ADAPTIVE RESOLUTION MANAGEMENT PREDICTION RESCALING

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of priority of U.S. Provisional Patent Application Serial No. 62/883,454, filed on August 6, 2019, and titled "ADAPTIVE RESOLUTION MANAGEMENT PREDICTION RESCALING," which is incorporated by reference herein in its entirety.

FIELD OF THE INVENTION

The present invention generally relates to the field of video compression. In particular, the present invention is directed to adaptive resolution management prediction rescaling.

BACKGROUND

A video codec may include an electronic circuit or software that compresses or decompresses digital video. It may convert uncompressed video to a compressed format or vice versa. In the context of video compression, a device that compresses video and/or performs some function thereof may typically be called an encoder, and a device that decompresses video and/or performs some function thereof may be called a decoder.

A format of compressed data may conform to a standard video compression specification. Compression may be lossy in that compressed video may lack some information present in an original video. A consequence of this may include that decompressed video may have lower quality than an original uncompressed video because there may be insufficient information to accurately reconstruct the original video.

There can be complex relationships between video quality, an amount of data used to represent a video, for instance as determined by bit rate, complexity of encoding and decoding algorithms, sensitivity to data losses and errors, ease of editing, random access, end-to-end delay such as without limitation latency, and the like.

Motion compensation may include an approach to predict a video frame or a portion thereof given a reference frame, such as previous and/or future frames, by accounting for motion of the camera and/or objects in the video. It may be employed in encoding and decoding of video data for video compression, for example in encoding and decoding using the Motion Picture Experts Group (MPEG)'s advanced video coding (AVC) standard (also referred to as H.264). Motion compensation may describe a picture in terms of transformation of a reference picture to

a current picture. A reference picture may be previous in time when compared to a current picture and/or from the future when compared to the current picture.

SUMMARY OF THE DISCLOSURE

In an aspect, a decoder includes circuitry configured to receive a reference frame, determine, for a current block, a scaling constant, determine a scaled reference block using the reference frame and the scaling constant, determine a scaled prediction block using the scaled reference block, and reconstructing pixel data of the current block and using the rescaled prediction block.

In another aspect, a method includes receiving a reference frame, determining, for a current block, a scaling constant, determining a scaled reference block using the reference frame and the scaling constant, determining a scaled prediction block using the scaled reference block, and reconstructing pixel data of the current block and using the rescaled prediction block.

The details of one or more variations of the subject matter described herein are set forth in the accompanying drawings and the description below. Other features and advantages of the subject matter described herein will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

For the purpose of illustrating the invention, the drawings show aspects of one or more embodiments of the invention. However, it should be understood that the present invention is not limited to the precise arrangements and instrumentalities shown in the drawings, wherein:

FIG. 1 is an illustration of an example reference frame and example predicted frames of various resolution scales;

FIG. 2 is an illustration depicting an example reference frame, an example rescaled reference frame, and an example subsequent block prediction process;

FIG. 3 is an illustration depicting an example reference block position before and after the scaling process;

FIG. 4 is an illustration depicting an example motion vector scaling including the reference block, predicted block, scaled reference block, and scaled predicted block;

FIG. 5 is a process flow diagram illustrating an example process according to some implementations of the current subject matter;

FIG. 6 is a system block diagram illustrating an example decoder capable of decoding a bit stream according to some implementations of the current subject matter;

FIG. 7 is a process flow diagram illustrating an example process of encoding a video according to some implementations of the current subject matter;

FIG. 8 is a system block diagram illustrating an example video encoder according to some implementations of the current subject matter; and

FIG. 9 is a block diagram of a computing system that can be used to implement any one or more of the methodologies disclosed herein and any one or more portions thereof.

The drawings are not necessarily to scale and may be illustrated by phantom lines, diagrammatic representations and fragmentary views. In certain instances, details that are not necessary for an understanding of the embodiments or that render other details difficult to perceive may have been omitted. Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

In many current state-of-the-art encoders, resolution is managed by re-coding and re-sending whole portion of the video known as group-of-pictures (GOP). This requires sending the intra-frame (I-frame) which can incur additional costs, since those frames are responsible for the bulk of the bits in the GOP.

Embodiments described in this disclosure relate to Adaptive Resolution Management (ARM), which is a technique that enables additional flexibility for a video encoder/decoder allowing bitrate savings in various use cases. In general, ARM includes performing a prediction using a reference frame of different resolution than a current frame. In current coding standards, reference frames have the same resolution as predicted frames. In ARM, a reference frame may be of a smaller or larger resolution than a frame being predicted. This approach may be used to downscale video resolution, thus reducing bitrate, or upscale video resolution, thus facilitating display characteristics of video playback.

ARM may alternatively or equivalently be referred to for the purposes of this disclosure as reference picture resampling (RPR); RPR and ARM may be used interchangeably.

Some implementations of the current subject matter may include using ARM for any number of frames, at any position within GOP, thus removing requirements for I-frame re-coding.

FIG. 1 is an illustration of a reference frame and predicted frames of various resolution scales. Frame 1 is smaller (lower resolution) than the reference frame, Frame 2 is of same size (same resolution), while Frame 3 is larger (greater resolution). "Resolution," as used in this disclosure, is a number of pixels in a picture, frame, sub-frame, and/or other displayed area or portion thereof used in video playback, compression, or the like, with a higher number of pixels corresponding to a higher resolution, and a lower number of pixels corresponding to a lower resolution. Resolution may be measured in terms of area, for instance and without limitation by using one or more dimensions of length, measured in pixels that define an area. For instance, a circular sub-frame or other region may have a resolution defined according to a radius. Alternatively or additionally, resolution may be defined by an overall number of pixels.

As an example, and with continued reference to FIG. 1, where reference frame and/or subframe has a geometric form for which area may be defined entirely in terms of two length parameters, such as without limitation a triangular, parallelogram, and/or rectangular form, reference frame and/or subframe may have a resolution W x H, where W and H may indicate a number of pixels describing, respectively, width (or base) and height dimensions of reference frame and/or subframe. Each predicted frame may also have a resolution, which may be determined similarly to resolution of reference frame; for instance, frame 1 may have a smaller resolution WS x HS, frame 2 may have same resolution as reference frame W x H, and frame 3 may have larger resolution WL x HL. Width and height of smaller and larger frames may be obtained by multiplying reference width and height by an arbitrary rescaling constant (Rc), also referred to as a scaling factor and/or constant. In case of smaller frames, Rc may have a value between 0 and 1. In case of larger frames, Rc may have a value greater than 1; for instance, Rc may have a value between 1 and 4. Other values are possible. Rescaling constant may be different for one resolution dimension than another; for instance, a rescaling constant Rch may be used to rescale height, while another rescaling constant Rcw may be used to rescale width.

Still referring to FIG. 1, ARM may be implemented as a mode. In case of ARM mode activation at some point during decoding, a decoder may have already received a reference frame at a resolution W x H and may rescale a predicted frame using rescaling constant. In some implementations, an encoder may signal to decoder which rescaling constant to use. Signaling may be performed in a sequence parameter set (SPS) corresponding to a GOP containing current picture and/or in a picture parameter set (PPS) corresponding to current picture. For instance,

and without limitation, encoder may signal rescaled parameters using fields such as a pps_pic_width_in_luma_samples, pps_pic_height_in_luma_samples, pps_scaling_win_left_offset, pps_scaling_win_right_offset, pps_scaling_win_top_offset, pps_scaling_win_bottom_offset, and/or sps_num_subpics_minus1.

Further referring to FIG. 1, W and H parameters as described above may be represented, without limitation, using variables CurrPicScalWinWidthL and CurrPicScalWinHeightL, respectively; these variables may be derived from signaled parameters as described above using one or more mathematical relationships between the signaled parameters and the variables. For instance, and without limitation, CurrPicScalWinWidthL may be derived according to the following equation:

CurrPicScalWinWidthL = pps_pic_width_in_luma_samples −
SubWidthC * (pps_scaling_win_right_offset + pps_scaling_win_left_offset)

As a further non-limiting example, CurrPicScalWinHeightL may be derived according to the following equation:

CurrPicScalWinWidthL = pps_pic_width_in_luma_samples −
SubWidthC * (pps_scaling_win_right_offset + pps_scaling_win_left_offset)

Persons skilled in the art, upon reviewing the entirety of this disclosure, will be aware of various alternative computations that may be used to derive the above-described variables. Encoder may alternatively or additionally signal one or more such variables, Rc, Rch, and/or Rcw directly for instance and without limitation in a PPS and/or SPS.

Alternatively or additionally, and still referring to FIG. 1, a rescaling constant and/or set of rescaling constants as described above may be signaled in a bitstream using a reference to an index of a stored scaling constant and/or constants, and/or of a frame and/or block that was signaled using a previously signaled and/or utilized scaling constant and/or constants. Reference to index of a stored scaling constant may be signaled explicitly and/or determined from one or more additional parameters signaled in bitstream. For instance, and without limitation, decoder may identify a reference frame and/or group of pictures containing a current frame; where a rescaling constant has previously been signaled and/or used in such a group of pictures, with a reference frame signaled as applicable to current frame and/or current group of pictures, or the like, decoder may identify that rescaling constant for use as a rescaling constant with the current frame.

In some implementations, and with continued reference to FIG. 1, ARM operation may be executed on a block level of encoded frames. For example, a reference frame may first be rescaled and subsequently prediction may be performed, as depicted in FIG. 2. FIG. 2 is an illustration depicting a reference frame, a rescaled reference frame, and a subsequent block prediction process. Block prediction process may be performed on a scaled reference frame (having a scaled resolution) rather than the original reference frame. Rescaling reference frame may include rescaling according to any parameter signaled by an encoder as described above; for instance and without limitation, where a reference frame to be used with current picture is signaled, such as via a reference to an index value associated with reference frame or the like, the signaled reference frame may be rescaled, prior to prediction, according to any method of rescaling described above. A rescaled reference frame may be stored in memory and/or in a buffer, which may include without limitation a buffer identifying frames contained therein by indices, according to which frame retrieval may be performed; buffer may include a decoded picture buffer (DCB) and/or one or more additional buffers implemented by decoder. Prediction process may include, for example, inter picture prediction including motion compensation.

Some implementations of block-based ARM may enable flexibility in applying optimal filters for each block, instead of applying same filter for a whole frame. In some implementations, a skip-ARM mode may be possible such that some blocks (based for example on the uniformity of the pixels and bitrate cost) can be in a skip-ARM mode (such that rescaling would not change the bitrate). Skip-ARM mode may be signaled in a bitstream; for instance, and without limitation, skip-ARM mode may be signaled in a PPS parameter. Alternatively or additionally, decoder may determine that skip-ARM mode is active based on one or more parameters set by decoder and/or signaled in bitstream. Spatial filters used in block-based ARM may include, without limitation, bicubic spatial filters that apply bicubic interpolation, bilinear spatial filters that apply bi-linear interpretation, Lanczos filters that use Lanczos filtering and/or Lanczos resampling using combinations of sinc filters, sinc-function interpolation and/or signal reconstruction techniques, or the like; persons skilled in the art, upon reviewing the entirety of this disclosure, will be aware of various filters that may be used for interpolation consistently with this disclosure.

Continuing to refer to FIG. 2, in a case where a sequence of frames is already encoded, post-encoding ARM may allow quick rescaling and transmission of predicted coding units

(blocks or frames). This may remove the need to re-encode video frames at a different resolution. Some implementations of current subject matter may reduce computational complexity at a relatively negligible cost in rate-distortion performance.

With further reference to FIG. 2, depending on a use case a rescaling may be done, in an encoding and decoding process where an encoder transmits to a decoder, on the encoder side (for downscaling, reduced bandwidth) or on the decoder side (for upscaling, again reducing the bandwidth needed for video transmission). In some implementations, scaling may be performed using interpolation filters such as sub-pixel prediction filters. Interpolation filters may include, as a non-limiting example, any filters described above, a low-pass filter, which may be used, without limitation, by way of an up-sampling process whereby pixels between pixels of block and/or frame previous to scaling may be initialized to zero, and then populated with an output of the low-pass filter. Alternatively or additionally, any luma sample interpolation filtering process may be used. Luma sample interpretation may include computation of an interpolated value at a half-sample interpolation filter index, falling between two consecutive sample values of a non-scaled sample array. Computation of interpolated value may be performed, without limitation, by retrieval of coefficients and/or weights from lookup tables; selection of lookup tables may be performed as a function of motion models of coding units and/or scaling ratio amounts, for instance as determined using scaling constants as described above. Computation may include, without limitation, performing weighted sums of adjacent pixel values, where weights are retrieved from lookup tables. Computed values may alternatively or additionally be shifted; for instance and without limitation, values may be shifted by Min(4, BitDepth − 8), 6, Max(2, 14 − BitDepth), or the like. Persons skilled in the art, upon reviewing the entirety of this disclosure, will be aware of various alternative or additional implementations that may be used for interpolation filters.

In such an approach, and still referring to FIG. 2, predicted elements may be rescaled by a factor Rc (scaling constant or factor), which may be signaled to decoder, for instance as described above. For instance, and without limitation, pps_scaling_window_explicit_signalling_flag equal to 1 may indicate that scaling window offset parameters are present in a PPS, while pps_scaling_window_explicit_signalling_flag equal to 0 may indicate that scaling window offset parameters are not present in the PPS. When

sps_ref_pic_resampling_enabled_flag is equal to 0, a pps_scaling_window_explicit_signalling_flag may be equal to 0.

Scaled elements may include intra and inter coding units (e.g., blocks), that may be rescaled using factor scaling such as depicted in FIG. 3. FIG. 3 is an illustration depicting reference block position before and after a scaling process.

Scaled elements may include motion vectors. FIG. 4 is an illustration depicting motion vector scaling including a reference block, predicted block, scaled reference block, and scaled predicted block. FIG. 4 illustrates a motion compensated prediction for an original size reference frame (top) and for a scaled reference frame (bottom).

FIG. 5 is a process flow diagram illustrating an exemplary embodiment of a process 500 of adaptive resolution management that may enable additional flexibility for a video encoder/decoder allowing bitrate savings in various use cases.

At step 505, and still referring to FIG. 5, a reference frame is received. Reference frame may be received (e.g., accessed from) memory. The reference frame may have been previously decoded from a bit stream. Bit stream may have been received by a decoder. Bit stream may include a current block. Current block may be contained within a bit stream that a decoder receives. Bit stream may include, for example, data found in a stream of bits that is an input to a decoder when using data compression. Bit stream may include information necessary to decode a video. Receiving may include extracting and/or parsing a block and associated signaling information from bit stream. In some implementations, a current block may include a coding tree unit (CTU), a coding unit (CU), and/or a prediction unit (PU).

At step 510, and with continued reference to FIG. 5, a scaling constant is determined for a current block. In some implementations, scaling constant may be determined from bit stream. For example, a scaling constant may be signaled within a bit stream, the bit stream may include an index to one of a number of predetermined scaling constants, or the like.

At step 515, and still referring to FIG. 5, a scaled reference block may be determined using reference frame and scaling constant. Determining a scaled reference block may include determining a position of scaled reference block. In some implementations, position may be determined at least by multiplying one or more dimensions of a reference block by one or more scaling factors; for instance, a horizontal component of a position of a reference block may be

multiplied by scaling factor and a vertical component of a the position of the reference block may be multiplied by the scaling factor.

At step 520, and with continuing reference to FIG. 5, a scaled prediction block may be determined using scaled reference block. In some implementations, determining scaled prediction block may include determining scaled motion information. Scaled motion information may be determined by at least multiplying a horizontal component of a motion vector by scaling constant and multiplying a vertical component of the motion vector by the scaling constant.

At step 525, and still referring to FIG. 5, pixel data of the current block may be reconstructed using a rescaled prediction block.

FIG. 6 is a system block diagram illustrating an example decoder 600 capable of adaptive resolution management prediction rescaling as described in this disclosure. Decoder 600 may include an entropy decoder processor 604, an inverse quantization and inverse transformation processor 608, a deblocking filter 612, a frame buffer 616, a motion compensation processor 620 and/or an intra prediction processor 624.

In operation, and still referring to FIG. 6, bit stream 628 may be received by decoder 600 and input to entropy decoder processor 604, which may entropy decode portions of bit stream into quantized coefficients. Quantized coefficients may be provided to inverse quantization and inverse transformation processor 608, which may perform inverse quantization and inverse transformation to create a residual signal, which may be added to an output of motion compensation processor 620 or intra prediction processor 624 according to a processing mode. An output of the motion compensation processor 620 and intra prediction processor 624 may include a block prediction based on a previously decoded block. A sum of prediction and residual may be processed by deblocking filter 612 and stored in a frame buffer 616.

FIG. 7 is a process flow diagram illustrating an exemplary embodiment of a process 700 of encoding a video with adaptive resolution management that may enable additional flexibility for a video encoder and/or decoder allowing bitrate savings in various use cases. At step 705, a video frame may undergo initial block segmentation, for example, using a tree-structured macro block partitioning scheme that may include partitioning a picture frame into CTUs and CUs.

At step 710, and still referring to FIG. 7, block-based adaptive resolution management may be performed, including resolution scaling of a frame or portion thereof.

At step 715, and with continued reference to FIG. 7, a block may be encoded and included in bit stream. Encoding may include utilizing inter prediction and intra prediction modes, for example.

FIG. 8 is a system block diagram illustrating an example video encoder 800 capable of adaptive resolution management prediction rescaling as described in this disclosure. Example video encoder 800 may receive an input video 804, which may be initially segmented or dividing according to a processing scheme, such as a tree-structured macro block partitioning scheme (e.g., quad-tree plus binary tree). An example of a tree-structured macro block partitioning scheme may include partitioning a picture frame into large block elements called coding tree units (CTU). In some implementations, each CTU may be further partitioned one or more times into a number of sub-blocks called coding units (CU). A final result of this portioning may include a group of sub-blocks that may be called predictive units (PU). Transform units (TU) may also be utilized.

Still referring to FIG. 8, example video encoder 800 may include an intra prediction processor 808, a motion estimation / compensation processor 812, which may also be referred to as an inter prediction processor, capable of constructing a motion vector candidate list including adding a global motion vector candidate to the motion vector candidate list, a transform /quantization processor 816, an inverse quantization / inverse transform processor 820, an in-loop filter 824, a decoded picture buffer 828, and/or an entropy coding processor 832. Bit stream parameters may be input to the entropy coding processor 832 for inclusion in the output bit stream 836.

In operation, and with continued reference to FIG. 8, for each block of a frame of input video 804, whether to process block via intra picture prediction or using motion estimation / compensation may be determined. Block may be provided to intra prediction processor 808 or motion estimation / compensation processor 812. If block is to be processed via intra prediction, intra prediction processor 808 may perform processing to output a predictor. If block is to be processed via motion estimation / compensation, motion estimation / compensation processor 812 may perform processing including constructing a motion vector candidate list including adding a global motion vector candidate to the motion vector candidate list, if applicable.

Further referring to FIG. 8, a residual may be formed by subtracting a predictor from input video. Residual may be received by transform / quantization processor 816, which may

perform transformation processing (e.g., discrete cosine transform (DCT)) to produce coefficients, which may be quantized. Quantized coefficients and any associated signaling information may be provided to entropy coding processor 832 for entropy encoding and inclusion in output bit stream 836. Entropy encoding processor 832 may support encoding of signaling information related to encoding a current block. In addition, quantized coefficients may be provided to inverse quantization / inverse transformation processor 820, which may reproduce pixels, which may be combined with a predictor and processed by in loop filter 824, an output of which may be stored in decoded picture buffer 828 for use by motion estimation / compensation processor 812 that is capable of constructing a motion vector candidate list including adding a global motion vector candidate to the motion vector candidate list.

With continued reference to FIG. 8, although a few variations have been described in detail above, other modifications or additions are possible. For example, in some implementations, current blocks may include any symmetric blocks (8x8, 16x16, 32x32, 64x64, 128 x 128, and the like) as well as any asymmetric block (8x4, 16x8, and the like).

In some implementations, and still referring to FIG. 8, a quadtree plus binary decision tree (QTBT) may be implemented. In QTBT, at a Coding Tree Unit level, partition parameters of QTBT may be dynamically derived to adapt to local characteristics without transmitting any overhead. Subsequently, at a Coding Unit level, a joint-classifier decision tree structure may eliminate unnecessary iterations and control the risk of false prediction. In some implementations, LTR frame block update mode may be available as an additional option available at every leaf node of QTBT.

In some implementations, and still referring to FIG. 8, additional syntax elements may be signaled at different hierarchy levels of bitstream. For example, a flag may be enabled for an entire sequence by including an enable flag coded in a Sequence Parameter Set (SPS). Further, a CTU flag may be coded at a coding tree unit (CTU) level.

Some embodiments may include non-transitory computer program products (i.e., physically embodied computer program products) that store instructions, which when executed by one or more data processors of one or more computing systems, cause at least one data processor to perform operations herein.

Embodiments disclosed herein may include a decoder having circuitry configured to receive a reference frame, determine, for a current block, a scaling constant, determine a scaled

reference block using the reference frame and the scaling constant, determine a scaled prediction block using the scaled reference block, and reconstruct pixel data of the current block and using the rescaled prediction block.

In some embodiments, the decoder may be further configured to determine the scaled prediction block by determining scaled motion information. The scaled motion information may be determined by at least multiplying a horizontal component of a motion vector by the scaling constant and multiplying a vertical component of the motion vector by the scaling constant. The decoder may be further configured to determine the scaled reference block by determining a position of the scaled reference block. The position may be determined by multiplying a horizontal component of a position of a reference block by the scaling factor and multiplying a vertical component of the position of the reference block by the scaling factor. The decoder may be further configured to receive a bit stream and determine, from the bit stream, the scaling constant, wherein the scaling constant is signaled within the bit stream. The bit stream may include an index to a predetermined scaling constant. The decoder may be further configured to receive the reference frame by accessing the reference frame from memory. The decoder may include an entropy decoder processor configured to receive a bit stream and decode the bit stream into quantized coefficients, an inverse quantization and inverse transformation processor configured to process the quantized coefficients including performing an inverse discrete cosine, a deblocking filter, a frame buffer, and an intra prediction processor. The current block may include a coding tree unit. The current block may include a coding unit. The current block may include a prediction unit.

Embodiments disclosed herein may include a method. The method may include receiving a reference frame, determining, for a current block, a scaling constant, determining a scaled reference block using the reference frame and the scaling constant, determining a scaled prediction block using the scaled reference block, and reconstructing pixel data of the current block and using the rescaled prediction block.

In some embodiments, determining the scaled prediction block may include determining scaled motion information. The scaled motion information may be determined by multiplying a horizontal component of a motion vector by the scaling constant and multiplying a vertical component of the motion vector by the scaling constant. Determining the scaled reference block may include determining a position of the scaled reference block. The position may be

determined by multiplying a horizontal component of a position of a reference block by the scaling factor and multiplying a vertical component of the position of the reference block by the scaling factor. The method may include receiving a bit stream and determining, from the bit stream, the scaling constant, wherein the scaling constant is signaled within the bit stream. The bit stream may include an index to a predetermined scaling constant. Receiving the reference frame may include accessing the reference frame from memory. At least one of the receiving, the determining, and the reconstructing may be performed by a decoder including an entropy decoder processor configured to receive a bit stream and decode the bit stream into quantized coefficients, an inverse quantization and inverse transformation processor configured to process the quantized coefficients including performing an inverse discrete cosine, a deblocking filter, a frame buffer, and an intra prediction processor. The current block may include a coding tree unit. The current block may include a coding unit. The current block may include a prediction unit.

It is to be noted that any one or more of the aspects and embodiments described herein may be conveniently implemented using digital electronic circuitry, integrated circuitry, specially designed application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs) computer hardware, firmware, software, and/or combinations thereof, as realized and/or implemented in one or more machines (*e.g.*, one or more computing devices that are utilized as a user computing device for an electronic document, one or more server devices, such as a document server, etc.) programmed according to the teachings of the present specification, as will be apparent to those of ordinary skill in the computer art. These various aspects or features may include implementation in one or more computer programs and/or software that are executable and/or interpretable on a programmable system including at least one programmable processor, which may be special or general purpose, coupled to receive data and instructions from, and to transmit data and instructions to, a storage system, at least one input device, and at least one output device. Appropriate software coding may readily be prepared by skilled programmers based on the teachings of the present disclosure, as will be apparent to those of ordinary skill in the software art. Aspects and implementations discussed above employing software and/or software modules may also include appropriate hardware for assisting in the implementation of the machine executable instructions of the software and/or software module.

Such software may be a computer program product that employs a machine-readable storage medium. A machine-readable storage medium may be any medium that is capable of storing and/or encoding a sequence of instructions for execution by a machine (*e.g.*, a computing device) and that causes the machine to perform any one of the methodologies and/or embodiments described herein. Examples of a machine-readable storage medium include, but are not limited to, a magnetic disk, an optical disc (*e.g.*, CD, CD-R, DVD, DVD-R, etc.), a magneto-optical disk, a read-only memory "ROM" device, a random access memory "RAM" device, a magnetic card, an optical card, a solid-state memory device, an EPROM, an EEPROM, Programmable Logic Devices (PLDs), and/or any combinations thereof. A machine-readable medium, as used herein, is intended to include a single medium as well as a collection of physically separate media, such as, for example, a collection of compact discs or one or more hard disk drives in combination with a computer memory. As used herein, a machine-readable storage medium does not include transitory forms of signal transmission.

Such software may also include information (*e.g.*, data) carried as a data signal on a data carrier, such as a carrier wave. For example, machine-executable information may be included as a data-carrying signal embodied in a data carrier in which the signal encodes a sequence of instruction, or portion thereof, for execution by a machine (*e.g.*, a computing device) and any related information (*e.g.*, data structures and data) that causes the machine to perform any one of the methodologies and/or embodiments described herein.

Examples of a computing device include, but are not limited to, an electronic book reading device, a computer workstation, a terminal computer, a server computer, a handheld device (*e.g.*, a tablet computer, a smartphone, etc.), a web appliance, a network router, a network switch, a network bridge, any machine capable of executing a sequence of instructions that specify an action to be taken by that machine, and any combinations thereof. In one example, a computing device may include and/or be included in a kiosk.

FIG. 9 shows a diagrammatic representation of one embodiment of a computing device in the exemplary form of a computer system 900 within which a set of instructions for causing a control system to perform any one or more of the aspects and/or methodologies of the present disclosure may be executed. It is also contemplated that multiple computing devices may be utilized to implement a specially configured set of instructions for causing one or more of the devices to perform any one or more of the aspects and/or methodologies of the present

disclosure. Computer system 900 includes a processor 904 and a memory 908 that communicate with each other, and with other components, via a bus 912. Bus 912 may include any of several types of bus structures including, but not limited to, a memory bus, a memory controller, a peripheral bus, a local bus, and any combinations thereof, using any of a variety of bus architectures.

Memory 908 may include various components (*e.g.*, machine-readable media) including, but not limited to, a random-access memory component, a read only component, and any combinations thereof. In one example, a basic input/output system 916 (BIOS), including basic routines that help to transfer information between elements within computer system 900, such as during start-up, may be stored in memory 908. Memory 908 may also include (*e.g.*, stored on one or more machine-readable media) instructions (*e.g.*, software) 920 embodying any one or more of the aspects and/or methodologies of the present disclosure. In another example, memory 908 may further include any number of program modules including, but not limited to, an operating system, one or more application programs, other program modules, program data, and any combinations thereof.

Computer system 900 may also include a storage device 924. Examples of a storage device (*e.g.*, storage device 924) include, but are not limited to, a hard disk drive, a magnetic disk drive, an optical disc drive in combination with an optical medium, a solid-state memory device, and any combinations thereof. Storage device 924 may be connected to bus 912 by an appropriate interface (not shown). Example interfaces include, but are not limited to, SCSI, advanced technology attachment (ATA), serial ATA, universal serial bus (USB), IEEE 1394 (FIREWIRE), and any combinations thereof. In one example, storage device 924 (or one or more components thereof) may be removably interfaced with computer system 900 (*e.g.*, via an external port connector (not shown)). Particularly, storage device 924 and an associated machine-readable medium 928 may provide nonvolatile and/or volatile storage of machine-readable instructions, data structures, program modules, and/or other data for computer system 900. In one example, software 920 may reside, completely or partially, within machine-readable medium 928. In another example, software 920 may reside, completely or partially, within processor 904.

Computer system 900 may also include an input device 932. In one example, a user of computer system 900 may enter commands and/or other information into computer system 900

via input device 932. Examples of an input device 932 include, but are not limited to, an alpha-numeric input device (*e.g.*, a keyboard), a pointing device, a joystick, a gamepad, an audio input device (*e.g.*, a microphone, a voice response system, etc.), a cursor control device (*e.g.*, a mouse), a touchpad, an optical scanner, a video capture device (*e.g.*, a still camera, a video camera), a touchscreen, and any combinations thereof. Input device 932 may be interfaced to bus 912 via any of a variety of interfaces (not shown) including, but not limited to, a serial interface, a parallel interface, a game port, a USB interface, a FIREWIRE interface, a direct interface to bus 912, and any combinations thereof. Input device 932 may include a touch screen interface that may be a part of or separate from display 936, discussed further below. Input device 932 may be utilized as a user selection device for selecting one or more graphical representations in a graphical interface as described above.

A user may also input commands and/or other information to computer system 900 via storage device 924 (*e.g.*, a removable disk drive, a flash drive, etc.) and/or network interface device 940. A network interface device, such as network interface device 940, may be utilized for connecting computer system 900 to one or more of a variety of networks, such as network 944, and one or more remote devices 948 connected thereto. Examples of a network interface device include, but are not limited to, a network interface card (*e.g.*, a mobile network interface card, a LAN card), a modem, and any combination thereof. Examples of a network include, but are not limited to, a wide area network (*e.g.*, the Internet, an enterprise network), a local area network (*e.g.*, a network associated with an office, a building, a campus or other relatively small geographic space), a telephone network, a data network associated with a telephone/voice provider (*e.g.*, a mobile communications provider data and/or voice network), a direct connection between two computing devices, and any combinations thereof. A network, such as network 944, may employ a wired and/or a wireless mode of communication. In general, any network topology may be used. Information (*e.g.*, data, software 920, etc.) may be communicated to and/or from computer system 900 via network interface device 940.

Computer system 900 may further include a video display adapter 952 for communicating a displayable image to a display device, such as display device 936. Examples of a display device include, but are not limited to, a liquid crystal display (LCD), a cathode ray tube (CRT), a plasma display, a light emitting diode (LED) display, and any combinations thereof. Display adapter 952 and display device 936 may be utilized in combination with processor 904

to provide graphical representations of aspects of the present disclosure. In addition to a display device, computer system 900 may include one or more other peripheral output devices including, but not limited to, an audio speaker, a printer, and any combinations thereof. Such peripheral output devices may be connected to bus 912 via a peripheral interface 956. Examples of a peripheral interface include, but are not limited to, a serial port, a USB connection, a FIREWIRE connection, a parallel connection, and any combinations thereof.

The foregoing has been a detailed description of illustrative embodiments of the invention. Various modifications and additions can be made without departing from the spirit and scope of this invention. Features of each of the various embodiments described above may be combined with features of other described embodiments as appropriate in order to provide a multiplicity of feature combinations in associated new embodiments. Furthermore, while the foregoing describes a number of separate embodiments, what has been described herein is merely illustrative of the application of the principles of the present invention. Additionally, although particular methods herein may be illustrated and/or described as being performed in a specific order, the ordering is highly variable within ordinary skill to achieve embodiments as disclosed herein. Accordingly, this description is meant to be taken only by way of example, and not to otherwise limit the scope of this invention.

In the descriptions above and in the claims, phrases such as “at least one of” or “one or more of” may occur followed by a conjunctive list of elements or features. The term “and/or” may also occur in a list of two or more elements or features. Unless otherwise implicitly or explicitly contradicted by the context in which it is used, such a phrase is intended to mean any of the listed elements or features individually or any of the recited elements or features in combination with any of the other recited elements or features. For example, the phrases “at least one of A and B;” “one or more of A and B;” and “A and/or B” are each intended to mean “A alone, B alone, or A and B together.” A similar interpretation is also intended for lists including three or more items. For example, the phrases “at least one of A, B, and C;” “one or more of A, B, and C;” and “A, B, and/or C” are each intended to mean “A alone, B alone, C alone, A and B together, A and C together, B and C together, or A and B and C together.” In addition, use of the term “based on,” above and in the claims is intended to mean, “based at least in part on,” such that an unrecited feature or element is also permissible.

The subject matter described herein can be embodied in systems, apparatus, methods, and/or articles depending on the desired configuration. The implementations set forth in the foregoing description do not represent all implementations consistent with the subject matter described herein. Instead, they are merely some examples consistent with aspects related to the described subject matter. Although a few variations have been described in detail above, other modifications or additions are possible. In particular, further features and/or variations can be provided in addition to those set forth herein. For example, the implementations described above can be directed to various combinations and sub-combinations of the disclosed features and/or combinations and sub-combinations of several further features disclosed above. In addition, the logic flows depicted in the accompanying figures and/or described herein do not necessarily require the particular order shown, or sequential order, to achieve desirable results. Other implementations may be within the scope of the following claims.

The current application discloses the following concepts:

1. A decoder, the decoder comprising circuitry configured to:
receive a reference frame;
determine, for a current block, a scaling constant;
determine a scaled reference block using the reference frame and the scaling constant;
determine a scaled prediction block using the scaled reference block; and
reconstruct pixel data of the current block and using the rescaled prediction block.
2. The decoder of concept 1, further configured to determine the scaled prediction block by determining scaled motion information.
3. The decoder of concept 2, wherein the scaled motion information is determined by at least multiplying a horizontal component of a motion vector by the scaling constant and multiplying a vertical component of the motion vector by the scaling constant.
4. The decoder of concept 1, further configured to determine the scaled reference block by determining a position of the scaled reference block.
5. The decoder of concept 4, wherein the position is determined by multiplying a horizontal component of a position of a reference block by the scaling factor and multiplying a vertical component of the position of the reference block by the scaling factor.
6. The decoder of concept 1, further configured to:
receive a bit stream; and
determine, from the bit stream, the scaling constant, wherein the scaling constant is signaled within the bit stream.
7. The decoder of concept 6, wherein the bit stream includes an index to a predetermined scaling constant.
8. The decoder of concept 1, further configured to receive the reference frame by accessing the reference frame from memory.
9. The decoder of concept 1, further comprising:
an entropy decoder processor configured to receive a bit stream and decode the bit stream into quantized coefficients;
an inverse quantization and inverse transformation processor configured to process the quantized coefficients including performing an inverse discrete cosine;

a deblocking filter;

a frame buffer; and

an intra prediction processor.

10. The decoder of concept 1, wherein the current block is a coding tree unit.

11. The decoder of concept 1, wherein the current block is a coding unit.

12. The decoder of concept 1, wherein the current block is a prediction unit.

13. A method comprising:

receiving a reference frame;

determining, for a current block, a scaling constant;

determining a scaled reference block using the reference frame and the scaling constant;

determining a scaled prediction block using the scaled reference block; and

reconstructing pixel data of the current block and using the rescaled prediction block.

14. The method of concept 13, wherein determining the scaled prediction block includes determining scaled motion information.

15. The method of concept 14, wherein the scaled motion information is determined by multiplying a horizontal component of a motion vector by the scaling constant and multiplying a vertical component of the motion vector by the scaling constant.

16. The method of concept 13, wherein determining the scaled reference block includes determining a position of the scaled reference block.

17. The method of concept 16, wherein the position is determined by multiplying a horizontal component of a position of a reference block by the scaling factor and multiplying a vertical component of the position of the reference block by the scaling factor.

18. The method of concept 13, further comprising:

receiving a bit stream; and

determining, from the bit stream, the scaling constant, wherein the scaling constant is signaled within the bit stream.

19. The method of concept 18, wherein the bit stream includes an index to a predetermined scaling constant.

20. The method of concept 13, wherein receiving the reference frame includes accessing the reference frame from memory.

21. The method of concept 13, wherein at least one of the receiving, the determining, and the reconstructing is performed by a decoder, the decoder comprising:

an entropy decoder processor configured to receive a bit stream and decode the bit stream into quantized coefficients;

an inverse quantization and inverse transformation processor configured to process the quantized coefficients including performing an inverse discrete cosine;

a deblocking filter;

a frame buffer; and

an intra prediction processor.

22. The method of concept 13, wherein the current block is a coding tree unit.

23. The method of concept 13, wherein the current block is a coding unit.

24. The method of concept 13, wherein the current block is a prediction unit.

What is claimed is:

1. An encoder configured to:

receive a video signal;

process the video signal to generate an encoded bitstream comprising a reference frame, a current picture having a resolution different from a resolution of the reference frame and including a first block and a second block, and information useful for determining a scaling constant, the encoder further configuring the bitstream to be decoded by a method including:

determining a scaling constant using the information in the bitstream;

determining a scaled prediction block from the reference frame by:

determining a location of the scaled prediction block by scaling a motion vector component using the scaling constant; and

scaling a resolution of a block of the reference frame using the scaling constant and applying a filter determined for the scaled prediction block and not determined for the whole reference frame;

reconstructing the first block using the scaled prediction block; and

reconstructing the second block without using a scaled prediction block.

2. The encoder of claim 1 wherein, the information in the bitstream useful for determining a scaling constant includes an index to a set of predetermined values and said index is utilized to determine the scaling constant.

3. An encoder configured to:

receive a video signal;

process the video signal to generate an encoded bit stream including a current coded picture comprising a first plurality of coded blocks and a second plurality of coded blocks, the encoder further configuring the encoded bitstream to be decoded by a method including:

decoding each block of the first plurality of coded blocks by:

determining a scaling constant using information in the bitstream; and

determining a scaled prediction block from a reference frame having a different resolution than a resolution of the current coded picture, by scaling a motion vector component to determine a location of the scaled prediction block, scaling a block from the reference frame using the scaling constant, and applying a filter determined individually for the scaled prediction block;

decoding each block of the first plurality of coded blocks using its scaled prediction block; and

decoding each block of the second plurality of coded blocks without using a scaled prediction block of a reference frame and without applying a resolution changing filter determined for a block of a reference frame.

4. The encoder of claim 3 wherein, the information in the bitstream includes an index to a set of predetermined values and said index is utilized to determine the scaling constant.

5. A video encoder having circuitry configured to receive a video signal and encode a bitstream comprising a reference frame, a current picture having a resolution different from the reference frame and including a first block and a second block, the second block being an intra-coded block, and information useful for determining a scaling constant, the encoded bitstream being configured to be decoded by a method comprising:

determining whether a block-based mode for decoding a current picture of one resolution using a reference frame of a different resolution is enabled;

determining a scaling constant using the information in the bitstream;

using the enabled mode, determine a scaled prediction block from the reference frame by:

determining a location of the scaled prediction block by scaling a motion vector component using the scaling constant; and

scaling the resolution of a block of the reference frame using the scaling constant and applying a filter determined specifically for the prediction block and not determined for the whole reference frame;

reconstructing the first block using the scaled prediction block; and

reconstructing the intra-coded second block without using the mode.

6. A video encoder having circuitry configured to encode a bitstream including a reference frame, a current picture having a resolution different from the reference frame and including a first block and a second block, the second block being an intra-coded block, signaling information for determining whether a block-based mode for decoding a current picture of one resolution using a reference frame of a different resolution is enabled, and information for determining a scaling constant, the encoder further configuring the bitstream to be decodable by a method comprising:

receiving the encoded bitstream;

determining a scaling constant using the information in the bitstream;

using the enabled mode, determine a scaled prediction block from the reference frame by:

determining a location of the scaled prediction block by scaling a motion vector component using the scaling constant; and

scaling the resolution of a block of the reference frame using the scaling constant and applying a filter determined specifically for the prediction block and not determined for the whole reference frame;

reconstructing the first block using the scaled prediction block; and

reconstructing the intra-coded second block without using the mode.

7. A non-transitory computer readable media storing an encoded bitstream comprising a reference frame, a current picture having a resolution different from the reference frame and including a first block and a second block, the second block being an intra-coded block, and information useful for determining a scaling constant, the encoded bitstream being decodable by a video decoder configured to:

receive the encoded bitstream;

determine whether a block-based mode for decoding a current picture of one resolution using a reference frame of a different resolution is enabled;

determine a scaling constant using the information in the bitstream;

using the enabled mode, determine a scaled prediction block from the reference frame by:

determining a location of the scaled prediction block by scaling a motion vector component using the scaling constant; and

scaling the resolution of a block of the reference frame using the scaling constant and applying a filter determined specifically for the prediction block and not determined for the whole reference frame;

reconstruct the first block using the scaled prediction block; and

reconstruct the intra-coded second block without using the mode.

8. A non-transitory computer readable media storing an encoded bitstream including a reference frame, a current picture having a resolution different from the reference frame and including a first block and a second block, the second block being an intra-coded block, signaling information for determining whether a block-based mode for decoding a current picture of one resolution using a reference frame of a different resolution is enabled, and information for determining a scaling constant, the encoded bitstream being decodable by a video decoder configured to:

receive the encoded bitstream;

determine a scaling constant using the information in the bitstream;

using the enabled mode, determine a scaled prediction block from the reference frame by:

determining a location of the scaled prediction block by scaling a motion vector component using the scaling constant; and

scaling the resolution of a block of the reference frame using the scaling constant and applying a filter determined specifically for the prediction block and not determined for the whole reference frame;

reconstruct the first block using the scaled prediction block; and

reconstruct the intra-coded second block without using the mode.