An Improved 3DRS Algorithm for Video De-interlacing

Songnan Li[1], *Jianguo Du*[1], *Debin Zhao*[1,2], *Qian Huang*[2,3], *Wen Gao*[2,3]

[1]Department of Computer Science and Engineering, Harbin Institute of Technology, Harbin 150001, China
[2]Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China
[3]Graduate School, Chinese Academy of Sciences, Beijing 100039, China

Abstract. In this paper, an improved 3D recursive searching (3DRS) algorithm for video de-interlacing is proposed. This algorithm solves the error propagation problem which exists in the original 3DRS by using bi-directional block matching. It also improves the precision of motion estimation by using multi-resolution searching to adjust the matching block size according to the video content. Experiments show that the proposed algorithm generates better subjective and objective results than the original 3DRS algorithm on most tested sequences.

Index Terms—video de-interlacing, 3D recursive searching, bi-directional block matching, multi-resolution searching

1. INTRODUCTION

Traditional television systems adopted interlaced scan as a tradeoff between frame rate and transmission bandwidth requirements [1]. However, interlaced scan suffers from a number of artifacts, such as interline flicker, inter-field flicker, twitter and serration [2]. It also makes video coding less efficient and more complex [3], while progressive scan, which increases spatio-temporal correlations within and between frames, simplifies the video codec and improves the coding efficiency. Moreover, nearly all the modern cameras and displays are progressive, whereas most of the video scan formats approved by the ATSC [4] in actual use are interlaced. Therefore the demand for conversion from interlaced videos to progressive ones grows day by day since people have piled up hundreds of thousands of valuable interlaced videos during the past decades. So nowadays, de-interlacing as a picture format conversion from interlaced to progressive pictures has been widely used to overcome the disadvantage caused by interlaced scan.

Over the last two decades, many de-interlacing algorithms have been proposed. They range from simple interpolation, via directional dependent filtering, up to advanced Motion-Compensated (MC) interpolation [5]. Generally speaking, MC methods generate better results than other ones despite of their massive requirements on computational resources and storage. 3D recursive searching (3DRS) algorithm [6-8], which has superiority at both true motion estimation and hardware complexity, is an efficient motion estimation method widely used in MC de-interlacing methods. Rather than taking all the possible candidate vectors, 3DRS algorithm calculates on a small number of spatial and temporal prediction vectors from a 3-D neighbourhood, which makes it efficient in yielding coherent vector fields that closely correspond to the true-motion of objects.

However, 3DRS algorithm uses forward block matching which suffers from latent error propogation and its fixed matching block size isn't suitable for predicting different kinds of motion. To overcome these deficiencies, we bring forward an improved 3DRS algorithm which introduces bi-directional block matching [9] to eliminate error propagation and adopts a multi-resolution searching method to adjust matching block size according to the video content. Multi-resolution searching, which makes use of block matching error and texture complexity to present the split criterion, is first proposed in [10]. In this paper we propose a new split criterion using motion similarity information to decide the search resolution.

The rest of this paper is organized as follows. Section 2 and Section 3 describe bi-directional block matching and multi-resolution searching respectively. Section 4 presents and discusses experimental results. Finally, we draw our conclusion in Section 5.

2. BI-DIRECTIONAL BLOCK MATCHING

The original 3DRS algorithm adopts forward block matching as Fig. 1 shows. Let f_{n-1} and f_n denote the previous and current fields respectively, where the solid lines represent the original lines and the dashed lines represent the missing lines. The missing lines in f_{n-1} have already been interpolated. For each block $B(x)$ of original pixels at block position x of f_n, the match error criterion is the Sum of Absolute Difference (SAD) which can be described as follow:

$$\varepsilon_1(d,x,\mathrm{n}) = \sum_{x' \in \mathrm{B}(x)} |f_{\mathrm{n}}(x') - f_{\mathrm{n}-1}(x'-d)|. \quad (1)$$

where d is the motion vector. This method doesn't place any restrictions on d, so the interpolation of the current field depends on both original and interpolated pixels in the previously de-interlaced field, which leads to the interpolation errors propagating into subsequent output frames.

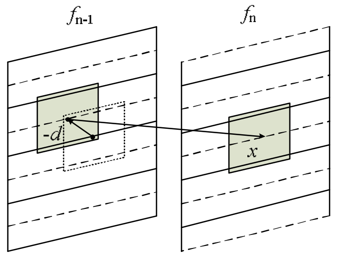


Fig. 1. Forward block matching for original 3DRS

We replace forward block matching by bi-directional block matching in our improved 3DRS algorithm. Bi-directional block matching is based on such an assumption that the forward and backward fields, which are of the same parity with different temporal position, have identical video contents placed upon different locations. The assumption holds for stationary areas and constant-speed motion areas, which compose the most part of a video sequence.

As shown in Fig. 2, let f_{n+1} denote the next field. For each block $B(x)$ of to-be-interpolated pixels at block position x of f_n, the match error criterion for bi-directional block matching is as follows:

$$\varepsilon_2(d, x, \mathrm{n}) = \sum_{x' \in \mathrm{B}(x)} | f_{\mathrm{n}+1}(x'+d) - f_{\mathrm{n}-1}(x'-d) | . \quad (2)$$

If motion vector d is represented by $(\mathrm{d}_x, \mathrm{d}_y)^{\mathrm{T}}$, then the following restriction must be satisfied:

$$\mathrm{d}_y \bmod 2 = 0. \quad (3)$$

which makes this algorithm get match error for d only by calculating on the original pixels from f_{n+1} and f_{n-1} without using the interpolated ones. Thus, error propagation is eliminated.

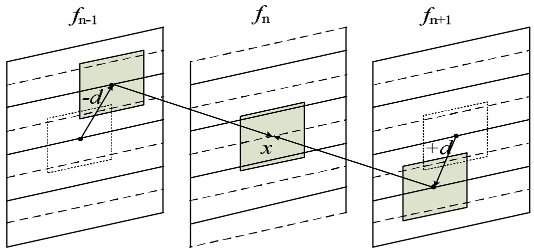


Fig. 2. Bi-directional block matching for improved 3DRS

3. MULTI-RESOLUTION SEARCHING

With regard to the block matching algorithm, a larger block size makes the motion estimation less sensitive to noise and reduces the aperture problem. However, large matching block isn't suitable to represent small-area motion details. To solve this problem we import a multi-resolution searching method which means to adjust matching block size according to the video content.

The basic algorithm can be briefly described as follows: First, at the lowest resolution we use the original 3DRS with a matching block sized 16x16 to estimate the motion and get a motion vector for each block; then by following the split criterion described below, we decide whether to go down to a higher resolution, that is, whether to split the current 16x16 block into four 8x8 blocks to continue the 3DRS motion estimation process. The same routine is applied to the generated 8x8 blocks until reaching the smallest block size 4x4. Fig. 3 illustrates a probable block-split case.

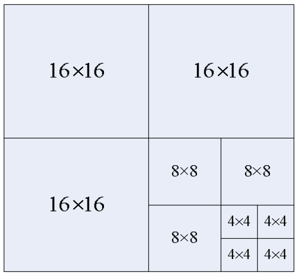


Fig. 3. A probable block-split case

3.1 Split Criterion

The key element for multi-resolution searching is about how to choose a proper searching resolution, in other words, making the decision of whether to split a block into four sub-blocks or not. In improved 3DRS algorithm we propose a simple split criterion which exploits motion similarity information around the current block to decide its searching resolution. This proposed split criterion is upon such a hypothesis that when you observe a moving object from a long distance, which means to observe at a low resolution, the object may contain many types of motion; but when you pay attention to a part of it, which means to observe at a high resolution, normally you will get a series of similar motions.

Fig. 4 shows the deeper-colored current block and its eight surrounding neighborhoods. The proposed split criterion can be formalized as follow:

$$R = f\,(MV_1, MV_2, \ldots, MV_9). \quad (4)$$

where MV_i (i=1,2,...,9) is the ith block's motion vector. Function f returns true or false depending on the similarity of these nine MVs. If at least 5 of them are equal to the current block's motion vector MV_5, then f returns true and makes our algorithm hold for the current search resolution. Otherwise f returns false and the current block is split into 4 smaller ones to continue 3DRS motion estimation at a higher resolution.

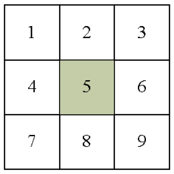


Fig. 4. Current block and its 8 neighborhoods

3.2 Candidate Set at Different Resolution

3DRS algorithm relies on motion vectors already estimated to construct a small set of candidate vectors (*candidate set*), from which the final MV dedicated to the current block is chosen. These already estimated motion vectors belong to the current block's spatio-temporal neighborhoods. The candidate set is composed of *spatial candidates*, *temporal candidates* and also some *update candidates* which are generated by taking a spatial candidate and adding a small vector to it. The small vetor is randomly chosen from the so-called *update set*.

To get better results, different candidate sets should be used at different resolutions (matching block sizes) [10], which can be achieved by choosing different spatio-temporal candidates or by changing the update set. Table 1 shows the candidate set for original 3DRS algorithm.

Table 1. Original 3DRS candidate set

Block size	Spatial candidate	Temporal candidate	Update set
8×8	S_1,S_2	T_1	U_1

S_1, S_2 are spatial candidates which are taken from neighboring blocks of the current field and T_1 is a temporal candidate which is taken from a neighboring block of the previous field. The candidate positions are depicted in Fig. 5, where C stands for the current block. Update set U_1 is defined as:

$$U_1 = \{\begin{pmatrix}0\\0\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}0\\-1\end{pmatrix},\begin{pmatrix}0\\2\end{pmatrix},\begin{pmatrix}0\\-2\end{pmatrix},\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}-1\\0\end{pmatrix},\begin{pmatrix}2\\0\end{pmatrix},\begin{pmatrix}-2\\0\end{pmatrix}\}$$

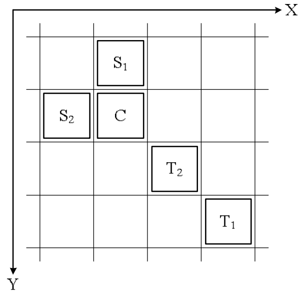


Fig.5. Positions for spatio-temporal candidates

Original 3DRS algorithm which generates experimental results shown in Section 4 employs five candidate vectors:

a. $MV(S_1)$
b. $MV(S_2)$
c. $MV(T_1)$
d. $MV(S_1)$ + $RANDOM(U_1)$
e. $MV(S_2)$ + $RANDOM(U_1)$

where MV(x) represents motion vector of block x and $RANDOM(U_1)$ represents a small vector randomly chosen from update set U_1.

At different resolutions improved 3DRS algorithm adopts different compositions of candidate set. The set of candidate sets employed by various resolutions is schematically listed in Table 2. At a lower resolution temporal candidate T_1 is replaced by T_2 to increase the reliability of the prediction and at a higher resolution update set U_1 is replaced by U_2 to increase horizontal search range.

$$U_2 = \{\begin{pmatrix}0\\0\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}0\\-1\end{pmatrix},\begin{pmatrix}0\\2\end{pmatrix},\begin{pmatrix}0\\-2\end{pmatrix},\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}-1\\0\end{pmatrix},\begin{pmatrix}2\\0\end{pmatrix},\begin{pmatrix}-2\\0\end{pmatrix},\begin{pmatrix}3\\0\end{pmatrix},\begin{pmatrix}-3\\0\end{pmatrix}\}$$

Table 2. Candidate sets at different resolution

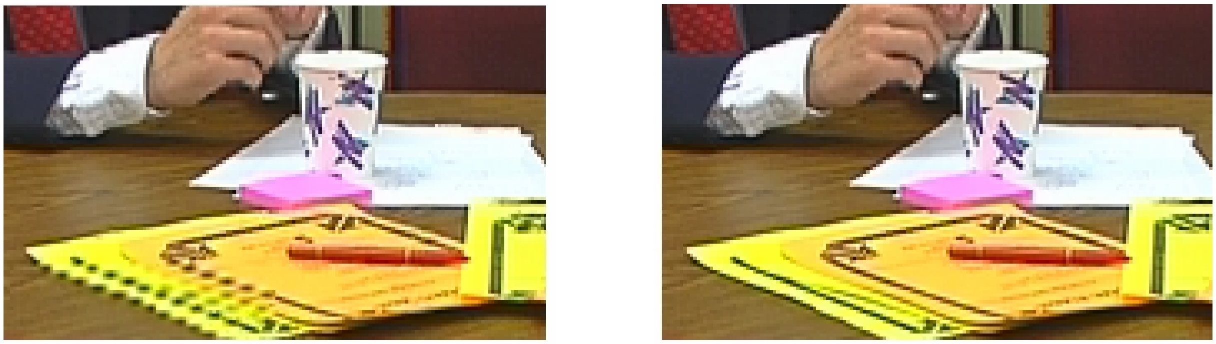
Block size	Spatial candidate	Temporal candidate	Update set
16×16	S_1,S_2	T_2	U_1
8×8	S_1,S_2	T_2	U_2
4×4	S_1,S_2	T_1	U_2

4. EXPERIMENTAL RESULTS

4.1 Subjective Visual Quality

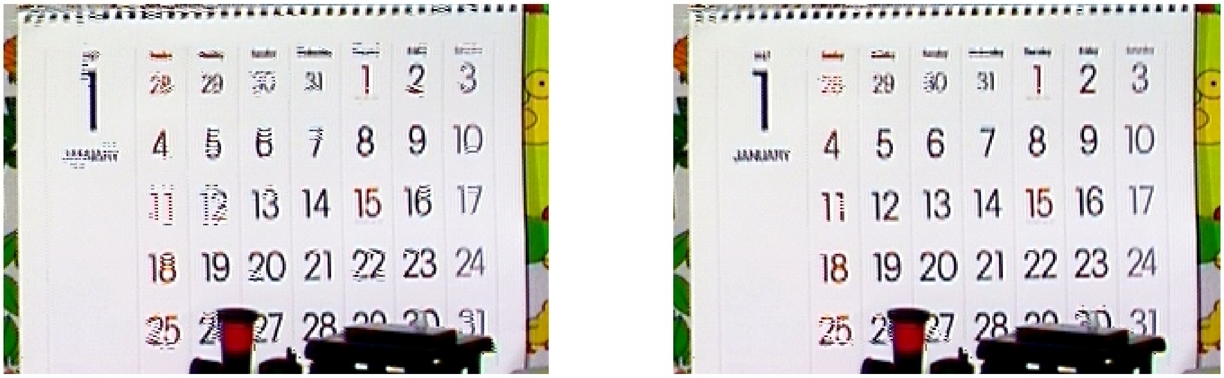
We extract interlaced videos from progressive sequences by dropping one field in each frame, then we de-interlace the generated videos using the original 3DRS algorithm and the improved 3DRS algorithm respectively. Both algorithms are followed by the AR error protection [11]. As shown in Fig. 6, improved 3DRS algorithm gets better subjective visual quality due to the good performance of bi-directional block matching on stationary areas and constant-speed motion areas.

Original 3DRS algorithm adopts forward block matching which suffers from error propogation, so it can't be utilized without error protection. To illustrate the advantage of multi-resolution searching used in our improved 3DRS algorithm, an experiment is conducted to test 3DRS with bi-directional block matching (Bi3DRS)



a) Original 3DRS with error protection b) Improved 3DRS with error protection

Fig. 6. De-interlaced pictures showing benefits from bi-directional block matching



a) Bi3DRS without error protection b) Improved 3DRS without error protection

Fig. 7. De-interlaced pictures showing benefits from multi-resolution searching

and the improved 3DRS algorithm with both bi-directional block matching and multi-resolution searching. These two algorithms are utilized without error protection in order to generate deinterlaced pictures from which we can see the accuracy of motion vetors directly. Fig. 7 shows that multi-resolution searching has a superority in finding the true motion.

4.2 Objective PSNR Performance

For an objective evaluation of our mechanism, the peak signal to noise ratio (PSNR) is used due to its mathematical tractability. Seven CIF progressive sequences and two HD progressive sequences are selected to generate interlaced seqeuces for objective evaluation of our proposed algorithm.

Experiments are conducted to test three de-interlacing algorithms: original 3DRS, Bi3DRS and improved 3DRS algorithm. Each of them is followed by the AR error protection to get the PSNR results shown in Table 3. It should be mentioned that the parameter C_1 of AR error protection determines how the temporal information from motion estimaiton and the spatial information from neighboring pixels are used to interpolate the missing pixels in the interlaced fields. Larger C_1 causes more spatial information used which can weaken the impact of error propagation but also reduce the benefits from accurate motion estimation. Since the improved 3DRS algorithm is apt to find ture motion vectors, we set C_1 to 0.2 instead of 0.3125 which is used in the original 3DRS algorithm [12] to take full advantage of the more accurate temporal information. Table 3 shows that the proposed approach generates better objective results than the original 3DRS algorithm on most of the tested sequences owing to the adoption of bi-directional block matching and multi-resolution searching.

5. CONCLUSION

3D recursive searching (3DRS) algorithm is an efficient motion estimation method for the video de-interlacing process. It has superiority at true motion estimation and hardware complexity but also has some weakness as well.

Table 3. PSNR results of the three different 3DRS algorithms

		C_1=0.3125			C_1=0.2
		3DRS	Bi3DRS	Improved 3DRS	Improved 3DRS
352 × 288	mobile	27.34	27.56	27.80	27.93
	paris	31.91	35.82	35.89	36.33
	bus	30.31	30.38	30.55	30.60
	foreman	32.50	35.77	35.97	35.95
	news	42.02	42.20	42.35	42.73
	template	32.30	32.71	32.81	33.07
	football	34.06	33.44	33.73	33.11
1280 ×720	night	35.15	36.10	36.29	36.48
	crew	39.21	40.10	40.20	39.46

For example, 3DRS algorithm uses forward block matching which suffers from latent error propagation and its fixed matching block size isn't suitable for predicting different kinds of motion. To overcome these deficiencies, we bring forward an improved 3DRS algorithm which introduces bi-directional block matching to eliminate error propagaion and adopts multi-resolution searching to adjust matching block size according to the video content. Experiments show that the proposed algorithm generates better subjective and objective results than the original 3DRS algorithm on most tested sequences.

ACKNOWLEDGEMENT

This work is partially supported by the National Natural Science Foundation of China under grant No. 60333020, the Natural Science Foundation of Beijing under grant No. 4041003. The authors would like to thank Honggang Qi, Longshe Huo, Peng Zhang, Yongbing Zhang for their valuable comments and support.

REFERENCES

1. Fukinuki, T.: Television: Past, Present, and Future. Proceedings of the IEEE. vol. 86, no. 5. (1998) 998–1004
2. Pigeon, S., Guillotel, P.: Advantages and Drawbacks of Interlaced and Progressive Scanning Formats. Europe: CEC RACE/HAMLET Deliverable (1995)
3. Vandendorpe, L., Cuvelier L.: Statistical Properties of Coded Interlaced and Progressive Image Sequences. IEEE Transactions on Image Processing. Vol. 8, No. 6. June (1999) 749-761
4. Advanced Television Systems Committee: http://www.atsc.org
5. de Haan, G., Bellers, E.B.: Deinterlacing – An overview. Proceedings of the IEEE. Vol. 86, No. 9, Sep (1998) 1839-1857
6. de Haan, G., Biezen, P., Huijgen, H., Ojo, O. A.: True motion estimation with 3-D recursive search block-matching. IEEE Trans. on Circ. and Syst. for Video Techn. 3 (1993) 368–388
7. de Haan, G., Biezen, P.: Sub-pixel motion estimation with 3–D recursive search block-matching. Signal Processing: Image Communication 6 (1994) 229-239
8. Braspenning, R.A., de Haan, G.: True-motion estimation using feature correspondence. SPIE, Proceedings of VCIP, Jan (2004) 396-407
9. Yoon, K., Kang-Sun, C., Jae-Young, P., Byung-Tae, C., Sung-Jea, K.: A Novel de-interlacing technique using bi-directional motion estimation. International Conference on Computational Science and its Applications (2003) 957-966
10. Braspenning, R.A., de Haan, G.: Effcient motion estimation with content-adaptive resolution. Proceedings of International Symposium on Consumer Electronics (ISCE), September (2002) E29-E34
11. de Haan, G., Bellers, E.B.: De-interlacing of video data. IEEE Trans. Consumer Electron. vol.43, Aug (1997) 819–825
12. de Haan, G., Biezen, P.W.A.C.: Time recursive de-interlacing for high-quality television receivers. Proc. of the Int. Workshop on HDTV and the Evolution of Television, Taipei, Taiwan, November (1995) 8B25–8B33