BBC RD 1996/9

Research and Development Report

A COMPARISON OF MOTION-COMPENSATED INTERLACE-TO-PROGRESSIVE CONVERSION METHODS

G.A. Thomas, M.A., Ph.D., C.Eng., M.I.E.E.

Research and Development Department
Policy and Planning Directorate
THE BRITISH BROADCASTING CORPORATION

A COMPARISON OF MOTION-COMPENSATED INTERLACE-TO-PROGRESSIVE CONVERSION METHODS

G.A. Thomas, M.A., Ph.D., C.Eng, M.I.E.E.

Summary

The conversion of interlaced video to progressive format may be carried out in a number of ways, including simple vertical interpolation, fixed or motion-compensated vertical-temporal interpolation, and approaches based on motion compensation using non-uniform sampling theory. Motion-compensated methods are the only ones which offer the possibility of maintaining high vertical resolution with minimal aliasing for the majority of vertical motion speeds. However, the accuracy and reliability of the vectors has a significant impact on the performance of such methods. For some applications, such as display conversion using vectors recovered from an MPEG-2 coded signal, the vector signal available will be less than perfect. This Report considers several motion-compensated interlace-to-progressive conversion methods, and assesses their performance with both accurate and inaccurate vectors. A way of making any motion-compensated method tolerant to gross vector errors is proposed and evaluated. Comparisons are made with non-motion-compensated methods, and a recommendation is made of a method suitable to use with vectors recovered from an MPEG-2 bitstream.

Issued under the Authority of

Research & Development Department
Policy & Planning Directorate
BRITISH BROADCASTING CORPORATION

Head of Research & Development
Research & Development Department

A COMPARISON OF MOTION-COMPENSATED INTERLACE-TO-PROGRESSIVE CONVERSION METHODS

G.A. Thomas, M.A., Ph.D., C.Eng., M.I.E.E.

A COMPARISON OF MOTION-COMPENSATED INTERLACE-TO-PROGRESSIVE CONVERSION METHODS

G.A. Thomas, M.A., Ph.D., C.Eng., M.I.E.E.

1. INTRODUCTION

The conversion of an interlaced image signal to a progressively-scanned one is a key step in any standards conversion process which operates in either the vertical or the temporal domains. The advent of flat-panel displays, which inherently are progressively-scanned devices, will create the need to perform this conversion in many video displays. It should be noted that there is no 50 Hz progressive mode in the main profile, main level of MPEG-2.

Much has been recorded in previous publications concerning interlace-to-progressive conversion methods. Motion-adaptive switching between vertical and temporal interpolation is one approach[1]; similar performance can be achieved without the need for switching by using fixed spatio-temporal interpolation filters[2]. Other adaptive methods include the use of vertical-temporal median filters[3], and techniques based on identifying the spatial slope of lines[4]. Potentially, the best performance may be achieved using motion-compensated methods[2,5], which offer the possibility of maintaining high vertical resolution with minimal aliasing for the majority of vertical motion speeds, except those very close to speeds of an odd number of picture lines per field period.

Since in the future many displays are likely to be fed by signals decoded from an MPEG-2 bitstream, this produces the possibility for giving a display device access to the motion vectors from the MPEG-2 coding process. This could allow motion-compensated interlace-to-progressive conversion to be carried out without the need to incorporate a motion estimator in the display converter. Nevertheless, the vectors recovered from the bitstream are unlikely to be of the highest quality: the MPEG-2 standard divides the vector field into ‘macroblocks’ of 16 × 16 samples, and quantises vertical displacements to either half a picture line or half a field line, depending on the prediction mode. Furthermore, vectors generally refer to the average motion over several field periods, rather than the motion between adjacent fields.

This Report compares a number of interlace-to-progressive conversion methods, and considers the impact of inaccuracies and errors in the vector field on those methods using motion compensation. The following areas are covered and summarised:

- conversion techniques;
- performance with ‘perfect’ vector information, to show what level of performance is theoretically possible;
- the effect of small inaccuracies in the vectors;
- the effect of noise in the image;
- how the motion-compensated algorithms may be modified to reduce the effect of large vector errors;
- the results obtained when the algorithms were used to process natural image sequences, using vectors recovered from an MPEG-2 bitstream;
- conclusions and summary of the work.

2. DESCRIPTION OF THE ALGORITHMS CONSIDERED

Various interlace-to-progressive interpolation methods were assessed and the following gives basic details of them:

2.1 1/2, 1/2 vertical filter

The use of a simple intra-field vertical filter.

2.2 Fixed vertical-temporal filter

A fixed vertical-temporal filter, with an aperture of three fields and nine picture lines, was next assessed. The coefficients in the outer fields summed to zero, guaranteeing full temporal bandwidth for low vertical frequencies and thereby avoiding motion blur[2]. The coefficient values, taken from Ref. 6, are shown in Table 1 *(overleaf)*.

2.3 Vertical-temporal filter with even motion compensation

The vertical-temporal filter described above was steered to follow vertical motion in steps of two pic-

Table 1: Coefficients of the vertical-temporal filter (the central '1' acting to pass existing lines unchanged).

	Lines								
Fields	0.031	0.0	–0.116	0.0	0.170	0.0	–0.116	0.0	0.031
	0.0	–0.026	0.0	0.526	1.0	0.526	0.0	–0.026	0.0
	0.031	0.0	–0.116	0.0	0.170	0.0	–0.116	0.0	0.031

ture lines per field period. That is, for vertical motion speeds below one picture line per field period it was unchanged from the fixed filter. For speeds between one and three picture lines per field period the coefficients in the outer fields were moved up and down by two picture lines in the direction of motion, and so on. Note that it is not possible to compensate a three-field filter such as this for fractional motion speeds without reducing the vertical resolution and the degree of alias suppression; this is because the temporal cut-off frequency cannot be reduced and is a direct consequence of using only three fields. More details of this are discussed in Section 2.4, following. The filter was also compensated for horizontal motion to a resolution of 0.25 pixels per field period, using a four-tap horizontal filter to perform sub-pixel interpolation.

2.4 Motion-compensated 6.25 Hz temporal low-pass filter

Interlace-to-progressive conversion may be achieved using a motion-compensated low-pass temporal interpolation filter to remove the alias spectra caused by interlaced sampling. A spatial interpolator must be used to allow the filter to be skewed for motions of fractions of a pixel or picture line per field period. The pass-band of the filter must be skewed so that it is centred on the wanted baseband spectrum, and it must be sufficiently narrow to reject the alias spectrum centred on 25 Hz and 288 cycles per active picture height (c/aph). The pass-band of the temporal filter should be progressively reduced as the motion speed rises towards the 'strobe' speed of one picture line per field period, since the baseband and alias spectra approach each other as they are skewed by the motion. This implies a maximum bandwidth of 12.5 Hz (with a null at 25 Hz) when the image is stationary, reducing to 6.25 Hz for a motion speed of half a picture line per field period, and 3.125 Hz for a motion speed of 0.75 picture lines per field period. Theoretically, such a filter allows the full vertical resolution to be maintained, without aliasing, for all motion speeds except those very close to the strobe speed. As the strobe speed is approached, the large number of temporal taps in the filter required to achieve a null at a suitably low frequency will lead to practical problems.

For the experiments reported here, a seven-field temporal low-pass filter with a cut-off frequency of 6.25 Hz was used for all motion speeds up to 0.5 picture lines per field period. This avoided the need for a very large number of temporal taps, and the need to change the filter pass-band for different motion speeds. For vertical motion, the filter was compensated to the nearest 0.25 picture line per field period, using an eight-tap vertical interpolator to perform vertical interpolation within each frame (although, in practice, four of the eight contributions always came from lines of inserted zeroes). The temporal filter had nulls at 25 Hz, 18.75 Hz and 12.5 Hz to ensure rejection of the alias (at 288 c/aph, 25 Hz) for each compensation velocity. The filter coefficients (multiplied by 16) were:

$$\left(2-\sqrt{2}\right), 2, \left(2+\sqrt{2}\right), 4, \left(2+\sqrt{2}\right), 2, \left(2-\sqrt{2}\right)$$

To avoid motion speeds near the strobe speed, the value of the vertical vector used was modified as follows:

- for rounded motion speeds of 0.75 and 1.0 picture lines per field period, a vertical vector of 0.5 picture lines per field period was used;
- for a rounded speed of 1.25, a value of 1.5 was used.

This pattern was repeated for all vertical motion speeds, in a 'modulo 2' manner. Avoiding the speed of 0.75 picture lines per field period period obviated the need for a temporal filter with a cut-off of 3.125 Hz, which would have required around 15 temporal taps for adequate performance. Ideally, a simple vertical filter should have been used for speeds around the strobe speed; the only reason that the temporal filter was used at this speed was to simplify software implementation. The filter was also compensated for horizontal motion to a resolution of 0.25 pixels per field period, in the same way as the vertical-temporal filter.

Strictly speaking, this filter was a vertical-temporal filter rather than a purely temporal filter, since vertical interpolation was necessary to deal with fractional vertical speeds. However, the vertical element of the filter was (ideally) of an all-pass nature; the predominant filtering action was low-pass temporal. Conversely, the *fixed vertical-temporal (fixed v-t)* and *vertical-temporal with even motion-compensation (v-t with* *even mc)* filters were designed to attenuate both vertical and temporal frequency components. Thus, it was felt to be justified to distinguish this filter from the previous two by the terms *temporal* and *vertical-temporal,* respectively.

2.5 General sampling theorem two-field filter

An alternative to an approach based on temporal filtering is to treat the samples in two successive fields as unevenly-spaced samples of the same signal, and use the so-called *general sampling theorem* (*gst*) to interpolate samples in the required position[5]. This approach also allows full vertical resolution to be obtained up to speeds close to the strobe speed. As the strobe speed is approached, the coefficients of the filter become large, leading to noise amplification.

A family of filters was designed according to the method described in eqn. (29) [in Ref. 5], with the vertical cut-off parameter *p* equal to 0.6 (corresponding to a Kell factor of 0.7)* and a vertical extent of 13 picture lines. Filters were designed for motion speeds of multiples of 0.25 picture lines per field period, and a '1/2, 1/2' vertical filter was used for odd numbers of picture lines per field period. Contributions were taken from the present and the preceding field, corresponding to the 'backward only' mode described in [Ref. 5]. The filter was also compensated for horizontal motion to a resolution of 0.25 pixels per field period (as for the previous filters).

2.6 General sampling theorem three-field filter

This approach was very similar to the two-field version described immediately above, but contributions were taken from both the preceding *and* following fields, by averaging the coefficients for a 'forward-only' and 'backward-only' version of the filter. This made a time-symmetrical filter.

3. PERFORMANCE WITH 'PERFECT' VECTORS

To assess the performance of these interlace-to-progressive conversion methods in the presence of vertical motion and to examine their tolerance to inaccurate vectors, some experiments were conducted using a sequence containing synthetic movement. A portion of the test picture 'Formal Pond' was selected

Fig. 1 - The portion of the test image 'Formal Pond' (size 360 × 288) that was moved synthetically to generate test sequences.

(Fig. 1), which contained a number of sharp straight lines at a range of angles. This was moved using a very high-order interpolator to produce a progressively-scanned sequence that was stationary for eight fields, then accelerated vertically at a rate of 0.04 picture lines per field period per field period (sic) until reaching a speed of two picture lines per field period, 58 fields after the start of the sequence. The speed was held constant at two picture lines per field period for the remaining ten fields of the sequence. The effect of camera integration was not modelled, so the sequence corresponded to that from a camera with a very short shutter time. An interlaced sequence was derived from the progressive version by discarding alternate lines (the vertical resolution of the test picture was judged to be not atypical of that from an interlaced source, so no pre-filtering was applied before interlacing). The sequence thus allowed an objective measurement of the performance of interlace-to-progressive converters to be assessed, by comparing the processed sequence to the progressive original. The subjective quality of the converted sequences was also assessed, using a 50 Hz progressive monitor. Fig. 2 *(overleaf)* shows the RMS error that each interpolator produced using 'perfect' vectors.

3.1 Comparison of simple and fixed vertical-temporal interpolators

The simple 1/2, 1/2 vertical interpolator gave a significant interpolation error of around seven to eight grey levels RMS; it fluctuated as the image moved at fractional motion speeds, since the alias signal changed phase rapidly. Poor vertical resolution and aliasing were visible in the converted image, most notably on diagonally-sloping edges. The aliasing became particularly visible as the image moved, since the aliases

* The ratio of the usable vertical resolution to the theoretical maximum vertical resolution (given by the Nyquist sampling criterion as half the number of scanning lines per picture height). Typical values:- 0.7-0.8 for a progressive display; 0.6-0.7 for interlaced displays.

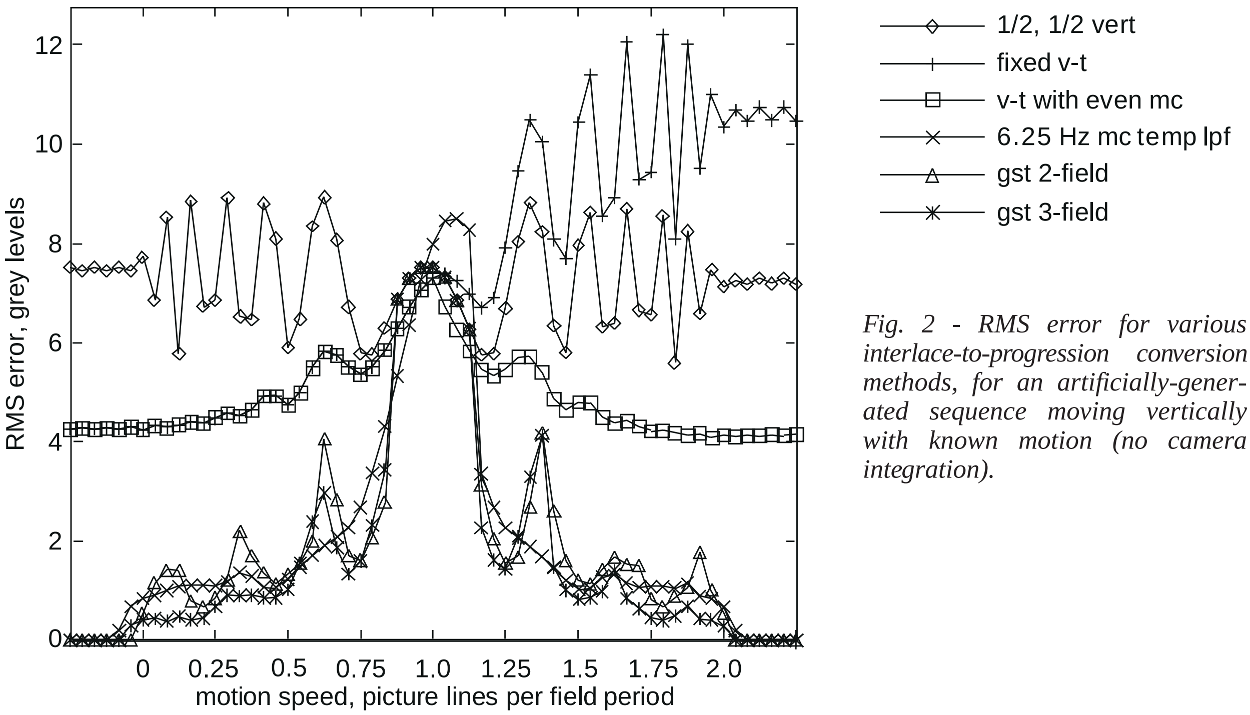


Fig. 2 - RMS error for various interlace-to-progression conversion methods, for an artificially-generated sequence moving vertically with known motion (no camera integration).

moved differently to the picture material. The fixed vertical-temporal interpolator performed better than the simple vertical interpolator for very low speeds, but aliasing quickly became apparent as the speed increased. For motion speeds over one picture line per field period, the fixed vertical-temporal interpolator performed worse than the simple vertical interpolator, since the contributions from adjacent fields were mispositioned significantly.

3.2 Application of motion compensation with *even* vertical accuracy

The application of motion compensation (quantised vertically to the nearest even number of picture lines per field period) to the vertical-temporal filter significantly improved its performance for velocities around two picture lines per field period. Indeed, for all velocities of an even number of picture lines per field period, its performance returns to that of the fixed filter acting on a stationary image. However, vertical aliasing was still clearly visible for intermediate speeds.

3.3 Application of motion compensation with *fractional* vertical accuracy

The three interpolators that used motion compensation to fractional vertical accuracy all performed very well, and all gave broadly similar results. The filters based on the 'general sampling theorem' produced slightly lower RMS errors than the motion-compensated temporal interpolator at the motion speeds for which they were designed (multiples of 0.25 picture lines per field period). The two-field filter was particularly sensitive to differences between the design and actual velocity, showing clear peaks in error at velocities mid-way between the design values; the three-field filter was much better in this respect. The 6.25 Hz temporal filter tended to give slightly higher errors overall, but was less sensitive to differences between the design and actual velocity. The temporal filter gave higher errors than all other filters tested for speeds around one picture line per field period, since no switch was made to a pure vertical filter for this speed, as explained earlier. Subjectively, all three interpolators that used motion compensation to fractional vertical accuracy performed very similarly, yielding much better picture quality than the other methods considered for all speeds except those close to one picture line per field period, for which they performed similarly.

3.4 The effect of camera integration

To assess the impact of camera integration, the experiments were repeated using a similar artificially-generated sequence, the generation of which included modelling the effect of a camera with an integration time of one field period. The results are shown in Fig. 3. Camera integration provides a pre-filtering action which attenuates high vertical frequencies in the presence of vertical movement, providing some degree of anti-alias pre-filtering. As expected, the RMS errors for all methods were reduced at the higher motion speeds, compared to the results with no integration. For example, at two picture lines per field period, the error from the vertical and the fixed vertical-temporal filters were both reduced by approximately 40%, and that from the vertical-temporal filter with even motion

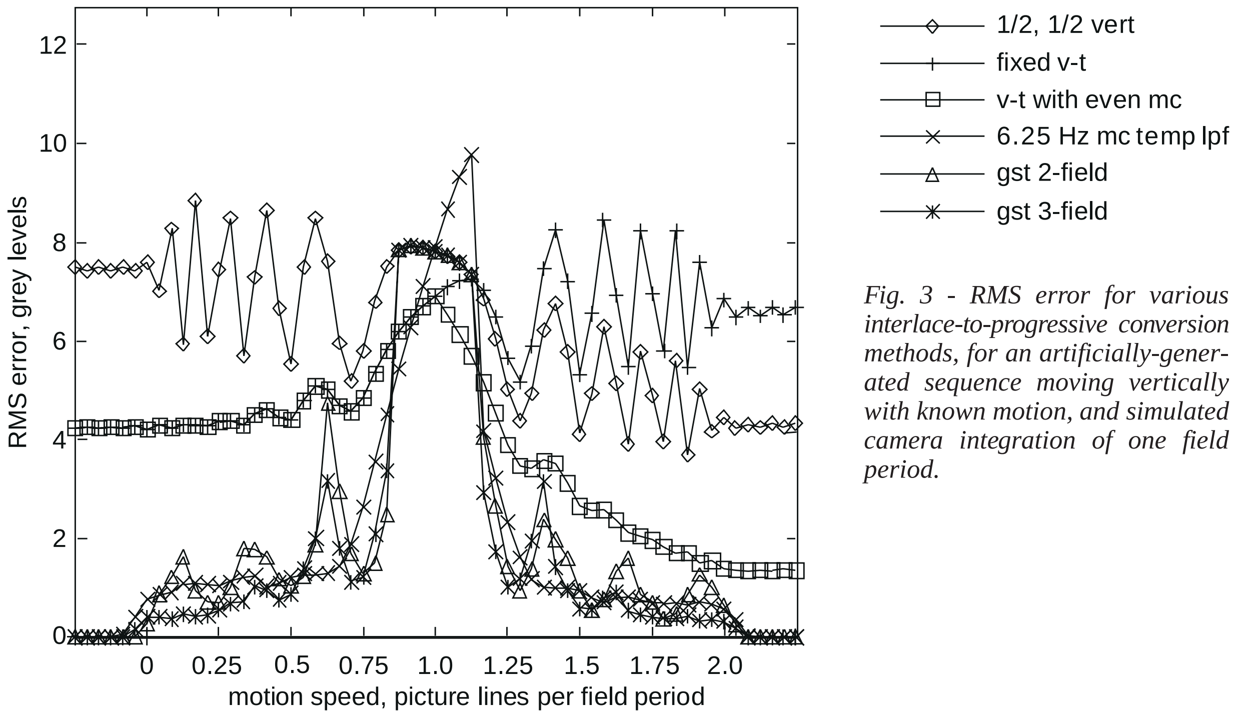


Fig. 3 - RMS error for various interlace-to-progressive conversion methods, for an artificially-generated sequence moving vertically with known motion, and simulated camera integration of one field period.

compensation was reduced by approximately 65%. The performance of the other motion-compensated systems, which gave low errors even in the absence of camera integration, were essentially unchanged. Thus the effect of camera integration was to reduce the differences in performance between the various methods studied. Although there was still a significant advantage to be gained by the use of motion compensation, the performance of the simplest motion-compensated method (*v-t with even mc*), Section 3.2 above, started to approach that of the more sophisticated methods as the motion speed rose above one picture line per field period.

4. ASSESSMENT OF INTERLACE-TO-PROGRESSIVE CONVERSION METHODS FOR VECTORS WITH SMALL ERRORS

Having confirmed that significant gains in picture quality of interlace-to-progressive conversion can be made, given accurate vectors, an experiment was carried out to see how tolerant the various algorithms were to small errors in the motion vector.

4.1 The effects of motion vector errors in the vertical component

The test sequence from the first experiment (without camera integration) was again used, but the motion vector used to compensate the filters was fixed at 0.5 picture lines per field period (this speed was chosen as it is mid-way between the ‘special case’ speeds of zero and one picture lines per field period). Fig. 4 *(overleaf)* shows the resulting RMS errors. Note that the filter *v-t with even mc* is not shown in this figure, since its performance is identical to the *fixed v-t* filter for this vector value, as it is below one picture line per field period. The plots for the fixed filters are, of course, the same as those in Fig. 2, and have been included for comparison.

It can be seen that the motion-compensated filters performed better than the fixed filters for velocities close to the correct motion velocity, but that their performance degraded significantly as the motion vector error increased, giving around twice the RMS error when the difference between assumed and true velocities was greater than about one picture line per field period. The two-field ‘general sampling theorem’ filter was the least tolerant to vector errors; for example, the fixed vertical-temporal filter gave better results than this filter for vector estimation errors of 0.25 picture lines per field period or more. The three-field version of this filter was significantly more tolerant to vector errors, and performed better than the *fixed v-t* filter for actual motion speeds within about ±0.5 picture lines per field period of the assumed speed. The motion-compensated 6.25 Hz temporal low-pass filter showed about the same overall tolerance to vector error as the three-field ‘general sampling theorem’ filter, being slightly worse than this filter for actual speeds below the assumed speed but slightly better for actual speeds above.

From these results it appears that the vertical component of the motion vector must be known to an accuracy of about 0.25 picture lines per field period in order to achieve a worthwhile gain over a simpler fixed vertical-temporal filter. The additional tolerance of the three-field ‘general sampling theorem’ filter

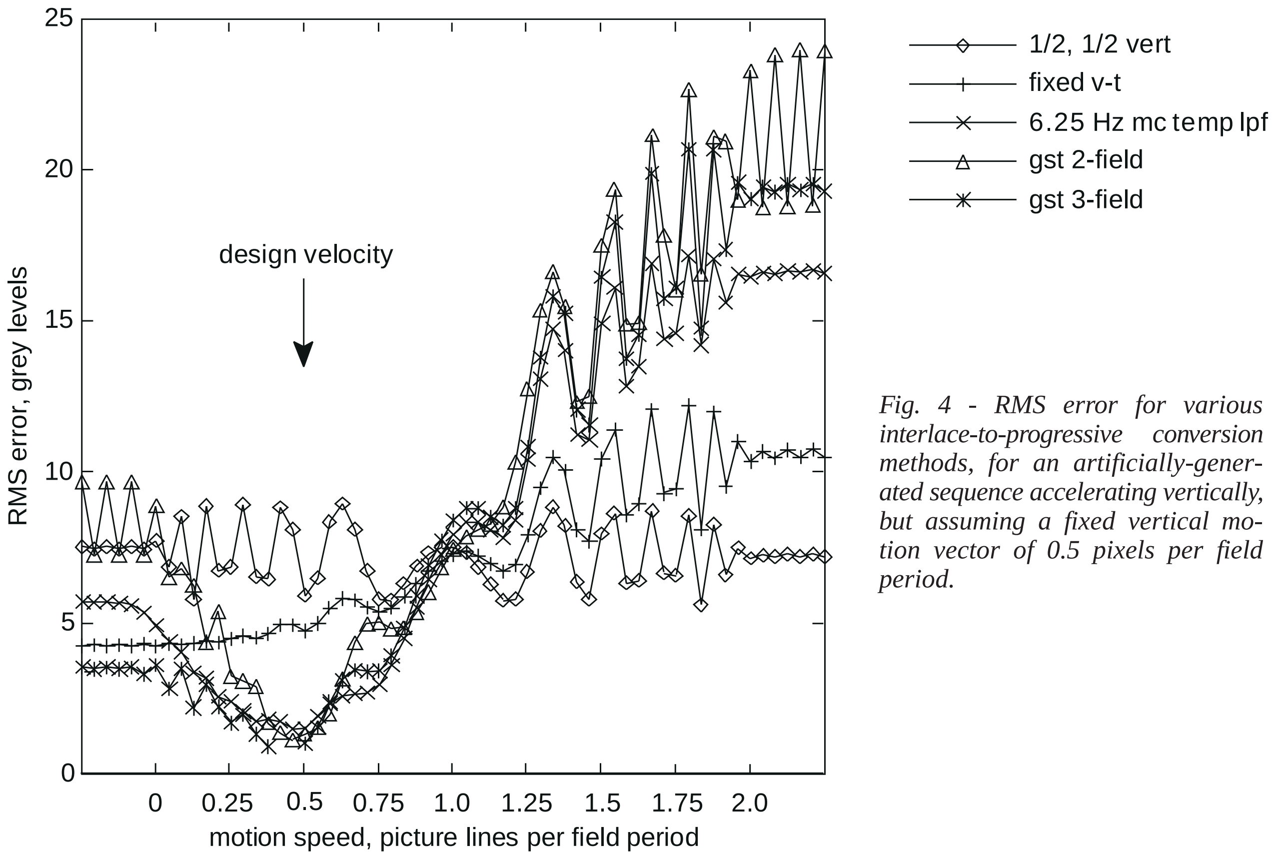


Fig. 4 - RMS error for various interlace-to-progressive conversion methods, for an artificially-generated sequence accelerating vertically, but assuming a fixed vertical motion vector of 0.5 pixels per field period.

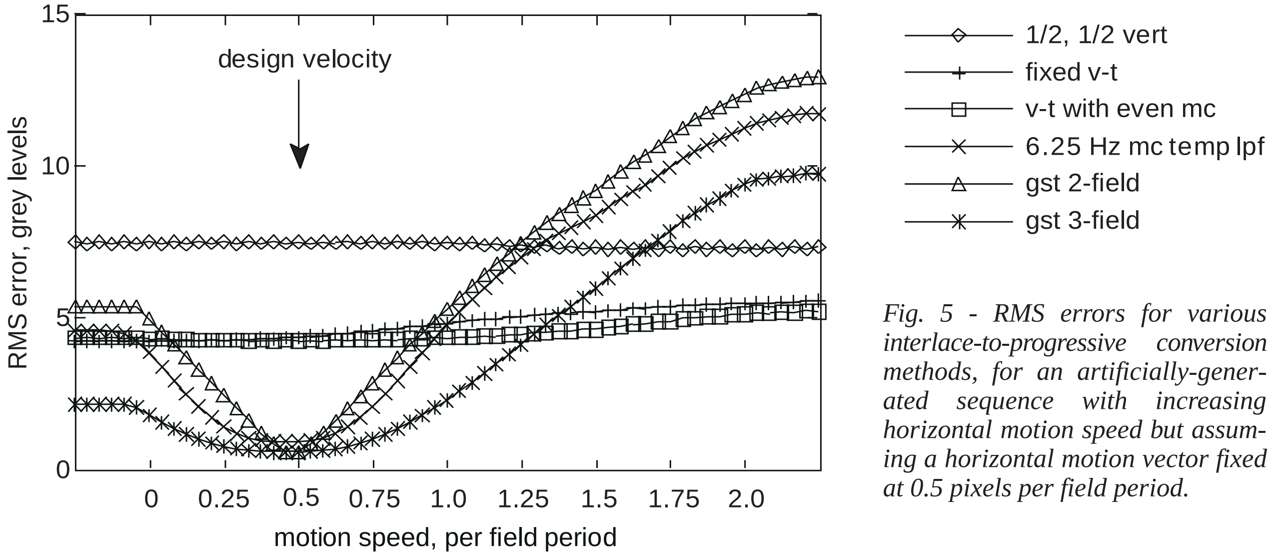


Fig. 5 - RMS errors for various interlace-to-progressive conversion methods, for an artificially-generated sequence with increasing horizontal motion speed but assuming a horizontal motion vector fixed at 0.5 pixels per field period.

over the two-field version is likely to be of benefit, particularly when motion vector errors regularly exceed 0.2 picture lines per field period.

4.2 The effects of motion vector errors in the horizontal component

The tolerance of the various methods was also assessed for errors in the horizontal motion vector, using a test sequence with horizontal artificial movement but otherwise similar to that used for the vertical motion experiments. The motion speed varied from zero to two pixels per field period. Each motion-compensated process was given a vector value fixed at 0.5 pixels per field period. Fig. 5 shows the results of this experiment.

Comparing Figs. 5 and 4 (which have deliberately been drawn with the same vertical scale) it is clear that errors in the horizontal component of the motion vector had a much smaller effect than errors in the vertical component. One reason for this is that uncompensated vertical motion can give rise to aliasing, whereas

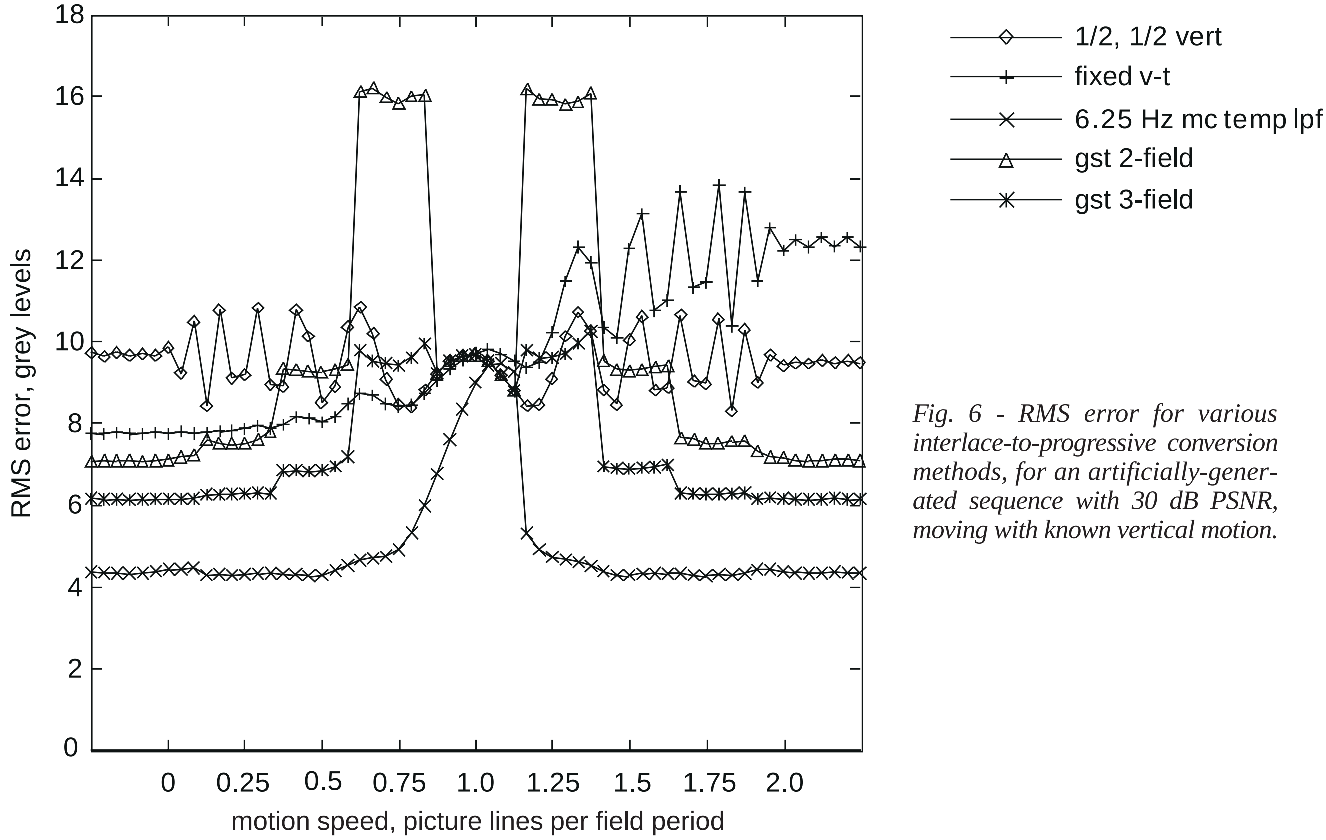


Fig. 6 - RMS error for various interlace-to-progressive conversion methods, for an artificially-generated sequence with 30 dB PSNR, moving with known vertical motion.

uncompensated horizontal motion merely reduces the resolution of the interpolated lines. The relative tolerance of the motion-compensated methods to horizontal vector errors was similar to that found earlier for vertical errors. It is interesting to note that the *v-t with even mc* filter is very tolerant to the presence of uncompensated motion, since it takes a relatively small amount of energy from adjacent fields, and furthermore has a small and symmetrical temporal aperture. However, its overall performance was poorer than the other motion-compensated filters. These results suggest that horizontal motion vector accuracy is not critical, and that the vector resolution of a quarter of a pixel per field period (the resolution likely to be available from vectors recovered from an MPEG-2 bitstream) is adequate for even the least tolerant method examined. Lower resolution, to the nearest half or integer pixel, would be sufficient for the *v-t with even mc* filter, and would introduce only small errors for the *gst three-field* filter.

5. THE EFFECT OF NOISE IN THE IMAGE SIGNAL

Another factor that needs to be considered is the tolerance to noise of the various conversion methods. This is of particular relevance when converting the *output* of an MPEG-2 decoder to progressive format, due to the presence of coding noise. In particular, filters based on the ‘general sampling theorem’ would be expected to have a poor noise performance for motion speeds around odd numbers of picture lines per field period, because the filters used at such speeds tend to have gains much greater than unity for certain frequencies. Conversely, an approach based on temporal low-pass filters might not be expected to be so sensitive, since such filters only ever *attenuate* components of the signal spectrum. Indeed, temporal filters may help to reduce the coding noise. To investigate this, a test sequence was made by adding noise corresponding to a peak signal-to-noise ratio of 30 dB to the interlaced test sequence with artificial vertical movement (without camera integration) used for the earlier experiments.

The sequence was converted to progressive using the methods described earlier, and the results compared to the original clean progressive sequence. Fig. 6 shows the RMS errors.

As expected, the filters based on the ‘general sampling theorem’ were much more sensitive to noise than the temporal filter. The two-field filter amplified the noise significantly; the RMS error for 0.75 picture lines per field period being significantly higher than that from the simple vertical interpolator. The three-field filter was less sensitive to noise, but still produced RMS errors which rose to about double those from the temporal filter at a speed of 0.75 picture lines per field period.

Subjectively, the sudden appearance of noise at certain speeds was annoying, particularly in the case of the *gst* *two-field* filter, reducing the overall picture quality to a similar level to that from the simple vertical or vertical-temporal filter. The temporal filter produced the best subjective results, but would of course be more expensive to implement, as the filter aperture was seven fields, compared to two or three.

6. ENSURING TOLERANCE TO VECTORS WITH LARGE ERRORS

The tolerance of the various interlace-to-progressive conversion algorithms has already been discussed with respect to small vector errors. However, algorithms also need to be tolerant to significant errors in the vector field, since perfect vectors will never be available 100% of the time. This is particularly important when considering display conversion using vectors from the MPEG-2 bitstream, since the block-based nature of the vectors makes it impossible to convey a perfect vector field.

One approach to the problem is to assess how well the vector signal matches the motion for each small region or pixel in the image, and switch to a non-motion-compensated method when the vectors become unreliable. However, the derivation of a reliable control signal is not a trivial matter, particularly in the presence of coding noise.

6.1 Deriving information from the present field

The philosophy that has been shown to work successfully for fixed vertical-temporal filters[2] was therefore considered; namely, ensuring that all low-frequency vertical information is derived from the present field. With this approach, incorrect vectors can never lead to degradation of low vertical frequencies, and significant blurring or combing is prevented. There is a price to pay in terms of vertical resolution and aliasing, since high vertical frequencies in the source signal can never be properly recovered, and will always manifest themselves as ‘twitter’ or vertical aliasing. However, real signal sources rarely generate much energy at such frequencies, and progressive displays cannot in any case represent high vertical frequencies accurately due to the absence of a proper vertical post-filter at the display, giving rise to the ‘Kell’ factor.

6.2 Filter designs

A low-pass intra-field vertical filter was designed with a cut-off frequency of 36 cycles per active picture height (a quarter of the vertical resolution supportable in one field). The filter had a vertical aperture of seven field lines, and was designed to represent a filter of a practical size that would provide protection for the lowest vertical frequencies without unduly compromising the performance of the interlace-to-progressive converters. Conversely, the fixed vertical-temporal filter that was tested had a cut-off frequency of around 100 c/aph for temporal frequencies of 25 Hz. It therefore provided better vertical resolution in moving areas but with poorer resolution and more aliasing in stationary areas.

The low-pass intra-field vertical filter was used to control which part of the vertical frequency spectrum was taken from a motion-compensated filter, and which part came from the simple ‘1/2 1/2’ intra-field vertical filter, using the arrangement shown in Fig. 7. This arrangement can be applied to any of the motion-compensated filters studied here; when applied, the system can be said to be ‘protected’.

The motion-compensated filter can be modified to include the action of the vertical filters in Fig. 7, rather than implementing the filters separately. For example, when this ‘protection’ strategy was applied to the motion-compensated temporal low-pass filter, its response for a motion speed of 0.5 picture lines per field period became that shown in Fig. 9, which can be compared to the response of the ‘unprotected’ filter, shown in Fig. 8. The contour lines correspond to responses of –1, –3, –6 and –20 dB. The ‘protected’ filter had unity gain for all temporal frequencies on the zero vertical frequency axis, and zero gain for all temporal frequencies at 288 c/aph. The low-level response in the upper left and lower right quadrants was due to the relatively slow roll-off of the vertical filter used when skewing the temporal filter by

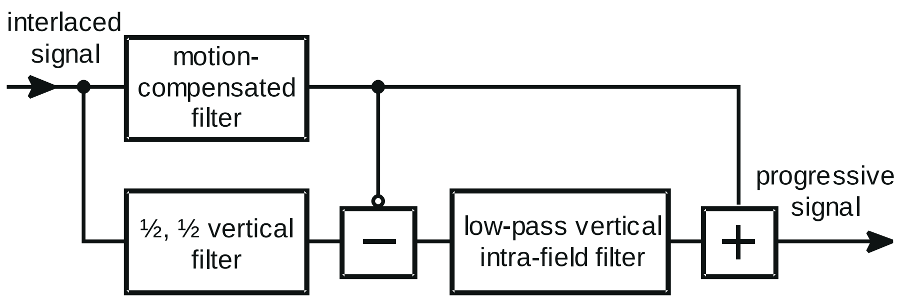


Fig. 7 - Use of a low-pass filter to ensure all low vertical frequencies are derived from an intra-field filter.

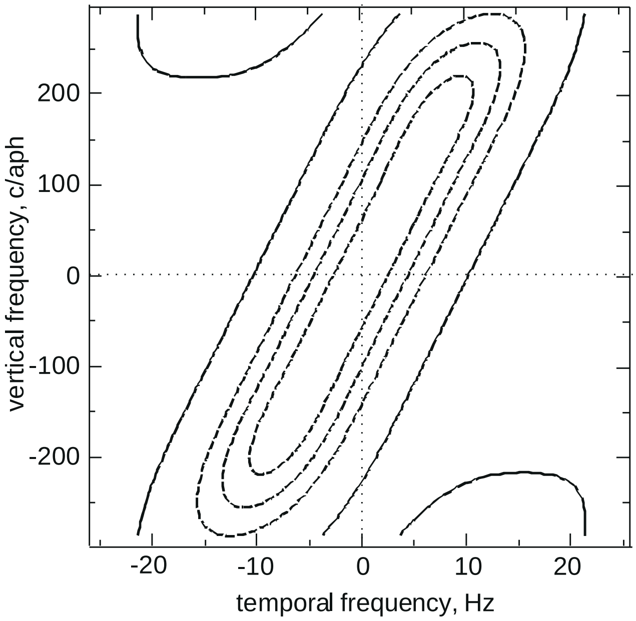


Fig. 8 - Response of the temporal low-pass filter, motion-compensated for a velocity of 0.5 picture lines per field period.

fractional amounts of a picture line. It can be thought of as the residual response of this filter above 288 c/aph, aliased down into the baseband; this can be most clearly appreciated from Fig. 8. For comparison, the response of the fixed vertical-temporal filter (which takes a larger proportion of low vertical frequencies from the current field) is shown in Fig. 10.

6.3 Performance of 'protected' filters with correct vectors

The plots in Fig. 11 *(overleaf)* show the slight reduction in performance that occurred in situations in which the vectors were correct, when using the low-pass intra-field filter arrangement described above with the motion-compensated temporal low-pass filter (similar results were obtained when applying this protection to the filters based on the 'general sampling theorem'). For comparison, the performance of the '1/2 1/2' fixed vertical filter and three-field vertical-temporal filter with motion compensation (to the nearest even number of picture lines per field period) is also shown. The reduction in performance of the motion-compensated temporal filter was typically around one grey level RMS. Subjectively, this manifested itself as a small degree of 25 Hz flickering in detailed areas, and more obvious vertical aliasing as the image moved. However, there was still a worthwhile improvement over the vertical-temporal filter with motion compensation (which is an example of a simpler filter that is robust in the presence of incorrect vectors). An experiment was also conducted using the test sequence that included camera integration; it was found that for speeds over one picture line per field period, the performance was reduced by a much lower figure (less than 0.5 grey levels). This was to be ex-

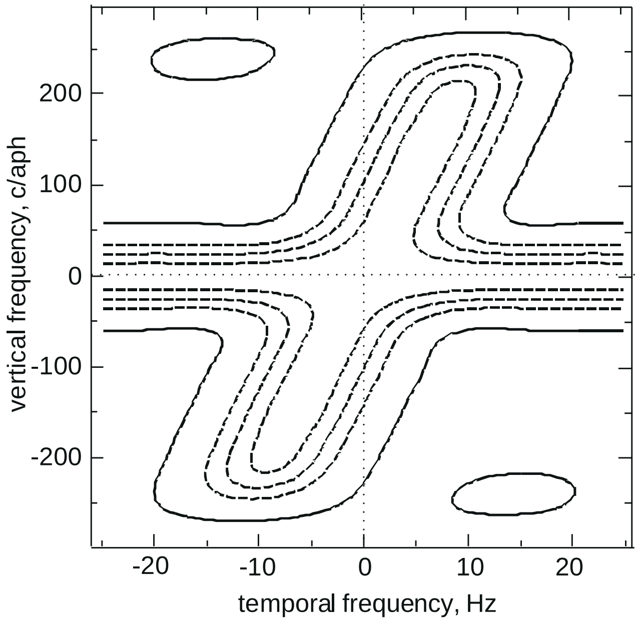


Fig. 9 - Response of the temporal low-pass filter, motion-compensated for a velocity of 0.5 picture lines per field period, modified to take low vertical frequencies from the

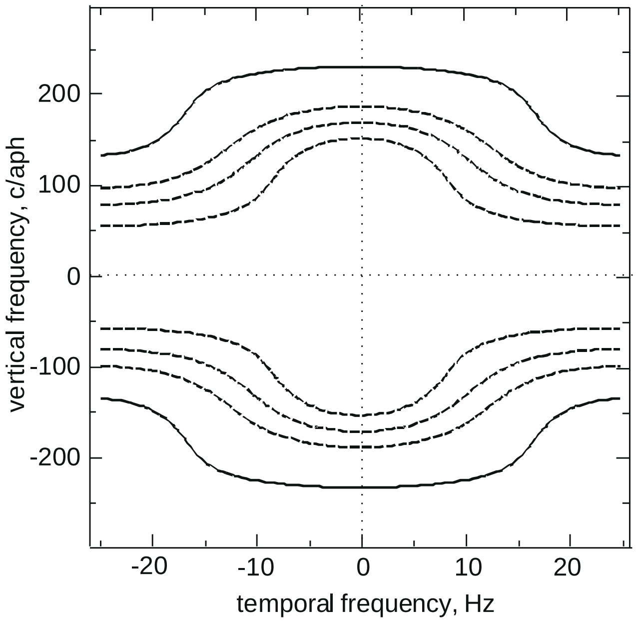


Fig. 10 - Response of the 3-field fixed vertical-temporal filter.

pected, as at these speeds the vertical resolution of the sequence is reduced, leaving little energy from high vertical frequencies for the 'protection' system to misinterpret as high temporal frequencies.

6.4 Performance of 'protected' filters with incorrect vectors

An experiment was conducted to test the performance of this protection method in the presence of incorrect vectors. The sequence used earlier, containing accelerating horizontal motion in the range of 0-2 pixels per field period, was processed assuming a fixed motion

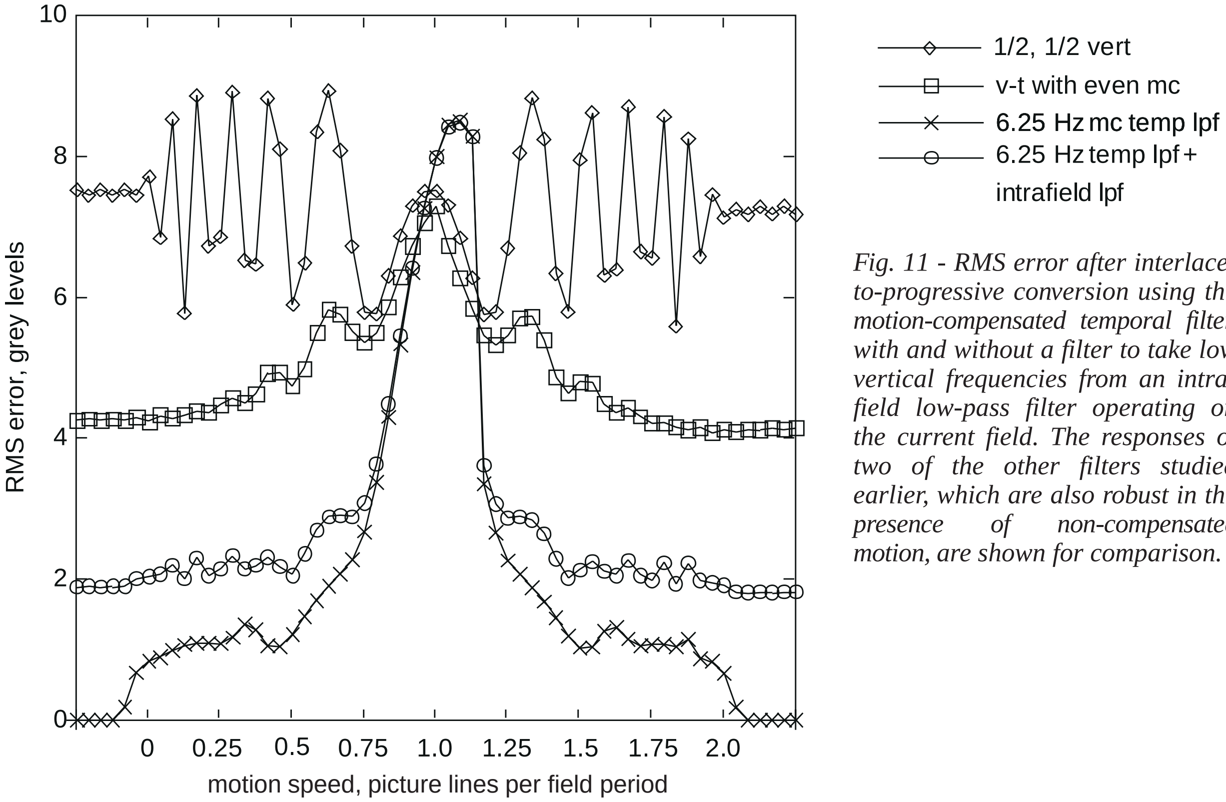


Fig. 11 - RMS error after interlace-to-progressive conversion using the motion-compensated temporal filter, with and without a filter to take low vertical frequencies from an intra-field low-pass filter operating on the current field. The responses of two of the other filters studied earlier, which are also robust in the presence of non-compensated motion, are shown for comparison.

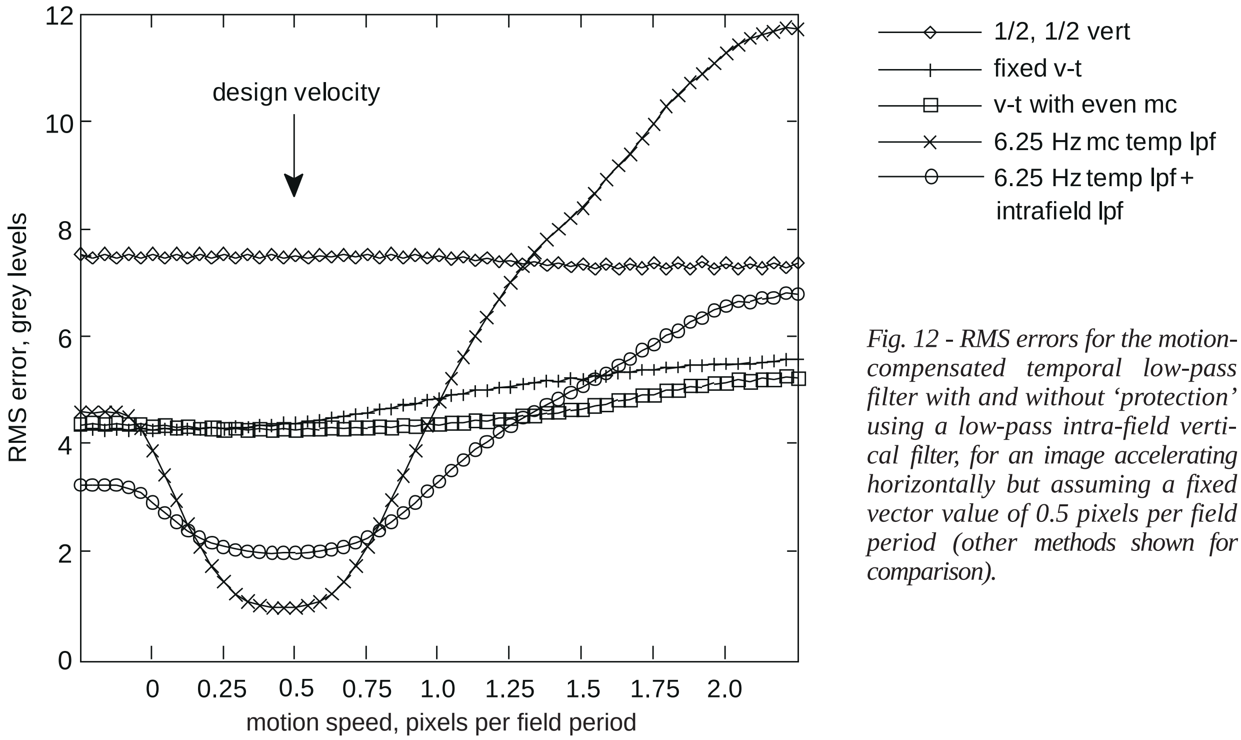


Fig. 12 - RMS errors for the motion-compensated temporal low-pass filter with and without ‘protection’ using a low-pass intra-field vertical filter, for an image accelerating horizontally but assuming a fixed vector value of 0.5 pixels per field period (other methods shown for comparison).

speed of 0.5 pixels per field period using the motion-compensated temporal filter with ‘protection’. The performance of this filter, together with that of the ‘unprotected’ version and some of the other filters studied, is shown in Fig. 12. The protection method gave an improvement when the vector was in error by more than about 0.25 pixels per field period, and succeeded in keeping the error below that from a simple vertical interpolator, even for image velocities up to 1.5 pixels per field period above the design speed of the filter (the maximum speed that occurs in this test sequence). The down-side, as shown previously, was a reduction in performance when the vector was correct.

7. TESTS WITH REAL IMAGE SEQUENCES & VECTORS RECOVERED FROM AN *MPEG-2* SIGNAL

Following the experiments with synthetically-moving sequences, the performance of the algorithms was examined using real image sequences. In order to allow an objective measure of performance to be derived, experiments were conducted using two progressive test sequences, which were converted to interlaced form after application of the well-known ‘HHI’ pre-filter, as used for example in Ref. 5.

7.1 Obtaining motion vectors

Vectors were derived by the phase correlation method, followed by a tracing and refinement process[7]; this approach has been shown to yield vectors that correspond closely with the true motion in the scene. These vectors were then used for coding the sequences into MPEG-2. Inter-field pixel-rate vectors were derived from the vectors used for coding, by dividing the displacement vectors by the number of field periods between the reference field and the field being predicted. Vectors in the corresponding blocks in the adjacent fields were examined to see whether the temporal distance of either of these vectors was lower, and if so, the vector with the lowest temporal distance was used instead. This helped to reduce errors in the recovered vectors due to acceleration or the revealing or obscuring of picture material between the reference and predicted image. This was particularly effective for ‘P’ frames, where the time period between reference and predicted images was the longest (three frames).

A four-point bilinear interpolator was used to generate a vector for every pixel from the vectors of the four nearest macroblocks. Whilst this produced a smooth vector field and worked well with the inherently uniform vector fields derived by phase correlation and tracing[7], artefacts could be produced if ever the original vector field was highly non-uniform (for example, if an exhaustive-search block match was used in the MPEG-2 encoder). Alternative approaches such as ‘block erosion’[8] may offer better subjective performance in such situations.

7.2 Comparing various interlace-to-progressive converters

The generated vectors were then used to control the various interlace-to-progressive converters, and the converted sequences were compared to the pre-filtered progressive original in order to compute the RMS error. The *gst two-field* filter was not tested, since the three-field version was expected to give a better indication of the performance obtainable from this class of filter. Note that the original (rather than the decoded) interlaced sequences were processed, so that artefacts from the interlace-to-progressive conversion could be separated from those due to coding.

Fig. 13 shows the RMS errors calculated for the two sequences used, *football* and *bbcdisc*. The relative subjective appearance of the converted sequences correlated reasonably well with these objective measurements. Examples of portions of processed images from some of the algorithms (128 pixels by 128 lines) are shown in Figs. 14(a) to (f) *(overleaf)*.

The performance of the fixed vertical-temporal filter (Fig. 14(b)) was relatively poor on these sequences (in terms of vertical resolution and aliasing), since virtually all parts of the sequences were moving.

The two ‘fully’ motion-compensated algorithms tested (‘general sampling theorem’ three-field, Fig. 14(d), and the seven-field 6.25 Hz temporal low-pass filter, Fig. 14(e)) both performed even worse than the non-motion-compensated algorithms, due to inaccuracies and errors in the recovered motion vectors. Some of these errors were present in the original pixel-rate vector field before conversion to GOP-based macroblock form; but others came from the block-based nature of the MPEG-2 vectors, and the vertical vector quantisation. Revealed and obscured background contributed further to the errors. The temporal filter tended to produce low-amplitude blurring where the vectors were wrong, whereas the *gst three-field* filter produced higher-amplitude errors but over a smaller

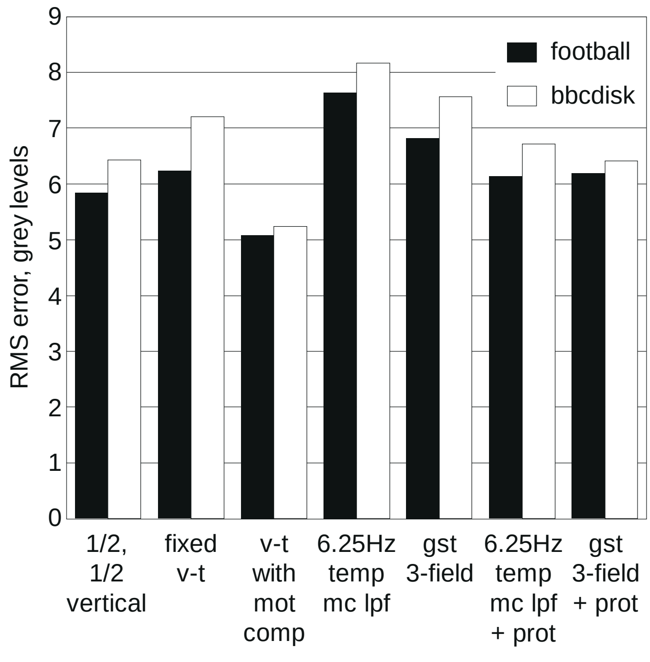
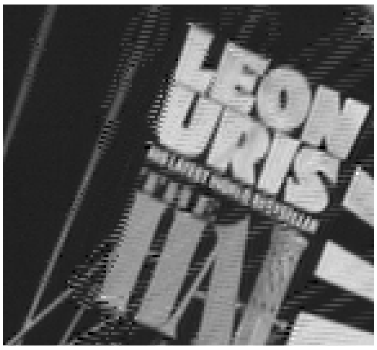


Fig. 13 - RMS errors from various interlace-to-progressive conversion methods, using vectors recovered from an MPEG-2 bitstream.

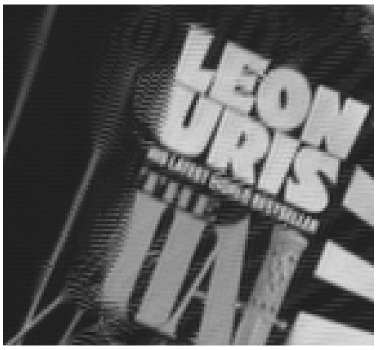
(a) - Image interpolated using simple vertical filter.

(b) - Image interpolated using fixed v-t filter.

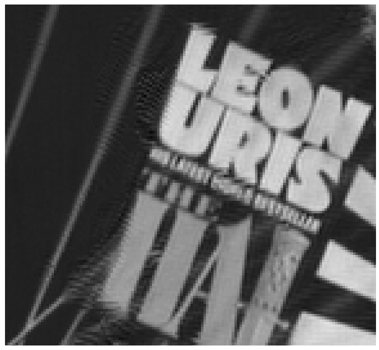
(c) - Image interpolated using v-t filter with motion comp to nearest even number of pic. lines.



(d) - Image interpolated using 'general sampling theorem' 3-field mot comp filter.



(e) - Image interpolated using motion compensated temporal filter.



(f) - Image interplated using motion-compensated temporal filter with protection.

Fig. 14 - Portions of progressive images from 'bbcdisc'.

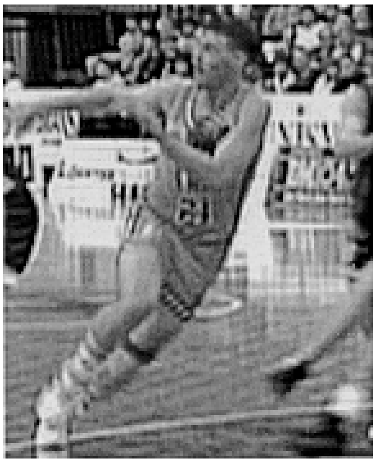
area, reflecting the shorter temporal extent of the filter. There were a few areas in both test sequences where the 'fully' motion-compensated algorithms gave improved vertical resolution or reduced aliasing compared to the fixed filters, but the effects were small compared to the level of artefacts. The improvements were not as widespread as might have been expected, suggesting that the accuracy of the recovered vectors, particularly in the vertical direction, was insufficient in many areas.

The use of the 'protection' system, discussed previously in Section 6.2, significantly improved the performance of the two 'fully' motion-compensated algorithms, both subjectively and objectively. Almost all motion artefacts were significantly reduced, with remaining errors taking the form of low-amplitude combing. An example of some artefacts in the *bbcdisc* sequence can be seen in Fig. 14(e), generated using the seven-field temporal interlace-to-progressive filter without protection. The same portion processed with the 'protected' form of the filter is shown in Fig. 14(f). In the absence of protection, artefacts from incorrect vectors in the region where the spokes of the bicycle wheel (moving up and to the left) cross over the book cover mounted on the rotating disc (moving down and to the right) are clearly visible. The curved nature of the artefacts in the spokes is due to the effect of the bilinear interpolator smoothing the vectors between macroblocks. The protected version of the interpolator shows artefacts at a much lower level.

The best performance, both subjectively and objectively, came from the vertical-temporal filter with motion compensation to an even number of picture lines per field period, Fig. 14(c). This filter was inherently more robust to vector errors than the other motion-compensated filters with protection, since it had a vertical passband for high temporal frequencies about three times that of the 36 c/aph filter used in the protection system. Picture resolution was much improved in many moving areas compared to both the simple vertical and the fixed vertical-temporal filters (see, for example, the spoke at the bottom left-hand corner of the picture, just left of centre).

Experiments were also conducted with a number of interlaced test sequences, which confirmed the results described above. An example of a portion of one of these sequences (*basketball*), processed with the '1/2, 1/2' vertical filter, and the fixed and motion-compensated vertical-temporal filters, is shown in Fig. 15(a) to (c). The poor vertical resolution and aliasing produced by simple vertical interpolation can be seen in Fig. 15(a), particularly in the lines on the floor and the player's shorts. The fixed vertical-temporal filter, Fig. 15(b) produced significantly better results for the lines on the floor, which are moving slowly, but per-

(a) - Field interpolated using ‘1/2, 1/2’ vertical filter.



(b) - Field interpolated using fixed vertical-temporal filter.

(c) - Field interpolated using vertical-temporal filter with even motion cempensation.

Fig. 15 - Portions of progressive images from ‘basketball’.

formed poorly on the player, who is moving more rapidly. Indeed, on the upper part of the player’s vest, it appeared slightly worse than the simple vertical filter. The vertical-temporal filter with even motion compensation, Fig. 15(c), produced good results on both the player and the floor, without introducing any artefacts, even in the revealed and obscured background areas around the player’s rapidly-moving arms and feet (despite the absence of any explicit detection or special processing for such areas).

8. CONCLUSION

This Report has considered various methods for performing interlace-to-progressive conversion. The basic objective has been to identify the best method to use when the only motion vector signal available is one which has been recovered from an MPEG-2 bitstream; this is likely to have limited resolution, both spatially and in terms of vector component magnitude.

It was found that by using the vectors recovered from the MPEG-2 bitstream, to motion-compensate a vertical-temporal filter in horizontal steps of a quarter of a pixel, and vertical steps of an even number of picture lines per field period, a worthwhile improvement in picture quality over fixed vertical and vertical-temporal filters could be obtained. Using the vectors to steer more sophisticated filters, which attempted to maintain good vertical resolution for all vertical speeds (except those close to the strobe speed), gave *worse* results than those obtained from the steering of the simpler vertical-temporal filter. This was because the filters generated objectionable artefacts in the presence of grossly incorrect vectors, since they placed much more reliance on information derived from adjacent fields. Even when the vectors were approximately correct, the accuracy of the recovered vertical vector component was not always good enough to allow high levels of resolution to be obtained. The vertical-temporal filter with even motion compensation is also simpler to implement, as it requires fewer coefficients than the more sophisticated filters; also the coefficient values remain fixed rather than varying as a function of vertical motion speed.

In order to minimise the appearance of artefacts caused by grossly incorrect vectors being used by the more sophisticated filters, a ‘protection’ scheme was proposed. This scheme ensured that low vertical frequencies, subjectively the most important, always came from the current field and were therefore not influenced by the motion vectors at all. It is possible that the use of such a scheme, coupled with a means of improving the accuracy of the vertical vector component in situations where it was slightly inaccurate, might significantly improve the performance of the more sophisticated filters in this application. For example, a small-range block-matcher might be used to ‘refine’ the recovered vectors vertically. This would be an interesting subject for future work. There is also scope for optimising the response of the low-pass vertical filter used in the ‘protection’ scheme. The cut-off frequency of 36 c/aph used here was too low, given the relatively poor quality of the vectors retrieved from the MPEG-2 bitstream. This value may be more appropriate for better quality vectors that have not had to pass through the ‘bottleneck’ of MPEG-2. There is also scope for optimising the response of the vertical-temporal filter, which was originally designed assuming no motion compensation. Its vertical response at 25 Hz could probably be reduced, and consequently increased at 0 Hz, to give better vertical solution at the expense of placing greater reliance on information from adjacent fields.

Of the more sophisticated filters examined, the two-field filter based on the 'general sampling theorem' was found to be the least tolerant to small inaccuracies in the estimated vectors, and the least tolerant to noise in the image. The three-field version of this filter performed much better in both these respects. The seven-field low-pass temporal filter that was tested proved to be the most tolerant to noise. When accurate vectors were available, all three of these filters proved to be capable of providing a worthwhile improvement in image quality, compared to the other simpler filters tested.

8.1 Acknowledgements

The work described here was supported by the European Commission as a part of the *RACE 'COUGAR'* project.

9. REFERENCES

1. BOCK, A.M., 1994. Motion-adaptive standards conversion between formats of similar field rates. Signal Processing: Image Communication, **6**(3), June, pp. 275-280.

2. WESTON, M., 1988. Fixed, adaptive and motion compensated interpolation of interlaced TV pictures. 2nd International Workshop on signal processing for HDTV, 29 February–2 March, (L'aquila, Italy).

3. DOYLE, T. *and* FRENCKEN, P., 1986. Median filtering of television images. ICCE Digest of Technical Papers, pp. 186-187.

4. SALONEN, J. *and* KALLI, S., 1993. Edge adaptive interpolation for scanning rate conversions, in: E. Dubois and L. Chiariglione, eds., Signal Processing of HDTV, **IV**, Elsevier, Amsterdam, pp. 757-764.

5. VANDENDORPE, L. *et al*., 1994. Motion compensated conversion from interlaced to progressive formats, Signal Processing: Image Communication, **6**(3), June, pp. 193-211.

6. BBC EUROPEAN PATENT, 1994.
   Interpolating lines of video signals.
   Inventor: WESTON, M.
   European Patent No. 0266079.

7. THOMAS, G.A. *and* DANCER, S.J., 1995. Improved motion estimation for MPEG coding within the RACE 'COUGAR' project, IBC '95, IEE Conference Publication No. 413, pp. 238-243.

8. De HAAN, G. *and* HUIJGEN, H., 1991. Motion estimator for TV-picture enhancement. Proceedings of the Fourth International Workshop on HDTV and beyond, **1**, 4–6 September. Turin, Italy.

Published by BBC RESEARCH & DEVELOPMENT DEPARTMENT, Kingswood Warren, Tadworth, Surrey, KT20 6NP