

VIDEO IMAGE COLOUR ENCODING

5 The present invention relates to the encoding and decoding of digital video image frames, and particularly to the coding of pixel colour values.

 A well known technique is run length coding, used for example as one of the coding modes in the Compact Disc interactive (CD-i) standard. The
10 technique allows sets of adjacent pixel values to be coded more compactly by specifying the colour once and then the count of the number (n) of identical pixels instead of repeating the colour code (n) times. Further information about run length coding may be found in, for example, "Principles of Interactive Computer Graphics" by W M Newman and R F Sproul,
15 International Student Edition, 1979, pp 287-289, pub McGraw-Hill, ISBN 0-07-066455-2, and a brief summary of the relevant points will be given below with reference to prior art Figures 1 to 3 of the accompanying drawings.

 Whilst run length coding is very efficient in terms of data compression
20 for long runs, this efficiency is reduced for shorter runs and frames with a large number of short runs can be computationally expensive to encode. Accordingly, it is an object of the present invention to provide a coding scheme which achieves at least a reasonable degree of compression without incurring large overheads due to complexity.

25 In accordance with the present invention there is provided a method of encoding pixel colour values for a digital video image frame in which each different colour within the image is assigned a colour value, where a predominant colour is identified for the image frame, and runs of at least two successive pixels of the predominant colour are encoded as a first code word
30 indicating a run and a second code word indicating the run length, and wherein pixels having colours other than the predominant colour are encoded

as codes containing at least the respective colour values.

Also in accordance with the present invention there is provided video image encoding apparatus arranged to encode pixel colour values for a digital video image frame by assigning to each different colour within the image a
5 respective colour value, wherein a predominant colour is identified for the image frame, the apparatus including means operable to identify runs of at least two successive pixels of the predominant colour and encode each such run as a first code word indicating a run and a second code word indicating the run length, and means operable to determine pixels having colours other
10 than the predominant colour and encode these as codes containing at least the respective colour values.

Still further in accordance with the present invention there is provided a video image signal comprising encoded frames of pixel colour values, wherein runs of two or more successive pixels of a predetermined
15 predominant colour for the frame are encoded in the form of a first code word indicating a run and a second code word indicating the run length, and pixels of colours other than the predetermined colour are encoded in the form of a code containing at least the respective colour value. A storage medium (such as a compact or floppy disc) carrying such a video image signal is also
20 provided, together with initialisation data including specification of the said predetermined predominant colour.

The invention additionally provides a video image display apparatus arranged to receive and decode signals such as described above, the said apparatus including at least one look-up table of pixel colours, including the
25 predominant colour, for the digital video image frames addressed by the encoded pixel colour values, and means operable to generate for display runs of at least two successive pixels of the predominant colour as well as individually generated pixels having colours other than the predominant colour.

30 In a first embodiment of the invention, only the predominant colour of the frame is run length coded. Thus, for the predominant colour it is not

necessary to specify the colour code each time a run is specified. Also, as the predominant colour, the efficiency is improved as the likelihood of runs is generally highest. A particularly common situation will be that there will be many runs of short to medium length in the predominant colour with fewer such runs in other colours. To achieve higher efficiency for this situation, all runs of, for example, two or more pixels of the same colour might be run-length encoded to achieve high data compression but still without the requirement for at least the short to medium length runs (e.g 3 to 9 pixels) to contain a colour code where the run is of the predominant colour.

The first code word may suitably include a sub-code indicating one of a number of ranges of run lengths, with the second code word indicating a run length from within the range indicated. In this way, the length of the second code word can be kept shorter than would be the case if the absolute length were required to be specified.

In order to allow for situations where the predominant colour changes within the frame (for example a "split-screen" with a different colour background in each part), a further code may be provided including specification of a colour value, wherein subsequent to the placing of the further code in a stream of pixel colour codes, the predominant colour specified for encoded runs is changed to the colour specified in the further code. This further code is referred to as a CHANGE BACKGROUND code in the following description, and may comprise a first code word indicating a run and a sub-code indicating a change of predominant colour, and a second code word comprising the colour value for the new predominant colour. By having the same sub-code in the first code word to indicate both a run and a change of predominant colour, with the difference being indicated by a specific sub-code of the first code word, the number of "control" codes is kept to a minimum, thus allowing a greater number of "free" codes to be assigned to colours.

To avoid the inefficiency associated with run-length coding of short

runs, runs of one or two pixels of the predominant colour may be respectively coded as one or two successive iterations of the colour value for the predominant colour, or unique short codes may be respectively assigned.

To allow for the situation where the predominant colour changes
5 between successive frames, the code sequence for a frame may suitably commence with the further code specifying the initial predominant colour, that is to say instructing a change to the specified predominant colour to be used for all subsequent runs, even if unchanged from the previous frame.

If it is the case that the predominant colour may change a number of
10 times during the course of a frame, the change command may prove inefficient. Accordingly, in a still further embodiment of the present invention a subset of the colours within the image may be identified as predominant colours, with colours (other than predominant) being assigned respective colour codes of a first length, predominant colours being assigned respective
15 colour codes of a second length shorter than the first, and runs of one of the predominant colours being encoded as a first code word indicating a predominant colour run and a second code word indicating in separate sub-codes the run length and (shorter) colour code respectively.

20 Preferred embodiments of the present invention will now be described, by way of example only, and with reference to the accompanying drawings in which:

Figure 1 represents a known colour code for a single pixel;

Figure 2 represents a known colour code for a pixel run of length L;

25 Figure 3 represents a sequence of pixel colour values converted to colour codes as in Figures 1 and 2;

Figure 4 is a table of code values according to a first embodiment of the invention;

Figure 5 is an alternative version of the table of Figure 4, representing
30 a first alternative embodiment of the invention;

Figures 6 and 7 are tables of code values for background and non-

background pixels respectively, representing a further alternative embodiment of the invention;

Figure 8 is a block schematic diagram of a pixel code detector stage for a decoder apparatus; and

5 Figures 9 to 13 are flow charts illustrating operation of the controller of Figure 8.

By way of illustration, it will initially be useful to consider a conventional scheme for encoding pixel colour values, with reference to
10 Figures 1 to 3. In the case where each pixel colour value occupies seven bits (Figure 1), as in the run length coding used for CD-i, the run specification (Figure 2) occupies two bytes. One byte contains the colour value C in seven of its bits and the other contains a count value L. For a run of more than two pixels length, the run length coded form will be more compact than
15 separately coding each pixel of the run.

However, in this case coding a single pixel as a run of length one would be inefficient since this would occupy two bytes whereas the single colour pixel could be contained in one byte. The run length coding scheme used for CD-i therefore allows two adjacent pixels to be defined in both a
20 repeated one byte code and a two byte run length sequence. The spare bit in the colour byte (the most significant bit) is used to indicate this difference. If a colour byte has its most significant bit set to zero then it is a single colour value as in Figure 1 while if the most significant bit is set to one then this byte is the start of a run specification and will be followed by a length
25 byte as in Figure 2. These one and two byte codes are combined as required to produce a coded sequence, a selection of which is shown in Figure 3.

The first embodiment of the invention provides coding for the particular case where an image principally consists of one background colour with few instances of significant runs of any other colour value. In such
30 circumstances it is preferred to run length code runs of pixels of the background colour but to individually colour code the pixels of other runs.

This eliminates the need to indicate in the coded data stream which colour values in the coded stream represent run length coded material and which do not: in this way the initial bit setting of 0 or 1 (as with Figures 2 and 3) is not required. The technique also ensures that the run length coded data need
5 only contain a count value; since one colour is specified as background, it is not necessary to specify a colour value for the run.

As will be described below, the method uses variable length code, which may comprise variable length code words, to further optimise performance. In the particular implementation to be described, all the
10 variable length codes are set to be an integer number of four-bit words (nibbles) long.

Each pixel colour value is defined in four bits. Conventionally, this would allow sixteen possible colours to be specified by the use of a colour look up table, although in this embodiment the number of different colours
15 is limited to fifteen, for reasons explained below, in order to improve the operation of the coding scheme.

Colour codes 1 to 15 (expressed as nibble values 0001 to 1111) are not compressed, but are simply added to the coded data stream without further compression. The coding used to indicate background colour run
20 lengths is shown in the table of Figure 4: the names given to the different codes (SHORT, MEDIUM etc) are merely illustrative and have no effect on the codes themselves. The benefits of run length coding can clearly be seen from the longest code shown in Figure 4, that for a run of 303 background colour pixels which, rather than requiring a colour code nibble for each pixel,
25 requires only 4 nibbles to define the run.

For background runs of 3 or more pixels, colour code 0 (binary 0000) is placed in the coded stream to indicate the start of a run coded sequence of colour 0, which is assumed to be the background colour. If the following four bit code is in the range 0011 to 1111 then the run (SHORT) is of a
30 length between 3 and 15 pixels, with the binary coded number of the second nibble directly representing the length of the run. Where the second nibble

is also code 0 (0000), the run (MEDIUM) is of a length between 17 and 31 pixels, with a third nibble specifying a value from 1 to 15 (binary 0001 to 1111) to be added to 16 to give the run length.

If, on the other hand, the second nibble of the sequence is 0001, then
5 the third nibble in the data stream has values between 0000 and 1111 representing run lengths (LONG) between 32 and 47 pixels.

The final case (LONGER) is that the second nibble has value 0010, and the two following four bit nibbles are taken together as an eight bit value representing run lengths between 48 and 303.

10 In order to allow a single pixel of background colour to be represented in a single four-bit nibble, without the following code being misconstrued as a run length, colour code 0001 is also taken to represent the background colour: this is the reason why only 15 colours can be represented in this scheme. Additionally, unless it directly follows colour code 0000 (as in the
15 codes for 17, 32 or 49 pixel run lengths), code 0001 represents a single pixel of the background colour. As will be noted from Figure 4, there will never be more than two colour 0 codes (0000) in sequence: this enables the scheme to satisfy MPEG constraints, where a series of such codes is used to indicate a re-synchronisation.

20 Since the coding arrangement described above only allows background colour runs of 3 or more pixels, a background colour run of length 2 (SINGLE +) is represented by two consecutive colour values of 0001. Also, a background colour run of a length 16 (SHORT +) is provided as a coded run of length 15 followed by a single pixel coded 0001. By the use of this
25 second colour code for the predominant or background colour, expansion is avoided - that is to say a screen of N pixels will be coded in a maximum of N codes, and generally much fewer.

In an environment where code lengths are not constrained to be a multiple of four bits, Huffman coding (assigning the shortest codes to the
30 most commonly specified run lengths) could give even more efficient compression. Codes of 8-bits or more may be used where it is necessary to

have a large range of available colours, and 2-bit codes may be used where very few colours are required.

In an alternative embodiment, the above-recited method may be modified as described below with reference to the table of Figure 5 to allow
5 the background colour represented by the colour code 0000 to be changed dynamically. This would be useful where the background colour changed part way down an image or where there is a large area of some other colour where the advantage of run coding outweighed the overhead of changing the run coded colour.

10 Comparison of Figures 4 and 5 will show that the two schemes are virtually identical save in that the codes for runs of 4 pixels upwards of the original embodiment now respectively represent run lengths one pixel shorter. the original embodiment code for a 3 pixel run (0000 0011) is now used as a CHANGE BACKGROUND code which indicates that the binary value (cccc)
15 from 1 to 15 of the following nibble (0 is reserved to the general designation of the background) is the colour code for the new background colour. The new background will be used until the code 0000 0011 0001 (change code + original background colour code) is received or a change to a third background colour is specified. Note that with a changed background, the
20 code for a run of 15 pixels length will have the colour code for that new background as the third nibble.

This alternative embodiment allows a change in the general background colour of the image to be specified at a cost of 12 bits of data. A temporary change to allow a long run of some non-background colour to
25 be coded will cost 24 bits (12 to change to the non-background colour and 12 to switch back to the background colour again), which gives savings for runs of 9 or more pixels.

In a further alternative embodiment, all runs of colour values (predominant/background or otherwise) may be run-length encoded, whilst
30 still using the identification of a predominant colour to produce savings for the most common occurrence, that is to say relatively short runs of

background colour pixels. The coding schemes for background and non-background runs are shown in Figures 6 and 7 respectively. In this scheme, colour codes 0 to 15 are available (indicated in Figure 7 as four-bit code cccc) with only code 0 reserved to the predominant colour. Since code 0 is also used to indicate the start of a run (of any colour) the code for a single background pixel (0000 1111) is wasteful but necessary to avoid anomalies. If a second colour code (e.g 1) were assigned to the background colour, the single background pixel could be coded in a single nibble, as with the schemes of Figures 4 and 5, albeit at the expense of a reduction in available colour codes. As shown by Figure 6, the particular saving for this scheme occurs with background runs of from three to nine pixels, each of which is encoded in a two nibble code. For background runs of 10 pixels or more, the same run-length coding scheme is used as for non-background colours (shown in Figure 7), that is to say comprising the run code (code 0), a range code (1100 or 1101 for MEDIUM or LONG runs respectively) the run length (4-bits for MEDIUM, 8-bits for LONG) and the colour code.

As shown in Figure 7, runs of 1 to 3 pixels of non-background colour are simply coded as separate instances of the colour code. Runs of from 4 to 7 pixels are coded as the run code (code 0) a run length code (4 to 7 indicated by respective binary values 1000 to 1011) and the colour code (cccc). In similar manner to the SHORT+ codes of the Figures 4 and 5 schemes, a run of eight non-background pixels is coded as a seven pixel run followed by a repetition of the colour code (i.e the code for a single pixel).

As the coded pixel values will generally be sent or stored as blocks of values, a further code is provided (0000 0000) to indicate the end of a block. Determination as to the start of a code will be handled at system level, such that the appearance of two consecutive nibbles of 0000 after the start of a code (as in the code for a run of 25 background pixels) will not be taken as indicating the end of a block.

In a still further alternative embodiment, the benefits arising from selection of a predominant or background colour may be extended to allow

for a subset of predominant colours to be specified: in this way the repeated use of the CHANGE BACKGROUND code of the scheme of Figure 5 may be avoided. The specification of a subset has particular benefit where longer codes are used, for example an 8-bit scheme where 256 colour codes are available. With such a scheme, the 16 most commonly used colours are coded as 4-bit (rather than 8-bit) values. In this way runs of between 3 and 9 pixels may be coded in two 8-bit words as

0000 0000 0LLL cccc

where the first word indicates a run of one of the predominant colours, the first half of the second word specifies the run length (0001 to 0111) and the second half of the second word comprises the 4-bit colour code. It will be noted that the first bit of the second word is 0 in this arrangement: this allows other coding possibilities including a full range of colours (indicated by using 1 at the start of this word).

As will be appreciated, whilst the benefits are greater for higher number bit schemes, the above technique of subsets could be applied to a 4-bit scheme, although the range of available run lengths or colour codes would be greatly restricted.

Turning now to apparatus for applying the above described schemes, encoding will generally be carried out by a suitably programmed microprocessor performing the functions of identification of the predominant colour, assignment of colour codes to pixels, and detecting single instances and runs of the predominant colour. Dedicated hardware may instead be provided but the flexibility of a reprogrammable system is preferred.

Considering in detail the implementation of the scheme of Figure 4, the code detector, for use in a decoder apparatus, may suitably be formed in hardware and have an arrangement as shown in Figure 8, where controller 20 controls the output nibble value (the colour code for each particular pixel) as well as an output clock signal, which clocks those nibble values. The controller requests input of a nibble on line 22 by sending a trigger signal on line 24. This input nibble is supplied to the first input of a multiplexer 26 as

well as to each of three comparator stages 28,30 and 32, the outputs of which are supplied to the controller 20. The three comparators 28,30,32 respectively compare the input nibble with codes 0000, 0001 and 0010: if the output from comparator 28 is positive, that is to say the input nibble is
5 code 0000, then the start of a run is indicated, in which case the controller sends a signal on select line 34 to the multiplexer 26 causing it to select, rather than the input nibble value, the colour code for the predominant (background) colour stored in register 36.

Having determined that a run is commencing, the next nibble will be
10 called for (via line 24) and, if the outputs of any of comparators 28,30 or 32 are positive, this will indicate the run is within the ranges identified as MEDIUM, LONG, or LONGER respectively as identified in Figure 4. If the output of none of the comparators is positive, then the run is in the SHORT range, that is to say between three and fifteen pixels and the input nibble
15 value is actually the duration of the run. This nibble value is clocked into an 8-bit counter 38 as the least significant nibble under control of a clock signal on line 40 from the controller 20. As the counter 38 counts down the value represented by the nibble, the output clock signal on line 32 clocks the background colour value from register 36 on the output line 44 until such
20 time as the nibble value has been counted out, when a zero detect signal from the counter is passed to the controller via line 46.

In the case where a run has been detected and the second nibble produces a positive output from one of comparators 28, 30 or 32, the following nibble (once called for) will be clocked into the counter 38 as the
25 least significant nibble (in the case of positive output from comparators 28 or 30) and the following two nibbles will be read into the counter as respectively the most and least significant nibbles in the case of a positive output from comparator 32. Where two nibbles are clocked into the 8-bit counter 38, that is to say in the case of a LONGER run the most significant
30 nibble is clocked into the counter under control of a further clock line 48 from the controller.



The operations of the controller 20 of Figure 8 in determining whether the input code describes a run, and if so, how long it is, will now be described with reference to the flow charts of Figures 9 to 13. From a start point A (Figure 9) a new input nibble is triggered at 71 following which the output of the comparators is checked, at step 72, to see whether the nibble is code 0000. If not, the code accordingly represents the colour code (colour other than background) for a pixel and, at step 73, the multiplexer 26 (Figure 8) is set to output the input nibble value. At step 74, one clock signal for this nibble value is also output following which the procedure returns to start point A for the next input nibble.

Where, at step 72, the nibble value is 0000 a run is indicated which is processed generally as shown in Figure 10. The processing of a run starts at step 81, with the triggering of the next input nibble and, at step 82, by setting the multiplexer 26 (Figure 8) to output colour code 0000 from register 36. Following this, the new input nibble is checked at steps 83, 84 and 85 to see whether it corresponds to one of the nibble 2 codes from Figure 4 representing a MEDIUM, LONG, or LONGER run as identified in that Figure. If the answer is "No" to all of these checks, then the run is in the range of three to fifteen pixels and the nibble value, loaded into the 8-bit counter, sets the value to be clocked. The repeated outputting of clock signals for the colour code 0000 is handled by loop stages 87, 88 and 89 which repeatedly decrement the clock value and check for zero whilst outputting the clock signal until such time as the counter reaches zero at which point the process returns to start A (Figure 9) for the next input nibble value.

Where any of the stages 83, 84 or 85 of Figure 10 answers "Yes", a run in the range of 17 to 303 pixels is specified. Where it is the output from stage 83 which is "Yes" (the second nibble value is 0000), the run is in the range of 17 to 31 pixels and is handled as shown in Figure 11. Initially, at stage 91, the value 0000 from register 36 is clocked out 16 times, following which a new input nibble is triggered at step 92 which nibble is clocked into the 8-bit counter at step 93. This least significant nibble value in counter 38

is then clocked out by countdown/zero detect loop (steps 94, 95 and 96) until the counter reaches zero, at which point the process returns to start A (Figure 9) for the next input nibble.

Where the nibble 2 value is detected as 0001, at step 84 of Figure 10, the run is between 32 and 47 pixels which is handled as shown in Figure 12. The handling of the LONG run in Figure 12 is substantially identical to that for the MEDIUM run of Figure 11, save in that the colour code 0000 is clocked out 32 times at stage 101 prior to triggering and clocking into the counter of the next nibble value at steps 102 and 103. Again, the countdown loop of steps 104, 105 and 106 causes output of the additional clock values beyond 32 forming the total run length.

Where the output of step 85 of Figure 10 is positive, that is to say the run is LONGER of length between 48 and 303 pixels, the run is handled as shown in Figure 13. In this case, the process starts by clocking out the value 0000 48 times at step 111 following which the next input nibble is triggered at step 112. Since, for a LONGER run, the run length beyond 48 pixels is specified as an 8-bit code, the nibble triggered at step 112 represents the most significant half of the 8-bit code and is clocked into the 8-bit counter as such under control of a signal from the controller on line 48 (Figure 8). Following this the next nibble is triggered at step 114 which nibble, comprising the least significant half of the 8-bit code is clocked in as the least significant nibble into 8-bit counter 38. Finally, the countdown loop formed by steps 116, 117 and 118 counts down the 8-bit value represented in the counter until this reaches zero at which point the process returns to start A of Figure 9.

Whilst not shown in Figure 8, it will be appreciated that, where a change of background is permitted (as with the table of Figure 5) the contents of register 36 may be modified under control of the controller 20 to contain the newly selected background colour value rather than 0000. In addition, the number of times this value is initially clocked out for a run, at steps 91, 101 and 111 of Figures 11, 12 and 13 respectively, will be 15, 31

and 47 rather than 16, 32 and 48 as previously described.

In order to effect the coding scheme of Figures 6 and 7, the modifications to the apparatus of Figure 8 will be generally apparent to the skilled reader and will not be discussed in detail. It will be understood that
5 the comparison stages 28,30,32 will need to be configured to check for the particular code words used, and that register 36 will have provision for receiving, storing and outputting (under control of controller 26) colour values other than the background colour value during runs of the non-background colours. Similar considerations apply to the embodiment supporting subsets
10 of predominant colours.

From reading the present disclosure, other modifications will be apparent to persons skilled in the art. Such modifications may involve other features which are already known in the field of video signal encoding systems and devices and component parts thereof and which may be used
15 instead of or in addition to features already described herein. Although claims have been formulated in this application to particular combinations of features, it should be understood that the scope of the disclosure of the present invention also includes any novel feature or any novel combination of features disclosed herein either explicitly or implicitly or any generalisation thereof, whether or not it relates to the same invention as presently claimed
20 in any claim and whether or not it mitigates any or all of the same technical problems as does the present invention. The applicants hereby give notice that new claims may be formulated to such features and/or combinations of features during the prosecution of the present application or of any further
25 application derived therefrom.

CLAIMS

1. A method of encoding pixel colour values for a digital video image frame in which each different colour within the image is assigned a colour value, wherein a predominant colour is identified for the image frame, and runs of at least two successive pixels of the predominant colour are encoded as a first code word indicating a run and a second code word indicating the run length, and wherein pixels having colours other than the predominant colour are encoded as codes containing at least the respective colour values.
2. A method as claimed in Claim 1, wherein each pixel having a colour other than the predominant colour is separately coded as its respective colour value only.
3. A method as claimed in Claim 1, wherein runs of two or more successive pixels of a colour other than the predominant colour are encoded together as a first code word indicating a run, a second code word indicating the run length and a third code word indicating the colour value.
4. A method as claimed in Claim 1, wherein the first code word includes a sub-code indicating one of a number of ranges of run lengths and the second code word indicates a run length from within the range indicated.
5. A method as claimed in any of Claims 1 to 4, in which a further code is provided including specification of a colour value, wherein subsequent to the placing of the further code in a stream of pixel colour codes, the predominant colour specified for encoded runs is changed to the colour specified in the further code.
6. A method as claimed in Claim 5, wherein the further code



comprises a first code word indicating a run and a sub-code indicating a change of predominant colour, and a second code word comprising the colour value for the new predominant colour.

5 7. A method as claimed in Claim 1, wherein runs of one or two pixels of the predominant colour are respectively coded as one or two successive iterations of the colour value for the predominant colour.

 8. A method as claimed in Claim 5, wherein the code sequence for
10 a frame commences with the further code specifying the initial predominant colour.

 9. A method as claimed in Claim 1, in which a subset of the colours within the image are identified as predominant colours, colours other
15 than predominant colours are assigned respective ones of a first range of colour value codes of a first length, the predominant colours are assigned respective ones of a second range of colour value codes of a second length shorter than the first length, and a run of at least two pixels of one of the predominant colours is encoded as a first code word indicating a predominant
20 colour run and a second code word indicating in separate sub-codes the run length and colour value code respectively.

 10. A method as claimed in Claim 1 wherein the second code word precedes the first code word in the code for a run.

25

 11. Video image encoding apparatus arranged to encode pixel colour values for a digital video image frame by assigning to each different colour within the image a respective colour value, wherein a predominant colour is identified for the image frame, the apparatus including means operable to
30 identify runs of at least two successive pixels of the predominant colour and encode each such run as a first code word indicating a run and a second



code word indicating the run length, and means operable to determine pixels having colours other than the predominant colour and encode these as codes containing at least the respective colour values.

5 12. A video image signal comprising encoded frames of pixel colour values, wherein runs of two or more successive pixels of a predetermined predominant colour for the frame are encoded in the form of a first code word indicating a run and a second code word indicating the run length, and pixels of colours other than the predetermined colour are encoded in the form
10 of a code containing at least the respective colour value.

 13. A storage medium carrying a video image signal as claimed in Claim 12, together with initialisation data including specification of the said predetermined predominant colour.

15

 14. Video image display apparatus arranged to receive and decode the signal of Claim 12, said apparatus including at least one look-up table of pixel colours, including the predominant colour, for the digital video image frames addressed by the encoded pixel colour values, and means operable to
20 generate for display runs of at least two successive pixels of the predominant colour as well as individually generated pixels having colours other than the predominant colour.

