Adaptive De-interlacing Algorithm Based on Motion Compensation

Gu Junxia[1,2], Gao Xinbo[1], Hiroyuki Ishimura[3]

1. School of Electronic Engineering, Xidian University, Xi'an China 710071
2. Department of Electronic Engineering, Tsinghua University, Beijing China 100084
3. Smart Vision Multimedia Consulting Corporation, Tokyo Japan

Email: xbgao@lab202.xidian.edu.cn

Abstract: An adaptive de-interlacing algorithm based on motion compensation is presented in this paper. In this algorithm, the motion blocks are detected accurately by using successive 4-field images. With the Kalman filter, the motion estimation module searches motion vector only for motion blocks, and the search model is adaptive to motion velocity and acceleration. Two de-interlacing methods are adopted to satisfy the different requirements of motion blocks and static blocks. Compared with full search algorithm, the proposed algorithm greatly reduces the computational amount with keeping the performance approximately.

Key Words: De-interlace, adaptive motion estimation, Kalman filter, motion compensation

1. Introduction

In the current worldwide television system, the interlaced scanning technique is widely adopted to reduce the bandwidth burden of the video signals. But a major drawback of the interlacing on the current bright high-resolution displays is the line flicker and jagged effects of moving edges. Thus, many de-interlacing algorithms have been proposed to reduce these artifacts.

Among the existing de-interlacing algorithms, motion compensation (MC) is the most promising method. As due to the inertia, it always takes time for objects to completely disappear, or geometrically distort, resulting a strong correlation of successive images. The motion compensation algorithms attempt to interpolate in the direction with the highest correlation, *i.e.*, interpolation along the motion trajectory.

In the MC algorithm, block-matching algorithm has been widely adopted for motion estimation. In theory, full search block-matching method can find the optimal solution, but it is hard to deal with visual sequence in real-time because of high computational complexity. Many fast block-matching algorithms have been proposed [1], such as three steps search (TSS), 2-D logarithmic search (TDL), orthogonal search algorithm (OSA), one at a time search (OTS), cross search algorithm (CSA) and hierarchical block matching algorithms (HBMA). If there is real motion, the searched motion vector can be directly used to compensate. However, the estimated motion vector is always inaccurate, so a temporal median filtering is often employed for further post-processing.

For this purpose, this paper presents an adaptive de-interlacing algorithm with better performance based on motion compensation, which consists of the motion detection with 4-field images and the motion estimation with the Kalman filtering.

2. Adaptive De-interlacing Algorithm Based on Motion Compensation

A large percentage of static blocks are encountered in video sequences [2]. For static blocks, field repetition algorithm is the best de-interlacing method. While for motion blocks, motion compensation can improve the de-interlacing effect. This suggests us to separately deal with the static and the motion parts, which forms the adaptive de-interlacing algorithm based on motion compensation. It follows three processing steps: motion blocks detection, adaptive motion vector estimation based on spatio-temporal correlation, de-interlacing. The block diagram of this algorithm is shown in Figure 1, where flag1, flag2 and flag are the motion block marker matrices.

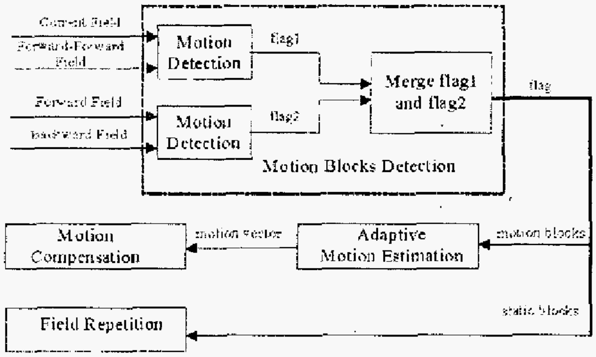

Figure 1 Block diagram of the proposed de-interlacing method

2.1 Motion Block Detection

The motion blocks are identified by computing the mean matching difference (MMD) between the blocks to be detected in the current field and the corresponding blocks in the referenced field. If the MMD is bigger than a given threshold, the block will be considered a motion block. To make more accurate detection of the motion blocks, the same-parity 4-field images are needed [3].

2.1.1 Choice of matching criteria

A matching criterion, or distortion function, is used to measure the similarity between the target block and candidate blocks. As well known, the mean square difference function is too complex to be used in real-time. While, the mean absolute difference function is simple, but easy to be interfered by noise. So, a double threshold (DT) criteria is proposed, which defines as

$$f(m,n)=\begin{cases}1 & \text{if}\left[\sum_{p=1}^{W}\sum_{q=1}^{H} ord\left(|A_{mn}(p,q)-B_{mn}(p,q)|>\alpha_1\right)\right]>\alpha_2 \\ 0 & \text{else}\end{cases} \quad (1)$$

where (m, n) is the block number, f is the motion marker matrix, A_{mn} and B_{mn} the luminance matrices of compared blocks respectively, W and H the block width and block height respectively, α_1 and α_2 the thresholds, and $ord(x)$ the logic function.

2.1.2 Choice of threshold and post-processing

In motion detection, the choice of threshold is a difficulty. Any erroneous detection will cause artifacts as spots in the video. So *over detection* is employed to detect motion blocks as many as possible. Then post-processing is performed on the motion marker matrix to remove the false alarms. By many experiments, we conclude that for DT criteria the choice of threshold is as follows:

$$\begin{cases}\alpha_1=1 & \alpha_2\in \text{round}([0.3,0.35]*W*H)\\ \alpha_1=2 & \alpha_2\in \text{round}([0.2,0.3]*W*H)\\ \alpha_1=3 & \alpha_2\in \text{round}([0.15,0.25]*W*H)\end{cases} \quad (2)$$

Morphological filtering, which includes the opening and closing operations, can eliminate the background noise of the over-detected motion marker picture. The opening is used to reduce the noise, while the closing is used to fill up holes in the motion object. Many times of opening and closing process operated alternately can improve the quality of post processing. From Figure 2, which shows the detected motion blocks picture of '*golf*' sequence, we can conclude that the post-processing reduces noise (small black dots) and fills up some holes in the motion object.

(a) (b)

Figure 2 Post Processing of the motion blocks detection: (a) the original picture of over-detected motion blocks; (b) the post-processed Picture

2.2 Adaptive Motion Vector Estimation Based on Spatio-temporal Correlation (AME-BSTC)

In spatial domain, if two adjacent blocks, such as the current block and its left block, belong to the same motion object, their motion vectors may be similar [2], and in temporal domain, motion vectors of motion objects have the characteristic of leaning to center [4]. By the spatial or temporal correlation, motion vector can be predicted. However, the algorithm based on spatial correlation will be false if the adjacent blocks do not belong to the same motion object, while algorithm based on temporal correlation depends on the previous estimation precision. So the two algorithms combine to make one better algorithm based on spatio-temporal correlation.

(1) Prediction of search center and search window

Here, Kalman filter is served as a backward prediction, because motion estimation is searching matching block in the previous field. The state vectors of the Kalman filter are defined as follows.

$$\text{Initial state: } X_k = [x_k, y_k, \hat{u}_k, \hat{v}_k]^T;$$

$$\text{Next state: } X'_k = [x'_k, y'_k, u'_k, v'_k]^T$$

Where (x_k, y_k) and $(\hat{u}_k, \hat{v}_k)$ are respectively the position and predicted velocity of some motion block in the field k; (x'_k, y'_k) and (u'_k, v'_k) are respectively predicted position and velocity of the matching block in the field k-1. In temporal domain, we are implicitly assuming that the object motion is almost linear within the field interval, *i.e.*, $\hat{u}_k = u'_k, \hat{v}_k = v'_k$, and in spatial domain, predicted velocity is equal to the velocity of its left block.

The state transition matrix A and the measurement matrix C are given by

$$A = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ and } C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

respectively (since they are time independent, we omitted the time subscript).

The state function and the measurement function are defined by function (3) and (4) respectively.

$$X'_k = AX_k + \omega_k \tag{3}$$

$$Y_k = CX'_k + V_k \tag{4}$$

with $\omega_k = [-a, -b, 0, 0]^T$ the input white noise and (a, b) for acceleration of motion block, and $V_k = [v_x, v_y]^T$ the observation white noise. We assume that in every direction acceleration is the homogeneous distributing, with $1/(2M)$ the probability density. Then the covariance matrix, Q_k and R_k, are given as follows.

$$Q_k = \begin{bmatrix} \frac{1}{3}M^2 & 0 & 0 & 0 \\ 0 & \frac{1}{3}M^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \text{ and } R_k = \begin{bmatrix} \sigma_x^2(k) & 0 \\ 0 & \sigma_y^2(k) \end{bmatrix};$$

Then the next state can be calculated by Kalman recursion function.

To the state vector it is associated the state covariance matrix P_k that encodes the uncertainty of the current state; the region of the phase space centered on the estimated state which contains the true state with a given probability c^2 is given by the ellipsoid.

$$(X'_k - \hat{X}'_k)(P_k)^{-1}(X'_k - \hat{X}'_k)^T \le c^2 \tag{5}$$

In the block-matching algorithm, the elliptical search region is too complex, so it is transformed into a rectangular search region by searching the leftmost, the rightmost, the highest and the lowest points.

(2) Adaptive multi-model search

Search model is categorized into three groups by the absolute difference, d, between temporal predicted motion vector and spatial predicted motion vector:

a) $\max(d) \le d_1$: the current motion block and its spatial adjacent block belong to the same motion object. The search model is based on spatial correlation.

b) $d_1 < \max(d) \le d_2$: compare the current block and the spatial predicted block and temporal predicted block respectively, and then choose the better matching block as the search center and the search model is correspondingly based on temporal or spatial correlation.

c) $\max(d) > d_2$: the prediction is wrong, and full search method is adopted.

Here d_1 and d_2 are the threshold values determined empirically.

In the search based on temporal or spatial correlation, the search model varies with absolute value of acceleration, a: when $\max(a) \le \beta_1$, the motion vector is evaluated by the predicted value; when $\beta_1 < \max(a) \le \beta_2$, search with TSS; when $\max(a) > \beta_2$: search with FS. Here β_1 and β_2 are both experiential thresholds. In TSS or FS search, the search center and search region are calculated by equation (3) and (5) respectively.

3. Experimental Results and Analysis

The MC-based de-interlacing algorithm requires 4-field memory, which increases the hardware costs, but the performance is better and the computational amount is less than the available MC algorithms. In this section, the proposed algorithm and two classical MC algorithms with TSS and FS search will be mutually compared.

3.1 Evaluation Method

Several methods can be used to evaluate the de-interlaced results ranging from objective to subjective evaluation. We prefer to use the objective measurement.

3.1.1 With progressive scanning sequence

Nowadays, there are many progressive sequences, with which it is easy to evaluate the performance of the tested de-interlacing algorithms. Thus, two criterions are defined as follows.

Error Ratio is defined as the percent of different element between the original progressive frame, *A*, and the de-interlaced image, *B* with size of *M**N*.

$$R_f = \frac{\sum_{p=1}^{M}\sum_{q=1}^{N}\sum_{r=1}^{3}(ord[(A(p,q,r)-B(p,q,r))\neq 0])}{M*N*3} \quad (6)$$

Significant Error Ratio is defined as the percent of different element, between *A* and *B*, whose absolute difference is greater than a given threshold, *t*.

$$R_g = \frac{\sum_{p=1}^{M}\sum_{q=1}^{N}\sum_{r=1}^{3}(ord[|A(p,q,r)-B(p,q,r)|>t])}{M*N*3} \quad (7)$$

3.1.2 With interlaced scanning sequence

Because the motion compensation algorithm with FS method can obtain optimal performance, other algorithms can be evaluated by being compared with FS method.

3.2 Experimental Results

In experiment, a progressive sequence, *'boy'*, and an interlaced sequence, *'golf'*, are selected as test-bed. They are both containing abundant motion information, such as static background, slow motion and fast motion.

Table 1 and Table 2 respectively show the experiment results of three algorithms with the test video sequences. In the two experiments, the value of each parameter is given as follows: the size of sub-block is 4*6 *pixel*; the size of search window for FS and TSS is 6 *blocks*; $c^2=0.9$, $M=10blocks$, $\sigma_i^2(0)=\sigma_j^2(0)=1$, $\alpha_1=1$, $\alpha_2=8$, $\beta_1=2pixels$, $\beta_2=10pixels$, $d_1=2blocks$, $t=6$, $d_2=5blocks$. From the two tables, we can find that the proposed de-interlacing algorithm greatly reduces the search amount and keeps the same performance with the full search algorithm.

Table 1 Result of *'boy'* sequence

Algorithm	Search amount	Error ratio (%)	Remarkable error ratio (%)
FS	1	41.20	7.41
TSS	0.0011	44.15	11.12
AME-BSTC	0.0018	43.36	8.51

Table 2 Result of *'golf'* sequence

Algorithm	Search amount	Error ratio (%)	Remarkable error ratio (%)
FS	1	[illegible]	[illegible]
TSS	0.0011	18.67	4.54
AME-BSTC	0.0018	13.13	1.43

4. Conclusions

This paper presents an adaptive de-interlacing algorithm based on motion compensation, which processes the static blocks and the motion blocks separately in a field to reduce search computational amount. Experiment results show that this algorithm greatly reduces the search amount and keeps the performance with the full search algorithm.

Acknowledgements

This project was supported by the National Natural Science Foundation of China (N0.60202004)

References:

[1] http://www.newmediarepublic.com/dvideo/compression.

[2] Yao Nie and Kai-Kuang Ma. Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation. *IEEE Trans. Image Pro.*, 2002, 11(12): 1442-1449.

[3] Shyh-Feng Lin, Yu-Ling Chang and Liang-Gee Chen. Motion Adaptive Interpolation with Horizontal Motion Detection for Deinterlacing. *IEEE Trans.Consumer Electronics*, 2003, 49(4): 1256-1265.

[4] Qin Bo, Wu Yuan-ming, Yang Jian-jun, et al. An Interlace/Progressive Transforming Algorithm Based on Motion Estimation. *Journal of UEST of China*, 2002, 31(2): 131-135. (in Chinese)