Improvement of conventional deinterlacing methods with extrema detection and interpolation

Jérôme Roussel, Pascal Bertolino, Marina Nicolas

IMPROVEMENT OF CONVENTIONAL DEINTERLACING METHODS WITH EXTREMA DETECTION AND INTERPOLATION

Jérôme ROUSSEL[1][2], Pascal BERTOLINO[2], Marina NICOLAS[1]

[1] ST Microelectronics S.A., 12 Rue Jules Horowitz B.P. 217, GRENOBLE - France
[2] Laboratory of Images and Signals, INPG, BP 46-38402 St Martin d'Hères - France

Abstract. This article presents a new algorithm for spatial deinterlacing that could easily be integrated in a more complete deinterlacing system, typically a spatio-temporal motion adaptive one. The spatial interpolation part often fails to reconstruct close to horizontal lines with a proper continuity, leading to highly visible artifacts. Our system preserves the structure continuity taking into account that the mis-interpolated points usually correspond to local value extrema. The processing is based on chained lists and connected graph construction. The new interpolation method is restricted to such structures, for the rest of the image, a proper traditional directional spatial interpolation gives satisfactory results already. Although the number of pixels affected by the extrema interpolation is relatively small, the overall image quality is subjectively well improved. Moreover, our solution allows to gain back one of the major advantages of motion compensation methods, without having to afford their complexity cost.

1 Introduction

The video signal is transmitted over the world in interlaced frames. For technical reasons, the TV signal frame frequency was selected according to the frequency of the electrical power supply and the requirements concerning large area flicker. Interlacing made it possible to cope both with the frame rate and the resolution requirements. However, new flat panels like plasma or L.C.D are progressive ones and thus require the display of the whole image at time t. Moreover, interlacing also causes flicker on objects containing high horizontal frequencies. There is thus a high interest in deinterlacing methods that allow a conversion from interlaced to progressive. They can be classified in two major families, the methods without motion compensation and the ones with motion compensation [1]. We will focus here on the methods without motion compensation. Those methods can in their turn be split in temporal, spatial and spatio-temporal adaptive methods. The adaptive method consists in going towards the temporal method in areas where there is no movement and towards the spatial method in moving areas [2] - [3]. The spatial interpolation part has its own limitations, mainly on the rendering of

close to horizontal lines and the technique proposed in this article addresses this particular point. First, the existing methods are reviewed showing that the limitation mentioned cannot be easily overcome. Then in a second part we describe our method, based on a new extrema detection and interpolation principle. The major improvements reached are then showed. Finally, we conclude on the cost of the method and give some hints about the remaining work.

2   Existing methods

From now on in the article, f_{in} represents the interlaced input image and $\tilde{f}$ the interpolated output image. (i, j) are the spatial coordinates where i represents the line position number and j the column position number. f_{in} is defined only for half of the lines, i.e. for i even or odd. Many methods and solutions have been proposed to perform spatial interpolation [4]. The first very basic one consists in using the average of the pixels above and below the missing one to interpolate the missing pixel, i.e.:

$$\tilde{f}(i,j) = \frac{(f_{in}(i-1,j) + f_{in}(i+1,j))}{2}. \tag{1}$$

This method does not make it possible to reconstruct high frequencies (contours) in a sharp way. Typically, it can introduce staircase contours, flicker or blur. To improve this technique, the next idea was to make the interpolation along the direction of contours, using the so-called E.L.A method (Edge-based Line Averaging) [5]. This latter method detects the best direction Dir for interpolation within a window centered on the missing pixel and then makes the interpolation according to the found direction:

$$\tilde{f}(i,j) = \frac{f_{in}(i-1,j-Dir) + f_{in}(i+1,j+Dir)}{2}. \tag{2}$$

Although it leads to a better interpolation of contours, the method still has several limitations. Indeed, the correlation is done at the local level and remains quite sensitive to noise. The direction of contours thus happens to be wrong which can lead to very annoying artifacts since it disrupts the structure of thin lines or contours. Many alternatives of this method [6] make it possible to correct wrong direction interpolation for a majority of the pixels of the image, for instance by computing the correlation between groups of pixels instead of pixel to pixel (*figure* 1). However, the results of these methods always remain dependent on the size of the window used, that determines the maximum angle allowed for the reconstruction of the contours. On the other side, the larger the window is, the higher the risk of bad interpolation [7]. Different existing features and metrics try to control an adequate window size and introduce weights to reduce the number of false directions [8] - [9]. But methods used to calculate this weights significantly increase the complexity of the solution. Still, all these alternatives only bring a final minor improvement and do not allow to reconstruct correctly close to horizontal lines and structures. This point is even more annoying in real

time where moving horizontal lines do not only look disrupted, but also instable and highly flickering.

Our goal was thus to propose a system that helps this problem significantly without having to pay for a motion compensation solution. The method proposed here overcomes the limitation of the current directional spatial interpolation method since it is not based anymore on a searching window. It also only affects pixels values in image areas with given characteristics that cannot be handled properly in existing methods. Typically it overrules the wrong interpolation of close to horizontal structures, keeping the results of the traditional spatial interpolation where the results are already satisfactory. It can be added to a classical spatial deinterlacing method that can itself be integrated in a motion adaptive one (*figure* 2).
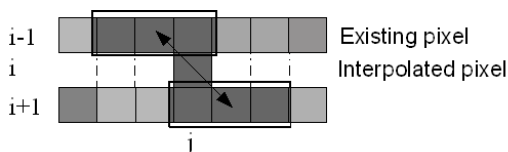

Fig. 1. Principle of modified E.L.A.

3 Extrema detection

The existing methods are not able to respect the continuity of close to horizontal thin structures. It is all the more awkward as the visual artifacts due to this problem are often very visible (disconnection, erroneous interpolation), as shown in the figure 3.c.

By comparing the modulus of the Fourier transform of a frame and the one of a whole image, one can observe in the case of the frame, on the one hand the spectrum folding phenomenon constituting the aliasing, and on the other hand the loss of the horizontal high frequencies. The difficulty consists in locating and reconstructing the continuity of these high frequencies structures which were partially destroyed and systematically disconnected by horizontal under-sampling (*figure* 3.b). Those correspond to local minima or maxima of the intensity function, in the vertical direction. The detection of these local extrema is carried out on the known lines of the image by comparing the value of each pixel $f_{in}(I, J)$ with the values of the closest lower and higher lines $f_{in}(i - 2, j)$ and $f_{in}(i + 2, j)$. Let $\mathcal{H}$ be the set of all maxima pixels and $\mathcal{L}$ the set of all minima pixels:

$$\mathcal{H} = \{f_{in}(i, j) \, / f_{in}(i, j) > \max(f_{in}(i - 2, j), f_{in}(i + 2, j)) + T\} \qquad (3)$$

$$\mathcal{L} = \{f_{in}(i, j) \, / f_{in}(i, j) < \min(f_{in}(i - 2, j), f_{in}(i + 2, j)) - T\} \qquad (4)$$

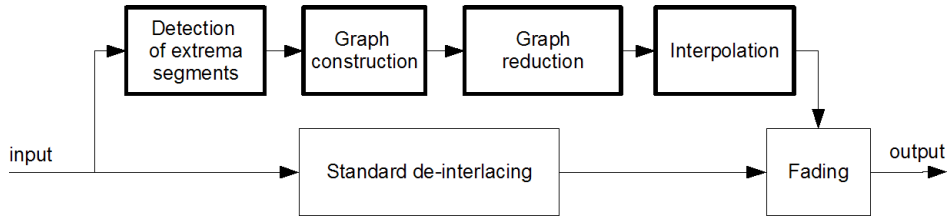T is a threshold value of minimum contrast ($T = 16$ in our experiments).

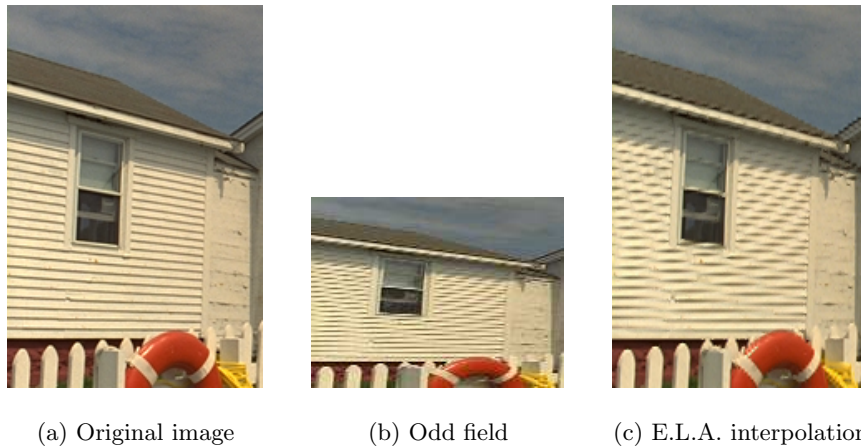**Fig. 2.** Block diagram of the process. The stages of the proposed method are in bold

(a) Original image   (b) Odd field   (c) E.L.A. interpolation

Fig. 3. The algorithms which use a search window cannot accurately rebuild the extrema-type structures

4 Segments and associated data structure

On the same line, the extrema of the same type can form connected components (called segments) according to the horizontal 2-connectivity (*figure* 4). As the continuation of the method is not founded on the traversing and the processing of pixels but on the traversing and the processing of segments, the traditional two-dimensional image structure is not suitable any more and is replaced with a high level structure: namely the segment. Each segment is an entity characterized by its coordinates (line, starting column), its length and type (minimum or

maximum). The chosen data structure is a compromise between the memory size needed and the complexity to traverse $\mathcal{H}$ and $\mathcal{L}$. The adopted solution is an array of lines (an entry for each frame line), each line being a chained list of the segments extracted in the corresponding frame line.

Fig. 4. Example of segments of the same type extracted (in black). The gray lines are known, the white lines are to be interpolated

5 Construction of connected graphs

This stage has two goals: first of all a practical goal to fill the structure presented above with all the segments, so that it can be easily traversed. Then a functional goal to interconnect the segments of $\mathcal{H}$ (resp. of $\mathcal{L}$) to constitute one or more connected graphs of maxima (resp. one or more connected graphs of minima). The traversing of the elements of $(\mathcal{H} \cup \mathcal{L})$ in the array of chained lists is done in the video scanning direction.

A segment S on a line i has at most 6 direct neighbors of the same type: 3 on the west side and 3 on the east side, 2 on each line $i-2$, i and $i+2$ (*figure* 5). The distance between two neighboring segments of the same type S_1 and S_2 is the Euclidean distance $d(S_1, S_2)$, calculated between the closest extremities of S_1 and S_2.

For a given side (west or east), S is connected to its closest neighbor. If two neighbors are closest at the same distance, S is connected to both (*figure* 6). Connections are bidirectional. An adaptive threshold is used in order to avoid not very reliable (too long) connections. L_1 and L_2 being the respective lengths of S_1 and S_2, the distance $d(S_1, S_2)$ must check the following condition so that S_1 and S_2 are connected:

$$d(S_1, S_2) < \min(L_1, L_2) + \delta \tag{5}$$

In our experiments δ is fixed at 2. Connections of a segment with its neighbors are stored in the segment itself (as pointers to segments). It should be noted that because of the intrinsic nature of local extrema, a segment cannot have more than two neighbors for a given side.

6 Graph reduction

The whole of the extracted graphs cannot be interpolated as it is. Certain connections must be removed (*figure* 7), which causes the division of a graph into
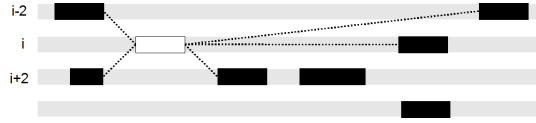
Fig. 5. The white segment has 5 direct neighbors

Fig. 6. Connected graph derived from the segments of figure 4

several subgraphs. The graphs comprising only one segment or which are on only one line are also removed. The suppression of connections must on the one hand privilege subgraphs having each one a prevalent direction and on the other hand remove the false positive ones (connections performed wrongly).

Let NW, W, SW, NE, E, SE be the directions associated to the 6 possible connections for a segment. When performing the in-depth traversal of the graph, let us call input direction the one by which the segment is reached. This direction is non-existent for the starting segment of the traversal. The output directions correspond to all connections of the segment except the input direction. The simplification complies with the following rules:

1. If there are 2 output connections on the same side, they are removed. This rule makes it possible not to connect potentially different structures wrongly.
2. If an output connection is on the same side that the input connection, it is removed. This rule makes it possible to preserve only structures that are stretched and not zigzag-like.

The subgraphs resulting from the traversal and the reduction rules are trees with only one branch. Long rectilinear or curve structures can be thus reconstituted.

Fig. 7. The remaining connections after the reduction of the graph of figure 6

7 Interpolation

The interpolation is the last stage (*figure* 8). It is carried out using a forward (from west to east) traversing of each branch.

Let S_1 and S_2 be two connected segments described by their lengths L_1 and L_2, their starting coordinates (Y_{S_1}, X_{start_1}) and (Y_{S_2}, X_{start_2}). The pixels to be interpolated with our method correspond to the segment S_I whose extremities X_{start_I} and X_{end_I} are linearly interpolated from the extremities of the segments S_1 and S_2 (*figure* 9).

$$X_{start_I} = X_{start_1} + \left| \frac{X_{start_2} - X_{start_1}}{2} \right| \tag{6}$$

$$X_{end_I} = X_{end_1} + \left| \frac{X_{end_2} - X_{end_1}}{2} \right| \tag{7}$$

$$\tilde{f}(i,j) = \frac{1}{2} f_{in} \left(Y_{S_1}, X_{start_1} + E\left(\frac{j \times L_1}{L_I} \right) \right)$$
$$+ \frac{1}{2} f_{in} \left(Y_{S_2}, X_{start_2} + E\left(\frac{j \times L_2}{L_I} \right) \right) \tag{8}$$
$$j \in [X_{start_I}, X_{end_I}]$$

The function E returns the nearest integer of its argument. L_I is the size of the segment to interpolate $X_{end_I} - X_{start_I} + 1$. Y_{S_1} and Y_{S_2} represent the ordinates $i - 1$ and $i + 1$ (or $i + 1$ and $i - 1$).

Fig. 8. The segments in dark gray represent the pixels interpolated thanks to the black neighboring segments

8 Results

The tests were carried out on a rather broad set of sequences, some coming from originally interlace material, others from originally progressive material that has been re-interlaced. The latter ones can be used as reference pictures to estimate the quality of the interpolation. The extrema pixels were interpolated following our method, as described in previous sections, while the E.L.A. method [8] was used to interpolate the other pixels in the image.
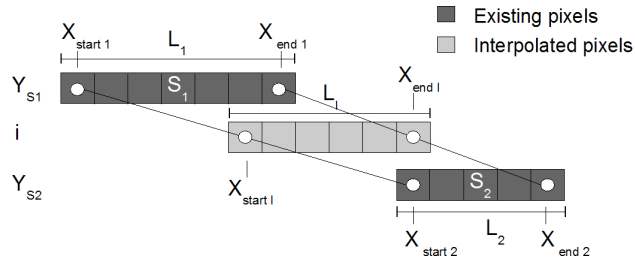
Fig. 9. Interpolation principle

If we consider the results obtained on fixed images with many horizontal details, such as the image of the lighthouse (*figure* 10) then we can see that the method reconstructs the continuity of the thin structures in a much better way than the E.L.A. method (*figure* 11).On the moving table tennis sequence, the improvement is also very visible (*figure* 12) already on a still frame. In real time, the benefit of our method is also noticeable since the stability of the moving horizontal lines is assured and the traditional flicker effect of conventional method is removed. The results obtained come closer to the results we can get with a motion compensation method. Generally, for the sequences tested, on the detected lines or curves the interpolation is correct, showing that the graph simplification stage, which is highly crucial in our algorithm, is robust enough. The close to horizontal structures which are not properly reconstructed by the traditional methods are almost identical to the structures present in the original progressive material.

The processing time hardly increases wrt the processing time of the original E.L.A. method since the method only performs very simple computations and since the amount of chained list data to be analysed is rather small. The added memory required is also very minor wrt to the memory requirements for the rest of a motion adaptive deinterlacing system. We analyzed the number of segments and the number of pixels interpolated by our method for a few sequences (table 1). On average the percentage of pixels interpolated by our method is about 2 % of the whole image. Because of this low percentage, the measurement of the PSNR is not very significant. However, the subjective improvement is noticeable since the eye is very sensitive to the continuity of the linear structures and to their flicker.

9   Conclusion

Our method is not based on the principle of existing methods. It thus does not suffer from the limitations of these methods. The method aims at correcting the most unpleasant artifacts for the human eye by detecting them directly. It is based on the continuity of object limits in order to reconstruct them. So the structures with strong contrast are more stable. Finally, our method can be

Sequences	Size	Number of pixels	Extrema pixels		Number of segments	Interpolated pixels	
			Number	%of the image		Number	%of the image
Lighthouse	768×512	393216	15000	3.8	7800	9500	2.4
Car 2	576×720	414720	10000	2.4	3500	5000	1.2
Calendar	576×720	414720	18500	4.5	13000	13000	3.1
BBC	576×720	414720	9500	2.3	3500	7500	1.8
Table Tennis	480×720	345600	9000	2.6	5600	4000	1.2
American banner	480×720	345600	4000	1.2	1900	3600	1.0
Means			11000	2.8	5883	7100	1.8

Table 1. Number of pixels and segments interpolated by our method for one still image and several video sequences

added to all the traditional methods to improve their weak point without a high added cost. We still have to find an automatic adjustment of the threshold used for the detection of the extrema according to local and global dynamics.

References

1. Sugiyama, K., Nakamura, H.: A method of de-interlacing with motion compensated interpolation. IEEE Trans. on Consumer Electronics **45** (1999) 611–616
2. Koivunen, T.: Motion detection of an interlaced video signal. IEEE Trans. on Consumer Electronics **40** (1994) 753–760
3. Lin, S.F., Chang, Y.L., Chen, L.G.: Motion adaptive interpolation with horizontal motion detection for deinterlacing. IEEE Trans. on Consumer Electronics **49** (2003) 1256–1265
4. de Haan, G., Bellers, E.: Deinterlacing - an overview. Proceedings of the IEEE **86** (1998) 1839–1857
5. T.Doyle: Interlaced to sequential conversion for edtv applications. in Proc. 2nd International Workshop Signal Processing of HDTV (1988) 412–430
6. Chen, T., Wu, H.R., Yu, Z.H.: Efficient deinterlacing algorithm using edge-based line average interpolation. Optical Engineering **39** (2000) 2101–2105
7. Yoo, H., Jeong, J.: Direction-oriented interpolation and its application to de-interlacing. IEEE Trans. on Consumer Electronics **48** (2002) 954–962
8. Park, M.K., Kang, M.G., Nam, K., Oh, S.G.: New edge depent deinterlacing algorithm based on horizontal edge pattern. IEEE Trans. on Consumer Electronics **49** (2003) 1508–1512
9. Byun, M., Park, M.K., , Kang, M.G.: Edi-based deinterlacing using edge patterns. IEEE International Conference on Image Processing **2** (2005) 1018–1021

(a) Original image

(b) De-interlacing with the proposed method

(c) Connection of minima

in the full size image
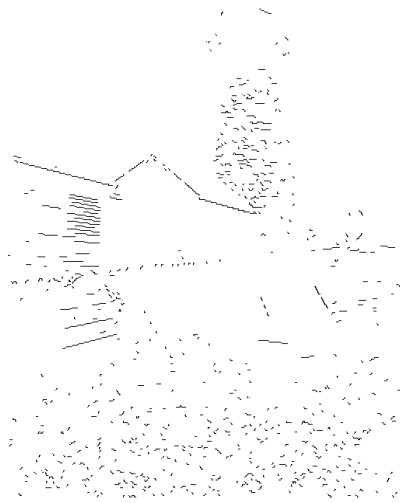
(d) Connection of maxima

in the full size image

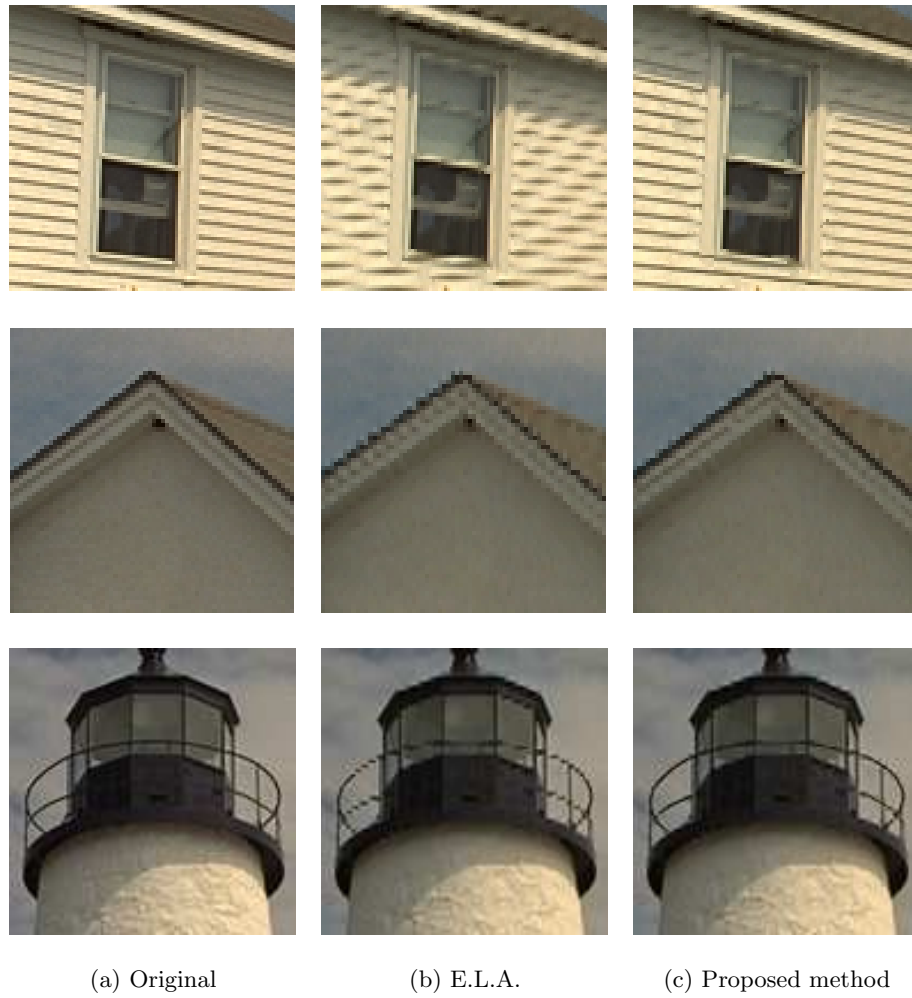**Fig. 10.** Reconstruction of thin structures using half of an original image

(a) Original   (b) E.L.A.   (c) Proposed method

Fig. 11. Comparison of the 2 methods for different parts of the lighthouse image

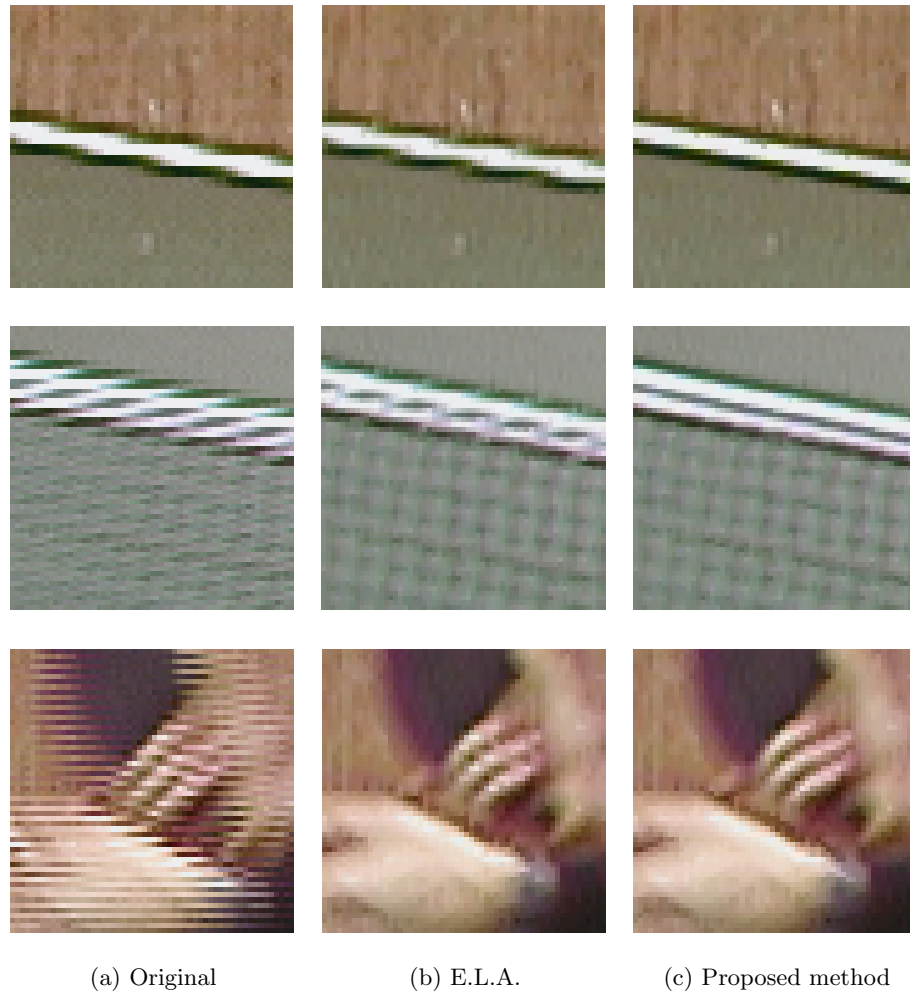(a) Original   (b) E.L.A.   (c) Proposed method

Fig. 12. Comparison of the 2 methods for different parts of the tennis sequence