

Wyner-Ziv Side Information Generation Using a Higher Order Piecewise Trajectory Temporal Interpolation Algorithm

Mobolaji O. Akinola¹, Laurence S. Dooley², Patrick K. C. Wong³

Department of Communications and Systems, Next Generation Multimedia Technology (XGMT) Group, The Open University, Milton Keynes, United Kingdom

¹m.o.akinola@open.ac.uk; ²l.s.dooley@open.ac.uk; ³k.c.p.wong@open.ac.uk

Abstract—Distributed video coding (DVC) reverses the traditional coding paradigm of complex encoders allied with basic decoding, to one where the computational cost is largely incurred by the decoder. This enables low-cost, resource-poor sensors to be used at the transmitter in various applications including multi-sensor surveillance. A key constraint governing DVC performance is the quality of *side information* (SI), a coarse representation of original video frames which are not available at the decoder. Techniques to generate SI have generally been based on linear temporal interpolation, though these do not always produce satisfactory SI quality especially in sequences exhibiting asymmetric (non-linear) motion. This paper presents a *higher-order piecewise trajectory temporal interpolation* (HOPTTI) algorithm for SI generation that quantitatively and perceptually affords better SI quality in comparison to existing temporal interpolation-based approaches.

Keywords—Distributed Video Coding (DVC); Side Information (SI); Wyner-Ziv; Trajectory; Interpolation.

I. INTRODUCTION

Distributed video coding (DVC) is the practical implementation of the lossy Wyner-Ziv (WZ) [1] and lossless Slepian-Wolf (SW) [2] theorems and represents a paradigm shift from conventional coding standards by allowing the deployment of simple, cheap, resource-constrained encoders in portable mobile devices [3]. The main focus of DVC research has to date, been to reduce the acknowledged performance gap between conventional *coder-decoders* (codec) like H.264/AVC [3-5]. A key DVC bottleneck is the quality of *side information* (SI) [4-7] which is a coarse approximation of the original video frames that are not available at the decoder. SI impacts in two ways upon DVC performance: i) *rate distortion* (RD) - the number of decoder bits required to provide a prescribed output quality; and ii) robustness (error resilience) – SI frames are constructed independent of channel fidelity at the decoder, so the better the SI quality, the more resilient the codec becomes compared to the situation where more bits are needed via error prone channels. Traditionally, *linear motion compensated temporal interpolation* (LMCTI) has been used in SI generation [3-5], with recently proposed enhancements including hierarchical temporal interpolation [6] and spatially-aided SI generation [7]. While LMCTI provides reasonable SI quality for sequences with slow-to-medium object motion, it tends not to generally be so successful for sequences exhibiting non-linear motion [7]. Higher-order trajectories [8, 9] for temporal variations have been modeled, leading to more

accurate sequence reconstructions, though at the pyrrhic cost of greater computational complexity. Modeling pixel intensity in a 2-D plane known as apparent motion (optical flow) [9], assumes the temporal correlations are highest in the direction of object motion therefore a more accurate model of object motion is required to fully exploit this correlation. In optical flow, higher-order trajectories and *motion vector* (MV) sampling at least one MV per pixel, give a better representation of object motion and reduce temporal redundancies [9]. Due to the high computational cost incurred, this requirement can be relaxed in terms of trajectory and MV sampling, which is the approach often adopted in DVC [3-5], where a linear trajectory model together with MV sampling at less than one per image pixel is assumed.

This paper presents a block-based, *higher-order piecewise trajectory temporal interpolation* (HOPTTI) algorithm for SI generation which is based upon the models in [8, 9]. Instead of tracking every pixel however, 4 x 4 pixel blocks are tracked so a block-based motion field is sampled using block-based MV estimation to reduce the computational overheads in terms of the number of MV and the search method. For instance, the *adaptive rood pattern search* (ARPS) algorithm [10] used by HOPTTI reduces the number of MV search points to about 9 compared with 255 for a full search. By using a higher-order motion trajectory models, HOPTTI provides superior SI quality in terms of the average *peak signal-to-noise ratio* (PSNR) compared with existing LMCTI techniques, with for certain sequences, an improvement of over 8dB achieved when using cubic polynomials instead of a linear model.

The remainder of the paper is organized as follows: Section II introduces the HOPTTI algorithm, while Section III presents a numerical and qualitative analysis of its SI generation performance. Section IV draws some conclusions.

II. HIGHER ORDER PIECEWISE TRAJECTORY TEMPORAL INTERPOLATION (HOPTTI) ALGORITHM

LMCTI may not always generate the requisite SI quality because of the inherent non-linear nature of certain object and global motion. In [8] and [9] for instance, asymmetric object motion was addressed by a quadratic trajectory model allied with MV sampling of one MV per pixel for temporal interpolation, which gave an overall improvement of up to 4dB over conventional linear-based models.

This was the motivation to investigate in a DVC context, a similar approach for SI generation, by employing higher-order polynomial trajectory models for temporal interpolation. Results for the HOPTTI algorithm confirm a consistent improvement in SI performance of at least 4dB when cubic models are employed, with full details of the results being presented in Section III.

A. The Piecewise Trajectory Formulation and Parametization

The SI generation technique which is the basis of HOPTTI adopts the motion compensated temporal interpolation concept in [8] and [9], except instead of a constant acceleration trajectory model comprising quadratic functions, it is extended to a variable acceleration (3rd order) paradigm. This enables objects exhibiting sudden accelerated motion, such as a surge (also popularly referred to as *jolt*) to be more accurately represented.

To illustrate the idea example segments of the motion trajectory of an object in 3-D x, y, t space between time t_1 and t_4 is shown in Figure 1. It is assumed the displacements (MV) of the blocks relating to the object at key frames K_1, K_2, K_3 and K_4 between t_1 and t_4 are respectively A_i, B_i, C_i and D_i . In HOPTTI, the MV of a block is evaluated by finding the position of best match in the next key frame. All key frames are available at the decoder, while WZ frames (denoted as *SI* in Figure 1) are produced using motion compensated temporal interpolation.

The motion trajectory $C(t)$ of an object can be represented by a set of piecewise cubic polynomials:

$$C(t) = \begin{cases} p_1(t) & \text{for } t_1 \leq t \leq t_4 \\ p_2(t) & \text{for } t_4 \leq t \leq t_8 \\ \vdots & \\ p_n(t) & \text{for } t_n \leq t \leq t_{n+3} \end{cases} \quad (1)$$

where each segment of the trajectory $p_i(t)$ is represented by an equation of motion considering a constant jolt given by:

$$p_i(t) = \frac{1}{6} j_i (t - t_i)^3 + \frac{1}{2} a_i (t - t_i)^2 + v_i (t - t_i) + d_i \quad (2)$$

For $i = 1, 2, \dots, n$. In (1), n is the number of available key frames, j_i is the average *jolt* (the rate of change of acceleration), a_i the average acceleration, v_i the average velocity between t_i and t_{i+1} and d_i the initial displacement at t_i .

To calculate the four parameters j_i, a_i, v_i and d_i , a minimum of 4 key frames are required, and if it is assumed the respective displacements of the blocks at these key frames are A_i, B_i, C_i and D_i , then the following holds:

$$d_i = A_i \quad (3)$$

$$v_i = \frac{B_i - A_i}{T} \quad (4)$$

$$a_i = \frac{v_{i+1} - v_i}{2T} = \frac{C_i - 2B_i + A_i}{2T^2} \quad (5)$$

$$j_i = \frac{a_{i+1} - a_i}{3T} = \frac{D_i - 3C_i + 3B_i + A_i}{6T^3} \quad (6)$$

where T is the time between two consecutive key-frames, $A_{i+1} = B_i, B_{i+1} = C_i, C_{i+1} = D_i$.

As the motion trajectory of an object can be evaluated using (1) – (6), this enables the MV of the object at any time between t_1 and t_{n+1} to be accurately interpolated. The backward motion trajectory is evaluated the same way as the forward one using (1) – (6) as described but in backward direction i.e. D_i, C_i, B_i and A_i .

B. The HOPTTI Algorithm

The HOPTTI framework is displayed in Figure 2 with the individual blocks now explained.

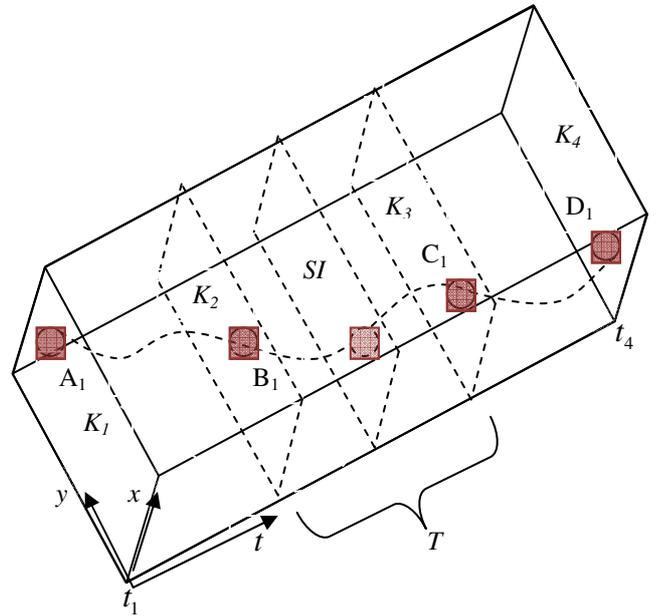


Figure 1. Example segments of the motion trajectory of an object in 3-D space between time t_1 and t_4 , where K are the key frames and *SI* is the side information of the Wyner-Ziv frame.

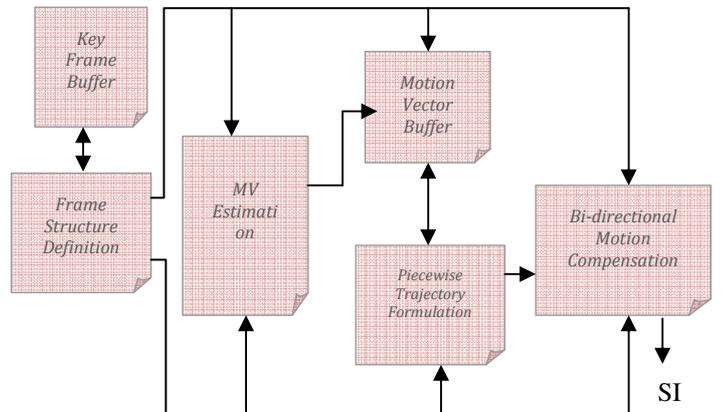


Figure 2. Block diagram of the HOPTTI algorithm

1) Frame structure definition:

The structure is as follows:

1. Estimate the MVs (A, B and C and D as shown in Figure 1) for both forward and backward path using ARPS [10].
2. Calculate the inter-frame distance between two successive key frames to obtain a normalized temporal distance of unity. A fractional weight ζ is introduced to locate the temporal SI position (the missing WZ frames) that gives the highest PSNR. For LMCTI $\zeta=0.5$ [4] while for higher-order models the value of ζ must take cognizance of the fact a MV may not necessarily intersect at the centre of a block.
3. Using the estimated parameters from Step 1, calculate the cubic polynomial motion trajectory; the SI using bi-directional motion compensation and the weight ζ to generate the final interpolated frame (see Figure 4).
4. Repeat Steps 1 to 3 for all SI frames.

2) *MV estimation*: uses ARPS [10] since it exploits the fact motion is generally coherent, so the MV of an adjacent block can predict the direction of the current block. A 4 x 4 pixel block is employed for ARPS in HOPTTI.

3) *Piecewise trajectory formulation and bi-directional motion compensation*: The HOPTTI formulation uses a trajectory formulation analogous to [8, 9]. The block-based motion estimation scheme does not capture all aspects of the motion field therefore the higher order piecewise trajectory and bi-directional motion estimation and compensation reduces the impact of this by refining the MV in a way similar to B-frames in conventional video coding. In this situation however, the interpolated block is not known and the corresponding residue is unavailable so a different refinement strategy following the higher order trajectory is applied. In addition, MV estimation for uncovered areas (corresponding to holes) are estimated from previous frames in both the forward and backward directions resulting in forward and backward interpolated frames. In bi-directional motion estimation and compensation in LMCTI schemes, there are

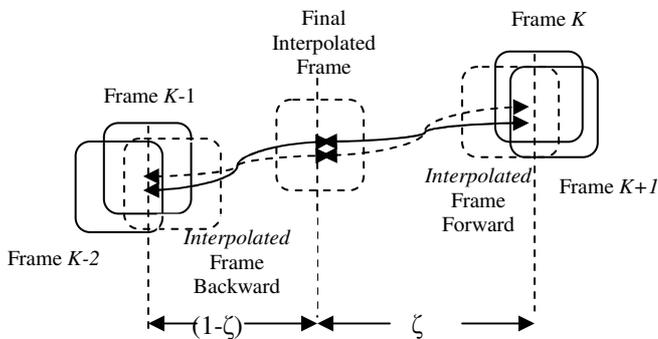


Figure 3. Bidirectional motion compensation with cubic trajectory and MV sampling estimation at decoder. Fractional weight ζ locates final SI frame.

two methods used to generate the final interpolated frame, namely: i) spatial hole tracking and filling as in [5], which involves pixel-wise searching of both forward and backward frames which is computationally very expensive, and ii) a temporal MV scheme where the forward and backward MV interpolated frames are averaged together. The second approach has been adopted most often due to its simplicity and lower computational cost. However, due to irregular object motion in real world video, the scheme which assumes linearity and regular object motion is not sufficient. HOPTTI introduces a weighting factor ζ to adjust the respective contributions of the forward and backward MVs to generate the interpolated frame, as illustrated in Figure 3. The best ζ is empirically determined to provide the highest average PSNR for each sequence. Figure 3 shows the use of the weight ζ to obtain the final interpolation frame from the forward and backward frames, while at the same time illustrating how the MV of block from 4 key frames is used to obtain the piecewise cubic motion trajectory.

III. EXPERIMENTAL RESULTS

The HOPTTI algorithm was implemented in Matlab version 7.5.0 (R2007b) running under Microsoft Windows XP on a PC with an Intel Duo Core CPU at 2.20 GHz. A *group-of-picture* size of 2 was chosen for all the experiments i.e. *KWKWKW*, where *K* and *W* denote key and WZ frames respectively. The cubic trajectory used in HOPTTI which is one order higher than the implementation in [8] and uses polynomial trajectory parameterization as described in Section IIA.

To numerically and qualitatively evaluate HOPTTI, various QCIF (Quarter Common Intermediate Format) test sequences were applied including *Carphone*, *Mother*, *Coastguard*, *Silent*, *Hall*, and *Foreman*, which provided a range of different types of motion and objects. Two interpolation-based SI generation approaches were used for comparison, namely the 3-D content *adaptive recursive search* (3DCARS) [11] which employs a quarter-pixel MV search implying 16 more MV searches than HOPTTI and the *pixel-domain WZ* (PD-WZ) codec [12]. Both these SI generation techniques use linear interpolation allied with various temporal and spatial refinements respectively.

Table I summarises the HOPTTI algorithm results for linear, quadratic and cubic-order trajectories for various sequences. The results confirm consistently superior SI quality is achieved when a cubic polynomial trajectory model is applied to the various sequences, with for instance *Foreman* providing an average improvement of up to 5dB and *Coastguard* an 8dB improvement compared with the linear HOPTTI model. Interestingly, Table I reveals the HOPTTI algorithm exhibits progressive SI quality improvement for increasing polynomial trajectory order, i.e., quadratic over linear and cubic over quadratic. Pragmatically however, these SI improvements are counterbalanced by a higher complexity overhead, which was the reason that even higher-order polynomial trajectories were not considered.

TABLE I. SI AVERAGE PSNR PERFORMANCE COMPARISON IN dB FOR VARIOUS TRAJECTORY ORDERS FOR HOPTTI

Sequences	HOPTTI Linear	HOPTTI Quadratic	HOPTTI Cubic
Carphone	30.9	34.0	35.3
Coastguard	28.4	34.3	36.4
Foreman	29.9	33.0	35.1
Mother	36.2	44.4	47.3
Hall	30.5	36.2	38.5
Silent	31.7	37.2	38.9

Table II shows the comparative results for HOPTTI with 3DCARS [11], and PD-WZ [12]. This reveals for example, for *Mother*, HOPTTI gave better SI quality with an improvement on average of 2.9dB and 9.0dB respectively compared with 3DCARS and PD-WZ. This is particularly noteworthy when noted the enhancements introduced in both [11] and [12]. 3DCARS for example used quarter-pixel interpolation accuracy compared with the integer accuracy for HOPTTI. The HOPTTI algorithm also provided better SI quality of more than 2dB over 3DCARS for *Coastguard*, though this was counterbalanced by its performance being less satisfactory compared to PD-WZ [11]. This was due to the influence of the significant temporal noise components produced by the water in this sequence.

Figure 4 shows the perceptual SI quality of two sample frames for *Hall*, where HOPTTI performed better in comparison with the other SI generation schemes, with Table II confirming the numerically PSNR improvements are 1.1dB and 1.7dB respectively. Figure 5 plots the corresponding frame-wise SI curve for the complete (150 frames) of the *Hall* sequence. In terms of the RD curves, Figure 6 reveals that while H.264 Inter remained the upper bound, HOPTTI provided better RD performance (up to 5 dB better than H.264 No Motion) for *Hall* over other established DVC codecs including: H.264 Intra, H.264 No Motion and DISCOVER [4]. This vindicates the rationale of employing high-order piecewise polynomial trajectories for temporal interpolation to improve the quality of SI.

TABLE II. COMPARISON OF AVERAGE PSNR IN dB PERFORMANCE FOR VARIOUS SI INTERPOLATION TECHNIQUES

Sequences	3DCARS [11]	PD-WZ [12]	HOPTTI
Carphone	34.9	-	35.3
Mother	44.4	38.3	47.3
Foreman	34.9	33.0	35.1
Silent	36.4	-	38.9
Coastguard	37.5	34.2	36.4
Hall	37.4	36.8	38.5

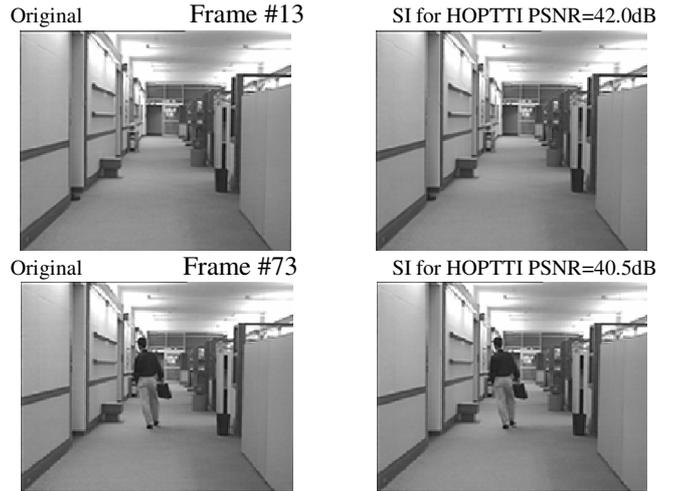


Figure 4. Sample frames for the *Hall* sequence showing the SI quality obtained using HOPTTI algorithm

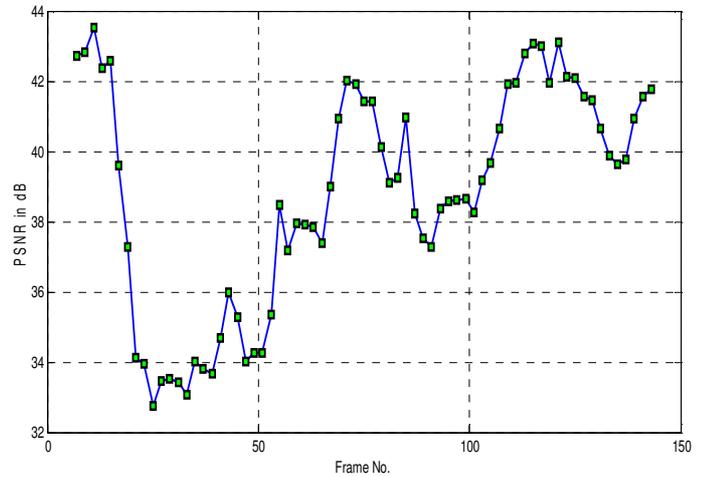


Figure 5. Frame-wise plot showing SI quality of HOPTTI algorithm for *Hall* sequence

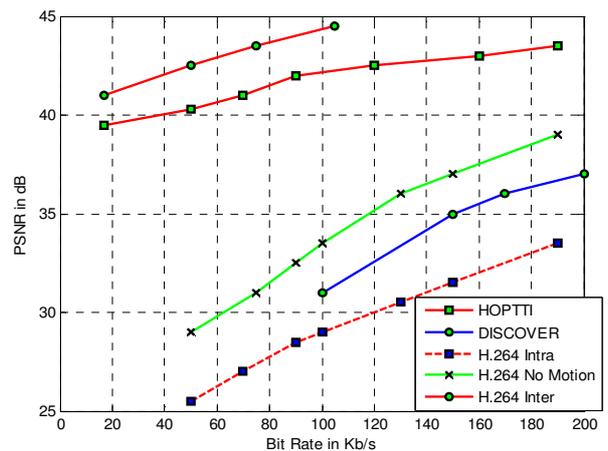


Figure 6. RD Curves showing HOPTTI PSNR performance in codec based on [5] for *Hall* sequence @ 15fps

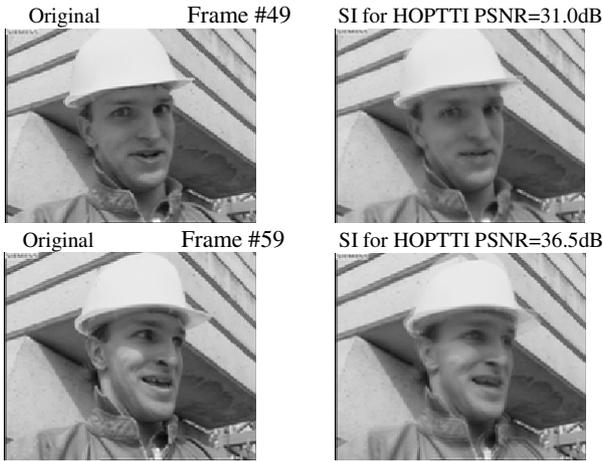


Figure 7. Sample frames for the *Foreman* sequence showing the SI quality obtained using HOPTTI algorithm

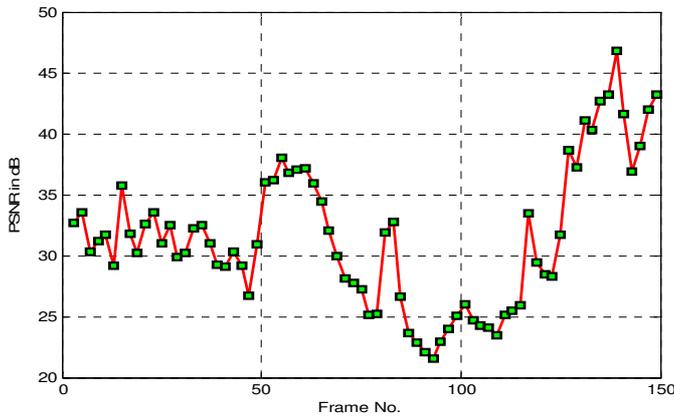


Figure 8. Frame-wise plot showing SI quality of HOPTTI algorithm for *Foreman* sequence

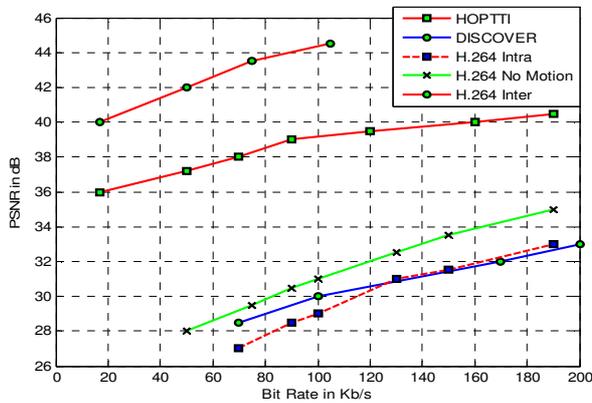


Figure 9. RD Curves showing HOPTTI PSNR performance in codec based on [5] for *Foreman* sequence @ 15fps

Figure 7 shows the perceptual quality of sample frames for *Foreman*, where the gain in using HOPTTI compared to [11] for instance is more than 2dB, while in Figure 8 the frame-

wise SI performance is plotted. As far as the corresponding RD curves shown in Figure 9, these again confirm HOPTTI outperformed (at least 5 dB compared to H.264 No Motion) the other conventional DVC codecs, with again H.264 Inter being the upper RD bound.

CONCLUSION

This paper presents a novel method of generating side information (SI) in Wyner-Ziv coding using a *higher-order piecewise trajectory temporal interpolation* (HOPTTI) algorithm. Both numerical and qualitative results confirm that HOPTTI consistently provided superior SI quality compared to a number of existing interpolation techniques, especially for sequences which exhibited non-linear object motion.

REFERENCES

- [1] D. Slepian, and J. Wolf., "Noiseless coding of correlated information sources," in *IEEE Transactions on Information Theory*, vol. 19, pp.471-480, 1973.
- [2] A. D. Wyner, and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," in *IEEE Transaction on Information Theory*, vol. 22, pp.1-10, 1973.
- [3] F. Pereira, L. Torres, C. Guillemot, T. Ebrahimi, R. Leonardi, and S. Klomp, "Distributed video coding: selecting the most promising application scenarios," in *Elsevier, Signal Processing: Image Communication*, vol. 23, pp. 339-352, 2008.
- [4] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Oualet, "The DISCOVER codec: architecture, techniques and evaluation," in *Proc. of Picture Coding Symposium*, vol 6, pp 14496-14410, Lisbon, Portugal, 2007.
- [5] X. Li, X., "On the importance of source classification in Wyner-Ziv video coding," in *SPIE conf. on Visual Communications and Image Processing*, Jan. Vol. 5308, pp. 520-528, 2008.
- [6] R. Liu, Z. Yue, and C. Chen, "Side information generation based on hierarchical motion estimation in distributed video coding," in *Chinese Journal of Aeronautics*, Volume 22, Issue 2, pp. 167-173, 2009.
- [7] S. Ye., M. Oualet., F. Dufaux, and T. Ebrahimi, "Improved side information generation for distributed video coding by exploiting spatial and temporal correlations," in *EURASIP journal on Image and Video Processing*, Vol. 2009, Article ID 683510, 15 pages., 2009.
- [8] M. Chahine, and J. Konrad, "Estimation and compensation of accelerated motion for temporal sequence interpolation," in *Signal Processing. Image Communication*, vol. 7, pp. 503--527, Nov. 1995.
- [9] M. Chahine, "Motion compensation interpolation using trajectories with acceleration," in *IS&T/SPIE Symposium on Electronic Imaging Digital Video Compression*, San Jose, CA, USA vol.2419, Feb. 1995.
- [10] H. Zhao, X. Yu, J. Sun, C. Sun, and H. Cong, "An enhanced adaptive rood pattern search algorithm for fast block-matching motion estimation", in *Congress on Image and Signal Processing*, Vol. 1, pp.416-420, 2008
- [11] S. Borchert, R. P. Westerlaken, R. K. Gunnewiek, and R. L. Lagendijk, "On extrapolating side information in distributed video coding," in *26th Picture Coding System*, Lisbon, Portugal, November 2007.
- [12] M Tagliasacchi, A. Trapanese, S. Tubaro, J. Ascenso, C. Brites, F. Pereira, "Exploiting spatial redundancy in pixel domain Wyner-Ziv video coding," in *IEEE International Conference on Image Processing*, Atlanta, USA, October 2006.