

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 serious 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 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:

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

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 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.

CLAIMS:

1. A storage medium for storing a bitstream, the bitstream comprising:
an encoded first video picture;
an encoded second video picture; and
an integer value that is based on an exponent of a representation of an order value, said order value representative of a position of the second video picture with reference to the first video picture in a sequence of video pictures.
2. The storage medium of claim 1, wherein the exponent is an exponent of a power of two integer.
3. The storage medium of claim 1, wherein the bitstream further comprises a slice header that is associated with said second video picture, said slice header comprising an encoded value that is based on the order value.
4. The storage medium of claim 1, wherein said order value is for computing a motion vector for the second video picture.
5. The storage medium of claim 1, wherein the order value is stored more than once in the bitstream.
6. The bitstream of claim 1, wherein said order value is a compressed order value.

7. The bitstream of claim 6, wherein said compressed order value is compressed in said bitstream by using variable length coding.
8. The bitstream of claim 6, wherein said compressed order value is compressed in said bitstream by using arithmetic coding.
9. The bitstream of claim 1, wherein the order value is representative of an order difference value between said second video picture and said first video picture.
10. A method of encoding a plurality of video pictures, the method comprising:
 - encoding a first video picture;
 - encoding a second video picture; and
 - encoding an integer value that is based on an exponent of a representation of an order value, the order value representative of a position of the second video picture with reference to the first video picture in a sequence of video pictures.
11. The method of claim 10, wherein the said encoded integer value, encoded first video picture and encoded second video picture are stored in a bitstream.
12. The method of claim 10, wherein encoding the second video picture comprises using the order value to compute a motion vector for the second video picture based on a motion vector of another video picture.

13. The method of claim 10, wherein the order value is representative of an order difference value between said second video picture and said first video picture.
14. The method of claim 10 further comprising encoding the order value in a slice header in the bitstream.
15. The method of claim 14, wherein the slice header is associated with the second video picture.
16. The method of claim 10, wherein the exponent is an exponent of a power of two integer.
17. The method of claim 10 further comprising calculating a motion vector for the second video picture by using said order value.
18. The method of claim 17, wherein encoding the second video picture comprises using the calculated motion vector for encoding a macroblock.
19. The method of claim 17 further comprising calculating another motion vector for the second video picture by using said order value.
20. The method of claim 17, wherein calculating the motion vector for the second video picture comprises calculating a particular value that is based on (i) a first order value difference between an order value of a third video picture and an order value of the first video picture, and

(ii) a second order value difference between the order value of the second video picture and the order value of the first video picture.

21. The method of claim 20, wherein calculating the motion vector for the second video picture further comprises multiplying a motion vector for the third video picture with the particular value.

22. The method of claim 17, wherein the calculated motion vector is not encoded in a bitstream.

23. The method of claim 10, wherein said encoded integer value is stored in a slice header associated with the encoded second video picture.

24. A method for decoding a plurality of video pictures, the method comprising:

receiving an encoded first video picture, an encoded second video picture and an integer value for the second video picture, said integer value based on an exponent of a representation of an order value, the order value representative of a position of the second video picture with reference to the first video picture in a sequence of video pictures; and

by a decoder, decoding the second video picture by using the order value.

25. The method of claim 24 further comprising receiving a slice header associated with the second video picture, wherein the slice header comprises an encoded value that is based on the order value.

26. The method of claim 24 further comprising calculating a motion vector for the second video picture by using said order value.
27. The method of claim 26 further comprising calculating another motion vector for the second video picture by using said order value.
28. The method of claim 26, wherein calculating the motion vector for the second video picture comprises calculating a particular value that is based on (i) a first order value difference between an order value of a third video picture and an order value of the first video picture, and (ii) a second order value difference between the order value of the second video picture and the order value of the first video picture.
29. The method of claim 28, wherein calculating the motion vector for the second video picture further comprises multiplying a motion vector for the third video picture with the particular value.
30. The method of claim 24, wherein the exponent is an exponent of a power of two integer.
31. The method of claim 24, wherein the order value is a power of two integer.
32. The method of claim 24, wherein the position of the second video picture with reference to the first video picture is a temporal position.

33. A machine readable storage medium storing a bitstream, the bitstream comprising:
- a plurality of encoded video pictures; and
 - a plurality of encoded order values, wherein a particular order value is a power of two value, wherein an integer value of the power of two value is stored in the bitstream to represent the power of two value.
34. The machine readable storage medium of claim 33, wherein said integer value is stored in a slice header of the bitstream.
35. A method for encoding a plurality of video pictures, the method comprising:
- encoding a plurality of video pictures;
 - encoding a plurality of order values; and
 - storing the encoded video pictures and the encoded order values in a bitstream, wherein a particular order value is a power of two value, wherein an integer value of the power of two value is stored in the bitstream to represent the particular order value.
36. The method of claim 35, wherein encoding the plurality of video pictures comprises using the plurality of order values to calculate a plurality of motion vectors.
37. A method for decoding a plurality of video pictures, the method comprising:
- receiving a bitstream comprising a plurality of encoded video pictures and a plurality of encoded values, each encoded value representing an order value, wherein a particular order value

is a power of two value, wherein an integer value of the power of two value is stored in the bitstream to represent the power of two value; and

encoding the bitstream.

38. The method of claim 37, wherein decoding the bitstream comprises decoding an order value by decoding one of said encoded values using said integer value.

39. The method of claim 38, wherein decoding the bitstream further comprises using the decoded order value to calculate a motion vector.

40. A computer readable medium storing a computer program that is executable by at least one processor, the computer program comprising sets of instructions for implementing the method according to any one of claims 10-23 and 35-36.

41. A computer readable medium storing a computer program that is executable by at least one processor, the computer program comprising sets of instructions for implementing the method according to any one of claims 24-32 and 37-39.

42. A computer system comprising means for implementing steps according to any one of claims 10-23 and 35-36.

43. A computer system comprising means for implementing steps according to any one of claims 24-32 and 37-39.

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

ABSTRACT

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 (105) and a nearby video picture is determined. The display time difference is then encoded (180) 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.

Figure 1