

Document downloaded from:

<http://hdl.handle.net/10251/55942>

This paper must be cited as:

Prades Nebot, J. (2014). On the Efficiency of Angular Intraprediction. IEEE Transactions on Image Processing. 23(12):5733-5742. doi:10.1109/TIP.2014.2369954.



The final publication is available at

<http://dx.doi.org/10.1109/TIP.2014.2369954>

Copyright Institute of Electrical and Electronics Engineers (IEEE)

Additional Information

# On the efficiency of Angular Intra Prediction

José Prades Nebot, *Member, IEEE*.

**Abstract**—Angular Intra Prediction (AIP) is a coding tool that has been incorporated into the video coding standards H.264/AVC and HEVC. In this paper, we study how the efficiency of AIP depends on its prediction parameters. To carry out this study, we first theoretically analyze the variance of the error incurred when a perfectly directional signal is predicted in a certain direction. The results of this analysis are then used to study the efficiency of AIP when it is applied to a distribution of directions. To facilitate mathematical derivations, we make several assumptions about the signal and the prediction process, and we use some approximations. This allows us to obtain simple expressions for the variance of the AIP prediction error as a function of signal and prediction parameters. Finally, we compare our theoretical results with the results obtained from the prediction of images containing rectilinear edges. This comparison shows that our theoretical expressions follow the main trends of the experimental results except when AIP is performed with a very high accuracy.

EDICS: COM-LOC

## I. INTRODUCTION

Video coding algorithms are crucial in today's visual communication since they reduce the huge bit rate of raw digital video [1]–[3]. Most video coding algorithms use a block-based hybrid coding approach. In this approach, each video frame is partitioned into blocks which are encoded using inter-frame coding (inter-blocks) or intra-frame coding (intra-blocks). Since motion-compensated prediction cannot be used in the encoding of intra-blocks, these are less efficiently encoded than inter-blocks.

To achieve an efficient coding of intra-blocks, some video coding algorithms use (spatially-based) intra prediction. In this technique, the pixels of a block are predicted using previously encoded and reconstructed neighboring pixels of that block. Then, the residual block is computed and subsequently encoded using transform coding. In order to adapt to the different structures that are present in images, a set of different predictors or *intra-prediction modes* are available, and, for each block, the prediction is switched to the mode that provides the best coding efficiency. Since images often contain locally rectilinear structures, some modes are chosen to efficiently predict structures of this type. These modes predict a block by propagating neighboring pixels inside the block in a certain direction. The prediction provided by these *directional modes* is called *angular intra prediction* (AIP) [4].

AIP was introduced in the video coding standard H.264/AVC [1]. AIP contributed significantly to improving the efficiency of H.264/AVC intra-frame coding with respect to previous standards [5]. For this reason, AIP was also used in subsequent standards such as the Audio-Video coding

Standards (AVS) [2] and the High Efficient Video Coding (HEVC) standard [3]. The use of AIP in these standards has motivated us to study the prediction efficiency of this coding tool.

The accuracy of AIP in the prediction of directional structures is determined by several factors such as the number of directional modes and the block size. In this paper, we study how these accuracy factors influence the prediction efficiency of AIP. To perform the study, we model images with one-dimensional random sequences that are translated in a certain direction. These pure directional images can be perfectly predicted using AIP with infinite accuracy. In this paper, we study the error introduced when practical AIP (i.e., AIP with finite precision) is used to predict these signals.

We first consider the simple case in which a directional image is predicted using an arbitrary direction. Using this simple set-up, we show how the variance of the prediction error depends on several signal and prediction parameters. The results obtained allow us to extend the study to the variance of the prediction error when images with a distribution of directions are predicted using AIP. By performing several approximations and simplifications, we derive expressions that show how prediction parameters influence AIP efficiency. Finally, we compare our theoretical results with experimental results obtained by predicting images with rectilinear edges. We show that our theoretical expressions follow the main trends of the experimental results except when AIP is performed with a very high accuracy. Consequently, our expressions can help in the design of AIP algorithms for future video coders.

The rest of the paper is organized as follows. In Section II, we describe the AIP algorithm. In Section III, we study the prediction error when a single directionality is predicted in a certain angle. The results of this section are used in Section IV to analyze the variance of prediction error in AIP. In Section V, we compare our theoretical analysis with experimental results obtained using synthetic images and comment on their similarities and differences. Finally, Section VI concludes the paper.

*Notation and definitions:* Independent variables are enclosed with parentheses in continuous signals (e.g.,  $s(x, y)$ ) and with brackets in discrete signals (e.g.,  $s[n, m]$ ). Although we deal with 2D signals, prediction often involves 1D processing in only one variable. To highlight this fact, for a given  $m_0$  ( $n_0$ ), the signal  $s_{m_0}[n]$  ( $s_{n_0}[m]$ ) refers to the 1D signal  $s[m_0, n]$  ( $s[m, n_0]$ ). The autocorrelation of a real deterministic sequence  $z[n]$  is  $r_{zz}[k] = \sum_{n=-\infty}^{\infty} z[n+k]z[n]$ . If  $h[n]$  is the impulsive response of a filter, its  $l$ th polyphase component (with respect to  $M$ ) is  $h_l[n] = h[nM + l]$  ( $0 \leq l < M$ ). We denote probability, expectation, and variance, as  $\mathbb{P}\{\cdot\}$ ,  $\mathbb{E}\{\cdot\}$ , and  $\text{Var}\{\cdot\}$ , respectively.

J. Prades Nebot is with the Institute of Telecommunications and Multimedia Applications, Universitat Politècnica de València, Valencia 46022, Spain (e-mail: jprades@dcom.upv.es).

## II. AIP

In this section, we describe the fundamentals of AIP. In AIP, there is a set of directional modes each of which has a prediction angle  $\tilde{\theta}$  (see Fig. 1(a)). For each block, the mode that minimizes a rate-distortion cost function is selected for prediction. This *switched prediction* adapts to the directional features of each block. The larger the number of directional modes, the more accurate the prediction (but the larger the complexity of the search for the optimal mode). Apart from the directional modes of AIP, intra-prediction algorithms also include some non-directional modes for blocks without a defined directionality (e.g., the DC and the Planar modes of H.264/AVC and HEVC [1], [4]).

Let  $s[n, m]$  be a monochrome digital image. Let us consider a block of  $B \times B$  pixels of  $s[n, m]$  that, without loss of generality, is formed by the pixels  $\{(n, m): 1 \leq n, m \leq B\}$  (see Fig. 1(b)). The directional prediction of a block is built using previously coded and reconstructed pixels that belong to the two 1D *reference sequences* of that block: its reference row and its reference column. The *reference row* is the set of pixels with coordinates  $\{(n, 0): 0 \leq n \leq 2B\}$ , while the *reference column* is the set of pixels with coordinates  $\{(0, m): 0 \leq m \leq 2B\}$ . Note that  $s[0, 0]$  belongs to both reference sequences.

All or part of the pixels of a reference sequence may not be available for prediction. Thus, blocks placed at the borders of the frame may partially or completely lack one or both reference sequences. Additionally, some pixels of the reference row or column of a block may not have been encoded yet, which prevents their use for closed-loop prediction. Thus, in H.264/AVC, the pixels  $s[0, m]$  with  $B + 1 \leq m \leq 2B$  are never used as reference since they are rarely available [1].

To derive the prediction with a certain angle  $\tilde{\theta}$  for a pixel, the pixel position is projected in that angle towards one of the two reference sequences. The reference sequence that is chosen depends on  $\tilde{\theta}$ . The pixel is projected towards the reference row when  $\tilde{\theta} \in [-\frac{\pi}{4}, 0]$  and towards the reference column when  $\tilde{\theta} \in [\frac{\pi}{2}, \frac{3\pi}{4}]$ . When  $\tilde{\theta} \in [0, \frac{\pi}{2}]$ , the projection of a pixel position only crosses one of the two sequences, and that sequence is used for that pixel (see Fig. 1(b)). If the projection points to a pixel of the chosen reference sequence, then that pixel is used as the prediction. Otherwise, the value is interpolated (linear interpolation is normally used [1], [4]).

## III. THE ERROR IN DIRECTIONAL PREDICTION

In this section, we study the error  $e[n, m]$  incurred when a real random sequence  $s[n, m]$  that is perfectly directional with an angle  $\theta$  is directionally predicted with an angle  $\tilde{\theta}$ . To measure the efficiency of this *directional prediction* in each pixel, we consider the variance of  $e[n, m]$ ,  $d_{n, m} = \text{Var}\{e[n, m]\}$ ; to globally measure the prediction efficiency in a block of  $B \times B$  pixels, we consider the average of  $d_{n, m}$  over that block

$$d = \frac{1}{B^2} \sum_{n=1}^B \sum_{m=1}^B d_{n, m}. \quad (1)$$

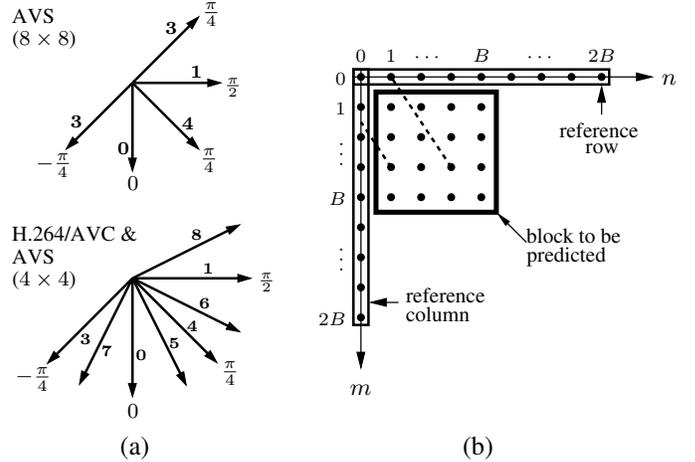


Fig. 1. AIP. (a) Prediction angles and directions for luma blocks in H.264/AVC and AVS. (b) Reference sequences of a block ( $B = 4$ ).

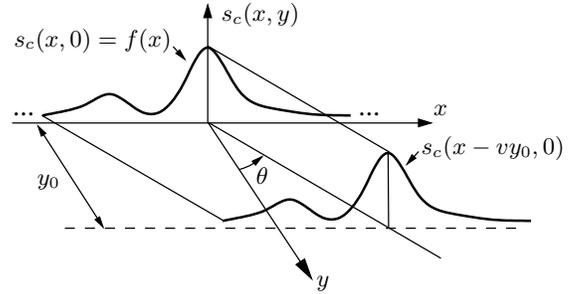


Fig. 2. Projection of  $s(x, 0)$  in a certain direction  $v = \tan \theta$ .

### A. Image Model

Let  $f(x)$  be a continuous and bandlimited wide-sense stationary (WSS) random process. Consider a 2D continuous signal  $s(x, y)$  that is obtained by displacing or *projecting*  $f(x)$  with a certain angle  $\theta$  (see Fig. 2) in such a way that

$$s(x, y) = f(x - vy) \quad (2)$$

where  $v = \tan \theta$ .

Now, suppose we sample  $s(x, y)$  and that for simplicity the sampling periods are  $T_x = T_y = 1$ . The resulting sequence  $s[n, m]$  fulfills

$$s[n, m] = s(n - mv, 0) = f(n - mv). \quad (3)$$

We refer to a sequence that fulfills (3) for some  $f(x)$  as a *directional sequence*, and we refer to  $\theta$  and  $v$  as its *angle* and its *direction*, respectively.

Let us assume that the sampling of  $s(x, y)$  was performed without aliasing and that we wish to predict any row  $s_m[n]$  (with  $m > 1$ ) from the infinite-length reference row  $s_0[n]$  ( $s_0[n] = s[0, n]$ ). In the frequency domain,  $s_m[n]$  and  $s_0[n]$  are related through [6]

$$S_m(e^{j\omega}) = e^{-jmv\omega} S_0(e^{j\omega}) \quad (4)$$

which in the time domain provides

$$s_m[n] = s_0[n] * \text{sinc}(n - mv). \quad (5)$$

In other words, each row  $s_m[n]$  can be predicted without error by filtering  $s_0[n]$  with a filter that implements a displacement of  $mv$  samples.

From (2), we can also write  $s(x, y) = s(0, y - x/v)$  and, after the sampling,  $s_n[m] = s(0, m - n/v)$ . Consequently,

$$s_n[m] = s_0[m] * \text{sinc}(m - n/v) \quad (6)$$

i.e., each column  $s_n[m]$  can be obtained by filtering  $s_0[m]$  with a sinc filter. Therefore, we can perfectly predict  $s[n, m]$  using either of the two reference sequences. In practice, however, it is preferable to use the reference sequence that involves a smaller displacement:  $s_0[n]$  if  $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$  and  $s_0[m]$  if  $\theta \in (\frac{\pi}{4}, \frac{3\pi}{4}]$ . In the rest of Section III, we assume that  $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$  and that  $s_0[n]$  is used for prediction. Nevertheless, the results obtained can be easily extended by symmetry to the case in which  $\theta \in (\frac{\pi}{4}, \frac{3\pi}{4}]$  and  $s_0[m]$  is used.

A practical implementation of (5) requires estimating  $v$  and approximating the sinc function with a finite-length filter. Hence, a prediction error is generally introduced. Once the statistics of  $f(x)$  are set, the prediction error *only* depends on how accurately we estimate  $v$  and approximate the sinc filter.<sup>1</sup> Hence, our signal model allows us to study how the accuracy of directional prediction influences performance. This approach is similar to the framework used in the works of [7], [8] where the efficiency of motion-compensated prediction in hybrid video coding is analyzed. In these works, video frames are modeled using a stationary random field  $s(x, y)$  and the only difference between consecutive frames is a translatory displacement.

In [9]–[12], the statistics of intra-prediction residuals are analyzed in order to derive optimal transforms [9]–[11] or to perform image partitions that improve the prediction [12]. These works disregard the implementation aspects of directional prediction and focus on the error introduced in the prediction of a pure 2D random sequence. In our work, the image model allows a theoretically exact prediction to be made and the prediction error is caused by an inaccurate implementation of (5).

### B. Practical directional prediction

We can approximate the filtering in (5) using the scheme shown in Fig. 3, which implements a fractional displacement of  $mN/M$  samples [13]. This scheme facilitates the analysis of the prediction error; however, practical directional prediction is implemented in a simpler way [4]. In Fig. 3,  $N$  and  $M$  are two coprime integers with  $M > 0$  and  $|N| \leq M$ , and  $h[n]$  is an interpolation filter of factor  $M$ . This scheme first interpolates  $s_0[n]$  by a factor  $M$  so that the required displacement can be implemented with an integer displacement ( $z^{-mN}$ ); the final downsampling by  $M$  restores the initial sampling frequency. Note that  $h[n]$  is fixed and that adapting the prediction to each  $m$  only requires changing the integer displacement  $z^{-mN}$ .

<sup>1</sup>If sampling introduces aliasing in  $s_0[n]$ ,  $s_m[n]$  cannot be generally recovered from  $s_0[n]$  using (5). In this case, we should also consider the error due to aliasing.

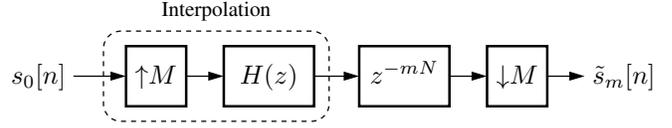


Fig. 3. Practical directional prediction of  $s_m[n]$  in a direction  $\tilde{v} = N/M$ .

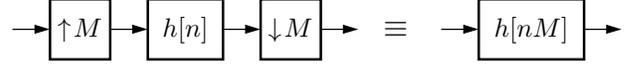


Fig. 4. Multirate identity.

Using the multirate identity shown in Fig. 4 (see [14], pp. 133), the prediction for the  $m$ th row  $\tilde{s}_m[n]$  can be expressed as

$$\tilde{s}_m[n] = s_0[n] * h[nM - mN]. \quad (7)$$

Let  $\tilde{v} = N/M$  and  $h_I[n] = \text{sinc}(n/M)$ . If  $h[n] = h_I[n]$  and  $\tilde{v} = v$ , then

$$h[nM - mN] = \text{sinc}(n - mv), \quad (8)$$

and hence  $\tilde{s}_m[n] = s_m[n]$  (i.e.,  $s_m[n]$  is predicted without error). When  $\tilde{v} \neq v$  or  $h[n] \neq h_I[n]$ , the prediction incurs an error  $e_m[n] = s_m[n] - \tilde{s}_m[n]$ . From (5) and (7), we have

$$e_m[n] = s_0[n] * (\text{sinc}(n - mv) - h[nM - mN]). \quad (9)$$

To gain insight into the causes of error in directional prediction, we express  $e_m[n]$  as the sum of two components:

$$e_m[n] = e_m^p[n] + e_m^i[n], \quad (10)$$

where  $e_m^p[n]$ , which is called the *projection error*, is

$$e_m^p[n] = s_0[n] * (\text{sinc}(n - mv) - \text{sinc}(n - m\tilde{v})) \quad (11)$$

and  $e_m^i[n]$ , which is called the *interpolation error*, is

$$e_m^i[n] = s_0[n] * (\text{sinc}(n - m\tilde{v}) - h[nM - mN]). \quad (12)$$

The projection error appears when  $\tilde{v} \neq v$  causing  $s_0[n]$  to be projected in direction  $\tilde{v}$  instead of in direction  $v$ . The interpolation error appears when  $h[n] \neq h_I[n]$  and it introduces distortion in the interpolation stage (see Fig. 3).

In the following sections, we study the variance of each of these errors. We will focus on the case where  $s_0[n]$  is AR(1) with a first-step correlation coefficient  $\rho$  that is close to 1, since this type of process has been widely used in the modeling of rows and columns of digital images [15].

### C. Variance of the projection error

When  $h[n] = h_I[n]$  and  $\tilde{v} \neq v$ , only the projection error is introduced ( $e_m[n] = e_m^p[n]$ ). This situation is only possible if  $\tilde{v} \in \{-1, 0, 1\}$ ; otherwise  $h_I[n]$  has infinite length. Let  $s_0[n]$  be a WSS process with variance  $\sigma_s^2$  and autocovariance  $C_{ss}[k]$ . For each  $m$ ,  $e_m^p[n]$  is a WSS process with zero mean and variance (see Appendix A)

$$d_m^p = 2\sigma_s^2 \left( 1 - \sum_{k=-\infty}^{\infty} c_{ss}[k] \text{sinc}(k - m|\epsilon|) \right) \quad (13)$$

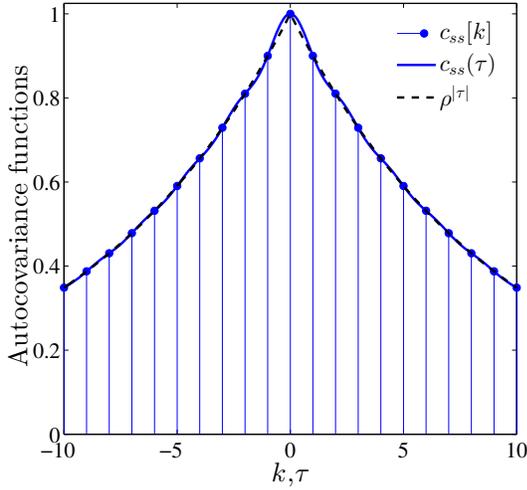


Fig. 5. Normalized autocovariance of an AR(1) random process with  $\rho = 0.9$  ( $c_{ss}[k] = 0.9^{|k|}$ ), its bandlimited reconstruction ( $c_{ss}(\tau)$ ), and  $\rho^{|\tau|}$ .

where  $c_{ss}[k]$  is the normalized autocovariance of  $s_0[n]$  ( $c_{ss}[k] = C_{ss}[k]/\sigma_s^2$ ) and  $\epsilon = v - \tilde{v}$ . Note that  $d_m^p$  does not depend on  $n$  (for this reason, we have dropped  $n$  from  $d_{n,m}^p$ ) nor does it depend on the mean of  $s_0[n]$  (if there is a DC component, it is always perfectly predicted). Since the ideal band-limited reconstruction of  $s_0[n]$  (with sampling period 1) is a zero-mean WSS continuous process with normalized autocovariance

$$c_{ss}(\tau) = \sum_{k=-\infty}^{\infty} c_{ss}[k] \text{sinc}(k - \tau), \quad (14)$$

we can also express (13) as

$$d_m^p = 2\sigma_s^2 (1 - c_{ss}(m|\epsilon|)). \quad (15)$$

When  $s_0[n]$  is AR(1) with  $\rho$  close to 1,  $c_{ss}(\tau)$  can be considered to be a monotonic decreasing function of  $|\tau|$  (see Fig. 5), and, consequently,  $d_m^p$  increases with increasing  $m$  and tends to  $2\sigma_s^2$ . The higher the  $|\epsilon|$ , the faster  $d_m^p$  increases with  $m$  (see Fig. 6). Hence, the average variance of  $d_m^p$  over a block of  $B \times B$  pixels increases with both  $|\epsilon|$  and  $B$ . Since  $d_m^p$  is independent of  $n$ , prediction is more efficient in a rectangular block of  $KB \times B/K$  pixels (with  $K > 1$ ) than in a block of  $B \times B$  pixels. This is the base of the block-splitting scheme proposed in [12].

#### D. Variance of the interpolation error

When  $\tilde{v} = v$  and  $h[n] \neq h_I[n]$ , only the interpolation error is introduced ( $e_m[n] = e_m^i[n]$ ). In this case, the error is caused by the use of a filter that is different to  $h_I[n]$ . The filters used in image interpolation are short-length FIR filters with the following properties: (i) they are zero-phase filters (i.e.,  $h[n] = h[-n]$ ); (ii) they are  $M$ th band filters (i.e.,  $h[Mn] = \delta[n]$ ); (iii) the frequency response of any polyphase component at  $\omega = 0$  is equal to 1 [16]. The ideal interpolation filter  $h_I[n]$  has these three properties (see Appendix B-A), and throughout the remainder of this paper we will assume that  $h[n]$  also has them.

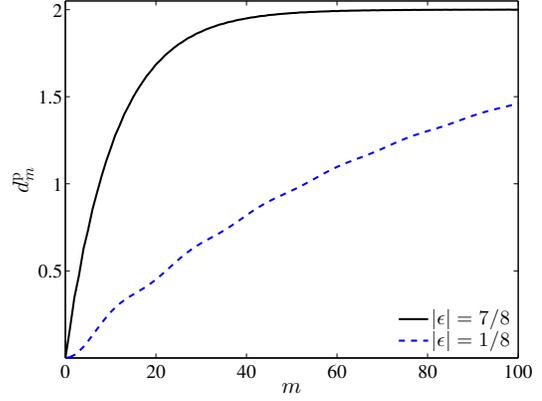


Fig. 6. Variance of  $e_m^p[n]$  when  $s_0[n]$  is AR(1) ( $\sigma_s^2 = 1$  and  $\rho = 0.9$ ) for  $|\epsilon| = 7/8$  and  $|\epsilon| = 1/8$ .

Let us define

$$u_m[n] = h_I[nM - mN] - h[nM - mN]. \quad (16)$$

Since  $h_I[nM - mN] = \text{sinc}(n - m\tilde{v})$ , we can rewrite (12) as

$$e_m^i[n] = s_0[n] * u_m[n]. \quad (17)$$

It can be shown (see Appendix B-B) that for each  $m$ ,  $e_m^i[n]$  is a zero-mean WSS random process. Its variance  $d_m^i$  is given by [17]

$$d_m^i = \sum_{k=-\infty}^{\infty} C_{ss}[k] r_{u_m u_m}[k]. \quad (18)$$

Sequence  $d_m^i$  has the following properties:

- 1)  $d_m^i$  is periodic with period  $M$ .
- 2)  $d_{mM}^i = 0$ .
- 3)  $d_{kM+l}^i = d_{(k+1)M-l}^i$  for  $0 \leq l < M$  and any  $k \in \mathbb{N}$ .

The proofs can be found in Appendix B-C.

Fig. 7 shows  $d_m^i$  ( $0 \leq m \leq 15$ ) when  $s_0[n]$  is AR(1) (with  $\sigma_s^2 = 1$  and  $\rho = 0.9$ ),  $N = 1$ ,  $M = 8$ , and for the filters used in linear interpolation, third-order cubic interpolation, and fourth-order cubic interpolation [16], [18]. These three filters represent different trade-offs between computational complexity and accuracy in approaching  $h_I[n]$ ; the linear filter requires the lowest number of computations and the fourth-order cubic filter is the one that best approaches  $h_I[n]$  [16], [18]. Note that the three curves in Fig. 7 exhibit the properties mentioned above. Also note that the better  $h[n]$  approaches  $h_I[n]$ , the smaller the  $d_m^i$ . Moreover,  $d_m^i \ll d_m^p$  except when  $m|\epsilon|$  is close to zero (compare Figs 6 and 7). In summary,  $d_m^i$  is periodic and small and mainly depends on  $h[n]$ .

#### E. Variance of the prediction error

When  $\tilde{v} \neq v$  and  $h[n] \neq h_I[n]$ , both the interpolation and the projection errors are present and they are correlated. If we define

$$z_m[n] = \text{sinc}(n - mv) - h[nM - mN] \quad (19)$$

then we can write (9) as  $e_m[n] = s_0[n] * z_m[n]$ . By following similar steps as for  $e_m^p[n]$  and  $e_m^i[n]$ , it can be shown that for

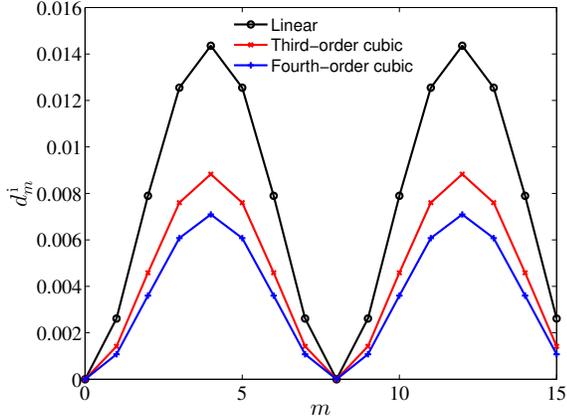


Fig. 7. Variance of  $e_m^i[n]$  as a function of  $m$  when  $s_0[n]$  is AR-1 ( $\sigma_s^2 = 1$  and  $\rho = 0.9$ ),  $M = 8$ ,  $N = 1$ , and for three different interpolation filters.

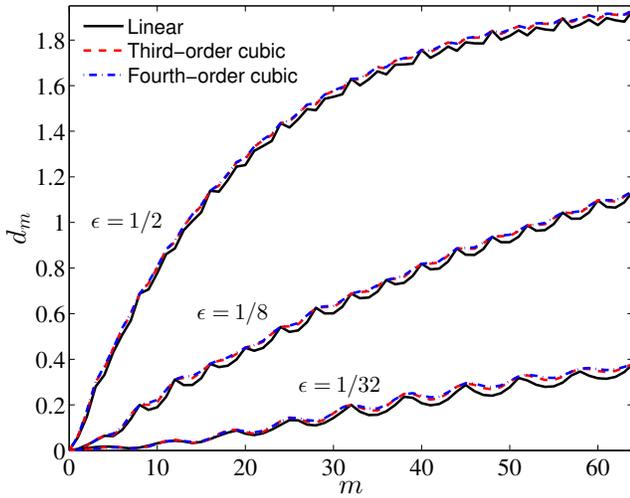


Fig. 8. Function  $d_m$  when  $s_0[n]$  is AR(1) (with  $\sigma_s^2 = 1$  and  $\rho = 0.9$ ) and  $v = -1/8$ . The function is plotted for three interpolation filters and for three values of  $\tilde{v}$ :  $-5/8$  ( $|\epsilon| = 1/2$ ),  $-1/2$  ( $|\epsilon| = 1/8$ ), and  $-5/32$  ( $|\epsilon| = 1/32$ ).

each  $m$ ,  $e_m[n]$  is a zero-mean WSS process whose variance  $d_m$  is given by

$$d_m = \sum_{k=-\infty}^{\infty} C_{ss}[k] r_{z_m z_m}[k]. \quad (20)$$

Fig. 8 shows  $d_m$  when  $s_0[n]$  is AR(1) (with  $\sigma_s^2 = 1$  and  $\rho = 0.9$ ) and  $v = -1/8$ ; the function is plotted for the three filters of Fig. 7 and for three values of  $\tilde{v}$ :  $-5/8$  ( $\epsilon = 1/2$ ),  $-1/4$  ( $\epsilon = 1/8$ ), and  $-5/32$  ( $\epsilon = 1/32$ ). Note that, like  $d_m^p$ ,  $d_m$  increases with  $m$  (the larger the  $|\epsilon|$ , the faster the increase). In contrast to  $d_m^p$ , however,  $d_m$  is not monotonically increasing. When  $m|\epsilon| \approx 0$ ,  $e_m[n] \approx e_m^i[n]$  and, hence,  $d_m \approx d_m^i$ . Therefore,  $d_m$  is approximately periodic when  $m|\epsilon| \approx 0$  (for instance, when  $|\epsilon| = 1/32$  and  $1 \leq m \leq 8$ ). Also note in Fig. 8 that the cubic filters perform almost equally and that the linear filter outperforms the other two for most values of  $m$ .

The two terms of  $z_m[n]$ ,  $\text{sinc}(n - mv)$  and  $h[nM - mN]$ , are centered around  $n = mv$  and  $n = mN/M$ , respectively. When  $m|\epsilon| \approx 0$ , both terms are approximately aligned. In that

case, the closer  $h[n]$  is to  $h_I[n]$ , the more  $h[nM - mN]$  cancels  $\text{sinc}(n - mv)$  in (19), and the smaller the  $d_m$ . Consequently, when  $m|\epsilon| \approx 0$ , those filters that better approach  $h_I[n]$  provide the smallest  $d_m$ . As  $m$  increases,  $\text{sinc}(n - mv)$  and  $h[nM - mN]$  separate, and  $h[nM - mN]$  progressively loses its capacity to cancel  $\text{sinc}(n - mv)$ . Hence, using a filter that approaches  $h_I[n]$  does not guarantee a smaller  $d_m$  when  $m|\epsilon|$  is not close to zero. Thus, when  $|\epsilon| = 1/32$ , the fourth-order cubic filter (very slightly) outperforms the other two for  $1 \leq m \leq 6$ , while the linear filter is the best for  $m \geq 11$ . However, when  $|\epsilon| = 1/2$ , the linear filter is the best for any  $m > 0$  (in that case,  $m|\epsilon|$  is never close to 0). In fact, when  $m|\epsilon| > 1/2$ , the filter  $h_I[n]$  performs worse than any of the three filters in most samples. Consequently, cubic or higher order interpolation filters should be avoided unless both  $B$  and  $|\epsilon|$  are very small.

#### F. Approximate expression for $d_m$

In this section, we derive an approximate expression for  $d_m$  when  $s_0[n]$  is AR(1) with  $\rho$  close to 1. Although our approximation  $\hat{d}_m$  may incur large errors, it notably facilitates the study of the AIP efficiency that is presented in Section IV.

To derive  $\hat{d}_m$ , we first approximate  $d_m$  with

$$d_m \approx 2\sigma_s^2 (1 - c_{ss}(m|\epsilon|)) \quad (21)$$

$$\approx 2\sigma_s^2 (1 - \rho^{m|\epsilon|}) \quad (22)$$

$$=: \bar{d}_m. \quad (23)$$

In (21), we neglect the interpolation error, and hence  $d_m \approx d_m^p$ . This approximation removes the oscillatory behavior of  $d_m$ . In (22), we approximate  $c_{ss}(x)$  with  $\rho^{|x|}$  (see Fig. 5). As shown in Fig. 9,  $\bar{d}_m$  increases monotonically with  $m$  and incurs large relative errors when  $m\epsilon \approx 0$ .

Finally,  $\hat{d}_m$  is the first-order Taylor approximation of  $\bar{d}_m$  at  $m|\epsilon| = 0$ :

$$\hat{d}_m = 2\sigma_s^2 \ln \rho^{-1} m|\epsilon|. \quad (24)$$

The reason of using this approximation is that, generally, in image coding,  $B$  is chosen small enough so that  $d_m$  be small in  $1 \leq m \leq B$ . Moreover, as shown in Section V,  $\hat{d}_m$  is a better approximation than  $\bar{d}_m$  when the directional sequence  $s[n, m]$  is a rectilinear edge.

From (24), the variance of the prediction error averaged over a block of  $B \times B$  pixels  $d$  is approximately

$$\hat{d} = \frac{1}{B^2} \sum_{n=1}^B \sum_{m=1}^B \hat{d}_m = 2\sigma_s^2 \ln \rho^{-1} (B+1)|\epsilon|. \quad (25)$$

Therefore, when  $s_0[n]$  is AR(1) with  $\rho$  close to 1, the efficiency of directional prediction is approximately determined by the product  $(B+1)|\epsilon|$ .

#### IV. VARIANCE OF THE PREDICTION ERROR IN AIP

In Section III, we studied the case where a directional sequence is predicted with a certain angle using its reference row. In this section, we extend the study by considering the prediction of directional sequences using AIP. Our study focuses on two distinctive features of AIP: the use of one or

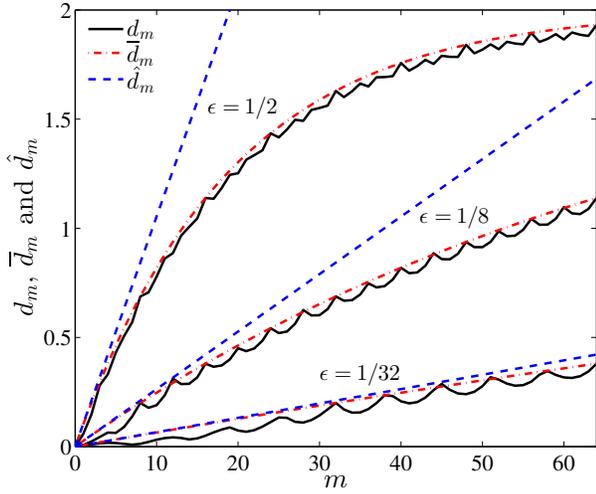


Fig. 9. Functions  $d_m$  (with linear interpolation),  $\bar{d}_m$ , and  $\hat{d}_m$ , when  $s_0[n]$  is AR(1) ( $\sigma_s^2 = 1$  and  $\rho = 0.9$ ), and  $v = -1/8$ . The three functions are plotted for three values of  $\tilde{v}$ :  $-5/8$  ( $\epsilon = 1/2$ ),  $-1/4$  ( $\epsilon = 1/8$ ), and  $-5/32$  ( $\epsilon = 1/32$ ).

two reference sequences depending on the prediction angle (Section IV-A), and the switching between a set of predefined prediction angles to adapt to any directional sequence (Section IV-B).

#### A. The use of one or two reference sequences in AIP

Depending on the value of  $\tilde{\theta}$ , one or both reference sequences are used for prediction in AIP. In the following, we study how this affects the efficiency of prediction. Consider a directional sequence  $s[n, m]$  of angle  $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$  and assume that  $s_0[n]$  is AR(1) with  $\rho$  close to 1. In this case, the normalized autocovariance of  $s_0[m]$  is approximately that of an AR(1) process with correlation coefficient  $\rho^{|\nu|}$  (see Appendix C). When  $\tilde{\theta} \in [-\frac{\pi}{4}, 0]$ , AIP only uses the reference row  $s_0[n]$  and, hence, we can approximate  $d_{n,m}$  with (24). In that case,  $\hat{d}_{n,m}$  is independent of  $n$  and grows linearly with  $m$  (see Fig. 10(a)). The case  $\tilde{\theta} \in (0, \frac{\pi}{4}]$  is a bit more complicated since AIP uses  $s_0[n]$  or  $s_0[m]$  depending on the pixel position. When  $n \geq m\tilde{v}$ , AIP uses  $s_0[n]$  and, therefore,  $d_{n,m}$  can be approximated with (24). When  $n < m\tilde{v}$ , AIP uses  $s_0[m]$ , and by symmetry with the previous case,

$$d_{n,m} \approx 2\sigma_s^2 \ln \rho^{-\nu} |\cot \theta - \cot \tilde{\theta}| n \quad (26)$$

$$= 2\sigma_s^2 \ln \rho^{-1} \tilde{v}^{-1} |\epsilon| n. \quad (27)$$

Summarizing, when  $\tilde{\theta} \in (0, \frac{\pi}{4}]$ ,  $d_{n,m}$  is approximately

$$\hat{d}_{n,m} = \begin{cases} 2\sigma_s^2 \ln \rho^{-1} |\epsilon| m, & n \geq m\tilde{v} \\ 2\sigma_s^2 \ln \rho^{-1} \tilde{v}^{-1} |\epsilon| n, & n < m\tilde{v} \end{cases}. \quad (28)$$

Hence, when  $\tilde{\theta} \in [0, \frac{\pi}{4}]$ ,  $\hat{d}_{n,m}$  depends on both  $n$  and  $m$  (see Fig. 10(b)). Similar expressions can be easily derived by symmetry when  $\tilde{\theta} \in (\frac{\pi}{4}, \frac{3\pi}{4}]$  and  $s_0[m]$  is an AR(1) process with  $\rho$  close to 1.

Let us now compare the prediction efficiency of using one or both reference sequences. Consider two pairs of angles,

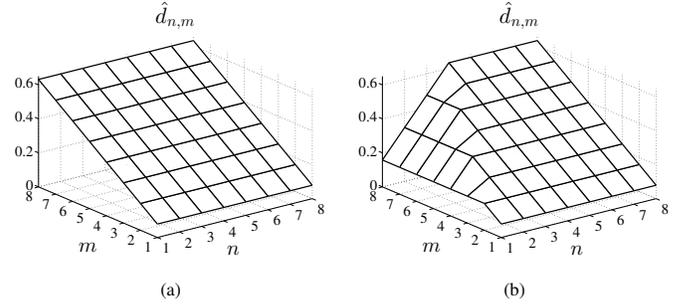


Fig. 10. Function  $\hat{d}_{n,m}$  for  $\sigma_s^2 = 1$ ,  $\rho = 0.9$ ,  $B = 8$  when  $\theta = -0.124$  and  $\tilde{\theta} = -0.464$  (a), and when  $\theta = 0.124$  and  $\tilde{\theta} = 0.464$  (b).

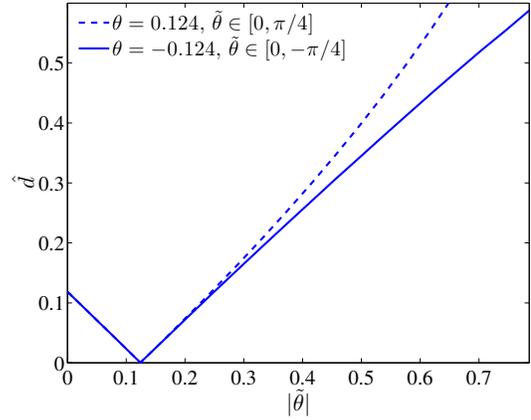


Fig. 11. Approximate average variance of the prediction error  $\hat{d}$  as a function of  $\tilde{\theta}$  for two values of  $\theta$  when  $s_0[n]$  is AR(1) ( $\sigma_s^2 = 1$ ,  $\rho = 0.9$ ) and  $B = 8$ . Both curves match when  $|\tilde{\theta}| < 0.124$  because, for these angles,  $n \geq m\tilde{v}$  for  $1 \leq n, m \leq 8$ .

$(\theta, \tilde{\theta})$  and  $(-\theta, -\tilde{\theta})$ , such that  $0 \leq \theta, \tilde{\theta} < \frac{\pi}{4}$ . From (24) and (28), we can write

$$\begin{cases} \hat{d}_{n,m}(\theta, \tilde{\theta}) = \hat{d}_{n,m}(-\theta, -\tilde{\theta}), & n \geq m \tan \tilde{\theta} \\ \hat{d}_{n,m}(\theta, \tilde{\theta}) < \hat{d}_{n,m}(-\theta, -\tilde{\theta}), & n < m \tan \tilde{\theta} \end{cases} \quad (29)$$

and, consequently,  $\hat{d}(\theta, \tilde{\theta}) \leq \hat{d}(-\theta, -\tilde{\theta})$ . Hence, even though  $|\epsilon|$  has the same value in both pairs of angles, using both reference sequences equals or improves the prediction efficiency with respect to using only one. For a given value of  $|\epsilon|$ , the improvement increases when  $\tilde{\theta}$  approaches  $\frac{\pi}{4}$  (see Fig. 11).

#### B. The use of switched prediction in AIP

In each block, AIP selects the optimal prediction direction. In this way, prediction is adapted to the directional features of each block. In this section, we study the efficiency of the switched prediction performed by AIP.

Let  $s[n, m]$  be a directional signal with angle  $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$  whose reference row  $s_0[n]$  is AR-1 with  $\rho$  close to 1. Let us assume that  $s[n, m]$  is predicted using  $s_0[n]$  irrespective of  $\theta$ .<sup>2</sup> Consequently, we can approximate  $d_{n,m}$  with  $\bar{d}_m$  and  $d$  with  $\hat{d}$ . To predict  $s[n, m]$  (where  $1 \leq n, m \leq B$ ), AIP considers a set of directions  $\{\tilde{v}_k\}_{k=1}^L$  and performs the prediction in the

<sup>2</sup>Even though this is not true when  $\tilde{\theta} \in [0, \frac{\pi}{4}]$ , this simplification of the prediction process greatly facilitates mathematical derivations.

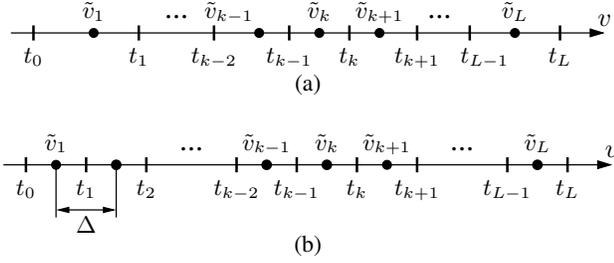


Fig. 12. Quantization of  $v$ . (a) General quantization. (b) Uniform quantization.

direction that minimizes  $d$ . If we approximate  $d$  with  $\hat{d}$ , the optimal direction is

$$\tilde{v}_k^* = \underset{\tilde{v}_k}{\operatorname{argmin}} \hat{d} = \underset{\tilde{v}_k}{\operatorname{argmin}} |v - \tilde{v}_k|. \quad (30)$$

Hence,  $\tilde{v}_k^*$  is the prediction direction that is closest to  $v$ . Selecting  $\tilde{v}_k^*$  can be seen as a scalar quantization of  $v$  [19] (see Fig. 12(a)). The *reproduction values* of the implicit quantizer are the directions  $\{\tilde{v}_k\}_{k=1}^L$ ; its *thresholds* are

$$t_k = \frac{\tilde{v}_k + \tilde{v}_{k+1}}{2}, \quad 1 \leq k < L, \quad (31)$$

with  $t_0 = -1$  and  $t_L = 1$ ; and the *quantization rule* is:  $Q(v) = \tilde{v}_k$  if  $v \in [t_{k-1}, t_k)$ .

Now, let us model the direction of  $s[n, m]$  as a continuous random variable  $V$  with probability density function (PDF)  $f_V(v)$  and domain  $[-1, 1]$ . Since  $d_{n,m}$  and  $\hat{d}_m$  are functions of  $V$ , they are also random variables. Then,

$$\mathbb{E}\{d_{n,m}\} \approx \mathbb{E}\{\hat{d}_m\} \quad (32)$$

$$= 2\sigma_s^2 \ln \rho^{-1} m \int_{-1}^1 f_V(v) |v - \tilde{v}_k| dv \quad (33)$$

$$= 2\sigma_s^2 \ln \rho^{-1} m \sum_{k=1}^L \int_{t_{k-1}}^{t_k} f_V(v) |v - \tilde{v}_k| dv. \quad (34)$$

Given  $f_V(v)$  and  $L$ , the optimal set  $\{\tilde{v}_k\}_{k=1}^L$  minimizes (34). The optimal  $\{\tilde{v}_k\}_{k=1}^L$  are equally spaced when  $f_V(v)$  is uniform [20]. This is not the case for typical images where vertical and horizontal edges appear much more frequently than the rest edges [21]. In AVS-Part 2, prediction directions are equally spaced  $\Delta = 1$ ; however, the outer directions share the same prediction mode 3 (see Fig. 1(a)) [2]. In H.264/AVC and AVS-Part 7, the  $\tilde{v}_k$  for predicting  $4 \times 4$  luma blocks are equally spaced  $\Delta = 1/2$  [1], [2]. In HEVC, the  $\tilde{v}_k$  are unequally spaced to adapt to the orientation statistics of images. Thus, in this standard, the minimum spacing is  $1/16$  (in the horizontal and vertical directions) and the maximum is  $3/16$  (in the diagonal directions) [4].

We can approximately solve (34) when  $Q$  is uniform with step size  $\Delta = 2/L$  (see Fig. 12(b)) and  $f_V(v)$  is approximately

constant in each interval.<sup>3</sup> In this case,

$$\begin{aligned} \mathbb{E}\{\hat{d}_m\} &\stackrel{(a)}{\approx} 2\sigma_s^2 \ln \rho^{-1} m \sum_{k=1}^L f_V(\tilde{v}_k) \int_{t_{k-1}}^{t_k} |v - \tilde{v}_k| dv \\ &\stackrel{(b)}{=} 0.5\sigma_s^2 \ln \rho^{-1} m \Delta \sum_{k=1}^L f_V(\tilde{v}_k) \Delta \\ &\stackrel{(c)}{\approx} 0.5\sigma_s^2 \ln \rho^{-1} m \Delta \int_{-1}^1 f_V(v) dv \\ &\stackrel{(d)}{=} 0.5\sigma_s^2 \ln \rho^{-1} m \Delta =: \hat{D}_m. \end{aligned} \quad (35)$$

In (a), the PDF is approximated by a constant (its midpoint value) in each interval; in (b) we solve the integral; in (c) we use the fact that  $\sum_{k=1}^L f_V(\tilde{v}_k) \Delta$  is an approximation of the integral of  $f_V(v)$ ; and, finally, in (d) we use the fact that the integral of any PDF is unity.

Let us now consider that  $\theta \in (\frac{\pi}{4}, \frac{3\pi}{4}]$ . If we assume hypotheses similar to those used to derive (35),<sup>4</sup> by symmetry with the previous case,  $d_{n,m}$  is a random variable whose mean is approximately

$$\hat{D}_n = 0.5\sigma_s^2 \ln \rho^{-1} n \Delta. \quad (36)$$

Finally, if  $\theta \in [-\frac{\pi}{4}, \frac{3\pi}{4}]$ , the approximate variance of the prediction error at pixel  $(n, m)$  is given by

$$\hat{D}_{n,m} = p_r \hat{D}_m + p_c \hat{D}_n \quad (37)$$

where

$$p_r = \mathbb{P}\left\{-\frac{\pi}{4} \leq \tilde{\theta} \leq \frac{\pi}{4}\right\} \quad \text{and} \quad p_c = \mathbb{P}\left\{\frac{\pi}{4} < \tilde{\theta} \leq \frac{3\pi}{4}\right\}.$$

Finally, by substituting (35) and (36) into (37), we obtain

$$\hat{D}_{n,m} = 0.5\sigma_s^2 \ln \rho^{-1} \Delta (p_r m + p_c n). \quad (38)$$

Function  $\hat{D}_{n,m}$  is a 2D function that increases linearly with both  $n$  and  $m$ . This contrasts with the distribution of the error variance in conventional motion-compensated prediction for video where variance is almost constant at the center of the block but increases when approaching each block border [22].

The average of  $\hat{D}_{n,m}$  over all the pixels of a block is

$$\hat{D} = \frac{1}{B^2} \sum_{n=1}^B \sum_{m=1}^B \hat{D}_{m,n} \quad (39)$$

$$= 0.25\sigma_s^2 \ln \rho^{-1} \Delta (B + 1). \quad (40)$$

Hence, apart from the signal parameters ( $\sigma_s^2$  and  $\rho$ ), prediction efficiency is determined by the product  $\Delta(B + 1)$ . From a coding perspective, the larger the  $B$ , the smaller the number of overhead bits and the greater the energy compaction of the transform; however, according to (40), the larger  $B$  is, the less efficient the prediction is. Nevertheless, we can increase  $B$  and still keep  $\hat{D}$  constant if  $\Delta$  is decreased so that  $\Delta(B + 1)$  remains constant. Thus, in those coders that use variable

<sup>3</sup>The hypotheses and the steps performed to derive (35) are similar to the ones assumed to obtain the high-resolution formula for uniform quantizers [19].

<sup>4</sup>The hypotheses are: only  $s_0[m]$  is used for prediction;  $s_0[m]$  is AR-1 with  $\rho$  close to 1,  $Q$  is uniform without step size  $\Delta = 2/L$ , and  $f_V(v)$  is approximately constant in each quantization interval.



Fig. 13. Synthetic images used to experimentally test prediction efficiency.

block-size partitioning, constant prediction efficiency can be achieved if a different number of prediction directions are used for each block size. This, however, increases the algorithmic complexity of coding, especially when the number of block sizes is large [3], [4].

## V. EXPERIMENTAL RESULTS

In this section, we present the results obtained after applying directional prediction and AIP to three directional images. We also compare these experimental results with the results of Sections III and IV in order to assess how accurately our theoretical expressions model the efficiency of practical prediction algorithms.

The experimental results were obtained by using three synthetic digital images, I1, I2, and I3, which contain rectilinear edges (see Fig. 13). We used this type of images because strong directional edges are the main structures that are targeted by AIP [3]. These three images were generated by performing squared zero-order hold filtering, rectangular sampling (with  $T_x = T_y = 1$ ), and 8-bit uniform quantization over rectilinear 2D step functions of different angles.<sup>5</sup> I1 and I2 contain a single edge of contrast 100 and angles  $\theta = -0.124$  and  $\theta = 0.124$ , respectively. I3 contains 61 edges with equally-spaced angles between  $-\frac{\pi}{4}$  and  $\frac{\pi}{4}$  (the contrast of each edge is 35). The edge spacing guarantees that when I3 is divided into square blocks of  $B \times B$  pixels, any block contains at most one edge if  $B \leq 64$ .

The process to generate the results of this section is the following. First, an image is first partitioned into blocks of  $B \times B$  pixels. Then, each block is predicted from its reference sequences using directional prediction or AIP. Those blocks that have incomplete reference sequences are discarded. Finally, the prediction error is obtained and the mean squared (prediction) error (MSE) is computed. The reference sequences contain original pixel values rather than decoded values as happens in video coding. The linear interpolation filter is used unless otherwise stated. AIP predictions are performed using the algorithm of Section II with equally spaced prediction directions and selecting the direction that provides the minimum MSE.

Fig. 14 shows the MSE as a function of  $m$  obtained when I1 is directionally predicted with  $B = 64$  and with the same directions and filters of Fig. 8. Note that Fig. 14 is similar to Fig. 8 except for the fact that, in Fig. 14, the MSE increases indefinitely with  $m$  instead of tending to a fixed value. Hence,  $\hat{d}_m$  is more appropriate than  $\bar{d}_m$  to approximate  $d_m$  when the blocks to be predicted contain a single edge. Similarly to Fig. 8, the two cubic filters perform almost equally and

<sup>5</sup>A 2D step  $u(x, y)$  of angle  $\theta$  is defined as  $u(x, y) = u(x - \tan \theta)$  where  $u(x)$  is the 1D step (i.e.,  $u(x) = 0$  for  $x < 0$  and  $u(x) = 1$  for  $x \geq 0$ ).

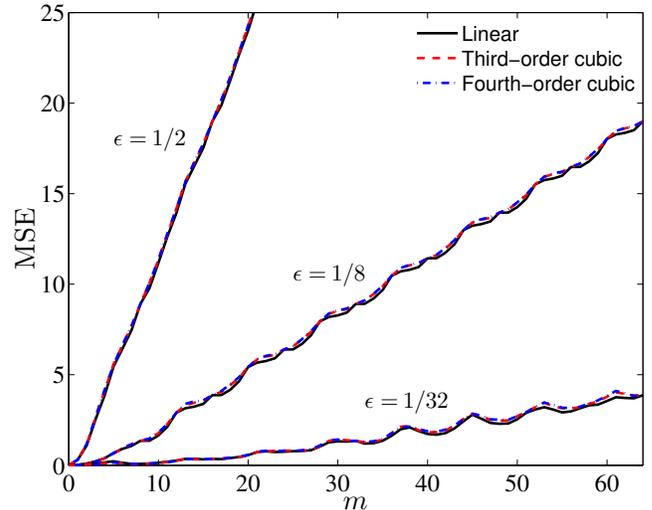


Fig. 14. MSE as a function of  $m$  when I1 ( $v = -1/8$ ) is directionally predicted with  $B = 64$  and three values of  $\bar{v}$ :  $-\frac{5}{8}$  ( $\epsilon = \frac{1}{2}$ ),  $-\frac{1}{4}$  ( $\epsilon = \frac{1}{8}$ ) and  $-\frac{5}{32}$  ( $\epsilon = \frac{1}{32}$ ).

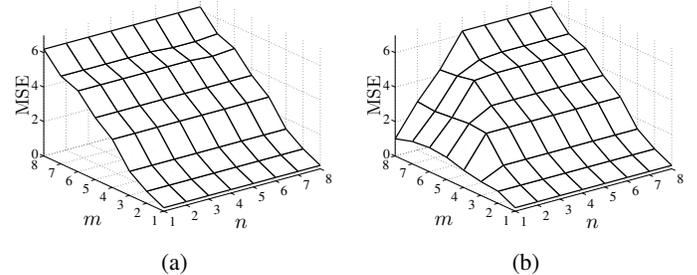


Fig. 15. MSE of each block pixel ( $B = 8$ ) when I1 (a) and I2 (b) are predicted using  $\hat{\theta} = -0.464$  and  $\hat{\theta} = 0.464$ , respectively.

the linear filter outperforms the other two except when  $m|\epsilon|$  is close to 0. Thus, when  $|\epsilon| = 1/32$ , the fourth-order cubic filter is the best for  $1 \leq m \leq 6$  while the linear filter is the best for  $m \geq 16$ . However, when  $|\epsilon| = 1/2$ , the linear filter is the best for any  $m > 0$ . These results confirm that linear filtering should be used unless both  $B$  and  $|\epsilon|$  are very small.

Fig. 15 shows the MSE of each block pixel obtained when I1 and I2 are directionally predicted using  $\hat{\theta} = -0.464$  and  $\hat{\theta} = 0.464$ , respectively. In both cases,  $B = 8$ . Since the values of  $\hat{\theta}$  and  $\theta$  in Fig. 15 are the same as those in Fig. 10, both figures show the same main trends. Nevertheless, Fig. 10 does not exhibit the oscillations of Fig. 15 because the approximation  $\hat{d}_m$  neglects the interpolation error. As expected, the MSE averaged over all block pixels is greater in I1 (3.08) than in I2 (2.65) since I2 benefits from being predicted using both reference sequences.

Fig. 16(a) shows the MSE of each block pixel obtained when I3 is predicted using AIP with  $B = 8$  and  $\Delta = 1/4$ . Note that the MSE depends on  $m$ , but it also depends slightly on  $n$ , while  $\hat{D}_m$  only depends on  $m$ . The reason for this discrepancy is that AIP uses both  $s_0[m]$  and  $s_0[n]$  for predicting blocks with  $\theta \in [0, \frac{\pi}{4}]$ , whereas  $\hat{D}_m$  was derived by assuming that only  $s_0[n]$  is used in those blocks. From the MSE of Fig. 16(a), we can obtain the MSE of predicting 121 equally-spaced

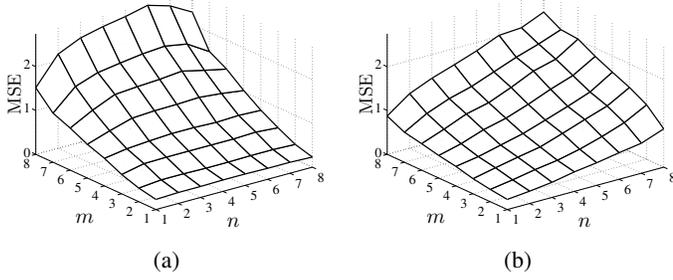


Fig. 16. MSE of each block pixel when using AIP with  $B = 8$  and  $\Delta = 1/4$  over I3 (a). MSE for 121 equally spaced edges between  $-\frac{\pi}{4}$  and  $\frac{3\pi}{4}$  (b).

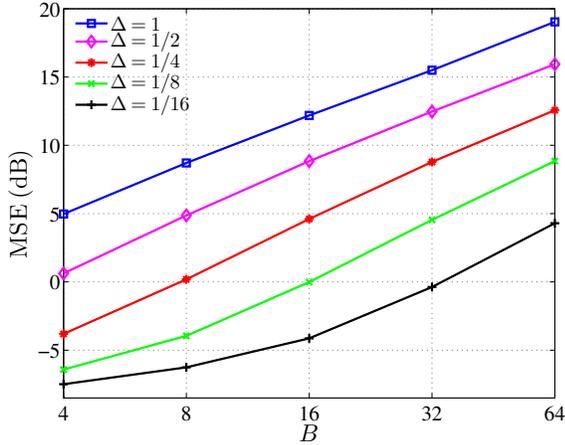


Fig. 17. MSE (in dB) as a function of  $B$  and for several values of  $\Delta$  obtained when I3 is predicted using AIP.

angles between  $-\frac{\pi}{4}$  and  $\frac{3\pi}{4}$ , which is shown in Fig. 16(b).<sup>6</sup> When  $p_r = p_c$ ,  $\hat{D}_{n,m}$  is a 2D linear function that increases equally fast with  $n$  and  $m$ . Note that this is approximately the behavior of the MSE in Fig. 16(b).

Fig. 17 shows the MSE (in dB) as a function of  $B$  and for several values of  $\Delta$  obtained when I3 is predicted using AIP. The minimum and maximum values of  $B$  and  $\Delta$  have been chosen to cover the range of values that these two parameters have in H.264/AVC, AVS, and HEVC. Note that, as in  $\hat{D}$ , the MSE is essentially determined by the product  $(B+1)\Delta$  (those points  $(B, \Delta)$  with similar values of  $(B+1)\Delta$  also have similar MSE values). According to  $\hat{D}$ , the MSE should increase approximately 3 dB when doubling  $B$  or halving  $\Delta$ . In Fig. 17, this increase is larger than 3 dB except when both  $B$  and  $\Delta$  are small. Thus, passing from  $(4, \frac{1}{16})$  to  $(8, \frac{1}{16})$  or to  $(4, \frac{1}{8})$  only increases the MSE by approximately 1.1 dB. In such highly accurate predictions, the interpolation error is not negligible, and, consequently,  $\hat{D}$  incurs large errors. In fact, when  $B = 4$  and  $\Delta = 1/16$  (i.e.,  $|\epsilon| \leq 1/32$ ), the projection error is negligible in relation to the interpolation error (see Fig. 8). Consequently, decreasing  $\Delta$  below  $1/16$  for such a small block cannot significantly decrease the MSE (we obtained a decrease of only 0.35 dB with  $\Delta = 1/32$ ).

<sup>6</sup>If  $MSE(n, m)$  is the MSE of I3 at each block pixel, then  $0.5(MSE(n, m) + MSE(m, n))$  is the MSE of predicting 121 equally-spaced angles between  $-\frac{\pi}{4}$  and  $\frac{3\pi}{4}$ .

## VI. CONCLUSION

In this paper, we have studied the efficiency of AIP in the prediction of directional images. First, we have studied the error incurred when a directional image is predicted in a certain direction. We show how the variance of the prediction error depends on factors such as the pixel position and the error in estimating the direction of the image. The results of this study have allowed us to analyze the efficiency of AIP in the prediction of images with a distribution of directions. By using some approximations, we have obtained simple expressions for the variance of the prediction error in AIP. Finally, we have compared these expressions with experimental results obtained from the prediction of rectilinear edges. This comparison has shown how accurately our theoretical expressions model AIP efficiency.

## REFERENCES

- [1] I. E. Richardson, *The H.264 advanced video compression standard*. Wiley, 2010.
- [2] K. R. Rao, D. N. Kim, and J. J. Hwang, *Video Coding Standards*. Springer, 2014.
- [3] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
- [4] J. Lainema, F. Bossen, W.-J. Han, J. Min, and K. Ugur, "Intra coding of the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1792–1801, Dec. 2012.
- [5] T. Halbach and M. Wien, "Concepts and performance of next-generation video compression standardization," in *Nordic Signal Processing Symposium (NORSIG)*, Norway, Oct. 2002.
- [6] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*. Prentice-Hall, 2009.
- [7] B. Girod, "The efficiency of motion-compensating prediction for hybrid coding of video sequences," *IEEE J. Sel. Areas Commun.*, vol. 5, no. 7, pp. 1140–1154, Aug. 1987.
- [8] —, "Motion-compensating prediction with fractional-pel accuracy," *IEEE Trans. Commun.*, vol. 41, no. 4, pp. 604–611, Apr. 1993.
- [9] J. Han, A. Saxena, V. Melkote, and K. Rose, "Jointly optimized spatial prediction and block transform for video and image coding," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1874–1884, Apr. 2012.
- [10] A. Saxena and F. C. Fernandes, "DCT/DST-based transform coding for intra prediction in image/video coding," *IEEE Trans. Image Process.*, vol. 22, no. 10, pp. 3974–3981, Oct. 2013.
- [11] C. Yeo, Y. H. Tan, Z. Li, and S. Rahardja, "Mode-dependent transforms for coding directional intra prediction residuals," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 4, pp. 545–554, Apr. 2012.
- [12] X. Cao, C. Lai, Y. Wang, L. Liu, J. Zheng, and Y. He, "Short distance intra coding scheme for high efficiency video coding," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 790–801, Feb. 2013.
- [13] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*. Prentice-Hall, 1983.
- [14] P. Vaidyanathan, *Multirate systems and filter banks*. Prentice-Hall, Inc, 1993.
- [15] W. K. Pratt, *Digital Image Processing*. Wiley-Interscience, 2007.
- [16] E. Meijering and M. Unser, "A note on cubic convolution interpolation," *IEEE Trans. Image Process.*, vol. 12, no. 4, pp. 477–479, Apr. 2003.
- [17] D. G. Manolakis, V. K. Ingle, and V. K. Ingle, *Statistical and Adaptive Signal Processing*. Artech House, 2005.
- [18] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 6, pp. 1153–1160, Dec. 1981.
- [19] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2325–2383, Oct. 1998.
- [20] N. S. Jayant and P. Noll, *Digital coding of waveforms*. Prentice-Hall, Inc, 1984.
- [21] N. Keskes, F. Kretz, and H. Maitre, "Statistical study of edges in TV pictures," *IEEE Trans. Commun.*, vol. 27, no. 8, pp. 1239–1247, Aug. 1979.
- [22] M. T. Orchard and G. T. Sullivan, "Overlapped block motion compensation: an estimation-theoretic approach," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 693–699, Oct. 1994.

## APPENDIX A

## STATISTICAL PROPERTIES OF THE PROJECTION ERROR

If we define

$$p_m[n] = \text{sinc}(n - mv) - \text{sinc}(n - m\bar{v}) \quad (41)$$

then  $e_m^p[n] = s_0[n] * p_m[n]$ . Thus, for each  $m$ ,  $e_m^p[n]$  is a WSS random process. The mean of  $e_m^p[n]$  is [17]

$$\mathbb{E}\{e_m^p[n]\} = \mathbb{E}\{s_0[n]\} P_m(e^{j0}), \quad (42)$$

and since  $P_m(e^{j0}) = 0$  for any  $m$ ,  $\mathbb{E}\{e_m^p[n]\} = 0$ . The variance of  $e_m^p[n]$ , denoted  $d_m^p$ , is [17]

$$d_m^p = \sigma_s^2 \sum_{k=-\infty}^{\infty} c_{ss}[k] r_{p_m p_m}[k] \quad (43)$$

where  $c_{ss}[k] = C_{ss}[k]/\sigma_s^2$ . The autocorrelation of  $p_m[n]$  is

$$r_{p_m p_m}[k] = 2\delta[k] - \text{sinc}(k - m\epsilon) - \text{sinc}(k + m\epsilon) \quad (44)$$

where  $\epsilon = v - \bar{v}$ . By substituting (44) into (43), we obtain

$$d_m^p = \sigma_s^2 \left( 2 - \sum_{k=-\infty}^{\infty} c_{ss}[k] (\text{sinc}(k - m\epsilon) + \text{sinc}(k + m\epsilon)) \right) \quad (45)$$

and, as  $c_{ss}[k]$  and  $\text{sinc}(x)$  are even functions, we finally have

$$d_m^p = 2\sigma_s^2 \left( 1 - \sum_{k=-\infty}^{\infty} c_{ss}[k] \text{sinc}(k - m|\epsilon|) \right). \quad (46)$$

## APPENDIX B

## STATISTICAL PROPERTIES OF THE INTERPOLATION ERROR

A. Properties of  $h_1[n]$ 

Since  $\text{sinc}(n/M)$  is an even function,  $h_1[n]$  has zero phase. Since  $h_1[nM] = \text{sinc}(n) = \delta[n]$ ,  $h_1[n]$  is an  $M$ -th band filter. The  $l$ th polyphase component of  $h_1[n]$  is  $h_{1,l}[n] = \text{sinc}(n + l/M)$  and its frequency response is  $H_{1,l}(e^{j\omega}) = e^{j\omega l/M}$ . Consequently,  $H_{1,l}(e^{j0}) = 1$  for any  $l$ .

B. Mean of  $e_m^i[n]$ 

Given a positive integer  $M$ , any integer  $n$  can be expressed as

$$n = q(n) \times M + ((n))_M \quad (47)$$

where

$$((n))_M = n \text{ modulo } M \quad (48)$$

is always an integer between 0 and  $M - 1$  and  $q(n) = \lfloor \frac{n}{M} \rfloor$ . Using (47), we can write

$$h[nM - mN] = h[M(n + q(-mN)) + ((-mN))_M], \quad (49)$$

i.e.,  $h[nM - mN]$  is a polyphase component of  $h[n]$  translated  $q(-mN)$  samples. Since the Fourier transform of any polyphase component of  $h[n]$  is 1 at  $\omega = 0$ , the Fourier transform of  $h[nM - mN]$  is also 1 at  $\omega = 0$ . A similar reasoning is valid for  $h_1[nM - mN]$ . Consequently,  $U_m(e^{j0}) = 0$  for any  $m$ , and hence,  $\mathbb{E}\{e_m^i[n]\} = 0$ .

C. Proof of the properties of  $d_m^i$ 

Property 1. From (16), we can easily derive

$$u_{i+kM}[n] = u_i[n - kN] \quad (50)$$

for any integers  $i$  and  $k$ . Using (47) and (50), we obtain

$$u_m[n] = u_{((m))_M}[n - \lfloor m/M \rfloor N] \quad (51)$$

for any  $m \geq 0$ . As  $e_m^i[n] = s_0[n] * u_m[n]$ , we can write

$$e_m^i[n] = e_{((m))_M}^i[n - \lfloor m/M \rfloor N]. \quad (52)$$

Finally, since displacing a WSS process does not alter its variance,  $d_m^i = d_{((m))_M}^i$ , i.e.,  $d_m^i$  is periodic with period  $M$ .

Property 2. Since  $h[n]$  is assumed to be an  $M$ -th band filter ( $h[nM] = \delta[n]$ ), we have  $u_0[n] = \text{sinc}(n) - h[nM] = 0$ . Hence,  $d_0^i = 0$ , and using Property 1, we obtain  $d_{mM}^i = 0$ .

Property 3. Consider two integers  $k$  and  $l$  with  $1 \leq l \leq M - 1$ . Then,

$$\begin{aligned} u_{(k+1)M-l}[N(2k+1) - n] &\stackrel{(a)}{=} u_{l-(k+1)M}[n - N(1+2k)] \\ &\stackrel{(b)}{=} u_{kM+l}[n]. \end{aligned} \quad (53)$$

In (a), we have used the fact that since  $\text{sinc}(x)$  and  $h[n]$  are even functions, then  $u_m[n] = u_{-m}[-n]$ ; and in (b) we have used (50). Then,

$$e_{(k+1)M-l}^i[N(2k+1) - n] = e_{kM+l}^i[n] \quad (54)$$

and since inverting and displacing a WSS process does not alter its variance, we finally obtain

$$d_{(k+1)M-l}^i = d_{kM+l}^i. \quad (55)$$

## APPENDIX C

CORRELATION COEFFICIENTS OF  $s_0[m]$  AND  $s_0[n]$ 

Let  $s_r(x) = s(x, 0)$  and  $s_c(y) = s(0, y)$ , and let  $c_{s_r s_r}(\tau_x)$  and  $c_{s_c s_c}(\tau_y)$  be the normalized autocorrelations of  $s_r(x)$  and  $s_c(y)$ , respectively. Since  $s_c(y) = s_r(-vy)$ , we have

$$c_{s_c s_c}(\tau_y) = c_{s_r s_r}(-v\tau_y). \quad (56)$$

When,  $s_0[n]$  is AR(1) with  $\rho$  close to 1,  $c_{s_r s_r}(\tau_x) \approx \rho^{|\tau_x|}$  (see Fig. 5). In this case, from (56) we obtain

$$c_{s_c s_c}(\tau_y) \approx \rho^{|v||\tau_y|}. \quad (57)$$

Finally, since  $s_0[m] = s_c(y)|_{y=m}$ , the normalized autocovariance of  $s_0[m]$  is approximately  $\rho^{|v||m|}$ , which is the autocovariance of an AR(1) process with correlation coefficient  $\rho^{|v|}$ .