

An effective error concealment scheme for heavily corrupted H.264/AVC videos based on Kalman filtering

Shihua Cui · Huijuan Cui · Kun Tang

Received: 22 April 2012 / Revised: 3 October 2012 / Accepted: 4 October 2012 / Published online: 18 October 2012
© Springer-Verlag London 2012

Abstract Error concealment at the decoder side is an economical approach to ensuring an acceptable and stable video quality in case of packet erasure or loss, and thus, it has attracted considerable research and application interest. Relevant techniques usually employ the spatial or temporal correlation to recover the motion vectors (MVs) of the missing blocks, and interpolation, extrapolation, or boundary-matching schemes are usually effective. However, for heavily corrupted sequences, e.g., with block loss rate beyond 50%, most methods might perform less satisfactorily. Inspired by the tracking efficiency of Kalman filter (KF), in the present work, we adopted it to predict the missing MVs, and the unpredicted ones (minority) were restored complementarily using a modified bilinear motion field interpolation (MFI) method. Since the KF prediction is independent of the loss rate, the present framework proves to be robust for heavily corrupted videos. Experimental results on typical sequences reveal that the proposed algorithm outperforms the boundary-matching algorithm embedded in the H.264/AVC reference code, the MFI and the MV extrapolation techniques in the literature.

Keywords Error concealment · Kalman filter · Motion vector prediction · H.264/AVC

1 Introduction

To ensure video quality through the error-prone network conditions, error concealment (EC) is a popular post-processing technique at the decoder side to alleviate the transmission errors, by exploiting the temporal or spatial information from available image regions. For a P-frame, the motion vectors (MVs) of the missing blocks can be recovered using the MVs of the neighboring blocks in the same frame or the MVs from the previous frames. The well-known boundary-matching algorithm (BMA) [1] has been provided in the H.264/AVC reference codec and has been modified to the outer boundary-matching algorithm (OBMA) [2]. Another popular approach is the motion field interpolation (MFI) technique [3–5], where the missing MVs are obtained via bilinear interpolation of those in the neighboring available macroblocks (MBs). However, since these methods depend on the availability and reliability of the neighboring MVs or pixels (for BMA), in the case of heavy corruption and when all the neighboring MBs are missing, they will generally degrade to the simple frame copy (FC) scheme. Therefore, they and their modifications are usually applied to situations with a MB loss rate (MLR) below 50%.

To handle heavy errors, temporally predicting the MVs is especially desirable. The optical flow estimation [6,7] is the mainstream technique in this direction, but it is not always better than the FC operation. MV extrapolation [8,9] is another way. It has been refined to the pixel level to remove block artifacts [10], and a hybrid scheme containing both block and pixel levels has been proposed recently as well [11]. In the present work, however, we put forward a new scheme for MV prediction, by introducing the Kalman filter (KF) [12] to the MV tracking. We assume that the MV evolution between frames is a Markov process and that the KF can conduct the predictions. Consequently, we use the

This work has been supported by the National Natural Science Foundation of China under Grant 60572081.

S.Cui (✉) · H.Cui · K.Tang
Tsinghua National Laboratory for Information Science and Technology,
Department of Electronic Engineering, Tsinghua University,
Beijing 100084, People's Republic of China
e-mail: seacui@vip.sina.com

predicted MVs to restore the missing MBs. KF [13–16] and its extensions such as the extended KF (EKF) [17–19] and the unscented KF (UKF) [20–22] have been applied widely in various fields. Within the video community, KF-related methods have been employed mainly to track objects, for applications of surveillance, visual inspection, human-computer interfaces [23], etc. However, unlike the normal one-target tracking [16], when applied in the EC framework several tricks are needed, for tracking the MVs of all the blocks is a multi-target tracking in cluttered environments with frequent occlusion, disappearance and (re)appearance of the targets. Handling such difficulties is challenging even in the video tracking community [24,25]. Fortunately, we have found some efficient solutions, and the experimental results seem quite promising.

It is worth noting that KF was applied previously [26, 27] to EC as a post-processing step to BMA to refine the MVs obtained via BMA. The framework is different from the present one, and it handles videos with low MLR, leading to PSNR improvements of 0.4–0.72 dB [26], or 0.1–0.2 dB [27].

The following sections will describe the algorithm and results in detail. In Sect. 2, the model and strategies will be stressed. The experimental settings and results will be offered in Sect. 3, along with our analyses. Section 4 will summarize our algorithm, and some prospective work will be stated.

2 Proposed scheme

In H.264/AVC, a MB (16×16) can be partitioned into sub-MBs of 16×16 , 16×8 , 8×16 , or 8×8 , and the 8×8 sub-MB can be further partitioned into blocks of 8×8 , 8×4 , 4×8 , or 4×4 . Therefore, the minimum block size is 4×4 , and MVs can be assigned to such blocks. From the consideration of accuracy, in the present work, we account for the MVs of such blocks, and the blocks hereinafter all mean such 4×4 ones (i.e., each MB contains 16 blocks). We assume that the evolution of the MV of each block from frame to frame is a Markov process, and thus, we adopt the KF to track it. We use a state vector \mathbf{x} containing four elements for each block, as in (1), where v means the MV, a stands for the evolution rate of v (the acceleration), and subscripts x and y reflect the components along the frame width and height directions, respectively. The true state \mathbf{x}_k of a block at frame k is assumed to evolve from that at frame $k - 1$ according to (2). We set the time step between two successive frames as unity and exploit a constant state transition matrix F as depicted in (3). In KF, the process noise w is assumed to be taken from a zero mean multivariate normal distribution. Here, we suppose a constant process noise correlated with a constant covariance Q as shown in (4), with $Q = 0.64I_4$ (I_4 is the identity matrix of size four). The Q and the following

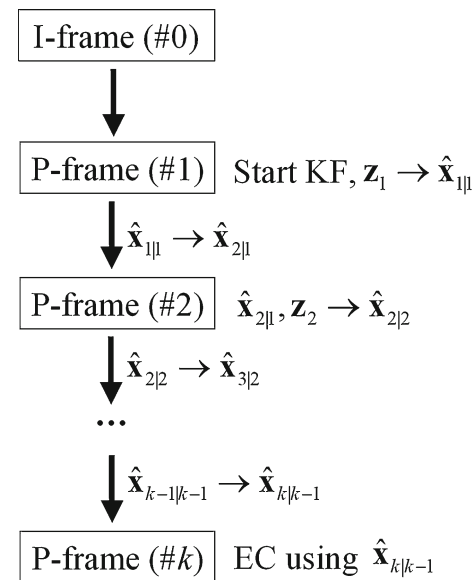


Fig. 1 Flow diagram of the overall KF-based EC procedure

R values are determined empirically based on tests on many sequences for generally optimal performance.

$$\mathbf{x} = (v_x \ v_y \ a_x \ a_y)^T \quad (1)$$

$$\mathbf{x}_k = F\mathbf{x}_{k-1} + w \quad (2)$$

$$F = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3)$$

$$w \sim N(0, Q) \quad (4)$$

The observation \mathbf{z}_k of \mathbf{x}_k of a block at frame k is made using (5), where we adopted a constant observation matrix H as given in (6). For the observation noise u , KF also presumes the zero mean Gaussian white noise, and here, we employ a constant one as in (7) with a constant covariance $R = 0.64I_2$ (the identity matrix of size two).

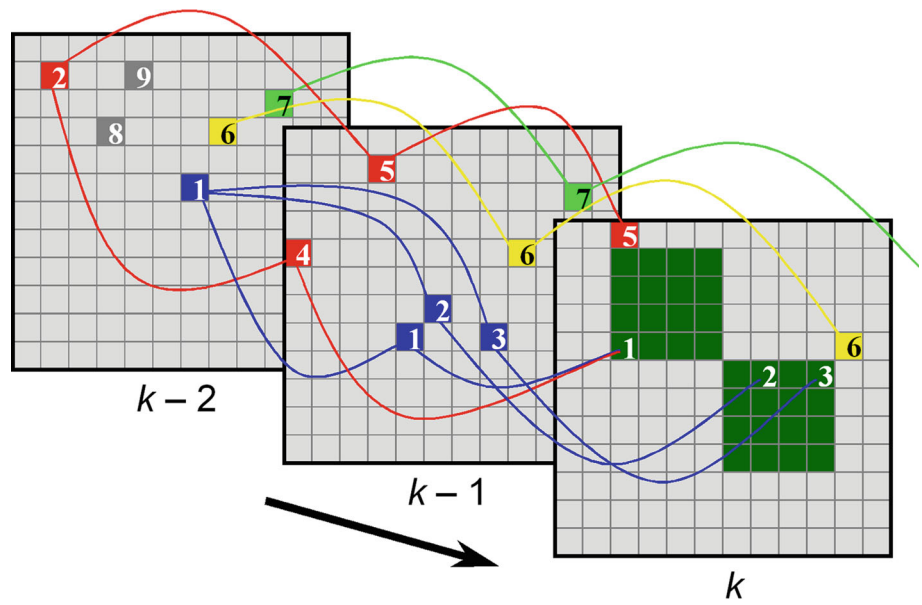
$$\mathbf{z}_k = H\mathbf{x}_k + u \quad (5)$$

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad (6)$$

$$u \sim N(0, R) \quad (7)$$

The detailed procedure of tracking the state of each block can be described through its alternate prediction and updating phases. During the prediction step, the predicted state $\hat{\mathbf{x}}_{k|k-1}$ of a block for frame k is produced using the updated state estimate $\hat{\mathbf{x}}_{k-1|k-1}$ of frame $k - 1$, as in (8). The predicted estimate of covariance $P_{k|k-1}$ of frame k is based on the updated estimate of covariance $P_{k-1|k-1}$ of frame $k - 1$ and the covariance Q from the process noise, as in (8). At the updating step, the updated state estimate $\hat{\mathbf{x}}_{k|k}$ at frame k is obtained using the observation \mathbf{z}_k and the observation noise

Fig. 2 Illustration of the block correspondence between successive frames during our KF-based tracking



R at frame k , as given in (10). Then, we get the updated estimate of covariance at frame k using (11).

$$\hat{\mathbf{x}}_{k|k-1} = F\hat{\mathbf{x}}_{k-1|k-1} \tag{8}$$

$$P_{k|k-1} = FP_{k-1|k-1}F^T + Q \tag{9}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + P_{k|k-1}H^T(H P_{k|k-1}H^T + R)^{-1}(\mathbf{z}_k - H\hat{\mathbf{x}}_{k|k-1}) \tag{10}$$

$$P_{k|k} = (I - P_{k|k-1}H^T(H P_{k|k-1}H^T + R)^{-1}H)P_{k|k-1} \tag{11}$$

We employed the IPPP mode (a first I-frame followed by all P-frames) and used the MV of each block at the first P-frame as the first two elements of its initial state estimate, with the other two as zero, as in (12). The initial estimate of covariance $P_{1|1}$ was set to I_4 , the identity matrix of size four.

$$\hat{\mathbf{x}}_{1|1} = (v_x \ v_y \ 0 \ 0)_1^T \tag{12}$$

The predicting and updating process was triggered from the second P-frame, and the whole procedure is illustrated in Fig. 1.

The prediction of a block’s MV is straightforward using (8), while during the updating stage, to get its observation \mathbf{z}_j at frame j , we search frame j (within the search range described in the next section) to find the corresponding block using the correlation of (13) (its position plus its MV at frame j matches the position of the concerned block in frame $j - 1$) where p means the position, and thus, the available MV at frame j is the observation \mathbf{z}_j . Such a search can fail or return more than one block at frame j . These cases will be analyzed afterward. For the corrupted frame k , we will do only prediction, but no updating for we have no observations, and the predicted MVs will be used as those of the corresponding missing blocks to fulfill the concealment. A block’s position at frame k is assumed to be its position at frame $j - 1$ minus

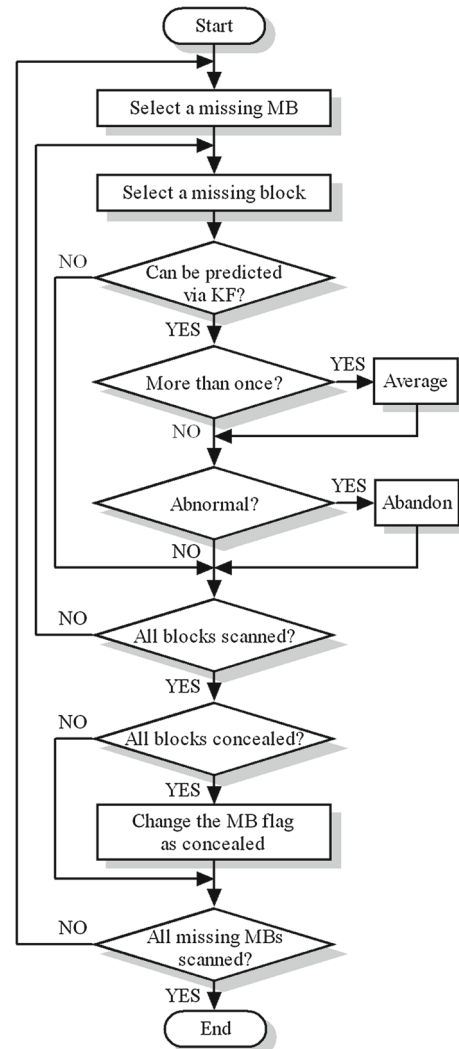


Fig. 3 Flowchart of the KF stage of our concealing scheme

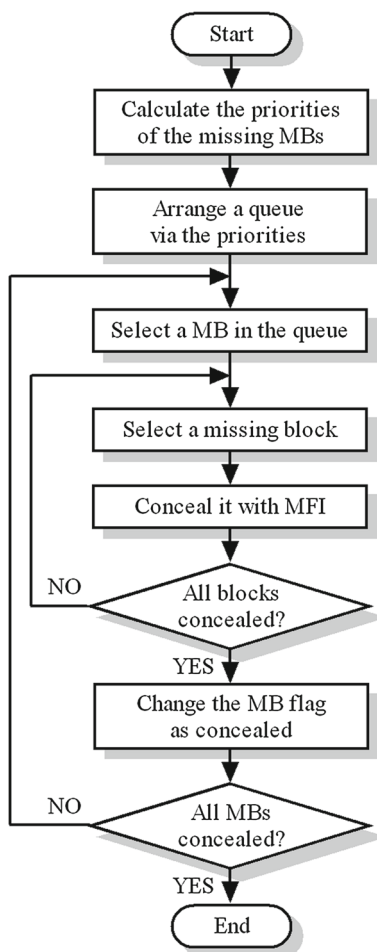


Fig. 4 Flowchart of the MFI stage of our concealing scheme

its predicted MV, i.e., $(\hat{v}_x, \hat{v}_y)_{k|k-1}$ (the first two elements of $\hat{\mathbf{x}}_{k|k-1}$), as in (14).

$$(p_x, p_y)_j + (v_x, v_y)_j = (p_x, p_y)_{j-1} \tag{13}$$

$$(p_x, p_y)_k = (p_x, p_y)_{k-1} - (\hat{v}_x, \hat{v}_y)_{k|k-1} \tag{14}$$

As partly mentioned above, there exists some complexity concerning tracking the blocks' MVs. First of all, this is a multi-target tracking process. For the QCIF format, we need to track the MVs of 1,584 blocks. As known in the tracking community, multi-target tracking is challenging due to targets' occlusions, appearing and disappearing. In the current framework, these cases occur frequently. We can take the usage of one reference frame for the P-frame as an example to simplify the descriptions (more reference frames might not change the situation qualitatively). Several blocks in frame $k - 1$ might be referenced to the same block in frame $k - 2$, and some blocks in frame $k - 2$ might not be referenced by any block in frame $k - 1$. Fortunately, since all the blocks in frame $k - 1$ can be traced back to frame $k - 2$ (for it is a P-frame), the problem can be handled through a simple trick. For the former case, we allow the blocks in frame

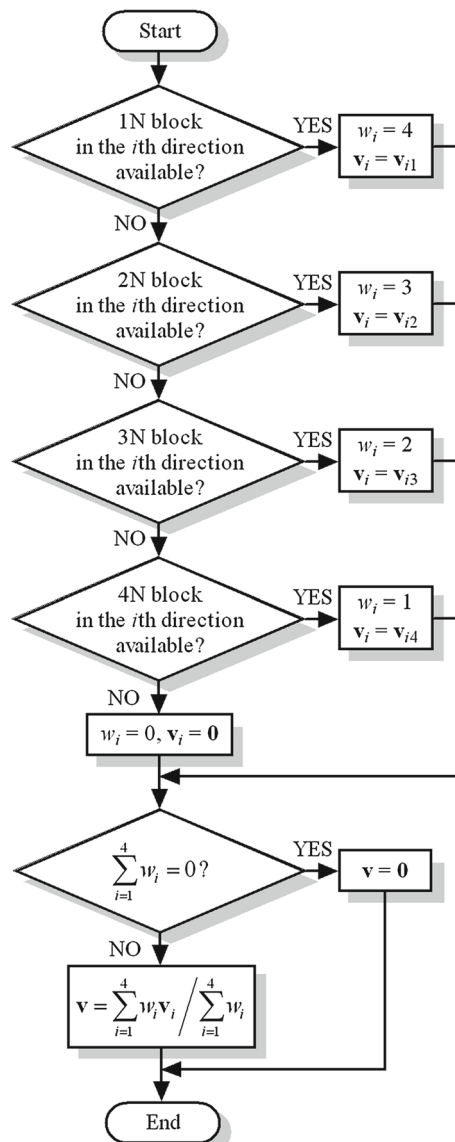
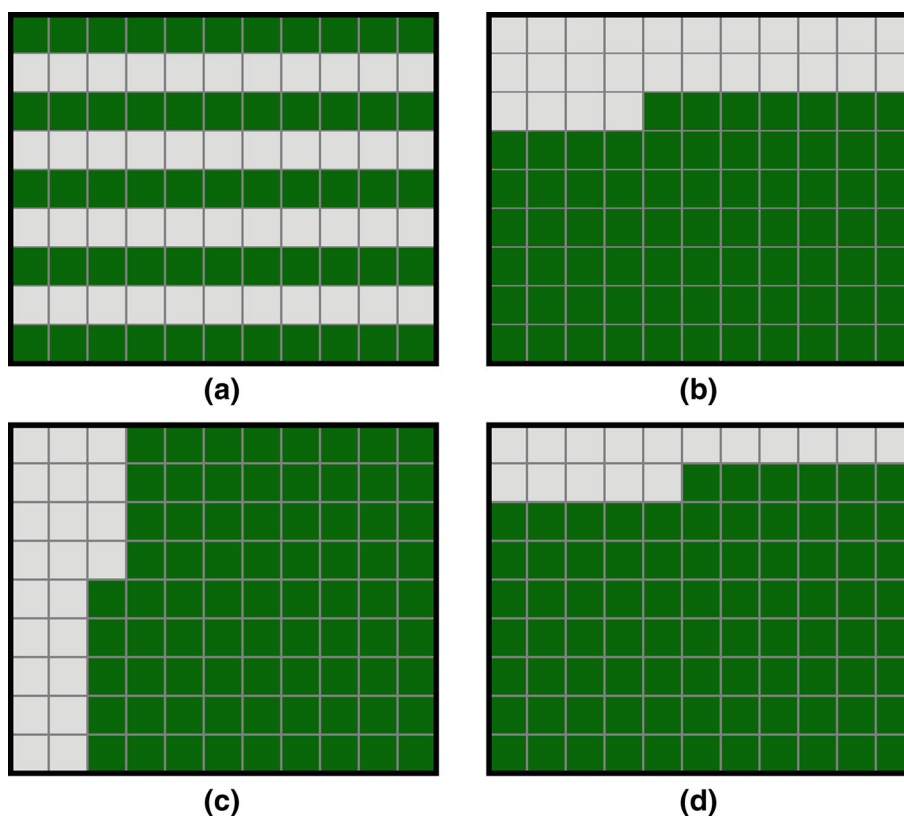


Fig. 5 Flowchart of our modified MFI scheme

$k - 1$ share the tracking history of the same reference block in frame $k - 2$, i.e., branching the tracking trajectories. For the latter case, we just terminate the tracking trajectories. As illustrated in Fig. 2, blocks #1 to #3 in frame $k - 1$ have been referenced to the same block #1 in frame $k - 2$, so we give the three blocks in frame $k - 1$ the same tracking trajectory of block #1 frame $k - 2$. The same thing holds for blocks #4 and #5 in frame $k - 1$, which correspond to the same block #2 in frame $k - 2$. Meanwhile, blocks #8 and #9 in frame $k - 2$ have never been referenced by any blocks in frame $k - 1$, so tracking them will terminate at frame $k - 2$. The remaining blocks exhibit the simple one-to-one correspondence between frame $k - 2$ and frame $k - 1$, e.g., blocks #6 and #7, and no special treatments are needed. When the tracking procedure reaches the corrupted frame k , we can

Fig. 6 The adopted slice loss patterns, corresponding to **a** MLR=55/99, **b** MLR=73/99, **c** MLR=77/99, and **d** MLR=83/99. Each *square* stands for a MB, and the *dark squares* represents lost MBs



still predict the MVs for the missing blocks (dark in Fig. 2), and then, we use these MVs to recover the blocks. Here arise two issues. Firstly, more than one tracking trajectory might point to the same missing block in frame k , e.g., blocks #1 and #4 in frame $k - 1$ point to the same missing block #1 in frame k , so it receives more than one predicted MV. As in the optical flow methods [6, 7], here we use the average of all the predicted MVs to execute the concealment. Secondly, the tracking trajectories might be insufficient to cover all the missing blocks in frame k for some blocks might receive more than one trajectory each, or some trajectories might point out of the frame and become invalid, e.g., that of block #7 in frame $k - 1$. In other words, some missing blocks in frame k cannot achieve predicted MVs from the tracking trajectories via the present KF approach. For this case, we adopt a modified MFI algorithm to finish the concealment, in contrast to the median filter employed in [6] and [7].

Another aspect is the removal of abnormal MVs, i.e., those distinctly different from their surroundings. As we know, the motion compensation technique in H.264/AVC is based on minimum residuals, and therefore, some MVs might be physically meaningless. Our tracking of such MVs might create abnormal predictions sometimes, so isolation and removal of them is necessary. We identify the abnormality of a predicted MV if the deviation of any of its two components from the available MV average of the MB where it lies is larger

than a preset threshold t , as expressed in (15). According to some experiments, $t = 16$ and $t = 4$ can bring a PSNR difference up to ± 0.5 dB (the higher can be for $t = 16$ or $t = 4$), depending on the sequence characteristics. For consistency, we choose $t = 16$ for all the sequences in the present work. However, adaptively setting this variable seems more desirable and optimal. The variable m in (15) represents the number of available (predicted) blocks in the missing MB.

$$\left| v_x - \frac{1}{m} \sum_{i=1}^m v_x(i) \right| > t \text{ or } \left| v_y - \frac{1}{m} \sum_{i=1}^m v_y(i) \right| > t \quad (15)$$

In summary, our scheme encompasses a dominant KF procedure and a supplementary MFI process. Experiments reveal that the former conceals most of the missing blocks, and the latter deals with the remaining minority. The flow-charts are presented in Figs. 3 and 4. We first conduct the KF concealment for all the missing MBs, but due to no prediction or abnormality, some blocks remain missing. During the subsequent MFI concealment, we first make a queue for the MBs still containing missing blocks according to the availability of their neighboring MBs. If a MB does not consist of any missing blocks, it is labeled as available. For a MB, we count the availability of its four neighboring MBs and set it as a priority value within 0–4. The higher

the priority value, the more forward the missing MB in the queue. For those with the same priority value, we arranged them in an arbitrary way, i.e., according to their positions in the frame, with the bottom-right prior to the top-left. We perform the MFI concealment in the order of the queue. Since our modified MFI ignores the MB boundaries, and many blocks in a corrupted MB might be available during this stage, queuing only once as adopted here seems sufficient, which is more time-saving than updating the queue frequently.

We need to explain a little on our modified MFI framework applied here, which is different from the conventional MFI method that utilizes only the boundary blocks of available neighboring MBs. Since this supplementary MFI concealment is conducted after the KF operation, a corrupted MB now usually contains only several missing blocks. The others (not necessarily on the MB boundaries) might have already been concealed in the preceding KF stage, and it seems better to utilize them during this subsequent MFI concealment. Therefore, in our modified MFI scheme, to rebuild the MV of a missing block, we neglect the MB boundaries and use the MVs of the nearest available blocks along four directions (upwards, downwards, leftwards and rightwards) within a distance range of four blocks, and set the weights according to their distances from the missing one (weights of 4, 3, 2, 1 correspond to distances of 1, 2, 3, 4 blocks, respectively). The missing MV is calculated as the weighted average of these nearest available MVs. If there are no available blocks within this range in a direction, the corresponding weight is set to zero. If all the four weights are zero, the simple FC method is employed to conceal the block. The procedure is schematized in Fig. 5, where \mathbf{v} means the MV of the missing block, $i = 1, 2, 3, 4$, indicating the four searching directions, \mathbf{v}_i is the available MV in the i th direction, w_i is the corresponding weight, 1N represents the 1st (nearest) neighboring, 2N the 2nd (next nearest) neighboring, and so on, and the MVs of the corresponding blocks (if available) are $\mathbf{v}_{i1}, \mathbf{v}_{i2}$, etc.

We considered only one corrupted frame in this work to evaluate the proposed scheme. To deal with more than one corrupted frame, additional strategies should be involved. For instance, for two inconsecutive corrupted frames, we can restart the KF tracking immediately after concealing the first one, and the performance might be similar to that by treating with them separately as here except for the effects of error propagation. As to two consecutive ones (this case is much less possible to occur than the inconsecutive case), other policies can be applied. One is to conceal the second corrupted frame using the restored MVs of the first corrupted frame. The alternative is to recover the MVs of the second corrupted frame via continuing the previous KF tracking by surpassing the first corrupted frame. Fulfilling and optimizing such ideas is just under way, and the results will appear in our future reports.

Table 1 Comparison of the average PSNR (in dB) performance

Sequence	MLR	Error	BMA	BMAr	MFI	MVE	KF0	Proposed
Foreman (41.86)	83/99	5.22	31.08	31.22	30.65	34.55	31.96	34.88
	77/99	5.60	30.98	30.99	31.62	34.02	31.55	34.53
	73/99	6.03	31.14	31.72	31.95	35.10	32.57	35.24
	55/99	6.74	33.53	33.26	35.11	35.61	33.17	35.83
Bus (40.88)	83/99	8.38	21.67	20.82	22.29	25.30	25.46	26.02
	77/99	8.81	22.17	22.00	23.28	26.81	26.55	27.64
	73/99	8.90	22.79	21.15	22.72	25.47	25.64	26.25
	55/99	10.12	27.85	25.72	26.59	27.83	27.96	28.65
Ice (44.15)	83/99	6.14	25.58	25.37	25.03	27.02	29.01	29.03
	77/99	6.57	25.25	25.32	24.26	27.08	29.05	28.87
	73/99	6.68	26.91	26.64	26.65	27.72	29.73	29.80
	55/99	7.87	30.46	29.86	28.66	28.81	30.72	30.54
Soccer (41.18)	83/99	6.11	25.17	23.57	26.00	27.54	28.50	29.10
	77/99	6.56	26.10	26.00	28.47	29.63	30.39	30.84
	73/99	6.46	26.28	24.76	26.63	27.64	28.67	29.24
	55/99	8.19	32.01	30.63	31.88	30.32	31.58	32.07
Stefan (40.94)	83/99	6.45	27.49	26.96	28.35	28.18	27.86	29.26
	77/99	6.68	26.16	25.96	24.16	27.88	27.48	28.95
	73/99	7.14	28.20	28.02	28.93	28.83	28.85	30.11
	55/99	8.19	29.23	8.59	29.53	29.35	28.88	30.61

3 Experimental results

The proposed KF-based technique was tested using the JM13.2 H.264/AVC codec. Preliminary comparisons with an existing MV extrapolation technique [9] reveal that our scheme is more suitable for sequences containing complex movements (for sequences with simple or little movements, e.g., “Coastguard”, where most MVs between adjacent frames are quite similar or the same, e.g., with value of zero, the simple MV extrapolation seems a little more appropriate than the present scheme maybe due to the introduction of process and observation noises in KF), so the standard video sequences, “Foreman”, “Bus”, “Ice”, “Soccer” and “Stefan” [28, 29] with QCIF resolution (176×144) were utilized. We adopted the IPPP frame structure with the frame rate of 30 and the quantization parameter (QP) of 22. We took advantage of the 1/4-pixel accuracy, and the number of reference frames was set as one with the search range of 32 pixels. We simulated the standard slice loss patterns as in Fig. 6, corresponding to MLR of 55/99, 73/99, 77/99, and 83/99 (99 is the total MB number in one QCIF frame), and including the interleaved, raster scan, and wipe scan FMO (flexible MB ordering) modes.

Table 2 Comparison of the average PSNR (in dB) performance on error propagation

Sequence	MLR	Error	BMA	BMAr	MFI	MVE	KF0	Proposed
Foreman (41.86)	83/99	5.27	31.41	31.53	30.96	34.80	32.33	35.15
	77/99	5.68	31.32	31.29	31.94	34.30	31.97	34.87
	73/99	6.10	31.52	32.01	32.26	35.34	32.92	35.51
	55/99	6.98	33.92	33.65	35.50	35.90	33.71	36.22
Bus (40.91)	83/99	8.73	22.23	21.47	22.78	25.79	26.18	26.56
	77/99	9.24	23.02	22.91	24.02	27.94	28.03	28.99
	73/99	9.29	23.36	21.80	23.23	26.08	26.48	26.91
	55/99	10.41	27.96	26.08	26.84	28.03	28.45	28.78
Ice (44.22)	83/99	6.19	26.10	25.90	25.53	27.64	29.71	29.67
	77/99	6.78	26.02	26.10	24.98	27.90	30.30	29.98
	73/99	6.76	27.37	27.10	27.23	28.37	30.51	30.54
	55/99	8.04	31.19	30.58	29.17	29.52	31.64	31.34
Soccer (41.21)	83/99	6.20	25.28	23.63	26.23	28.08	29.39	29.84
	77/99	6.48	25.95	25.79	28.14	29.60	30.69	31.15
	73/99	6.55	26.58	24.82	26.81	28.11	29.56	29.99
	55/99	8.37	32.80	31.17	32.45	30.91	32.49	32.85
Stefan (40.95)	83/99	6.65	28.41	28.10	29.12	28.93	28.83	30.22
	77/99	6.88	27.17	27.05	24.84	28.71	28.41	29.93
	73/99	7.32	29.03	29.11	29.62	29.64	29.86	31.14
	55/99	8.47	30.07	29.60	30.18	30.00	29.69	31.36

For comparison, we implemented several other schemes as well, including the BMA embedded in the H.264/AVC reference software, the BMA plus KF refinement [26], the conventional MFI, and the MV extrapolation [9], denoted as BMA, BMAr, MFI, and MVE, respectively. To check the necessity of considering the MV acceleration, we also conducted a scheme without the acceleration terms, labeled as KF0. The BMA implements a fixed scanning order. As for the MFI here, we introduced the simple queuing technique as that in our modified MFI described before to improve its performance. The difference was that the queue was updated after the concealment of each MB to utilize the concealed MBs as many as possible. In the BMAr approach, we adopted the covariance parameters in the literature [26]. In the MVE method, we employed the scheme in [9], but instead of median filtering, our modified MFI was used as the post-processing technique for the blocks with no or abnormal projected MVs. In other words, in this MVE, we exploited the same framework (including the projection rules, the averaging for multiple projections, and the criteria for abnormal projections) as in the proposed scheme, except that we used the projected MVs of the previous frame in MVE, rather than the KF-tracked MVs. As in the proposed scheme, the projection here was not determined via the overlapping area, but via the position

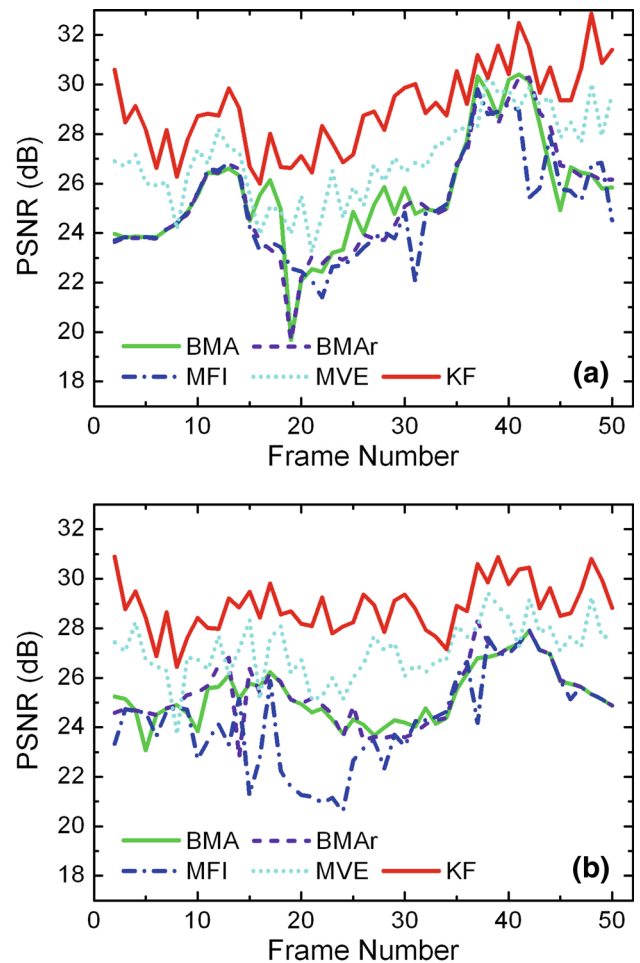


Fig. 7 PSNR versus frame number from 2 to 50 for the “Ice” sequence with **a** MLR=83/99, **b** MLR=77/99

of the block top-left corner. The block where the projected corner falls is the projected block. Our experiments show that this simple quantization yields better results than the criteria of overlapping area for the present sequences. In KF0, we used a state vector $\mathbf{x} = v_x v_y^T$, $\hat{\mathbf{x}}_{1|1} = v_x v_y^T$, $Q = 0.64I_2$, and $F = H = P_{1|1} = I_2$, i.e., without any MV acceleration term.

The average PNSR results over frames 2–50 are listed in Table 1. The values in the parentheses following the sequence names are for error-free frames, and those of the erroneous frames are provided as well. The results were obtained by successively setting the missing MBs in one frame each time. To investigate the error propagation, we observed the five frames just following the corrupted frame as well, and the average PNSR values are collected in Table 2. The corresponding error-free values are listed as well, which might be a little different from those in Table 1, for they pertain to different frames (e.g., frames #3 to #7 for the corruption of frame #2, and frames #51–#55 for corrupted frame #50). It is apparent that our proposed method outperforms all the BMA, BMAr, MFI, and MVE approaches in these cases. At

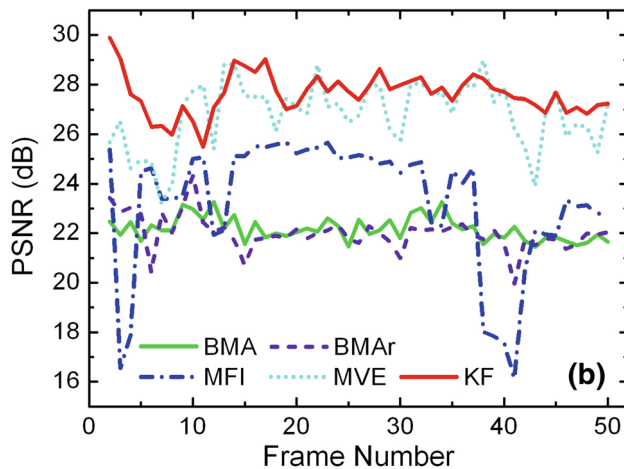
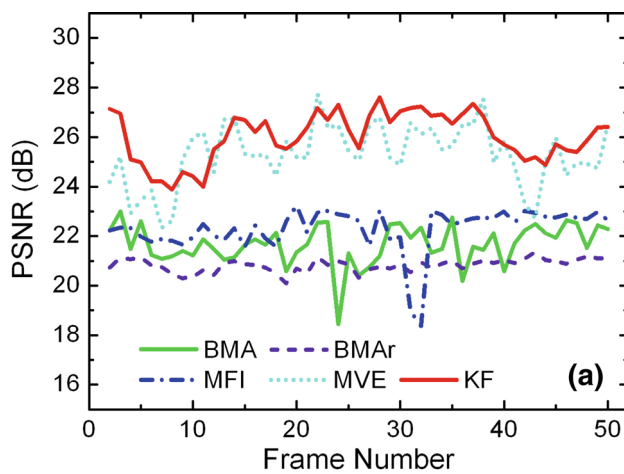


Fig. 8 PSNR versus frame number from 2 to 50 for the “Bus” sequence with **a** MLR=83/99, **b** MLR=77/99

high MLR, the differences can be beyond 5 dB. It is found that BMAR performs worse than BMA in most of current cases, indicating that the MVs of adjacent MBs in a large or complex (e.g., interleaved) slice (as here) might not always be highly correlated as assumed in BMAR, especially when the sequence contains complex movements. The KF0 results indicate that considering the MV acceleration is necessary for most cases, and in the following we will not provide further details for this simplified version of the proposed scheme. We have also tried different QP values (not shown here), and generally, lower QP yields larger differences, indicating that the present algorithm prefers high-quality images. The threshold t for checking MV’s abnormality in our proposed scheme was set to 16 here, and a value of 4 can cause a PSNR difference up to approximately ± 0.5 dB (not shown here) for these sequences, with the sign and the value dependent on the sequence features. Although some sequences seem insensitive to this threshold, for most sequences, adaptively selecting this value might further improve the performance of our approach.

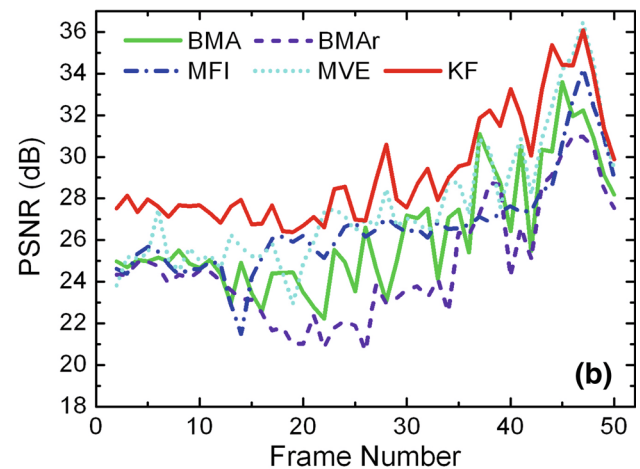
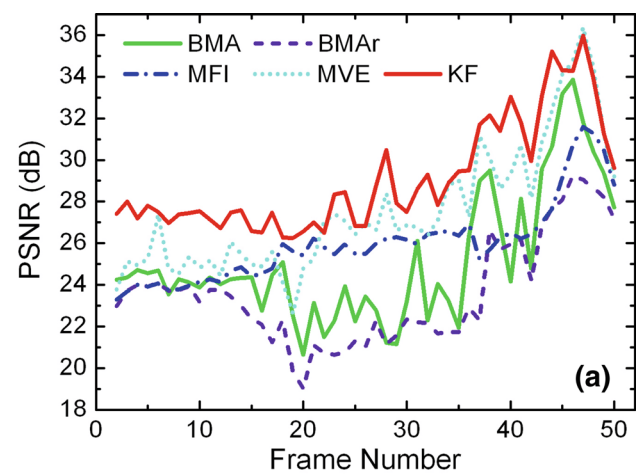


Fig. 9 PSNR versus frame number from 2 to 50 for the “Soccer” sequence with **a** MLR=83/99, **b** MLR=73/99

In Figs. 7, 8, and 9, we present the detailed comparison of PSNR at each frame from 2 to 50 for the “Ice”, “Bus”, and “Soccer” sequences, respectively, with different MLR values. The superiority of our schemes to BMA, BMAR, MFI, and MVE seems apparent here, and at some frames, the differences are drastic.

Besides the PSNR results, typical frames from “Stefan” and “Foreman” are displayed in Figs. 10 and 11, respectively, for subjective evaluation. In Fig. 10, differences seem obvious on the legs, and in Fig. 11, the different face orientations can be noticed. The wrong orientation (via BMA, BMAR, or MFI) is due to the simple FC from the previous frame. The subjective qualities are generally consistent with their PSNR differences (see the figure legends), but as widely recognized, the agreement is not always justified (e.g., Fig. 10d and f), and thus, other objective metrics such as the structural similarity (SSIM) has been put forward. The better performance of MFI over BMA in some slice loss modes might partially be attributed to the queuing technique we applied in the MFI method, which might help to take advantage of the concealed

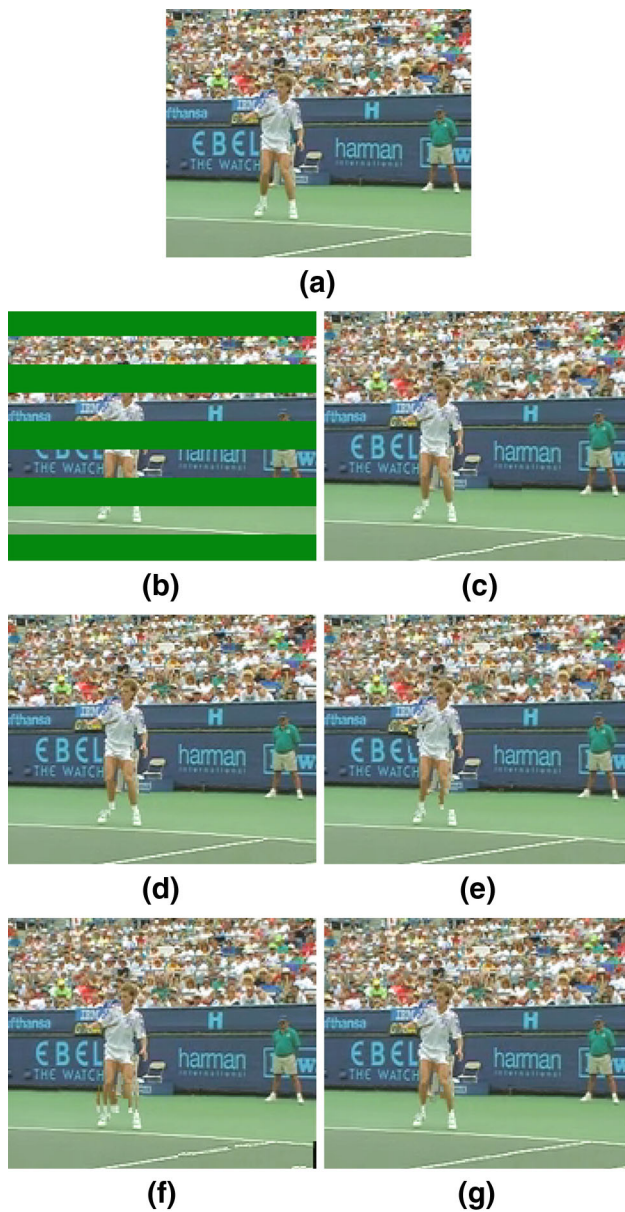


Fig. 10 Comparison of the 37th frame of the “Stefan” sequence with $MLR=55/99$. **a** Error-free frame (40.94 dB); **b** Corrupted frame (8.23 dB); **c** Concealed with BMA (30.03 dB); **d** Concealed with BMAR (30.62 dB); **e** Concealed with MFI method (29.12 dB); **f** Concealed with MVE (30.91 dB); **g** Concealed with our proposed scheme (33.49 dB)

MBs as soon and as many as possible. In contrast, the embedded BMA in the JM13.2 H.264/AVC reference codec adopts a fixed MB order for concealing.

As the last aspect, we would like to point out the computational complexity of our scheme. We recorded the time consumption as a rough estimate. On a laptop (Intel Centrino 2 GHz CPU, 2 GB memory), we find that the time consumption of the BMA, BMAR, MFI, MVE, KF0 and our technique is around 1.00:1.02:1.05:1.06:1.26:1.43. As we know, the decoding in H.264/AVC is orders of magnitude more

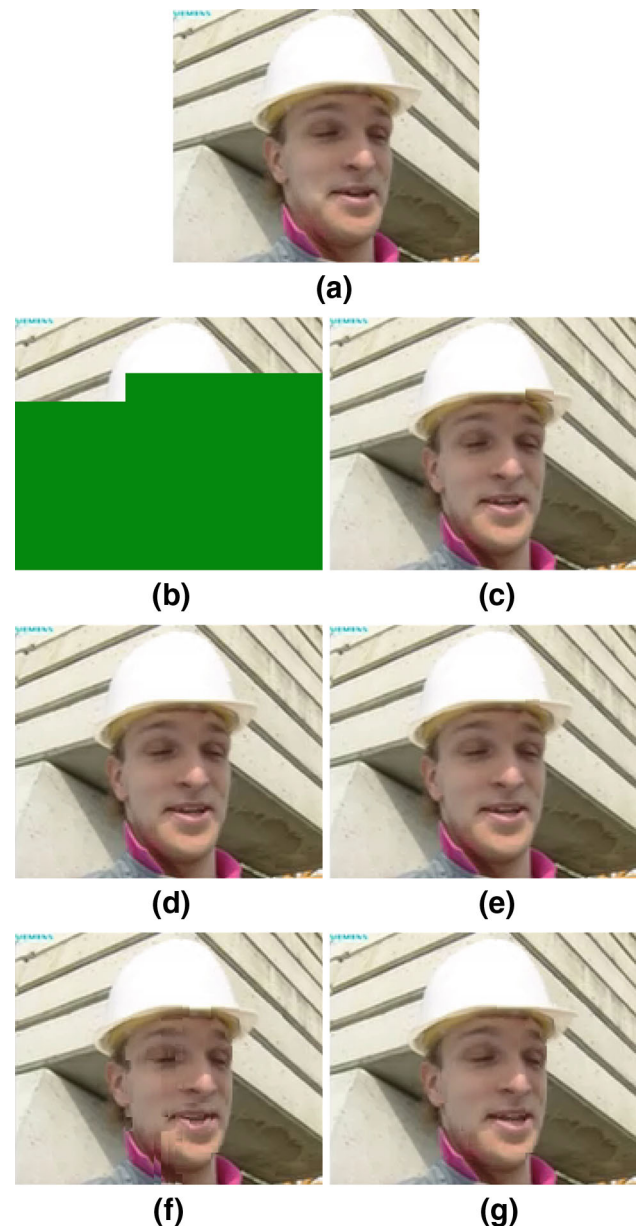


Fig. 11 Comparison of the 2nd frame of the “Foreman” sequence with $MLR=73/99$. **a** Error-free frame (41.87 dB); **b** Corrupted frame (6.11 dB); **c** Concealed with BMA (29.89 dB); **d** Concealed with BMAR (29.87 dB); **e** Concealed with MFI method (30.36 dB); **f** Concealed with MVE (33.36 dB); **g** Concealed with our proposed scheme (35.14 dB)

time-saving than the encoding, the enhanced complexity in our approach at the decoder side still seems within an acceptable range.

4 Conclusion

It is known that concealing the erroneous sequences with MLR larger than 50% is usually quite challenging, for the available spatial information is pretty limited, and the

performances of the normal temporal-spatial algorithms such as MFI and BMA have been deteriorated. Therefore, in such cases, it is desirable to make use of the temporal information as much as possible. Besides the existing techniques, in this work, we have introduced KF to the EC application, by predicting the MVs of the missing blocks using KF. For the remaining blocks (not predicted via KF, or the predicted MVs are abnormal), we proposed a modified MFI method to fulfill the complementary concealment. Experimental results show that the performance of our proposed scheme on concealing heavily corrupted sequences is encouraging. We hope that our present work could pave a new way to predicting missing MVs. To further improve the performance, we expect that the current approach can be applied to the prediction of residuals. Since KF deals only with linear motion (actually in the present work, we have assumed linear acceleration movements), to enhance the tracking accuracy and to account for nonlinear motion, the more sophisticated UKF might be more suitable though accompanied by more computational overheads.

References

- Lam, W.M., Reibman, A.R., Liu, B.: Recovery of lost or erroneously received motion vectors. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 5, pp. 417–420 (1993)
- Thaipanich, T., Wu, P.-H., Kuo, C.-C.J.: Low-complexity video error concealment for mobile applications using OBMA. In: IEEE Trans. Consum. Electron. **54**(2), 753–760 (2008)
- Al-Mualla, M.E., Canagarajah, N., Bull, D.R.: Error concealment using motion field interpolation. In: Proceeding of the 1998 IEEE International Conference on Image Processing, vol. 3, pp. 512–516 (1998)
- Al-Mualla, M.E., Canagarajah, N., Bull, D.R.: Temporal error concealment using motion field interpolation. *Electron. Lett.* **35**(3), 215–217 (1999)
- Al-Mualla, M.E., Canagarajah, N., Bull, D.R.: Motion field interpolation for temporal error concealment. In: IEE Proceedings—Vision, Image, and Signal Processing **147**(5), 445–453 (2000)
- Belfiore, S., Grangetto, M., Magli, E., Olmo, G.: An error concealment algorithm for streaming video. In: Proceedings of the 2003 IEEE International Conference on Image Processing, 3:649–652 (September 2003).
- Belfiore, S., Grangetto, M., Magli, E., Olmo, G.: Low bit-rate video based on multiframe optical flow estimation. In: IEEE Trans. Multimed. **7**(2), 316–329 (2005)
- Peng, Q., Yang, T., Zhu, C.: Block-based temporal error concealment for video packet using motion vector extrapolation. In: IEEE 2002 International Conference on Communications, Circuits and Systems and West Sino. Expositions, vol. 1, pp. 10–14 (2002)
- Wu, Z., Boyce, J.M.: An error concealment scheme for entire frame losses based on H.264/AVC. In: Proceedings of the International Symposium on Circuits and Systems, pp. 4463–4466 (2006)
- Chen, Y., Yu, K., Li, S.: An error concealment algorithm for entire frame loss in video transmission. Presented at IEEE Picture Coding Symposium (2004)
- Yan, B., Gharavi, H.A.: hybrid frame concealment algorithm for H.264/AVC. In: IEEE Trans. Image Process **19**(1), 98–107 (2010)
- Kalman, R.E.: A new approach to linear filtering and prediction problems. *J. Basic Eng.* **82**(1), 35–45 (1960)
- Lundquist, C., Orguner, U., Gustafsson, F.: Extended target tracking using polynomials with applications to road-map estimation. In: IEEE Trans. Signal Process. **59**(1), 15–26 (2011)
- Soper, T.D., Haynor, D.R., Glenny, R.W., Seibel, E.J.: In vivo validation of a hybrid tracking system for navigation of an ultrathin bronchoscope within peripheral airways. In: IEEE Trans. Biomed. Eng. **57**(3), 736–745 (2010)
- Gao, J., Kosaka, A., Kak, A.C.: A multi-Kalman filtering approach for video tracking of human-delineated objects in cluttered environments. *Comput. Vis. Image Underst.* **102**(3), 260–316 (2006)
- Yin, S., Na, J.H., Choi, J.Y., Oh, S.: Hierarchical Kalman-particle filter with adaptation to motion changes for object tracking. *Comput. Vis. Image Underst.* **115**(6), 885–900 (2011)
- Breidt, F.J., Carriquiry, A.L.: Highest density gates for target tracking. In: IEEE Trans. Aerosp. Electron. Syst. **36**(1), 47–55 (2000)
- Graetzl, C.F., Nelson, B.J., Fry, S.N.A.: dynamic region-of-interest vision tracking system applied to the real-time wing kinematic analysis of tethered drosophila. In: IEEE Trans. Autom. Sci. Eng. **7**(3), 463–473 (2010)
- Rosales, R., Sclaroff, S.A.: framework for heading-guided recognition of human activity. *Comput. Vis. Image Underst.* **91**(3), 335–367 (2003)
- Reddy, J.B.V., Dash, P.K., Samantaray, R., Moharana, A.K.: Fast tracking of power quality disturbance signals using an optimized unscented filter. In: IEEE Trans. Instrum. Meas. **58**(12), 3943–3952 (2009)
- Dorini, L.B., Goldenstein, S.K.: Unscented feature tracking. *Comput. Vis. Image Underst.* **115**(1), 8–15 (2011)
- Paul, A.S., Wan, E.A.: RSSI-based indoor localization and tracking using sigma-point Kalman Smoothers. In: IEEE J. Sel. Topics Signal Process. **3**(5), 860–873 (2009)
- Dominguez, S.M., Keaton, T., Sayed, A.H.: A robust finger tracking method for multimodal wearable computer interfacing. In: IEEE Trans. Multimed. **8**(5), 956–972 (2006)
- Zhang, X., Hu, W., Qu, W., Maybank, S.: Multiple object tracking via species-based particle swarm optimization. In: IEEE Trans. Circuits Syst. Video Technol. **20**(11), 1590–1602 (2010)
- Lanz, O.: Approximate Bayesian multibody tracking. In: IEEE Trans. Pattern Anal. Mach. Intell. **28**(9), 1436–1449 (2006)
- Gao, Z.-W., Lie, W.-N.: Video error concealment by using Kalman-filtering technique. In: Proceedings of the International Symposium on Circuits and Systems, vol. 2, pp. 69–72 (2004)
- Mochnáč, J., Marchevský, S., Gamec, J.: Kalman filter based error concealment algorithm. In: Proceedings of 54th Internationales Wissenschaftliches Kolloquium, pp. 1–4 (2009)
- <http://trace.eas.asu.edu/yuv/index.html>
- <http://media.xiph.org/video/derf>