

Method and Apparatus for Variable Accuracy Inter-Picture Timing Specification for Digital Video Encoding

FIELD OF THE INVENTION

The present invention relates to the field of multimedia compression systems. In particular the present invention discloses methods and systems for specifying variable accuracy inter-picture timing.

BACKGROUND OF THE INVENTION

Digital based electronic media formats are finally on the cusp of largely replacing analog electronic media formats. Digital compact discs (CDs) replaced analog vinyl records long ago. Analog magnetic cassette tapes are becoming increasingly rare. Second and third generation digital audio systems such as Mini-discs and MP3 (MPEG Audio - layer 3) are now taking market share from the first generation digital audio format of compact discs.

The video media has been slower to move to digital storage and transmission formats than audio. This has been largely due to the massive amounts of digital information required to accurately represent video in digital form. The massive amounts of digital information needed to accurately represent video require very high-capacity digital storage systems and high-bandwidth transmission systems.

However, video is now rapidly moving to digital storage and transmission formats. Faster computer processors, high-density storage systems, and new efficient compression and encoding algorithms have finally made digital video practical at consumer price points. The DVD (Digital Versatile Disc), a digital video system, has been one of the fastest selling consumer electronic products in years. DVDs have been rapidly supplanting Video-Cassette Recorders (VCRs) as the pre-recorded video playback system of choice due to their high video quality, very high audio quality, convenience, and extra features. The antiquated analog NTSC (National Television Standards Committee) video transmission system is currently in the process of being replaced with the digital ATSC (Advanced Television Standards Committee) video transmission system.

Computer systems have been using various different digital video encoding formats for a number of years. Among the best digital video compression and encoding systems used by computer systems have been the digital video systems backed by the Motion Pictures Expert Group commonly known by the acronym MPEG. The three most well known and highly used digital video formats from MPEG are known simply as MPEG-1, MPEG-2, and MPEG-4. VideoCDs (VCDs) and early consumer-grade digital video editing systems use the early MPEG-1 digital video encoding format. Digital Versatile Discs (DVDs) and the Dish Network brand Direct Broadcast Satellite (DBS) television broadcast system use the higher quality MPEG-2 digital video compression and encoding system. The MPEG-4 encoding system is rapidly being adapted by the latest computer based digital video encoders and associated digital video players.

The MPEG-2 and MPEG-4 standards compress a series of video frames or video fields and then encode the compressed frames or fields into a digital bitstream. When encoding a video frame or field with the MPEG-2 and MPEG-4 systems, the video frame or field is divided into a rectangular grid of macroblocks. Each macroblock is independently compressed and encoded.

When compressing a video frame or field, the MPEG-4 standard may compress the frame or field into one of three types of compressed frames or fields: Intra-frames (I-frames), Unidirectional Predicted frames (P-frames), or Bi-Directional Predicted frames (B-frames). Intra-frames completely independently encode an independent video frame with no reference to other video frames. P-frames define a video frame with reference to a single previously displayed video frame. B-frames define a video frame with reference to both a video frame displayed before the current frame and a video frame to be displayed after the current frame. Due to their efficient usage of redundant video information, P-frames and B-frames generally provide the best compression.

SUMMARY OF THE INVENTION

A method and apparatus for variable accuracy inter-picture timing specification for digital video encoding is disclosed. Specifically, the present invention discloses a system that allows the relative timing of nearby video pictures to be encoded in a very efficient manner. In one embodiment, the display time difference between a current video picture and a nearby video picture is determined. The display time difference is then encoded into a digital representation of the video picture. In a preferred embodiment, the nearby video picture is the most recently transmitted stored picture.

For coding efficiency, the display time difference may be encoded using a variable length coding system or arithmetic coding. In an alternate embodiment, the display time difference is encoded as a power of two to reduce the number of bits transmitted.

Other objects, features, and advantages of present invention will be apparent from the company drawings and from the following detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

The objects, features, and advantages of the present invention will be apparent to one skilled in the art, in view of the following detailed description in which:

Figure 1 illustrates a high-level block diagram of one possible a digital video encoder system.

Figure 2 illustrates a series of video pictures in the order that the pictures should be displayed wherein the arrows connecting different pictures indicate inter-picture dependency created using motion compensation.

Figure 3 illustrates the video pictures from **Figure 2** listed in a preferred transmission order of pictures wherein the arrows connecting different pictures indicate inter-picture dependency created using motion compensation.

Figure 4 graphically illustrates a series of video pictures wherein the distances between video pictures that reference each other are chosen to be powers of two.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

A method and system for specifying Variable Accuracy Inter-Picture Timing in a multimedia compression and encoding system is disclosed. In the following description, for purposes of explanation, specific nomenclature is set forth to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that these specific details are not required in order to practice the present invention. For example, the present invention has been described with reference to the MPEG-4 multimedia compression and encoding system. However, the same techniques can easily be applied to other types of compression and encoding systems.

Multimedia Compression and Encoding Overview

Figure 1 illustrates a high-level block diagram of a typical digital video encoder **100** as is well known in the art. The digital video encoder **100** receives an incoming video stream of video frames **105** at the left of the block diagram. Each video frame is processed by a Discrete Cosine Transformation (DCT) unit **110**. The frame may be processed independently (an intra-frame) or with reference to information from other frames received from the motion compensation unit (an inter-frame). Next, a Quantizer (Q) unit **120** quantizes the information from the Discrete Cosine Transformation unit **110**. Finally, the quantized video frame is then encoded with an entropy encoder (H) unit **180** to produce an encoded bitstream. The entropy encoder (H) unit **180** may use a variable length coding (VLC) system.

Since an inter-frame encoded video frame is defined with reference to other nearby video frames, the digital video encoder 100 needs to create a copy of how decoded each frame will appear within a digital video decoder such that inter-frames may be encoded. Thus, the lower portion of the digital video encoder 100 is actually a digital video decoder system. Specifically, an inverse quantizer (Q^{-1}) unit 130 reverses the quantization of the video frame information and an inverse Discrete Cosine Transformation (DCT^{-1}) unit 140 reverses the Discrete Cosine Transformation of the video frame information. After all the DCT coefficients are reconstructed from iDCT, the motion compensation unit will use the information, along with the motion vectors, to reconstruct the encoded frame which is then used as the reference frame for the motion estimation of the next frame.

The decoded video frame may then be used to encode inter-frames (P-frames or B-frames) that are defined relative to information in the decoded video frame. Specifically, a motion compensation (MC) unit 150 and a motion estimation (ME) unit 160 are used to determine motion vectors and generate differential values used to encode inter-frames.

A rate controller 190 receives information from many different components in a digital video encoder 100 and uses the information to allocate a bit budget for each video frame. The rate controller 190 should allocate the bit budget in a manner that will generate the highest quality digital video bit stream that complies with a specified set of restrictions. Specifically, the rate controller 190 attempts to generate the highest quality compressed video stream without overflowing buffers (exceeding the amount of available memory in a decoder by

sending more information than can be stored) or underflowing buffers (not sending video frames fast enough such that a decoder runs out of video frames to display).

Multimedia Compression and Encoding Overview

In some video signals the time between successive video pictures (frames or fields) may not be constant. (Note: This document will use the term video pictures to generically refer to video frames or video fields.) For example, some video pictures may be dropped because of transmission bandwidth constraints. Furthermore, the video timing may also vary due to camera irregularity or special effects such as slow motion or fast motion. In some video streams, the original video source may simply have non-uniform inter-picture times by design. For example, synthesized video such as computer graphic animations may have non-uniform timing since no arbitrary video timing is created by a uniform video capture system such as a video camera system. A flexible digital video encoding system should be able to handle non-uniform timing.

Many digital video encoding systems divide video pictures into a rectangular grid of macroblocks. Each individual macroblock from the video picture is independently compressed and encoded. In some embodiments, sub-blocks of macroblocks known as 'pixelblocks' are used. Such pixel blocks may have their own motion vectors that may be interpolated. This document will refer to macroblocks although the teachings of the present invention may be applied equally to both macroblocks and pixelblocks.

Some video coding standards, e.g., ISO MPEG standards or the ITU H.264 standard, use different types of predicted macroblocks to encode video pictures. In one scenario, a macroblock may be one of three types:

1. I-macroblock – An Intra (I) macroblock uses no information from any other video pictures in its coding (it is completely self-defined);
2. P-macroblock – A unidirectionally predicted (P) macroblock refers to picture information from one preceding video picture; or
3. B-macroblock – A bi-directional predicted (B) macroblock uses information from one preceding picture and one future video picture.

If all the macroblocks in a video picture are Intra-macroblocks, then the video picture is an Intra-frame. If a video picture only includes unidirectional predicted macro blocks or intra-macroblocks, then the video picture is known as a P-frame. If the video picture contains any bi-directional predicted macroblocks, then the video picture is known as a B-frame. For the simplicity, this document will consider the case where all macroblocks within a given picture are of the same type.

An example sequence of video pictures to be encoded might be represented as

$I_1 B_2 B_3 B_4 P_5 B_6 B_7 B_8 B_9 P_{10} B_{11} P_{12} B_{13} I_{14} \dots$

where the letter (I, P, or B) represents if the video picture is an I-frame, P-frame, or B-frame and the number represents the camera order of the video picture in the sequence of video pictures. The camera order is the order in which a camera

recorded the video pictures and thus is also the order in which the video pictures should be displayed (the display order).

The previous example series of video pictures is graphically illustrated in **Figure 2**. Referring to **Figure 2**, the arrows indicate that macroblocks from a stored picture (I-frame or P-frame in this case) are used in the motion compensated prediction of other pictures.

In the scenario of **Figure 2**, no information from other pictures is used in the encoding of the intra-frame video picture I_1 . Video picture P_5 is a P-frame that uses video information from previous video picture I_1 in its coding such that an arrow is drawn from video picture I_1 to video picture P_5 . Video picture B_2 , video picture B_3 , video picture B_4 all use information from both video picture I_1 and video picture P_5 in their coding such that arrows are drawn from video picture I_1 and video picture P_5 to video picture B_2 , video picture B_3 , and video picture B_4 . As stated above the inter-picture times are, in general, not the same.

Since B-pictures use information from future pictures (pictures that will be displayed later), the transmission order is usually different than the display order. Specifically, video pictures that are needed to construct other video pictures should be transmitted first. For the above sequence, the transmission order might be

$I_1 P_5 B_2 B_3 B_4 P_{10} B_6 B_7 B_8 B_9 P_{12} B_{11} I_{14} B_{13} \dots$

Figure 3 graphically illustrates the above transmission order of the video pictures from Figure 2. Again, the arrows in the figure indicate that macroblocks from a stored video picture (I or P in this case) are used in the motion compensated prediction of other video pictures.

Referring to Figure 3, the system first transmits I-frame I_1 which does not depend on any other frame. Next, the system transmits P-frame video picture P_5 that depends upon video picture I_1 . Next, the system transmits B-frame video picture B_2 after video picture P_5 even though video picture B_2 will be displayed before video picture P_5 . The reason for this is that when it comes time to decode B_2 , the decoder will have already received and stored the information in video pictures I_1 and P_5 necessary to decode video picture B_2 . Similarly, video pictures I_1 and P_5 are ready to be used to decode subsequent video picture B_3 and video picture B_4 . The receiver/decoder reorders the video picture sequence for proper display. In this operation I and P pictures are often referred to as stored pictures.

The coding of the P-frame pictures typically utilizes Motion Compensation, wherein a Motion Vector is computed for each macroblock in the picture. Using the computed motion vector, a prediction macroblock (P-macroblock) can be formed by translation of pixels in the aforementioned previous picture. The difference between the actual macroblock in the P-frame picture and the prediction macroblock is then coded for transmission.

Each motion vector may also be transmitted via predictive coding. For example, a motion vector prediction may be formed using nearby motion

vectors. In such a case, then the difference between the actual motion vector and the motion vector prediction is coded for transmission.

Each B-macroblock uses two motion vectors: a first motion vector referencing the aforementioned previous video picture and a second motion vector referencing the future video picture. From these two motion vectors, two prediction macroblocks are computed. The two predicted macroblocks are then combined together, using some function, to form a final predicted macroblock. As above, the difference between the actual macroblock in the B-frame picture and the final predicted macroblock is then encoded for transmission.

As with P-macroblocks, each motion vector (MV) of a B-macroblock may be transmitted via predictive coding. Specifically, a predicted motion vector is formed using nearby motion vectors. Then, the difference between the actual motion vector and the predicted is coded for transmission.

However, with B-macroblocks the opportunity exists for interpolating motion vectors from motion vectors in the nearest stored picture macroblock. Such interpolation is carried out both in the digital video encoder and the digital video decoder.

This motion vector interpolation works particularly well on video pictures from a video sequence where a camera is slowly panning across a stationary background. In fact, such motion vector interpolation may be good enough to be used alone. Specifically, this means that no differential information

needs be calculated or transmitted for these B-macroblock motion vectors encoded using interpolation.

To illustrate further, in the above scenario let us represent the inter-picture display time between pictures i and j as $D_{i,j}$, i.e., if the display times of the pictures are T_i and T_j , respectively, then

$$D_{i,j} = T_i - T_j \quad \text{from which it follows that}$$

$$D_{i,k} = D_{i,j} + D_{j,k}$$

$$D_{i,k} = -D_{k,i}$$

Note that $D_{i,j}$ may be negative in some cases.

Thus, if $MV_{5,1}$ is a motion vector for a P_5 macroblock as referenced to I_1 , then for the corresponding macroblocks in B_2 , B_3 and B_4 the motion vectors as referenced to I_1 and P_5 , respectively, would be interpolated by

$$MV_{2,1} = MV_{5,1} * D_{2,1} / D_{5,1}$$

$$MV_{5,2} = MV_{5,1} * D_{5,2} / D_{5,1}$$

$$MV_{3,1} = MV_{5,1} * D_{3,1} / D_{5,1}$$

$$MV_{5,3} = MV_{5,1} * D_{5,3} / D_{5,1}$$

$$MV_{4,1} = MV_{5,1} * D_{4,1} / D_{5,1}$$

$$MV_{5,4} = MV_{5,1} * D_{5,4} / D_{5,1}$$

Note that since ratios of display times are used for motion vector prediction, absolute display times are not needed. Thus, relative display times may be used for $D_{i,j}$ display time values.

This scenario may be generalized, as for example in the H.264 standard. In the generalization, a P or B picture may use any previously transmitted picture for its motion vector prediction. Thus, in the above case picture B₃ may use picture I₁ and picture B₂ in its prediction. Moreover, motion vectors may be extrapolated, not just interpolated. Thus, in this case we would have:

$$MV_{3,1} = MV_{2,1} * D_{3,1} / D_{2,1}$$

Such motion vector extrapolation (or interpolation) may also be used in the prediction process for predictive coding of motion vectors.

In any event, the problem in the case of non-uniform inter-picture times is to transmit the relative display time values of D_{i,j} to the receiver, and that is the subject of the present invention. In one embodiment of the present invention, for each picture after the first picture we transmit the display time difference between the current picture and the most recently transmitted stored picture. For error resilience, the transmission could be repeated several times within the picture, e.g., in the so-called slice headers of the MPEG or H.264 standards. If all slice headers are lost, then presumably other pictures that rely on the lost picture for decoding information cannot be decoded either.

Thus, in the above scenario we would transmit the following:

D_{5,1} D_{2,5} D_{3,5} D_{4,5} D_{10,5} D_{6,10} D_{7,10} D_{8,10} D_{9,10} D_{12,10} D_{11,12} D_{14,12} D_{13,14} ...

For the purpose of motion vector estimation, the accuracy requirements for $D_{i,j}$ may vary from picture to picture. For example, if there is only a single B-frame picture B_6 halfway between two P-frame pictures P_5 and P_7 , then it suffices to send only:

$$D_{7,5} = 2 \text{ and } D_{6,7} = -1$$

Where the $D_{i,j}$ display time values are relative time values. If, instead, video picture B_6 is only one quarter the distance between video picture P_5 and video picture P_7 then the appropriate $D_{i,j}$ display time values to send would be:

$$D_{7,5} = 4 \text{ and } D_{6,7} = -1$$

Note that in both of the two preceding examples, the display time between the video picture B_6 and video picture P_7 is being used as the display time "unit" and the display time difference between video picture P_5 and picture video picture P_7 is four display time "units".

In general, motion vector estimation is less complex if divisors are powers of two. This is easily achieved in our embodiment if $D_{i,j}$ (the inter-picture time) between two stored pictures is chosen to be a power of two as graphically illustrated in Figure 4. Alternatively, the estimation procedure could be defined to truncate or round all divisors to a power of two.

In the case where an inter-picture time is to be a power of two, the number of data bits can be reduced if only the integer power (of two) is transmitted instead of the full value of the inter-picture time. Figure 4 graphically illustrates a case wherein the distances between pictures are chosen to be powers of two. In such a case, the $D_{3,1}$ display time value of 2 between video picture P_1 and picture

video picture P_3 is transmitted as 1 (since $2^1 = 2$) and the $D_{7,3}$ display time value of 4 between video picture P_7 and picture video picture P_3 can be transmitted as 2 (since $2^2 = 4$).

In some cases, motion vector interpolation may not be used. However, it is still necessary to transmit the display order of the video pictures to the receiver/player system such that the receiver/player system will display the video pictures in the proper order. In this case, simple signed integer values for $D_{i,j}$ suffice irrespective of the actual display times. In some applications only the sign may be needed.

The inter-picture times $D_{i,j}$ may simply be transmitted as simple signed integer values. However, many methods may be used for encoding the $D_{i,j}$ values to achieve additional compression. For example, a sign bit followed by a variable length coded magnitude is relatively easy to implement and provides coding efficiency.

One such variable length coding system that may be used is known as UVLC (Universal Variable Length Code). The UVLC variable length coding system is given by the code words:

1 =	1
2 =	0 1 0
3 =	0 1 1
4 =	0 0 1 0 0
5 =	0 0 1 0 1
6 =	0 0 1 1 0
7 =	0 0 1 1 1
8 =	0 0 0 1 0 0 0...

Another method of encoding the inter-picture times may be to use arithmetic coding. Typically, arithmetic coding utilizes conditional probabilities to effect a very high compression of the data bits.

Thus, the present invention introduces a simple but powerful method of encoding and transmitting inter-picture display times. The encoding of inter-picture display times can be made very efficient by using variable length coding or arithmetic coding. Furthermore, a desired accuracy can be chosen to meet the needs of the video decoder, but no more.

The foregoing has described a system for specifying variable accuracy inter-picture timing in a multimedia compression and encoding system. It is contemplated that changes and modifications may be made by one of ordinary

skill in the art, to the materials and arrangements of elements of the present invention without departing from the scope of the invention.

We Claim:

1. A method of specifying digital video information, said method comprising:
determining a first order difference between a first video picture and a second video picture; and
encoding said first video picture and said first order difference into a first digital video picture.
2. The method of claim 1 further comprising transmitting said first video picture and said first order difference.
3. The method of claim 2, wherein said second video picture is transmitted before the first video picture .
4. The method of claim 1, wherein said first order difference is stored into a slice header associated with the first video picture.
5. The method of claim 1, wherein said first order difference is stored more than once in said first digital video picture.
6. The method of claim 1, wherein said first order difference comprises a relative time value.
7. The method of claim 1, wherein said first order difference is in a compressed format.
8. The method of claim 7, wherein said first order difference is stored by using a variable length coding.
9. The method of claim 7, wherein said first order difference is stored by using arithmetic coding.
10. The method of claim 1, wherein said first order difference comprises an integer.
11. A bitstream comprising:

a first digital video picture comprising an encoding of a first video picture; and
a first order difference, said first order difference is for specifying a difference
between an order of said first video picture and an order of a second video picture.

12. The bitstream of claim 11, wherein said first order difference is stored within said first digital video picture.

13. The bitstream of claim 11 further comprising a second digital video picture comprising an encoding of the second video picture, wherein said second digital video picture is for transmission before the transmission of the first digital video picture .

14. The bitstream of claim 11, wherein said first order difference is stored into a slice header of said first digital video picture.

15. The bitstream of claim 11, wherein said first order difference is stored more than once in said first digital video picture.

16. The bitstream of claim 11, wherein said first order difference comprises a relative time value.

17. The bitstream of claim 11, wherein said first order difference is in a compressed format.

18. The bitstream of claim 17, wherein said first compressed order difference is stored by using a variable length coding.

19. The bitstream of claim 17, wherein said first compressed order difference is stored by using arithmetic coding.

20. The bitstream of claim 11, wherein said first order difference comprises an integer.

21. A method comprising:

receiving a first picture, a second picture and at least one order value, said order value indicative of a relationship of the second picture with reference to the first picture; and

decoding at least one picture by using the order value.

22. The method of claim 21, wherein the order value represents a time value.

23. The method of claim 21, wherein the order value does not represent a time value.

24. The method of claim 21, wherein decoding at least one picture comprises computing at least one motion vector for the second picture by using the order value.

25. The method of claim 24, wherein computing the motion vector comprises using the order value to compute the motion vector from a motion vector for another picture.

26. The method of claim 25, wherein said another picture is a third picture.

27. The method of claim 25, wherein said another picture is the first picture.

28. The method of claim 21 further comprising outputting the decoded picture based on the order value.

29. The method of claim 21, wherein the order value is in a compressed format.

30. The method of claim 21, wherein the second picture comprises a frame.

31. The method of claim 21, wherein the second picture comprises a field.

32. The method of claim 21, wherein the second picture comprises a slice.

33. The method of claim 32, wherein the order value is stored in a slice header for the slice of the second picture.

34. A computer readable medium storing a computer program that is executable by at least one processor, the computer program comprising sets of instructions for:

receiving a first picture, a second picture and at least one order value, said order value indicative of a relationship of the second picture with reference to the first picture; and

decoding at least one picture by using the order value.

35. The computer readable medium of claim 34, wherein the order value represents a time value.

36. The computer readable medium of claim 34, wherein the order value does not represent a time value.

37. The computer readable medium of claim 34, wherein the set of instructions for decoding at least one picture comprises a set of instructions for computing at least one motion vector for the second picture by using the order value.

38. The computer readable medium of claim 37, wherein the set of instructions for computing the motion vector comprises a set of instructions for using the order value to compute the motion vector from a motion vector for another picture.

39. The computer readable medium of claim 38, wherein said another picture is a third picture.

40. The computer readable medium of claim 38, wherein said another picture is the first picture.

41. The computer readable medium of claim 34, wherein the computer program further comprises a set of instructions for outputting the decoded picture based on the order value.

42. The computer readable medium of claim 34, wherein the order value is in a compressed format.

43. The computer readable medium of claim 34, wherein the second picture comprises a frame.
44. The computer readable medium of claim 34, wherein the second picture comprises a field.
45. The computer readable medium of claim 34, wherein the second picture comprises a slice.
46. The computer readable medium of claim 45, wherein the order value is stored in a slice header for the slice of the second picture.
47. A method comprising:
receiving a first picture, a second picture and at least one order value, said order value indicative of a relationship of the second picture with reference to the first picture;
and
outputting at least one picture based on the order value.
48. The method of claim 47, wherein the order value represents a time value.
49. The method of claim 47, wherein the order value does not represent a time value.
50. The method of claim 47, wherein the order value is stored in a slice header associated with the second picture.
51. A computer readable medium storing a computer program that is executable by at least one processor, the computer program comprising sets of instructions for:
receiving a first picture, a second picture and at least one order value, said order value indicative of a relationship of the second picture with reference to the first picture;
and
outputting at least one picture based on the order value.
52. The computer readable medium of claim 51, wherein the order value represents a time value.

53. The computer readable medium of claim 51, wherein the order value does not represent a time value.
54. The computer readable medium of claim 51, wherein the order value is stored in a slice header associated with the second picture.
55. A method comprising:
receiving a first video picture;
receiving an order value; and
calculating a first motion vector for the first video picture by using said order value.
56. The method of claim 55, wherein said order value relates said first video picture to a second video picture.
57. The method of claim 56, wherein calculating the first motion vector comprises using said order value to perform an interpolation operation on a known second motion vector to calculate the first motion vector.
58. The method of claim 57, wherein the known second motion vector is a motion vector for a third video picture, wherein the third video picture is after the first and second video pictures in an order sequence.
59. The method of claim 56, wherein calculating the first motion vector comprises using said order value to perform an extrapolation operation on a known second motion vector to calculate the first motion vector.
60. The method of claim 59, wherein the known second motion vector is a motion vector for a third video picture, wherein the first video picture is after the second and third video pictures in an order sequence.
61. The method of claim 59, wherein the known second motion vector is a motion vector for a third video picture, wherein the first video picture is before the second and third video pictures in an order sequence.

62. The method of claim 55, wherein said order value is in a compressed format.
63. The method of claim 55 further comprising receiving a second video picture and a third video picture, wherein calculating said first motion vector comprises:
determining (i) a first order difference between an order value for the first video picture and an order value for the second video picture, and (ii) a second order difference between an order value for the third video picture and the order value for the second video picture; and
computing the first motion vector for said first video picture by multiplying a motion vector for the third video picture with a value that is derived from a calculation comprising said first order difference and said second order difference.
64. The method of claim 63, wherein the first video picture is after the second video picture in a sequence of video pictures.
65. The method of claim 64, wherein the sequence is an output order sequence for the video pictures.
66. The method of claim 63, wherein the order values are for indicating an output order sequence for the video pictures, wherein the order value of the first video picture is larger than the order value of the second video picture.
67. The method of claim 55, wherein the first video picture comprises a frame.
68. The method of claim 55, wherein the first video picture comprises a field.
69. The method of claim 55, wherein the first video picture comprises a slice.
70. The method of claim 69, wherein the received order value is for the slice of the first video picture.
71. The method of claim 69, wherein the received order value is stored in a slice header of the slice of the first video picture.

72. A computer readable medium storing a computer program that is executable by at least one processor, the computer program comprising sets of instructions for:
- receiving a first video picture;
 - receiving an order value; and
- calculating a first motion vector for the first video picture by using said order value.
73. The computer readable medium of claim 72, wherein said order value relates said first video picture to a second video picture.
74. The computer readable medium of claim 73, wherein the set of instructions for calculating the first motion vector comprises a set of instructions for using said order value to perform an interpolation operation on a known second motion vector to calculate the first motion vector.
75. The computer readable medium of claim 74, wherein the known second motion vector is a motion vector for a third video picture, wherein the third video picture is after the first and second video pictures in an order sequence.
76. The computer readable medium of claim 73, wherein the set of instructions for calculating the first motion vector comprises a set of instructions for using said order value to perform an extrapolation operation on a known second motion vector to calculate the first motion vector.
77. The computer readable medium of claim 76, wherein the known second motion vector is a motion vector for a third video picture, wherein the first video picture is after the second and third video pictures in an order sequence.
78. The computer readable medium of claim 76, wherein the known second motion vector is a motion vector for a third video picture, wherein the first video picture is before the second and third video pictures in an order sequence.
79. The computer readable medium of claim 72, wherein said order value is in a compressed format.

80. The computer readable medium of claim 72, wherein the computer program further comprises a set of instructions for receiving a second video picture and a third video picture, wherein the set of instructions for calculating said first motion vector comprises a set of instructions for:

determining (i) a first order difference between an order value for the first video picture and an order value for the second video picture, and (ii) a second order difference between an order value for the third video picture and the order value for the second video picture; and

computing the first motion vector for said first video picture by multiplying a motion vector for the third video picture with a value that is derived from a calculation comprising said first order difference and said second order difference.

81. The computer readable medium of claim 80, wherein the first video picture is after the second video picture in a sequence of video pictures.

82. The computer readable medium of claim 81, wherein the sequence is an output order sequence for the video pictures.

83. The computer readable medium of claim 80, wherein the order values are for indicating an output order sequence for the video pictures, wherein the order value of the first video picture is larger than the order value of the second video picture.

84. The computer readable medium of claim 72, wherein the first video picture comprises frame.

85. The computer readable medium of claim 72, wherein the first video picture comprises a field.

86. The computer readable medium of claim 72, wherein the first video picture comprises a slice.

87. The computer readable medium of claim 86, wherein the received order value is for the slice of the first video picture.

88. The computer readable medium of claim 86, wherein the received order value is stored in a slice header of the slice of the first video picture.

89. For a bitstream comprising a first video picture, a second video picture and a third video picture, a method comprising:

determining (i) a first order difference between an order value for the third video picture and an order value for the first video picture, and (ii) a second order difference between an order value for the second video picture and said order value for said first video picture; and

computing a particular motion vector for said second video picture by multiplying a motion vector for the third video picture with a particular value that is derived from a calculation comprising said first order difference and said second order difference.

90. The method of claim 89, wherein computing the particular motion vector for said second video picture comprises using an interpolation operation on the motion vector for the third video picture to compute the particular motion vector.

91. The method of claim 89, wherein computing the particular motion vector for said second video picture comprises using an extrapolation operation on the motion vector for the third video picture to compute the particular motion vector.

92. The method of claim 89, wherein said order value for said second video picture is stored into a slice header that is associated with said second video picture.

93. The method of claim 89, wherein the first video picture comprises a frame.

94. The method of claim 89, wherein the first video picture comprises a field.

95. The method of claim 89, wherein the first video picture comprises a slice.

96. The method of claim 95, wherein the order value is stored in a slice header of the slice of the first video picture.

97. A computer readable medium storing a computer program that is executable by at least one processor, the computer program for computing motion vectors for a stream comprising a first video picture, a second video picture and a third video picture, the computer program comprising sets of instructions for:

determining (i) a first order difference between an order value for the third video picture and an order value for the first video picture, and (ii) a second order difference between an order value for the second video picture and said order value for said first video picture; and

computing a particular motion vector for said second video picture by multiplying a motion vector for the third video picture with a particular value that is derived from a calculation comprising said first order difference and said second order difference.

98. The computer readable medium of claim 97, wherein the set of instructions for computing the particular motion vector for said second video picture comprises a set of instructions for using an interpolation operation on the motion vector for the third video picture to compute the particular motion vector.

99. The computer readable medium of claim 97, wherein the set of instructions for computing the particular motion vector for said second video picture comprises a set of instructions for using an extrapolation operation on the motion vector for the third video picture to compute the particular motion vector.

100. The computer readable medium of claim 97, wherein said order value for said second video picture is stored into a slice header that is associated with said second video picture.

101. The computer readable medium of claim 97, wherein the first video picture comprises a frame.

102. The computer readable medium of claim 97, wherein the first video picture comprises a field.

103. The computer readable medium of claim 97, wherein the first video picture comprises a slice.
104. The computer readable medium of claim 103, wherein the order value is stored in a slice header of the slice of the first video picture.
105. For a bitstream comprising a first video picture, a second video picture, and a third video picture, a method comprising:
- computing a particular value that is based on (i) a first order difference value between an order value for the third video picture and an order value for the first video picture, and (ii) a second order difference value between an order value for the second video picture and the order value for the first video picture; and
 - computing a particular motion vector for the second video picture from the particular value and a motion vector for the third video picture.
106. The method of claim 105, wherein the particular value is proportional to the second order difference value.
107. The method of claim 106, wherein the particular value is inversely proportional to the first order difference value.
108. The method of claim 105, wherein computing the particular value and the particular motion vector are performed during a decoding operation of the second video picture, wherein the first and third video pictures are decoded before the decoding of the second video picture.
109. The method of claim 105, wherein the order value for the second video picture is derived from a value stored in a slice header that is associated with the second video picture.
110. The method of claim 105, wherein the first video picture comprises a frame.
111. The method of claim 105, wherein the first video picture comprises a field.

112. The method of claim 105, wherein the first video picture comprises a slice.
113. A computer readable medium storing a computer program that is executable by at least one processor, the computer program for computing motion vectors for a stream comprising a first video picture, a second video picture, and a third video picture, the computer program comprising sets of instructions for:
- computing a particular value that is based on (i) a first order difference value between an order value for the third video picture and an order value for the first video picture, and (ii) a second order difference value between an order value for the second video picture and the order value for the first video picture; and
 - computing a particular motion vector for the second video picture from the particular value and a motion vector for the third video picture.
114. The computer readable medium of claim 113, wherein the particular value is proportional to the second order difference value.
115. The computer readable medium of claim 114, wherein the particular value is inversely proportional to the first order difference value.
116. The computer readable medium of claim 113, wherein the set of instructions for computing the particular value and the particular motion vector are performed during a decoding operation of the second video picture, wherein the first and third video pictures are decoded video before the decoding of the second video picture.
117. The computer readable medium of claim 113, wherein the order value for the second video picture is derived from a value stored in a slice header that is associated with the second video picture.
118. The computer readable medium of claim 113, wherein the first video picture comprises a frame.
119. The computer readable medium of claim 113, wherein the first video picture comprises a field.

120. The computer readable medium of claim 113, wherein the first video picture comprises a slice.

121. A bitstream comprising:

a first picture;

a second picture; and

an order value specifying a relationship between said second picture and said first picture.

122. The bitstream of claim 121, wherein the order value is for the second picture.

123. The bitstream of claim 121, wherein the first and second pictures are stored in the bitstream in an encoded format.

124. The bitstream of claim 121, wherein said order value is stored more than once in said bitstream.

125. The bitstream of claim 121, wherein said order value is in a compressed format.

126. The bitstream of claim 125, wherein said order value is compressed by using variable length coding.

127. The bitstream of claim 126, wherein a coding length for the order value is based on a conditional probability of the order value.

128. The bitstream of claim 126, wherein the variable length coding comprises universal variable length coding ("UVLC").

129. The bitstream of claim 125, wherein said order value is compressed by using arithmetic coding.

130. The bitstream of claim 121, wherein said order value is for computing a motion vector for the second picture.

131. The bitstream of claim 121, wherein the second picture comprises a frame.
132. The bitstream of claim 121, wherein the second picture comprises a field.
133. The bitstream of claim 121, wherein the second picture comprises a slice.
134. The bitstream of claim 133, wherein the order value is stored in a slice header associated with the slice of the second picture.
135. The bitstream of claim 121, wherein the bitstream is stored on a computer readable medium.
136. A bitstream comprising:
a first video picture and a second video picture; and
an order value for the first video picture for computing at least one motion vector for the first video picture from a motion vector for the second video picture.
137. The bitstream of claim 136 further comprising a third video picture, wherein computing the motion vector for the first video picture comprises:
computing a particular value from (i) a difference between the order value for the first video picture and an order value for the third video picture, and (ii) a difference between an order value for the second video picture and the order value for third video picture; and
multiplying the particular value by the motion vector of the second video picture.
138. The bitstream of claim 136 further comprising a slice header associated with the first video picture, wherein the slice header comprises an encoding of at least one order value for the first video picture.
139. The bitstream of claim 136, wherein the first video picture comprises a frame.
140. The bitstream of claim 136, wherein the first video picture comprises a field.

141. The bitstream of claim 136, wherein the first video picture comprises a slice.
142. The bitstream of claim 136, wherein the bitstream is stored in a computer readable medium.
143. A bitstream comprising:
 - a picture;
 - a slice header associated with the picture; and
 - an order value stored in the slice header.
144. The bitstream of claim 143, wherein the order value is for the picture.
145. The bitstream of claim 143, wherein the order value is stored in the slice header in an encoded format.
146. The bitstream of claim 143, wherein the order value is stored in the slice header in a compressed format.
147. The bitstream of claim 143, wherein said order value is for computing a motion vector for the picture.
148. The bitstream of claim 143, wherein the picture comprises a frame.
149. The bitstream of claim 143, wherein the picture comprises a field.
150. The bitstream of claim 143, wherein the picture comprises a slice.
151. The bitstream of claim 150, wherein the slice header is associated to the slice.
152. A method comprising:
 - determining an order value between a first picture and a second picture; and
 - encoding said order value into a digital bitstream.

153. The method of claim 152, wherein said order value is indicative of a relationship of the first picture with reference to the second picture.
154. The method of claim 152, wherein said order value is stored more than once in the digital bitstream.
155. The method of claim 152, wherein the encoding comprises storing a compressed order value into the digital bitstream.
156. The method of claim 152, wherein the encoding comprises using a variable length coding to compress the order value.
157. The method of claim 156, wherein a coding length of the compressed order value is less than a coding length of another compressed order value.
158. The method of claim 156, wherein a coding length for the compressed order value is based on a conditional probability of the compressed order value.
159. The method of claim 156, wherein the variable length coding comprises universal variable length coding ("UVLC").
160. The method of claim 155, wherein encoding the compressed order value comprises using arithmetic coding to compress the order value.
161. The method of claim 152, wherein said order value comprises an integer.
162. The method of claim 152, wherein said order value is for specifying an order for outputting a sequence of pictures.
163. The method of claim 152, wherein said order value is for computing at least one motion vector for the first picture from a motion vector of a third picture.
164. The method of claim 152, wherein encoding said order value comprises storing said order value in a slice header associated with the first picture.

165. The method of claim 152, wherein the first picture comprises a frame.
166. The method of claim 152, wherein the first picture comprises a field.
167. The method of claim 152, wherein the first picture comprises a slice.
168. The method of claim 167, wherein the order value is stored in a slice header of the slice of the first picture.
169. A computer readable medium storing a computer program that is executable by at least one processor, the computer program comprising sets of instructions for:
determining an order value between a first picture and a second picture; and
encoding said order value into a digital bitstream.
170. The computer readable medium claim 169, wherein said order value is indicative of a relationship of the first picture with reference to the second picture.
171. The computer readable medium of claim 169, wherein the set of instructions for encoding comprises a set of instructions for storing a compressed order value into the digital bitstream.
172. The computer readable medium of claim 169, wherein the set of instructions for encoding comprises a set of instructions for using a variable length coding to compress the order value.
173. The computer readable medium of claim 169, wherein said order value is for specifying an order for outputting a sequence of pictures.
174. The computer readable medium of claim 169, said order value for computing at least one motion vector for the first picture from a motion vector of a third picture.

175. The computer readable medium of claim 169, wherein the set of instructions for encoding said order value comprises a set of instruction for storing said order value in a slice header associated with the first picture.

176. The computer readable medium of claim 169, wherein the first picture comprises a frame

177. The computer readable medium of claim 169, wherein the first picture comprises a field.

178. The computer readable medium of claim 169, wherein the first picture comprises a slice.

179. The computer readable medium of claim 178, wherein the order value is stored in a slice header of the slice of the first picture.