A 2-dimensional generalised sampling theory and application to de-interlacing

Călina Ciuhu and Gerard de Haan
Philips Research Laboratories, Prof. Holstlaan 4, Eindhoven, The Netherlands

ABSTRACT

Yen’s generalisation of the sampling theorem has been proposed as the theoretical solution for de-interlacing by Delogne and Vandendorpe. The solution results in a vertical interpolation filter with coefficients that depend on the motion vector value, which uses samples that exist in the current field and additional samples from a neighbouring field shifted over (part of) a motion vector. We propose a further generalisation, where we design vector-adaptive inseparable 2D filters, which use samples from the current and the motion compensated previous field that are not available for all vectors on a vertical line. The resulting inseparable filters give a better interpolation quality at a given number of input pixels. We will show that the algorithm can be made robust against the sensitivity to inaccurate motion vectors.

Keywords: de-interlacing, generalised sampling theorem, motion compensation.

1. INTRODUCTION. THE GENERALISED SAMPLING THEOREM

According to the sampling theorem, a bandwidth–limited signal with a maximum frequency of \( f_s \), with \( f_s \) being the sampling frequency, can exactly be reconstructed after sampling at a frequency higher than \( f_s \) (Nyquist criterion).

In 1956, Yen showed a generalisation of the sampling theorem (GST), proving that a signal with a bandwidth of \( 0.5f_s \) can be reconstructed from \( N \) independent sets of samples, all obtained by sampling the signal at \( f_s/N \) (see also Figure 1).

1.1. De-interlacing and GST

Yen’s generalisation of the sampling theorem has been proposed as the solution for de-interlacing by Delogne and Vandendorpe. We shall refer to this method as the “GST de–interlacing” method.

As shown in Figure 2 for de-interlacing, the first of the two required independent sets of samples is created by shifting the samples from the previous field over the motion vector towards the current temporal instance. The second set of samples contains all pixels of the current field. The two sets are assumed to be independent, which is true unless a so-called “critical velocity” occurs, i.e. a velocity leading to an odd integer pixel displacement between two successive fields. In case the assumption is valid, Yen’s generalisation of the sampling theorem can be applied to generate a progressively scanned output signal. The output sample results, according to the theory as a weighted sum (GST-filter) of samples from both fields.

Using \( F(\vec{x}, n) \) for the luminance value of the pixel at position \( \vec{x} \) in image number \( n \), and \( F_i \) for the interpolated pixels at the missing line, we can define the output of the GST de-interlacing method as:

\[
F_i(\vec{x}, n) = \sum_{k} F(\vec{x} - (2k + 1)\vec{u}_y, n)h_1(k, \delta_y) + \sum_{m} F(\vec{x} - \vec{e}(\vec{x}, n) - 2m\vec{u}_y, n - 1)h_2(m, \delta_y),
\]

\( k, m = \{ \ldots - 1, 0, 1, 2, 3, \ldots \} \) (1)

---

Further author information: (Send correspondence to C.C.)
C.C.: E-mail: calina.ciuhu@philips.com, Telephone: +31 (0)40 27 45 356 Address: Philips Research Laboratories, Prof. Holstlaan 4 (WO), 5656 AA Eindhoven, The Netherlands
with $h_1$ and $h_2$ defining the GST filter, and the modified motion vector $\vec{c}(\vec{x}, n) = (e_x(\vec{x}, n), e_y(\vec{x}, n))^T$ defined as:

$$\vec{c}(\vec{x}, n) = \left( \begin{array}{c} d_x(\vec{x}, n) \\ 2\text{Round}\left( \frac{d_y(\vec{x}, n)}{2} \right) \end{array} \right)$$

with $d_x$, $d_y$ being the displacements in the $x$ and $y$ directions respectively, Round() rounding to the nearest integer value, and $\delta_y$ the vertical motion fraction defined by:

$$\delta_y(\vec{x}, n) = \left| d_y(\vec{x}, n) - 2\text{Round}\left( \frac{d_y(\vec{x}, n)}{2} \right) \right|$$

Note that in the above equations, motion is modelled by the linear GST filters $h_1$ and $h_2$.

The GST filter, composed of $h_1$ and $h_2$, depends on the vertical motion fraction $\delta_y(\vec{x}, n)$ and on the sub–pixel interpolator type. Assume that the current field contains the odd scanning lines only. Then, the corresponding even field, $F^e(\vec{x}, n)$, is defined by:

$$F^e(\vec{x}, n) = \sum_k F(\vec{x} - (2k + 1)\vec{u}_y, n)h_1(k, \delta_y) + \sum_m F(\vec{x} - \vec{c}(\vec{x}, n) - 2m\vec{u}_y, n - 1)h_2(m, \delta_y)$$

In line with the literature, we assume separate horizontal and vertical interpolators and focus on the interpolation in the $y$–direction. Equation 4 then simplifies to:

$$F^e(y, n) = \sum_k F(y - (2k + 1), n)h_1(k) + \sum_m F(y - e_y - 2m, n - 1)h_2(m)$$

Later in this paper, we will show that the assumption of separable interpolators is not always optimal, particularly if the video sequences have diagonal motion, rather than pure vertical motion.

If a progressive image $F^p$ would be available, $F^e$ could be determined as:

$$F^e(y, n) = \sum_k F^p(y - k, n - 1)h(k)$$

Since it is convenient to derive the filter coefficients in the $z$–domain, Equation 6 is transformed into:

$$F^e(z, n) = (F^p(z, n - 1)H(z))_e = F^o(z, n - 1)H^o(z) + F^e(z, n - 1)H^e(z)$$

where $(X)_e$ is the even field of $X$. Similarly:

$$F^o(z, n) = (F^p(z, n - 1)H(z))_o = F^o(z, n - 1)H^o(z) + F^e(z, n - 1)H^e(z)$$
which can be rewritten as:

\[ F^o(z, n - 1) = \frac{F^o(z, n) - F^e(z, n - 1)H^o(z)}{H^o(z)} \]  

(9)

Substituting Equation 9 into 7 results in:

\[ F^e(z, n) = H_1(z)F^o(z, n) + H_2(z)F^e(z, n - 1) \]  

(10)

with

\[ H_1(z) = \frac{H^o(z)}{H^e(z)} \]

\[ H_2(z) = H^e(z) - \left(\frac{H^o(z)}{H^e(z)}\right)^2 \]  

(11)

The GST filter coefficients are solely determined by the interpolator \( H(z) \). Vandendorpe et al.\(^3\) apply the sinc-waveform interpolator for deriving the GST filter coefficients:

\[ h_1(k) = (-1)^k \text{sinc}(\pi(k - \frac{1}{2})) \frac{\sin(\pi \delta_y)}{\cos(\pi \delta_y)} \]

\[ h_2(k) = (-1)^k \text{sinc}(\pi(k+\delta_y)) \frac{\sin(\pi \delta_y)}{\cos(\pi \delta_y)} \]  

(12)

Although the sinc-waveform interpolator represents the ideal filter, it is of theoretical value mainly. In practice, the number of coefficients is limited by the reduced validity of the vector for more distant neighbouring samples. Furthermore, motion vectors are usually of limited accuracy, which leaves the advantage of accurate interpolation useless. In the next sub-section, therefore, a more practical interpolator shall be introduced.

1.2. GST applying a first–order linear interpolation

The first–order linear interpolator is the preferred interpolator in many application, because of its favourable price–performance ratio. Therefore, we will use this interpolator in our initial experiments with the GST de-interlacer.

If we assume a first–order linear interpolator \( H(z) = (1 - \delta_y) + \delta_y z^{-1} \) with \( 0 \leq \delta_y \leq 1 \), then \( H_1(z) \) and \( H_2(z) \) are given by:

\[ H_1(z) = \frac{\delta_y}{1 - \delta_y} z^{-1} \]

\[ H_2(z) = (1 - \delta_y) - \left(\frac{\delta_y}{1 - \delta_y}\right)^2 z^{-2} \]  

(13)

As an example, we consider a motion of 0.5 pixels per field, i.e. \( \delta_y = 0.5 \). Then Equation (10) yields:

\[ F^e(z, n) = z^{-1}F^o(z, n) + \frac{1}{2}(1 - z^{-2})F^e(z, n - 1) \]  

(14)

Figure 2. The GST de–interlacing algorithm.
The inverse \( z \)-transform of \( F_e(z, n) \) gives the spatio–temporal expression for \( F_e(y, n) \):

\[
F_e(y, n) = F_o(y + 1, n) + \frac{1}{2} F_e(y, n - 1) - \frac{1}{2} F_e(y + 2, n - 1)
\] (15)

This is illustrated in Figure 3.

Consequently, the first–order linear interpolator, as defined above, results in a three taps GST interpolation filter. The above calculations assume linearity between two neighbouring pixels on the frame grid, as illustrated in Figure 4a. Since we started the derivation of the filter coefficients in Equation (6) by shifting samples from the previous frame to the current field, we inherently defined the interpolator function, \( H(z) \), on the previous frame grid. As such, the region of linearity for a first–order linear interpolator starts at the position of the MC sample as illustrated in Figure 4a. We may also center the region of linearity to the center of the nearest original and MC sample as illustrated in Figure 4b. The resulting GST filters now have four taps as shall be discussed in Section 2, i.e. the aperture of the GST interpolation filter is increased without modifying the interpolation order.

The additional tap in this four taps GST filter increases the contribution of spatially neighbouring sample values and decreases the relative importance of the less reliable motion estimated samples from the previous field. It has been experimentally validated,\(^7\) that this increases the robustness for incorrect motion vectors.

We note however, that this improvement of the robustness of the GST filter although visible, is far from sufficient for practical applications. A real solution for the robustness problem shall be introduced in Section 3.
2. A FURTHER GENERALISATION TO AN INSEPARABLE 2D GST

In Section 1.2, we described the application of the generalised sampling theorem (GST) to the de-interlacing of video signals. We made use of two independent sets of samples, i.e. a set of pixels from the current field, and a set of pixels from the previous/next field, shifted over the estimated motion vector \((d_x, d_y)\). The GST was only applied to design the vertical interpolation.

We shall prove, that the indirect usage of horizontally neighbouring pixels from the previous field, as it results from sub-pixel motion compensation, leads to a sub-optimal solution. We shall further indicate how a 2-dimensional inseparable GST interpolator can be designed based on a 2-dimensional linear interpolator.

To this end, we first analyse the design of the 1D-GST filter based on a linear interpolator in sub-section 2.1, followed by the introduction of the generalised GST filter to an inseparable 2D GST filter in sub-section 2.2.

2.1. Analysis of the GST interpolator

Linear interpolation is assumed to be valid within the region of linearity, which equals the pixel-distance on the frame grid. Depending on the location of this region of linearity, we can have 2, 3 or 4-taps interpolators. Figure 5 illustrates how the location of the region of linearity is related to the aperture of the GST interpolation filter. If the pixel to be interpolated, \(P\), is situated at the border between two regions of linearity, as in Figure 5a, then the corresponding triangular–wave pattern, that indicates the weight of each pixel as a function of its position, is located such that a linear relation can be established between the pixels \(P\), \(A\) and \(C\). This leads to the simplest, 2-taps, interpolator. Similarly, a choice of the regions of linearity as in Figure 5b, results in the most complex, 4-taps, interpolator. Experiments show, that the more pixels are being used, the better the interpolation quality.

For clarity, we will first analyse the 2-taps interpolator. The region of linearity starts at the position of the pixels \(A\) belonging to the current field, and ends up at the position of the pixels \(P\) to be interpolated. Thus, the borders of the triangle-wave pattern are located at the positions \((x + p, y + q)\) in the vertical direction, with \(p\) and \(q\) integers, as illustrated in Figure 5a. The pixels contributing to the interpolation are the ones within the region of linearity centered at the position of the pixel \(P\). These pixels are \(A \equiv F(x + \text{sign}(d_y)u_y, n)\), belonging to the current temporal field \(n\), and \(C \equiv F(x, y + \delta_y, n - 1)\), being motion compensated pixel from the previous field, \(n - 1\). Using \(P \equiv F_i(x, n)\) we find the 2-taps interpolation formula:

\[
P = \frac{-\delta_y A + C}{1 - \delta_y}.
\]

Figure 5. 1D GST; a) Position of the region of linearity and the triangle shaped interpolator leading to a 2-taps interpolator, b) Position of the region of linearity and the triangle shaped interpolator resulting in a 4-taps interpolator.
Shifting the region of linearity in the vertical direction, we can increase the aperture of the GST interpolation filter to include 4 taps. This situation is illustrated in Figure 5b. In addition to A and C, the pixels contributing to the interpolation are in this case $B = F(\vec{x} - \text{sign}(d_y)\vec{u}_y, n)$, belonging to the current temporal field $n$, and $D = F(x, y - 2\text{sign}(d_y) + \delta_y, n - 1)$, being motion compensated pixel from the previous field, $n - 1$.

Unless the horizontal component of the motion vector has an integer value, the values of the pixels C and D needed in the GST interpolation have to be interpolated. Assuming a horizontal interpolator of at most 4 taps, $C$ is calculated as:

$$C = \alpha F(x - \text{sign}(d_x) + \delta_x, y + \delta_y, n - 1) + \beta F(x + \delta_x, y + \delta_y, n - 1) + \gamma F(x + \text{sign}(d_x) + \delta_x, y + \delta_y, n - 1) + \delta F(x + 2\text{sign}(d_x) + \delta_x, y + \delta_y, n - 1),$$

(17)

where $\alpha$, $\beta$, $\gamma$, and $\delta$ are coefficients to be optimised. The same horizontal interpolator is applied to calculate D. Equation (17) performs an interpolation in the horizontal direction, we do not need the GST, as there is no sub-Nyquist sampling in the horizontal domain. The combination of the horizontal interpolation and the vertical GST-filter interpolation gives rise to a separable 10-taps filter, as indicated in Fig 6. In the following, we will refer to this interpolator as a 1D GST, 4-taps interpolator, the 4 referring to the vertical GST-filter only.

In Figure 6, we illustrate the result of the 4-taps interpolation described above. Encircled are the pixels used in this interpolation. We notice that, on a region of 2 pixels distance from the interpolated pixel $P$, there are pixels not being used, while some of the pixels $D$ being used are located more than a 2 pixels distance from the pixel $P$. We conclude that the distribution of the used pixels seems sub-optimal as using the $D$-pixels rather than a pixel $A$ or $B$ of coordinates $(x \pm 1, y \pm \text{sign}(d_y))$ implies a larger distance from the interpolated pixel.

2.2. Design of an inseparable 2D-GST interpolator

As it was formulated before, the GST applies to signals that are functions of time and one spatial direction. As such, we applied it to a de-interlacing algorithm in the vertical direction. As the video signals are functions of time and two spatial directions, however it is possible to define a de-interlacing algorithm that treats both spatial directions equally.

In this sub-section, we shall show a further generalisation of the theory and prove that it leads to a better price performance of the resulting de-interlacing algorithm.

First, we generalise the idea of region of linearity to a grid defining a 2D region of linearity. Mathematically, we have to find the reciprocal lattice of the frequency spectrum, which can be formulated with a simple equation

$$\hat{f} \hat{x} = 1,$$

(18)
where \( \vec{f} = (f_h, f_v) \) is the frequency in the \( \vec{z} = (x, y) \) direction.

In Figure 7, we illustrate the two reciprocal lattices. The region of linearity is now a square that has the semi-diagonal equal to one pixel size. Further, the triangular-wave pattern indicating the weights of the interpolator, takes the shape of a pyramidal interpolator, as shown in Figure 8. As in the 1D-GST case, shifting the region of linearity in the vertical or horizontal directions leads to different number of filter taps. In particular, if the pyramids are centered at positions \( (x + p, y) \), with \( p \) an arbitrary integer, the 1D case results, as shown in Figure 8a.

In the 2D situation, we can freely shift the position of the lattice in the horizontal direction. The simplest calculations result when the centers of the pyramids are at the positions \( x + p + \delta_x \) in the horizontal direction, with \( p \) arbitrary integers, see Figure 8b. This leads to a larger aperture of the 2D GST interpolation filter in the horizontal direction. If the vertical coordinate of the centers of pyramids are \( y + m \), we obtain a generalisation.

\[ (a) \ 5\text{-taps interpolator} \quad (b) \ 10\text{-taps interpolator} \]

**Figure 8.** 2D GST; a) Position of the region of linearity and the pyramidal interpolator leading to a 5-taps interpolator; b) Position of the region of linearity and the pyramidal interpolator resulting in a 10-taps interpolator.
of the 1D 2-taps interpolator, resulting in a 5-taps inseparable 2D interpolator. The expression used for this 2-taps interpolator is

$$P = -\delta_y |\delta_x|(1 - |\delta_x|)A^{(-1)} + (|\delta_x|^2 + (1 - |\delta_x|)^2)A + |\delta_x|(1 - |\delta_x|)A^{(+1)}$$

$$+ \frac{(1 - |\delta_x|)C^{(0)} + |\delta_x|C^{(+1)}}{1 - \delta_y}$$

where $A^{(\pm 1)} \equiv F(x \pm 1, y + \text{sign}(d_y), n)$, $C^{(0)} \equiv F(x + \delta_x, y + \delta_y, n - 1)$ and $C^{(+1)} \equiv (x + \text{sign}(d_x) + \delta_x, y + \delta_y, n - 1)$. We note that in Equation (19) the horizontal motion comes out naturally in the weighting factors of the taps used in the interpolation. Besides, this Equation reduces to the 1D case, described by Equation (16), in case $\delta_x = 0$. Based on the above, we expect that the 1D GST interpolator works well for sequences with vertical motion only, but is sub-optimal for motion in a diagonal direction, i.e. motion with both non-zero vertical and horizontal components.

The generalisation of the 4-taps interpolator is straightforward. The 4-taps interpolator additionally uses the pixels symmetrically situated with respect to $P$. This can be realised by shifting the centers of the pyramids from the vertical coordinate $y + m$ corresponding to the 2-taps interpolator, to the positions $y + m + \frac{\delta_x}{2}$. This shift enables the use of the pixels $B^{(-1)} \equiv F(x - 1, y - \text{sign}(d_y), n)$, $B \equiv F(x, y - \text{sign}(d_y), n)$, and $B^{(+1)} \equiv F(x + 1, y - \text{sign}(d_y), n)$ from the current field, as well as $D^{(0)} \equiv F(x + \delta_x, y - 2\text{sign}(d_y) + \delta_y, n - 1)$, $D^{(+1)} \equiv F(x + \text{sign}(d_x) + \delta_x, y - 2\text{sign}(d_y) + \delta_y, n - 1)$ from the previous field, as in the case of a 4-taps GST interpolator $^*$. As a consequence, the GST filter has an increased horizontal aperture, as compared to the 1D case, when this aperture is limited to the use of an average value $C$ given by Equation (17) of the motion compensated pixels in the horizontal direction, instead of just the value $F(x + \delta_x, y, n - 1)$. The resulting interpolating filter has the general form

$$P = -\frac{\delta_y(2 - \delta_y)}{4(1 - \delta_y)} \left\{ |\delta_x|(1 - |\delta_x|)A^{(-1)} + (|\delta_x|^2 + (1 - |\delta_x|)^2)A + |\delta_x|(1 - |\delta_x|)A^{(+1)} \right\}$$

$$+ \frac{\delta_y(2 - \delta_y)}{4(1 - \delta_y)} \left\{ |\delta_x|(1 - |\delta_x|)B^{(-1)} + (|\delta_x|^2 + (1 - |\delta_x|)^2)B + |\delta_x|(1 - |\delta_x|)B^{(+1)} \right\}$$

$$+ \frac{(2 - \delta_y)^2}{4(1 - \delta_y)} \left\{ (1 - |\delta_x|)C^{(0)} + |\delta_x|C^{(+1)} \right\}$$

$$+ \frac{\delta_y^2}{4(1 - \delta_y)} \left\{ (1 - |\delta_x|)D^{(0)} + |\delta_x|D^{(+1)} \right\}.$$  (20)

$^*$The name of 2D 4-taps GST given to this filter derives from the fact that it is a generalisation of the previously described 1D 4-taps GST filter. However, counting the pixels used, we conclude that this is actually a 10-taps interpolator.
Figure 10. The results $P_{n-1,n}$ and $P_{n+1,n}$ are two solutions theoretically identical.

The graphic representation of the interpolator (20) is given in Figure 9.

All the pixels used by the 2D GST interpolation filter are now situated within a distance of less than two units on the pixel grid from the interpolated pixel $P$. It comes out that the density of points on the lattice that represents the distribution of pixels is equal to one pixel per lattice cell. This leads to a maximal number of 12 pixels covered by a circle of a radius equal to the distance between 2 pixels, while the interpolator (20) is a 10-taps one. The remaining two pixels situated within the same distance from $P$ can also be accounted for, combining, for example, the coefficients of the pixels $C$ with a 3-taps filter $\{-\alpha, 1+2\alpha, -\alpha\}$, where $\alpha$ is a parameter to be optimised.

However, from Equation (20), we notice that the GST interpolation filter has a singularity for values of $\delta_y \to 1$, that is for critical motion vectors, which are vectors leading to a odd number of pixels between two successive fields. In this situation, the first and the second sets of samples required by the generalised sampling theorem are no longer independent and a fall-back algorithm needs to be applied for de-interlacing.

3. ROBUST DE-INTERLACING ALGORITHM BASED ON THE GST

For critical motion, the generalised sampling theorem provides no solution to our de-interlacing problem. Moreover, even for vectors that are not critical, but have values close to a critical vector, the GST algorithm becomes very sensitive for small inaccuracies of the estimated motion, as one can derive from Equation (20) taking a value of $\delta_y$ close to 1.

In this section, we shall introduce a solution to make the de-interlacing algorithm robust against motion vector inaccuracies, which is the main weakness of the GST-de-interlacing as shown before.\[8\]

Our solution is based on the observation that the GST cannot only be applied to de-interlace a video signal using samples from the current and the previous field, but that equally well the samples could be taken from the current and the next field $^\dagger$, as it is indicated in Figure 10. Therefore, Equation (20) is valid also if we replace the motion compensated samples from the previous field, $n-1$ with motion compensated samples from the next field, $n+1$. Using both options, two output samples result that are theoretically the same. If this is not the case, we must conclude that the motion vector is, at least locally, unreliable. In other words: the difference between the two samples provides a quality indicator for every interpolated pixel enabling us to discriminate between areas where we need a protection method and image portions where we have a (near) perfect result and should not touch the calculated samples.

$^\dagger$More general but likely less preferable, it is possible to de-interlace the current field with a first and a second field taken from different time instances to obtain two candidate progressive outputs.
More formally, we define a first output candidate following Equation (1):

\[
C_1(\vec{x}, n) = \sum_k F(\vec{x} - (2k + 1)\vec{u}_y, n)h_1(k, \delta_y) + \sum_m F(\vec{x} - \vec{c}(\vec{x}, n) - 2m\vec{u}_y, n - 1)h_2(m, \delta_y)
\]  

(21)

and a second candidate according to:

\[
C_2(\vec{x}, n) = \sum_k F(\vec{x} - (2k + 1)\vec{u}_y, n)h_1(k, -\delta_y) + \sum_m F(\vec{x} + \vec{c}(\vec{x}, n) - 2m\vec{u}_y, n + 1)h_2(-m, -\delta_y)
\]  

(22)

With these two candidates, we calculate a quality indicator \(Q_{GST}(\vec{x}, n)\):

\[
Q_{GST}(\vec{x}, n) = |C_1 - C_2|^{-1}
\]

(23)

which can be used, e.g., to fade between the average of the two outputs, in case they are considered reliable, and a fall-back option, e.g., line averaging, otherwise:

\[
F_i(\vec{x}, n) = \frac{Q_{LA}(C_1 + C_2) + Q_{GST}(A + B)}{2(Q_{LA} + Q_{GST})}
\]

(24)

with

\[
Q_{LA}(\vec{x}, n) = |A - B|^{-1}
\]

(25)

Alternatively to (23), one can define the quality factor \(Q_{GST}\) over a block of pixels. For a block of nine pixels, \(Q_{GST}\) is then defined as follows:

\[
Q_{GST}(\vec{x}, n) = \left| \frac{1}{9} \sum_{-1 \leq p, q \leq 1} (C_1(\vec{x} + p\vec{u}_x + 2q\vec{u}_y, n) - C_2(\vec{x} + p\vec{u}_x + 2q\vec{u}_y, n)) \right|^{-1}
\]

(26)

Experimentally, this block averaging was found to be an advantage in fine textured areas.

### 4. EVALUATION

We evaluated the performance of the 2D GST de-interlacing method by comparing it on a set of video sequences with some important alternatives, such as the Adaptive Recursive (AR) algorithm, which proved to be the best of the previously investigated methods.\(^8\) We calculated the error with respect to the progressive sequences using the mean square error (MSE) method. Results for a set of five video sequences, shown in Figure 11, are presented in Table 1. The first column indicates the MSE for the AR de-interlacing algorithm. The MSE for the separable 1D GST, 10-taps interpolation, is shown in the second column. The inseparable (2D) GST is also a 10-taps interpolation, for which results are given in the third column. The lower values of the MSE obtained

<table>
<thead>
<tr>
<th>Algorithm/Sequence</th>
<th>Adaptive Recursive</th>
<th>1D GST</th>
<th>2D GST</th>
<th>1D GST (Robust)</th>
<th>2D GST (Robust)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Siena</td>
<td>12.16</td>
<td>5.45</td>
<td>5.45</td>
<td>4.84</td>
<td>4.84</td>
</tr>
<tr>
<td>Kiel</td>
<td>84.73</td>
<td>107.04</td>
<td>76.43</td>
<td>97.11</td>
<td>75.95</td>
</tr>
<tr>
<td>Football</td>
<td>27.28</td>
<td>52.60</td>
<td>40.09</td>
<td>40.91</td>
<td>33.28</td>
</tr>
<tr>
<td>Renata</td>
<td>23.29</td>
<td>14.01</td>
<td>13.60</td>
<td>6.08</td>
<td>6.08</td>
</tr>
<tr>
<td>Bicycle</td>
<td>32.28</td>
<td>91.38</td>
<td>60.05</td>
<td>54.24</td>
<td>32.71</td>
</tr>
</tbody>
</table>
with the 2D-GST algorithm are due to a more compact distribution of the pixels used in interpolation, as it results from a comparison of Figures 6 and 9. Besides, less pixels are motion compensated, and therefore suffer from inaccuracies, with respect to the previous, 1D GST method.

Our study, based on the MSE results shown in Table 1, reveals the fact that a larger improvement is due to this method particularly for sequences with a diagonal motion, such as Kiel, Football, and Bicycle.

The robustness improvement, described in Section 3, leads to a significant improvement of the performance of the algorithm compared to the initial 1D- and 2D-GST methods in sequences with more complicated motion that cannot always be correctly estimated, such as Football or Bicycle. In the last two columns of the Table 1, we indicate the MSE for the robust 1D and 2D algorithms. To also show a subjective evaluation, the improved robustness can be judged on a screen photo of the sequence Bicycle in Fig. 12. Apparently, this problem is solved in the robust 2D-GST algorithm, which gives much better results for the most of the test sequences, compared to the AR algorithm.

5. CONCLUSIONS

De-interlacing is the primary resolution determinator of high-end video displays to which important emerging non-linear spatial up-scaling techniques, like DRC and PixelPlus, can only add finer details. With the advent of new display technologies like LCD and PDP the limitation in the image resolution is no longer in the display, but rather in the source or transmission system. At the same time these displays require a progressively scanned video input. Therefore, high quality de-interlacing is an important pre-requisite for superior image quality on these emerging displays.

GST-based de-interlacing is the theoretically optimal way to generate progressive images from interlaced video. In this paper, we introduced an extension of the initial generalised sampling theorem which is more appropriate for 2-dimensional signals such as video signals. With respect to the initial 1D-GST filter, the
Figure 12. Comparison between (a) the GST algorithm and (b) the robust GST-based algorithm. Some visible distortions in (a), are due to inaccurate motion vectors, and are strongly suppressed by using the proposed protection described in Section 3.

2D generalisation has a larger aperture in the horizontal direction, the filter coefficients depending explicitly on both vertical and horizontal motion vector fractions. This yields an improvement in the picture quality particularly for sequences with a diagonal motion.

However, the 1D- and the 2D-GST filters show a high sensitivity to inaccuracies of the motion vectors. It is our new robustness improvement method, based on the possibility to use both motion compensated samples from the previous as well from the next field, that opens the applicability of the GST de-interlacing method in products that have to cope with non-perfect vectors occasionally.

REFERENCES