De-interlacing of video data

G. de Haan  E.B. Bellers

1 Introduction

De-interlacing is a basic requirement for video scanning format conversions. Since perfection under all circumstances is impossible to achieve, many different algorithms to realize a good quality have been proposed. The products currently available on the consumer electronics market, either use linear Vertical-Temporal (VT) filtering [1, 2], MEDian filtering (MED) [6], or in the most advanced product, Motion Compensated MEDian filtering (mcMED) [8].

In the literature more advanced motion compensated algorithms have been described, including Time-Recursive (TR) methods [3], and (GST) methods based on a Generalized Sampling Theorem [4, 5].

This paper evaluates the existing commercially available methods\(^1\), the alternative methods as described in the literature cited above, as well as a new, Adaptive Recursive concept (AR), which could be shown to outperform all others. The AR de-interlacer is a further elaboration of ideas presented in [10].

2 The new AR algorithm

The basic assumption in our new algorithm is that the consistency along the motion trajectory for interpolated pixels should ideally equal that of the original pixels. To achieve this, the interpolated luminance value, \( F_i(\vec{x}, n) \) at position \( \vec{x} \) in picture number \( n \), obtained from any initial de-interlacing algorithm, is additionally filtered with an adaptive first order recursive motion compensated temporal filter, according to:

\[
F_{\text{out}}(\vec{x}, n) = \begin{cases} 
F(\vec{x}, n) & , y \bmod 2 = n \bmod 2 \\
kF_i(\vec{x}, n) + (1 - k)F_{\text{out}}(\vec{x} - \vec{D}, n - 1) & , \text{otherwise}
\end{cases}
\]

where \( \vec{D} \) is the motion vector, and \( F_{\text{out}}(\vec{x}, n) \) is the output of the de-interlacing algorithm. The original input pixels, \( F(\vec{x}, n) \), are just copied as indicated in equation (1), but may also be filtered [10].

Applying \( \vec{y}_u = (0, 1)^T \), with \( T \) for transpose, our basic assumption leads to the equation:

\[
\left| F_{\text{out}}(\vec{x}, n) - F_{\text{out}}(\vec{x} - \vec{D}, n - 1) \right| = \\
\left| F_{\text{out}}(\vec{x} - \vec{y}_u, n) - F_{\text{out}}(\vec{x} - \vec{y}_u - \vec{D}, n - 1) \right|
\]

Combining equation (1) and (2), we find:

\[
k(\vec{x}, n) = \frac{|F(\vec{x} - \vec{y}_u, n) - F_{\text{out}}(\vec{x} - \vec{y}_u - \vec{D}, n - 1)| + \delta}{|F_i(\vec{x}, n) - F_{\text{out}}(\vec{x} - \vec{D}, n - 1)| + \delta}
\]

where a small constant, \( \delta \), prevents division by zero, and \( k \) is clipped between 0 and 1. The consistency of the interpolated pixel should rather take the average of that of its vertical neighbours, instead of equaling that of its lower neighbour only. This and other sophistications shall be detailed in the full paper.

Although the AR algorithm can be applied to improve any existing de-interlacing technique, we used a straightforward intra-field line average as a basis to calculate \( F_i(\vec{x}, n) \):

\[
F_i(\vec{x}, n) = \frac{1}{2} \left( F(\vec{x} - \vec{y}_u, n) + F(\vec{x} + \vec{y}_u, n) \right)
\]

to illustrate the robustness of the improvement. A possible implementation is shown in Figure 2.

3 Evaluation and conclusion

Experiments have shown very good results of the AR algorithm as compared to the earlier mentioned alternatives. Figure 2 shows the mean-square-errors (MSE) and motion-trajectory-inconsistencies (MTI), calculated according to [5, 10]:

\[
MSE = \sum_{\vec{x}, n} \left( F(\vec{x}, n) - F_{\text{out}}(\vec{x} - \vec{D}, n - 1) \right)^2
\]
\[ MTI = \sum_{\bar{x}, n} \left( F_{\text{out}}(\bar{x}, n) - F_{\text{out}}(\bar{x} - D, n - 1) \right)^2 \] (6)

as obtained with both the commercially available devices, the methods described in the referenced literature, and the new algorithm. \(MSE\) and \(MTI\) are calculated here as an average over various sequences, identical to the ones used in the evaluation of [5].

In the category of simple non-motion compensated methods, the VT filter suffered from serious degradation on fast and particularly vertically moving sequences, whereas the the MED filter approach showed some quality loss in material with small details in the vertical direction.

The category of motion compensated methods showed that vector error protection means were indispensable. The lack of it, seriously decreases the score of the GST-based method, whereas the TR-method [3] profits from its median protection. The difference showed mainly in the complex motion sequences. In all categories, the new AR algorithm shows the best performance as can be concluded from Figure 2. The sub-pel accurate Recursive Search block-matcher of [7] is applied in all motion compensated methods.

Acknowledgements
The authors wish to thank Paul Biezen for his contribution to the optimization of the algorithm.

References


