Support Vector Machine based Fusion for Multi-view Distributed Video Coding

Frederic Dufaux

SUPPORT VECTOR MACHINE BASED FUSION FOR MULTI-VIEW DISTRIBUTED VIDEO CODING

Frederic Dufaux

Laboratoire de Traitement et Communication de l'Information (LTCI) - CNRS UMR 5141
Télécom ParisTech, F-75634 Paris Cedex 13, France
frederic.dufaux@telecom-paristech.fr

ABSTRACT

In multi-view Distributed Video Coding, Side Information can be computed either from previously decoded frames in the same view, or from previously decoded frames in adjacent views. In this paper, we address the problem of effectively fusing these two predictions. For this purpose, we propose two Support Vector Machine based fusion algorithms. We also identify a number of features to be used in classification. Experimental results show the efficiency of the proposed approach and its robustness over test sequences with greatly varying characteristics.

Index Terms— Multi-view, Distributed Video Coding, Support Vector Machine, fusion, Side Information

1. INTRODUCTION

Digital media content is becoming ubiquitous, thanks to rapid progress in information and communication technologies. Nevertheless, digital video is demanding in terms of resources. The emergence of this digital age has been enabled by tremendous advances in video coding technologies. In particular, MPEG and ITU-T have produced a number of standards for video coding, based on the two principles of predictive and transform coding.

More recently, multi-view video coding has gained a lot of interests. Indeed, this representation is attractive for a number of emerging applications such as stereoscopic video, 3D TV, free viewpoint video, or camera networks for surveillance and monitoring.

Distributed Video Coding (DVC) has emerged as a novel coding paradigm a few years ago. Its theoretical foundation is based on two information theory theorems: Slepian-Wolf [1] and Wyner-Ziv [2]. By exploiting these results, the first practical DVC schemes have been proposed in [3][4]. Recent DVC developments are reviewed in [5][6].

The DVC framework presents a number of advantages. First, the complexity can be flexibly allocated between the encoder and decoder. In particular, DVC allows for low complexity encoding. Second, DVC is robust to transmission errors, thanks to its intrinsic joint source-channel formulation. Third, DVC provides with a new type of codec independent scalability. Finally, with its ability to exploit inter-camera correlation at the decoder side, without communication between cameras, DVC is also well-suited for multi-view video coding where it offers a noteworthy architectural advantage. This may prove a significant advantage from a system implementation standpoint, avoiding complex and power consuming networking.

In the case of multi-view DVC, the Side Information (SI) can be estimated either from previously coded in the same view by motion compensated temporal interpolation, or from previously decoded frames in the adjacent views by disparity compensated view prediction. In this paper, we address the challenge to effectively fuse these two sources of information. SI fusion can be seen as a classification problem with two classes. Therefore, we propose to use Support Vector Machine (SVM), a powerful supervised machine learning technique. More specifically, we introduce to variants, performing either binary decision or linear combination of the two SI. We also indentify a number of features which are suitable in the SVM classification process.

This paper is structured as follow. Related work is first reviewed in Sec. 2. After presenting the multi-view DVC architecture, the proposed SVM-based SI fusion algorithm is introduced in Sec. 3. We also discuss the selection of features to be used for classification. Next, the performance of the proposed approach is assessed in Sec. 4. Finally, Sec. 5 draws conclusion and outlines future perspectives.

2. RELATED WORK

DVC is promising for multi-view video coding. Indeed, in this context, it allows for an architecture where cameras do not need to communicate, while still enabling the exploitation of inter-view correlation during joint decoding. This may prove a significant advantage from a system implementation standpoint.

The major difference in multi-view DVC, in contrast with mono-view, is that the SI can be computed not only

from previously decoded frames in the same view, but also from frames in neighboring views.

A number of techniques have been proposed in order to perform inter-view prediction. As a straightforward extension of Motion Compensated Temporal Interpolation (MCTI) [7], in Disparity Compensation View Prediction (DCVP) [8] the prediction is carried out by motion compensation of the frames in other views using disparity vectors. With Multi-View Motion Estimation (MVME) [9], motion vectors are estimated in the side views and then applied to the view to be WZ encoded. This requires to estimate disparity vectors between views. In [10], inter-view prediction is carried out using a homography model estimated by robust global motion estimation. It results in significant SI quality improvement. In View Synthesis Prediction (VSP) [11], pixels are projected to the 3D world coordinates using intrinsic and extrinsic camera parameters, and are subsequently used to predict other views. However, VSP requires depth information and its performance depends on the accuracy of the camera calibration as well as the depth estimation. Finally, View Morphing (VM) [12] is commonly used to synthesized virtual views, using principles of projective geometry.

In multi-view DVC, the SI can be generated either from frames in the same view using motion compensated temporal interpolation, or from frames in adjacent views using inter-view interpolation. The next challenge is to optimally fuse these multiple SI at the decoder side. In [13], a technique is proposed to fuse intra-view temporal and inter-view homography side information. It exploits the previous and next key frames to choose the best predictor on a pixel basis. Two fusion techniques are introduced in [14]. They rely on a binary mask to estimate the reliability of each prediction. The latter is computed on the side views and projected on the view to be WZ encoded. However, depth information is required for inter-camera disparity estimation. The technique in [15] combines a discrete wavelet transform and turbo codes. Fusion is performed between intra-view temporal and inter-view homography side information, based on the amplitude of motion vectors. The method in [16] follows a similar approach, but relies on the H.264/AVC mode decision applied on blocks in the side views. Taking a different approach, in [8] a binary mask is computed at the encoder and then transmitted to the decoder in order to help the fusion process. Video sensors to encode multi-view video are described in [17]. The scheme exploits both inter-view correlation by disparity compensation from other views, as well as temporal correlation by motion compensated lifted wavelet transform. The proposed scheme leads to a bit rate reduction by performing joint decoding when compared to separate decoding. In [18], three fusion algorithms are proposed. The first two approaches are based on the residual error in temporal or inter-view motion compensated interpolation. The third approach also takes into account the motion vectors norm. A fusion technique based on genetic algorithm is used to optimally combined multiple SI in [19], for the case of mono-view DVC. Finally, an overview and assessment of different inter-view prediction techniques for SI is given in [20]. A new SI and fusion algorithm is also proposed, based on an iterative reconstruction process.

Note that in all the above techniques, inter-view correlation is exploited, although the cameras do not need to communicate.

3. PROPOSED FUSION ALGORITHM

In this section, the introduce the proposed SVM-based fusion algorithm. We first present the multi-view DVC architecture, which is based on the DISCOVER DVC codec [21]. Next, we explicitly state the fusion problem as considered in this paper. We then describe in more details the proposed fusion algorithm. Finally, we motivate the selection of features to be used in the SVM classifier.

3.1. Multi-view DVC Architecture

In this paper, we more specifically consider the DISCOVER DVC codec [21], as illustrated in Fig. 1.

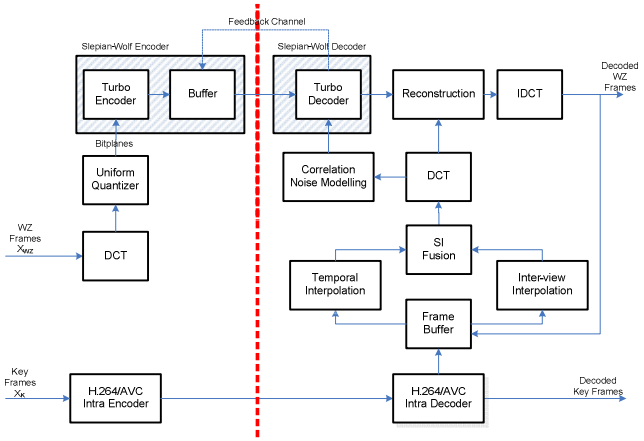

Fig. 1. Proposed multi-view DVC architecture.

Input frames are first split into Group of Pictures (GOP). Key frames, corresponding to the first frame of each GOP, are conventionally encoded using H.264/AVC Intra coding. In turn, WZ frames undergo a DCT transform followed by uniform quantization. The quantized values are then split into bitplanes which go through a Turbo encoder. At the decoder, SI approximating the WZ frames is generated from the previously decoded reference frames. At this stage, both temporal and inter-view interpolation are used to produce two SI, which are then fused. The resulting SI after fusion is used in the Turbo decoder, along with the parity bits of the WZ frames requested via a feedback channel, in order to reconstruct the bitplanes, and subsequently the decoded video sequence.

3.2. Problem Statement

Hereafter, without loss of generality, we more specifically consider the multi-view configuration with three views as depicted in Fig. 2, and we consider the coding of the central view v.

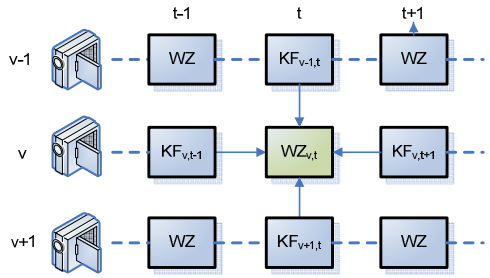

Fig. 2. Multi-view configuration under consideration.

Each view is encoded using a GOP size of 2, with a time lag between adjacent views. In this way, the WZ frame in view v at time t, $\mathrm{WZ}_{v,t}(\mathbf{p})$ is surrounded by four key frames: the preceding and following key frames in the same view, $K_{v,t-1}(\mathbf{p})$ and $K_{v,t+1}(\mathbf{p})$, and the two adjacent key frames at time t in views v-1 and v+1, $K_{v-1,t}(\mathbf{p})$ and $K_{v+1,t}(\mathbf{p})$.

In this paper, we consider the fusion of MCTI [7] and DCVP [8]. More precisely, temporal interpolation $P_\tau(\mathbf{p})$ at pixel location $\mathbf{p}$ is given by averaging backward and forward motion compensated key frames $\widetilde{K}_{v,t-1}(\mathbf{p})$ and $\widetilde{K}_{v,t+1}(\mathbf{p})$.

$$P_\tau(\mathbf{p}) = \frac{1}{2}\left(\widetilde{K}_{v,t-1}(\mathbf{p}) + \widetilde{K}_{v,t+1}(\mathbf{p})\right) \tag{1}$$

Similarly, inter-view prediction $P_\kappa(\mathbf{p})$ is computed as the average of left and right disparity compensated key frames $\widetilde{K}_{v,t-1}(\mathbf{p})$ and $\widetilde{K}_{v,t+1}(\mathbf{p})$.

$$P_\kappa(\mathbf{p}) = \frac{1}{2}\left(\widetilde{K}_{v-1,t}(\mathbf{p}) + \widetilde{K}_{v+1,t}(\mathbf{p})\right) \tag{2}$$

The fusion problem can then be cast as the effective combination of $P_\tau(\mathbf{p})$ and $P_\kappa(\mathbf{p})$.

3.3. Previous Fusion Algorithms

We now explicitly formulate previously proposed fusion algorithms in the context of our multi-view configuration.

Firstly, an Oracle fusion can be defined using the original WZ frame. While not practical, it gives a performance bound for optimal fusion. With Oracle fusion, the SI is straightforwardly given by

$$\mathrm{SI}(\mathbf{p}) = \begin{cases} P_\kappa(\mathbf{p}) & \text{if } \left|P_\kappa(\mathbf{p}) - \mathrm{WZ}(\mathbf{p})\right| < \left|P_\tau(\mathbf{p}) - \mathrm{WZ}(\mathbf{p})\right| \\ P_\tau(\mathbf{p}) & \text{otherwise} \end{cases} \tag{3}$$

In [13], Pixel Difference (PD) fusion is performed by comparing the temporal prediction, respectively inter-view prediction, with the previous and next key frames

$$\mathrm{SI}(\mathbf{p}) = \begin{cases} P_\kappa(\mathbf{p}) & \text{if } \left|P_\kappa(\mathbf{p}) - K_{v,t-1}(\mathbf{p})\right| < \left|P_\tau(\mathbf{p}) - K_{v,t-1}(\mathbf{p})\right| \\ & \text{and } \left|P_\kappa(\mathbf{p}) - K_{v,t+1}(\mathbf{p})\right| < \left|P_\tau(\mathbf{p}) - K_{v,t+1}(\mathbf{p})\right| \\ P_\tau(\mathbf{p}) & \text{otherwise} \end{cases} \tag{4}$$

Exploiting the temporal and inter-view residuals $E_\tau(\mathbf{p})$ and $E_\kappa(\mathbf{p})$,

$$E_\tau(\mathbf{p}) = \left|\widetilde{K}_{v,t+1}(\mathbf{p}) - \widetilde{K}_{v,t-1}(\mathbf{p})\right| \tag{5}$$

$$E_\kappa(\mathbf{p}) = \left|\widetilde{K}_{v+1,t}(\mathbf{p}) - \widetilde{K}_{v-1,t}(\mathbf{p})\right| \tag{6}$$

Motion and Disparity Compensated Difference binary fusion (MDCD-Bin) [18] is defined as

$$\mathrm{SI}(\mathbf{p}) = \begin{cases} P_\kappa(\mathbf{p}) & \text{if } E_\kappa(\mathbf{p}) < E_\tau(\mathbf{p}) \\ P_\tau(\mathbf{p}) & \text{otherwise} \end{cases} \tag{7}$$

Similarly, Motion and Disparity Compensated Difference linear fusion (MDCD-Lin) [18] uses a linear combination of both temporal and inter-view interpolations instead of a binary decision

$$\mathrm{SI}(\mathbf{p}) = \frac{1}{E_\tau(\mathbf{p}) + E_\kappa(\mathbf{p})}\left(E_\tau(\mathbf{p})P_\kappa(\mathbf{p}) + E_\kappa(\mathbf{p})P_\tau(\mathbf{p})\right) \tag{8}$$

3.4. Proposed SVM-based Fusion Algorithm

SI fusion can be seen as a classification problem with two classes: pixels to be predicted by MCTI on the one hand, and pixels to be predicted by DCVP on the other hand.

SVM is a powerful tool for supervised machine learning [22]. Given training samples, namely data points labeled as belonging to one of two classes, a model is first built during the training phase. Subsequently, during the prediction phase, each new input data point is classified as a member of one of the two possible classes.

Data points consist of n-dimensional feature vectors. SVM computes the optimal (n-1)-dimensional hyperplane which separates the two classes and maximizes the distance with nearest data points on each side. This is known as a linear classifier. In this paper, we use the SVM^{light} software implementation [23][24].

The proposed SVM-based fusion algorithm is illustrated in Fig. 3. The following processes take place at the decoder. The first WZ frame of the sequence, $\mathrm{WZ}_{v,t=1}$, is decoded with high quality and without using the proposed fusion algorithm. For instance, fusion algorithms such as defined in Eqs. (4), (7) or (8) can be used. Next, training is carried out using $\mathrm{WZ}_{v,t=1}$ as well as its four surrounding key frames. Namely, features are extracted from the key frames and labels are defined using Oracle fusion. For subsequent WZ frames, features are extracted from the neighboring key

frames. The SVM classifier then assign each pixel to one of the two classes: C_τ for MCTI interpolation or C_κ for DCVP interpolation. This information is then exploited in SI fusion in order to effectively combine MCTI and DCVP predictions.

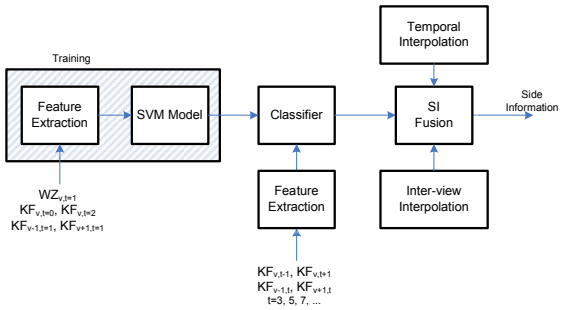

Fig. 3. Proposed SVM-based fusion algorithm.

Based on this SVM classification procedure, we then define two fusion algorithms. SVM-based binary (SVM-Bin) fusion directly uses the classification label of the feature vector $\mathbf{F}(\mathbf{p})$ at pixel $\mathbf{p}$ as belonging to one of the two classes:

$$\mathrm{SI}(\mathbf{p}) = \begin{cases} P_\kappa(\mathbf{p}) & \mathbf{F}(\mathbf{p}) \in C_\kappa \\ P_\tau(\mathbf{p}) & \mathbf{F}(\mathbf{p}) \in C_\tau \end{cases} \tag{9}$$

In a second method, for pixels with low classification confidence, namely the value of the SVM decision function $d(\mathbf{p})$ is smaller than a threshold T, a linear combination of MCTI and DCVP is rather used. For pixels with high classification confidence, binary decision is still taken. This leads to the SVM-based linear (SVM-Lin) fusion:

$$\mathrm{SI}(\mathbf{p}) = \begin{cases} P_\kappa(\mathbf{p}) & \mathbf{F}(\mathbf{p}) \in C_\kappa, d(\mathbf{p}) \geq T \\ \dfrac{(T+d)P_\kappa(\mathbf{p}) + (T-d)P_\tau(\mathbf{p})}{2T} & \mathbf{F}(\mathbf{p}) \in C_\kappa, d(\mathbf{p}) < T \\ P_\tau(\mathbf{p}) & \mathbf{F}(\mathbf{p}) \in C_\tau, d(\mathbf{p}) \geq T \\ \dfrac{(T+d)P_\tau(\mathbf{p}) + (T-d)P_\kappa(\mathbf{p})}{2T} & \mathbf{F}(\mathbf{p}) \in C_\tau, d(\mathbf{p}) < T \end{cases} \tag{10}$$

3.5. Features Selection

The next step is to identify discriminative features to be used in SVM to classify pixels as belonging to the MCTI or DCVP class.

For this purpose, we study the statistical distribution of the following scalar features:

$$\mathrm{SAD}_{Temp} = \sum \left| K_{v,t+1}(\mathbf{p}) - K_{v,t-1}(\mathbf{p}) \right| \tag{11}$$

$$\mathrm{SAD}_{Interview} = \sum \left| K_{v+1,t}(\mathbf{p}) - K_{v-1,t}(\mathbf{p}) \right| \tag{12}$$

$$\mathrm{SAD}_{MCTI} = \sum \left| \tilde{K}_{v,t+1}(\mathbf{p}) - \tilde{K}_{v,t-1}(\mathbf{p}) \right| \tag{13}$$

$$\mathrm{SAD}_{DCVP} = \sum \left| \tilde{K}_{v+1,t}(\mathbf{p}) - \tilde{K}_{v-1,t}(\mathbf{p}) \right| \tag{14}$$

$$\mathrm{SAD}_{PDMCTI} = \frac{1}{2} \sum \left(\left| P_\tau(\mathbf{p}) - K_{v,t-1}(\mathbf{p}) \right| + \left| P_\tau(\mathbf{p}) - K_{v,t+1}(\mathbf{p}) \right| \right) \tag{15}$$

$$\mathrm{SAD}_{PDDCVP} = \frac{1}{2} \sum \left(\left| P_\kappa(\mathbf{p}) - K_{v,t-1}(\mathbf{p}) \right| + \left| P_\kappa(\mathbf{p}) - K_{v,t+1}(\mathbf{p}) \right| \right) \tag{16}$$

The summation is performed in a small window around pixel $\mathbf{p}$, e.g. in our experiments we use a 5x5 window.

Additionally, we also consider the norm of the motion vectors obtained in MCTI, MV_{MCTI}, and the norm of the disparity vectors computed in DCVP, DV_{DCVP}.

In the training stage, data points representative of their respective classes should be used. Moreover, data points for which the right decision results in a significant gain over the wrong decision are preferred.

Hence, for pixels classified as MCTI by an Oracle, we have the following additional constraints:

$$\left| P_\tau(\mathbf{p}) - \mathrm{WZ}(\mathbf{p}) \right| < T_1 \text{ and } \left| P_\tau(\mathbf{p}) - P_\kappa(\mathbf{p}) \right| > T_2 \tag{17}$$

where T_1 and T_2 are threshold values. Similarly, the following conditions are used for pixels classified as DCVP by an Oracle:

$$\left| P_\kappa(\mathbf{p}) - \mathrm{WZ}(\mathbf{p}) \right| < T_1 \text{ and } \left| P_\tau(\mathbf{p}) - P_\kappa(\mathbf{p}) \right| > T_2 \tag{18}$$

Fig. 4 shows the distribution of the features under consideration for data points classified as MCTI or DCVP by an Oracle and satisfying the conditions in Eqs. (17) and (18), for the first WZ frame of the test sequence "Breakdancers".

Based on the analysis of Fig. 4, it appears that the following 6 features are discriminative: SAD_{Temp}, $\mathrm{SAD}_{Interview}$, SAD_{MCTI}, $\mathrm{SAD}_{PD\text{-}MCTI}$, MV_{MCTI}, and DV_{DCVP}. Per consequent, we use these 6 features in the proposed SVM-based fusion.

4. PERFORMANCE ASSESSMENT

In this section, experimental results are presented in order to assess the performance of the proposed SVM-based fusion algorithm.

4.1. Test conditions

Simulations are performed using the DISCOVER DVC codec [21]. Three multi-view test sequences are used: "Breakdancers", "Book Arrival" and "Outdoor", with a spatial resolution of 256 x 192 pixels and a frame rate of 15 frames per second. The first frame of each view is shown in Fig. 5.

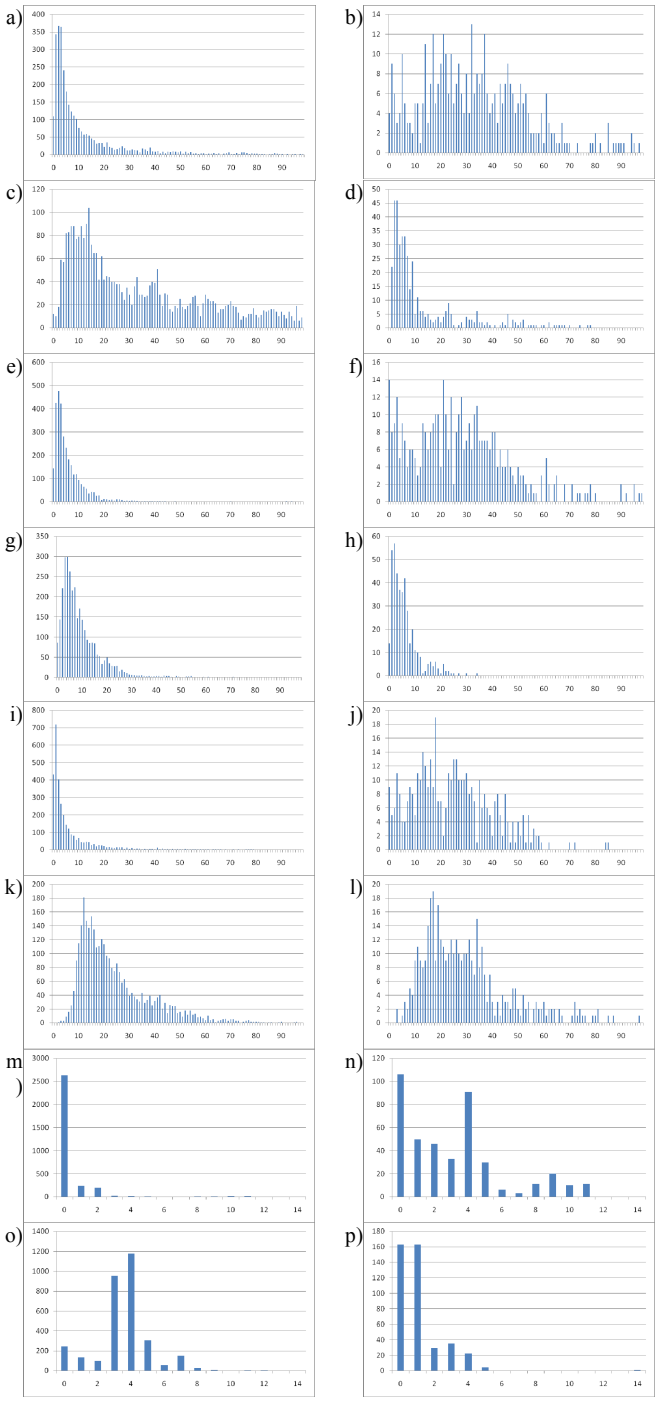

Fig. 4. Distribution for candidate features: a) & b) SAD_{Temp}, c) & d) $SAD_{Interview}$, e) & f) SAD_{MCTI}, g) & h) SAD_{DCVP}, i) & j) $SAD_{PD\text{-}MCTI}$, k) & l) $SAD_{PD\text{-}DCVP}$, m) & n) MV_{MCTI}, o) & p) DV_{DCVP}, a) c) e) g) i) m) o) pixels classified as MCTI by Oracle, b) d) f) h) j) l) n) p) pixels classified as DCVP by Oracle.

Fig. 5. First frame from left, central, and right views: a) Breakdancers, b) Book Arrival, c) Outdoor.

Experiments are carried out at four rate-distortion points, by varying quantization parameters.

4.2. Side Information

The quality of the SI resulting from the proposed SVM-based approaches is first assessed. For this purpose, we compare the SI obtained using 5 fusion algorithms: PD [13], MDCD-Bin [18], MDCD-Lin [18], SVM-Bin and SVM-Lin. The SI obtained by MCTI, DCVP and Oracle fusion are also considered for reference.

Table I shows the corresponding SI average PSNR values for the three test sequences. We observe that the proposed SVM-Lin outperforms the other fusion techniques for all three sequences. SVM-Lin is also significantly better than MCTI and DCVP for "Breakdancers" and "Book Arrival". However, DCVP achieves the highest PSNR for "Outdoor".

TABLE I
SIDE INFORMATION PSNR

Method	Breakdancers	Book Arrival	Outdoor
Oracle	30.76	41.84	42.08
MCTI	26.11	35.46	32.69
DCVP	22.88	36.45	**37.81**
PD	26.20	36.09	33.07
MDCDBin	25.55	38.56	35.91
MDCDLin	26.65	38.82	36.26
SVMBin	26.77	38.63	36.04
SVMLin	**27.08**	**38.92**	36.47

4.3. Rate Distortion

We now assess the rate-distortion performance of the proposed SVM-based fusion techniques. For comparison, we use the same approaches as in Sec. 4.2.

The rate-distortion results are shown in Fig. 6 for the three test sequences. For "Breakdancers", SVM-Lin leads to

the best performance among fusion techniques. However, it only achieves small coding gains when compared to MCTI. We can also observe that for this sequence, both MDCD-Bin and MDCD-Lin performs poorly. For "Book Arrival" and "Outdoor", SVM-Lin and MDCD-Lin reach nearly the same performance, and outperform other fusion approaches. They also result in noticeable coding gains when compared to either MCTI or DCVP. For these two sequences, PD fusion performs poorly.

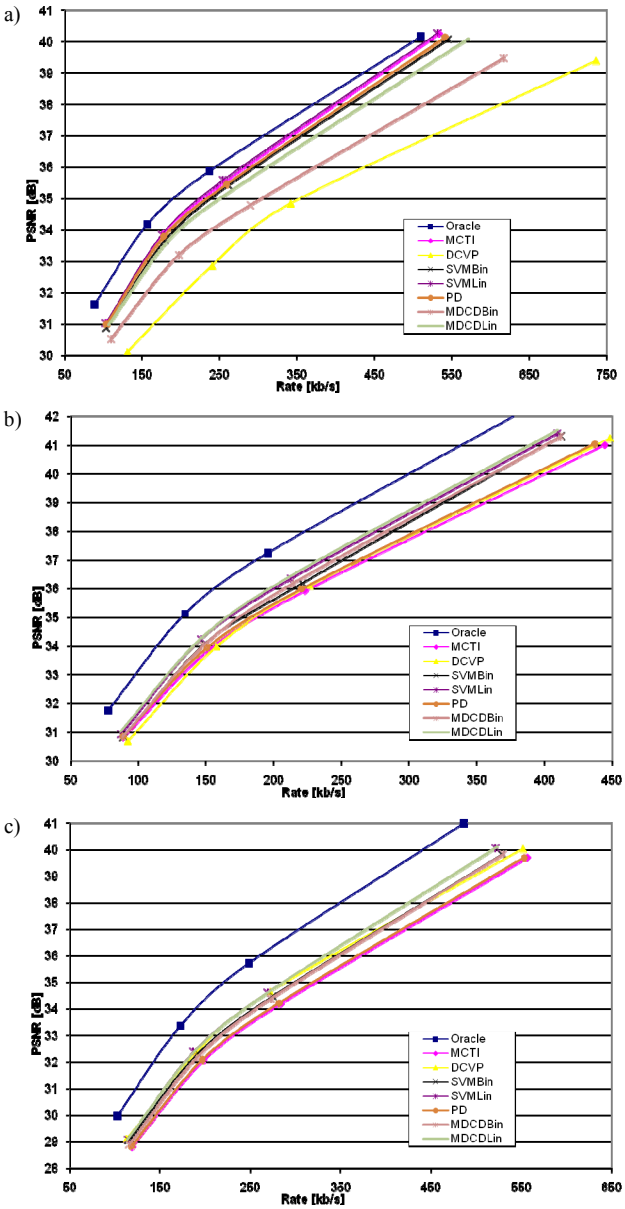

Fig. 6. Rate-distortion performance: a) Breakdancers, b) Book Arrival, c) Outdoor.

5. CONCLUSIONS

In this paper, we considered the problem of SI fusion in multi-view DVC, where SI can be computed either from intra-view temporal interpolation or inter-view interpolation. More specifically we introduced two SVM-based fusion algorithms. The first one, SVM-Bin, simply performs a binary decision based on the SVM classification. The second one, SVM-Lin, performs a linear combination when SVM achieves low classification confidence. Moreover, we presented a number of features which are appropriate for SI fusion. Performance assessment showed that the proposed SVM-Lin fusion algorithm consistently outperformed previously published fusion techniques in terms of SI PSNR. Moreover, SVM-Lin also achieved strong performance in terms of rate-distortion, equivalent or better than previous fusion techniques. Finally, SVM-Lin results in noticeable coding gain when compared to either MCTI or DCVP.

Nevertheless, both proposed SVM-based fusion algorithms still have a significant performance gap when compared to the upper-bound reached by an Oracle decision. Future efforts will concentrate on further improving the approach in order to reduce this gap.

Future work will focus on further improving the proposed SVM-based fusion. First, efforts will aim at optimizing the SVM classifier. Second, we will also consider successive refinements of the SI after the decoding of each bitplane.

6. ACKNOWLEDGMENT

The author would like to acknowledge the use of the DISCOVER codec, a software which started from the IST WZ software developed at the Image Group from Instituto Superior Técnico (IST) of Lisbon by Catarina Brites, João Ascenso and Fernando Pereira. The author would like also to acknowledge the Interactive Visual Media Group at Microsoft Research for the Breakdancers multi-view video sequence [25].

7. REFERENCES

[1] J. Slepian and J. Wolf, "Noiseless Coding of Correlated Information Sources", IEEE Trans. on Information Theory, vol. 19, no. 4, July 1973.

[2] A. Wyner and J. Ziv, "The Rate-Distortion Function for Source Coding with Side Information at the Decoder", IEEE Trans. on Information Theory, vol. 22, no. 1, January 1976.

[3] R. Purit and K. Ramchandran, "PRISM: A new robust video coding architecture based on distributed compression principles", in Proc. Allerton Conference on Communication, Control and Computing, Allerton, IL, USA, October 2002.

[4] A. Aaron, R. Thang, and B. Girod, "Wyner-Ziv coding of motion video", in Proc. Asilomar Conference on Signals and Systems, Pacific Grove, CA, USA, November 2002.

[5] C. Guillemot, F. Pereira, L. Torres, T. Ebrahimi, R. Leonardi and J. Ostermann, "Distributed Monoview and Multiview Video Coding", IEEE Signal Processing Magazine, vol. 24, no. 5, 2007.

[6] F. Dufaux, W. Gao, S. Tubaro, A. Vetro, Distributed Video Coding: Trends and Perspectives, EURASIP Journal on Image and Video Processing, vol. 2009, Article ID 508167, doi:10.1155/2009/508167, 2009.

[7] J. Ascenso, C. Brites, F. Pereira, "Improving Frame Interpolation with Spatial Motion Smoothing for Pixel Domain Distributed Video Coding", in Proc. EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services, Slovak Republic, June-July 2005.

[8] M. Ouaret, F. Dufaux, and T. Ebrahimi, "Multiview Distributed Video Coding with Encoder Driven Fusion", in Proc. European Conference on Signal Processing (EUSIPCO '07), Poznan, Poland, September 2007.

[9] X. Artigas, F. Tarres, and L. Torres, "Comparison of Different Side Information Generation Methods for Multiview Distributed Video Coding", in Proc. International Conference on Signal Processing and Multimedia Applications (SIGMAP '07), Barcelona, Spain, July 2007.

[10] F. Dufaux, M. Ouaret, and T. Ebrahimi, "Recent Advances in Multiview Distributed Video Coding", in Proc. SPIE Mobile Multimedia/Image Processing for Military and Security Applications, vol. 6579, Orlando, FL, April 2007.

[11] E. Martinian, A. Behrens, J. Xin, and A. Vetro, "View Synthesis for Multiview Video Compression", in Proc. Picture Coding Symposium (PCS '06), Beijing, China, April 2006.

[12] S. M. Seitz and C. R. Dyer, "View Morphing", in Proc. 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96), New Orleans, LA, August 1996.

[13] M. Ouaret, F. Dufaux, and T. Ebrahimi, "Fusion-Based Multiview Distributed Video Coding", in Proc. ACM International Workshop on Video Surveillance and Sensor Networks (VSSN'06), Santa Barbara, CA, October 2006.

[14] X. Artigas, E. Angeli, and L. Torres, "Side Information Generation for Multiview Distributed Video Coding Using a Fusion Approach", in Proc. of the 7th Nordic Signal Processing Symposium (NORSIG'06), Reykjavik, Iceland, June 2007.

[15] X. Guo, Y. Lu, F. Wu, W. Gao, and S. Li, "Distributed Multiview Video Coding", in Proc. SPIE Visual Communications and Image Processing (VCIP), vol. 6077, San Jose, CA, January 2006.

[16] X. Guo, Y. Lu, F. Wu, D. Zhao, and W. Gao, "Wyner-Ziv-Based Multiview Video Coding", IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no. 6, pp. 713–724, June 2008.

[17] M. Flierl and B. Girod, "Coding of Multi-View Image Sequences with Video Sensors", in Proc. IEEE International Conference on Image Processing (ICIP '06), Atlanta, GA, USA, October 2006.

[18] T. Maugey, W. Miled, M. Cagnazzo, B. Pesquet-Popescu, "Fusion Schemes for Multiview Distributed Video Coding", in Proc. EUSIPCO, Glasgow, Scotland, Aug. 2009.

[19] T. Maugey, C. Yaacoub, J. Farah, M. Cagnazzo, B. Pesquet-Popescu, "Side information enhancement using an adaptive hash-based genetic algorithm in a Wyner-Ziv context", IEEE MMSP2010, St. Malo, France, Oct. 2010.

[20] M. Ouaret, F. Dufaux and T. Ebrahimi, "Iterative Multiview Side Information for Enhanced Reconstruction In Distributed Video Coding", Eurasip Journal on Image and Video Processing, vol. 2009, Article ID 591915, doi:10.1155/2009/591915, 2009.

[21] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, M. Ouaret, "The Discover Codec: Architecture, Techniques and Evaluation", in Proc. of Picture Coding Symposium, Lisboa, Portugal, November 2007.

[22] V.N. Vapnik, "The Nature of Statistical Learning Theory", Springer, 1995.

[23] http://www.cs.cornell.edu/People/tj/svm_light/

[24] T. Joachims, Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

[25] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," ACM SIGGRAPH and ACM Trans. on Graphics, Los Angeles, CA, Aug. 2004.