demonstrate that the uplink transmission rate is reduced by up to 96.7% when emulating a conventional FVV usage scenario. Both prediction algorithms infer a substantial decrease in the mobile terminal's power consumption and reduce the network's uplink bandwidth utilization.

**Keywords**-Free-Viewpoint; Recursive Least Squares; Linear Regression; Wireless Transmission.

## I. INTRODUCTION

In the last decade, both mobile and multimedia communications have experienced unequalled growth and commercial success [1]. The achievement of the latter was augmented with the introduction of multiview broadcasting and similar interactive technologies that have consistently increased the consumer's expectations [2-3]. Technologies such as Free-Viewpoint Video (FVV), in which the user can interactively select a desired viewpoint and observe a three-dimensional panoramic scene from arbitrary perspectives [4], are able to expand the user's experience far beyond what is presented by existing conventional media [5].

The realization of such a technology, which entails the capturing of a unique scene from multiple views, requires the implementation of a number of calibrated cameras distributed around the site. Feasibility issues however, demand that the number of cameras is restricted [6], and proximity of adjacent cameras be limited to at least the physical separation, as depicted in Fig. 1. Thus, for the viewer to observe a continuous stream of viewpoints without being limited to the actual camera locations, techniques of Intermediate View Reconstruction (IVR) must be adopted [5-6]. This involves synthesizing virtual viewpoints between

the physical cameras to provide a gradual change in perspective whilst moving from one tangible camera location to the next. Alas, these techniques present a high-computational cost and can only be implemented in a real-time scenario when executed on custom computing architectures.

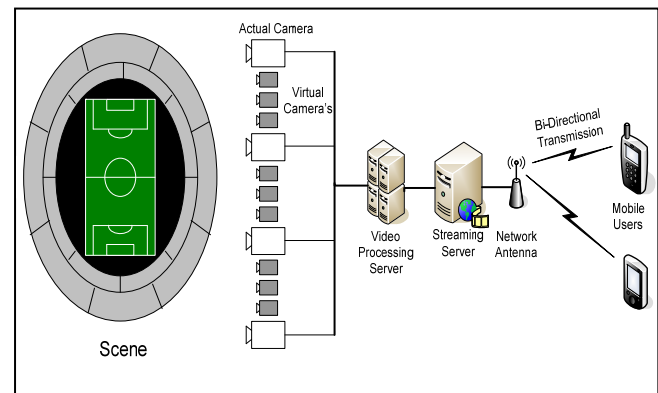


Figure 1. Block diagram of the entire system representing the scene capturing by multiple cameras, video-based rendering servers, the streaming server and the wireless network.

Concurrent to the progress in multimedia, emerging mobile client devices are all equipped with a liquid crystal display and sufficient processing power to allow real-time presentation of multimedia information [7]. Out of all the modalities of desirable future mobile multimedia, motion video is most likely to be one of the major applications [1]. Nonetheless, this technology provides a strong impact on the network infrastructure due to its strict delay requirements and wireless bandwidth restrictions. Beyond these limitations, the harsh wireless transmission environment presents additional technical challenges such as attenuation, fading, multi-user interference and spatio-temporal varying channel conditions [8]. Such issues, together with current business models in emerging wireless systems, where the end-user's costs are proportional to the reserved bit-rate or the number of bits transmitted over a radio link [9], drive future multimedia network systems to provide a more efficient framework for the deployment of services. A further bottleneck encountered in the implementation of FVV technology is intrinsic to mobile terminals. Due to their hand-held nature, such devices are severely constrained in processing power, storage capacity and energy resources [10].

These implications make the transmission of several views for rendering at the mobile terminal impractical under several perspectives. Thus, a sensible implementation for FVV is to implement the video-based rendering (VBR) techniques on the server's side and transmit only the required view to the mobile terminal [11]. Inherently, this strategy demands the implementation of a feedback channel from the mobile device to request the required view perspective stipulated by the user. This necessitates that a request is made on the uplink channel after each received view, leading to service delays, bandwidth usage and terminal power consumption.

This paper builds on [12] to further reduce the amount of feedback transmissions necessary for successful free-viewpoint operation. Prediction strategies are implemented on the multimedia server to forecast the ensuing user's view request. The server interprets the lack of feedback as a confirmation that the correct estimate was generated. Thus, the mobile terminal only needs to transmit when the received view does not match the user's indicated standpoint, at which point the algorithm is retrained to converge to the new FVV pattern. The reduced use of the mobile terminal's transmitter preserves its battery power and reduces the uplink bandwidth utilization. The algorithms implemented also manage to minimize the round-trip delays incurred by the system due to the otherwise continuous transmissions on the uplink channel.

This paper is organized as follows; the prediction algorithms used in this work are discussed in Section II together with the details on the parameters employed. Section III presents the simulation results which highlight the curtailment in feedback transmission attained by each solution and gives a quantitative measure of the battery power saved. Finally, comments and conclusions are presented in Section IV.

## II. VIEW PREDICTION ALGORITHMS

The use of prediction algorithms for estimating system behavior requires the utilization of a subset of the observed readings acquired previously from the system, to construct the original set of data within some pre-defined precision tolerance [13]. Viewpoint prediction is performed by employing a combination of received data, preceding data predictions, and a priori knowledge of the system's operation. This work studies and compares the performance of two algorithms, the recursive least square algorithm and the linear regression algorithm, in FVV applications in an effort to reduce uplink transmission requests and increase the battery lifetime of mobile terminals.

### A. Recursive Least Squares

A possible solution is to employ the Recursive Least Squares (RLS) algorithm which aims at minimizing the sum of squares difference between the modeled filter output and the desired signal [14]. This is achieved by calculating the optimum filter weights using an exponentially weighted estimate of the input autocorrelation and cross-correlation [15]. Owing to the prediction filter nature of the algorithm, the implementation during learning mode entails an iterative

process whereby current weight vectors are employed for future data prediction, and subsequently measurement data is considered as reference for updating of the internal weight vectors.

The adaptation of the RLS is renowned to pursue fast convergence in time-varying environments even in cases where the eigenvalue spread of an input signal correlation matrix is large [16]. Unfortunately, this benefit is attained at the cost of high complexity, having an order of  $\mathcal{O}(N^2)$  FLOPS per sample ( $N$  being the filter length) [17], and an increased sensitivity to mismatch when compared to similar least squares algorithms [18]. When new samples of the incoming signal are received, the coefficient vector update solution for the least-squares problem can be computed in recursive form using [19]:

$$\underline{w}(k+1) = \underline{w}(k) + \underline{g}(k)e(k), \quad (1)$$

where  $w(k)$  is the coefficient vector of the adaptive filter of length  $N$  at time  $k$ ,  $e(k)$  is the difference between the generated filter output and the desired signal, and  $g(k)$  is known as the Kalman gain. In order to render the algorithm feasible for real-time applications, the matrix inversion lemma technique [14] is applied to obtain a simple update for the inverse of the data input equation:

$$P^{-1}(k) = \frac{1}{\lambda} \left[ P^{-1}(k-1) - \underline{g}(k)\underline{x}^T(k)P^{-1}(k-1) \right], \quad (2)$$

where  $x(k)$  is the input data vector, and  $\lambda \in (0,1]$  is called the forgetting factor. The latter parameter has direct implication on the memory of the algorithm, with unity implying infinite memory and is only suitable for statistically stationary systems. The algorithm contains no a priori information on the system at initialization, thus, an approximate initialization technique is employed for the covariance matrix by setting  $P^{-1} = \delta I$ , a scaled version of the identity matrix [19].

### B. Linear Regression

Another possibility is to exploit the Linear Regression (LR) algorithm which is a statistical method utilized to model the relationship between two or more variables by fitting a least squares linear equation to the observed data [20]. The implementation of a linear model provides several advantages for a real-time system, particularly due to its computational simplicity and inherent ease of use [23]. This statistical technique holds validity under the assumption that the output is linearly dependent on the input [21], thus having a system model of the form:

$$\hat{Y} = \beta_1 \times X + \beta_0 + \varepsilon, \quad (3)$$

where  $\hat{Y}$  denotes the predicted value of the output for a given dependent variable input  $X$ . The additional amount  $\varepsilon$  represents the residual error from the regression line, whilst the variables in the model function  $\beta_0$  and  $\beta_1$  are referred to

as the model parameters and are estimated from a training set of  $n$  observations, in the form of  $(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)$ , using [22]:

$$\beta_1 = \frac{\sum_{i=1}^n (X_i - \bar{X}) \times (Y_i - \bar{Y})}{\sum_{i=1}^n (X_i - \bar{X})^2}, \quad (4)$$

$$\beta_0 = \bar{Y} - b_1 \times \bar{X}, \quad (5)$$

where  $\bar{X}$  and  $\bar{Y}$  represent the mean values of the respective variables.

The calculations defined in equations (4) and (5) are only performed during the training phase of the algorithm so that the model functions converge. Once the aforementioned parameters are established, the computational load for executing the LR algorithm on the server, is only that of computing a single multiplication and two additions as described by equation (3).

### III. SIMULATION AND RESULTS

To simulate and analyze the employment of a free-viewpoint scenario objectively, a typical situation was modeled in Matlab<sup>®</sup> for two separate FVV usage profiles. The FVV system considered allows the view of nine distinct virtual views in between each physical camera location. The simulated scenarios start with the user receiving a view from a single camera and not requesting any view changes. At an arbitrary point in time, the user starts changing the viewing angle at a particular velocity, thus performing free-viewpoint operation. Subsequently, this motion pattern will change casually, with each epoch having a different rate of change and duration of the varying FVV perspective. This results in a linear motion as illustrated in the video sequence shown in Fig. 2, whereby the change of perspective is gradual from one camera outlook to another, and this can be modeled with straight lines.



Figure 2. Video sequence for linear Free-Viewpoint motion.

The simulated usage profiles are: (i) system A which allows only a velocity pattern in which free-viewpoint

motion can be either static or else change with at least one virtual view per time step, and (ii) system B which allows the same functions of system A, together with the increased facility of providing the user with the capability to scroll through views more leniently and hence take more than one time step to change from a virtual standpoint to a neighboring one. Due to the quantized nature by which view changes can be requested, the option proposed by system B will introduce a jittered pattern of motion. In this situation, the view will remain constant for a small period of time and then alter by the minimum quantized amount, giving the impression of a slow changing perspective. Thus, the speed for this scenario is proportional to the number of time steps in which the view remains constant between changes.

The performance evaluation of the RLS and LR algorithms is compared to the system specifications described by the reference FVV architecture found in [11]. This reference methodology, whose view request profiles for systems A and B are shown in Fig. 3(a) and Fig. 3(d) respectively, portrays the operation of an FVV system where the mobile terminal transmits to the server in each time step to demand the view being requested by the user. The server processes the requested viewpoint and transmits the perspective back to the terminal. The image from the modified outlook would then be viewed on the mobile terminal in the subsequent time slot.

The first implementation of the proposed system involves the adaptation of the RLS algorithm on the server node to track the user demands for FVV viewing. A training set of 8 samples is transmitted from the mobile terminal each time there is a discrepancy between the received view and the user's demanded one. Upon receiving the initial feedback packet, the server turns the RLS algorithm online and commences a training routine. During this period, a recursive approach is utilized by the algorithm in order to adapt itself to the new linear viewing pattern. Following the updating process, the RLS algorithm is turned back offline, and the server predicts the views that will be demanded by the user using the algorithm's converged values.

The simulation result for the performance of the RLS algorithm on system A, shown in Fig. 3(b), shows that although the filter length was limited to five, to conserve the memory footprint, convergence to the desired output is still achieved at a satisfactory speed. This feature was optimized heuristically by assigning different values to  $\lambda$ . The latter was varied adaptively during the simulation from an initial small value to promote speed-up to subsequently a value closer to unity to avoid overshoot, oscillations and prevent system instability. Although convergence is obtained within a few iterations, some additional epochs must be computed prior to executing the RLS algorithm offline. This is done to ensure extensive reduction of the trailing error between signals, which would otherwise influence the performance of the system once it operates in open loop. The RLS algorithm however, struggled to achieve suitable performance when applied to the system B scenario, as can be visualized in Fig. 3(e). This occurs because of the constant gradient changes in the quantized system at low velocity which the RLS algorithm finds difficult to predict once in offline mode.

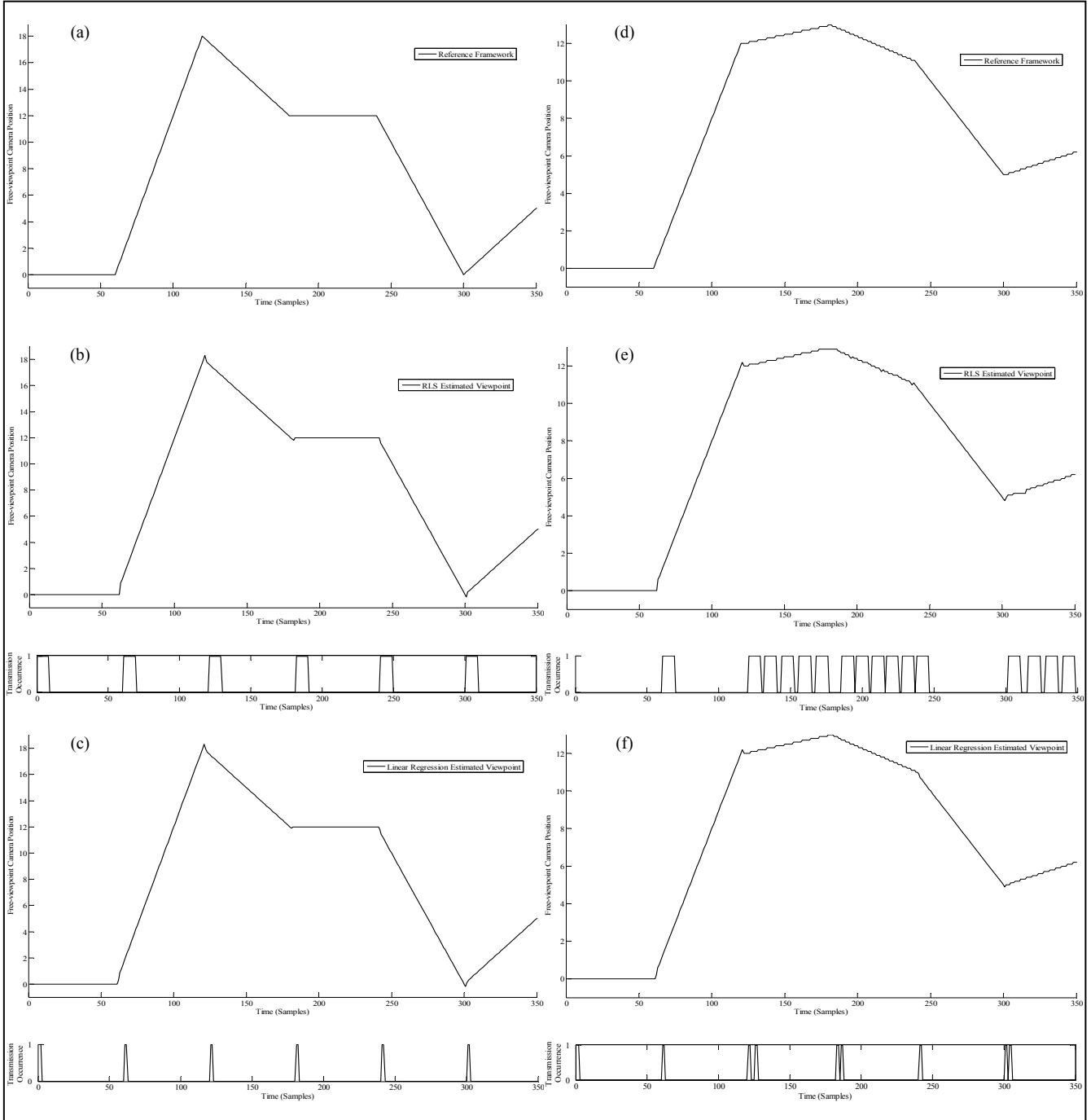


Figure 3. Simulation results and the associate transmission occurrences comparing the proposed algorithms in a typical FVV scenario for System A: (a) reference, (b) Recursive Least Squares, (c) Linear Regression, and System B: (d) reference, (e) Recursive Least Squares, (f) Linear Regression.

The RLS algorithm provides a significant improvement on the standard reference system. With a manageable increase in computational time and resources on the server, an 87% reduction in traffic on the uplink channel for FVV feedback transmissions is obtained for the typical FVV scenario described by system A. However, the gain in performance reduces to 63% in system B as the RLS suffers when slow profile changes are applied.

The second solution employs the LR algorithm on the server to predict the desired view sequences requested by the user. The results implicitly expose the benefits derived from the adaptation characteristics of this algorithm. Instead of attempting to predict the next viewpoint, the server predicts the velocity by which the viewpoints are changing between defined time steps, thus generating a regression line which indicates the rate of change demanded by the user.

Due to the inherent nature of the system operation, the linear algorithm for prediction was limited to a first-order model, as this reduced the complexity of the algorithm, whilst still providing reliable accuracy in the regression line generated. Since observations done at the mobile terminal are inputted directly by the user, there are no sources of error during the acquisition of information. Thus, the training set used by the algorithm is composed of only two samples, as this is the minimum amount of observations required for the LR algorithm to produce a robust regression line.

After training, the mobile terminal simply checks whether the received view matches that demanded by the user. If the prediction is correct, the mobile terminal refrains from transmitting feedback, and the server assumes that the estimate it has calculated is correct. Otherwise, feedback is delivered to the server which will wait for the second data packet to be transmitted before restarting the execution of the algorithm. Once data is received from the first feedback packet, the view is altered in the desired direction to signify that the request has been detected and adhered to. At the second iteration, the server computes the regression line for the velocity of view changes and adjusts for the previous delay, hence re-synchronizing the prediction process with the demand of the user on the mobile terminal.

For the simplistic system A scenario, portrayed in Fig. 3(c), the LR algorithm converges faster than the RLS counterpart to the desired signal output, since the algorithm takes direct advantage of the linear nature in which the FVV viewpoint is modified. Apart from its outstanding convergence speed and reduced amount of necessary feedback transmission, the LR algorithm also manages to cope with the demand for slow free-viewpoint movement. As evident in Fig. 3(f), the algorithm keeps track of the amount of time steps that the user spends on a fixed viewpoint between view changes by means of a dedicated counter. This information is utilized once an error is retransmitted by the mobile terminal, at which point a linear regression is computed taking into account the new position and the previous samples. Since the system is quantized in nature, the LR will perform well when the output is truncated to the resolution of the view. Hence the algorithm will be able to predict any user's input pattern after two error signals.

The LR algorithm provides better results compared to the RLS counterpart, since, for the system A scenario the reduction in uplink transmissions is of 96.7%. A consistent performance gain is also registered with the linear regression in the more flexible scenario represented by system B, were on average, the uplink transmission reduction achieved by the algorithm amounted to 92%. The small discrepancy between both systems adopting the LR algorithm comes from the additional transmissions required to cater for the slow movements in viewpoint patterns. Due to the small amount of training samples required by the LR algorithm, this methodology is also able to better accommodate dynamic view pattern changes.

To obtain a quantitative analysis of the energy saved at the mobile terminal by the implementation of any one of the proposed solutions, a simulation of the energy consumed by the transceiver of a mobile station during FVV was

performed. The mobile terminal is assumed to be equipped with a 1500mAh battery and consumes a current of 25mA during transmission and a current of 20mA during the reception of data. The considered scenario is that of a terminal operating in a network employing HSPA technology with 7.2Mbps/s and 1.4Mbps/s on the downlink and uplink respectively. The free-viewpoint operation requires a single packet of 50 bytes to send view requests and a mean of four 200 byte packets on the downlink channel. The transmitted data consists of a continuous video stream in standard CIF resolution using the H.264/AVC baseline profile.

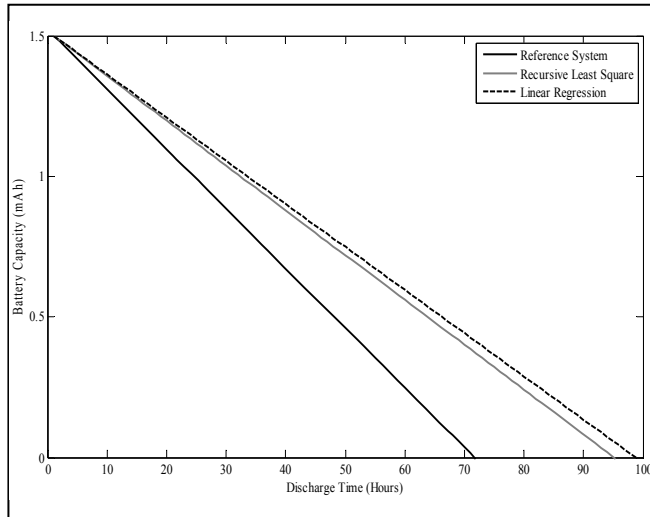


Figure 4. Simulation results showing the battery power consumption of the proposed RLS algorithm and Linear Regression algorithm when compared to the current reference system.

As evidenced in Fig. 4, the drastic reductions in the necessary transmissions on the uplink channel achieved by both solutions significantly reduce the battery power consumed by the mobile terminal during FVV operation. Although the amount of data which is transmitted on the uplink channel is much smaller than the received video, the energy saved is still significant for a resource limited device. This considerable drop in energy consumption occurs because the transmitter is the most power hungry component of any mobile terminal. Furthermore, the solutions enhance the Quality of Service (QoS) provided by the FVV system. This occurs because the algorithms offer a pro-active network implementation by predicting and providing the required view perspective to the user instantaneously, thus reducing the round-trip delays incurred in the reference method.

#### IV. CONCLUSION

This paper has presented the adaptation of prediction algorithms to free-viewpoint video technology. The small computational costs incurred at the server node by this implementation are justified by a drastic reduction of up to 96.7% in the amount of feedback packets transmitted on the wireless uplink channel. The requests for demanding a

different perspective of the scene in a typical FVV usage profile are replaced by the server's predictions through either the RLS or the LR algorithm. The benefits achieved from such systems enable a considerable gain in terms of power conservation for the multimedia mobile terminal as well as a reduced utilization of the uplink bandwidth. Moreover, the average round-trip delay is drastically reduced making the real-time experience more pleasing.

Research is currently ongoing to implement test bed infrastructures for subjective evaluation. Furthermore, the exploitation of the proposed methodologies to several application scenarios of FVV multimedia will be studied.

#### REFERENCES

- [1] A. Kubotka, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, and C. Zhang, "Multiview imaging and 3DTV," *IEEE Trans. Signal Processing*, vol. 24, pp. 10–21, Nov. 2007.
- [2] 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 Int. Broadcast Conf. Amsterdam*, 2002, pp. 357–365.
- [3] 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 *SPIE Visual Communications and Image Processing 2006*, California, 2006 pp. 298–305.
- [4] A. Smolic, H. Kimata, and A. Vetro, "Development of MPEG standards for 3D and free viewpoint video," in *SPIE Conf. Optics East 2005: Communications, Multimedia & Display Technologies*, Boston, 2005, pp. 262–273.
- [5] J. Carranza, C.M. Theobalt, M. Magnor, and H.P. Siedel, "Free-viewpoint video of human actors," *ACM Transactions on Graphics*, vol. 22, pp. 569–577, Jul. 2003.
- [6] C. Zhang, L. Huo, C. Xia, W. Zeng, and W. Gao, "A virtual view generation method for free-viewpoint video system," in *Proc. of the 2007 Int. Symp. on Intelligent Signal Processing and Communication Systems*, Xiamen, 2007, pp. 361–364.
- [7] T. Stockhammer, and M.M. Hannuksela, "H.264/AVC Video for Wireless Transmission," *IEEE Trans. Wireless Communications*, vol. 12, pp. 6–13, Aug. 2005.
- [8] T. Stockhammer, M.M. Hannuksela, and T. Wiegand, "H.264/AVC in Wireless Environments," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, pp. 657–673, Jul. 2003.
- [9] T. Stockhammer, and T. Wiegand, "H.264/AVC for wireless applications," in *Proc. of the IEEE Inter. Workshop on Mobile Multimedia Communications*, Munich, 2003.
- [10] H.K. Arachchi, and W.A.C. Fernando, "H.264/AVC in mobile environment: a review," *SPIE Optical Engineering*, vol. 44, pp. 097006.1–097006.7, Sep. 2005.
- [11] 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*, Harbin, 2008, pp. 1321–1234.
- [12] C.J. Debono and C. De Raffaele, "Minimizing Uplink Data in Wireless Free-Viewpoint Video Transmission Applications," in *Proc. of the 2009 Third Int. Conf. on Next Generation Mobile Applications, Services and Technologies (NGMAST 2009)*, Cardiff, September 2009, pp. 298–302.
- [13] C.J. Debono, and N.P. Borg, "The implementation of an adaptive data reduction technique for wireless sensor networks," in *Proc. of the IEEE 8th Int. Symp. on Signal Processing and Information Technology*, Sarajevo, 2008, pp. 402–406.
- [14] G.C. Goodwin, and R.L. Payne *Dynamic System Identification: Experiment Design and Data Analysis*. Academic Press, New York, 1977.
- [15] P.C. Wei, J.R. Zeidler, and W.H. Ku, "Adaptive recovery of a chirped signal using the RLS algorithm," *IEEE Trans. Signal Processing*, vol. 45, pp. 363–376, Feb. 1997.
- [16] P.S.R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, 3rd ed. Springer Publishers, New York, 2008.
- [17] Y.V. Zakharov, G.P. White, and J. Liu "Low-complexity RLS algorithms using dichotomous coordinate descent iterations," *IEEE Trans. Signal Processing*, vol. 56, pp. 3150–3161, Jul. 2008.
- [18] Z. Tian, K.L. Bell, and H.L. Van Trees "A recursive least squares implementation for LCMP beamforming under quadratic constraint," *IEEE Trans. Signal Processing*, vol.49, pp. 1138–1145, Jun. 2001.
- [19] S. Haykin, *Adaptive Filter Theory* 3rd ed. Prentice Hall, New Jersey, 1996.
- [20] K.B. Song, S.K. Ha, J.W. Park, D.J. Kweon, and K.H. Kim, "Hybrid load forecasting method with analysis of temperature sensitivities," *IEEE Trans. Power Systems*, vol. 21, pp. 869–876, May 2006.
- [21] A. Gu, and A. Zakhor, "Optical proximity correction with linear regression," *IEEE Semiconductor Manufacturing*, vol. 21, pp. 263–271, May 2008.
- [22] N.R. Draper, and H. Smith, *Applied Regression Analysis*, 3rd ed. John Wiley & Sons Inc., New York, 1998.
- [23] M.R. Gupta, E.K., Garcia, and E. Chin, "Adaptive local linear regression with application to printer color management," *IEEE Trans. Image Processing*, vol. 17, pp. 936–945, Jun. 2008.