A Motion Compensated De-interlacing Algorithm for Motive Object Capture

Lei Gao, Chao Li, Chengjun Zhu, and Zhang Xiong

School of Computer Science and Engineering, Beihang University,
37th Xueyuan Road,Haidian District, Beijing, China, 10083
squirelive@gmail.com, licc@buaa.edu.cn, zhucj@cse.buaa.edu.cn,
xiongz@buaa.edu.cn

Abstract. A motion compensated de-interlacing algorithm is proposed to recover the defects of interlaced video frame for capturing motion object. In this algorithm, two anti-noise background fields are formed by analyzing the temporal correlation of pixels between adjacent same parity fields. To each field, the subtraction with the corresponding background is used to detect motion object. To avoid the inaccurate detection caused by the difference between the spatial scanning positions of odd and even field, the motion objects are detected with same parity field and background field. Then motion estimation technology is used to measures the inter-field motion, find out the motion vector between the odd field and even field. Based on the motion vector, an interpolation filter is designed to shift the pixels of the motion object in the two temporally displaced fields to a common point in time. This de-interlacing algorithm maximizes the vertical resolution of the motion objects. Experimental results show that the proposed algorithm could achieve higher image quality on motion object, and the computational complexity is acceptable for consumer computer applications.

Keywords: de-interlacing, motion compensation, motion estimation, motion detect, motion object.

1 Introduction

The conventional television system based on interlaced video signal has been widely used for many decades. There are many systems transmit or record video in PAL or NTSC format. The interlaced video signal means that during the image capture process, the camera outputs the odd lines at one instant in time, and then several milliseconds later, outputs the even lines. Odd lines make up odd field and even lines make up even field. Interlacing reduces the bandwidth by half, is a way to display the nonmoving parts with full resolution and the moving parts with half resolution, but fluidly. It is a very clever way to cut bandwidth without sacrificing much quality [1]. But in some cases it is necessary to get one full frame from the two fields. A traffic surveillance and automobile license plate identification system, for example, is required to capture moving objects and to allow viewing of details within a moving image.

V.G. Duffy (Ed.): Digital Human Modeling, HCII 2007, LNCS 4561, pp. 74–81, 2007.

Fig. 1. An example of interlaced effect on moving object

Because of a time delay between two half-frames which is approximately 16ms for NTSC standard and approximately 20 ms for PAL and SECAM standards [2], a temporal shift occurs between the odd and even lines of the image. After combing two adjoining half-frames an interlaced effect appears on moving object’s edges. This problem is illustrated in Figure 1.

Therefore, the de-interlacing process attempts to remove interlace effect by creating a clean frame from the two fields. Many de-interlacing algorithms have been proposed [3-7]. These methods can be classified into two basic categories: motion compensated and non-motion compensated. When restoring a full frame from two half-frames, most of non-motion compensated methods base on one of the half-frames, and calculate moving pixels which need to be corrected from the basic one using an interpolation method. These mathematical methods lead inevitably to information loss right up to losing all the information of one of the half-frames [6], especially in regions with significant motion. In contrast with non-motion compensated method, motion compensated de-interlacing using maximum information of both half-frames. It is the most advanced method of de-interlacing [7], but it requires significant compute resources to implement.

In this paper, we present a low computation requirement de-interlacing algorithm based on motion compensation. It adapts to divide quick moving object from background and interpolate moving pixels based on motion vector. Comparing with conventional methods, it produces higher image quality on motive object.

The organization of this paper is as follows. The motion compensated de-interlacing is reviewed in section 2. The proposed algorithm is presented in section 3. Experiment results are presented in section 4. Finally, our conclusion is given in section 5.

2 Motion Compensated De-interlacing

Since the position of a moving object will be different in the two half-frames, if one measure the inter-filed motion vector and then align data between the two half-frames, more inter-field information can be used. Motion compensated de-interlacing is the technique attempts to interpolate pixels base on motion vectors.

Motion compensated de-interlacing can be symbolized as:

$$F(\vec{x},n)=\begin{cases} f(\vec{x},n), & y \bmod 2 = n \bmod 2 \\ f(\vec{x}-\vec{d}(\vec{x},n),n-1), & otherwise \end{cases} \tag{1}$$

where $f(\vec{x},n)$ is the brightness value of the pixel in field n, at position $\vec{x}=[x,y]^t$, t for transpose, $f(\vec{x},n)$ the input field and $F(\vec{x},n)$ the de-interlaced output frame, $\vec{d}(\vec{x},n)=(d_x(\vec{x},n),d_y(\vec{x},n))^t$ for the motion vector at spatial position $\vec{x}$ in field n.

Obviously, the accuracy of motion vector $\vec{d}(\vec{x},n)$ is key factor to de-interlace frame correctly. In general, finding out the motion vector is typically implemented as a block-matching motion estimation process. Typical block sizes range from 4x4 to 8x8. However, the complexity of real-time block matching is significantly higher. The extra hardware is required to perform the complex motion estimation and compensation process. For example, real-time motion estimation for CIF (352 x 288) 30 frames per second (fps) video with [-16, +15] search range requires 9.3 Giga-operations per second (GOPS). If the frame size is enlarged to D1 (720 x 480) 30fps with [-32, +32] search range, 127 GOPS is required [8]. To reduce computation in real-time system, an improved motion compensated de-interlacing algorithm for motive object capture is proposed.

3 The Proposed Motion Compensated De-interlacing Algorithm

The proposed algorithm consists of three stages, that is, motive object detection, motion estimation and motion compensated de-interlacing filtering. We will discuss them in detail in the following.

3.1 Motive Object Detection

Motive object detection is the essential pre-stage of motion estimation. In order to extract motive objects, background subtraction method is adopted. Widely used in surveillance system, background subtraction is an efficient method to discriminate moving objects from the still background [9]. The idea of background subtraction is to subtract the current image from the still background, which is acquired before the objects move in.

To avoid the inaccurate detection caused by the difference between the spatial scanning positions of odd and even field, the proposed method form two anti-noise background fields by capturing the statistics from the first N background frames:

$$B_0(\vec{x}) = \frac{1}{N}\sum_{k=0}^{N-1} f(\vec{x},2k)$$
$$B_1(\vec{x}) = \frac{1}{N}\sum_{k=0}^{N-1} f(\vec{x},2k+1) \quad (2)$$

where $B_0(\vec{x})$ is the brightness value of the pixel in odd background field, at position $\vec{x} = [x, y]^t$ and $B_1(\vec{x})$ is the even one. Then the motion objects located displaced in adjoining fields are detected by subtracting the current field from background field with same parity. The brightness difference value $D_0(\vec{x})$ and $D_1(\vec{x})$ are defined as:

$$D_0(\vec{x}) = \left| f(\vec{x},n) - B_0(\vec{x}) \right| \quad n \bmod 2 = 0$$
$$D_1(\vec{x}) = \left| f(\vec{x},n) - B_1(\vec{x}) \right| \quad otherwise \quad (3)$$

If $D_0(\vec{x})$ or $D_1(\vec{x})$ is larger than a given threshold, current pixel $\vec{x}$ is classified to the motive object. Otherwise, the pixel $\vec{x}$ belongs to the background. Then an assumable motion vector can be calculated by comparing the centroids of the motion object located displaced in the two half-frame.

3.2 Motion Estimation

As described in previous section, the accuracy of motion vector $\vec{d}(\vec{x},n)$ is key factor to de-interlace frame correctly. The goal of the motion estimation is to get an accurate motion vector of inter-field object moving. To achieve this goal, an improved block-matching method is adopted.

Considering low computation requirement, only one block within the area of motive object is chosen in the even field, and the motion of the block in the even field is estimated by matching to candidate blocks in the previous odd field. The size and position of the block depend on the size and the user most interested part of motive object. In order to reduce search positions, the search positions are calculated base on the assumable motion vector which was found in motive object detection stage. Then a search of lowest brightness distortion block among the entire candidate blocks is performed.

The proposed block-matching can be expressed as:

$$SAD(x,y)=\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left|f([i,j]^t,n+1)-f([i+x,j+y]^t,n)\right|,\quad n \bmod 2=0$$
$$x_a-p\le x\le x_a+p-1$$
$$y_a-p\le y\le y_a+p-1 \tag{4}$$

$$\vec{d}(\vec{x},n+1)=\min(SAD(x,y)) \tag{5}$$

where $SAD(x,y)$ is the sum of absolute differences of the candidate block at search position (x,y), the block size is $M\times N$, $\{f([i,j]^t,n+1)|0\le i\le M-1, 0\le j\le N-1\}$ means current block data in even field $n+1$, $\{f([i+x,j+y]^t,n)|0\le i\le M-1, 0\le j\le N-1\}$ means candidate block data at search position (x,y) in odd filed n, $[-p,p-1]$ is the search range, $[x_a,y_a]$ is the assumable motion vector which was found in motive object detection stage, and $\vec{d}(\vec{x},n+1)$ is the motion vector of current block with minimum *SAD* among $(2p)^2$ search positions.

3.3 Motion Compensated De-interlacing Filter

In the previous two sections, the inter-field motion vector of motive object has been calculated by the proposed method. Based on the motion vector, an interpolation filter is designed and the missing pixels in motive object could be interpolated by applying median filter on the spatial neighbors in the vertical direction and the predict position in the next even field.

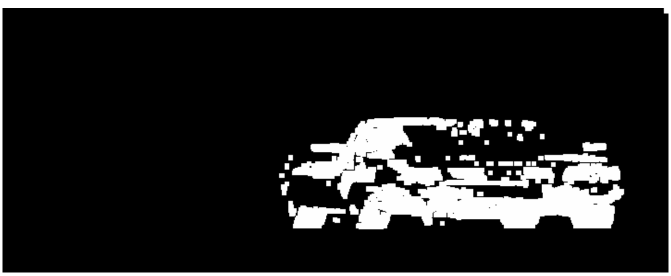
The final output can be defined as:

$$F(\vec{x},n)=\begin{cases} f(\vec{x},n), & y \bmod 2=0, n \bmod 2=0 \\ median\begin{pmatrix} f(\vec{x}-\begin{bmatrix}0\\1\end{bmatrix},n) \\ f(\vec{x}+\begin{bmatrix}0\\1\end{bmatrix},n) \\ f(\vec{x}-\vec{d}(\vec{x},n+1),n+1) \end{pmatrix}, & y \bmod 2=1, n \bmod 2=0 \end{cases} \tag{6}$$

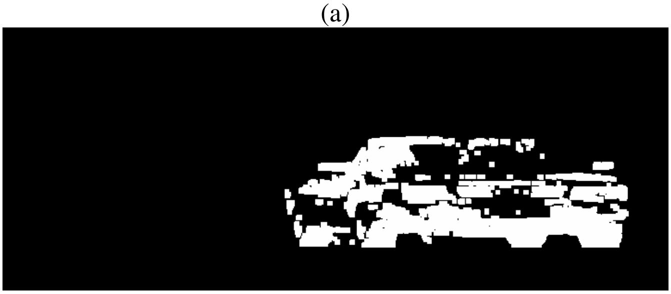
4 Experimental Results

The proposed algorithm was simulated on a Celeron CPU 2.53GHz PC for performance evaluation. We used interlaced video sequences of spatial size 720 x 576

Fig. 2. An interlaced frame of the test sequences including a fast moving object



(a)



(b)

Fig. 3. Motive object detection results: (a) odd field, (b) even field

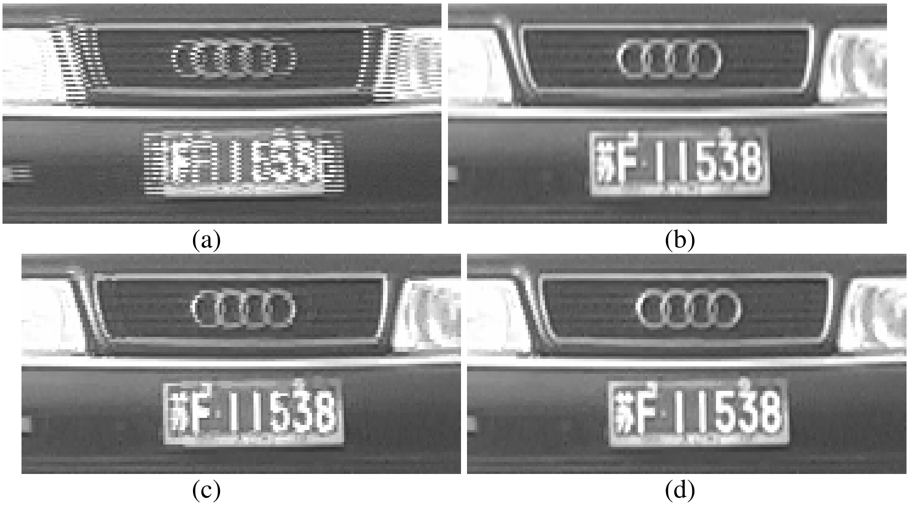


Fig. 4. Partial image of motive object: (a) original, (b) line average, (c) median filtering, (d) proposed method

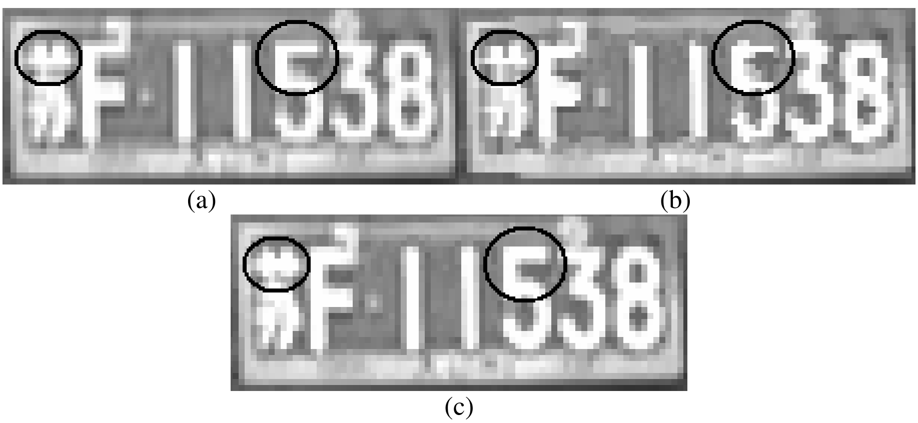


Fig. 5. Enlarged image of license plate areas: a) line average, (b) median filtering, (c) proposed method

which are captured on road for testing. The average processing speed is over 25 frames per second. For comparison, tow simple de-interlacing methods, line average and median filtering, were applied to same sequences.

Figure 2 is an interlaced frame of the test sequences including a fast moving object. The motive object detection results are shown in Figure 3. The de-interlacing results on this moving object are shown in Figure 4. Figure 5 are enlarged images of license plate areas. It can be seen that the image quality of the proposed method outperforms the line average and median filtering method.

5 Conclusion

The proposed motion compensated de-interlacing algorithm employed maximum information of both half-frames to interpolate the missing pixels of the motive object, maximized the vertical resolution of the motion objects. It could achieve higher image quality on motion object, and the computational complexity is acceptable for consumer computer applications.

References

1. What is Deinterlacing? Facts, solutions, examples http://www.100fps.com/
2. Wang, Y., Ostermann, J., Zhang, Y.-Q.: Video processing and communications. Prentice Hall, Upper Saddle River, NJ (2002)
3. Kuo, C.J., Liao, C., Lin, C.C.: Adaptive Interpolation Technique for Scanning Rate Conversion. IEEE Transactions on Circuits and System for Video Technology 6(3), 317–321 (1996)
4. Lin, S.F., Chang, Y.L., Chen, L.G.: Motion Adaptive Interpolation with Horizontal Motion Detection for De-interlacing. IEEE Transactions on Consumer Electronics 49(4), 1256–1264 (2003)
5. Tai, S.C., Yu, C.S., Chang, F.J.: A Motion and Edge Adaptive Deinterlacing Algorithm. In: IEEE International Conference on Multimedia and Expo (ICME), 27-30 June 2004, vol. 1, pp. 659–662 (2004)
6. Khudeev, R., Osipov, S., Afanasyev, V.: De-Interlacing Using Area Based Motion Compensation. In: Confernence on Control and Communications SIBCON, 2005, 21-22 October 2005, pp. 154–160 (2005)
7. Deame, J.: Motion Compensated De-Interlacing: The Key to Digital Video Transition. In: SMPTE 141st Technical Conference in NY (November 19-22, 1999)
8. Huang, Y.-W., Chen, C.-Y., Tsai, C.-H., Shen, C.-F., Chen, L.-G.: Survey on Block Matching Motion Estimation Algorithms and Architectures with New Results. Journal of VLSI Signal Processing 42, 297–320 (2006)
9. Pan, J., Lin, C.-W., Gu, C., Sun, M.-T.: A robust video object segmentation scheme with pre-stored background information. In: IEEE International Symposium on Circuits and Systems (ISCAS), 2002, 26-29 May 2002, vol. 3, pp. 803–806 (2002)