De-interlacing—an overview

G. de Haan and E.B. Bellers
Philips Research Laboratories
5656 AA Eindhoven, The Netherlands

Abstract: **The question 'to interlace or not to interlace' divides the TV and the PC communities. A proper answer requires a common understanding of what is possible nowadays in de–interlacing video signals. This paper outlines the most relevant methods, and provides a relative comparison.**

1 Introduction

The human visual system is less sensitive to flickering details than to large–area flicker [1]. Television broadcast standards apply interlacing to profit from this fact. Interlace, however, complicates many image processing tasks [2]. Particularly, it complicates scanning–format conversions. These were necessary in the past mainly for international programme exchange, but with the advent of high-definition television, videophone, Internet, and video on PCs, many scanning formats have been added to the broadcast formats, and the need for conversion between formats is increasing [3].

This increasing need, not only in professional but also in consumer equipment, has restarted the discussion 'to interlace or not to interlace'. Particularly, this issue divides the TV and the PC communities. The latter seems biased towards the opinion that present–day technologies are powerful enough to produce progressively scanned video at high rate and do not need to trade–off vertical against time resolution through interlacing. On the other hand, the TV world seems more conservative, and biased towards the opinion that present–day technologies are powerful enough to adequately de–interlace video material, which reduces, or even eliminates, the need to introduce incompatable standards and sacrifice the investments of so many consumers.

It appears that the two camps have had disjunct expertises for a long time. In a world where the two fields are expected by many to be converging, it becomes inevitable to appreciate and understand each other's techniques to some extent. Currently, the knowledge in the PC community on scan rate conversion in general, and on de–interlacing in particular, seems to be lagging behind on the expertise available in the TV world. Given the availablility of advanced motion–compensated scan rate conversion techniques in consumer TV–sets since some years [4], it is remarkable that at the 1997 WinHec conference Gates [5] proposes the PC community to lower the picture rate of broadcast PCs to 60 Hz and to consider long–persistent phosphors to alleviate large area flicker, since good quality scan rate conversion would not be affordable in a consumer product.

The question, 'to interlace or not to interlace', touches various issues. Whether present–day technologies are powerful enough to produce progressively scanned video at high rate and good signal to noise ratio is not evident [6]. Moreover, a visual–communication system also involves display and transmission of video signals. Concerning the channel, the issue translates into: 'Is interlacing and de-interlacing still the optimal algorithm for reducing the signal bandwidth with a factor of two?' Before answering this question, it is necessary to know what can be achieved with de–interlacing techniques nowadays. Although there is evidence that an all–progressive chain gives at least as good an image quality as an all–interlaced chain with the same channel bandwidth [7], experiments with advanced de–interlacing in the receiver have not been reported. In fact, recent research [8] suggests that motion–compensated temporal interpolation, in a different context, can improve the efficiency of even highly efficient compression techniques. It seems appropriate, therefore, to evaluate the available options in de–interlacing, before jumping to conclusions.

Over the last two decades, many de–interlacing algorithms have been proposed. They range from simple spatial interpolation, via directional dependent filtering, up to advanced Motion–Compensated (MC) interpolation. Some methods are already available in products, while the more recent ones will appear in products when technology economically justifies their complexity.

Our paper outlines the most relevant algorithms, avail-

able either in TV and PC products or in recent literature, and compares their performance. This comparison provides figures of merit, such as Mean Square Errors (MSE). Also screen photographs are included showing the typical artifacts of the various de–interlacing methods. A 'foot–print' indicates the relative strengths and weaknesses of individual methods in a single graph.

We cannot hope that this overview shall silence the discussions on interlace. We do hope, however, that it serves to provide a common knowledge basis for the two divided camps. This can be a starting point for further experiments that will contribute to the final *technical* answer. The debate is unlikely to end even there, as introducing incompatible new TV standards in the past proved difficult, and balancing technical and non–technical issues may prove to be difficult.

The paper is organized as follows: Section 2 formulates the de-interlacing problem. Section 3 outlines the main non-MC techniques and Section 4 the most relevant MC methods. Section 5 presents the performance evaluation and we draw our conclusions in Section 6.

2 Problem statement

Figure 1 illustrates the de–interlacing task. The input video *fields*, containing samples of either the odd or the even vertical grid positions (*lines*) of an image, have to be converted to *frames*. These frames represent the same image as the corresponding input field but contain the samples of all lines. Formally, we shall define the output frame $F_{\mathrm{O}}(\vec{x}, n)$ as:

$$F_{\mathrm{O}}(\vec{x}, n) = \begin{cases} F(\vec{x}, n), y \bmod 2 = n \bmod 2 \\ F_{\mathrm{I}}(\vec{x}, n) \qquad\qquad , \text{otherwise} \end{cases} \tag{1}$$

with $\vec{x} = (x, y)^{\mathrm{T}}$ designating the spatial position, field number n and T for transpose, $F(\vec{x}, n)$ the input field defined for $y \bmod 2 = n \bmod 2$ only, and $F_{\mathrm{I}}(\vec{x}, n)$ the interpolated pixels.

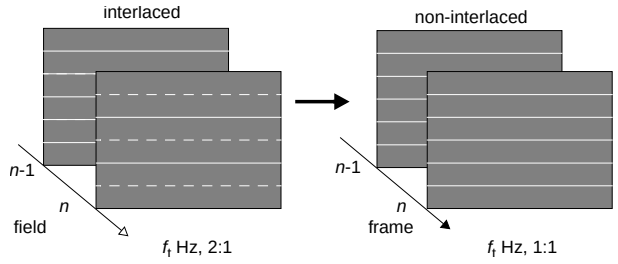


Figure 1: The de-interlacing task.

De-interlacing increases the vertical sampling density, and aims at removing the first repeat spectrum caused by the interlaced sampling of the video. It is not, however, a straightforward linear sampling rate up–conversion problem [9], as TV signals do not fulfil the demands of the sampling theorem: the prefiltering *prior* to sampling, required to suppress frequencies above half the sampling frequency, is missing. As in a TV system the pick–up device in the camera samples the scene (vertically and temporally), the prefilter should be in the optical path. This is hardly feasible, or at least absent in practical systems.

On top of this practical issue, there is a fundamental problem. The temporal frequencies at the retina of an observer have an unknown relation with the scene content [10]. High frequencies, due to object motion, are mapped to DC at the retina if the observer tracks the object. Consequently, suppression of such apparently high and less relevant frequencies results in significant blurring for this viewer. Temporal filtering of a video signal therefore degrades the picture quality.

Figure 2a shows the Vertical–Temporal (VT) video spectrum of a static scene. This spectrum includes baseband and spectral replicas due to the interlaced sampling. The sampling lattice results in a quinqunx pattern of the centers of the spectral replicas. The vertical detail of the scene determines the extent of the VT spectrum support, while vertical motion changes its orientation, as illustrated in Figure 2b [11]. Figure 3a illustrates the general spectrum for an interlaced signal with motion, and Figure 3b shows the ideally resulting spectrum from the de–interlacing process. Clearly, de–interlacing is a spatio–temporal problem, and the fundamental problem is highly relevant.

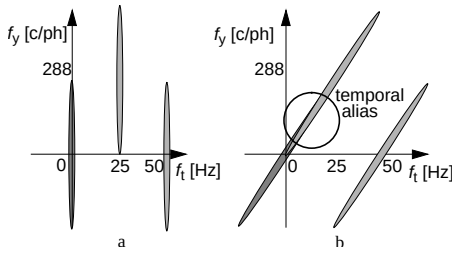


Figure 2: VT spectrum of a 50-Hz video signal, with vertical frequencies in cycles/picture height (c/ph). a: no motion, b: vertical motion

Due to these practical and fundamental problems, researchers have proposed many de–interlacing algorithms. Some neglected the problems with linear theory, and showed that acceptable results could nevertheless be achieved. Until the end of the seventies, this was the common approach for TV applications. From

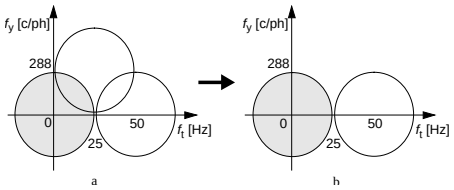


Figure 3: Spectrum of the interlaced input (a), and the target spectrum of the de–interlacer (b).

roughly the early eigthies onwards, others suggested that with non-linear means, linear methods can sometimes be outperformed. Next, motion compensation has been suggested to escape from problems in scenes with motion, but was considered to be too expensive for non–professional applications until the beginning of the nineties, when a breakthrough in motion estimation enabled a single–chip implementation for consumer TV [4]. Also in the nineties, video appeared in the PC, where up till now only the linear methods are applied. We shall discuss the relevant categories in the Sections 3 and 4.

3 Non-MC methods

We distinguish two categories of non-motion–compensated de-interlacing algorithms: linear and non-linear techniques. Both categories contain spatial (or intra-field), temporal (or inter-field), and spatio-temporal algorithms.

3.1 Linear techniques

The spatial and temporal filters are no longer popular in TV products. For multi–media PCs, however these techniques, under the name 'Bob' and 'Weave' [5] are currently proposed. Together with the spatio-temporal linear filters they are available in commercial products, and equally deserve our attention. All linear methods are defined by:

$$F_{\mathrm{O}}(\vec{x}, n) =$$

$$\begin{cases} F(\vec{x}, n) & y \bmod 2 = n \bmod 2 \\ \sum_k F(\vec{x} + k\vec{u}_y, n+m) h(k,m) & \text{otherwise} \\ & (k, m \in \{\ldots, -1, 0, 1, 2, \ldots\}) \end{cases} \quad (2)$$

with $h(k,m)$ the impulse response of the filter in the VT domain, and $\vec{u}_y = (0,1)^{\mathrm{T}}$. Similar to $\vec{u}_y$ we also define $\vec{u}_x = (1,0)^{\mathrm{T}}$. The actual choice of $h(k,m)$ determines whether it is a spatial, a temporal or a spatio-temporal filter.

3.1.1 Spatial filtering

Spatial de–interlacing techniques exploit the correlation between vertically neighbouring samples in a field when interpolating intermediate pixels. Their allpass temporal frequency response guarantees the absence of motion artifacts. Defects occur with high vertical frequencies only. The strength of spatial or intra–field methods is their low implementation cost.

The simplest form is *line repetition*, which results by selecting $h(k,0) = 1$ for $k = -1$, and $h(k,m) = 0$ otherwise. The frequency response of this interpolator is given by:

$$H_{\mathrm{y}}(f_{\mathrm{y}}) = |\cos(\pi f_{\mathrm{y}})| \quad (3)$$

with f_{y} the vertical frequency (normalized to the vertical sampling frequency), and $H_{\mathrm{y}}(f_{\mathrm{y}})$ the frequency response in the vertical direction. This frequency characteristic has no steep roll-off. As a consequence, the first spectral replica is not much suppressed, while the baseband is partly suppressed. This causes alias and blur in the output signal.

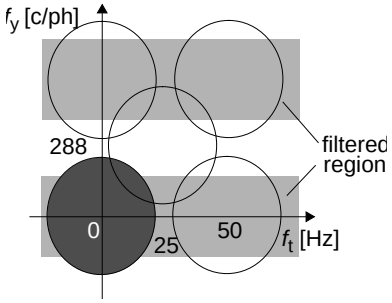


Figure 4: VT frequency response with a spatial filter

The alias suppression can be improved by increasing the order of the interpolator. Line averaging, or 'Bob' as it is called by the PC community, is one of the most popular methods, for which $h(k,0) = 0.5$ for $k = -1, 1$, and $h(k,m) = 0$ otherwise. Its response:

$$H_{\mathrm{y}}(f_{\mathrm{y}}) = \frac{1}{2} + \frac{1}{2}\cos(2\pi f_{\mathrm{y}}) \quad (4)$$

indicates a higher alias suppression. However, this suppresses the higher part of the baseband spectrum as well. Generally, purely spatial filters cannot discriminate between baseband and repeat spectrum regardless their length. They always balance between alias and resolution, as illustrated for a 50–Hz format in Figure 4. The gray shaded area indicates the passband, that either suppresses vertical detail, or passes the alias.

3.1.2 Temporal filtering

Temporal de-interlacing techniques exploit the correlation in the *time* domain. Pure temporal interpolation, implies a spatial allpass. Consequently, there is no degradation of *stationary* images.

The analogy in the temporal domain of the earlier line repetition method of the previous subsection is *field repetition* or *field insertion*. It results from selecting $h(0,-1) = 1$, and $h(k,m) = 0$ otherwise. The frequency characteristic of field repetition too, is the analogy of line repetition. It is defined by replacing f_y by f_t in (3).

Field insertion, also called 'Weave' in the PC world, provides an allpass characteristic in the vertical frequency domain. It is the best solution in case of still images, as all vertical frequencies are preserved. However, moving objects are not shown at the same position for odd and even lines of a single output frame. This causes serration of moving edges, which is a very annoying artifact illustrated in Figure 17.

Longer temporal FIR filters require multiple–field storage. They are are therefore economically unattractive, particularly as they also cannot discriminate between baseband and repeat spectra, as shown in Figure 5.

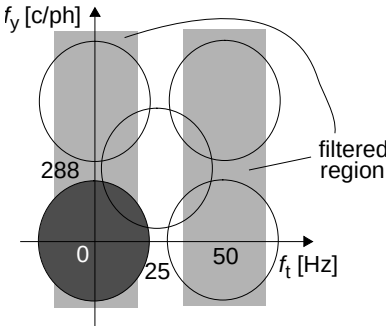


Figure 5: Frequency response of the temporal filter

3.1.3 VT filtering

A VT interpolation filter would theoretically solve the de–interlacing problem if the signal were bandwidth–limited prior to interlacing. The required pre–filter would be similar to the up–conversion filter. The required frequency characteristic is shown in Figure 6. Although the pre–filter is missing, and there are problems with motion–tracking viewers [10], Figure 6 illustrates that the VT filter is certainly the best linear approach, in that it prevents both alias and blur in stationary images. The vertical detail is gradually reduced with increasing temporal frequencies. Such a loss of resolution with motion is not unnatural.

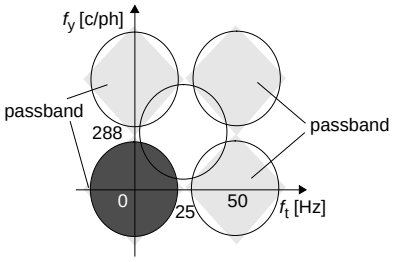


Figure 6: Video spectrum and a VT-filter

The filter is usually designed such that the contribution from the neighbouring fields is limited to the higher vertical frequencies. As a consequence, motion artifacts are absent for objects without vertical detail that move horizontally. In the evaluation we shall use such a filter with k and m selected as:

$$18h(k,m) = \begin{cases} 1,8,8,1 & ,(k=-3,-1,1,3) \wedge (m=0) \\ -5,10,-5 & ,(k=-2,0,2) \wedge (m=-1) \\ 0 & ,(\text{otherwise}) \end{cases} \tag{5}$$

3.2 Non-linear techniques

Linear temporal interpolators are perfect in the absence of motion. Linear spatial methods have no artifacts in case no vertical detail occurs. It seems logical, therefore, to adapt the interpolation strategy to motion and/or vertical detail. Many such systems have been proposed, mainly in the eigthies, and the detection of motion/detail can be explicit, or implicit. In this subsection, we describe some detail and motion detectors, some methods applying them, and finally some implicitly adaptive, non–linear de–interlacing algorithms. This last category seemed the best affordable de–interlacing technique for TV–receivers until, in the nineties, single–chip motion–compensated methods became feasible [4].

3.2.1 Motion–adaptive algorithms

To detect motion, the difference between two pictures is calculated. Unfortunately, due to noise, this signal does not become zero in all picture parts without motion. Some systems have additional problems. For example, colour subcarriers cause non–stationarities in coloured regions, interlace causes non–stationarities in vertically detailed parts, and timing jitter of the sampling clock is particularly harmful in horizontally detailed areas.

These problems imply that the motion detector output should be a multi-level signal, rather than a binary, indicating the *probability* of motion. Clearly, motion detection is not trivial. Therefore, assumptions are necessary to realize a practical motion detector that yields an adequate performance *in most cases*. Common (but not always valid) assumptions to improve the detector are:

1. Noise is small and signal is large;
2. The spectrum part around the colour carrier carries no motion information;
3. The low–frequency energy in the signal is larger than in noise and alias;
4. Objects are large compared to a pixel.

The general structure of a motion detector based on these assumptions is shown in Figure 7. A time–domain difference signal is first low–pass (and carrier reject) filtered to profit from assumptions 2) and 3) above. This filter also reduces 'nervousness' near edges in the event of timing jitter. After the rectification, another low–pass filter improves the consistency of the output signal, relying on assumption 4). Finally, the non–linear (but monotonous) transfer function in the last block translates the signal in a probability figure for the motion, $P_{\rm m}$, using 1). This last function may be adapted to the expected noise level. Low–pass filters are not necessarily linear. More than one detector can be used, working on more than just two fields in the neighbourhood of the current field, and a logical or linear combination of their outputs may lead to a more reliable indication of motion.

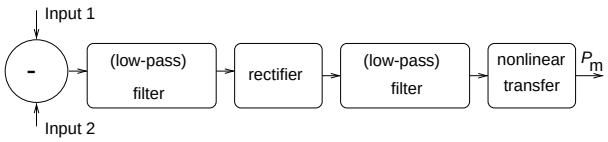


Figure 7: General structure of a motion detector

The Motion Detector (MD) is applied to switch or preferably fade between two processing modes, the one optimal for stationary and the other for moving image parts. Achiha *et al.* [12] and Prodan [13], mention that temporal and vertical filters may be combined to reject alias components and preserve true frequency components in the two–dimensional VT frequency domain by applying motion-adaptive fading. Bock [14] also mentioned the possibility to fade between an interpolator optimized for static image parts and one for moving image parts according to:

$$F_{\rm O}(\vec{x},n) = \begin{cases} F(\vec{x},n) & ,y \bmod 2 = n \bmod 2 \\ \alpha F_{\rm st}(\vec{x},n) + & \\ (1-\alpha)F_{\rm mot}(\vec{x},n) & ,\text{otherwise} \end{cases} \tag{6}$$

with $F_{\rm st}$ the result of interpolation for static image parts and $F_{\rm mot}$ the result for moving image parts. A motion detector determines the mix factor α.

Seth–Smith and Walker [15] suggested that a well defined VT filter can perform as well as the best motion–adaptive filter, at a lower price.

Filliman *et al.* [16] propose to fade between more than two interpolators. The high–frequency information for the interpolated line is extracted from the previous line. The low–frequency information is determined by a motion–adaptive interpolator.

$$F_{\rm O}(\vec{x},n) = \begin{cases} F(\vec{x},n) & ,y \bmod 2 = n \bmod 2 \\ F_{\rm HF}(\vec{x}+\vec{u}_y,n) + \alpha F_{\rm av}(\vec{x},n) & \\ +(1-\alpha)F_{\rm LF}(\vec{x},n-1) & ,\text{otherwise} \end{cases} \tag{7}$$

with $F_{\rm HF}$ and $F_{\rm LF}$ the highpass and low-pass filtered version of input signal F respectively, $F_{\rm av}$ defined by:

$$F_{\rm av} = \frac{F_{\rm LF}(\vec{x}-\vec{u}_y,n) + F_{\rm LF}(\vec{x}+\vec{u}_y,n)}{2} \tag{8}$$

with α controlled by the motion detector. The motion detector of Filliman *et al.* uses the frame difference. Field insertion results for the lower frequencies in the absence of motion, and line averaging in case of significant motion. Small frame differences yield an intermediate output.

Hentschel [17, 18] proposed to detect vertical edges, rather than motion, *within* a field. The edge detector output signal ED is defined by

$$ED(\vec{x},n) = g\left\{F\left(\vec{x}-\vec{u}_y,n\right) - F\left(\vec{x}+\vec{u}_y,n\right)\right\} \quad y \bmod 2 \neq n \bmod 2 \tag{9}$$

with $g()$ being a non–linear function that determines the presence of an edge. The output of $g()$ is either 0 or 1. Note that this detector does not discriminate between still and moving areas, but merely shows where temporal interpolation could be advantageous.

3.2.2 Edge–dependent interpolation

Doyle *et al.* [19] use a larger neighbourhood of samples to include information of the edge orientation. If intra–field interpolation is necessary because of motion, then

the interpolation should preferably preserve the baseband spectrum. After determining the least harmful filter orientation, the signal is interpolated in that direction. As shown in Figure 8, the interpolated sample

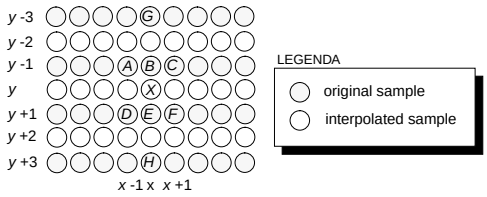


Figure 8: Apperture of edge dependent interpolators

X is determined by a luminance gradient indication calculated from its direct neighbourhood:

$$X = \begin{cases} X_{\mathrm{A}} & , \left((|A-F| < |C-D|) \wedge (|A-F| < |B-E|) \right) \\ X_{\mathrm{C}} & , \left((|C-D| < |A-F|) \wedge (|C-D| < |B-E|) \right) \\ X_{\mathrm{B}} & , \text{otherwise} \end{cases} \tag{10}$$

where X_{A}, $X_{\mathrm{B}}$and X_{C} are defined by:

$$X_{\mathrm{A}} = \tfrac{A+F}{2}, \quad X_{\mathrm{B}} = \tfrac{B+E}{2}, \quad X_{\mathrm{C}} = \tfrac{C+D}{2} \tag{11}$$

and the pixels A, B, C, D, E and F are the ones indicated in Figure 8, and defined by:

$$\begin{array}{ll} A = F(\vec{x} - \vec{u}_x - \vec{u}_y, n) & B = F(\vec{x} - \vec{u}_y, n) \\ C = F(\vec{x} + \vec{u}_x - \vec{u}_y, n) & D = F(\vec{x} - \vec{u}_x + \vec{u}_y, n) \\ E = F(\vec{x} + \vec{u}_y, n) & F = F(\vec{x} + \vec{u}_y + \vec{u}_x, n) \\ G = F(\vec{x} - 3\vec{u}_y, n) & H = F(\vec{x} + 3\vec{u}_y, n) \end{array} \tag{12}$$

In a preferred variant, X_{B} is replaced by a VT median filter, as described in the next section. Further modifications to this algorithm have been proposed [21].

It is uncertain whether a zero difference between pairs of neighbouring samples indicates the spatial direction in which the signal is stationary. For example, noise, or more fundamentally alias (edge detection on interlaced data), can negatively influence the decision. An edge detector can be applied to switch or fade between at least two processing modes, each of them optimal for interpolation of a certain orientation of the edge.

It is possible to increase the edge detection consistency [22], by checking also the edge orientation at the neighbouring pixel. In [22] *directional–edge detection operators* are defined. For example, the error measure for a vertical orientation is defined by:

$$angle90 = |B-E| + |C-F| \tag{13}$$

and for an edge under 116 degrees:

$$angle116 = |A-E| + |B-F| \tag{14}$$

Consistency of edge information is further increased by looking for a dominating main direction in a near neighbourhood. However, the problem of alias remains.

3.2.3 Implicitly adapting methods

Next to the adaptive linear filters for de-interlacing, non-linear filters have been described that implicitly adapt to motion or edges. Median filtering [23] is by far the most popular example. The simplest version is the three-tap VT median filter, illustrated in Figure 9. The interpolated samples are found as the median

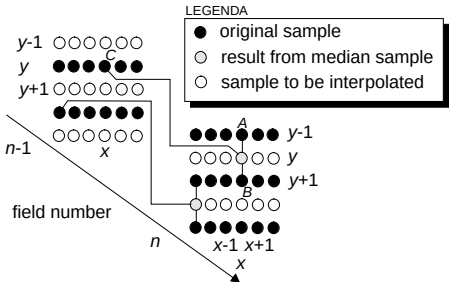


Figure 9: VT median filtering

luminance value of the vertical neighbours (A and B), and the temporal neighbour in the previous field (C):

$$F_{\mathrm{O}}(\vec{x}, n) = \begin{cases} F(\vec{x}, n) & , y \bmod 2 = n \bmod 2 \\ \mathrm{med}\left(F(\vec{x} - \vec{u}_y, n), F(\vec{x} + \vec{u}_y, n), F(\vec{x}, n-1)\right) & , \text{otherwise} \end{cases} \tag{15}$$

where $\mathrm{med}(A, B, C)$ is defined by [1]:

$$\mathrm{med}(A, B, C) = \begin{cases} A, & (B < A < C) \vee (C < A < B) \\ B, & (A \le B \le C) \vee (C \le B \le A) \\ C, & \text{otherwise} \end{cases} \tag{16}$$

The underlying assumption is that, in case of stationarity, $F(\vec{x}, n-1)$ is likely to have a value between that of its vertical neighbours in the current field. This results in temporal interpolation. However, in case of motion, intra-field interpolation often results, since then the correlation between the samples in the current field is likely to be the highest. Median filtering automatically realizes this 'intra/inter' switch on pixel basis.

If signals are corrupted by noise, the median filter allows noise breakthrough near edges. This is a flaw which can be reduced by applying order statistical smoothing prior to median filtering as proposed by Hwang *et al.* [24].

The major drawback of median filtering is that it distorts vertical details and introduces alias. However, its superior properties at vertical edges and its low hardware cost have made it very successful [25].

[1] The definition of the median is given here for three input values only. It is assumed to be known that this definition can be generalized to any number of input values

3.2.4 Hybrid methods

In the literature, many combinations of the earlier described methods have been proposed. Lehtonen and Renfors [26] combine a VT filter with a 5-point median. The output of the VT filter is one of the inputs of a 5-point median. The remaining four inputs are nearest neighbours on the VT sampling grid.

Salo *et al.* [27] extend the aperture of the median filter in the horizontal domain to enable implicit edge adaptation. The 3-point median was extended to a 7-point median, and the output of the median filter is defined by:

$$F_{\mathrm{O}}(\vec{x},n) = \mathrm{med}(A,B,C,D,E,F,F(\vec{x},n-1)) \quad (17)$$

where A, B, C, D, E and F are the pixels as indicated in Figure 8, and defined in (12). Haavisto *et al.* [28], extend this concept with a motion detector. They propose a 7–point spatio-temporal window as a basis for *weighted median filtering*. The motion detector controls the importance or 'weight' of these individual pixels at the input of the median filter. The output is defined by:

$$F_{\mathrm{O}}(\vec{x},n) = \begin{cases} F(\vec{x},n) & , y \bmod 2 = n \bmod 2 \\ \mathrm{med}\begin{pmatrix} A,B,C,D,E,F, \\ \alpha F(\vec{x},n-1), \\ \beta(B+E) \end{pmatrix} & , \text{otherwise} \end{cases} \quad (18)$$

where α and β are the integer weights. αA indicates the number of A's that occur in (18). (For example $3A$ means A, A, A). A large value of α increases the probability of field insertion, whereas a large β increases the probability of line averaging at the output.

Simonetti [29] describes yet another combination of implicit/explicit edge and motion adaptivity. His deinterlacing algorithm uses a hierarchical three-level motion detector which provides indications of static, slow and fast motion. Based on this analysis, one of the three different interpolators is selected. In case of static images, a temporal FIR filter is selected, in case of slow motion the so called Weighted Hybrid Median Filter (WHMF) is used, and in case of fast motion, a spatial FIR filter is used as the interpolator. Applying the definitions (12) and perusal of Figure 8 yields:

$$F_{\mathrm{O}}(\vec{x},n) = \begin{cases} F(\vec{x},n) & , y \bmod 2 = n \bmod 2 \\ \frac{1}{2}(F(\vec{x},n-1) + (\vec{x},n+1)) & , \text{static} \\ \mathrm{med}\begin{pmatrix} \alpha_0 \frac{A+F}{2}, \\ \alpha_1 \frac{B+E}{2}, \\ \alpha_2 \frac{C+D}{2}, \\ \alpha_3 \frac{G+H}{2} \end{pmatrix} & , \text{slow motion} \\ c_0 B + c_1 E + c_2 G + c_3 H, & \text{fast motion} \end{cases} \quad (19)$$

The coefficients α_i are calculated according to Webers law [30] ('the eye is more sensitive to small luminance differences in dark areas than in bright areas'). Using:

$$\beta_0 = \frac{A+F}{|A-F|}, \quad \beta_1 = \frac{B+E}{|B-E|}, \quad \beta_2 = \frac{C+D}{|C-D|}, \quad \beta_3 = \frac{G+H}{|G-H|} \quad (20)$$

and assuming β_i is the minimum, then $\alpha_i = 2$ and $\alpha_j = 1 \; \forall \; j \neq i$. Simonetti proposes a motion detector with a temporal aperture of 3 fields.

Kim *et al.* [31] detect motion by comparing an environment within the previous field with the same environment in the next field. Motion is detected if the (weighted) sum of absolute difference between corresponding pixels in the two environments exceeds a motion threshold value. Furthermore, vertical edges are detected by comparing the absolute difference of vertically neighbouring samples with a threshold value.

Depending on the edge and motion detectors, their output at interpolated lines switches between temporal averaging

$$F_{\mathrm{I}}(\vec{x},n) = \frac{1}{2}\left(F(\vec{x},n-1) + F(\vec{x},n+1)\right) \quad (21)$$

and edge–dependent interpolation according to

$$F_{\mathrm{I}}(\vec{x},n) = \frac{1}{2}\left(F(\vec{x}+l\vec{u}_x - \vec{u}_y, n) + F(\vec{x}+(m-l)\vec{u}_x + \vec{u}_y, n)\right) \quad (22)$$

where the interpolation directions, i.e. l and m, are determined using 'wide vector' correlations:

$$D(l,m) = \sum_{i=-L}^{L} |U(i) - V(i)|\,\gamma_i \quad (23)$$

where γ_i represents the weight, while

$$U = \begin{bmatrix} F(\vec{x}-(L-l)\vec{u}_x - \vec{u}_y, n) \\ \vdots \\ F(\vec{x}+\vec{u}_x - \vec{u}_y, n) \\ \vdots \\ F(\vec{x}+(L+l)\vec{u}_x - \vec{u}_y, n) \end{bmatrix} \quad (24)$$

and

$$V = \begin{bmatrix} F(\vec{x} - (L + l - m)\vec{u}_x + \vec{u}_y, n) \\ \vdots \\ F(\vec{x} + (m - l)\vec{u}_x + \vec{u}_y, n) \\ \vdots \\ F(\vec{x} + (L - l + m)\vec{u}_x + \vec{u}_y, n) \end{bmatrix} \quad (25)$$

The smallest $D(l, m)$ determines l and m.

4 MC methods

The most advanced de-interlacing algorithms use motion compensation. It is only since the mid–nineties that motion–estimators became feasible at consumer price level. They are currently available in studio scan rate convertors, in the more advanced TV receivers [4], and in single–chip consumer MPEG2 encoders [32].

We will assume the availability of motion vectors, but not discuss motion estimation. Since motion vectors can be incorrect, robustness of the de-interlacer against vector errors is important. In Section 5, robustness is discussed. We shall describe motion using $\vec{d}(\vec{x}, n) = (d_x(\vec{x}, n), d_y(\vec{x}, n))^{\mathrm{T}}$ with $d_x(\vec{x}, n)$ and $d_y(\vec{x}, n)$ the displacement or motion in the horizontal and vertical direction respectively.

Similar to many previous algorithms, MC methods try to interpolate in the direction with the highest correlation. With motion vectors available, this is an interpolation along the motion trajectory. Motion compensation allows us to virtually convert a moving sequence into a stationary one. Methods that perform better for stationary than for moving image parts will profit from motion compensation. Replacing the pixels $F(\vec{x}, n+m)$ with $F(\vec{x}+m\vec{d}(\vec{x}, n), n+m)$ converts a non-MC method in a MC version. Indeed, MC field insertion, MC field averaging, MC VT filtering, MC median filtering, and combinations with edge adaptivity have been proposed.

In this section we shall focus on methods that cannot readily be deduced from the non-MC algorithms. The common feature of these methods, is that they provide a solution to the fundamental problem of motion compensating subsampled data. This problem arises if the motion vector used to modify coordinates of pixels in a neighbouring field, does not point to a pixel on the interlaced sampling grid. In the horizontal domain this causes no problem, as sampling rate conversion theory is applicable. In the vertical domain, however, the demands for applying the sampling theorem are not satisfied, prohibiting correct interpolation.

4.1 Temporal backward projection

A first approximation to cope with this fundamental problem, is to nevertheless perform a spatial interpolation whenever the motion vector points at a non-existing sample, or even round to the nearest pixel.

Woods *et al.* [33] depart from this approximation. However, before actually performing an intra–field interpolation the motion vector is extended into the pre–previous field to check whether this *extended* vector arrives in the *vicinity* of an existing pixel. Figure 10 illustrates the procedure. Only if this is not the case spatial interpolation in the previous field is proposed:

$$F_{\mathrm{O}}(\vec{x}, n) = \begin{cases} F(\vec{x}, n) & , \; y \bmod 2 = n \bmod 2 \\ F(\vec{x} - \vec{d}(\vec{x}, n) - \vec{\varepsilon}, n-1) \; , & \\ (y - d_y - \varepsilon_y) \bmod 2 = (n-1) \bmod 2 & \\ F(\vec{x} - 2\vec{d}(\vec{x}, n) - 2\vec{\varepsilon}, n-2) \; , & \\ \quad (y - 2d_y - 2\varepsilon_y) \bmod 2 = n \bmod 2 & \\ F(\vec{x} - \vec{d}(\vec{x}, n), n-1) & , \text{otherwise} \end{cases} \quad (26)$$

where $\vec{\varepsilon} = (0, \varepsilon_y)^{\mathrm{T}}$, and ε_y is the small error resulting from rounding to the nearest grid position. This ε_y has to be smaller than a threshold. If no motion compensated pixel appears in the vicinity of the required position, it would be possible to find one even further backwards in time. This, however, is not recommended as the motion vector loses validity by extending it too much.

The algorithm implicitly assumes uniform motion over a two–field period, which is a drawback. Furthermore, the robustness to incorrect motion vectors is poor, since no protection is proposed. In Section 5, the consequences shall become evident.

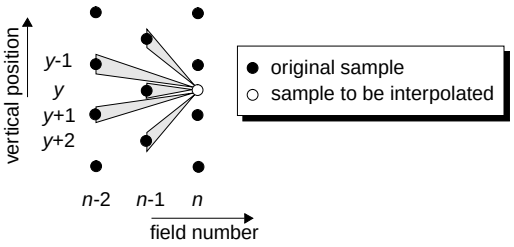


Figure 10: Temporal backward projection

4.2 Time–recursive de–interlacing

The MC Time-Recursive (TR) de-interlacer of Wang *et al.* [34] uses the previously de-interlaced field (frame) instead of the previous field in a 'field'-insertion algorithm. Once a perfectly de-interlaced image is available, and the motion vectors are accurate, sampling

rate conversion theory can be used to interpolate the samples required to de-interlace the current field:

$$F_{\rm O}(\vec{x},n) = \begin{cases} F(\vec{x},n) & , \; y \bmod 2 = n \bmod 2 \\ F_{\rm O}(\vec{x}-\vec{d}(\vec{x},n),n-1) & , \text{otherwise} \end{cases} \quad (27)$$

As can be seen in Figure 11, the interpolated samples generally depend on previous *original* samples as well as previously *interpolated* samples. Thus, errors originating from an output frame, can propagate into subsequent output frames. This is inherent to the recursive approach, and is the most important drawback of this method. To prevent serious errors from propagat-

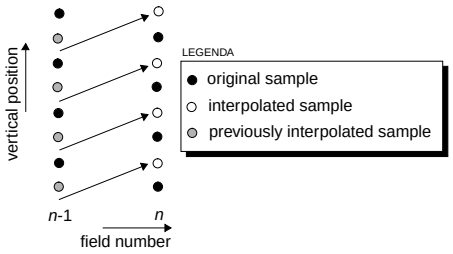


Figure 11: Time–recursive de–interlacing

ing, solutions have been described in [34]. Particularly, the median filter is recommended for protection. As a consequence, the TR de-interlacing becomes similar to the motion compensated median filter approach, albeit that the previous image consists of a previously de-interlaced field instead of the previous field. The output is defined by:

$$F_{\rm O}(\vec{x},n) = \begin{cases} F(\vec{x},n) & , \; y \bmod 2 = n \bmod 2 \\ \text{med}\left(\begin{array}{l} F_{\rm O}(\vec{x}-\vec{d}(\vec{x},n),n-1), \\ F\left(\vec{x}-\vec{u}_y,n\right), \\ F\left(\vec{x}+\vec{u}_y,n\right), \end{array} \right) & , \text{otherwise} \end{cases} \quad (28)$$

This is a very effective method, although the median filter can introduce aliasing in the de-interlaced image as illustrated in Figure 21.

4.3 Adaptive–recursive de–interlacing

Aliasing at the output of the de-interlacer results in non–stationarity along the motion trajectory. Such non–stationarities can be suppressed using a filter. Cost-effective filtering in the (spatio-) temporal domain can best be realised with a recursive filter. De Haan *et al.* [36] proposed a MC first–order recursive temporal filter:

$$F_{\rm O}(\vec{x},n) \begin{cases} kF(\vec{x},n) + (1-k)F_{\rm O}(\vec{x}-\vec{d}(\vec{x},n),n-1) \\ \qquad , \; y \bmod 2 = n \bmod 2 \\ pF_{\rm I}(\vec{x},n) + (1-p)F_{\rm O}(\vec{x}-\vec{d}(\vec{x},n),n-1) \\ \qquad , \text{otherwise} \end{cases} \quad (29)$$

where k and p are adaptive parameters, and $F_{\rm I}$ is the output of any initial de-interlacing algorithm. Preferably, a simple method is used, e.g. line averaging, which we selected for the evaluation. The derivation of k is fairly straightforward, and comparable to what we see in edge-preserving recursive filters, e.g. for motion–adaptive noise reduction.

A similar derivation for p is not obvious, since the difference would heavily depend upon the quality of the initial de-interlacer. In order to solve this problem, the factor p is selected such that the non–stationarity along the motion trajectory of the resulting output for interpolated pixels equals that of the vertically neighbouring original pixels. This assumption leads to:

$$p(\vec{x},n) = \frac{|A+B|+\delta}{2|F_{\rm I}(\vec{x},n) - F_{\rm O}(\vec{x}-\vec{d}(\vec{x},n),n-1)|+\delta} \quad (30)$$

with

$$\begin{aligned} A &= F_{\rm O}(\vec{x}-\vec{u}_y,n) - F_{\rm O}(\vec{x}-\vec{d}(\vec{x},n)-\vec{u}_y,n-1) \\ B &= F_{\rm O}(\vec{x}+\vec{u}_y,n) - F_{\rm O}(\vec{x}-\vec{d}(\vec{x},n)+\vec{u}_y,n-1) \end{aligned} \quad (31)$$

where δ, a small constant, prevents division by zero.

The recursion is an essential ingredient of the concept. Consequently, this method, similar to the TR algorithm of the previous subsection, has the risk of error propagation as its main disadvantage.

4.4 Interlace and generalized sampling

The sampling theorem states that a bandwidth–limited signal with maximum frequency $0.5f_s$ can exactly be reconstructed if this signal is sampled with a frequency of at least f_s. In 1956, Yen [37] showed a generalization of this theorem. Yen proved that any signal that is limited to a frequency of $0.5f_s$ can be exactly reconstructed from N independent sets of samples, respresenting the same signal with a sampling frequency f_s/N. This theorem can effectively be used to solve the problem of interpolation on a subsampled signal, as first presented by Delogne [38] and Vandendorpe [39]. We shall call this method the GST (Generalized Sampling Theorem) de-interlacer method.

Figure 12 shows the calculation of the samples to be interpolated. Samples from the previous field are shifted

over the motion vector towards the current field in order to create two independent sets of samples valid at the same temporal instance. A filter calculates the output sample. Appropriate filter coefficients are derived

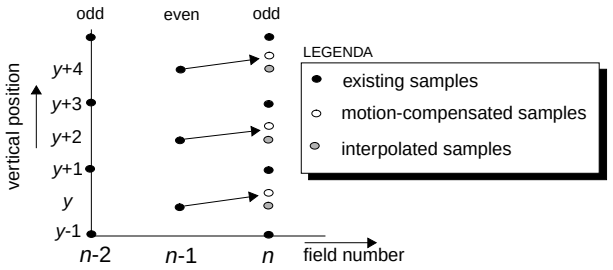


Figure 12: De-interlacing and generalized sampling

in the papers of Delogne [38] and Vandendorpe [39]. Kalker [40] shows an alternative (algebraic) derivation which does not require Fourier transforms (see also [41]). The de-interlaced output is defined by:

$$F_O(\vec{x},n) = \begin{cases} F(\vec{x},n) & , \; y \bmod 2 = n \bmod 2 \\ \sum_k F(\vec{x}-(2k+1)\vec{u}_y,n)h_1(k) + \\ \sum_m F(\vec{x}-d(\vec{x},n)-2m\vec{u}_y,n-1)h_2(m) \\ & , \text{otherwise} \end{cases} \tag{32}$$

The equations shows that output samples are completely determined by the *original* samples of the current and the previous field. No previously interpolated samples are used. Therefore, errors will not propagate, which is a clear advantage over the TR and the AR-algorithms.

To improve the robustness of this algorithm, some protection is necessary as shown by Bellers *et al.* [42]. The protection mentioned in this paper consists of a selective median filter, which activates the median only in the most critical situations. This prevents the disadvantages of median to outweigh the improved robustness. In the evaluation, this method will be referred to as 'GST de-interlacer with selective median'.

4.5 Hybrids with motion compensation

It is possible to combine MC and non-MC de-interlacing methods. Nguyen [43] and Kovacevic [44] describe de-interlacing methods that mix four methods: line averaging ($F_1(\vec{x},n)$), edge–dependent interpolation ($F_2(\vec{x},n)$), field averaging ($F_3(\vec{x},n)$) and MC field averaging ($F_4(\vec{x},n)$).

The output frame is defined by:

$$F_O(\vec{x},n) = \begin{cases} F(\vec{x},n), \; y \bmod 2 = n \bmod 2 \\ \sum_{j=1}^{4} k_j F_i(\vec{x},n) \quad , \text{otherwise} \end{cases} \tag{33}$$

The weights k_j associated with the corresponding interpolation methods, are determined by calculating the 'likely correctness' of the corresponding filter. The weights are calculated from the absolute difference of the corresponding method within a small region around the current position.

Kwon *et al.* [45] advocate switching instead of fading, and propose a decision on block basis. They include no edge adaptivity, but extend the number of MC-interpolators by distinguishing forward, and backward field insertion, as well as MC field averaging.

The fundamental problem with suchs hybrids is that averaging of the different methods introduces blurring, while switching requires a reliable quality ranking of the methods which is usually hard to achieve.

5 Evaluation

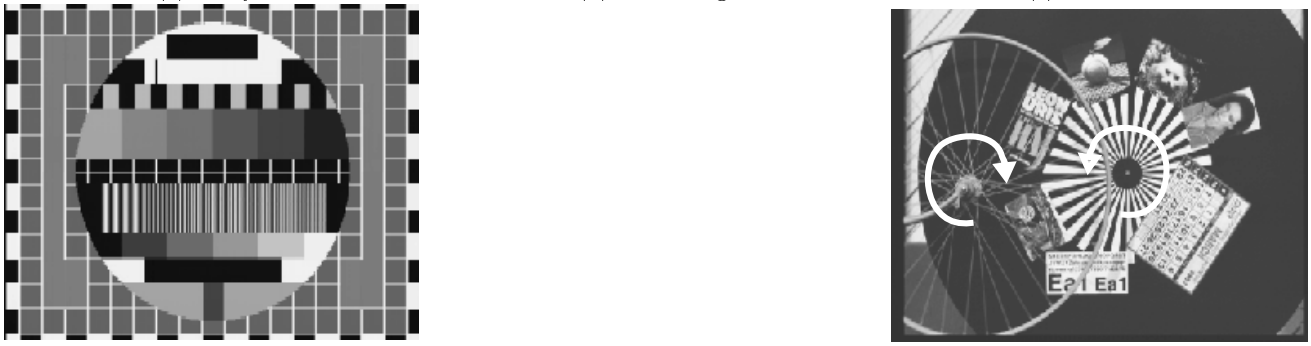
Video quality still is a subjective matter, as it proves difficult to design a reliable objective measure reflecting the subjective impression. Although many attempts have been reported [47], non of these appears to be widely accepted. Furthermore, we experienced difficulties in applying recent proposals, e.g. [46], since publications often do not provide all details while software is not (yet) made available. Some authors expressed their doubt whether their measure was applicable to evaluate de–interlacing. One alternative, the 'subjective MSE', that we experimented with [48] did not lead to different conclusions than the common MSE. We therefore see no good alternative yet for the much criticized MSE.

We conclude that objective measurements can help to rank the performance of the different methods, though screen pictures for the moment remain necessary to check conclusions. We will use Mean–Squared–Errors (MSE) and introduce an alternative applicable for interlaced originals. A consequence of the weak relation with subjective quality is that the conclusions based on our experiments have a rather qualitative character.

We selected some of the reviewed methods for evaluation. The selection criteria were popularity, availability in a product, or representativeness for a category. This led to twelve algorithms in the comparison: 1. line averaging, 2. field insertion, 3. linear VT–filtering [2], 4. VT median filtering, 5. weighted and edge-dependent median filtering, 6. MC median [3], 7. MC VT filtering, 8. TBP, 9. TR, 10. AR, 11. GST, and 12. GST with selective median [4].

[2] The first three methods are used in PC ICs, e.g. [15, 20]
[3] The methods 4., 5., and 6. are used in TV ICs, e.g. [4, 25, 49]
[4] The number of tested MC methods is largest, as technology

(a) Tokyo (b) Shopping (c) Calendar



(d) Circle (e) Bicycle

Figure 13: Images from each test sequence

.

We shall first introduce a criterion, applicable for interlaced originals, and the test sequences used in the evaluation. Thereafter, the score of the various methods is discussed, and screen photographs are presented to illustrate typical artifacts of the evaluated methods.

5.1 Performance measurement

In the literature on de–interlacing, the MSE is frequently used as an objective performance criterion. It requires progressively scanned original sequences, however, which are not necessarily representative for sequences recorded with an interlaced camera.

To cope with that problem, while preventing discussions on how to prefilter progressive originals prior to interlacing, an alternative error criterion, the so called Motion Trajectory Inconsistency MTI, was introduced by de Haan [35] and further used in [41, 42]:

$$MTI = \frac{1}{N_{\mathrm{MW}}} \sum_{\vec{x} \in \mathrm{MW}_n} \left(F_{\mathrm{O}}(\vec{x} - \vec{d}(\vec{x}, n), n-1) - F_{\mathrm{O}}(\vec{x}, n) \right)^2 \tag{34}$$

where MW_n indicates the Measurement Window in field n, and N_{MW} is the number of samples within that window summed over the length of the sequence.

The assumption is that two consecutive output images from a perfect de–interlacer are identical, even though one is derived from an odd input field and the other from an even. Obviously, motion has to be either absent or compensated. It is the mean square deviation from this ideal which is measured by the MTI. Due to the motion compensation, the score reflects not only the quality of the de-interlacer, but also that of the vectors. This may seem strange for non-MC methods, as their performance does not depend on motion vectors. However, as all methods suffer equally from erroneous vectors, MTI scores can be used for ranking, even though the values have no strict meaning.

Unfortunately, it is possible to design an algorithm – which consists in switching the output signal to zero– suggesting that even an MTI as low as zero, is insufficient to prove good quality. To validate our conclusions, we checked the MTI for the algorithms under test by calculating the classical MSE figures as well for those sequences of which a progressive original was available. We further added some screen photographs to give a subjective impression.

The motion estimator that we used is the so called 3-D RS block-matcher proposed by de Haan *et al.* [50]. This estimator yields close to true-motion vectors with

will soon justify these techniques in commercial products.

a quarter pel accuracy, and has a low complexity which earlier justified its use in consumer ICs [4].

5.2 Test sequences

We used five test sequences in the five categories: no motion, horizontal motion, vertical motion, zoom, and complex motion. Pictures from these test sequences are shown in Figure 13, along with an arrow indicating the motion. This material proved to be critical, and discriminates well between the various algorithms. From the sequences Bicycle, Circle, and Tokyo, progressive originals were available and used for an MSE score.

5.3 Results

To enable a quick performance comparison of the various methods we designed a 'star-graph' showing all MTI scores in a single graph. The star-graph shows five axes, each corresponding with a particular test sequence. The centre, where the axes meet, corresponds to the worst score obtained with any of the methods. The end of each axis corresponds to the best score obtained on the corresponding sequence. The MTI scores of a particular algorithm were drawn on the corresponding axes, connected with a line. The area of the star now indicates the overall quality, while the form suggests strengths and weaknesses. The figure in the graph indicates the average MTI score for the method, and the MSE score for the relevant sequences are indicated between brackets next to the name of the sequence. The Figures 14 and 15, show the results for the non-MC methods and the MC methods, respectively, from which we will draw our conclusions.

First of all, the MSE scores indicate that all methods produce realistic output signals, i.e. that the MTI is a good measure for ranking, even if no progressive original exists.

Field insertion shows the best MTI score for still images (Circle), but is worst of all for moving sequences. The line averaging shows the best average MTI score of all non-MC methods, but is worst for stationary images (Circle). Typical artifacts of line averaging and field insertion are shown in Figures 16 and 17, respectively.

The median filters perform better on still images than the linear ones, while the weighted median shows a relative good score on Bicycle (The rotor in this image is best interpolated along diagonal edges). The VT median introduces some alias in the highest vertical frequencies as shown in Figure 20, while the weighted median is weak in all detailed picture parts without dominant edges (Shopping, Tokyo). Figure 19 illustrates the effect which is due to a noisy interpolation direction. Robustness against erroneous edge detections is apparently lacking.

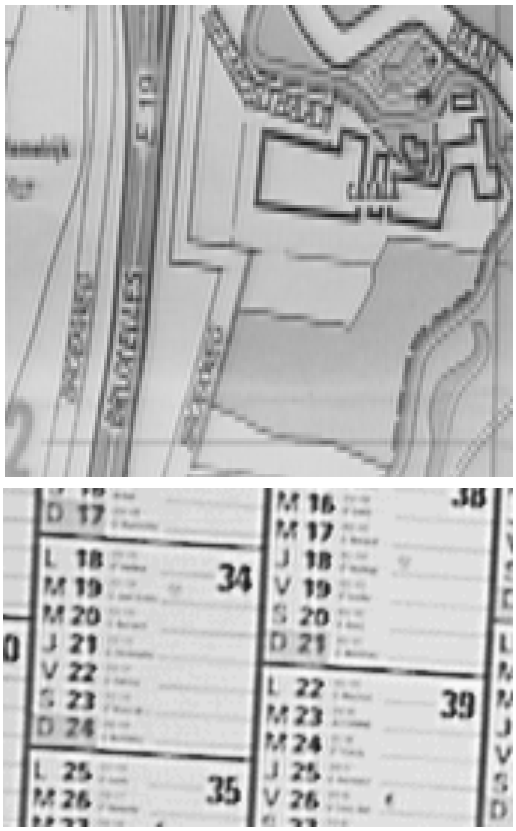

(a) (b)

Figure 16: a) Artefact due to spatial averaging filtering, b) Output of AR-method

The VT filter is typically weak on vertically moving sequences (Calendar, Bicycle), but best of the non-MC methods for horizontal motion (Tokyo). A screen photograph, shown in Figure 18, illustrates the problem with vertical motion clearly. Its performance on stationary images (Circle) is below average, resulting particularly in line–flicker.

MC methods perform better than non-MC algorithms, but TBP and GST have problems with complex motion (Bicycle), indicating that robustness against vector errors can be improved. Such lack of robustness can be dramatic, as is illustrated –for the TR-method in which we switched–off the protection– in Figure 20.

Motion compensation is less effective for methods that are poor on still images. An example is the VT filter.

The best tested methods are the AR, the GST with

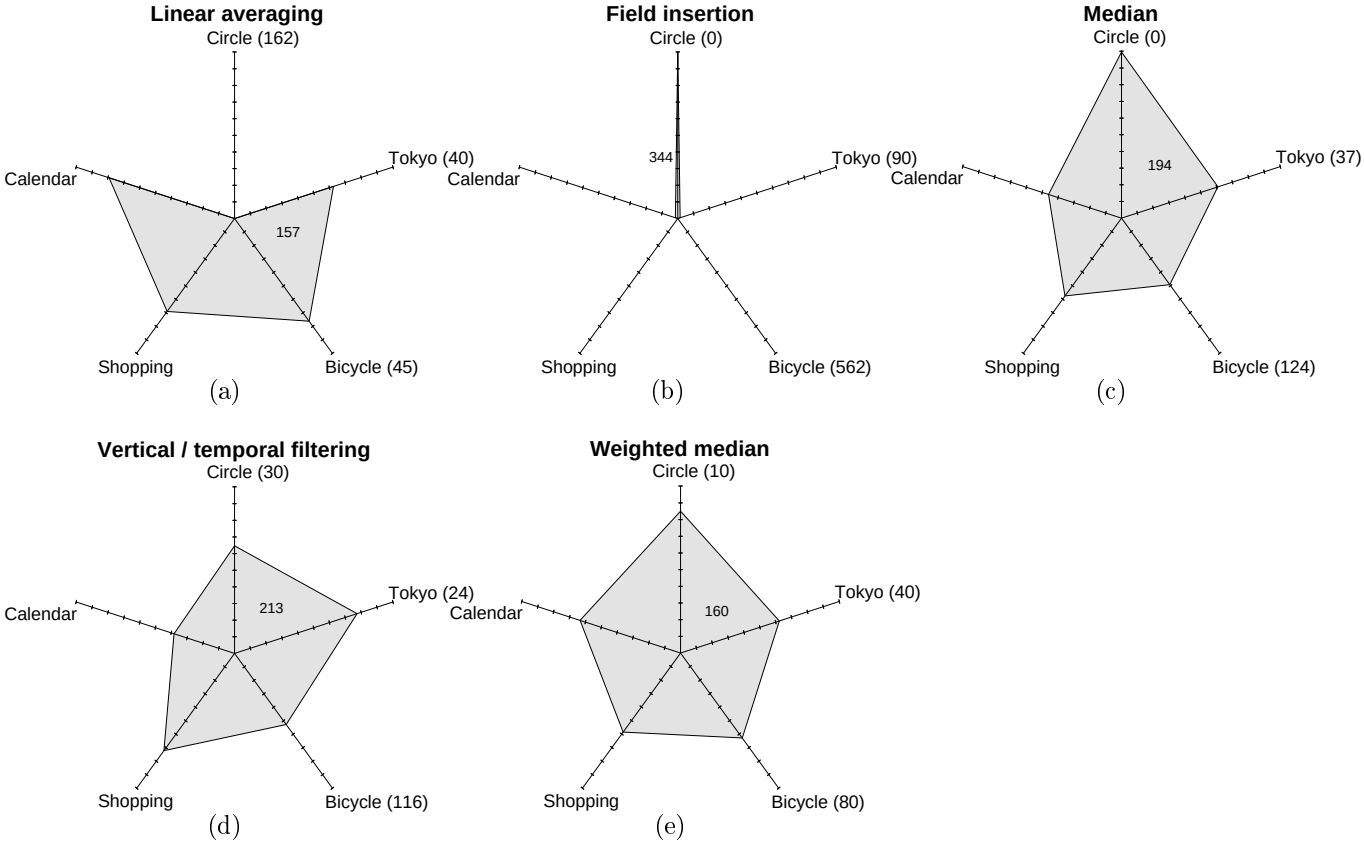


Figure 14: Results of the evaluation for the non MC de-interlacing algorithms

selective median, the TR, and the MC median method. Of these, AR and TR require a frame memory for recursion, and are therefore somewhat more expensive than the other two methods for which a field memory suffices. Their recursiveness makes the output sequence more 'stable' or less 'noisy', but this may be interpreted equally well as a loss of sharpness, particularly when compared with the GST methods.

The score of the MC median indicates that neglecting the SRC problem on subsampled data is better than lacking robustness.

6 Concluding remarks

We have presented an overview of de-interlacing techniques, ranging from simple linear methods to advanced motion compensated algorithms. We selected twelve methods for a performance comparison. These twelve include algorithms that are already available in (PC and TV) products, as well as algorithms from recent literature that could appear in future consumer products.

In the evaluation section we have compared the algorithms on critical test sequences. We included stationary, horizontally and vertically moving sequences, zooms, and material with complex motion. We showed objective scores, MSE and MTI, and screen photographs of the typical artifacts. The MTI scores were presented in a 'star-graph', a footprint of a method immediately showing its strenghts and weaknesses.

We conclude that the use of additional information extracted from the sequence, using motion detectors, edge detectors, and motion estimators, requires measures to garantee robustness in case of errors that inevitably occur in these extracted features. Nevertheless, the advantage of motion compensation was evident, and some algorithms had the required robustness to allow the use of a cost–effective motion estimator. We therefore expect that the de–interlacing quality of coming products shall greatly improve, and non-MC methods become obsolete in all but the least advanced products.

We further conclude that the methods used in TV receivers have shown clear improvements over time. Particularly, the MC methods recently introduced in consumer TV are considerably better than the earlier lin-

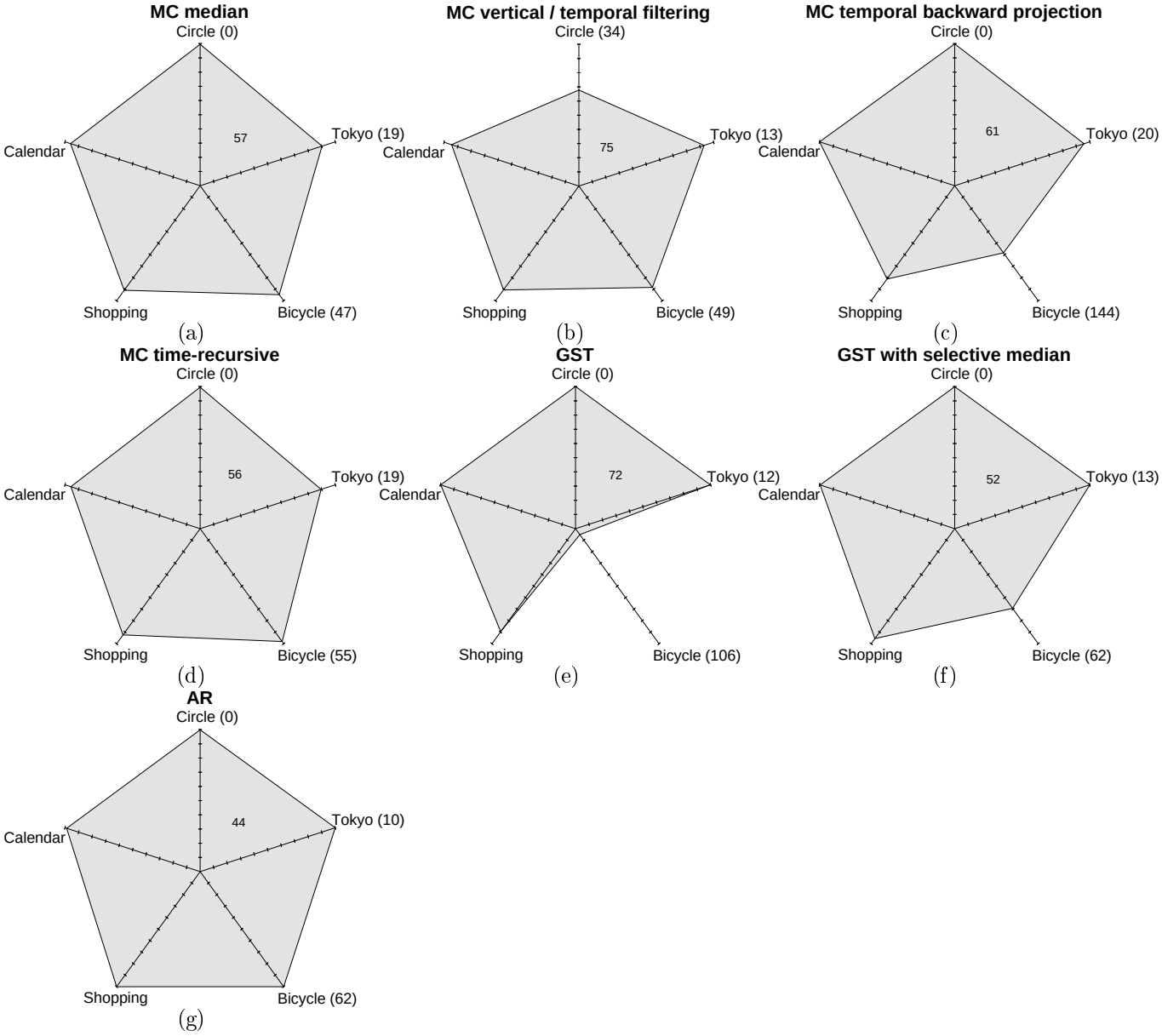


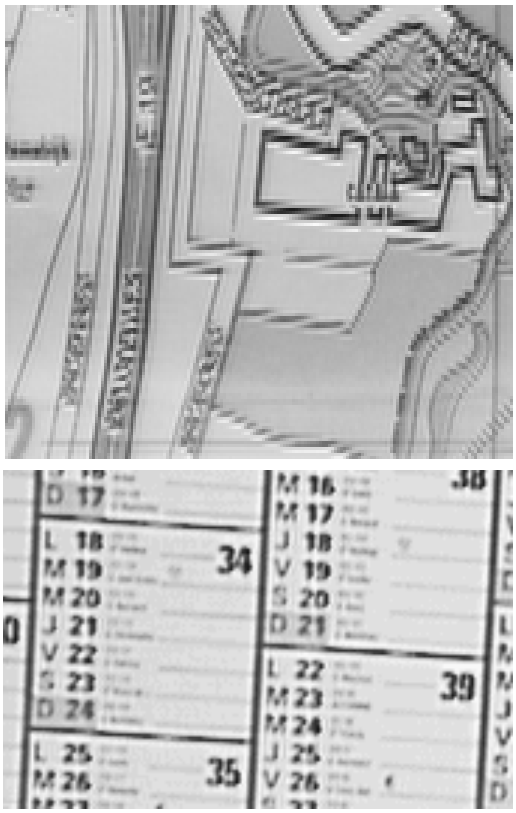
Figure 15: Results of the evaluation for the MC de-interlacing algorithms

ear and non–linear methods. Nevertheless, the older linear methods are state of the art in PC–products, although our evaluation indicates a relative poor performance. In addition, the evaluation showed that there is room for further improvement. The most recently proposed MC algorithms, not yet available in products, appear to be still better and affordable.

Concerning the question whether or not interlace should be part of future scanning format, we believe that the drawbacks of interlace are easily overestimated unless one is familiar with the recent developments in de–interlacing. The use of more advanced techniques in TV products may be a consequence of the belief of the TV community that compatibility with historical choices is a necessity in the very large TV market. On the other hand, the reluctance to embrace interlace may have caused out–dated techniques for de–interlacing in PCs. Our overview contributes to the discussion by providing a common knowledge basis. It does not prove that interlace is currently the best *technical* solution to reduce the transmission channel capacity with a factor of two. Further experiments with both advanced de–interlacing and coding techniques are required to quantify that and the most difficult task, of balancing the technical and non-technical issues, remains.

(a) (b)

Figure 17: a) Alias artefact due to field insertion, b) Output of AR-method



(a) (b)

Figure 18: a) Artefact due to vertical-temporal filtering in case of vertical motion, b) output from AR-method

As a closing remark it seems good to remember that many broadcasted pictures are originated on cine–film, i.e. can be perfectly de–interlaced within the PC or TV without deviating from the current interlaced transmission formats. The motion judder due to the low picture update frequency of cine–film can be eliminated. This requires the same motion vectors that enable successful MC de–interlacing of non–film material, and was proven by the award–winning 'Natural Motion' concept, commercially available in TV in Europe [4]. That concept also shows that motion portrayal problems can be solved economically when converting from one picture rate to another. Motion judder, resulting from repetition of the most recent picture and common procedure in all PC video cards, can be solved better with signal processing than with adapted display rates and long–persistent phosphors.

References

[1] E.W. Engstrom, 'A study of Television Image Characteristics. Part Two. Determination of Frame Frequency for Television in terms of Flicker Characteristics', *Proc. of the I.R.E.*, Vol. 23, no. 4, 1935, pp. 295-310.

[2] S. Pigeon and P. Guillotel, 'Advantages and Drawbacks of Interlaced and Progressive Scanning Formats', HAMLET report, RACE 2110, 1996.

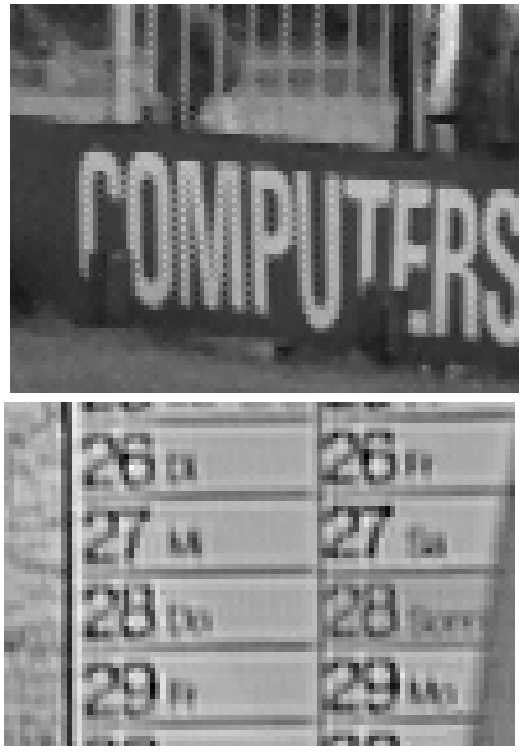
[3] E. Dubois, G. de Haan and T Kurita, 'Motion estimation and compensation technologies for standards conversion', Preface to special issue of Signal Processing: *Image Communication*, no. 6, 1994, pp.189-192.

[4] G. de Haan, J. Kettenis, and B. De Loore, 'IC for motion-compensated 100 Hz TV with natural-motion movie-mode', *IEEE Tr. on Cons. Electr.*, May 1996, pp. 165-174.

[5] Microsoft, 'Broadcast-Enabled Computer Hardware Requirements', *WinHEC'97, Broadcast Techologies White Paper*, 1997, pp. 11-12.

[6] L.J. Thorpe and T. Hanabusa, 'If Progressive Scanning is so good, how bad is Interlace?', *SMPTE Journal*, Dec. 1990, pp. 972-986.

[7] P. Guillotel and G. D'Agostino, 'Towards the Use of a Progressive Transmission Format', *Proc. Int.*



(a) (b)

Figure 19: a) Artefact due to weighted median filtering, b) Output of AR-method

(a) (b)

Figure 20: a) Artefact due to incorrect motion vector and without protection, b) Original proscan image

Workshop on HDTV and the Evolution of Television, Nov. 1995, Taipei, pp. 4A- 31-38.

[8] D. Bagni, G. de Haan, and V. Riva, 'Motion compensated post–processing for low bit rate video-conferencing on ISDN lines', To appear in *IEEE Tr. on Cons. Electr. Aug. 1997.*

[9] A.W.M. van den Enden and N.A.M. Verhoeckx, *Discrete-time signal processing*, Prentice Hall, ISBN 0-13-216763-8, 1989, pp. 233-.

[10] G.J. Tonge, 'Television motion portrayal' Presented at *Les Assises des Jeunes Chercheurs*, Sep. 1985, Rennes (Fr.).

[11] A. M. Tekalp, '*Digital video processing*', Prentice Hall 1995, ISBN 0-13-190075-7, pp. 250–.

[12] M. Achiha, K. Ishikura and T. Fukinuki, 'A Motion-Adaptive High-Definition Converter for NTSC Color TV Signals', *SMPTE Journal*, Vol. 93, no. 5, May 1984, pp. 470-476.

[13] R.S. Prodan, 'Multidimensional Digital Signal Processing for Television Scan Conversion', *Philips J. of Research*, Vol. 41, no. 6, 1986, pp. 576-603.

[14] A.M. Bock, 'Motion-Adaptive Standards Conversion Between Formats of Similar Field Rates', Signal Processing: *Image Communication*, Vol. 6, no. 3, Jun. 1994, pp. 275-280.

[15] N. Seth-Smith and G. Walker, 'Flexible Upconversion for High Quality TV and Multimedia Displays', *Proc. of the ICCE*, Jun. 1996, Chicago, pp. 338-339.

[16] P.D. Filliman, T.J. Christopher and R.T. Keen, 'Interlace to progressive scan converter for IDTV', *IEEE Tr. on Cons. Electr.*, Vol. 38, no. 3, Aug. 1992, pp. 135-144.

[17] C. Hentschel, 'Comparison between median filtering and vertical edge controlled interpolation for flicker reduction', *IEEE Tr. on Cons. Electr.*, Aug. 1989, Vol. 35, no. 3, pp. 279–289.

[18] C. Hentschel, *Television with increased image quallity; Flicker reduction through an increased picture rate in the receiver*, Ph.D. thesis, Institut fur Nachrichtentechnik, Tech. Univ. Braunschweig, Germany, Feb. 1990, (in German).

[19] T. Doyle and M. Looymans, 'Progressive Scan Conversion using Edge Information', in: *Signal*


(a)


(b)

Figure 21: a) Alias artefact due to VT–median filtering, b) Original proscan picture

Processing of HDTV, II, L. Chiariglione, Ed., Elsevier Science Publishers, 1990, pp. 711-721.

[20] *Preliminary data sheet of Genesis gmVLD8, 8 bit Digital Video Line Doubler*, version 1.0, Jun. 1996.

[21] M. Lee, J. Kim, J. Lee, K. Ryu and D. Song, 'A new algorithm for interlaced to progressive scan conversion based on directional correlations and its IC design', *IEEE Tr. on Cons. Electr.*, Vol. 40, no. 2, May 1994, pp. 119-129.

[22] J. Salonen, S. Kalli, 'Edge Adaptive Interpolation for Scanning Rate Conversion', in *Signal Processing of HDTV, IV*, E. Dubois and L. Chiariglione, Eds., Elsevier Science Publishers, 1993, pp. 757-764.

[23] M.J.J.C. Annegarn, T. Doyle, P.H. Frencken, and D.A. van Hees, 'Video signal processing circuit for processing an interlaced video signal', European Patent Application no. EP-A 0 192 292.

[24] H. Hwang, M.H. Lee and D.I. Song, 'Interlaced to progressive scan conversion with double smoothing', *IEEE Tr. on Cons. Electr.*, Vol. 39, No. 3, Aug. 1993, pp. 241-246.

[25] Philips Semiconductors, 'Datasheet SAA4990, PROZONIC', 1995.

[26] A. Lethonen and M. Renfors, 'Non-linear Quincunx Interpolation Filtering', *Proc. Vis. Comm. and Im. Proc. '90*, Lausanne, Oct. 1990, pp. 135-142.

[27] J. Salo, Y. Nuevo and V. Hameenaho, 'Improving TV Picture Quality with Linear-median Type operations', *IEEE Tr. on Cons. Electr.*, Vol. 34, no. 3, Aug. 1988, pp. 373-379.

[28] P. Haavisto, J. Juhola, and Y. Neuvo, 'Scan rate up-conversion using adaptive weighted median filtering', in: *Signal Processing of HDTV, II*, L. Chiariglione, Ed., Elsevier Science Publishers, 1990, pp. 703-710.

[29] R. Simonetti, S. Carrato, G. Ramponi and A. Polo Filisan, 'Deinterlacing of HDTV Images for Multimedia Applications', in *Signal Processing of HDTV, IV*, E. Dubois and L. Chiariglione, Eds., Elsevier Science Publishers, 1993, pp. 765-772.

[30] R.A.K. Jain, '*Fundamentals of Digital Image Processing*', Prentice-Hall International Inc., Englewood Cliffs, NJ, 1989.

[31] Y. Kim and Y. Cho, 'Motion Adaptive Deinterlacing Algorithm Based on Wide Vector Correlations and Edge Dependent Motion Switching', *Proc. of HDTV Workshop '95*, 1995, pp. 8B9 - 8B16.

[32] W. Bruls, A. van der Werf, R. Kleihorst, T. Friedrich, E. Salomons and F. Jorritsma, 'A single-chip MPEG-2 encoder for consumer storage applications', *Digest of the ICCE*, 1997, Chicago, pp. 262-263.

[33] J.W.Woods and Soo-Chul Han, 'Hierarchical Motion Compensated De-interlacing', *Proc. of the SPIE*, Vol. 1605, 1991, pp. 805-810.

[34] F.M. Wang, D. Anastassiou, and A.N. Netravali, 'Time-Recursive Deinterlacing for IDTV and Pyramid Coding', *Signal processing: Image Communications 2*, Elsevier, 1990, pp.365-374.

[35] G. de Haan and P.W.A.C. Biezen, 'Time-recursive de-interlacing for high-quality television receivers', *Proc. of Int. Workshop on HDTV'95*, Taipei, Taiwan, Nov. 1995, pp. 8B25-8B33.

[36] G. de Haan and E.B. Bellers 'De-interlacing of video data', *to be published in IEEE Tr. on Cons. Electr.*, Aug. 1997.

[37] J.L. Yen, 'On Nonuniform Sampling of Bandwidth-Limited Signals', *IRE Tr. on Circ. Th.*, vol. CT-3, Dec. 1956, pp. 251–257.

[38] P. Delogne, L. Cuvelier, B. Maison, B. Van Caillie, and L. Vandendorpe, 'Improved Interpolation, Motion Estimation and Compensation for Interlaced Pictures', *IEEE Tr. on Im. Proc.*, Vol. 3, no. 5, Sep. 1994, pp. 482–491.

[39] L. Vanderdorpe, L. Cuvelier, B. Maison, P. Quelez, and P. Delogne, 'Motion-compensated conversion from interlaced to progressive formats', Signal Processing: *Image Communication 6*, Elsevier, 1994, pp. 193–211.

[40] A.A.C. Kalker, 'Motion Estimation for Interlaced Video', to be published in *IEEE tr. on Image Processing.*

[41] E.B. Bellers and G. de Haan, 'Advanced deinterlacing techniques', *Proc. of the ProRISC/IEEE Workshop on Circ., Syst. and Sig. Proc.*, Mierlo, Nov. 27-28, 1996, pp. 7-17.

[42] E.B. Bellers and G. de Haan, 'Advanced motion estimation and motion compensated de-interlacing', *Proc. of the Int. Workshop on HDTV*, Oct. 1996, Los Angeles, Session A2.

[43] A. Nguyen and E. Dubois, 'Spatio-temporal adaptive interlaced-to-progressive conversion', in *Signal Processing of HDTV, IV*, E. Dubois and L. Chiariglione, Eds., Elsevier Science Publishers, 1993, pp. 749-756.

[44] J. Kovacevic, R.J. Safranek and E.M. Yeh, 'Deinterlacing by Successive Approximation', *IEEE Tr. on Im. Proc.*, Vol. 6, no. 2, Feb. 1997, pp. 339-344.

[45] J.S. Kwon, K. Seo, J. Kim and Y. Kim, 'A Motion-Adaptive De-interlacing Method', *IEEE Tr. on Cons. Electr.*, Vol. 38, no. 3, Aug. 1992, pp. 145-150.

[46] C J. van den Branden Lambrecht and O. Verscheure, 'Perceptual Quality Measure using a Spatio-Temporal Model of the Human Visual System', *Proceedings of the SPIE*, Vol. 2668, San Jose, Jan. 1996, pp. 450-461.

[47] A. Eskicioglu and P. Fisher, *'A survey of quality measures for gray scale image compression'*, *Proceedings of the Space and Earth Science Data Compression Workshop*, Apr. 1993, Snowbird, pp. 49-61.

[48] H. Marmolin, *IEEE Tr. on Systems, Man and Cybernetics*, Vol. 16, no. 3, May 1986. pp. 486-489.

[49] V. D'Alto, A. Cremonesi, C. Heintz, K. Oistamo, J. Urban, M. Karlsson, A. Vindigni and S. Dal Poz, 'A highly integrated processor for improved quality television', *Digest of the ICCE*, Chicago, Jun 1995, pp. 42-43.

[50] G. de Haan and P.W.A.C. Biezen, 'Sub-pixel motion estimation with 3-D Recursive Search Block Matching', Signal Processing: *Image Communication* 6, 1994, pp. 229-239.

[51] H. Hwang, 'Interlaced to progressive scan conversion for HD-MAC application', *IEEE Tr. on Cons. Electr.*, Vol. 38, no. 3, Aug. 1992, pp. 151-156.

Gerard de Haan was born in Leeuwarden, The Netherlands, on April 4, 1956. He received the B.Sc., the M.Sc., and the Ph.D. degree from Delft University of Technology in 1977, 1979 and 1992 respectively. In 1979 he joined the Philips Research Laboratories in Eindhoven. He led research projects in the area of image processing, and participated in European projects. He has coached students from various universities, and teaches since 1988 for the Philips Centre for Technical Training. In 1991/1992, he was a visiting researcher in the Information Theory Group of Delft University. At present, he is a Senior Scientist in the group Television Systems of Philips Research and has a particular interest in algorithms for motion estimation, scan rate conversion, and image enhancement. His work in these areas has resulted in about 35 patents and patent applications. He is a Senior Member of the IEEE, and received the first price in the 1995 ICCE Outstanding Paper Awards program. The Philips 100 Hz TV with 'Natural Motion', based on his PhD-study received the European Innovation Award of the Year 95/96 from the European Imaging and Sound Association.

Erwin Bellers was born in Enschede, The Netherlands, on November 14, 1965. He received the B.Sc degree from the Technische Hogeschool Enschede in 1987 and the M.Sc degree (with distinction) from the University of Twente in 1993. In December 1993, he joined Philips Research Laboratories in Eindhoven as a Research Scientist in the Television Systems Group. He worked on several video processing projects, with

major interest in video enhancement algorithms. At present Mr. Bellers contributes to the area of motion estimation, motion compensation, de-interlacing and scan rate conversion.