VIDEO ENCODING BY FILTER SELECTION

RELATED APPLICATIONS

[0001] This application claims priority to U.S. Provisional Application 61/078,642, filed on July 7, 2008, the contents of which are hereby incorporated by reference in their entirety.

BACKGROUND

Field of the Invention

[0002] This application relates to video encoding, more specifically, to techniques for interpolating reference frames in video compression systems.

Description of the Related Art

[0003] Digital video capabilities can be incorporated into a wide range of devices, including digital televisions, digital direct broadcast systems, wireless communication devices, personal digital assistants (PDAs), laptop computers, desktop computers, digital cameras, digital recording devices, cellular or satellite radio telephones, and the like. These and other digital video devices can provide significant improvements over conventional analog video systems in creating, modifying, transmitting, storing, recording and playing full motion video sequences.

[0004] A number of different video encoding standards have been established for communicating digital video sequences. The Moving Picture Experts Group (MPEG), for example, has developed a number of standards including MPEG-1, MPEG-2 and MPEG-4. Other standards include ITU H.263, QuickTime™ technology developed by Apple Computer of Cupertino Calif., Video for Windows.™ developed by Microsoft Corporation of Redmond, Wash., Indeo™ developed by Intel Corporation, RealVideo™ from RealNetworks, Inc. of Seattle, Wash., and Cinepak™ developed by SuperMac, Inc. These and other standards, including standards yet to be developed, will continue to evolve.

[0005] Video encoding standards achieve increased transmission rates by encoding data in a compressed fashion. Compression can reduce the overall amount of data that needs to be transmitted for effective transmission of image frames. The MPEG standards, for example, utilize graphics and video compression techniques designed to

facilitate video and image transmission over a narrower bandwidth than could be achieved without the compression. In particular, the MPEG standards incorporate video encoding techniques that utilize similarities between successive image frames, referred to as temporal or inter-frame correlation, to provide inter-frame compression. The inter-frame compression techniques exploit data redundancy across frames by using motion compensated prediction, i.e. by predicting a frame from another after estimating the motion of the scene. In addition, the video encoding techniques may utilize similarities within image frames, referred to as spatial or intra-frame correlation. Frame compression is typically based upon conventional processes for compressing still images, such as discrete cosine transform (DCT) encoding. Discrete cosine transform (DCT) encoding is also used to compress the motion compensated prediction.

[0006] One DCT technique is known as the Adaptive Block Size Discrete Cosine Transform (ABSDCT) method. This technique is disclosed in U.S. Pat. No. 5,021,891, entitled "Adaptive Block Size Image Compression Method And System," assigned to the assignee of the present invention and incorporated herein by reference. DCT techniques are also disclosed in U.S. Pat. No. 5,107,345, entitled "Adaptive Block Size Image Compression Method And System," assigned to the assignee of the present invention and incorporated herein by reference. Further, the use of the ABSDCT technique in combination with a Differential Quadtree Transform technique is discussed in U.S. Pat. No. 5,452,104, entitled "Adaptive Block Size Image Compression Method And System," also assigned to the assignee of the present invention and incorporated herein by reference. The systems disclosed in these patents utilize what is referred to as "intra-frame" encoding, where each frame of image data is encoded without regard to the content of any other frame. Using the ABSDCT technique, the achievable data rate may be reduced from around 1.5 billion bits per second to approximately 50 million bits per second without discernible degradation of the image quality.

[0007] The ABSDCT technique may be used to compress either a black and white or a color image or signal representing the image. The color input signal may be in a YIQ format, with Y being the luminance, or brightness, sample, and I and Q being the chrominance, or color, samples for each 4.times.4 block of pixels. Other known formats such as the YUV, YC.sub.bC.sub.y or RGB formats may also be used. Because of the low spatial sensitivity of the eye to color, most research has shown that a sub-sample of the color components by a factor of four in the horizontal and vertical directions is

reasonable. Accordingly, a video signal may be represented by four luminance components and two chrominance components.

[0008] To support the compression techniques, many digital video devices include an encoder for compressing digital video sequences, and a decoder for decompressing the digital video sequences. In many cases, the encoder and decoder comprise an integrated encoder/decoder (CODEC) that operates on blocks of pixels within frames that define the sequence of video images. In the MPEG-4 standard, for example, the encoder of a sending device typically divides a video image frame to be transmitted into macroblocks comprising smaller image blocks. For each macroblock in the image frame, the encoder searches macroblocks of the immediately preceding video frame to identify the most similar macroblock, and encodes the difference between the macroblocks for transmission, along with a motion vector that indicates which macroblock from the previous frame was used for encoding. The decoder of a receiving device receives the motion vector and encoded differences, and performs motion compensation to generate video sequences. Motion vectors may have full, half or quarter pixel precisions, depending on the level of precision selected by an encoder. When motion vectors with fractional pixel values are used, a better prediction block is obtained. Interpolation is carried out to determine the values of the fractional pixels (sub-pixels). In one example, pixel values include bits representing the intensity of a luminance, chrominance, or color component.

[0009] The video encoding process is computationally intensive. In particular, the process of comparing video blocks to previously transmitted video blocks requires large numbers of computations. Improved encoding techniques are highly desirable, particularly for use in wireless devices or other portable video devices where computational resources are more limited and power consumption is a concern.

SUMMARY

[0010] One embodiment is a method of encoding digital video, the method including; identifying a reference frame and a current frame, interpolating the reference frame using a default filter to create a default interpolated frame, interpolating the reference frame using an alternative filter to create an alternative interpolated frame, determining for each sub-pixel position whether to use the default filter or the alternative filter based on a lower sum error to generate a final reference frame.

[0011] Another embodiment is a video encoding device, including; a motion compensation unit configured to interpolate a first reference frame using a default filter to create a default interpolated frame and configured to interpolate a second reference frame using an alternative filter to create an alternative interpolated frame, a reference frame store configured to store the default interpolated frame and store the alternative interpolated frame, a filter decision unit configured to receive a current frame, receive the default interpolated frame, receive the alternative interpolated frame, and determine, for each full and sub-pixel position, whether to use a default filter or an alternative filter in a final frame; and generate the final frame.

[0012] Still anther embodiment is a video encoding device, including; a means for receiving a current frame, a means for storing a plurality of interpolated reference frames in a reference frame store, a means for receiving the plurality of interpolated reference frames from the reference frame store, a means for producing a final frame, and a means for determining for each full and sub pixel position in the final frame whether to use a default filter or an alternative filter.

[0013] One other embodiment is a video encoding device, including: a motion compensation unit configured to interpolate a first reference frame using a default filter to create a default interpolated frame and configured to interpolate a second reference frame using an alternative filter to create an alternative interpolated frame, a filter decision unit configured to receive a current frame, receive the default interpolated frame, receive the alternative interpolated frame, determine, for each full and sub-pixel position, whether to use a default filter or an alternative filter in a final frame, and generate the final frame.

[0014] Another embodiment is a video encoding device, including; a means for receiving a current frame, a means for receiving a plurality of interpolated reference frames, a means for producing a final frame, and means for determining for each full

and sub pixel position in the final frame whether to use a default filter or an alternative filter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] The foregoing aspects and many of the attended advantages of this application are illustrated by way of example, and not by way of limitation, in the accompanying drawings, wherein;

[0016] FIG. 1 is a block diagram illustrating a video encoding and decoding system.

[0017] FIG. 2 is a block diagram illustrating a video encoder.

[0018] FIG. 3 is a block diagram illustrating a reference frame store and a filter decision unit.

[0019] FIG. 4 is a flow diagram describing the process of determining best combination of filters.

[0020] FIG. 5 is a block diagram of a video frame showing fractional pixel positions for interpolation.

[0021] FIG. 6 is a block diagram illustrating video motion estimation.

[0022] FIG. 7 is a block diagram illustrating a video frame including an intra frame, a predicted frame and a bi-directional frame.

[0023] FIG. 8 is block diagram illustrating a video frame comprising a B slice comprising two motion vectors.

[0024] FIG. 9 is a flow diagram describing the process of creating a final reference frame.

DETAILED DESCRIPTION

[0025] Specific embodiments of the application will now be described with reference to the drawings. The terminology used in the description presented herein is not intended to be interpreted in any limited or restrictive manner, simply because it is being utilized in conjunction with a detailed description of certain specific embodiments of the application.

[0026] In one embodiment, the system interpolates a given block or frame of video using a plurality of filters to create multiple reference frames. For each filter, the system then calculates motion vectors from the given reference frame. A calculation is then made to determine, for each pixel, which filter provided the lowest error and distortion rate. In some embodiments, the expected performance of one or a

combination of filters for each sub-pixel and integer pixel within a video frame is examined. Then, the best filter for encoding the sub-pixel and integer pixel values is selected for each position in the video frame.

[0027] In one embodiment, the system accumulates the sum of absolute difference (SAD) error measurements for all available positions in a single frame, using a plurality of filters. By measuring the SAD, the system can determine whether it's advantageous to use an adaptive filter (for example, an adaptive interpolation filter (AIF)) corresponding to that position or whether it may be more convenient (in terms of SAD and weighted bit cost) to use a default interpolation filter, or use no filter at all to encode the pixel values.

[0028] In another embodiment, a process of using optimal selection criteria for a combination of available filters leading to a minimal computational cost (or SAD plus weighted encoding bits used by non-default interpolation filters) is used. Given a matrix and a vector, it may be possible to computer the error corresponding to each of the decision vectors and select the filter for each sub-pixel position indicating the minimal computational cost.

[0029] In still another embodiment, an iterative process is utilized. This process includes starting from a possible solution (any valid decision vector) and finding the position for which an alternative filter provides the biggest error decrease. This new position is then updated with the value of the new filter choice and the process is repeated until no more improvements are possible, the improvements are negligible (below a predetermined threshold), or a maximum number of iterations is reached.

[0030] In still another embodiment, a simplified heuristic is used to find an approximation of the minimal cost using symmetry classes. This method considers only error contributions in which the positions of two references blocks use the same filer. This happens when the positions are identical or when they belong to the symmetry class.

[0031] In another embodiment, a quantization table is used that improves the encoding of the filter coefficients used on the full-pixel positions, wherein the quantization precision depends on the coefficient location using an uneven quantization process.

[0032] Block matching is the most common method of motion estimation. Typically each macroblock (i.e. 16×16 pixels) in a video frame is compared with shifted regions

of the same size from a previous frame, and the shift which results in the minimum error is selected as the best motion vector for that macroblock.

[0033] Intra-frames (or I-frames) are frames coded without reference to any other frame, and may be generated by an encoder to create a random access point. Intra-frames typically require more bits to encode than other picture types. Often, I-frames are used for random access and are used as references for decoding of other frames.

[0034] Predicted frames (or P frames) require the prior decoding of some other frames(s) in order to be decoded and may contain both image data and motion vector displacements. In H.264, predicted frames can use multiple previously-decoded frames as references during decoding, and may have any arbitrary display-order relationship relative to the frames(s) used for its prediction. Also, typically, predicted frames require fewer bits for encoding than intra-frames do.

[0035] Bi-directional predicted frames (or B frames) require the prior decoding of some other frame(s) in order to be decoded, may contain both image data and motion vector displacements, and include some prediction modes that form a prediction by averaging the predictions obtained using two different previously-decoded reference regions. In H.264, bi-directional predicted frames may or may not be used as references for the decoding of other pictures. B frames typically require fewer bits for encoding than either intra-frames or predicted frames do.

[0036] FIG. 1 is a block diagram illustrating a video encoding and decoding system 10. As shown in FIG. 1, system 10 includes a source device 12 that transmits encoded video to a receive device 14 via a communication channel 16. Source device 12 may include a video source 18, video encoder 20 and a transmitter 22. The transmitter 22 may be a wired/wireless transmitter. Receive device 14 may include a receiver 24, video decoder 26 and video display device 28. In some aspects, receiver 24 may be a wireless receiver, such as a wireless receiver in a wireless communication device handset. System 10 may be configured to support alternative filtering (or adaptive filtering) of motion compensated prediction blocks to improve both visual quality and processing efficiency.

[0037] In the example of FIG. 1, communication channel 16 may comprise any wireless or wired communication medium, such as a radio frequency (RF) spectrum or one or more physical transmission lines, or any combination of wireless and wired media. Channel 16 may form part of a packet-based network, such as a local area

network, wide-area network, or a global network such as the Internet. Communication channel 16 generally represents any suitable communication medium, or collection of different communication media, for transmitting video data from source device 12 to receive device 14.

[0038] Source device 12 generates video for transmission to destination device 14. In some cases, however, devices 12, 14 may operate in a substantially symmetrical manner. For example, each of devices 12, 14 may include video encoding and decoding components. Hence, system 10 may support one-way or two-way video transmission between video devices 12, 14, e.g., for video streaming, video broadcasting, or video telephony.

[0039] Video source 18 may include a video capture device, such as one or more video cameras, a video archive containing previously captured video, or a live video feed from a video content provider. As a further alternative, video source 18 may generate computer graphics-based data as the source video, or a combination of live video and computer-generated video. In some cases, if video source 18 is a camera, source device 12 and receive device 14 may form so-called camera phones or video phones, including satellite or mobile wireless telephones, or other wireless communication devices. Hence, in some aspects, the techniques described in this application may be implemented within a mobile wireless communication device handset, such as a mobile telephone handset. In each case, the captured, pre-captured or computer-generated video may be encoded by video encoder 20 for transmission from video source device 12 to video decoder 26 of video receive device 14 via transmitter 22, channel 16 and receiver 24. Display device 28 may include any of a variety of display devices such as a liquid crystal display (LCD), plasma display or organic light emitting diode (OLED) display.

[0040] Video encoder 20 and video decoder 26 may be configured, in some aspects of this application, to support scalable video coding for spatial, temporal and/or signal-to-noise ratio (SNR) scalability. Encoder 20 and decoder 26 may support various degrees of scalability by supporting encoding, transmission and decoding of a base layer and one or more scalable enhancement layers. For scalable coding, a base layer carries video data with a minimum level of quality. One or more enhancement layers carry additional bitstream to support higher spatial, temporal or SNR levels.

[0041] Video encoder 20 and video decoder 26 may operate according to a video compression standard, such as MPEG-2, MPEG-4, ITU-T H.263, or ITU-T H.264/MPEG-4, Part 10 (AVC). Although not shown in FIG. 1, in some aspects, video encoder 20 and video decoder 22 may be integrated with an audio encoder and decoder, respectively, and include appropriate MUX-DEMUX units, or other hardware and software, to handle encoding of both audio and video in a common data stream or separate data streams. If applicable, MUX-DEMUX units may conform to the ITU H.223 multiplexer protocol, or other protocols such as the user datagram protocol (UDP).

[0042] The H.264 standard was developed by the ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group (MPEG), as the product of a partnership known as the Joint Video Team (JVT). The H.264 standard is described in ITU-T Recommendation H.264, Advanced Video coding for Generic Audiovisual Services, dated March 2005, which may be referred to herein as the H.264 standard or H.264 specification, or the H.264/AVC standard or specification. In some aspects, techniques described in this application may be applied to devices that generally conform to the H.264 standard, or other devices that do not generally conform to the H.264 standard.

[0043] The Joint Video Team (JVT) continues to work on a scalable video coding (SVC) extension to H.264/MPEG-4 AVC. The specification of both H.264/MPEG-4AVC and the evolving SVC extension are in the form of a Joint Draft (JD). The Joint Scalable Video Model (JSVM) created by the JVT implements tools for use in scalable video, which may be used within system 10 for various coding tasks described in this application.

[0044] In some aspects, for video broadcasting, embodiments relate to an application to Enhanced H.264 video coding for delivering real-time video services in terrestrial mobile multimedia multicast (TM3) systems using the Forward Link Only (FLO) Air Interface Specification, "Forward Link Only Air Interface Specification for Terrestrial Mobile Multimedia Multicast," to be published as Technical Standard TIA-1099 (the "FLO Specification"). The FLO Specification includes examples defining bitstream syntax and semantics and decoding processes suitable for the FLO Air Interface. Alternatively, video may be broadcasted according to other standards such as DVB-H (digital video broadcast-handheld), ISDB-T (integrated services digital broadcast-

terrestrial), or DMB (digital media broadcast). Hence, in some cases, source device 12 may be a mobile wireless terminal, such as a wireless communication device handset, a video streaming server, or a video broadcast server. However, the techniques described in this application are not limited to any particular type of broadcast, multicast, or point-to-point system.

[0045] Video encoder 20 and video decoder 26 each may be implemented as one or more microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), discrete logic, software, hardware, firmware or any combinations thereof. Hence, the techniques described herein may be implemented within one or more integrated circuit devices, which may be referred to collectively as an integrated circuit device. Such an integrated circuit device may be provided within a communication device, such as a wireless communication device handset. Each of video encoder 20 and video decoder 26 may be included in one or more encoders or decoders, either of which may be integrated as part of a combined encoder/decoder (CODEC) in a respective mobile device, subscriber device, broadcast device, server, or the like. In addition, video source device 12 and video receive device 14 each may include appropriate modulation, demodulation, frequency conversion, filtering, and amplifier components for transmission and reception of encoded video, as applicable, including radio frequency (RF) wireless components and antennas sufficient to support wireless communication. For ease of illustration, however, such components are not shown in FIG. 1.

[0046] As mentioned above, videos includes a series of video frames. Video encoder 20 operates on blocks of pixels within individual video frames in order to encode the video data. The video blocks may have fixed or varying sizes, and may differ in size according to a specified coding standard. As an example, the ITU-T H.264 standard supports intra prediction in various block sizes, such as 16x16, 8x8, 4x4 for luma components, and 8x8 for chroma components. The ITU-T H.264 standard also supports, inter prediction in various block sizes, such as 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4 for luma components and corresponding scaled sizes for chroma components. Smaller video blocks can provide better resolution, and may be used for locations of a video frame that include higher levels of detail. In general, macroblocks and the various smaller blocks may be considered to be video blocks. In some cases, the smaller blocks may be referred to as sub-blocks. After prediction, a transform may be

performed on the 8x8 residual block or 4x4 residual block. Also, an additional transform may be applied to the DC coefficients of the 4x4 blocks for chroma components or luma component if the intra sub -- 16x16 prediction mode is used.

[0047] FIG. 2 is a block diagram illustrating more details of the video encoder 20. Video encoder 20 may perform intra- and inter-coding of blocks within video frames. Intra-coding relies on spatial prediction to reduce or remove spatial redundancy in a given video frame. Inter-coding relies on temporal prediction to reduce, or remove, temporal redundancy in video within adjacent frames. For inter-coding, video encoder 20 performs motion estimation to track the movement of matching video blocks between two or more adjacent frames.

[0048] As shown in FIG. 2, video encoder 20 receives a current video block 21 within a video frame to be encoded. In the example of FIG. 2, video encoder 20 includes motion estimation unit 23, reference frame store 25, motion compensation unit 27, block transform unit 29, quantization unit 31, inverse quantization unit 33, inverse transform unit 35 and entropy coding unit 37. Video encoder 20 also includes summer 39 and summer 41. FIG. 2 illustrates the temporal prediction components of video encoder 20 for inter-coding of video blocks. Although not shown in FIG. 2 for ease of illustration, video encoder 20 also may include spatial prediction components for intra-coding of some video blocks.

[0049] Motion estimation unit 23 compares video block 21 to blocks in one or more adjacent video frames to generate one or more motion vectors. The adjacent frame or frames may be retrieved from reference frame store 25. Motion estimation may be performed for blocks of variable sizes, e.g., 16x16, 16x8, 8x16, 8x8 or smaller block sizes. Motion estimation unit 23 identifies a block in an adjacent frame that most closely matches the current video block 21, e.g., based on a rate distortion model, and determines a displacement between the blocks. On this basis, motion estimation unit 23 produces a motion vector that indicates the magnitude and trajectory of the displacement.

[0050] Motion vectors may have half-pixel or quarter-pixel precision, or even finer precision, allowing video encoder 20 to track motion with higher precision than integer pixel locations and obtain a better prediction block. When motion vectors with fractional pixel values are used, interpolation operations may be carried out in motion compensation unit 27. For example, in the H.264/AVC standard, to obtain a luma signal

at half-pixel positions, the 6-tap Wiener filter with coefficients (1, -5, 20, 20, -5, 1)/32 may be used. To obtain luma signals at quarter-pixel locations, bilinear filtering on the values at integer pixel locations and the interpolated values at half pixel locations may be used. The bilinear filter also may be used in fractional pixel interpolation for the chroma components, which may have up to 1/8-pixel precision.

[0051] Motion estimation unit 23 identifies the best motion vector for a video block using a rate-distortion model. Using the resulting motion vector, motion compensation unit 27 forms a prediction video block by motion compensation. Video encoder 20 forms a residual video block by subtracting the prediction video block produced by motion compensation unit 27 from the original, current video block 21 at summer 39. Block transform unit 29 applies a transform to the residual block. Quantization unit 31 quantizes the transform coefficients to further reduce bit rate. Entropy coding unit 37 entropy codes the quantized coefficients to even further reduce bit rate. Video decoder 26 performs inverse operations to reconstruct the encoded video.

[0052] Inverse quantization unit 33 and inverse transform unit 35 apply inverse quantization and inverse transformation, respectively, to reconstruct the residual block. Summer 41 adds the reconstructed residual block to the motion compensated prediction block produced by motion compensation unit 27 to produce a reconstructed video block (or a filter decision for a current frame) for storage in reference frame store 25. The motion compensation unit comprising a filter decision unit 44 configured to receive reference frames from the reference frame store 25, receive a current frame, and produce a reconstructed video block to be stored in the filter reference frame store 25. The reconstructed video block may be used by motion estimation unit 23 and motion compensation unit 27 to encode a block in a subsequent video frame.

[0053] When performing motion compensation for a given block in the current video frame 21, motion compensation unit 27 may use a fixed set of filters to interpolate a reference block from a reference frame. One reference block is needed if the current block is uni-directionally predicted or two reference blocks are needed if the current block is bi-directionally predicted. In H.264, multiple reference frames in forward and backward directions may be used in some cases. The actual filters used by motion compensation unit 27 depend on the fractional part of the motion vector. For example, if the motion vector points to a half-pixel location in the reference frame in a given dimension, to obtain the value of the half-pixel location, a default interpolation filter

may be used in that dimension with a half-pixel motion vector. If both motion vector components point to integer locations, the pixel values from the reference frame in reference frame store 25 may be used directly without performing any interpolation filtering operation.

[0054] FIG. 3 is a block diagram illustrating a reference frame store 25 and a filter decision unit 44 as shown in FIG. 2. The illustrative block diagram of the reference frame store 25 and filter decision union 44 of FIG. 3 includes a current frame 30, a reference frame store 25, and a filter decision unit 44. The current frame 30 may be transmitted to the filter decision unit 44 to be used to create a final (or reconstructed) reference frame to be stored in the reference frame store 25. The reference frames store 25 may maintain a set of reference frames and transmit the maintained reference frames to the motion compensation unit 27. In one embodiment, a set of reference frames interpolated with default filers 32, 34, and 36 are stored in the reference frame store 25 and are sent to the filter decision unit 44. In another embodiment, a set of reference frames interpolated with alternative filters (or non-default filters) 38, 420, 42 are stored in the reference frame store 25 and sent to the filter decision unit 44. In another embodiment, if memory capacity is limited, interpolated frames may be generated on-the-fly, as needed. In one embodiment, the interpolated frames are generated in real-time as long as the original (non-interpolated) frames, the interpolating filters and the filter choices are available. The alternative filter (not shown) residing in a video encoder 20 may be an adaptive interpolation filter, a separable adaptive interpolation filter, a fixed filter. In another embodiment, an alternative filter may an adaptive interpolation filter (AIF) that offers large coding gains, especially on video sequences with high resolution (e.g. 720p). The video encoder 20 has a fixed set of alternative filters to interpolate a block in a reference frame. In yet another embodiment, the video encoder 20 may support a 2D non-separable adaptive interpolated filter, a separable adaptive interpolated filter, a directional adaptive interpolated filter, and an enhanced adaptive interpolated filter. A 2D non-separable adaptive interpolation filter uses a two dimensional matrix to interpolate frames. A separable adaptive interpolation filter uses vertical then horizontal scans to interpolate frames. A directional adaptive interpolation filter uses diagonal scans to interpolate frames.

[0055] In another embodiment, an enhanced adaptive interpolation filter (E-AIF) may be used by adding a full-pixel position filter and a filter offset to each full- and sub-

pixel position. For the full and each sub-pixel position E-AIF also estimates the expected rate distortion gain of the alternative filter over the default interpolation and decides whether to use and signal to the decoder the alternative filter or not.

[0056] The filter decision unit 44 may compare the sum bit overhead (or cost) of encoding the received current frame 30 using reference frames interpolated with default filter 32, 34, 36 and reference frames interpolated with alternative filters 38, 40, 42. In one embodiment, the filter decision unit 44 may uses a scaling factor (not shown) to reconstruct the pixel positions to determines the absolute error corresponding to the use of either the default filters or the alternative filters. In another embodiment, the filter decision unit 44 may compare the absolute error value for each pixel, select between available filter, and produces a decision vector that minimizes bit error rate and distortion.

[0057] The filter decision unit 44 may also receive motion vectors (or motion vector data) 48. In one embodiment, the filter decision unit may receive the current frame 30, the motion vectors 48, interpolated reference frames, such as reference frames interpolated with a default filter 32, 34, 36 and reference frames interpolated with an alternative filter 38, 40, 42 in order to create a final (or reconstructed) reference frames 45, 46, 47 to be stored in the reference frame store 25. As a result, the filter decision unit 44 provides interpolated reference frames 45, 46, and 47, wherein each full or sub-pixel is interpolated using either a default or alternative filter. In another embodiment, the filter decision unit 44 may output to the reference frame store 25 a decision array (not shown). The decisions array (not shown) may provide decision values describing the reconstructed reference frames 45, 46, 47. In one embodiment, the decision values provided by the decision array (not shown) may be, for example, [a (1), b (0), c (1), d (0)…]; wherein a sub-pixel position is identified by an alpha character and a filter selection decision is identified in binary (0 or 1) format as either using the default or the alternative filter available the video encoder 20. In another embodiment, the decision values provided by the decision array may be for example, [a(1), b(2), c(3), d(4), …], wherein a sub-pixel position is identified by an alpha character and a filter selection decision is identified using a number format used to identify the actual filter type to be used per sub-pixel position. In one embodiment, multiple default and multiple adaptive interpolation filters may be compared against one another in order to determine the most efficient filter for encoding each sub-pixel position. In another embodiment, multiple

adaptive interpolation filters are compared against one another in order to determine the most efficient adaptive filter to be applied during encoding for each sub-pixel position.

[0058] FIG. 4 depicts an illustrative process 50 for determining a best combination of filters using the system shown in FIG. 3. The process begins at block 51 with the video encoder 20 calculating motion vectors of frames of reference to a current frame. The reference frames may be an intra-frame or predicted frames. The current frame 30 may be a bi-directional predicted frame. In one embodiment, a current frame references a previous intra-frame and a limited search using block matching is applied to calculate a motion vector describing the previous displacement. In another embodiment, a current frame references a future predicted frame and a limited search using block matching is applied to calculate a motion vector describing the predicted displacement. After motion vector data is calculated, it is transmitted to the filter decision unit 44 as described in FIG. 3.

[0059] The process 50 then moves to block 52 and calculating coefficients of alternative filters. In one embodiment, an adaptive interpolation filter is selected and coefficients, typically two vectors comprising a coefficient vector and an offset vector are sent. For example, a coefficient vector comprising six coefficients may be [1,0,1,0,1,1,], and corresponding offset vector [-1, 0, 2, 1, 3, 0]. One offset value is sent for each sub-pixel position, plus one offset for the full pixel position. So, there may be two vectors, one for the filter selection, and another, having same dimension, for the offsets. In another embodiment, a set of fixed, pre-calculated filters are used instead of determining adaptively the coefficients for each frame.

[0060] The process 50 then moves to block 53 and reference frames using alternative filters are generated. In one embodiment, a reference frame using an enhanced adaptive interpolation filter is generated and stored in the reference frame store 25. The reference frame may be interpolated using an adaptive interpolation filter, an enhanced adaptive interpolation filter, or a combination of non-default filters.

[0061] The process 50 then moves to block 54 and computing an error for each block with every combination of filters available. In one embodiment, a current frame 30 is compared with an interpolated reference frame using an alternative interpolation filter and the rate and distortion error is determined. In another embodiment, a current frame 30 is compared with an interpolated reference frame using a default H.264 filter, and the rate and distortion error is determined.

[0062] The process 50 then moves to block 55 and determining a cost of encoding filters coefficients. In one embodiment, a filter decision unit 44 of FIG. 3 determines a cost for each sub pixel position of encoding filter coefficients using a default interpolation filter verses using an alternative interpolation filter, wherein the cost is determined in bit overhead.

[0063] If some positions are considered symmetric, a single filter can be determined and chosen from more than one sub-pixel position. For example, sub-pixel position 'a' and 'c' could use filters that are mirror images of the other.

[0064] The process 50 then moves to block 56 and determines a best combination of filters. In one embodiment, for each full and sub pixel position in the final frame, a determination is made as to whether to use a default filter or an alternative filter based on the filter having a lower error and a low distortion rate when compared.

[0065] The mathematical computations used to determine the distortion rate per pixel are described below.

SAD_{Std}, the sum of absolute differences of the motion-compensated prediction error based on the default interpolation filter is calculated;

SAD_{AIF}, the motion-compensated prediction error based on the alternative interpolation filters is calculated;

$\lambda \cdot nBits$, the cost (in bit) of the side information necessary to encode the filters coefficients is calculated, multiplied by a scaling factor lambda and added to SAD_{AIF};

Finally, SAD_{Std} and $SAD_{AIF} + \lambda \cdot nBits$ are compared and the lowest value used to determine whether the alternative or the default interpolation filters should be used.

[0066] In one embodiment, if the filter selection uses predetermined and pre-calculated filters, and the encoder receives only the information related to the filter selection. For example, using the iterative method that finds a cost that is a local minimum approximation to the optimal cost, no side information, such as the cost (in bits) needs to be transmitted.

[0067] FIG. 5 is a block diagram of a video frame showing fractional pixel positions for interpolation. Also, FIG. 5 illustrates a set of fractional pixel (or sub-pixel) positions in a video frames. In FIG. 5, there are altogether 15 sub-pixel positions, labeled "a"

through "o" in FIG. 5. Moreover, this block diagram may be used to provide symmetry data and may lead to reduces computation overhead. For example, the ½-pixel positions 'a' 61 & 'c' 63 may be symmetric, reducing computational overhead involved in the interpolation processing. In yet another example, sub-pixel position 'C3' 65 may be symmetric to sub-pixel 'C4' 69 resulting in reduce computational overhead during interpolation processing.

[0068] A simplified heuristic method considers only error contributions in which the positions of the two reference blocks use the same filter. This happens when the positions are identical or when they belong to the same symmetry class (for example *e*, *g*, *m* and *o* in FIG. 5).

[0069] With this approach, it is possible to accumulate the error in a vector $C_{d_i,i}$ and the problem can be stated as:

$$\min_{D=(d_1,d_2,\ldots,d_n)} \sum_{i=1}^{n} \left(C_{d_i,i} + d_i \cdot \lambda \cdot nBits_i \right)$$

[0070] Due to the independence of the decisions, the solution can be found by minimizing each d_i independently.

[0071] FIG. 6 is a block diagram illustrating motion estimation. In general, motion estimation is used to determine motion vectors for each macroblock in a current frame as compared to a reference frame. The process of calculating a motion vector comprises executing a search algorithm that determines a best match between a macroblock in the current frame and a block of pixels from a reference frame. The search algorithm is generally performed using a Sum of Absolute Difference (SAD) computation, which is well known in the art. In particular, the motion estimation compares each macroblock in the current frame with its surrounding area in the previous frame and attempts to find a match indicated by the minimum SAD.

[0072] Moreover, FIG. 6 illustrates a relationship between a sequential set of video frames. FIG. 6 depicts an intra-frame (or I-frame) 60 as a reference frame and a predicted frame (or P frame) 62 as a current frame. The predicted frame (P frame) 62 defining a first macroblock 66 and performing motion estimation by defining a second macroblock 64 in the Intra-frame 60 (or reference frame). The pixels within the first macroblock 66 and the second macroblock 64 are compared and the sum absolute difference (SAD) is calculated, and a motion vector 68 is determined as a result.

[0073] FIG. 7 is a block diagram illustrating a video frame including an intra-frame, predicted frames, and bi-directional predicted frames. As illustrated, the intra-frame 70 does not reference any other frame, except itself. In one embodiment, an intra-frame is a first frame received in a sequence of frames. Generally, intra-frames do not reference any preceding frames and do not have the capability to reference frames in the future. On the other hand, predicted frames may reference previous intra-frames or future predicted frames. As shown in the diagram, the first predicted-frame 76 references the intra-frame 70 and does not reference the first B frame 72 or the second B frame 74. B frames, typically reference either intra-frames or predicted frames, both in the past and in the future. As shown in FIG. 7, a first B frame 72 references both the first intra-frame 70 and the first predicted frame 76 to determine the best motion vector. As shown in FIG. 7, a second B frame 74 references both a first intra-frame 70 and a first predicted frame 76.

[0074] FIG. 8 is block diagram illustrating a video frame comprising a bi-directional predicted slice comprising two motion vectors. In one embodiment, a first bi-directional predicted frame 72 defining a first bi-directional predicted macroblock 814 having a size of 16x16 referencing an intra frame 70 (a preceding frame) defining a second intra-frame macroblock 806 having a size of 16x16 and a first predicted frame 76 (a future frame) defining a third predicted macroblock 810 having a size of 16x16. In another embodiment, the reference frames may also be two bi-directional predicted frames, two predicted frames, or a combination of a predicted frame and a bi-directional predicted frame. The first bi-directional predicted frame 72 calculating and identifying two motion vectors, a first intra motion vector 820 and a second predicted motion vector 822 as illustrated in FIG. 8. The intra frame 70 identifying a first intra-pixel 808 having the smallest sum of absolute difference value with reference with the first bi-directional predicted frame 72. As shown in FIG. 8, the first predicted frame 76 identifying a first predicted pixel 812 having the smallest sum of absolute difference value with reference with the first predicted frame 72.

[0075] Upon identification of a first intra pixel 808 and the first predicted pixel 812, a matrix is then constructed, as described below; wherein 'a' represents the first intra pixel 808 and 'b' represents the first predicted pixel 812.

```
matrix C =    0, 0, a, b
              0, 1, a, b
              1, 0, a, b
```

```
1, 1, a, b
```

Matrix C contains errors accumulated during the prediction, wherein 'a' and 'b' are sub-pixel positions. So, in Fig 8, the first intra-motion vector 820 points to a sub-pixel position 'a' and the second predicted motion vector 822 points to a sub-pixel position 'b'. To matrix C = [0,0,a,b] the encoder will add the prediction error computed by applying default filters (choice 0) to both the first intra-pixel 808 and the first predicted pixel 812. To matrix C = [1,0,a,b] the encoder will be updated by adding the prediction error obtained by interpolating the first intra-pixel 808 with the adaptive filter and first predicted pixel 812 with the default filter (choice 1 0). To matrix C = [0,1,a,b] the encoder will add the prediction error obtained by interpolating the first intra-pixel 808 with a default filter and the first predicted pixel 812 with the adaptive filter (choice 0 1) . Finally, to matrix C = [1,1,a,b] encoder will be updated with the prediction error obtained from the interpolation of both pixels with an adaptive filter (choice 1 1).

[0076] For each full and sub-pixel position, a decision vector $D = (d_1, d_2, ..., d_n)$, with $d_i \in \{0,1\}$ and $i = 1,...,n$ can be computed.

[0077] A filter flag of '1' or '0' for each sub-pixel position representing a unique set of filter coefficients sent to indicate whether a default or an alternative filter for the sub-pixel position is used. If the filter flag is set to'1', then the alternative filter coefficients for this sub-pixel position are predicted, quantized, coded and applied to corresponding sub-pixel positions. If the filter flag is set to '0', then the default interpolation filters are used to interpolate this sub-pixel position and its symmetric sub-pixel positions.

[0078] In the heuristic solution, using symmetry classes, an error is accumulated if both intra pixel 808 and predicted pixel 812 are pointing to the same sub-pixel position. For example, the matrix may be simplified to:

```
matrix  =    0, 0, a, a
             0, 1, a, a
             1, 0, a, a
             1, 1, a, a
```

[0079] In a bi-predicted block, the full or sub-pixel positions for the two reference blocks used in the prediction can be different, so the SAD error has to be collected for each combination of positions. It is possible to iterate through all blocks and, in a single pass, accumulate the absolute differences between the original and the interpolated

pixels into a four-dimensional matrix where $C_{d_i,d_j,i,j}$ represents the sum of absolute differences accumulated for the positions $i,j=1,...,n$ when the filter decisions are $d_i,d_j \in \{0,1\}$.

[0080] For each block being encoded, the positions i and j refer to the second intra frame macroblock 806 (A) and the third predicted macroblock 810 (B). Then, for each pixel in the block 814, four predictions are computed by using pixStdX and pixAifX, the pixels interpolated with the default and with the alternative filters:

```
pix_00 = (1 - b_scaling) * pixStdA + b_scaling * pixStdB;
pix_01 = (1 - b_scaling) * pixStdA + b_scaling * pixAifB;
pix_10 = (1 - b_scaling) * pixAifA + b_scaling * pixStdB;
pix_11 = (1 - b_scaling) * pixAifA + b_scaling * pixAifB;
```

[0081] the reconstructed pixels are finally computed as:

```
rec_00 = max(0, min(max_pixel_value, (int)(pix_00 + 0.5)));
rec_01 = max(0, min(max_pixel_value, (int)(pix_01 + 0.5)));
rec_10 = max(0, min(max_pixel_value, (int)(pix_10 + 0.5)));
rec_11 = max(0, min(max_pixel_value, (int)(pix_11 + 0.5)));
```

[0082] and used to determine the absolute error corresponding to the use of the default or of the alternative interpolation filter. The error will be accumulated in the matrix as:

[0083]
```
C[0][0][i][j] += (double)abs(orig - rec_00);
C[1][1][i][j] += (double)abs(orig - rec_11);
C[0][1][i][j] += (double)abs(orig - rec_01);
C[1][0][i][j] += (double)abs(orig - rec_10);
```

[0084] The scaling factor b-scaling can be either 0.5 or computed according to the sequence of the current and reference frames. This second method may be used to weight more the contributions of predictions originated from a temporally closer reference frame.

[0085] If one of the two reference blocks is missing, then the error contribution is only accumulated for one of the two positions. Furthermore, the error contributions to

C[0][1] and C[1][0] are not accumulated when $i = j$ and the position is the same for both reference blocks.

[0086] For each used alternative filter the filter coefficients may be explicitly signaled, so each position $i = 1,...,n$ (whether full- or sub-pixel) has an associated weighted cost $\lambda \cdot nBits_i$, with the weighting factor reflecting the contribution of the bits to the rate-distortion performance. If the filters are predetermined or pre-calculated and available to the encoder, then information, such as, cost (in bits) may not need to be transmitted.

[0087] With this formalism, the application consists of finding the binary decision vector $D = (d_1, d_2, ..., d_n)$ that minimizes:

$$\min_{D=(d_1,d_2,...,d_n)} \sum_{i=1}^{n} \sum_{j=1}^{n} C_{d_i,d_j,i,j} + \sum_{i=1}^{n} d_i \cdot \lambda \cdot nBits_i$$

[0088] Given the matrix $C_{d_i,d_j,i,j}$ and the vector $\lambda \cdot nBits_i$, it is possible to exhaustively compute the error corresponding to each of the 2^n decision vectors and select the smallest.

[0089] An iterative method that closely approximates the optimal strategy includes starting from a possible solution (any valid decision vector) and finding the position for which an alternative filter has the largest error reduction. This position is then updated with the value of the new filter choice and the process is repeated until no more improvements are possible, the improvement is negligible (below a predetermined threshold) or a maximum number of iterations is reached. The three stopping criteria may be combined together.

[0090] A pseudo-code description of the method is as follows:

```
D[0,..,n-1] = {0, 0, 0, ..., 0};          //A starting solution
SAD = ComputeSAD(D, C);                    // C is the cost matrix
while(stopping criteria are not satisfied)
{ (P, F) = FindPosition(D, C, SAD);  // Find position and filter
                                           // decreasing SAD the most
  D[P] = F;                                // Update D
  SAD = ComputeSAD(D, C);                  // Update SAD }
```

[0091] This method is able to converge to a local minimum in 8 to 10 iterations in many cases. If n is the number of positions and k is the number of choices (filters)

available for each position, each iteration has $n \cdot (k-1)$ possibilities (the remaining $k-1$ filters on each of the n positions). If we limit the maximum number of iterations to be at most c, this method will have a computational complexity that is linear in the number of positions and filters.

[0092] A method of using uneven quantization, that is, the quantization precision, depends on the coefficient location. It has been found that for coefficients with smaller magnitude (these are coefficients farther away from the center of the filter), better precision is needed compared to coefficients with larger magnitude (these are coefficients closer to the center of the filter).

[0093] The matrix Q^{FP} is used to specify the quantization precision of the coefficients of the integer-pixel filter. The numbers of bits in the matrix includes 1 bit for sign so, for example, the center coefficient will be quantized with 9 bits of which 1 used for the sign and 8 bits for the error magnitude.

$$Q^{FP} = \begin{bmatrix} 11 & 11 & 11 & 11 & 11 \\ 11 & 10 & 10 & 10 & 11 \\ 11 & 10 & 9 & 10 & 11 \\ 11 & 10 & 10 & 10 & 11 \\ 11 & 11 & 11 & 11 & 11 \end{bmatrix}$$

[0094] It has been found experimentally that Q^{FP} provides the best quantization for P-frames, while B-frames are better encoded by using the finer quantization described by $\hat{Q}^{FP}$:

$$\hat{Q}^{FP} = \begin{bmatrix} 12 & 12 & 12 & 12 & 12 \\ 12 & 11 & 11 & 11 & 12 \\ 12 & 11 & 10 & 11 & 12 \\ 12 & 11 & 11 & 11 & 12 \\ 12 & 12 & 12 & 12 & 12 \end{bmatrix}$$

[0095] FIG. 9 is a flow diagram describing a process 92 for creating a final reference frame based on filter selection decisions using the system shown in FIG. 3. The process begins at block 93 by identifying a set of reference frames and a current frame. For example, the set of reference frames stored in the reference frame store 25 and the current frame transmitted to the motion compensation unit 27 within the video encoder 26. In one embodiment, the set of reference frames comprise an intra-frame and predicted frames.

[0096] The process 92 then moves to block 94 and interpolates the set of reference frames using a default filter to create a default interpolated frame. The set of reference frames in the reference frame store 25 previously identified, may be interpolated using a default filter and transmitted to the reference frame store 25 for storage and later retrieval. The default filter may be any default filter (i.e. H.264 filter) resident within a video encoder 20 that is declared to be normative by a standards organization. In one embodiment, the set of reference frames are interpolated using a default H.264 filter and stored in the reference frame store 25 within the video encoder 26.

[0097] The process 92 then moves to block 95 and interpolates the set of reference frames using an alternative filter to create an alternative interpolated frame. The set of reference frames in the reference frame store 25 previously identified, may be interpolated using an alternative filter and transmitted to the reference frame store 25 for storage and later retrieval. The alternative filter may reside within the motion compensation unit 27, or may reside in another component within the encoder. The alternative filter may be any adaptive filter (i.e. 2D adaptive interpolation filter, E-AIF, etc) resident within a video encoder 26. In other words, an alternative filter is any filter that is available in the video encoder 20 which is not labeled a default filter. In one embodiment, the set of reference frames are interpolated using an enhanced adaptive interpolation filter and stored in the reference frames store 25 within the video encoder 26.

[0098] The process 92 then moves to block 96 and comparing for each sub pixel position, if the interpolated reference frames using the default filter have a lower sum error than the interpolated reference frames using the alternative filter; wherein the lower sum error based on the lowest combination of error rate and distortion. The sub pixel position may be one half of a pixel, one quarter of a pixel, one eighth of a pixel, or any fractional value less than one. In one embodiment, a sub pixel position is determined to have a lower sum error using a default filter than an alternative filter after comparing both interpolated reference frames.

[0099] The process 92 then moves to block 97, wherein the alternative filter is selected if it has the lower sum error value. In one embodiment, for each sub pixel position to be generated in a final frame, an alternative filter is selected if the alternative filter (for example, an adaptive interpolation filter) is determined to have a lower error rate and distortion rate.

[0100] Otherwise, the process 92 moves to block 98 wherein the default filter is selected if the alternative filter did not have the lower sum error value. In one embodiment, for each sub pixel position to be generated in a final frame, a default filter is selected if the default filter is determined to have a lower error and distortion rate.

[0101] The process 92 then moves to block 99 wherein for each sub pixel position, the best combination of either a default or an alternative filter is selected to generate a final reference frame. In one embodiment, the final reference frame is generated using only a default filter for each sub pixel position. In another embodiment, the final reference frame is generated using alternative filters, determined to have a lower sum error rate. In another embodiment, the final reference frame is generated using a combination of alternative and default filters determined to have a lower sum error rate per sub pixel position.

[0102] Conditional language, such as, among others, "can," "could," "might," or "may," unless specifically stated otherwise, or otherwise understood within the context as used, is generally intended to convey that certain embodiments include, while other embodiments do not include, certain features, elements and/or steps. Thus, such conditional language is not generally intended to imply that features, elements and/or steps are in any way required for one or more embodiments or that one or more embodiments necessarily include logic for deciding, with or without user input or prompting, whether these features, elements and/or steps are included or are to be performed in any particular embodiment.

[0103] Any process descriptions, elements, or blocks in the flow diagrams described herein and/or depicted in the attached FIG.s should be understood as potentially representing modules, segments, or portions of code which include one or more executable instructions for implementing specific logical functions or steps in the process. Alternate implementations are included within the scope of the embodiments described herein in which elements or functions may be deleted, executed out of order from that shown or discussed, including substantially concurrently or in reverse order, depending on the functionality involved, as would be understood by those skilled in the art.

[0104] The techniques described herein may be implemented in hardware, software, firmware, or any combination thereof. If implemented in software, the techniques may be realized at least in part by one or more stored or transmitted instructions or code on a

computer-readable medium. Computer-readable media may include computer storage media, communication media, or both, and may include any medium that facilitates transfer of a computer program from one place to another. A storage media may be any available media that can be accessed by a computer.

[0105] By way of example, and not limitation, such computer-readable media can comprise RAM, such as synchronous dynamic random access memory (SDRAM), read-only memory (ROM), non-volatile random access memory (NVRAM), ROM, electrically erasable programmable read-only memory (EEPROM), EEPROM, FLASH memory, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium that can be used to carry or store desired program code in the form of instructions or data structures and that can be accessed by a computer.

[0106] Also, any connection is properly termed a computer-readable medium. For example, if the software is transmitted from a website, server, or other remote source using a coaxial cable, fiber optic cable, twisted pair, digital subscriber line (DSL), or wireless technologies such as infrared, radio, and microwave, then the coaxial cable, fiber optic cable, twisted pair, DSL, or wireless technologies such as infrared, radio, and microwave are included in the definition of medium. Disk and disc, as used herein, includes compact disc (CD), laser disc, optical disc, digital versatile disc (DVD), floppy disk and blu-ray disc where disks usually reproduce data magnetically, while discs reproduce data optically, e.g., with lasers. Combinations of the above should also be included within the scope of computer-readable media.

[0107] The code associated with a computer-readable medium of a computer program product may be executed by a computer, e.g., by one or more processors, such as one or more digital signal processors (DSPs), general purpose microprocessors, application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry. In some aspects, the functionality described herein may be provided within dedicated software modules or hardware modules configured for encoding and decoding, or incorporated in a combined video encoder-decoder (CODEC).

[0108] Various aspects have been described. These and other aspects are within the scope of the following claims.

WHAT IS CLAIMED IS:

1. A method of encoding digital video, the method comprising:

identifying a reference frame and a current frame;

interpolating the reference frame using a default filter to create a default interpolated frame;

interpolating the reference frame using an alternative filter to create an alternative interpolated frame; and

determining for each sub-pixel position whether to use the default filter or the alternative filter based on a lower sum error to generate a final reference frame.

2. The method of claim 1, wherein the reference frame is a predicted frame.

3. The method of claim 1, wherein the reference frame is a bi-directional predicted frame.

4. The method of claim 1, wherein the default filter is a H.264 filter

5. The method of claim 1, wherein the alternative filter is an adaptive interpolation filter.

6. The method of claim 1, wherein the lower sum error is based on the lowest combination of error rate and distortion.

7. A video encoding device, comprising:

a motion compensation unit configured to interpolate a first reference frame using a default filter to create a default interpolated frame and interpolate a second reference frame using an alternative filter to create an alternative interpolated frame;

a reference frame store configured to store the default interpolated frame and the alternative interpolated frame;

a filter decision unit configured to

receive a current frame, the default interpolated frame, and the alternative interpolated frame;

determine, for each full and sub-pixel position, whether to use a default filter or an alternative filter in a final frame; and

generate the final frame.

8. The video encoding device of claim 7, wherein the interpolated reference frames are generated using a H.264 filter.

9. The video encoding device of claim 7, wherein the final reference frame is created using a combination of filters determined to have a low error rate and a low distortion rate.

10. The video encoding device of claim 7, wherein the sub-pixel position is a ½ pixel, a ¼ pixel, or 1/8 pixel.

11. The video encoding device of claim 7, wherein the default filter is a H.264 filter.

12. The video encoding device of claim 7, wherein the alternative filter is an adaptive interpolation filter.

13. A video encoding device, comprising:

means for receiving a current frame;

means for storing a plurality of interpolated reference frames in a reference frame store;

means for receiving the plurality of interpolated reference frames from the reference frame store;

means for producing a final frame; and

means for determining for each full and sub pixel position in the final frame whether to use a default filter or an alternative filter.

14. The video encoding device of claim 13, wherein the plurality of interpolated reference frames are generated using a H.264 filter.

15. The video encoding device of claim 13, wherein the final reference frame is created using a combination of filters determined to have a low error rate and a low distortion rate.

16. The video encoding device of claim 13, wherein the sub-pixel position is a ½ pixel, a ¼ pixel, or 1/8 pixel.

17. The video encoding device of claim 13, wherein the default filter is a H.264 filter.

18. The video encoding device of claim 13, wherein the alternative filter is an adaptive interpolation filter.

19. A computer readable medium comprising instructions to cause a processor to:

identify a reference frame and a current frame;

interpolate the reference frame using a default filter to create a default interpolated frame;

interpolate the reference frame using a alternative filter to create an alternative interpolated frame; and

determine, for each sub-pixel position, whether to use the default filter or the alternative filter based on a lower sum error to generate a final reference frame.

20. The method of claim 19, wherein the reference frame is an intra-frame.

21. The method of claim 19, wherein the reference frame is a predicted frame.

22. The method of claim 19, wherein the default filter is a H.264 default filter

23. The method of claim 19, wherein the alternative filter is an adaptive interpolation filter.

24. The method of claim 19, wherein the lower sum error is based on the lowest combination of error rate and distortion.

25. A video encoding device, comprising:

a motion compensation unit configured to interpolate a first reference frame using a default filter to create a default interpolated frame and interpolate a second reference frame using an alternative filter to create an alternative interpolated frame;

a filter decision unit configured to receive a current frame, the default interpolated frame, and the alternative interpolated frame,

wherein the filter decision unit is further configured to determine, for each full and sub-pixel position, whether to use a default filter or an alternative filter in a final frame and to generate the final frame.

26. The video encoding device of claim 25, wherein the alternative filter is one of a plurality of pre-calculated filters.

27. A video encoding device, comprising:

means for receiving a current frame;

means for receiving a plurality of interpolated reference frames

means for producing a final frame; and

means for determining for each full and sub pixel position in the final frame whether to use a default filter or an alternative filter.

28. The video encoding device of claim 27, wherein the alternative filter is one of a plurality of pre-calculated interpolation filters.