

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
10 May 2007 (10.05.2007)

PCT

(10) International Publication Number
WO 2007/051985 A1

(51) International Patent Classification:
H04N 5/14 (2006.01) *H04N 7/26* (2006.01)

(21) International Application Number:
PCT/GB2006/004016

(22) International Filing Date: 27 October 2006 (27.10.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0522169.2 31 October 2005 (31.10.2005) GB

(71) Applicant (for all designated States except US): **SONY UNITED KINGDOM LIMITED** [GB/GB]; The Heights, Brooklands, Weybridge, Surrey KT13 0XW (GB).

(72) Inventor; and

(75) Inventor/Applicant (for US only): **LIVING, Jonathan** [GB/GB]; 3 Woodstock Close, Wordsley, Nr. Stourbridge, West Midlands DY8 5HZ (GB).

(74) Agents: **TURNER, James, Arthur** et al.; D Young & Co, 120 Holborn, London EC1N 2DY (GB).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM,

AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

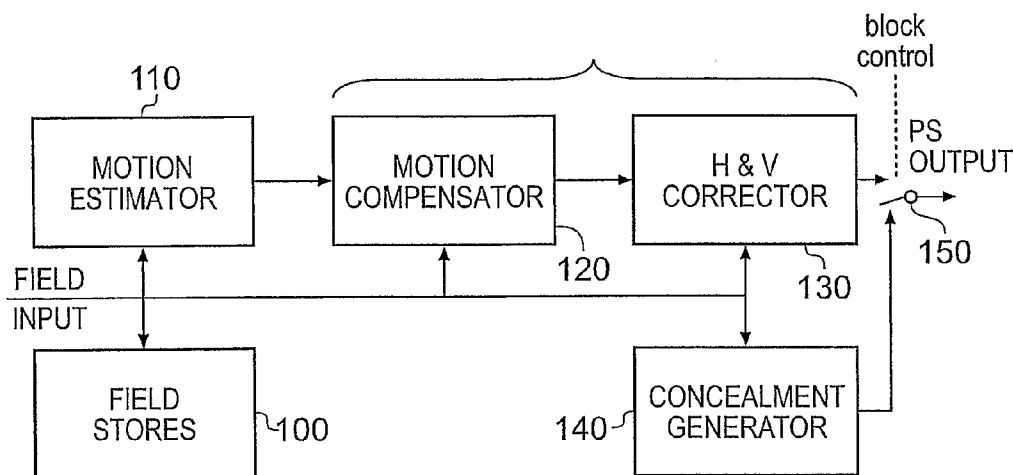
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

- with international search report
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: VIDEO MOTION DETECTION



(57) Abstract: Video motion detection apparatus comprises means for applying a motion test to detect inter-image motion between two or more test images, the motion test providing a test result providing a motion vector and an indication of the integrity of the detected motion vector; and means for applying the motion test to one of the test images, to detect intra-image similarity within that test image; in which the motion vector generated by the inter-image motion test is made available as an output of the apparatus if its associated test result indicates a greater integrity than the test result obtained by the intra-image similarity test; and the motion vector generated by the inter-image motion test is not made available as an output of the apparatus if its associated test result does not indicate a greater integrity than the test result obtained by the intra-image similarity test.

WO 2007/051985 A1

VIDEO MOTION DETECTION

This invention relates to video motion detection.

Video motion detection is used in video compression or other processing systems. In
5 a compression system, detecting inter-image motion allows a portion of an image to be
encoded with reference to a similar portion of another image. In other applications such as
standards conversion, definition conversion (e.g. standard definition to high definition or vice
versa) or interlaced to progressive scan conversion, a portion of a newly generated image is
derived or interpolated from one or more other temporally nearby images. In such
10 arrangements, the motion detection allows a suitable area of the nearby images to be used in
the interpolation process.

Image motion is often expressed in terms of motion “vectors”, on an image area by
image area basis. Commonly, motion vectors are detected in respect of image blocks, and a
motion vector is assigned either to a whole block or to pixels or groups of pixels within such a
15 block.

In compression systems, an incorrect motion vector would mean that an image area is
harder to encode – i.e. it would require more data to encode it because it would be relatively
dissimilar to the portion of another image referenced by the motion vector.

However, in the other systems based on generating or interpolating a new image from
20 one or more nearby images, the situation is even worse. An incorrect motion vector would
mean that the wrong part of a nearby image was used as the basis for interpolation. This
could lead to highly visible artefacts being generated in the output (new) image.

There is therefore a constant need to obtain high quality and accurate motion
information in video motion detection arrangements.

25 This invention provides video motion detection apparatus comprising:

means for applying a motion test to detect inter-image motion between two or more
test images, the motion test providing a test result providing a motion vector and an indication
of the integrity of the detected motion vector; and

means for applying the motion test to one of the test images, to detect intra-image
30 similarity within that test image;

in which the motion vector generated by the inter-image motion test is made available
as an output of the apparatus if its associated test result indicates a greater integrity than the
test result obtained by the intra-image similarity test; and

the motion vector generated by the inter-image motion test is not made available as an output of the apparatus if its associated test result does not indicate a greater integrity than the test result obtained by the intra-image similarity test.

In the present invention, a motion detection test is applied within one of the test
5 images itself. If a motion detection result within that one image is better than a motion detection result between the two images, then the inter-image result is considered to be of poor quality and is discarded.

Further respective aspects and features of the invention are defined in the appended claims.

10 Embodiments of the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1 schematically illustrates a flat-screen display arrangement;

Figure 2 schematically illustrates video mixing operation in a studio environment;

Figure 3 schematically illustrates an interlaced to progressive scan converter;

15 Figures 4a and 4b schematically illustrate "normal" and generalised sampling theory (GST);

Figure 5 schematically illustrates a part of a conversion process using sub-pixel positional correction;

Figure 6 schematically illustrates sub-pixel errors;

20 Figure 7a schematically illustrates horizontal sub-pixel correction;

Figure 7b schematically illustrates vertical sub-pixel correction;

Figures 8a to 8c schematically illustrate polyphase interpolation;

Figure 9 schematically illustrates a commutator;

Figure 10 shows an example image;

25 Figure 11 schematically illustrates edge detection using a Gx Sobel operator;

Figure 12 schematically illustrates edge detection using a Gy Sobel operator;

Figure 13 schematically illustrates a block match size map;

Figure 14 schematically illustrates a block match vector acceptance result;

Figure 15 schematically illustrates motion vector verification;

30 Figure 16 schematically illustrates vertical half-band filtering;

Figures 17a to 17c schematically illustrate aspects of GST filter design; and

Figures 18a to 18e schematically illustrate aspects of dealing with moving image objects.

Figure 1 schematically illustrates a flat screen display arrangement 10 comprising a source of interlaced video material 20, an interlace to progressive scan converter 30 and a display panel 40 such as a liquid crystal (LCD) or plasma display. This illustrates a typical use of interlace to progressive scan conversion, in that many broadcast signals are in the interlaced format whereas many flat panel displays operate most successfully in a progressive scan format. Accordingly, in Figure 1, a broadcast signal received by the source of interlaced material 20 is used to generate an interlaced signal for display. This is passed to the interlace to progressive scan converter 30 to generate a progressive scan signal from the interlaced signal. It is the progressive scan signal which is passed to the display 40.

It will be appreciated that the source of interlaced material 20 need not be a broadcast receiver, but could be a video replay apparatus such as a DVD player, a network connection such as an internet connection and so on.

Figure 2 schematically illustrates a video mixing operation in a studio environment, in order to give another example of the use of interlace to progressive scan conversion. Here, a source of interlace material 50 and source of progressive scan material 60 are provided. These sources could be cameras, video replay apparatus such as video tape recorders or hard disk recorders, broadcast receivers or the like.

The interlaced output from the source of interlaced material 50 is supplied to an interlace to progress scan converter 70 to generate a progressive scan signal. This can be processed by the vision mixer 80 along with the progressive scan material from the source 60 to generate a processed progressive scan output. Of course, the progressive scan output of the vision mixer 80 can be converted back to an interlaced format if required, e.g. for subsequent broadcast or recording. It will also be appreciated that the vision mixer 80 is just one example of video processing apparatus; instead, a digital video effects unit, for example, could be used at this position in Figure 2.

Figure 3 schematically illustrates an interlaced to progressive scan converter which receives a field-based input signal and generates a progressive scan frame-based output signal. In the present embodiment the output signal has one frame for each field of the input signal.

The converter of Figure 3 comprises one or more field stores 100, a motion estimator 110, a motion compensator 120, a horizontal and vertical positional corrector 130, a concealment generator 140 and an output selector 150. The motion compensator 120 and the positional corrector 130 are shown as separate items for clarity of the description; in reality, it is likely that both of these functions would be carried out as part of the same operation.

An input field is stored in the field store(s) 100 and is also passed to the motion estimator 110. Using block-based motion estimation techniques to be described below, and with reference to the field store(s) 100, the motion estimator 110 derives motion vectors indicative of image motion between the current field and another field (e.g. the preceding field). The motion vectors are derived to sub-pixel accuracy.

The motion compensator 120 is used to generate “missing” pixels to augment the pixels of the current field, in order to generate an output frame. So, the pixels of the current field are retained, and the empty lines between those pixels are populated with pixels from the stored field(s) using motion compensation. The operation of the motion compensator 120 will be described in more detail below.

The horizontal and vertical positional corrector is employed because the output of the motion compensator, while correct to the nearest pixel, is normally not exactly aligned with the sampling points (pixel positions) in the output frame. This is because motion estimation is performed to sub-pixel resolution.

Horizontal positional errors are corrected using polyphase filtering. Vertical positional errors are corrected using a filter employing a special case of the so-called Generalised Sampling Theorem. These operations will be described in more detail below.

The concealment generator 140 is arranged to provide a pixel value in case the motion dependent compensation arrangement fails to do so. It might be needed in the case of a failure to complete the processing needed to derive correct motion vectors in respect of each pixel, for example because the nature of the images made deriving motion vectors inaccurate or processor-intensive. In actual fact, the concealment generator is included within the functionality of the motion compensator / positional corrector, but is shown schematically in Figure 3 as a separate unit. Similarly, the selector 150 is part of the functionality of the motion compensator / positional corrector / concealment generator, but is shown separately to illustrate its operation. The selector 150 selects (on a block-by-block basis) a concealment pixel when a motion compensated pixel cannot be generated.

Figures 4a and 4b provide an overview of the generalised sampling theory (GST). In particular, Figure 4a schematically illustrates the “normal” sampling theory, whereas Figure 4b schematically illustrates the GST.

In Figure 4a, the familiar situation is illustrated whereby a signal having a maximum frequency of $f_s/2$ can be perfectly reconstructed by sampling at a rate of f_s , which is to say that sampling points occur regularly every $1/f_s$. This analysis is equally valid for a time-based

system or a spatially-based system, i.e. the sampling rate f_s can be expressed in samples per second or samples per spatial unit.

Figure 4b schematically illustrates an instance of the GST. According to the GST, it is not in fact necessary to sample with one fixed sampling period ($1/f_s$). Instead, a signal having
5 a maximum frequency of $f_s/2$ can be perfectly reconstructed if it is sampled by two sampling points every period of $2/f_s$.

Figure 5 schematically illustrates a part of the conversion process carried out by the apparatus of Figure 3, to illustrate the need for GST-based positional correction. Fields 0, 1 and 2 are evenly spaced in time. The intention is to create a progressively scanned frame,
10 frame 1, using existing pixels from field 1 and also motion compensated pixels (to fill in the missing lines) derived in this instance from fields 0 and 2 by a motion compensation technique using block based motion estimation. The missing pixels are inserted between the lines of pixels in field 1 to create frame 1. But the motion compensated pixels in frame 1 have sub-pixel positional errors. Note that in other embodiments the missing pixels are
15 derived from one field only.

As mentioned above, the sub-pixel positional errors are corrected by two techniques. Horizontal sub-pixel errors are corrected using polyphase filtering. Vertical errors are corrected using GST filtering.

Figure 6 schematically illustrates these sub-pixel errors. White circles 170 indicate the
20 required positions of motion compensated pixels to fill in the missing lines of field 1 to produce frame 1. Grey pixels 180 indicate the positions of real pixels from field 1. Dark pixels 190 indicate the positions of the motion compensated pixels in this example. It can be seen that the motion compensated pixels 190 are close to, but not exactly aligned with, the required positions 170.

Figure 7a schematically illustrates the use of a polyphase filter to correct the
25 horizontal position. The technique of polyphase filtering will be described in more detail below, but in general terms a filter 200 receives a group of motion compensated pixel values as inputs. The filter comprises P sets of filter taps h , each of which sets is arranged to generate an output value at a different phase (in the case of pixels, horizontal position) with
30 respect to the input motion compensated pixels. The phases are indicated schematically (210) in Figure 7a as running from 0 (in this example, phase 0 is aligned with a left-hand real pixel) to P-1 (in this example, phase P-1 is aligned with a right hand real pixel). In other words, the horizontal positional error is quantised to a sub-pixel accuracy of $1/P$ pixel spacings.

A schematic commutator 220 selects the correct set of taps to generate a new pixel value 190' which is horizontally aligned with a real pixel 170.

Figure 7b schematically illustrates the use of the GST to correct the vertical position. Here, the pixels 190' are shown having had their horizontal position corrected as described
5 above.

In the vertical direction, in each spatial period of two (frame) lines, two pixels are provided: a real pixel 180 from field 1, and a horizontally-corrected pixel 190'. The presence of two valid sampling points in a two-line spatial period means that the "original" value of each respective pixel 170 can be recovered by a vertical filtering process. A group of
10 properly vertically-aligned pixels 230 suffers little or no aliasing. On the contrary, a group of incorrectly vertically aligned pixels 240 suffers with vertical aliasing.

The equation of a suitable GST filter is as follows:

$$x_c(n) \sim \sum_{k=-L/2}^{+L/2} x_c(kN+n_0) \frac{1}{\sin(p(n-n_0)/N)} (-1)^{2k} \prod_{q=0}^1 \frac{\sin(p(n-n_q)/N)}{(n-kN-n_0)} \\ + \sum_{k=-L/2}^{+L/2} x_c(kN+n_1) \frac{1}{\sin(p(n-n_0)/N)} (-1)^{2k} \prod_{q=0}^1 \frac{\sin(p(n-n_q)/N)}{(n-kN-n_1)} \quad 15$$

where 2 sub-sampled data sequences form sample sets $nN + n_p$ ($p=0 \dots (N-1)$), N is the maximum number of discrete equally spaced samples per Nyquist period and n is a sample number.

In summary, therefore, the GST can be used to reconstruct a quasi-perfect progressive frame from two or more interlaced fields. The process involves the copying of pixels from one field and positionally restoring the remaining pixels (obtained from the other field) in the progressive scan frame.
20

Subsequently, horizontal phase correction and vertical GST reconstruction yields pixel
25 values that complete a quasi-perfect progressive scan frame.

However, to restore the position and phase of pixels from the second field, motion vectors accurate to some fraction of the spatial sampling resolution must be known. Accordingly, there now follows a description of the operation of the motion estimator 110.

Motion estimation in general aims to detect the magnitude and direction of real vectors
30 using some local minimisation of error between an image and spatially shifted versions of it. However, if image data is sub-sampled (as is the case for fields of an interlaced source), there may be little or even zero correspondence between versions with different displacements, inhibiting the detection of motion this way.

Several motion estimation methods are known. These include:

1. Gradient method: in its simplest form, this technique assumes a constant luminance gradient over a localised area to translate changes in pixel or small-block-average luminance into motion using a linear (straight-line) relationship.

2. Block based method: this method generally involves block matching between two or more successive frames of a video sequence to establish the correct displacement. The match criterion used is a minimum pixel difference measurement, usually the MSE (Mean Squared Error) between corresponding blocks.

3. Fourier-transform method: this technique is generally the same as block based methods, but uses the Fourier transform to calculate rotational convolution in two dimensions. This significantly reduces the computational effort required to compute block search results over a large area.

Block based methods are generic in operation (i.e. the outcome of a block based search should be the same as the outcome after applying the Fourier method) and yield a more accurate result mathematically than the gradient method supported by its associated assumptions.

The block match method is used in the present embodiment, but it will be appreciated that other methods could be used..

However, a known disadvantage of the block search method is calculation of the wrong motion vector by an incorrect MSE minimisation search. This can occur for at least three possible reasons:

1. Blocks chosen for the search lack sufficient detail to ensure any displacement yields an MSE larger than zero displacement.

2. The summation within the MSE calculation can be overloaded with pixel differences, causing larger errors to be reported for block displacements closer to the truth than other clearly incorrect displacements.

3. Blocks chosen for the search auto-correlate to produce a lower (intra-frame) MSE than that obtainable using true inter-frame vector displacement of the block.

These possible failings are addressed in the present embodiment using a specific technique for each one.

Referring now to Figures 8a to 8c, poly-phase interpolation is the method used to analyse sub-pixel motion between successive frames, imparted as a result of non-integral pixel shifts of the original source image caused by the process of generating an interlaced field. Poly-phase interpolation for a sub-block MSE search can be viewed as a computationally-efficient method of firstly inserting samples in a data sequence by applying

the original bandwidth constraint, and secondly selecting a regular set of samples with the desired sub-pixel shift.

A method of poly-phase interpolation can be derived from the schematic diagrams of Figures 8a to 8c. Figure 8a schematically illustrates an original discrete-time sampled signal. Figure 8b schematically illustrates the original signal of Figure 8a, zero-padded. In other words, zero-valued samples have (at least notionally) been inserted between “real” samples of the signal of Figure 8a. Figure 8c schematically illustrates the signal of Figure 8b, having been filtered to reapply the original bandwidth constraint (i.e. the bandwidth of the signal of Figure 8a).

Both the original signal and the filter are assumed to be discrete-time series sampled at instances $0 + nT$, where $n = 0, 1, 2$, etc. For the purposes of simplifying the present analysis, the substitution $T=1$ is made to normalise the sampling period.

The original signal referred to as $x(n)$ (rather than $x(nT)$, because $T=1$) is firstly zero-padded to reflect the interpolation ratio. For example, interpolation by a factor N requires insertion of $N-1$ zeros between original (real) samples to yield a sample sequence length N times the original.

Convolution of the zero-padded input sequence with a (length $L+1$) filter $h(n)$ applying the original bandwidth constraint (now N th-band) yields the sequence of results $y(n)$:

$$\begin{aligned} y(0) &= x(0)h(0); \\ y(1) &= x(1)h(0); \\ &\vdots \\ y(N-1) &= x(N-1)h(0) \dots \\ y(N) &= x(N)h(0) + x(0)h(N) + \dots \\ &\vdots \end{aligned}$$

Clearly, $y(0)$, $y(N)$, $y(2N)$, etc results are computed as a convolution of $x(n)$ with filter coefficients $h(0)$, $h(N)$, $h(2N)$, etc. Similarly, $y(1)$, $y(N+1)$, $y(2N+1)$, etc are computed by convolution with filter coefficients $h(1)$, $h(N+1)$, $h(2N+1)$, etc. These short-form computations can be neatly expressed in the form of a schematic commutator selecting between coefficient sets P as shown in Figure 9.

The commutator selects the sub-pixel phase required. An efficiency derives from this operation, as only the multiplications and additions required to provide that particular result

need to be computed. Generally, a gain factor of N is applied at the output as the zero-padded original sample sequence is considered to have $1/N$ th the energy of the original.

The poly-phase computation is used both vertically and horizontally in the block-matching algorithm. Accordingly, the motion vectors are generated with sub-pixel resolution.

5 The maximum search range in pixels (i.e. the maximum permitted tested displacement between a block in one field and a block in another field, which may be an absolute or may be a multiple of the test block size) is translated into the number of sub-pixels this represents. For any given offset from zero the required phase is the modulus of this shift measured in sub-pixels divided by the interpolation ratio. The absolute displacement in pixels is the
10 integer division of this shift by the interpolation ratio.

A method of variable block size selection is used for robust frame-based motion estimation. Each block is allocated a minimum and maximum power-of-two size in the horizontal (upper-case X) and vertical (upper-case Y) directions.

To begin, the sizes of all blocks are set to a predetermined maximum power of two
15 (for example 5, giving a maximum block size of 2^5 pixels) but with the frame's outer dimensions as a constraint such that block sizes can be reduced in X and/or Y from the outset to ensure edge fitting.

An iterative process of division of each block into two halves either vertically or horizontally (the later takes precedence) is undertaken, based on edge content detected and
20 measured using the Sobel operator. The general principle is that a block is divided (subject to a minimum block size – see below) if it is found to contain more than a desired edge content.

The Sobel operator takes the form of and is applied as two separate two-dimensional 3×3 coefficient filters. The first, G_x , shown to the left below, detects vertical edges and the second, G_y , shown to the right below, detects horizontal edges.

25

-1	0	1
-2	0	2
-1	0	1

30

+1	+2	+1
0	0	0
-1	-2	-1

 G_x G_y

Due to the coefficient value range of G_x and G_y , these filters exhibit maximum gains of +4 and -4 when convolved with image data in the range 0 to 1. The results obtained from applying these filters are therefore first normalised to the range -1 to +1 through division by 4. (As an alternative, normalised coefficients could be used in the Sobel operators)

5 Figure 10 illustrates one image of a source video sequence against which some of the present techniques were applied. The source video sequence was actually generated artificially by starting from a $4096 * 1696$ pixel basic image. Whole-pixel shifts, simulating camera panning, were applied to impart motion to a sequence of such images. The final output fields were obtained by n th-band filtering and subsequent sub-sampling by the same
10 factor, where a value of $n = 8$ provided a finished size of $512 * 212$ pixels. So, each field in the source video sequence involved motion with respect to neighbouring fields and also represented a sub-sampled version of the basic image.

Taking the absolute values of each operator's (i.e. G_x 's and G_y 's) results in turn and accepting only absolute (normalised) values of 0.2 and above (i.e. applying a "greater-than"
15 threshold of 0.2), the application of G_x and G_y to the source image shown in Figure 10 produces the two edge-detection images shown in Figures 11 and 12. In particular, Figure 11 schematically illustrates edges detected using the G_x operator and Figure 12 schematically illustrates edges detected using the G_y operator. Pixels are therefore identified and flagged as "edge" pixels.

20 With regard to each pixel block proposed for use in block matching, the total count of detected edge pixels (of minimum normalised magnitude 0.2) is subject to further threshold testing to establish whether the block may be split. Each block is notionally sub-divided into four quarters (vertically and horizontally by two).

If each quarter contains both a horizontal and vertical edge pixel count greater or equal
25 to the number of pixels in the predetermined minimum (indivisible) block size, the block division is accepted. However, if only the horizontal count is deficient, block quarter boundaries are merged and vertical division by two is accepted. Finally, if only the vertical count is deficient, block quarter boundaries are merged and horizontal division by two is accepted.

30 If both counts are deficient the block is not divided, marking the stopping criterion in each case. When there are no more sub-divisions, the block-match mapping is complete. Applying this technique to the source image shown in Figure 10 with edge threshold results as shown in Figures 11 and 12 results in the block division pattern shown schematically in Figure 13.

To prevent or at least avoid the mean squared error calculation used to assess block similarity from returning erroneous minima, pixel difference limiting is employed to prevent saturation of the sum for block displacements within a small range around the ground truth. The standard MSE calculation used in block matching is shown in Equation 1.

$$MSE = \sqrt{\frac{\sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (A_{x,y} - B_{x+j,y+k})^2}{N * M}} \quad \text{Eq. 1}$$

In Equation 1 the block size is N*M pixels and is indexed as $A_{x,y}$ in one frame and $B_{x+j,y+k}$ in the next frame, where j and k are the whole-pixel horizontal and vertical displacements applied during the minimisation search. Of course, $B_{x+j,y+k}$ references the appropriate phase of image according to those derived using Figure 9 and the modulus of the actual displacement required (in sub-pixels) for this analysis.

The kernel difference calculation is replaced with one that limits the overall error per pixel, as shown in Equation 2.

$$MSE = \sqrt{\frac{\sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \min \{ (A_{x,y} - B_{x+j,y+k})^2, q \}}{N * M}} \quad \text{Eq. 2}$$

In Equation 2, q is an appropriate constant. For image data in the range 0...1, a value of $q = 10^{-2}$ has been found to work well.

The limiting of pixel differences in this way has been found to provide greater definition and distinction of the ground truth displacement on the complete two-dimensional error surface generated by the block search.

To prevent or at least reduce erroneous (or "rogue") vector generation by the block-search method, advanced warning of the potential for a rogue result to occur can be obtained by block intra-frame (auto) correlation.

To apply this technique, a block search is first performed within the required range in the same image. The minimum MSE is recorded. A block search in the second frame is then performed as before, however if the smallest MSE recorded is greater than the intra-frame MSE, a vector resolved from the search is discarded.

A maximum MSE criterion can also be applied using intra-frame correlation results. By only allowing displacements of at least one sub-pixel in X and Y, a measure of the worst permissible MSE for a confident match can be obtained. This then is the upper limit for any

inter-frame MSE when the resolved displacement (motion vector) should ideally be within one sub-pixel of the ground truth.

The validity of each vector according to the described method of rogue vector elimination is shown in Figure 14, where only those blocks which the algorithm confirms to provide true motion estimation results are given an outline.

Further techniques for applying motion vector estimation to the method of interlaced to progressive frame conversion using the generalised sampling theorem will now be described.

Robust frame-based motion vector estimation has already been described above. However, in the present embodiment it is the frame data that does not exist and is to be reconstructed using the GST. Frame data cannot be created without quasi-perfect motion vectors to restore pixel positions from one field into the other and detect the phase alignment for GST filter selection. Neither frame data nor perfect motion vectors can exist without the other but either is difficult to derive in the first place.

One option is field-based motion estimation. Unfortunately, field data is aliased due to the 2:1 sub-sampling in the conversion from a progressive format or due to the inherent workings of the capture device generating the source material for subsequent display using the interlaced format.

Sub-sampling affords no guarantee that a block of image data will match at all with any supposedly identical block in another image as the chosen representation may naturally exclude some or all of the features in one sample set that are apparent in the other. However, there is some likelihood that at least some data will be aliased in the same way, and an inter-field match with the correct displacement will be obtained.

With regard to the three improvements made to ensure robust frame-based motion estimation, not all of these are applicable to field-based estimation.

Firstly, field data may be the result of sampling in a way that excludes significant detail from one or more areas, whereas in reality (or in another instance of the field later in time) this detail is present. Using detail analysis for variable block size selection is therefore not relevant for field data.

However, modification of the MSE calculation kernel to prevent error sum overflow due to large pixel differences is valid for field data. The best case is fields that do not contain aliasing artefacts due to the nature of the original signal content; modification of the kernel calculation therefore enhances the ability of the search algorithm to discern the minimum error attributable to the real displacement vector.

The same can be said for the rogue vector avoidance technique. It is an addition to the block search algorithm and can only improve performance for fields without significant aliasing.

For significantly aliased fields, there are fundamental reasons why the block search
5 algorithm may fail as already discussed - retaining the MSE kernel modification or rogue vector elimination methods will not degrade performance further under these conditions.

The field-based motion estimation algorithm is described below, initially in terms of the replacement for block selection by detail analysis and subsequently by further enhancements that make the technique more successful in field-based systems.

10 In the GST motion estimation algorithm, block sizes used for field-based MSE searches are variable by power-of-two divisions in X and Y from some maximum initial dimensions. However, these divisions are controlled by an allowable pixel area, below which the block cannot shrink.

This method supports awkward image sizes not dimensioned to be a multiple of any
15 power of two in X or Y while ensuring a sufficient number of pixels are included in the block matching calculation to achieve the desired accuracy of correlation results (i.e. the MSE minimum is the ground truth displacement).

Starting values for block sizes are typically up to 2^6 in X and Y but with an overall initial minimum area value of 2048 pixels. Final block dimensions as small as 2^2 in X and Y
20 are supported with a minimum area of 2^5 pixels.

Motion estimation for the GST includes inter-field block searches for representative motion, intra-field searches for block similarity and inter-frame block displacement verification. Both stages of the algorithm are implemented to support variable block sizes, as will be discussed later.

25 Application of the sub-pixel motion vector search algorithm to field data generates a distribution of motion vectors around ground truth vectors, even with inclusion of the MSE kernel calculation modification and rogue vector removal technique. This is wholly due to aliasing and the lack of repeatability of image data between fields.

For example, a test sequence in which successive images were shifted in X and Y at a
30 rate of 9 and 3 sub-pixels ($1/8$ pixels in this example) per frame respectively generated the following distribution of motion vectors in Table 1.

Displacement in X for minimum MSE, in sub-pixels	Displacement in Y for minimum MSE, in sub-pixels	Number of blocks returning this vector
3	3	4
18	3	155
2	3	2
18	5	3
-6	2	1
2	-2	1
17	5	3
20	3	2
17	3	12
18	4	3
15	5	1
11	5	1
13	3	2
14	3	2
15	3	4
16	3	3
11	4	2
19	3	2
22	4	4
15	4	1
21	3	1

From Table 1, the most popular vector is X, Y = 18, 3 which is indeed correct. Field-based motion estimation is applied between fields of the same type (either even or odd) which means that effectively there is double the motion in X and Y between the frames these fields are generated from. However, by using only collections of field lines, this doubling is subsequently halved in Y and only X is actually reported as twice the actual motion. Hence, a displacement of 9 and 3 pixels in X and Y between frames used to build fields is detected as a displacement of 18 and 3 pixels.

In the example, synthetic panning-only motion ensures one dominant vector is detected. However, actual inter-field motion may be significantly more complex than this. In general, candidate motion vectors are sorted in order of the number of blocks that support them and one or more vectors in order of popularity can be chosen for further processing and verification.

Candidate motion vectors obtained by field search are verified to ensure (or at least increase the likelihood of) their validity. The method used in the present embodiment involves repeated reconstruction of frames from two consecutive (even followed by odd or vice-versa) fields using the GST.

The motion vectors used for reconstruction are those obtained from field-based motion estimation, sorted in order of popularity. Once two successive frames have been reconstructed, block-based matching is employed to verify each vector's correctness.

The block size used for matching is variable, and is based on the fixed-area criterion
5 as described for field block size selection previously.

It is useful to assume the motion being verified is constant across four fields. Vectors obtained from one field pair match can be combined with those from the next field pair match forming the first stage of the filtering process. For example, if a vector is not supported by at least one block from each field pair, it is discarded.

10 Figure 15 schematically illustrates the overall process of vector verification. Candidate motion vectors are generated between fields of the same type (even or odd) within the four-field sequence. Combination of these vector lists, sorting in order of popularity and threshold discarding of entries if they do not appear at least twice (for example, once between each field pair) all help to build a prioritised set of vectors that ensure the success of the GST
15 for frame reconstruction.

Once the GST reconstructs two frames using neighbouring fields of the same type, the field vector used for that instance is the one applied to blocks mapped in one frame when compared with the other.

The match criterion is an MSE better than any intra-frame (auto) correlation of the
20 block with a displacement greater than or equal to one sub-pixel. This can be considered to be a threshold relating to the energy and complexity of the video within the block being verified and implies that the motion vector being used by the GST must be correct to within one sub-pixel for the block match between frames to succeed.

This verification threshold works well for all but the least detailed blocks, where the
25 intra-frame error is small and artefacts caused by the GST calculation exceed it.

Blocks that verify motion vectors are committed to the final output frame result. The candidate motion vector list obtained from field analysis can then be referenced for the next most popular vector and the process repeated until the largest possible proportion of the output frame has been derived using the block sizes given by the minimum area constraints.

30 The acceptance criterion for motion vectors described above can tend to leave a proportion of the reconstructed frame blocks unverified. The MSE threshold set by auto (intra-frame) correlation is particularly stringent and tends to reject blocks if:

1. The source frame detail within the block area is particularly low, generating a very small auto-correlation MSE that cannot be bettered by inter-frame correlation no matter how good the GST reconstruction.

2. The source frame has complex motion (more than one representative vector) within the block area being analysed. No good block match will be obtained between frames due to revealed or covered pixels (though see the discussion of Figures 18a to 18e below).

3. As a special case of (2) above, blocks positioned at the edges of the frame suffer a loss of current pixels and gain of new pixels due to panning motion and do not match well with blocks in other frames.

10 All of these problems can be dealt with to some extent by block size reduction. In the case of (2) and (3) above, smaller blocks will better fit to a part of the frame whose motion can be described by a single vector, refining object and background areas up to, but not including, their outlines.

The minimum block areas for field-based motion estimation and frame-based motion verification are then reduced and the process described above is repeated. Minimum block areas as small as 16 pixels (X and Y dimensions of 4 pixels) are currently permitted in the present embodiment.

The philosophy behind large-to-small block area selection is as follows. In starting with the largest block area of around 2048 pixels, the most accurate field-based motion estimation and frame-based motion verification are obtained. Smaller blocks that may be more susceptible to MSE minima not representing ground truth displacement are dealt with subsequently, such that any small reconstruction errors are better concealed.

After each round of frame-based vector verification is complete, any resolved picture areas are excluded from the block selection for field-based candidate motion vector generation using smaller block areas as follows.

A mask of unresolved frame pixels is constructed and decimated by 2 vertically by simple sub-sampling. This mask is overlaid onto field data for the next round of candidate vector generation. Any field block that is more than 90% complete is excluded from the analysis as any vector that could possibly be resolved using it, already has been.

30 Other block areas that do not reconstruct with an MSE below the decided threshold are those along the bottom and left edges of the frame that are subject to new pixel gain and current pixel loss due to global panning motion (point 3 above).

Pixels with unresolved motion are replaced with half-band interpolated existing field pixels.

Plain block areas lack high frequency detail that would otherwise constitute aliasing. Their interpolated counterparts are generally subjectively undetectable in the final output image.

In general terms, and purely by way of example, the overall motion estimation algorithm described so far may be set out as the following list of steps. These take place for successive block sizes from the largest motion vector detection block size down to the smallest motion vector detection block size.

1. Generate a list of motion vectors for all block positions using a lowest MSE match criterion between fields 0 and 2, discarding any rogue vectors for which an intra-field similarity is better than any non-zero inter-field similarity found during the block search.
2. Repeat step 1 in respect of fields 1 and 3.
3. Pool the two vector lists. Remove vectors that do not appear at least twice in the pooled list (i.e. twice in either list or once in both lists).
4. Sort the list in order of vector popularity (most frequently occurring vector first)
5. For each vector in the list order:
 - 5.1 Reconstruct a test output image using field 0 as the current field and field 2 as the motion compensated field, using the selected vector from the pooled, sorted list.
 - 5.2 Repeat step 5.1 but using field 1 as the current field and field 3 as the motion compensated field.
 - 5.3 For successive block sizes from the largest verification block size down to the smallest verification block size:
 - 5.3.1 Obtain an intra-image match threshold block similarity measure using a displacement of one sub-pixel for the block in the test output image created from fields 0 and 2
 - 5.3.2 Match the block between the test output frame created from fields 0 and 2 and the test output frame created from fields 1 and 3.
 - 5.3.3 If the inter-test-frame match is better than the intra-frame threshold then accept the vector and commit the area covered by the block in the test output frame created using fields 0 and 2 to the final output image.

Motion generation and motion verification stages therefore work independently and both use variable block sizes (areas of around 2048 [up to 64*32] pixels to start, and as small as 4 pixels [e.g. 2 * 2] to finish) with a repeated division by 2 for size reduction.

There is an overlap rule that is used in the feedback of the results of motion vector verification for subsequent motion vector verification at a smaller block size. This is needed

because complex areas of the final output image may exist due to successful verification at various block sizes, even before the next variable block size is used to generate more motion vectors.

Any blocks that are verified in the final output image are marked as such. A “field-sized” representation of this mask is generated, i.e. a vertically sub-sampled version of a frame mask, where each location in the frame mask is “1” (in this example) if motion for that pixel has been verified (i.e. it is part of a block that has been verified) or “0” if not. The field-sized mask is then used to exclude areas of fields for the next block size motion vector generation. At the next motion vector generation block size, if a block overlaps the mask of already-verified output pixels by more than 90%, it is not used to generate motion vectors. that way, subsequent pools of motion vectors between fields should converge to the motion of unresolved image areas as the remainder of the output frame is resolved / verified. The intention is that dominant motion is always at the top of the pooled candidate motion vector list.

Starting with larger areas, especially when trying to estimate motion using potentially aliased field data, normally generates more accurate vectors requiring subsequent verification. this is a main reason for starting with larger blocks. Motion in objects around the same size or smaller than the block is probably undetected – hence the need to reduce the block size.

Various detailed aspects of the apparatus of Figure 3 will now be described.

Figure 16 schematically illustrates a half-band filtering approach. In Figure 16, rows of known pixels are indicated by shaded rows 410 and rows of motion compensated pixels by white rows 410. Assume that all of the pixels have been successfully motion compensated except for a particular pixel 420. Horizontal and vertical phase (sub-pixel positional) correction is about to be performed.

As part of this, it will be necessary to horizontally phase-correct a pixel (e.g. a pixel 440) adjacent to (or at least within a half-filter length of) the missing pixel 420. To apply the horizontal phase correction a polyphase filter is used, as described above. But such a filter would require a value for the pixel 420 as one of its inputs. There is no such value, so one has to be generated before phase correction of nearby pixels can be performed. Without such a value, the phase correction of the adjacent or nearby pixel 440 will be incorrect. An error of that type would be amplified by a subsequent vertical phase correction, and could lead to a subjectively disturbing artefact on the output frame.

It is therefore appropriate to find a good concealment value for the pixel 420. This is done as follows.

First, vertical half-band interpolation is used to generate a row of vertically interpolated pixel values disposed around the pixel 420, the number of vertically interpolated pixel values being sufficient for each tap of the horizontal polyphase filter. Vertical interpolation filters 430 are schematically indicated in Figure 16 by vertical broken-line
5 boxes. Each vertical interpolation filter generates a pixel value in the same row as the pixel 420. Note that the motion compensated values in the rows 410 are temporarily laid aside for this process; the vertical half-band filter refers only to real pixel values in the rows 400.

The above process generates a row of half-band interpolated pixel values around the pixel 420. These do not replace any valid motion compensated values in that row, but instead
10 are used just to arrive at a useful concealment value for the pixel 420.

A “reverse” horizontal phase shift is then applied by polyphase filter to this group. The “reverse” phase shift is a phase shift equal and opposite to the phase shift that is to be applied to the nearby or adjacent pixel 440. So, the inputs to this reverse phase shift filter are the half-band interpolated pixels in the group created around the pixel 420. The result of the
15 reverse phase shifting is a concealment pixel value for the pixel 420.

This concealment value for the pixel 420 is then used, as normal, for the horizontal phase shifting of the pixel 440.

This technique can be extended to situations where more than one pixel (within a filter size of a pixel to be horizontally phase shifted) is missing. The missing pixels and those
20 around them are generated by vertical half-band filtering. Then a reverse phase shift is applied to each one. The pixel to be phase shifted is then filtered using the polyphase filter, with at least some inputs to the filter being provided by the reverse phase-shifted pixels.

The motion vectors obtained in this way can then be used by the motion compensator to obtain missing pixels from one or more fields, generally one or two fields which are
25 temporally adjacent to the current field.

Figures 17a to 17c schematically illustrate aspects of GST filter design.

In particular, Figure 17a schematically illustrates a typical spatial frequency spectrum of an interlaced signal. The field contains spatial frequencies up to the field Nyquist limit (half of the field sampling rate), but because of the interlaced sub-sampling process, some of
30 these frequency components will in fact be aliased, as shown by a shaded area in Figure 17a.

However, it has been noted that the frequency content of a progressively scanned frame often does not extend as far as the frame Nyquist limit, which means that when the interlaced field was formed the alias components (which are “folded” about the field Nyquist limit) tend not to extend down to zero frequency.

The present embodiment can make use of this feature of interlaced signals, bearing in mind that the purpose of the GST spatial positional correction filter is to reduce alias effects. In frequency regions where aliasing is not present, it may not be necessary or even appropriate to apply the GST correction.

5 Figure 17b schematically illustrates a low pass ("LP") – high pass ("HP") filter response, whereby the frequency range up to the field Nyquist limit is divided into a lower frequency region and a higher frequency region. The cross-over point between the two regions is set in this embodiment to about 20% of the field Nyquist limit, based on empirical trials. In general, therefore, it is to be expected that the lower frequency region will not tend
10 to contain any alias frequency components, whereas the higher frequency region will contain alias frequency components.

The filter responses shown in Figure 17b are applied to the pixels on which the GST filter operates. The higher frequency region is subject to GST spatial positional correction, whereas the lower frequency components are not. The two are then added back together. In
15 empirical tests this has been found to give an improvement in signal to noise response of the overall system.

Figure 17c schematically illustrates an arrangement for implementing this filtering and part-correction technique.

In particular, the arrangement of Figure 17c shows the situation after the motion
20 compensation process has been carried out to generate motion compensated pixels from a field of the opposite polarity to the current field.

Referring to the current field pixels, these are upsampled by a factor of 2 at an upsampler 500. Upsampling is used because the low frequency / non-aliased component is being used to create a frame. This process is in fact an upsampling and filtering process – in
25 the implementation it is carried out as interpolation with the 20% field Nyquist frequency response applied to the filter used.

The upsampled pixels are then supplied in parallel to a low pass filter 510 and a compensating delay element 520. The low pass filter 510 generates the lower frequency region shown in Figure 17b. This is passed to a downsampler 530 and from there to an added
30 540.

The lower frequency output of the filter 510 is also subtracted from the delayed version of the original signal by a subtractor 550. This generates the higher frequency region which is downsampled by a downsampler 560, the result being passed to a GST correction filter 570.

With regard to the motion compensated pixels, these follow a similar path via an upsampler 580, a low pass filter 590, a compensating delay 600, a subtractor 610 and a downsampler 620, so that the higher frequency components of the motion compensated pixels are passed to the GST filter 570.

- 5 The output of the GST filter is added back to the lower frequency components of the current field pixels by the adder 540.

Note that generally speaking, the low frequency component obtained from the known field has little or no motion. The higher frequency contribution from the known field and the unknown field are treated by the positional correction filters to provide pixel values at the
10 positions required. This gives phase corrected high frequency information. This is added back to the low frequency contribution, which is basically a vertical interpolation of the known field.

Techniques for dealing with object and image edges, and with revealed pixels, will now be described with reference to Figures 18a to 18c.

- 15 Figure 18a schematically illustrates an image in which an object 700 is moving in a certain direction and the image background is moving in a different direction. A schematic initial block match grid is illustrated, marking the positions of the initial (largest) blocks used in the block match motion vector detection process.

Various potential problems can arise even with the simple situation of Figure 18a. For
20 example, at the trailing edge of the object 700, pixels will be uncovered as the object moves past. Such pixels cannot be derived from a preceding field because they did not exist in that field. At the boundary between the object and the background, it will be difficult to select the correct motion vector. Also, the GST filter as applied to pixels at or very near to the boundary will take in pixel values from the other side of the boundary. So, a filter which is
25 intended to improve the image by applying a sub-pixel correction to a boundary pixel could in fact harm the image by blurring the edge of the object 700.

As described earlier, during the motion vector generation stage, various different motion vectors are generally produced in respect of an image, but for the image of Figure 18a two vectors will be the most frequently occurring. These are a vector representing the motion
30 of the object 700 and a vector representing the motion of the background.

The verification of these vectors should be successful away from the boundary between the object 700 and the background. But the verification process will struggle at the boundary.

Figure 18b schematically illustrates the smallest block match grid which can be used in the block match process described above. Even with this smallest grid, there remain blocks (shown as dark squares) at the boundary between the object 700 and its moving background for which a motion vector cannot be properly resolved.

5 Reference will now be made to four blocks at the boundary region between the object 700 and the background. These blocks are shown schematically in Figures 18c to 18e.

In Figure 18c, an example is shown of a horizontal polyphase filter 720 applied to correct the phase of a pixel 710 just inside the background. Another example is shown of a horizontal polyphase filter 740 applied to correct the phase of a pixel 730 just inside the
10 object.

The filter 720 will be “contaminated” with object pixels (which will have an incorrect phase with respect to the background), and the filter 740 will be contaminated by background pixels (which will have an incorrect phase with respect to the object). It would be better to avoid such contamination. The same concerns apply to vertical GST filters (not shown in
15 Figure 18c).

It would be possible to use a mirroring process to re-use pixels within the correct area (object or background) so as to avoid this contamination. Figure 18d is a schematic example of such a process, in which taps in the polyphase filters 720, 740 which fall the “wrong side” of the boundary are actually applied to pixel values from the correct side of the boundary. As
20 illustrated, the mirroring process is symmetrical about the filter centre (the pixel 710 or 730) but the reflection could instead be symmetrical about the boundary. Similar considerations apply to vertical GST filters.

But unfortunately, such a mirroring process relies on a knowledge of where the boundary lies. The location of the boundary requires a successful motion vector verification
25 stage. So, this is a circular problem; the location of the boundary is needed to locate the boundary correctly.

The present embodiment addresses this problem by the elegantly simple technique of using shorter positional correction (polyphase / GST) filters for motion vector verification than for pixel output.

30 It is desired to retain longer filters for the final output image, because of the general increase in quality that this provides. Shorter filters can cause unwanted artefacts such as “ringing” in the output image.

But for motion vector verification, whereby a motion vector is assigned to each pixel, shorter filters give less risk of contamination and provide an increased chance of being able to assign motion vectors correctly near to a motion boundary.

Figure 18e schematically illustrates two short filters 720' and 740' applied to the motion vector verification stage. Longer filters such as those shown schematically in Figure 18c, possibly with mirroring as described with reference to Figure 18d, would be used for generation of the final output image. The same considerations can apply vertically as well as horizontally.

Typical filter tap lengths are as follows:

	Vertical	Horizontal
Motion vector verification	11	11
Final output image	41	31

10

It will be appreciated that the embodiments of the invention can be implemented in programmable or semi-programmable hardware operating under the control of appropriate software. This could be a general purpose computer or arrangements such as an ASIC (application specific integrated circuit) or an FPGA (field programmable gate array). The software could be supplied on a storage medium such as a disk or solid state memory, or via a transmission medium such as a network or internet connection, or via combinations of these.

20

CLAIMS

1. Video motion detection apparatus comprising:
 - means for applying a motion test to detect inter-image motion between two or more
 - 5 test images, the motion test generating a test result providing a motion vector and an indication of the integrity of the detected motion vector; and
 - means for applying the motion test to one of the test images, to detect intra-image similarity within that test image;
 - in which the motion vector generated by the inter-image motion test is made available
 - 10 as an output of the apparatus if its associated test result indicates a greater integrity than the test result obtained by the intra-image similarity test; and
 - the motion vector generated by the inter-image motion test is not made available as an output of the apparatus if its associated test result does not indicate a greater integrity than the test result obtained by the intra-image similarity test.
- 15 2. Apparatus according to claim 1, in which the motion vector detected in respect of intra-image similarity is constrained to be a motion vector indicative of non-zero image motion.
- 20 3. Apparatus according to claim 1 or claim 2, in which:
 - the means for applying a motion test comprises means for comparing a test area of an image with a candidate areas in that or another image to detect a displacement between the test area and a candidate area giving a greatest similarity between the test area and that candidate area;
 - 25 the motion vector is dependent on the displacement between the test area and that candidate area; and
 - the integrity of the motion vector is dependent on the detected similarity.
4. Apparatus according to claim 3, in which the candidate areas lie within a
- 30 predetermined maximum displacement with respect to the test area.
5. Apparatus according to claim 3 or claim 4, in which the integrity of the motion vector is inversely associated with a degree of image difference between the test area and that candidate area.

6. Apparatus according to claim 5, in which the degree of difference is a mean squared error between pixel values in the test area and corresponding pixel values in that candidate area.

5

7. Video processing apparatus comprising:
video motion detection apparatus according to any one of the preceding claims; and
means for processing a video signal in dependence on detected motion vectors.

10 8. Video processing apparatus according to claim 7, the apparatus being a scan conversion apparatus.

9. A video motion detection method comprising the steps of:
applying a motion test to detect inter-image motion between two or more test images,
15 the motion test generating a test result providing a motion vector and an indication of the integrity of the detected motion vector;
applying the motion test to one of the test images, to detect intra-image similarity within that test image;
making available the motion vector generated by the inter-image motion test as an
20 output if its associated test result indicates a greater integrity than the test result obtained by the intra-image similarity test; and
not making available the motion vector generated by the inter-image motion test as an output if its associated test result does not indicate a greater integrity than the test result obtained by the intra-image similarity test.

25

10. Computer software having program code which, when executed by a computer, is arranged to cause the computer to carry out a method according to claim 9.
- 5 11. A medium by which software according to claim 10 is provided.
12. A medium according to claim 11, the medium being a storage medium.
13. A medium according to claim 11, the medium being a transmission medium.

1/14

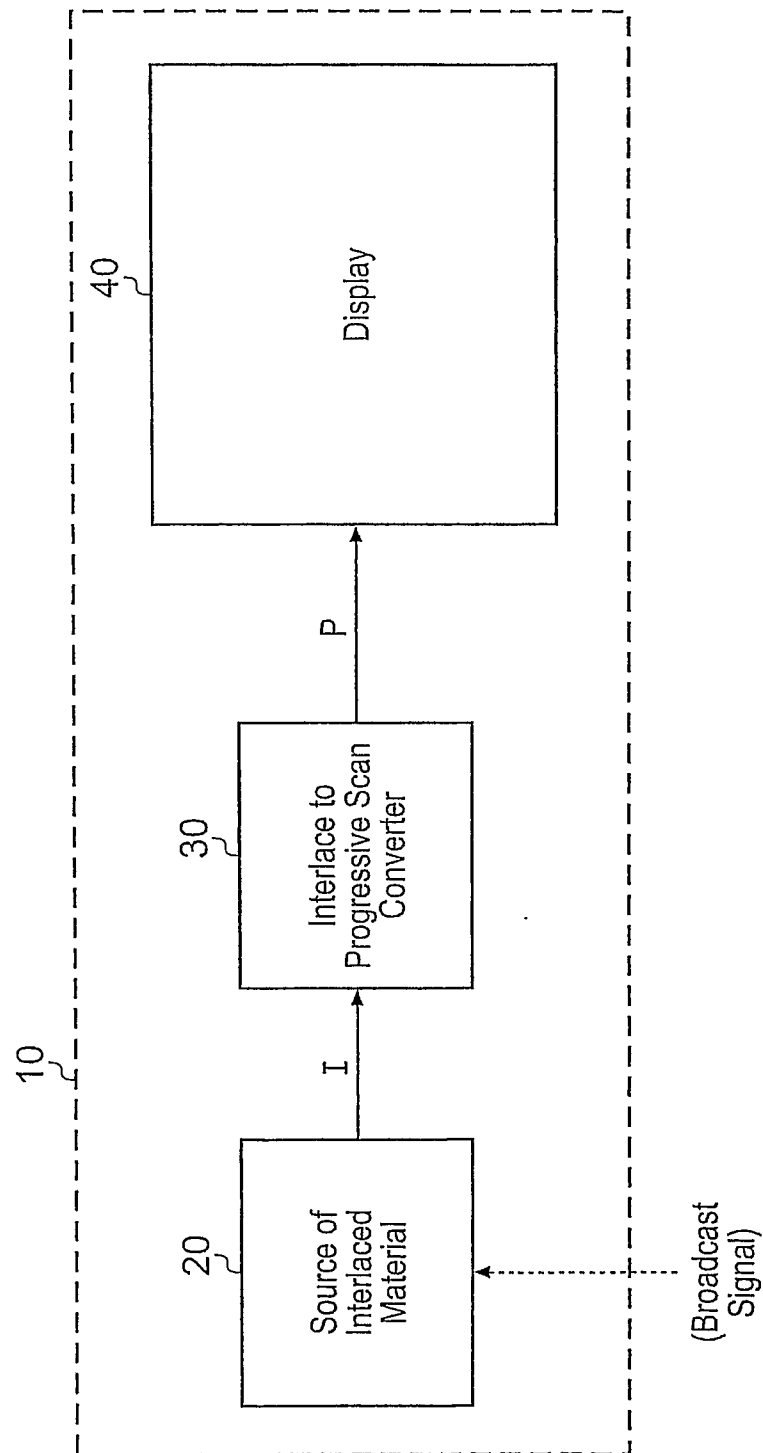


Fig. 1

2/14

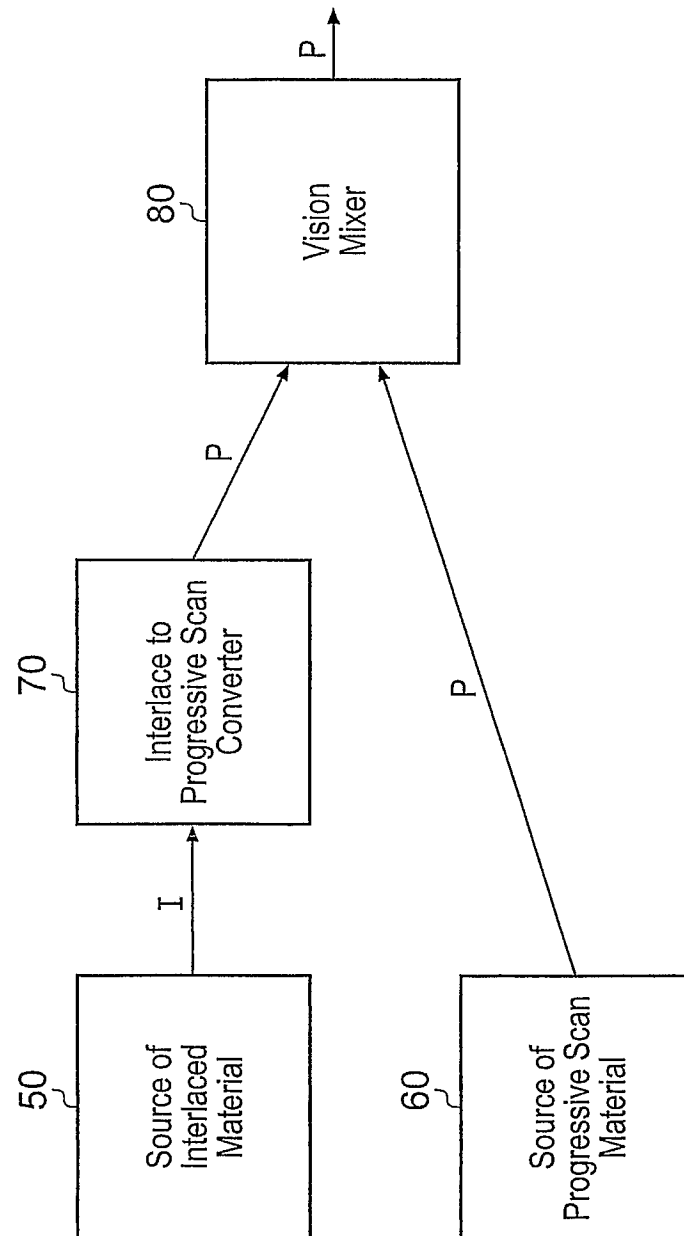


Fig. 2

3/14

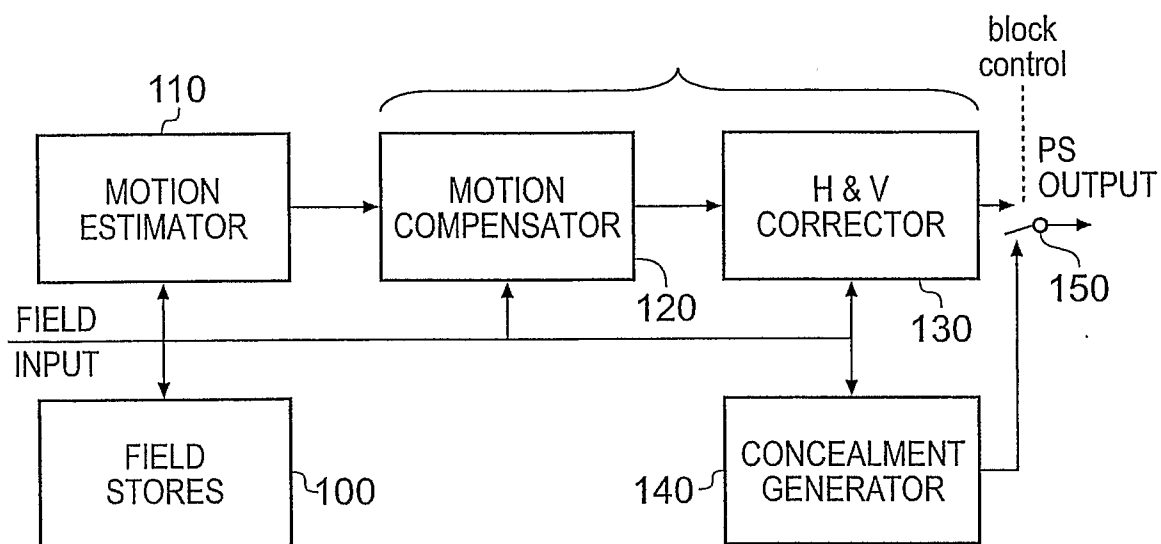


Fig. 3

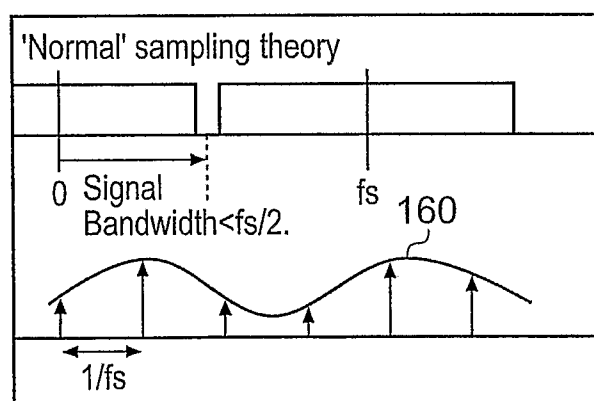


Fig. 4A

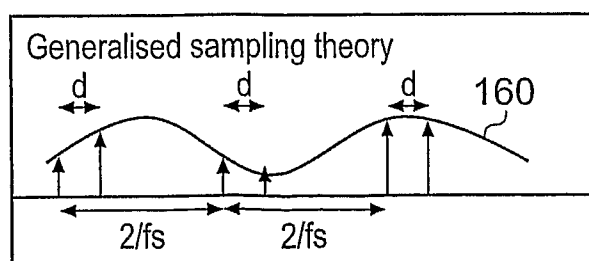


Fig. 4B

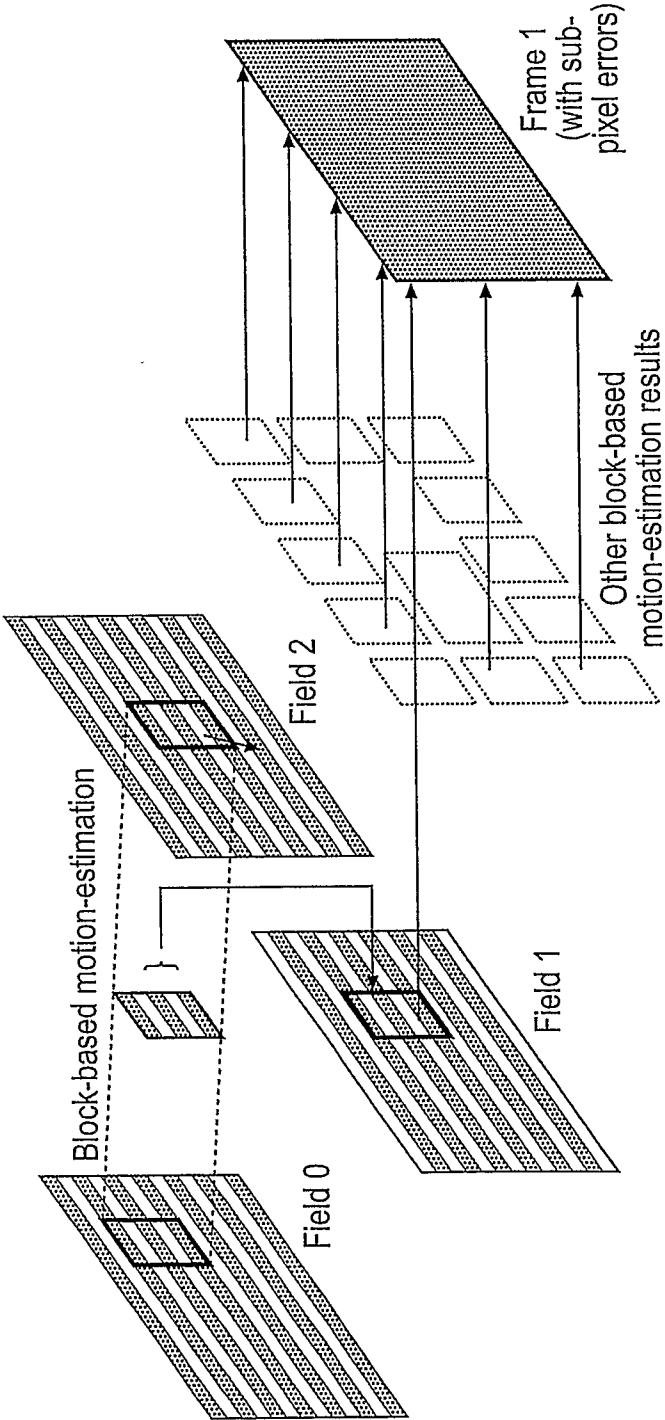


Fig. 5

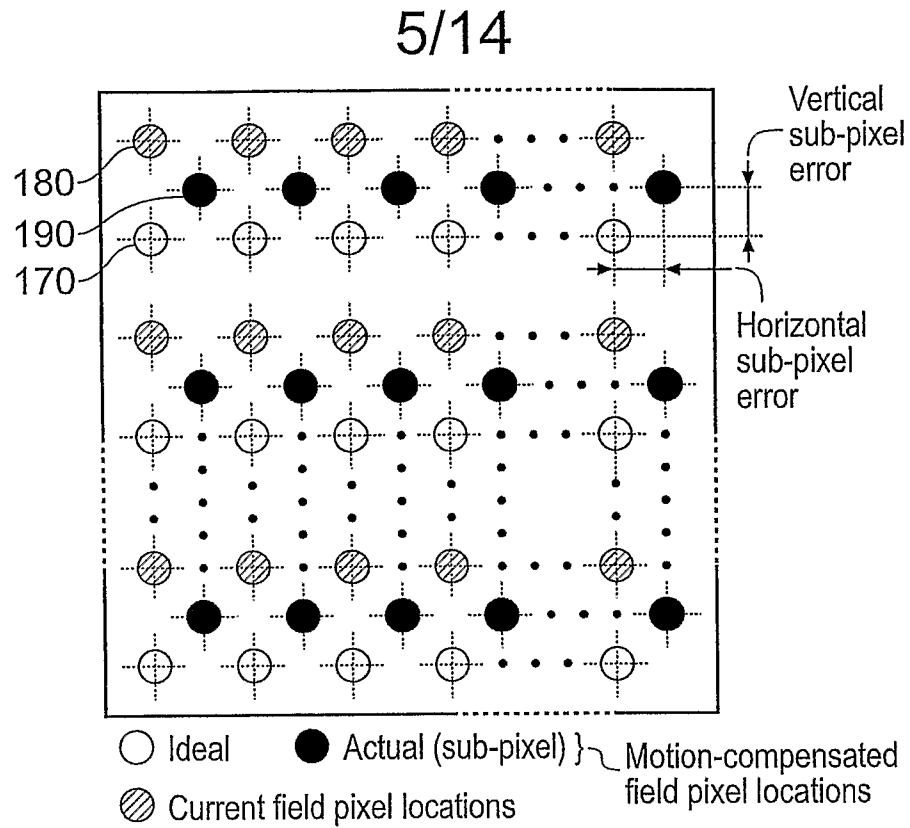


Fig. 6

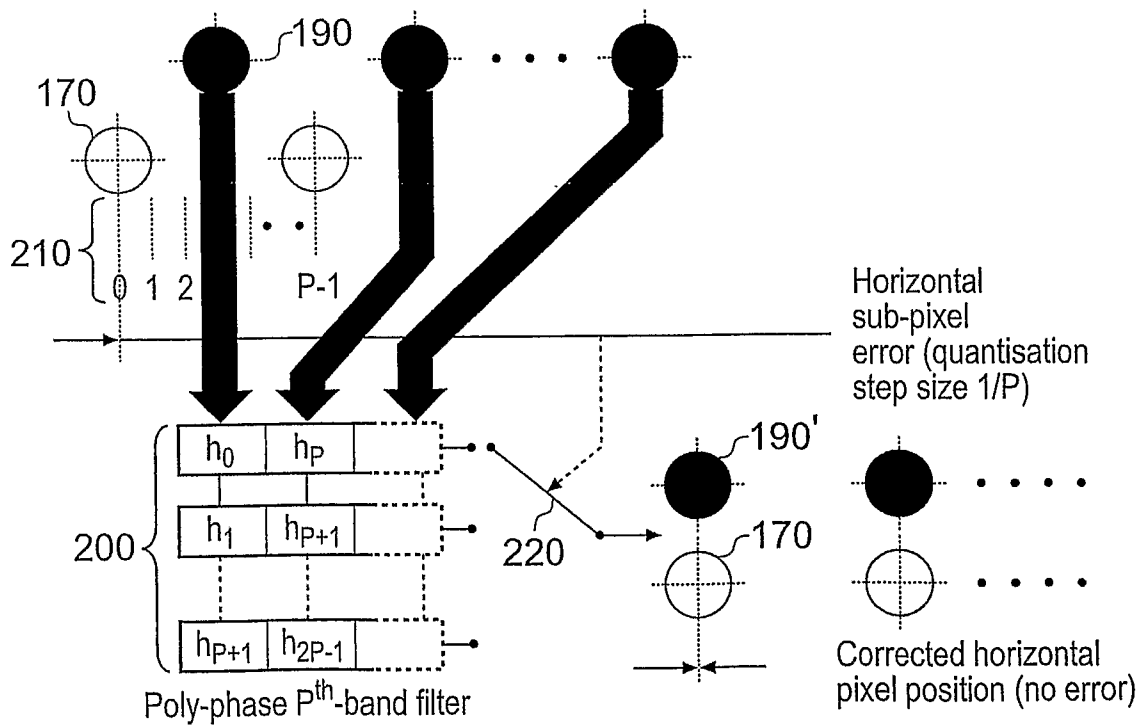


Fig. 7A

6/14

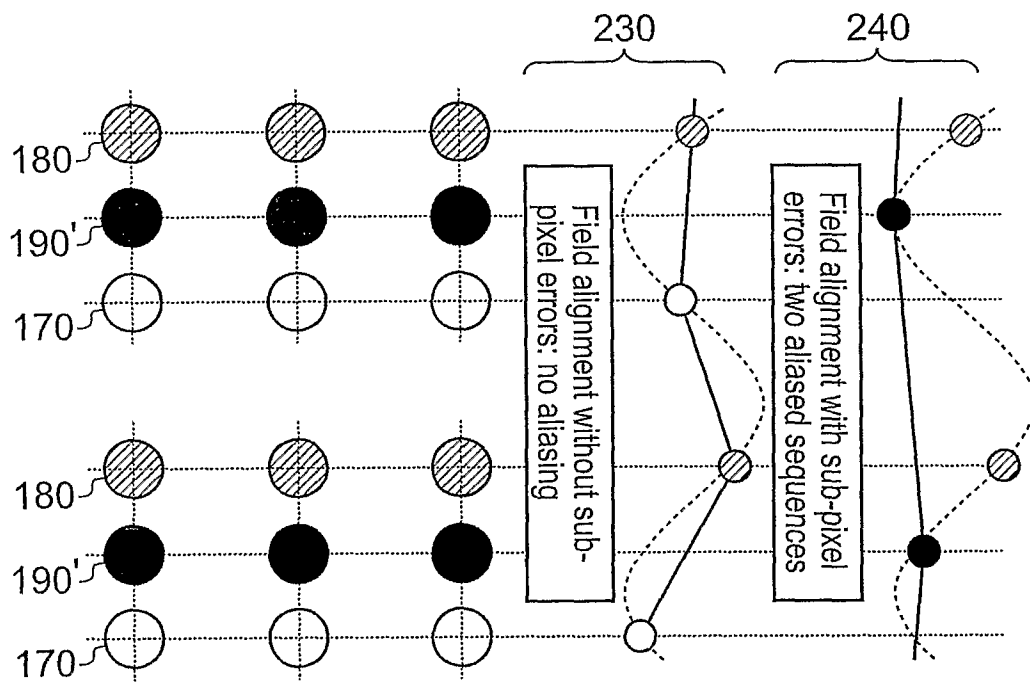


Fig. 7B

7/14

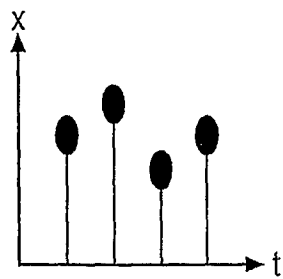


Fig. 8A

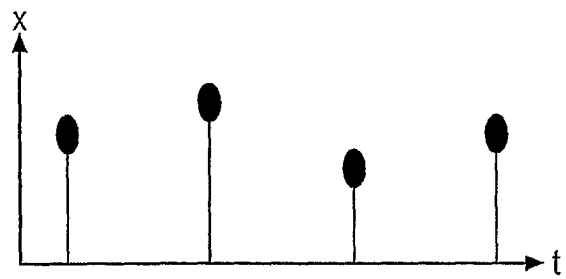


Fig. 8B

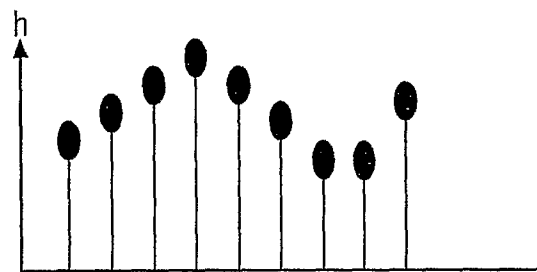


Fig. 8C

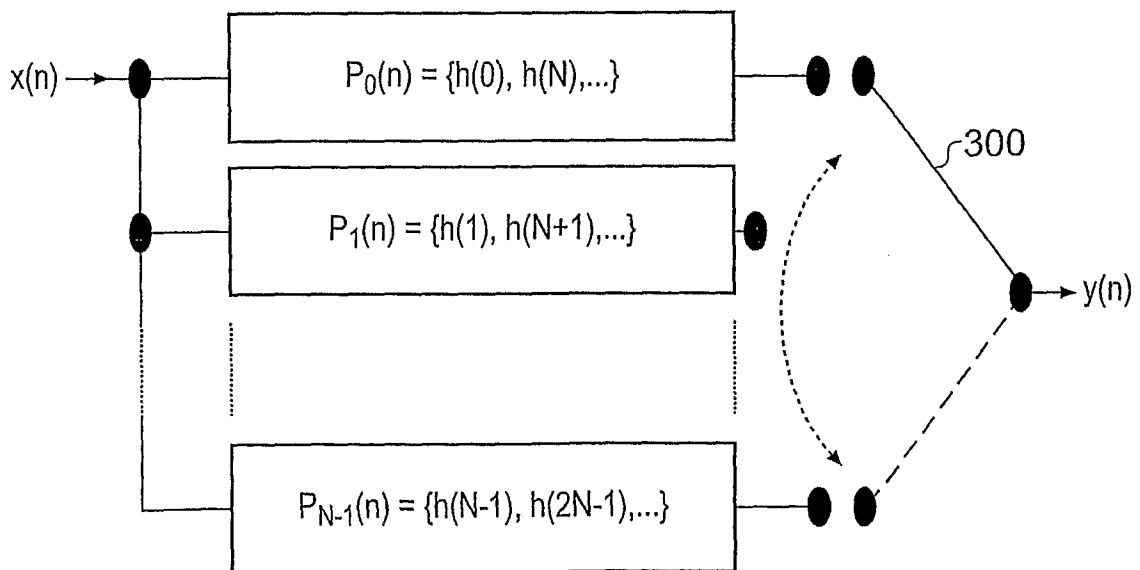


Fig. 9

8/14



Fig. 10



Fig. 11



Fig. 12

9/14

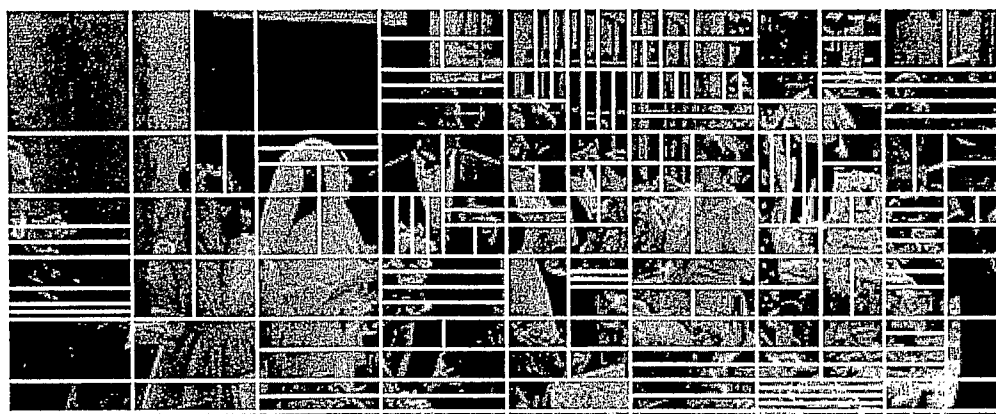


Fig. 13

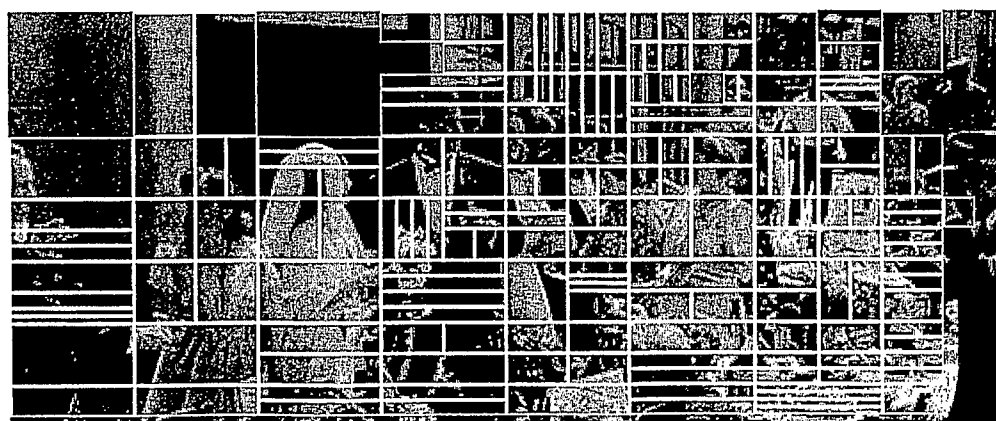


Fig. 14

10/14

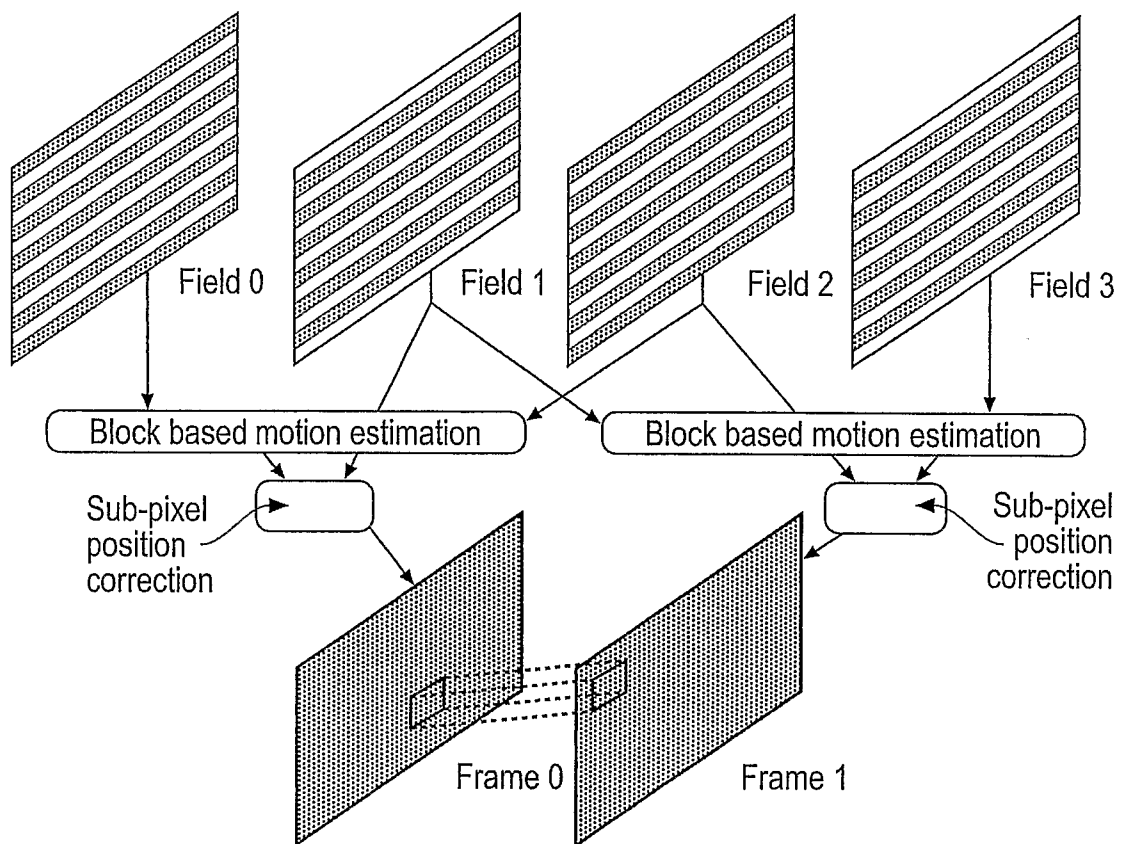


Fig. 15

11/14

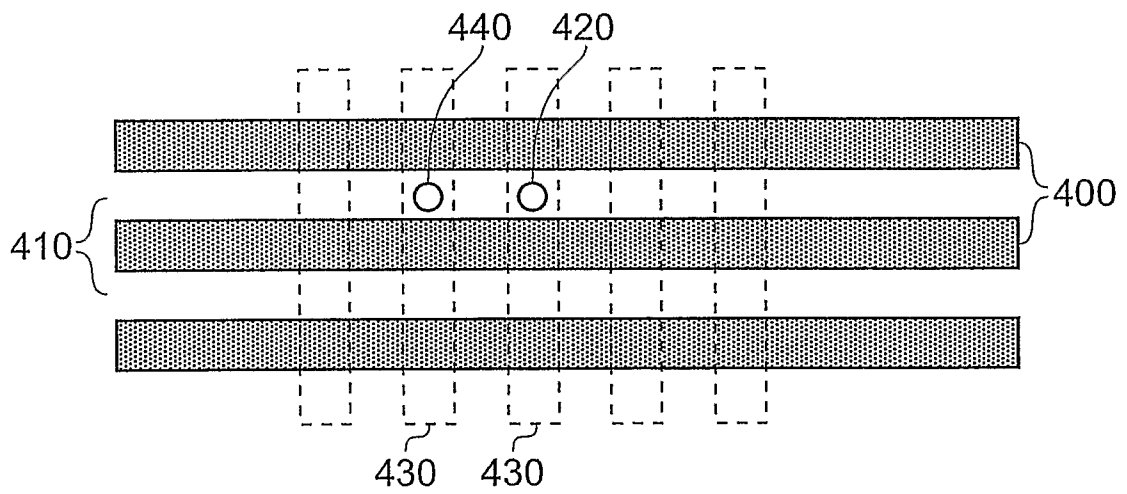


Fig. 16

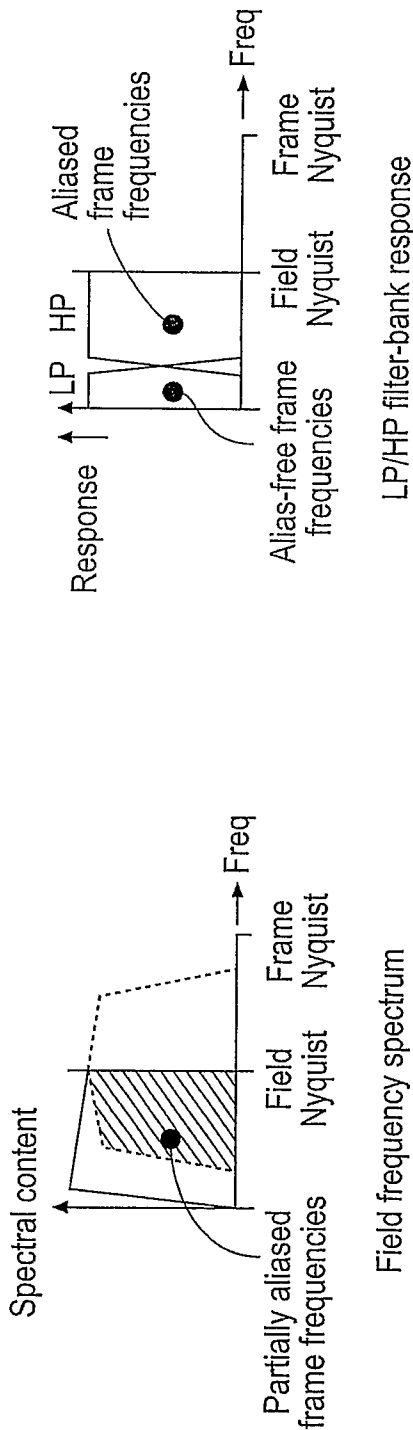


Fig. 17B

LP/HP filter-bank response

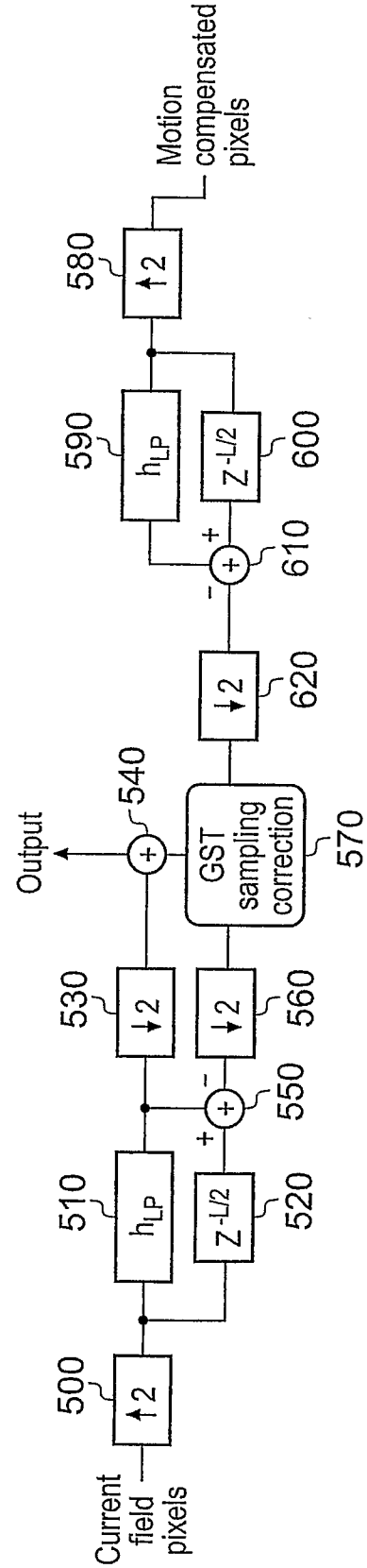


Fig. 17C

13/14

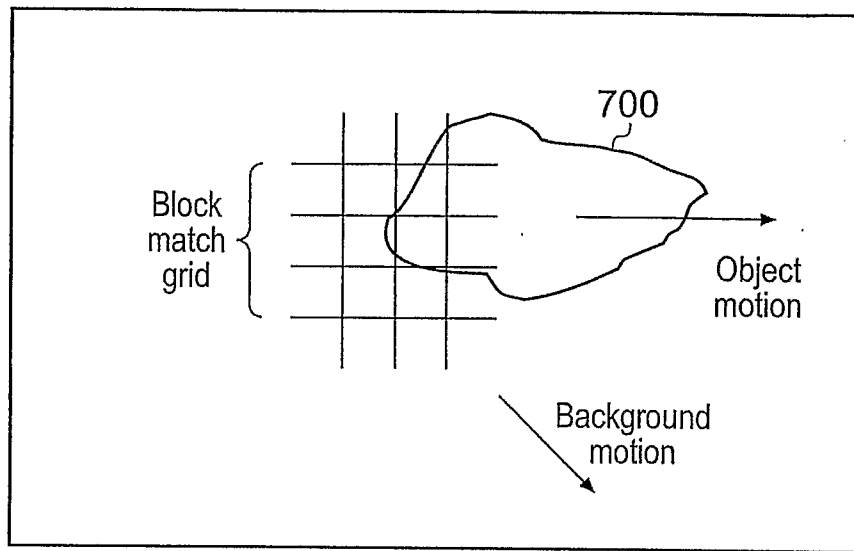


Fig. 18A

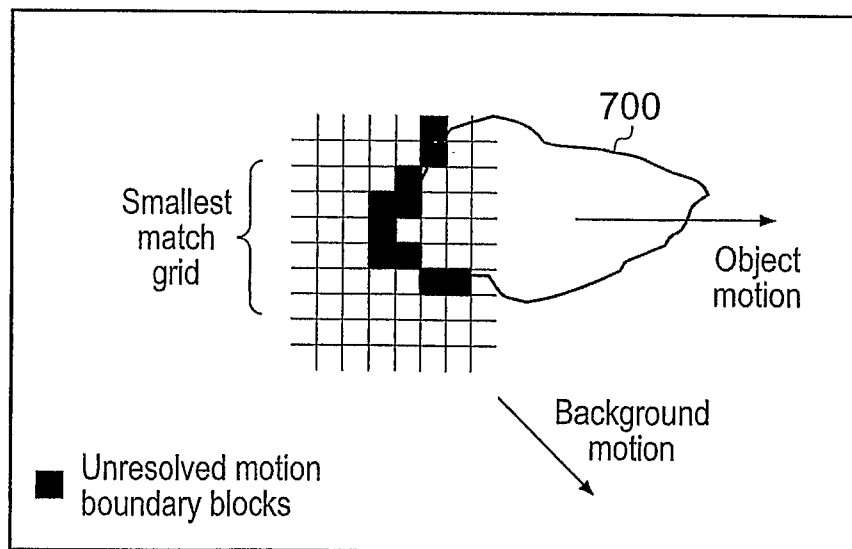


Fig. 18B

14/14

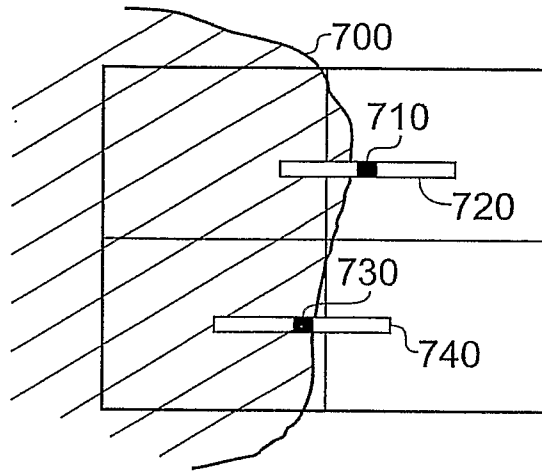


Fig. 18C

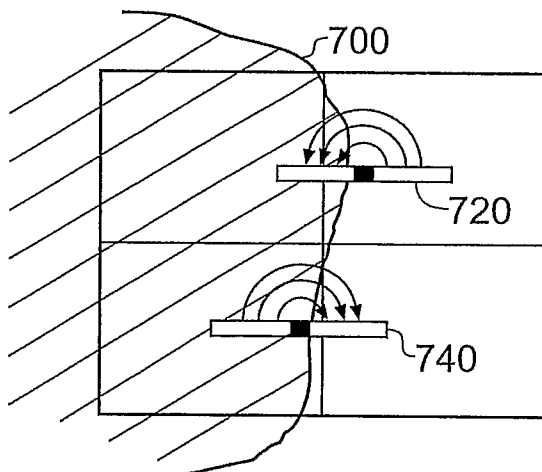


Fig. 18D

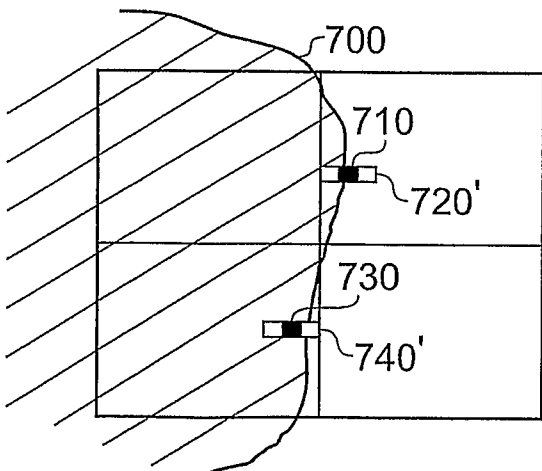


Fig. 18E

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2006/004016

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N5/14 H04N7/26

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EP0-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2004/156552 A1 (LARRIEU JEROME [FR] ET AL) 12 August 2004 (2004-08-12) abstract	1-13
A	US 4 924 310 A (VON BRANDT ACHIM [DE]) 8 May 1990 (1990-05-08) column 13, line 42 - column 14, line 4; figure 4	1-13
A	JP 2000 102014 A (SONY CORP) 7 April 2000 (2000-04-07) abstract	1-13
A	US 2002/054643 A1 (WATANABE HIROMI [JP] ET AL) 9 May 2002 (2002-05-09) the whole document	1-13

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- *Z* document member of the same patent family

Date of the actual completion of the international search

23 February 2007

Date of mailing of the international search report

06/03/2007

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Bardella, Xavier

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/GB2006/004016

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2004156552 A1	12-08-2004	FR 2850827 A1 WO 2004070651 A2	06-08-2004 19-08-2004
US 4924310 A	08-05-1990	EP 0293644 A1 JP 1092885 A	07-12-1988 12-04-1989
JP 2000102014 A	07-04-2000	NONE	
US 2002054643 A1	09-05-2002	NONE	