Innovation, Sciences et Développement économique Canada
Office de la Propriété Intellectuelle du Canada

Innovation, Science and Economic Development Canada
Canadian Intellectual Property Office

CA 2786910 C 2020/06/30

(11)(21) 2 786 910

(12) BREVET CANADIEN
CANADIAN PATENT

(13) C

(86) **Date de dépôt PCT/PCT Filing Date:** 2011/02/05
(87) **Date publication PCT/PCT Publication Date:** 2011/08/18
(45) **Date de délivrance/Issue Date:** 2020/06/30
(85) **Entrée phase nationale/National Entry:** 2012/07/11
(86) **N° demande PCT/PCT Application No.:** US 2011/023839
(87) **N° publication PCT/PCT Publication No.:** 2011/100174
(30) **Priorité/Priority:** 2010/02/11 (US12/704,047)

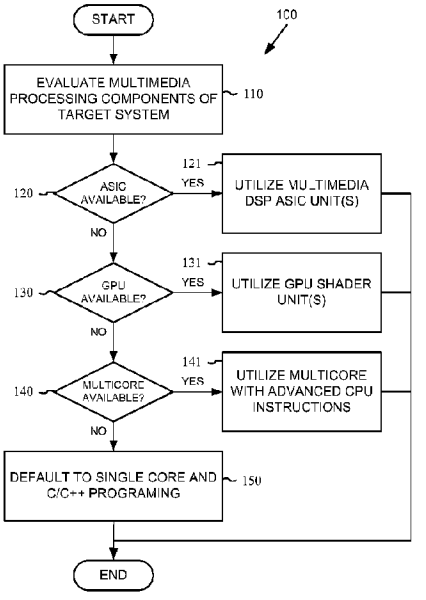
(51) **Cl.Int./Int.Cl.** ***H04N 5/235*** (2006.01)

(72) **Inventeurs/Inventors:**
WU, YONGJUN, US;
BORISOV, NIKOLA, US;
ZHAO, WEIDONG, US;
SADHWANI, SHYAM, US;
THUMPUDI, NAVEEN, US

(73) **Propriétaire/Owner:**
MICROSOFT TECHNOLOGY LICENSING, LLC, US

(74) **Agent:** SMART & BIGGAR LLP

(54) **Titre : STABILISATION D'IMAGE VIDEO DE PLATEFORME GENERIQUE**
(54) **Title: GENERIC PLATFORM VIDEO IMAGE STABILIZATION**



(57) **Abrégé/Abstract:**
Video image stabilization provides better performance on a generic platform for computing devices by evaluating available multimedia digital signal processing components, and selecting the available components to utilize according to a hierarchy structure for video stabilization performance for processing parts of the video stabilization. The video stabilization has improved motion vector estimation that employs refinement motion vector searching according to a pyramid block structure relationship starting from a downsampled resolution version of the video frames. The video stabilization also improves global motion transform estimation by performing a random sample consensus approach for processing the local motion vectors, and selection criteria for motion vector reliability. The video stabilization achieves the removal of hand shakiness smoothly by real-time one-pass or off-line two-pass temporal smoothing with error detection and correction.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 August 2011 (18.08.2011)

PCT

(10) International Publication Number
WO 2011/100174 A3

(51) **International Patent Classification**:
H04N 5/235 (2006.01)

(21) **International Application Number**:
PCT/US2011/023839

(22) **International Filing Date**:
5 February 2011 (05.02.2011)

(25) **Filing Language**: English

(26) **Publication Language**: English

(30) **Priority Data**:
12/704,047 11 February 2010 (11.02.2010) US

(71) **Applicant** *(for all designated States except US)*: **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) **Inventors**: **WU, Yongjun**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **BORISOV, Nikola**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **ZHAO, Weidong**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **SADHWANI, Shyam**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **THUMPUDI, Naveen**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *with international search report (Art. 21(3))*

[Continued on next page]

(54) **Title**: GENERIC PLATFORM VIDEO IMAGE STABILIZATION

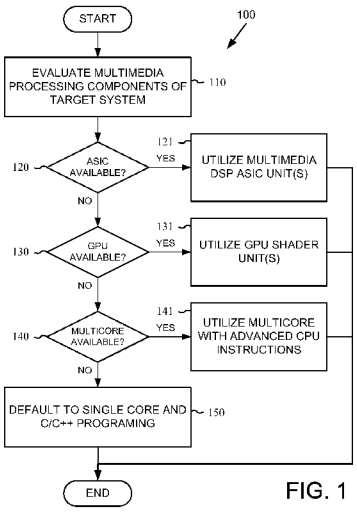


FIG. 1

(57) **Abstract**: Video image stabilization provides better performance on a generic platform for computing devices by evaluating available multimedia digital signal processing components, and selecting the available components to utilize according to a hierarchy structure for video stabilization performance for processing parts of the video stabilization. The video stabilization has improved motion vector estimation that employs refinement motion vector searching according to a pyramid block structure relationship starting from a downsampled resolution version of the video frames. The video stabilization also improves global motion transform estimation by performing a random sample consensus approach for processing the local motion vectors, and selection criteria for motion vector reliability. The video stabilization achieves the removal of hand shakiness smoothly by real-time one-pass or off-line two-pass temporal smoothing with error detection and correction.

WO 2011/100174 A3

WO 2011/100174 A3

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

(88) Date of publication of the international search report:
17 November 2011

GENERIC PLATFORM VIDEO IMAGE STABILIZATION

Background

[001] It is increasingly common for video camera capabilities to be incorporated into multi-purpose, small and light-weight handheld electronic devices, such as mobile phones, compact cameras and media players. Unlike the larger dedicated-purpose video camcorders, such multi-purpose handheld devices typically lack any mechanical or optical mechanism to reduce jittery video motion due to a shaky or unsteady hand. Due to their lighter weight and typical use of a wide angle lens, the multi-purpose handheld devices can be more susceptible to jittery video motion from hand shakiness. Further, as the availability of these inexpensive multi-purpose handheld devices spreads the popularity of shooting video beyond the community of amateur and professional videographers, consumer video is more and more commonly produced by users with very little training or experience in how to shoot quality video. There is also a growing popularity among consumers to share short video clips over the Internet via email, blogs, video sharing web sites and social networking sites. Particularly for those users who are not videography professionals or hobbyists, the video clips are often uploaded to the web site (sometimes directly from the video capture device) without any editing or other processing. For these various reasons, the quality of video clips shared on the web is very often quite poor, and the video clips commonly suffer from jittery video motion due to hand shakiness.

[002] Digital video image stabilization is a digital signal processing technique that can be applied to video to correct jittery video motion from hand shakiness. In one exemplary implementation, the technique involves calculating local motion estimation for macro blocks of each image of the video sequence relative to its preceding image; processing the local motion vectors of the macro blocks to produce an estimate of the global motion due to jitter; and then compensating for the jittery video motion by digital shifting or warping the image in a direction opposite to the estimated jitter motion.

[003] One drawback of known digital video image stabilization is that the technique is quite computationally intensive. When uploading video to a blog, video sharing web site or social networking site, the video may be uploaded from devices that vary in processing capabilities. Moreover, the casual user may be more interested in

immediacy of quickly posting their video to a video sharing or social networking site, such that any time consuming processing of the video is undesirable. For example, video may be uploaded directly from a multi-purpose handheld device, such as over a cellular network. However, the multi-purpose handheld device (such as a mobile phone) often has limited processing resources, or must share processing resources with other capabilities of the multi-purpose device. Alternatively, the handheld device also could be connected to a PC, laptop, netbook or like devices with internet connectivity to upload video to a video sharing or social networking site. However, these internet-connected PCs also vary greatly in processing resources. Also, in the interest of immediacy, any processing delays during posting via an internet-connected device can be undesirable.

[004] For these reasons, a video image stabilization technique that operates effectively across the various available computing platforms would be useful and desired.

Summary

[005] In summary, the detailed description relates generally to digital video processing for image stabilization of a video sequence.

[006] The following Detailed Description pertains to systems and methods that implement video stabilization on generic platform, which effectively utilizes available multimedia digital signal processing components to provide better performance. For example, the video stabilization on generic platform is programmed in an executable library that provides programming functions to utilize any of various multimedia digital signal processing components that may be provided on computing devices having the generic platform. The implementation of the video stabilization operates on the generic platform by evaluating the computing device on which it is being run for any available multimedia digital signal processing components on the computing device. The implementation of the video stabilization on generic platform then determines which available multimedia digital signal processing component or components to utilize for processing one or more parts of the video stabilization according to a hierarchy structure that orders multimedia digital signal processing components by criteria including at least performance for video image stabilization. The implementation then processes a video for video image stabilization including executing the executable library functions utilizing the determined multimedia digital signal processing components on those parts of processing of the video image stabilization.

[007] In some implementations presented in the following Detailed Description, the video image stabilization uses a form of local motion vector estimation that emphasizes consistency of motion vectors among neighboring blocks of a video frame. In this local motion vector estimation, the frame and its reference frame are down sampled one or more times to form a plurality of versions of the frame and reference frame at a plurality of resolutions. A lowest resolution version of the frame has a plurality of blocks associated as a pyramid structure to a successively larger number of blocks at corresponding locations in each higher resolution version of the frame. The motion vector estimation process begins with the lowest resolution version of the frame, estimating motion vectors of the blocks of the lowest resolution version frame using a full search of the associated lowest resolution version reference frame. Then, for each successively higher resolution version of the frame, the motion vectors for the blocks of the higher resolution version frame are estimated using a refinement search starting from the motion vector estimated for the associated block of the preceding lower resolution version frame according to the pyramid structure. After repeating for each higher resolution version up to the full resolution version of the frame, this produces estimated local motion vectors for the blocks of the full resolution version frame that better emphasize consistency among neighboring blocks. This more consistent estimate of local motion vectors better emphasizes the global motion of the frame.

[008] In some implementations presented in the following Detailed Description, the video image stabilization includes ways to process motion vectors for more reliably estimating global motion of frames in the video. These include selecting a set of motion vectors that more reliably reflect the frames global motion, such as by eliminating motion vectors on picture boundaries, eliminating motion vectors that produce high motion compensation residuals, and motion vectors of blocks with low variance. The video image stabilization also may include detection and removal of motion vector outliers by estimating parameters of a global motion transform using a random sample consensus approach. The video image stabilization also may estimate the global motion transform based on a similarity motion model that may better model jitter video motion from hand shakiness.

[009] In some implementations presented in the following Detailed Description, the video image stabilization may also subject the estimation of the global motion transform corresponding to jittery video motion to limit thresholds for detection and correction of errors in the global motion estimation. The video image stabilization

compares the parameters of the estimated global motion transform to the two limits. If the upper limit is exceeded, the estimate of the global motion transform is reset. If the parameters exceed lower limit, the parameters of the estimated global motion transform are limited to their lower limit. Otherwise, the estimated global motion transform is used unchanged.

[010] In some implementations presented in the following Detailed Description, the video image stabilization performs temporal smoothing of the global motion transforms estimated for the frames.

[011] The following Detailed Description presents variations of the video image stabilization that may be employed in real time playback or transcoding scenarios, as well as variations suitable for off-line video stabilization. In the real-time mode, the video image stabilization may be performed using a single pass processing of the local motion vectors to produce the estimates of global motion of the frames. In the off-line mode, the video image stabilization may perform the motion vector processing in two passes. For example, the first motion vector processing pass may be used to gather statistical information about the motion vectors, such as probability distributions of the motion vectors across the frames of the scene of the video segment. This statistical information can be used to adaptively determine the limits applied to the global motion transform estimation. The information from a first pass also may be used in the temporal smoothing of the global motion transforms. For example, the information can be used to determine constraint limits for constrained global optimization of the temporal smoothing. In addition, the first pass in off line mode also can be used for detecting scene changes in a video segment.

[012] This Summary is provided to introduce a selection of concepts in a simplified form that is further described below in the Detailed Description. This summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter. Additional features and advantages of the invention will be made apparent from the following detailed description of embodiments that proceeds with reference to the accompanying drawings.

[012a] According to one aspect of the present invention, there is provided one or more computer-readable media storing computer-executable instructions for causing a computing device programmed thereby to perform a method of digital video stabilization, the one or more computer-readable media including non-volatile memory or a storage device, the method comprising: evaluating the computing device to determine availability of any of various multimedia digital signal processing ("DSP") components on the computing device; determining from the results of said evaluating which available multimedia DSP component to utilize for at least one part of digital video stabilization according to a hierarchical structure for ordering multimedia DSP components by criteria comprising at least performance of the digital video stabilization, the digital video stabilization comprising at least local motion vector estimation to produce estimated local motion vectors, processing the estimated local motion vectors for estimation of a global motion transform representing jittery video motion, and image warping based on the global motion transform to compensate the jittery video motion; and performing digital video stabilization of a video segment on the computing device utilizing the determined available multimedia DSP component for said at least one part of the digital video stabilization, wherein the performing the digital video stabilization includes warping at least one frame of the video segment based on the global motion transform that applies for the at least one frame of the video segment.

[012b] According to another aspect of the present invention, there is provided a method of digital video stabilization, the method comprising performing steps on a computing device of: performing local motion vector estimation for a frame of a plurality of frames of a scene of a video segment, wherein the local motion vector estimation comprises for said frame: downsampling said frame and a reference frame associated with said frame by one or more times to form a plurality of versions of said frame at a plurality of resolutions, wherein a lowest resolution version of said frame has a plurality of blocks associated as a pyramid structure to a successively larger number of blocks at corresponding locations in each higher resolution version of said frame; for the lowest resolution version of said frame, estimating motion vectors of the blocks of said lowest resolution version frame using an initial motion vector estimation processing of the associated lowest resolution version reference frame; for each successively higher resolution version of said frame, estimating motion vectors of the

blocks of said successively higher resolution version frame using a refinement search starting from the motion vector estimated for the associated block of the preceding lower resolution version frame according to the pyramid structure; and producing estimated local motion vectors for the blocks of the original resolution version of said frame; processing the estimated local motion vectors of said frame for estimation of a global motion transform representing jittery video motion, including selecting a set of motion vectors to use for estimation of the global motion transform out of the produced estimated local motion vectors of said frame, wherein said selecting the set of motion vectors comprises one or more of: excluding motion vectors on picture boundaries, excluding motion vectors having a high motion compensation residual, and excluding motion vectors for blocks with low image content variation; and performing image warping on said frame based on the global motion transform estimated for said frame to compensate for the jittery video motion.

[012c] According to still another aspect of the present invention, there is provided a digital video processing device for processing video to apply digital video stabilization to the video, the digital video processing device comprising: a memory storing a generic platform video stabilization library program; at least one digital signal processing component; a processing unit operating to execute the video stabilization library program from the memory, wherein said execution of the video stabilization library program comprises: evaluating the at least one digital signal processing component of the digital video processing device; determining which of the at least one digital signal processing component to utilize for at least one part of digital video stabilization according to a hierarchical structure for ordering digital signal processing components by criteria comprising at least performance of the digital video stabilization; and performing digital video stabilization on the video in part by processing a frame of the video to estimate local motion vectors of the frame, processing the estimated local motion vectors for estimation of a global motion transform representing jittery video motion, and warping the frame based on the global motion transform to compensate for the jittery video motion, wherein said performing the digital video stabilization executes programming functions in the generic platform video stabilization library program to utilize the determined digital signal processing component for the at least one part of the digital video stabilization.

[012d] According to yet another aspect of the present invention, there is provided a handheld computing device comprising: a central processing unit (CPU); one or more memory units; a camera; a graphics processing unit (GPU) including a shader unit; and the one or more memory units storing computer-executable instructions for causing the handheld computing device, when programmed thereby, to perform real-time digital video stabilization that includes: for each of multiple frames of a video sequence captured by the camera: estimating, using the CPU, a motion transform that represents jittery motion associated with shakiness of the handheld computing device; and automatically warping, using the shader unit of the GPU, at least a portion of the frame based on the motion transform to compensate for the jittery motion; and initiating uploading of the stabilized video, from the handheld computing device to a video sharing site or social networking site, over a connection to a network.

[012e] According to a further aspect of the present invention, there is provided a computer-readable memory or storage device storing computer-executable instructions for causing a computing device that includes a graphics processing unit (GPU), when programmed thereby, to perform video sharing with real-time digital video stabilization, the video sharing with real-time digital video stabilization including: for each of multiple frames of a video sequence captured by a camera of the computing device: estimating a motion transform that represents jittery motion associated with shakiness of the computing device; and automatically warping, using a vertex shader on the GPU, at least a portion of the frame based on the motion transform to compensate for the jittery motion; forming stabilized video around a display port, including, for each of at least some of the multiple frames of the video sequence, cropping boundaries of the frame, wherein the display port depends on the warping for the multiple frames, respectively; and initiating uploading of the stabilized video to a video sharing site or social networking site over a connection to a network.

[012f] According to yet a further aspect of the present invention, there is provided in a computing device that includes a graphics processing unit (GPU), a method of video sharing with real-time digital video stabilization, the method comprising: for each of multiple frames of a video sequence captured by a camera of the computing device: estimating a motion transform that represents jittery motion associated with shakiness of the computing

device; and automatically warping, using a vertex shader on the GPU, at least a portion of the frame based on the motion transform to compensate for the jittery motion; forming stabilized video around a display port, including, for each of at least some of the multiple frames of the video sequence, cropping boundaries of the frame, wherein the display port depends on the warping for the multiple frames, respectively; and initiating uploading of the stabilized video, from the computing device to a video sharing site or social networking site, over a connection to a network.

[012g] According to still a further aspect of the present invention, there is provided a method of real-time sharing of stabilized digital video for multiple frames of a captured video sequence, comprising: warping at least a portion of a frame, among the multiple frames of the captured video sequence, based on a motion transform that represents jittery motion of a video capture device to compensate for the jittery motion of the video capture device; and initiating uploading of stabilized video from the video capture device to a server device associated with a service for video sharing or social networking; wherein the motion transform has been estimated prior to warping.

[012h] According to another aspect of the present invention, there is provided a device comprising: one or more processing units; one or more memory units; a camera; and the one or more memory units storing computer-executable instructions for causing the device, when programmed thereby, to perform real-time digital video stabilization that includes: for each of multiple frames of a video sequence captured by the camera: estimating, using at least one of the processing units, a motion transform that represents jittery motion of the device; and warping at least a portion of the frame based on the motion transform to compensate for the jittery motion; and initiating transfer of stabilized video to a social media storage location in network communication with the device.

[012i] According to yet another aspect of the present invention, there is provided computer-readable memory storing computer-executable instructions that, when executed, cause a computing device to perform video sharing with real-time digital video stabilization comprising: capturing multiple frames of a video sequence into a captured video sequence; warping at least a portion of a frame, among the multiple frames of the captured video sequence, to compensate for the jittery motion of a video capture device; and initiating uploading of stabilized video from the video capture device to a social networking or video sharing website; wherein the warping is based on an estimated motion transform that represents jittery motion.

Brief Description Of The Drawings

[013] Figure 1) is a flow diagram of a method for a generic platform digital video image stabilization technique to most effectively utilize available multimedia digital signal processing components according to a computation performance hierarchy.

[014] Figure 2) is a flow diagram of a three part digital video image stabilization process, which utilizes available multimedia digital signal processing components of the target computing device according to the hierarchical utilization process of Figure 1).

[015] Figure 3) is a diagram illustrating down sampling of a video frame for pyramid block based local motion estimation.

[016] Figures 4), 5), 6) and 7) are diagrams illustrating processing of an example video frame for pyramid block based local motion estimation.

[017] Figure 8) is a diagram illustrating global motion transform models, including a similarity model of global motion transform used for digital video image stabilization.

[018] Figure 9) is a flow diagram illustrating a single pass processing of the motion vectors for a video frame to estimate a global motion transform of jitter motion due to hand shakiness, such as for use in a real time mode digital video image stabilization.

[019] Figure 10) is a flow diagram illustrating a two pass processing of the motion vectors for a video frame to estimate a global motion transform of jitter motion due to hand shakiness, such as for use in an off-line mode digital video image stabilization.

[020] Figure 11) is a block diagram of a computing environment in which the digital video image stabilization techniques may be practiced.

[021] Figures 12) and 13) are block diagram of a cloud computing network environment in which the digital video image stabilization techniques may be practiced.

Detailed Description

[022] The following detailed description concerns systems and techniques to provide digital video image stabilization, and more particularly concerns digital video image stabilization techniques operable on a generic computation hardware platform while effectively utilizing available multimedia digital signal processing (DSP) hardware components. The digital video image stabilization techniques can be practiced across a variety of hardware devices, including handheld and portable computing devices, video cameras, mobile phones, entertainment consoles (e.g., video game consoles and television

set-top box), various network client computers (e.g., personal computers, laptop, netbook, and tablet computers), as well as server computers. The digital video image stabilization techniques can also be used in a variety of usage and computation scenarios, including video processing on a dedicated video capture device, video processing on a stand-alone computer, video processing on a network client computer and video processing on a server computer. Further, various parts of the digital video image stabilization technique can be performed in parallel or cooperatively on multiple computing devices, such as in a client/server, network "cloud" service or peer computing arrangement, among others. Accordingly, it should be recognized that the techniques can be realized on a variety of different electronic and computing devices, including both end user consumer operated devices as well as server computers that may provide the techniques as part of a service offering to customers.

Hierarchical Selection of Multimedia DSP Components On Generic Platform for Digital Video Image Stabilization

[023] The processing of video, and more particularly the processing of the digital video image stabilization techniques described herein, is inevitably computationally intensive. On the other hand, the digital video image stabilization techniques can be practiced on a variety of computing devices, whose configuration of multimedia processing hardware components can vary widely from each other. One way for the digital video image stabilization techniques described herein to achieve better potential performance and quality is that the techniques evaluate the target computing device on which they are run, and choose to utilize available multimedia processing components according to a hierarchy constructed in a way that considers one or more aspects of performance, quality, power consumption, conformance, and robustness. In this way, the digital video image stabilization techniques are developed for a generic operating platform, and then adapt to best utilize multimedia processing capabilities of the actual target computing hardware on which they are run.

[024] In one example implementation, the digital video image stabilization techniques described below are implemented in an executable program to be run on a computing device (described in more detail below), such as a dynamic link library file (DLL) or as an application program executable file. When implemented as a DLL or other executable library file, the executable program implements the digital video image

stabilization as a set of library functions, and provides a programmatic interface for application programs to make programmatic use of the digital video image stabilization functionality implemented by the executable program. The executable program runs on a generic platform or computing environment, which is to say that the executable program can run on a variety of computers and computing devices that may include varying multimedia digital signal processing (DSP) components.

[025] With reference to Figure 1), the digital video image stabilization library program includes programming to perform video processing for the digital video image stabilization using various multimedia DSP components that potentially may be available on the target computing device on which the program is run. The program then adapts to best utilize the multimedia DSP components via the multimedia utilization hierarchy process 100. In a first action 110 of this process 100, the library program evaluates the multimedia DSP components of the target computing system or device on which it has been installed and is being run. The library program can perform this evaluation by making a call to an operating system service for querying system information, by examining system information recorded in a registry or other system database, by calling programming interfaces associated with multimedia DSP components, by attempting access to multimedia DPS components or by other like ways to access system information and/or query the presence of hardware components.

[026] Based on the results of the evaluation in action 110, the library program then chooses from the multimedia DSP components determined to be available on the target computing device according the processing hierarchy. In the illustrated example, the library program first chooses to utilize any available ASIC units that are adapted for multimedia DSP in the target computing device as shown in actions 120-121. Examples of current commercially available such ASIC units include video cards from Intel (known by the name Larrabee) and Nvidia (named Tesla). If no such ASIC units are available on the target device, the library program next chooses to utilize any available graphics processing unit with shader capability on the target device as shown in actions 130-131. If no ASIC or GPU shader components are present, the library program chooses to utilize any available multi-core processor or central processing unit with advanced multimedia instruction set (e.g., SSEx instructions). Finally, if none of these higher performance multimedia DSP components are available on the target device, the library program falls back to performing video processing of the digital video image stabilization techniques on a single core CPU using C/C++ programming as shown in action 150. In other alternative

implementations, other hierarchies of multimedia DSP components can be used based on consideration of the above-mentioned aspects of performance, quality, power consumption, conformance, and robustness for multimedia DSP components that may then be available in commercially available computing device hardware of the generic platform. Such alternative implementations can include utilization of fewer or additional multimedia DSP components than in the illustrated hierarchical multimedia DSP utilization process 100.

Digital Video Image Stabilization On Generic Platform

[027] As illustrated in Figure 2), the digital video image stabilization techniques performs processing of a subject video involving generally three parts: local motion estimation 210, motion vector processing to estimate a global motion transform corresponding to video jitter from hand shakiness 220, and image warping to compensate the video jitter motion. The local motion estimation 210 and image warping 230 are computationally intensive, and most desirably are done using the choice from any available multimedia DSP components on the target computing device chosen according to the hierarchical DSP utilization shown in Figure 1) and discussed above. For example, a form of the local motion estimation 210 that is particularly suited for processing utilizing GPU shaders can be implemented in the library program and used to accomplish the local motion estimation part of the digital video image stabilization when the target computing device has a GPU shader capability among its available multimedia DPS components. On the other hand, the library program can include the functionality to perform the image warping 230 using a D3D API call when the evaluated target computing device is determined to posses that capability among its available multimedia DSP components. The second part 220, processing local motion vectors to estimate the global motion transform is done on the CPU in the example implementation.

Pyramid Block Based ME on GPGPU

[028] In one example implementation, the library program for the digital video image stabilization includes programming to implement the local motion estimation 210 using a pyramid block based motion estimation technique illustrated in Figures 3)-7) that is particularly well suited for processing in a GPU shader (when determined to be available on the target computing device by the process 100 discussed above). As

compared to individually estimating local motion for blocks of the full resolution video frames, the pyramid block-based motion estimation technique also emphasizes global motion of the video frame, which is well suited to estimating overall global motion of video frames for the digital video image stabilization.

[029] The pyramid block-based motion estimation performs motion estimation for each frame of the video relative to a reference frame. In general, the reference frame is a consecutive frame (e.g., the immediately preceding or following frame) in the video sequence. Alternatively, every third frame can be used as a reference for the next two frames of video, or other selection of reference frame can be made. For the motion estimation the video frame is divided into a regular grid of blocks, and the motion estimation calculates a motion vector or displacement of each block of the video frame relative to closely matching image content in the reference frame.

[030] The pyramid block-based motion estimation begins by repeatedly down-sampling the resolution of both the video frame and its reference frame by half using a down sampling shader on the GPU of the computing device. In the example shown in Figure 3), the video frame and reference frame are down sampled by half three times over to produce versions of the video frame and reference frame at full resolution, ½ resolution, ¼ resolution, and 1/8 resolution. The video frame and reference frame can be down sampled in resolution a fixed number of times, or can be down sampled a variable number of times dependent on the size and original resolution of the video frame according to available image buffer space and processing resources of the computing device. Alternatively, the video frame and reference frame can be down sampled fewer or more times than the example shown in Figure 3). It should be understood that the example illustrated in Figure 3) is illustrative only, and actual video frames processed by the digital video image stabilization technique generally would have a larger number of pixels and higher original resolution than the example shown.

[031] For each resolution of the video frame, the pyramid block-based motion estimation divides the video frame into a grid of blocks. The same size of block is used at each resolution. Preferably a relatively large block size, such as 16x16 or 8x8 pixels is used. In this way, each block of the lowest resolution image will split into 4 blocks of the same block size at the corresponding location in the next higher resolution image, which effectively provides a pyramid structure or hierarchy relationship of blocks in the lowest resolution image to those at the corresponding location in the higher resolution images. For purposes of illustration, the blocks in the higher resolution versions of the video frame

(i.e., 1/1, ½ and ¼ resolution) that are in a pyramid structure relationship to the upper-left block of the 1/8 resolution image are shown within the thick line boundary in Figure 3).

[032] Next, the pyramid block-based motion estimation performs a full search based on sum of absolute difference (SAD) on each block in the lowest resolution image over the full reference frame to find a position of matching image content in the reference frame. This produces a local motion vector of each block in the lowest resolution video frame corresponding to its displacement from matching image content in the reference frame. This full search is performed at the lowest resolution version of the video frame and reference frame only as shown in Figure 4).

[033] As illustrated in Figures 5)-7), the pyramid block-based motion estimation then performs refinement motion estimation searching at each successive higher resolution version of the video frame relative to the reference frame up to the original resolution video frame. As previously mentioned, each block of a lower resolution image splits into four blocks at the corresponding location of the next higher resolution version of the video frame. The motion vector calculated via the motion estimation search for the block in the lower resolution image is used as the starting point of refinement motion estimation searches for each of the four blocks in the pyramid structure at the next higher resolution version of the video frame. This refinement search using the pyramid structure relationship of the blocks in the lower resolution video frame to the four corresponding blocks in the next higher resolution video frame emphasizes consistency in the motion estimation calculation among neighboring blocks. As the refinement search is repeated for each successive higher resolution version of the video frame, this emphasized consistency among neighbors provides a more consistent set of local motion vectors for the blocks in the original resolution image upon which to base the global motion transform estimation discussed below.

[034] Once the GPU shader finishes the local motion estimation at the original video frame resolution, the motion vector along with the sum of differences result and variance for each block of the original resolution video frame is copied to the CPU for the global motion transform estimation.

Motion Vector Processing For Global Motion Transform Estimation

[035] With reference again to Figure 2), the motion vector processing to estimate global motion transform part 220 of the digital video image stabilization process 200 in

one example implementation is performed on the central processing unit (CPU) of the computing device. The digital video image stabilization techniques can use a single pass mode of the motion vector processing, which is suitable for a real time processing during video playback, video streaming or transmission. Alternatively, a two pass mode of the motion vector processing can be employed that provides a better video stabilization quality at a cost of increased computation time. The two pass mode therefore may be more suitable for off-line video processing scenarios, which are not subject to the time constraints of real time video playback and for which a high video stabilization quality is desired.

[036] Figure 8) illustrates various global motion transform models that could be used to estimate jittery video motion from hand shakiness, including translation similarity, Euclidean, project and affine motion transform models. In an example implementation presented herein, the motion vector processing uses the similarity global transform model with translation, rotation and zoom parameters as shown in the following equation (1):

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s\cos\beta & s\sin\beta & tx \\ -s\sin\beta & s\cos\beta & ty \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

where x and y are the original position, x' and y' are the transformed position, and s, β, t_x, t_y are zoom, rotation and translation parameters, respectively.

[037] In general, most users will experience shaky motion effects on translation, rotation and zoom only. Accordingly, the similarity motion transform model fits the application of video stabilization well to model the hand shakiness of the camera operator. However, alternative implementations of the global motion transform estimation could use other models that model additional or fewer aspects of video jitter from hand shakiness.

One Pass Motion Vector Processing In Real Time Mode

[038] Figure 9) illustrates an example implementation of a one pass motion vector processing 900 for estimating global motion transform of video frames, which is suitable to a real time processing mode. The motion vector processing is applied to the motion vectors for each video frame of a scene in a video sequence, such as those produced via the pyramid block-based motion estimation described above. The example implementation begins at action 910 by receiving information of the motion vectors for a video frame. In addition, the example motion vector processing 900 also receives

information of the residual value for each block relative to that of its matching block at the motion vector position in the reference frame, which is the sum of absolute differences (SAD) value calculated during the motion vector estimation. Further, the example motion vector processing 900 receives a value of the variance in image content of the original block in the video frame, which may also be calculated during the motion vector estimation.

[039] In action 920, the one pass motion vector processing 900 selects a set of the more reliable motion vectors upon which to base the global motion transform estimation. In one example implementation, the one pass motion vector processing 900 selects motion vectors based on three rules. First, the motion vectors on picture boundaries may not be reliable, and therefore are not selected. For example, after camera motion from frame to frame, picture contents at the boundary blocks may be partially missing. Second, a large residual error after motion compensation of a block (e.g., the SAD value for the block is large) may indicate unreliable motion estimation. The motion vectors for blocks with large motion compensation residuals (as indicated by large SAD value calculated during motion estimation for the block) therefore also are not selected. Third, the motion estimation for blocks with low original variance may not be reliable, so the motion vectors for such blocks also are not selected. In alternative implementations, the processing 900 can base the selection of the set of motion vectors on fewer or additional criteria of reliability.

[040] In action 930, the motion vector processing 900 calculates an estimate of the global motion transform of each frame based on the motion vectors selected in action 920. In one example implementation, the global motion transform is estimated using equation (1) for the similarity global motion model discussed above. Alternatively, the global motion transform can be estimated based on another motion model. For estimating the global motion transform, the motion vector processing 900 uses a Random Sample Consensus (RANSAC) method to detect and eliminate motion vectors that are outliers. More particularly, in the RANSAC method, two motion vectors are chosen at random out of the set of motion vectors selected in action 920. The motion vector processing then solves equation (1) using Least Mean Square Error (LMSE) fit to determine values of s, β, t_x, t_y parameters for the two randomly selected motion vectors. The RANSAC method then determines a number of the other motion vectors that also are consistent with this solution of the global motion transform equation parameters. The RANSAC method repeats the random selection of other pairs of motion vectors until a largest group of

motion vectors consistent with the solution is found. This eliminates motion vectors that are considered outliers, or inconsistent with the consensus LMSE solution for the parameters of the global motion transform.

[041] In action 940-944, the motion vector processing performs error detection and correction by subjecting the estimate of the parameters for global motion transform to two limits for each of zoom, rotation and translation motion. In the flow diagram, the two limits are denoted as a lower limit (T_{1x}) and a higher limit (T_{2x}), where $T_{1x} < T_{2x}$ and x represents zoom, rotation or translation. The two limits may be empirically derived through experimentation on a representative large sample of videos that exhibit jittery video motion, and represent a statistical probability as being global motion representative of jittery video motion. As shown in the actions 940-944, if the global motion transform parameters (s, β, t_x, t_y) are less than their lower limit, the global motion transform as estimated in action 930 is used. Otherwise, if the global motion transform parameters for zoom, rotation and translation exceed the lower limit, but remain less than an upper limit, then the motion vector processing limits the global motion transform to the lower limit as shown at action 943. If the global motion transform estimated at action 930 exceeds the upper limit, then the global motion transform is reset. These actions are thus intended to detect excessive global motion that would not be characteristic of jittery video motion from a shaky hand, and then correct from over-compensation.

[042] A last action 950 of the motion vector processing 900 applies temporal smoothing to the global motion transform. In one example implementation, the global motion transform estimates for a sequence of video frames of a scene are filtered by a 31-tap Gaussian filter, as shown in the following equation (2).

$$W_t^{'} = C_{t-15} \times (W_{t-1}W_{t-2}...W_{t-15})^{-1} + C_{t-14} \times (W_{t-1}W_{t-2}...W_{t-14})^{-1} + ... + C_t \times I + ... + C_{t+14}(W_{t+1}W_{t+2}...W_{t-14}) + C_{t+15}(W_{t+1}W_{t+2}...W_{t+15}) \quad (2)$$

The notation $(W_{t-1}W_{t-2}...W_{t-15})^{-1}$ denotes matrix inversion. In this equation (2), W_t' is the global motion transform after smoothing at time t, and W_{t-i} is the global motion transform before smoothing at time (t-i). The values C_{t-i} are the Gaussian filter coefficients, and I is the identity matrix. In real time video playback or transcoding scenarios, this size of filter imposes a delay of 15 pictures. Alternative implementations can use a longer or shorter filter size, and alternatively can use other type filter including an adaptive filter.

Two Pass Motion Vector Processing In Off Line Mode

[043] The estimation of the global motion transform can alternatively be done using a two pass motion vector processing (such as an example two pass motion vector processing 1000 shown in Figure 10)), which offers better quality of video image stabilization but at a cost of a much increased computation time. Two pass processing also generally imposes less convenience on the user, because the user is forced to wait until the end of the video stabilization processing to view the results. Due to its increased computational load, the two-pass motion vector processing typically is better suited to run as an off-line mode. The two-pass arrangement of the motion vector processing permits better optimization of the error detection and correction part, and the temporal smoothing part of the motion vector processing. In a first pass, the motion vector processing can gather additional information for use in these parts of the processing, which can then be applied in a second motion vector processing pass.

[044] As illustrated at action 1005 in Figure 10), the two-pass motion vector processing 1000 performs scene change detection over the video segment during a first pass. The scene change detection can be performed using conventionally known methods. Then, in action 1010, the two-pass motion vector processing 1000 produces estimates of the global motion transform for each video frame of the scene for the first pass, such as by applying the same actions as in actions 910, 920 and 930 of the single pass motion vector processing 900 of Figure 9). In this first pass, these estimates of the global motion transform are merely gathered to produce statistical information to better optimize global motion transform estimates in a second pass. Accordingly, the global motion estimates in the first pass are not directly used in the image warping 230 of the video image stabilization process 200 to compensate jittery video motion. Instead, at action 1020, the two-pass motion vector processing 1000 calculates minimum and maximum values, and probability distributions for the translation, rotation and zoom parameters of the global motion transform (e.g., s_{min}, s_{max}, β_{min}, β_{max}, t_{xmin}, t_{xmax}, t_{ymin}, t_{ymax}, $p(s)$, $p(\beta)$, $p(t_x)$, and $p(t_y)$) across all frames on each scene of the video segment . Alternatively, the probability statistics could be gathered for the video segment as a whole, and not per individual scene.

[045] Based on these empirical probability distributions of the parameters for each scene, the two-pass motion vector processing 1000 at action 1030 determines the lower and upper limits (T_{1x} and T_{2x}) in each scene for each of the global motion transform parameters (s, β, t_x, t_y). For example, the lower and upper limits can be chosen to

correspond to certain probability thresholds p_1 and p_2. In one example, the probability thresholds p_1 and p2 are 95% and 98%, respectively. In other words, 95% of the values of the parameter observed in the first pass for the scene are within the limit T_{1x} ; and 98% of the observed parameter values are under the upper limit T_{2x}. In alternative implementations, the two-pass motion vector processing can determine the limits based on other probability thresholds or on some other empirical probability-based criteria. In addition, the lower and upper limits could be bounded by hard limits. As compared to the single-pass motion vector processing 900 in Figure 9), this setting of the parameter limits (T_{1x} and T_{2x}) based on empirical probability statistics of the video permits the error correction and detection applied to the global motion transform to be adaptive to the particular content of the video.

[046] At action 1030, the two-pass motion vector processing 1000 then performs a second pass of the motion vector processing. For this second pass, the processing 1000 performs processing as done in actions 910, 920, 930 and 940-944 of the single pass motion vector processing 900. For the limits (T_{1x} and T_{2x}) applied in actions 940-944 of this second pass, the two-pass process uses the limits determined based on the probability statistics of the video that were gathered in the first pass. Because these limits are adapted to the content of the subject video, the two-pass motion vector processing 1000 should perform better quality error detection and correction using these adaptive limits.

[047] The two-pass motion vector processing also can achieve better quality digital video image stabilization (compared to the single pass motion vector processing) by also performing the temporal smoothing globally over the video segment. In one example implementation of the two-pass motion vector processing 1000, the processing performs temporal smoothing using a constrained global optimization, instead of the sliding window approach of the single-pass motion vector processing. One example of such constrained global optimization includes determining a constraint for over-smoothed global motion transforms from averaging the global motion transforms for a number N of video frames as an upper limit for temporal smoothing, where N can be a value in the range [10, 15] for over-smoothing. A convex optimization is then performed under the over-smoothing constraint. In one example implementation, the target function For example, one target function could be specified as,

$$T = \arg\min_{W_i}\left(\sum_{i=0}^{M}\sum_{j=-N}^{+N} C_i \times D(W_i, W_{ij})\right) + \rho\left(\sum_{i=0}^{M} D(W_i, W_{i0})\right) \quad (3)$$

[048] In the target function equation (3), the first term means the final smooth transform W_i should be similar to original transform and its neighbor transforms, W_{ai}, with different weighting coefficients C_i in a window. The second term means final smooth transform W_i should be similar to the over-smoothed W_{ai} as a constraint with weighting ρ. The value D is a distance measurement metric, which could be the distance between the transformed coordinates of the four image corners with W_i and W_{ai}. Convex optimization could apply to minimize the target function.

Image Warping With Global Motion Transform

[049] With reference again to Figure 2), the digital video image stabilization process 200 finally performs image warping 230 based on the global motion transforms produced using either the single or two-pass motion vector processing 220. The image warping may be done in the GPU simply by using the GPU D3D APIs and/or a customized bi-cubic vertex shader to apply an image warping in an opposite direction from the estimate of global motion transform due to jittery video motion. In order to provide enough video content at the boundaries of the video frame to permit image warping, the video frame is cropped in one example implementation by about τ% at the boundaries, which could typically be 10% with the one-pass processing mode and a variable (adaptive) percent for different scenes in the two-pass processing mode. This allows a display port to be moved within the content of the video frame by the image warping, which then forms the image-stabilized video for output.

[050] As can be understood from action 944 in the motion vector processing, when the global motion exceeds the amount of jitter video motion that could be due simply to a shaky hand, the estimated global motion transform is simply reset. In this way, no image warping is applied for global video motion that exceeds the limits characteristic of jitter from hand shakiness. The video frame is instead played with its actual motion, without any compensation for hand shakiness.

Example Computing Environment

[051] Figure 11) illustrates a generalized example of a suitable computing environment 11)00 in which described embodiments, techniques, and technologies may be implemented. For example, the computing environment 11)00 can be one of the

computing devices (e.g., a computer server, personal computer, laptop or other computing device) on which the digital video image stabilization process 200 of Figure 2) is run.

[052] The computing environment 11)00 is not intended to suggest any limitation as to scope of use or functionality of the technology, as the technology may be implemented in diverse general-purpose or special-purpose computing environments. For example, the disclosed technology may be implemented with other computer system configurations, including hand held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The disclosed technology may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

[053] With reference to Figure 11), the computing environment 11)00 includes at least one central processing unit 11)10 and memory 11)20. In Figure 11), this most basic configuration 11)30 is included within a dashed line. The central processing unit 11)10 executes computer-executable instructions and may be a real or a virtual processor. In a multi-processing system, multiple processing units execute computer-executable instructions to increase processing power and as such, multiple processors can be running simultaneously. The memory 11)20 may be volatile memory (e.g., registers, cache, RAM), non-volatile memory (e.g., ROM, EEPROM, flash memory, etc.), or some combination of the two. The memory 11)20 stores software 11)80 that can, for example, implement the technologies described herein.

[054] In addition to the central processing unit 11)10, the computing environment can include other processing resources, such as digital signal processing DSP or multimedia components 11)15. The DSP components 11)15 may be any of the resources that can be utilized advantageously for the digital video image stabilization process by the generic platform library implementation discussed above in connection with Figure 1). For example, the DSP components may include multimedia DSP ASIC units, GPU shader units, multicore CPU, advanced multimedia instruction sets for the CPU, and etc.

[055] A computing environment may have additional features. For example, the computing environment 11)00 includes storage 11)40, one or more input devices 11)50, one or more output devices 11)60, and one or more communication connections 11)70. An interconnection mechanism (not shown) such as a bus, a controller, or a network, interconnects the components of the computing environment 11)00. Typically, operating

system software (not shown) provides an operating environment for other software executing in the computing environment 11)00, and coordinates activities of the components of the computing environment 11)00.

[056] The storage 11)40 may be removable or non-removable, and includes magnetic disks, magnetic tapes or cassettes, CD-ROMs, CD-RWs, DVDs, or any other medium which can be used to store information and which can be accessed within the computing environment 11)00. The storage 11)40 stores instructions for the software 11)80, which can implement technologies described herein.

[057] The input device(s) 11)50 may be a touch input device, such as a keyboard, keypad, mouse, pen, or trackball, a voice input device, a scanning device, or another device, that provides input to the computing environment 11)00. For audio, the input device(s) 11)50 may be a sound card or similar device that accepts audio input in analog or digital form, or a CD-ROM reader that provides audio samples to the computing environment 11)00. The output device(s) 11)60 may be a display, printer, speaker, CD-writer, or another device that provides output from the computing environment 11)00.

[058] The communication connection(s) 11)70 enable communication over a communication medium (e.g., a connecting network) to another computing entity. The communication medium conveys information such as computer-executable instructions, compressed graphics information, or other data in a modulated data signal.

[059] Computer-readable media are any available media from which data and processor instructions that can be accessed within a computing environment 11)00. By way of example, and not limitation, within the illustrated computing environment 11)00, computer-readable media include memory 11)20 and/or storage 11)40. As should be readily understood, the term computer-readable storage media includes the media for storage of data and program instructions such as memory 11)20 and storage 11)40, and not modulated data signals alone.

Example Cloud Computing Network Environment

[060] Figures 12) and 13) illustrate a generalized example of a suitable networking environment 1200 for cloud computing in which the above described digital video image stabilization may be practiced.

[061] In example cloud computing network environment 12)00, various types of computing services for video sharing, storage or distribution (e.g., video sharing or social

networking sites) are provided by a collection of network-accessible computing and storage resources, referred to as the cloud 12)10. For example, the cloud 12)10 can comprise a collection of server computing devices, which may be located centrally or at distributed locations, that provide cloud-based services to various types of users and devices connected via a network such as the Internet.

[062] In example environment 12)00, the cloud 12)10 provides services (such as video storage, video sharing or social networking services, among other examples) for user computing devices. Services can be provided in the cloud 1210 through cloud computing service providers, or through other providers of online services. For example, the cloud-based services can include a video storage service, a video sharing site, a social networking site, or other services via which user-sourced video is distributed for viewing by others on connected devices 1320A-N.

[063] The user may use various mobile video capture devices to record video, such as video camcorders, digital cameras with video mode, mobile phones, and handheld computing devices. The user can upload video to a service on the cloud 1210 either directly (e.g., using a data transmission service of a telecommunications network) or by first transferring the video first to a local computer 1230, such as a laptop, personal computer or other network connected computing device.

[064] As shown in Figure 13), video can be later downloaded, streamed and/or otherwise played back from cloud based video storage or sharing site to other connected computer devices which may have a variety of screen display size factors 1320A-N. Connected device 1320A represents a device with a mid-size display screen, such as may be available on a personal computer, a laptop, a tablet or other like network connected devices.

[065] Connected device 1320B represents a device with display screen with form factors designed to be highly portable (e.g., a small size screen). For example, connected device 1320B could be a mobile phone, smart phone, personal digital assistant, and the like.

[066] Connected device 1320N represents a connected device with a large viewing screen. For example, connected device 1320N could be a television screen (e.g., a smart television) or another device that provides video output to a television or a video projector (e.g., a set-top box or gaming console), or other devices with like video display output.

[067] In the illustrated cloud-computing network environment 1200, the digital video image stabilization can be implemented and performed at various stages of the video sharing, storage and distribution, and by various of the depicted devices depending on the desired use scenario. In one example scenario, the digital video image stabilization is implemented in software on the local computer 1230, and applied when the video is either initially transferred to the local computer or when uploaded to the cloud-based service. In another scenario, the digital video image stabilization is implemented in the cloud, and applied to video as it is uploaded to and stored in the cloud. In another scenario, the digital video image stabilization is implemented by cloud computing services and applied when the video is played, transferred to or distributed to another connected device or service. In yet another scenario, the digital video image stabilization is implemented by cloud computing services, and applied when trans-coding the video for presentation at suitable resolution or streaming at suitable transmission bandwidth for the connected device on which it is to viewed. In still other scenarios, the digital video image stabilization can be performed on the connected device at playback.

Example Alternatives and Combinations

[068] Any of the methods described herein can be performed via one or more computer-readable media (e.g., storage or other tangible media) comprising (e.g., having or storing) computer-executable instructions for performing (e.g., causing a computing device to perform) such methods. Operation can be fully automatic, semi-automatic, or involve manual intervention.

[069] Having described and illustrated the principles of our innovations in the detailed description and accompanying drawings, it will be recognized that the various embodiments can be modified in arrangement and detail without departing from such principles. It should be understood that the programs, processes, or methods described herein are not related or limited to any particular type of computing environment, unless indicated otherwise. Various types of general purpose or specialized computing environments may be used with or perform operations in accordance with the teachings described herein. Elements of embodiments shown in software may be implemented in hardware and vice versa.

[070] In view of the many possible embodiments to which the principles of our invention may be applied, we claim as our invention all such embodiments as may come within the scope of the following claims and equivalents thereto.

CLAIMS:

1. One or more computer-readable media storing computer-executable instructions for causing a computing device programmed thereby to perform a method of digital video stabilization, the one or more computer-readable media including non-volatile memory or a storage device, the method comprising:

evaluating the computing device to determine availability of any of various multimedia digital signal processing ("DSP") components on the computing device;

determining from the results of said evaluating which available multimedia DSP component to utilize for at least one part of digital video stabilization according to a hierarchical structure for ordering multimedia DSP components by criteria comprising at least performance of the digital video stabilization, the digital video stabilization comprising at least local motion vector estimation to produce estimated local motion vectors, processing the estimated local motion vectors for estimation of a global motion transform representing jittery video motion, and image warping based on the global motion transform to compensate the jittery video motion; and

performing digital video stabilization of a video segment on the computing device utilizing the determined available multimedia DSP component for said at least one part of the digital video stabilization, wherein the performing the digital video stabilization includes warping at least one frame of the video segment based on the global motion transform that applies for the at least one frame of the video segment.

2. The one or more computer-readable media of claim 1 wherein the hierarchical structure comprises in order multimedia DSP ASIC units, GPU shader units, multicore CPU and single core CPU.

3. The one or more computer-readable media of claim 1 wherein, when said evaluating determines a GPU shader is available on the computing device, then the digital video stabilization utilizes the GPU shader for the local motion vector estimation.

4. The one or more computer-readable media of claim 3 wherein, when said evaluating determines the GPU shader is available on the computing device, then the digital video stabilization further utilizes the GPU shader for the image warping.

5. The one or more computer-readable media of claim 1 wherein a library of an operating system includes a set of library functions to control the digital video stabilization, and wherein the library provides a programmatic interface for applications to make use of the digital video stabilization, the library including at least one call to an operating system service or a DSP interface as part of the evaluating and including at least one call to a DSP interface to perform the at least one part of the digital video stabilization.

6. The one or more computer-readable media of claim 1 wherein an application executable controls the digital video stabilization, the application executable including at least one call to an operating system service or a DSP interface as part of the evaluating and including at least one call to a DSP interface to perform the at least one part of the digital video stabilization.

7. The one or more computer-readable media of claim 1 wherein a library of an operating system controls the digital video stabilization, the library including at least one call to a DSP interface as part of the evaluating and including at least one call to a DSP interface to perform the at least one part of the digital video stabilization.

8. A method of digital video stabilization, the method comprising performing steps on a computing device of:

performing local motion vector estimation for a frame of a plurality of frames of a scene of a video segment, wherein the local motion vector estimation comprises for said frame:

downsampling said frame and a reference frame associated with said frame by one or more times to form a plurality of versions of said frame at a plurality of resolutions, wherein a lowest resolution version of said frame has a plurality of blocks associated as a

pyramid structure to a successively larger number of blocks at corresponding locations in each higher resolution version of said frame;

for the lowest resolution version of said frame, estimating motion vectors of the blocks of said lowest resolution version frame using an initial motion vector estimation processing of the associated lowest resolution version reference frame;

for each successively higher resolution version of said frame, estimating motion vectors of the blocks of said successively higher resolution version frame using a refinement search starting from the motion vector estimated for the associated block of the preceding lower resolution version frame according to the pyramid structure; and

producing estimated local motion vectors for the blocks of the original resolution version of said frame;

processing the estimated local motion vectors of said frame for estimation of a global motion transform representing jittery video motion, including selecting a set of motion vectors to use for estimation of the global motion transform out of the produced estimated local motion vectors of said frame, wherein said selecting the set of motion vectors comprises one or more of:

excluding motion vectors on picture boundaries,

excluding motion vectors having a high motion compensation residual, and

excluding motion vectors for blocks with low image content variation; and

performing image warping on said frame based on the global motion transform estimated for said frame to compensate for the jittery video motion.

9. The method of claim 8 wherein said processing the estimated local motion vectors of said frame to estimate the global motion transform processes the local motion vectors of said frame using a random sample consensus for removal of outlier motion vectors.

10. The method of claim 8 wherein said global motion transform is based on a similarity model of video motion.

11. The method of claim 8 wherein said processing the estimated local motion vectors of said frame comprises:

comparing the global motion transform estimated from processing the local motion vectors of said frame to lower and upper limits;

if the estimated global motion transform exceeds the upper limit, resetting the global motion transform;

if the estimated global motion transform exceeds the lower limit but not the upper limit, limiting the global motion transform to the lower limit; and

otherwise, using the estimated global motion transform to be representative of jittery video motion.

12. The method of claim 11 further comprising:

as a first pass over the frames in the scene of the video segment, processing the frames to estimate global motion transforms for the frames in the scene of the video segment;

determining probability distributions of parameters of the global motion transforms for the frames in the scene of the video segment;

determining said lower limit and said upper limit based on the probability distributions; and

applying said lower limit and upper limit to estimates of the global motion transform in a second pass processing the frames in the scene of the video segment.

13. The method of claim 8 wherein said processing the estimated local motion vectors of said frame comprises:

processing the motion vectors of said frame using random sample consensus and least mean square error to estimate parameters for a global motion transform based on a similarity motion model;

comparing the estimated parameters of the global motion transform to lower and upper limits on said parameters;

if any of the estimated parameters of the global motion transform exceed their upper limit, resetting the global motion transform;

if any of the estimated parameters of the global motion transform exceed the lower limit but not the upper limit, limiting the estimated parameters of the global motion transform to their lower limit; and

otherwise, using the estimated global motion transform to be representative of jittery video motion.

14. The method of claim 8 wherein said processing the estimated local motion vectors of said frame comprises applying temporal smoothing to the estimated global motion transform using a Gaussian filter.

15. The method of claim 8 wherein said processing the estimated local motion vectors of said frame comprises applying temporal smoothing to the estimated global motion transform using a constrained global optimization over the frames in the scene of the video segment.

16. The method of claim 8 wherein a library of an operating system includes a set of library functions to control the digital video stabilization, and wherein the library provides a programmatic interface for applications to make use of the digital video stabilization, the library including at least one call to a DSP interface to perform one or more of the local motion vector estimation and the image warping.

17. The method of claim 8 wherein an application executable controls the digital video stabilization, the application executable including at least one call to a DSP interface to perform one or more of the local motion vector estimation and the image warping.

18. The method of claim 8 wherein a library of an operating system controls the digital video stabilization, the library including at least one call to a DSP interface to perform one or more of the local motion vector estimation and the image warping.

19. A digital video processing device for processing video to apply digital video stabilization to the video, the digital video processing device comprising:

a memory storing a generic platform video stabilization library program;

at least one digital signal processing component;

a processing unit operating to execute the video stabilization library program from the memory, wherein said execution of the video stabilization library program comprises:

evaluating the at least one digital signal processing component of the digital video processing device;

determining which of the at least one digital signal processing component to utilize for at least one part of digital video stabilization according to a hierarchical structure for ordering digital signal processing components by criteria comprising at least performance of the digital video stabilization; and

performing digital video stabilization on the video in part by processing a frame of the video to estimate local motion vectors of the frame, processing the estimated local motion vectors for estimation of a global motion transform representing jittery video motion, and warping the frame based on the global motion transform to compensate for the jittery video motion, wherein said performing the digital video stabilization executes programming functions in the generic platform video stabilization library program to utilize

the determined digital signal processing component for the at least one part of the digital video stabilization.

20. The digital video processing device of claim 19 wherein said processing the frame to estimate local motion vectors of the frame comprises causing the determined at least one digital signal processing component to perform acts of:

downsampling the frame and a reference frame associated with the frame by one or more times to form a plurality of versions of the frame at a plurality of resolutions, wherein a lowest resolution version of the frame has a plurality of blocks associated as a pyramid structure to a successively larger number of blocks at corresponding locations in each higher resolution version of the frame;

for the lowest resolution version of the frame, estimating motion vectors of the blocks of said lowest resolution version frame using an initial motion vector estimation processing of the associated lowest resolution version reference frame;

for each successively higher resolution version of the frame, estimating motion vectors of the blocks of said successively higher resolution version frame using a refinement search starting from the motion vector estimated for the associated block of the preceding lower resolution version frame according to the pyramid structure; and

producing the estimated local motion vectors for the blocks of the original resolution version of the frame.

21. The digital video processing device of claim 19 wherein said processing the estimated local motion vectors for estimation of the global motion transform representing jittery video motion comprises performing acts of:

selecting a set of motion vectors to use for estimation of the global motion transform out of the estimated local motion vectors of the frame, wherein said selecting the set of motion vectors comprises:

excluding motion vectors on picture boundaries;

excluding motion vectors having a high motion compensation residual; and

excluding motion vectors for blocks with low image content variation.

22. The digital video processing device of claim 19 wherein said processing the estimated local motion vectors for estimation of the global motion transform representing jittery video motion comprises processing the estimated local motion vectors of the frame using random sample consensus and least mean square error to estimate parameters for the global motion transform based on a similarity motion model.

23. The digital video processing device of claim 19 wherein said processing the estimated local motion vectors for estimation of the global motion transform representing jittery video motion comprises:

comparing estimated parameters of the global motion transform to lower and upper limits;

if any of the estimated parameters of the global motion transform exceed their upper limit, resetting the global motion transform;

if any of the estimated parameters of the global motion transform exceed the lower limit but not the upper limit, limiting the estimated parameters of the global motion transform to their lower limit; and

otherwise, using the estimated global motion transform to be representative of jittery video motion.

24. The digital video processing device of claim 23 wherein said upper limit and said lower limit are set in a first pass of the digital video stabilization, and wherein said upper limit and said lower limit are compared to the estimated parameters of the global motion transform in a second pass of the digital video stabilization.

25. The digital video processing device of claim 19 wherein said processing the estimated local motion vectors for estimation of the global motion transform representing

jittery video motion further comprises applying temporal smoothing to the estimated global motion transform using a Gaussian filter.

26. A handheld computing device comprising:

a central processing unit (CPU);

one or more memory units;

a camera;

a graphics processing unit (GPU) including a shader unit; and

the one or more memory units storing computer-executable instructions for causing the handheld computing device, when programmed thereby, to perform real-time digital video stabilization that includes:

for each of multiple frames of a video sequence captured by the camera:

estimating, using the CPU, a motion transform that represents jittery motion associated with shakiness of the handheld computing device; and

automatically warping, using the shader unit of the GPU, at least a portion of the frame based on the motion transform to compensate for the jittery motion; and

initiating uploading of the stabilized video, from the handheld computing device to a video sharing site or social networking site, over a connection to a network.

27. The handheld computing device of claim 26, wherein the estimating the motion transform includes calculating a warping matrix that compensates for one or more of rotation, zooming and translation of the handheld computing device during video capture.

28. The handheld computing device of claim 26, wherein the real-time digital video stabilization further includes:

forming stabilized video around a display port, including, for each of at least some of the multiple frames of the video sequence, cropping boundaries of the frame, wherein the display port depends on the warping for the multiple frames, respectively.

29. The handheld computing device of claim 28, further comprising a display, wherein the storage further stores computer-executable instructions for causing the handheld computing device, when programmed thereby, to perform:

outputting the stabilized video to the display for playback.

30. The handheld computing device of claim 26, wherein the shader unit of the GPU uses a vertex shader.

31. The handheld computing device of claim 26, wherein the handheld computing device is a mobile phone.

32. A computer-readable memory or storage device storing computer-executable instructions for causing a computing device that includes a graphics processing unit (GPU), when programmed thereby, to perform video sharing with real-time digital video stabilization, the video sharing with real-time digital video stabilization including:

for each of multiple frames of a video sequence captured by a camera of the computing device:

estimating a motion transform that represents jittery motion associated with shakiness of the computing device; and

automatically warping, using a vertex shader on the GPU, at least a portion of the frame based on the motion transform to compensate for the jittery motion;

forming stabilized video around a display port, including, for each of at least some of the multiple frames of the video sequence, cropping boundaries of the frame, wherein the display port depends on the warping for the multiple frames, respectively; and

initiating uploading of the stabilized video to a video sharing site or social networking site over a connection to a network.

33. The computer-readable memory or storage device of claim 32, wherein the estimating the motion transform includes calculating a warping matrix that compensates for one or more of rotation, zooming and translation of the computing device during video capture.

34. The computer-readable memory or storage device of claim 32, wherein the real-time digital video stabilization uses single-pass processing consistent with time constraints for playback or streaming of stabilized video during the real-time digital video stabilization.

35. The computer-readable memory or storage device of claim 32, wherein the video sharing with real-time digital video stabilization further includes:

applying temporal smoothing during the estimating the motion transform.

36. The computer-readable memory or storage device of claim 35, wherein the temporal smoothing includes selecting between different filter sizes and/or filter types having different delays.

37. The computer-readable memory or storage device of claim 32, wherein the video sharing with real-time digital video stabilization further includes:

comparing the motion transform to one or more thresholds, wherein performance of the warping depends on the motion transform satisfying the one or more thresholds.

38. The computer-readable memory or storage device of claim 32, wherein a library implements the real-time digital video stabilization and provides a programmatic interface for an application program to use of the real-time digital video stabilization.

39. The computer-readable memory or storage device of claim 38, wherein the real-time digital video stabilization further includes:

evaluating digital signal processing (DSP) components of the computing device; and

choosing, from among the DSP components of the computing device, which of the DSP components to use for different stages of the real-time digital video stabilization.

40. In a computing device that includes a graphics processing unit (GPU), a method of video sharing with real-time digital video stabilization, the method comprising:

for each of multiple frames of a video sequence captured by a camera of the computing device:

estimating a motion transform that represents jittery motion associated with shakiness of the computing device; and

automatically warping, using a vertex shader on the GPU, at least a portion of the frame based on the motion transform to compensate for the jittery motion;

forming stabilized video around a display port, including, for each of at least some of the multiple frames of the video sequence, cropping boundaries of the frame, wherein the display port depends on the warping for the multiple frames, respectively; and

initiating uploading of the stabilized video, from the computing device to a video sharing site or social networking site, over a connection to a network.

41. The method of claim 40, wherein the estimating the motion transform includes calculating a warping matrix that compensates for one or more of rotation, zooming and translation of the computing device during video capture.

42. The method of claim 40, wherein the real-time digital video stabilization uses single-pass processing consistent with time constraints for playback or streaming of the stabilized video during the real-time digital video stabilization.

43. The handheld computing device of claim 26, wherein the real-time digital video stabilization uses single-pass processing consistent with time constraints for playback or streaming of the stabilized video during the real-time digital video stabilization.

44. The handheld computing device of claim 26, wherein the real-time digital video stabilization further includes:

applying temporal smoothing during the estimating the motion transform.

45. The method of claim 40, further comprising, during the estimating the motion transform, applying temporal smoothing.

46. A method of real-time sharing of stabilized digital video for multiple frames of a captured video sequence, comprising:

warping at least a portion of a frame, among the multiple frames of the captured video sequence, based on a motion transform that represents jittery motion of a video capture device to compensate for the jittery motion of the video capture device; and

initiating uploading of stabilized video from the video capture device to a server device associated with a service for video sharing or social networking;

wherein the motion transform has been estimated prior to warping.

47. The method of claim 46, wherein said warping comprises use of a vertex shader of a processing unit.

48. The method of claim 46, further comprising cropping two or more of the multiple frames of the captured video sequence.

49. The method of claim 48, wherein the cropping comprises adaptive cropping depending on characteristics of the two or more frames.

50. The method of claim 46, wherein the warping further comprises moving a display port within the content of one of the multiple frames of a captured video sequence.

51. The method of claim 50, wherein the stabilized video reflects the moved display port.

52. The method of claim 46, wherein said warping and initiating uploading is carried out, at least in part, by a mobile device.

53. The method of claim 46, wherein the estimation of the motion transform occurs in real time during playback.

54. The method of claim 46, wherein the estimation of the motion transform relies on statistical information from a first past estimation of the motion of the video capture device.

55. The method of claim 46, wherein the estimation of the motion transform comprises adaptive filtering.

56. The method of claim 46, wherein the warping comprises warping in real time during playback.

57. A device comprising:

one or more processing units;

one or more memory units;

a camera; and

the one or more memory units storing computer-executable instructions for causing the device, when programmed thereby, to perform real-time digital video stabilization that includes:

for each of multiple frames of a video sequence captured by the camera:

estimating, using at least one of the processing units, a motion transform that represents jittery motion of the device; and

warping at least a portion of the frame based on the motion transform to compensate for the jittery motion; and

initiating transfer of stabilized video to a social media storage location in network communication with the device.

58. The device of claim 57, wherein at least one of said processing units includes a shader unit and wherein said warping comprises using said shader unit.

59. The device of claim 57, wherein the real-time digital video stabilization further includes cropping two or more of the multiple frames of the captured video sequence.

60. The device of claim 59, wherein the cropping comprises adaptive cropping depending on characteristics of the two or more frames.

61. The device of claim 57, wherein the warping comprises moving a display port within the content of one of the multiple frames of a captured video sequence.

62. The device of claim 61, wherein the stabilized video reflects the moved display port.

63. Computer-readable memory storing computer-executable instructions that, when executed, cause a computing device to perform video sharing with real-time digital video stabilization comprising:

capturing multiple frames of a video sequence into a captured video sequence;

warping at least a portion of a frame, among the multiple frames of the captured video sequence, to compensate for the jittery motion of a video capture device; and

initiating uploading of stabilized video from the video capture device to a social networking or video sharing website;

wherein the warping is based on an estimated motion transform that represents jittery motion.

64. The computer-readable memory of claim 63, wherein the video sharing with real-time digital video stabilization further comprises adaptively cropping two or more of the multiple frames of the captured video sequence.

65. The computer-readable memory of claim 63, wherein the video sharing with real-time digital video stabilization further comprises moving a display port within the content of one of the multiple frames of a captured video sequence.

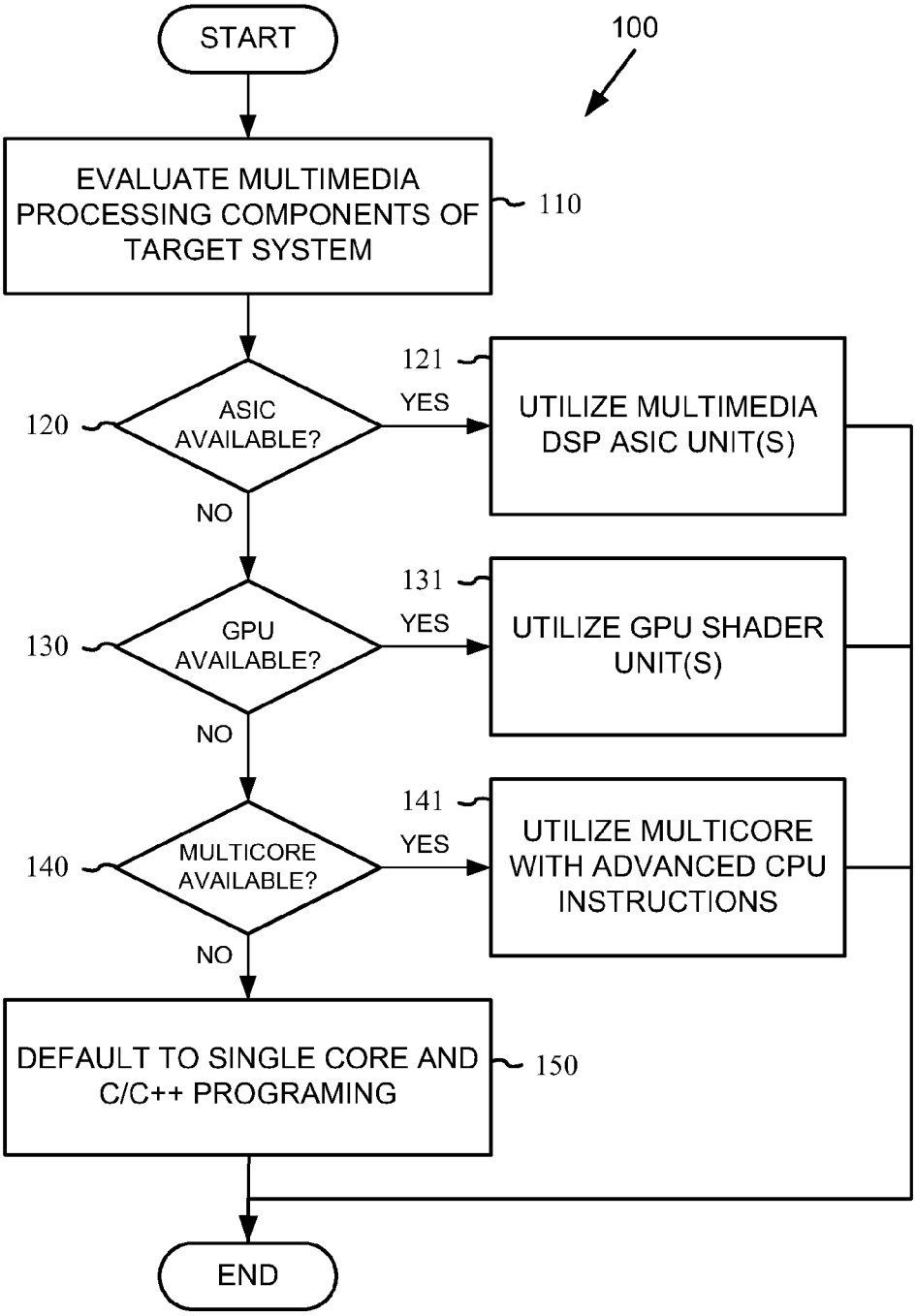

START
100
EVALUATE MULTIMEDIA PROCESSING COMPONENTS OF TARGET SYSTEM
110
ASIC AVAILABLE?
120
121
YES
UTILIZE MULTIMEDIA DSP ASIC UNIT(S)
NO
GPU AVAILABLE?
130
131
YES
UTILIZE GPU SHADER UNIT(S)
NO
MULTICORE AVAILABLE?
140
141
YES
UTILIZE MULTICORE WITH ADVANCED CPU INSTRUCTIONS
NO
DEFAULT TO SINGLE CORE AND C/C++ PROGRAMING
150
END

FIG. 1

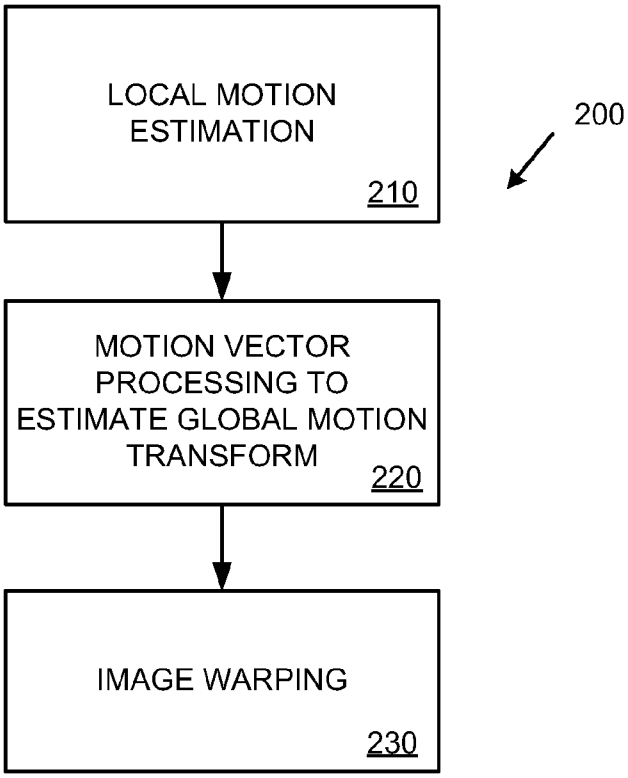
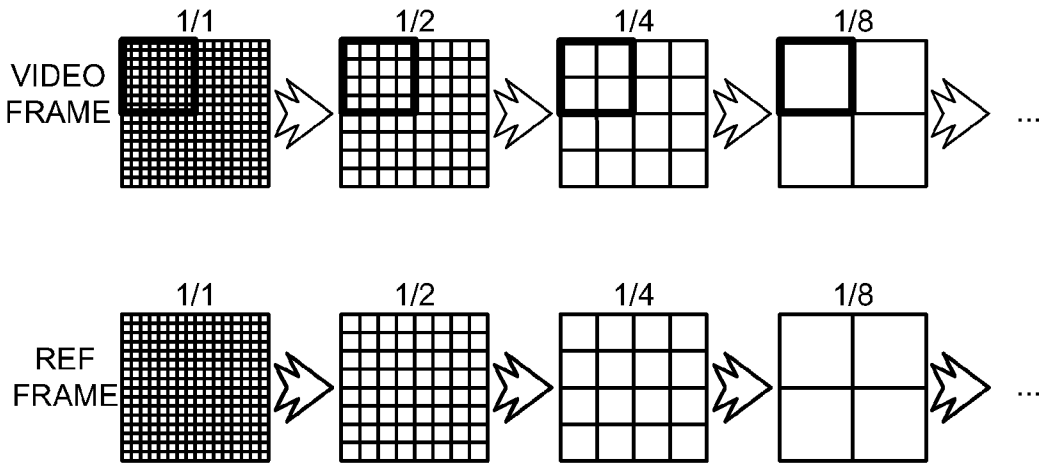


FIG. 2



DOWNSAMPLING FOR PYRAMIDAL BLOCK MOTION ESTIMATION

FIG. 3

FIG. 4

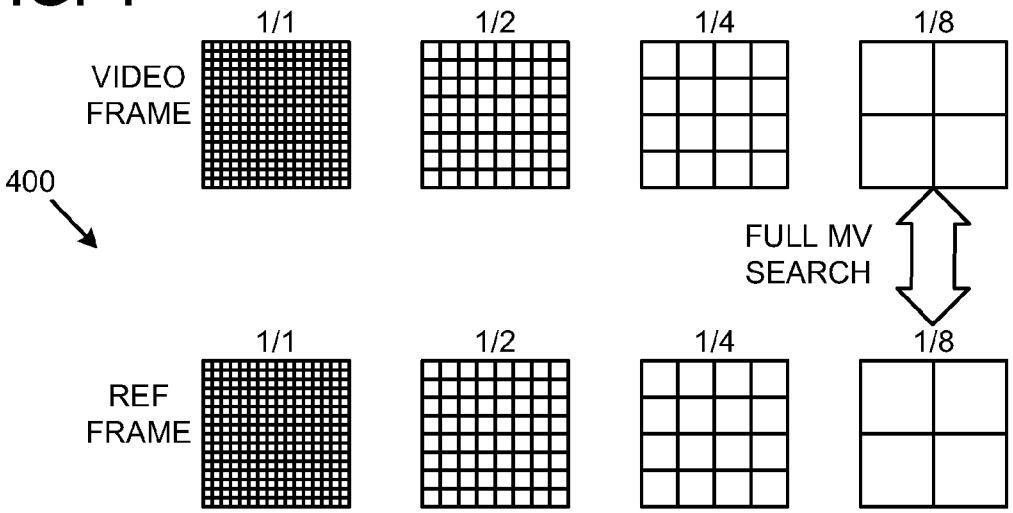


FIG. 5

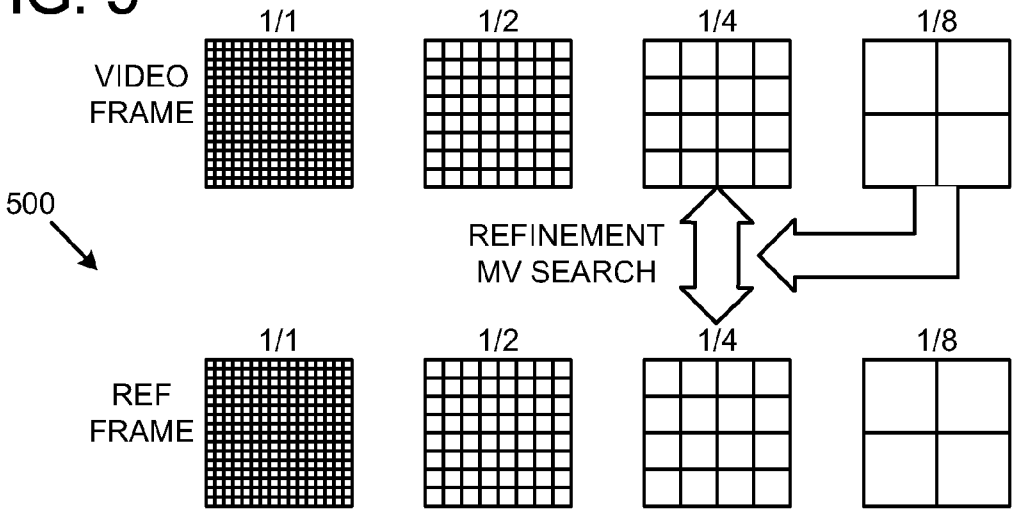
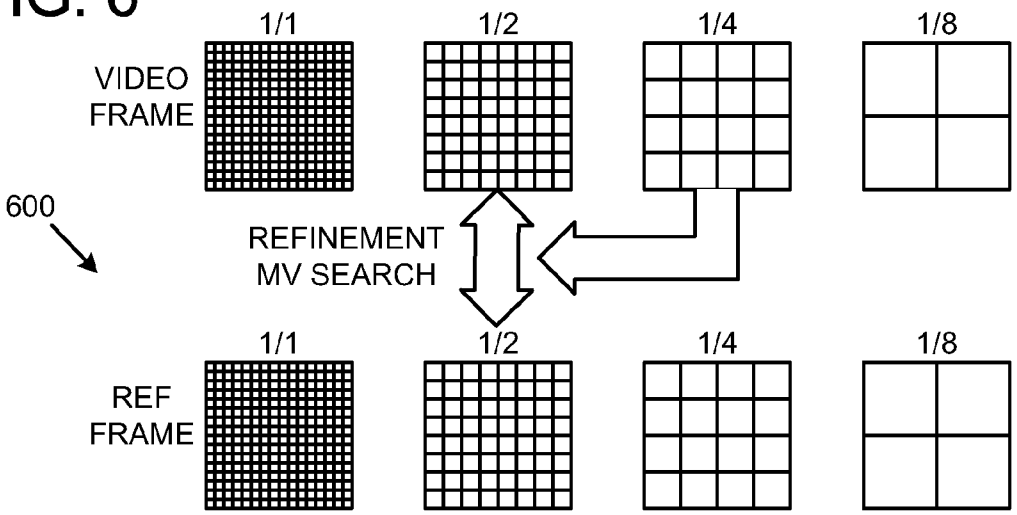


FIG. 6



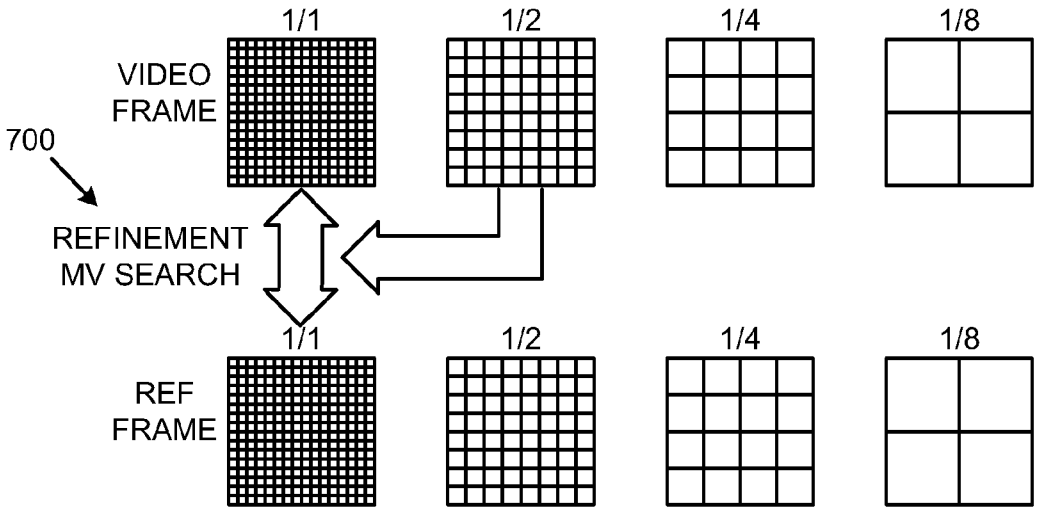

1/1
1/2
1/4
1/8
VIDEO FRAME
700
REFINEMENT MV SEARCH
1/1
1/2
1/4
1/8
REF FRAME

FIG. 7

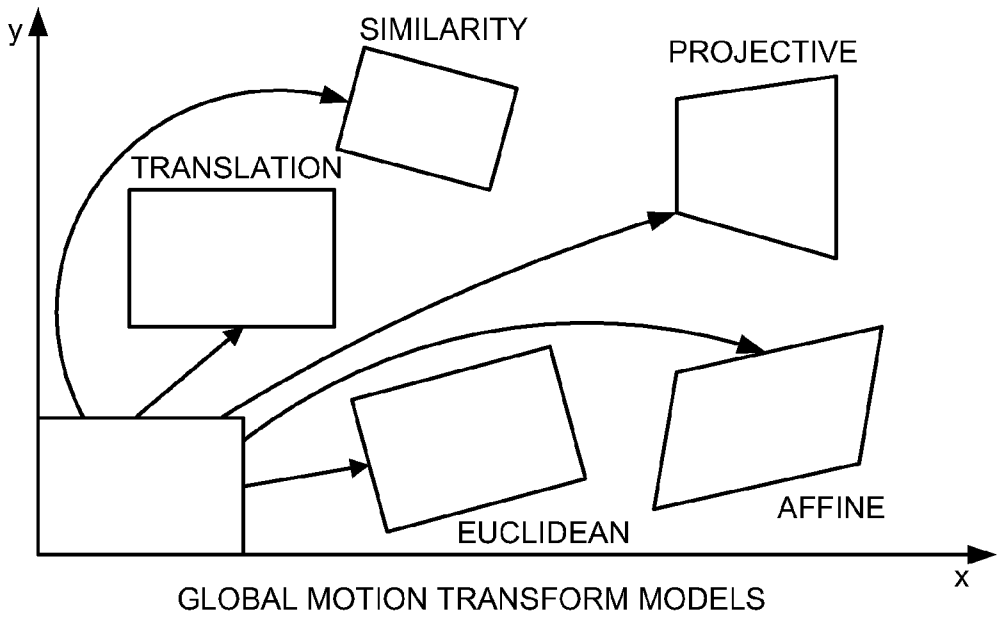

y
SIMILARITY
PROJECTIVE
TRANSLATION
EUCLIDEAN
AFFINE
x
GLOBAL MOTION TRANSFORM MODELS

FIG. 8

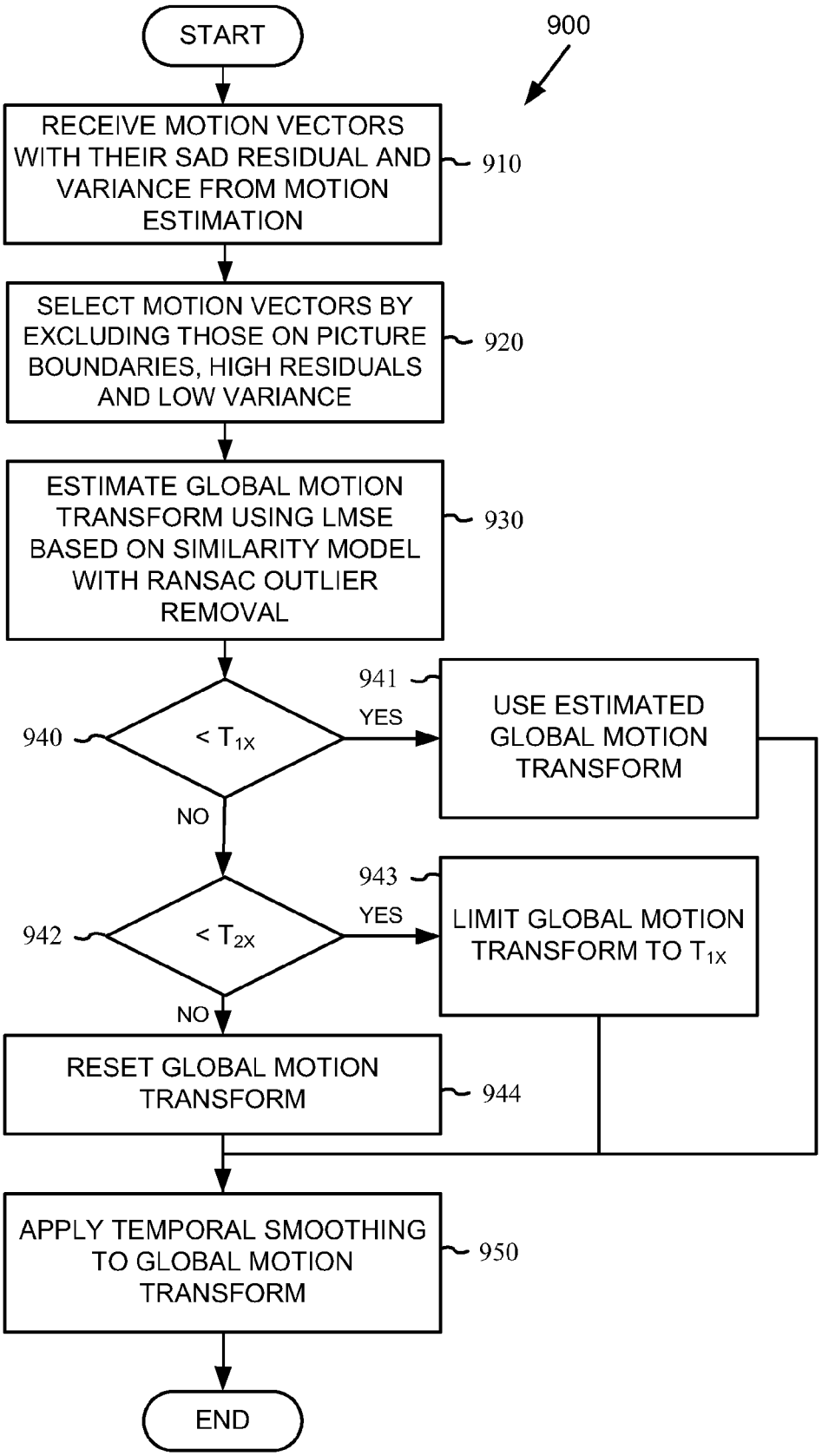

START
900
RECEIVE MOTION VECTORS WITH THEIR SAD RESIDUAL AND VARIANCE FROM MOTION ESTIMATION
910
SELECT MOTION VECTORS BY EXCLUDING THOSE ON PICTURE BOUNDARIES, HIGH RESIDUALS AND LOW VARIANCE
920
ESTIMATE GLOBAL MOTION TRANSFORM USING LMSE BASED ON SIMILARITY MODEL WITH RANSAC OUTLIER REMOVAL
930
940
< T1X
941
YES
USE ESTIMATED GLOBAL MOTION TRANSFORM
NO
942
< T2X
943
YES
LIMIT GLOBAL MOTION TRANSFORM TO T1X
NO
RESET GLOBAL MOTION TRANSFORM
944
APPLY TEMPORAL SMOOTHING TO GLOBAL MOTION TRANSFORM
950
END

FIG. 9

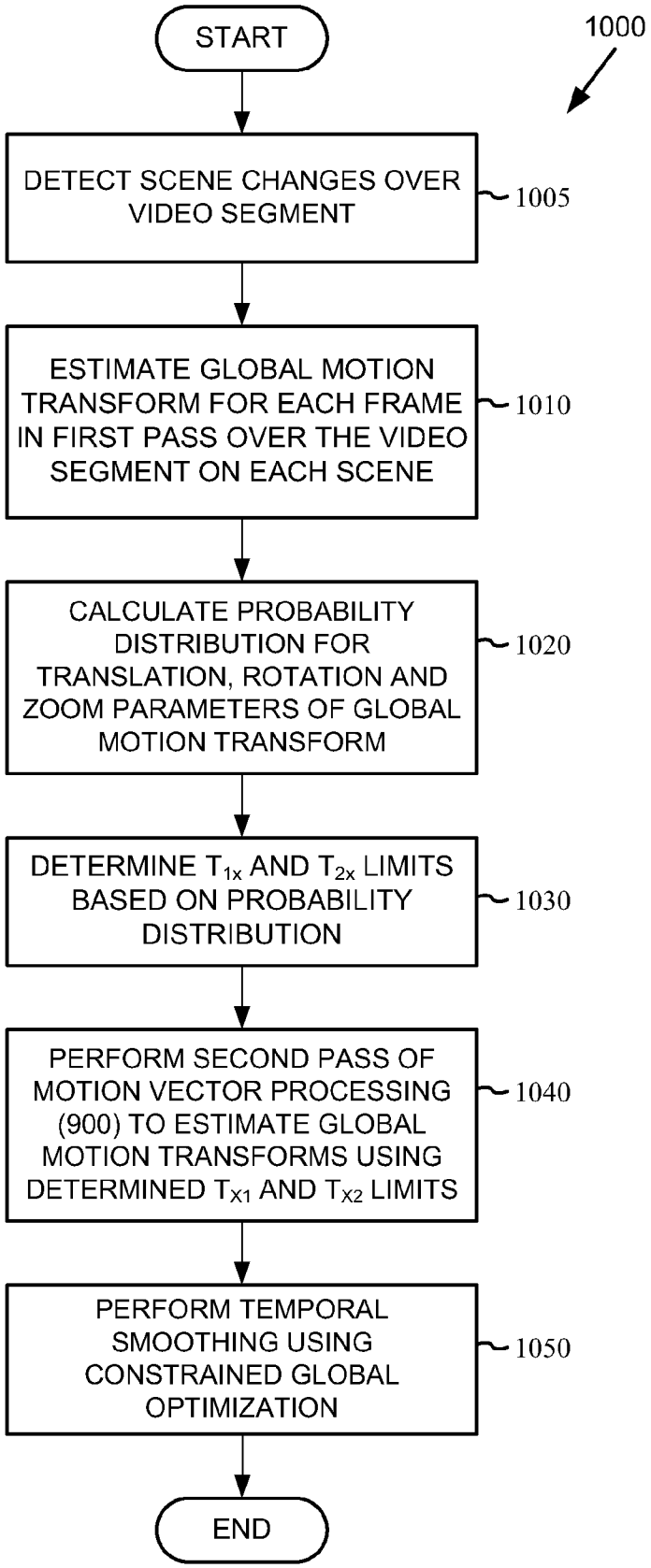

START
1000
DETECT SCENE CHANGES OVER VIDEO SEGMENT
1005
ESTIMATE GLOBAL MOTION TRANSFORM FOR EACH FRAME IN FIRST PASS OVER THE VIDEO SEGMENT ON EACH SCENE
1010
CALCULATE PROBABILITY DISTRIBUTION FOR TRANSLATION, ROTATION AND ZOOM PARAMETERS OF GLOBAL MOTION TRANSFORM
1020
DETERMINE T_{1x} AND T_{2x} LIMITS BASED ON PROBABILITY DISTRIBUTION
1030
PERFORM SECOND PASS OF MOTION VECTOR PROCESSING (900) TO ESTIMATE GLOBAL MOTION TRANSFORMS USING DETERMINED T_{X1} AND T_{X2} LIMITS
1040
PERFORM TEMPORAL SMOOTHING USING CONSTRAINED GLOBAL OPTIMIZATION
1050
END

FIG. 10

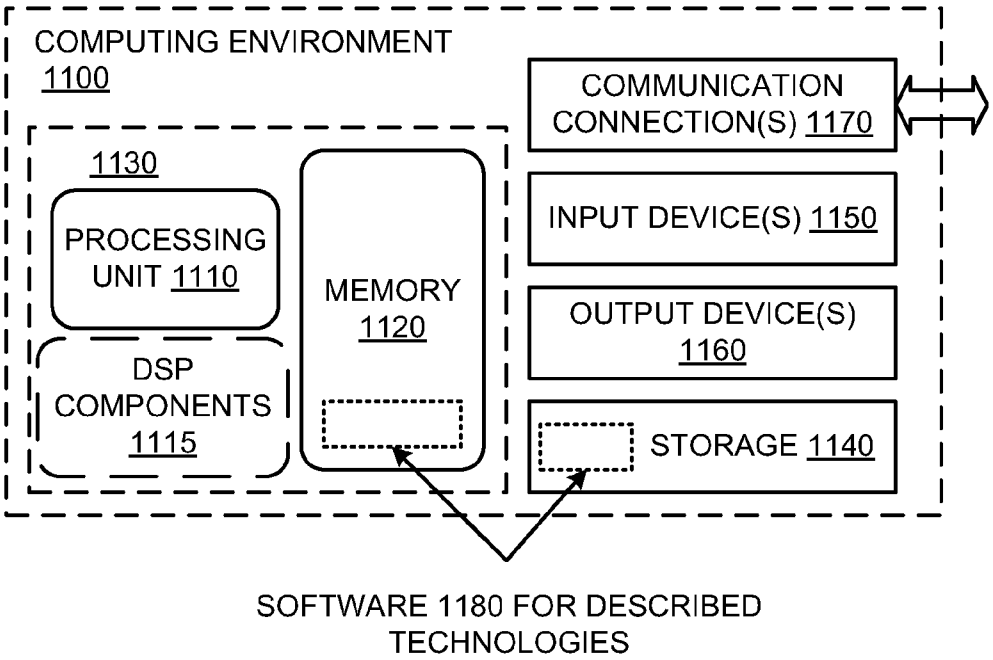

COMPUTING ENVIRONMENT
1100
1130
PROCESSING UNIT 1110
DSP COMPONENTS 1115
MEMORY 1120
COMMUNICATION CONNECTION(S) 1170
INPUT DEVICE(S) 1150
OUTPUT DEVICE(S) 1160
STORAGE 1140
SOFTWARE 1180 FOR DESCRIBED TECHNOLOGIES

FIG. 11

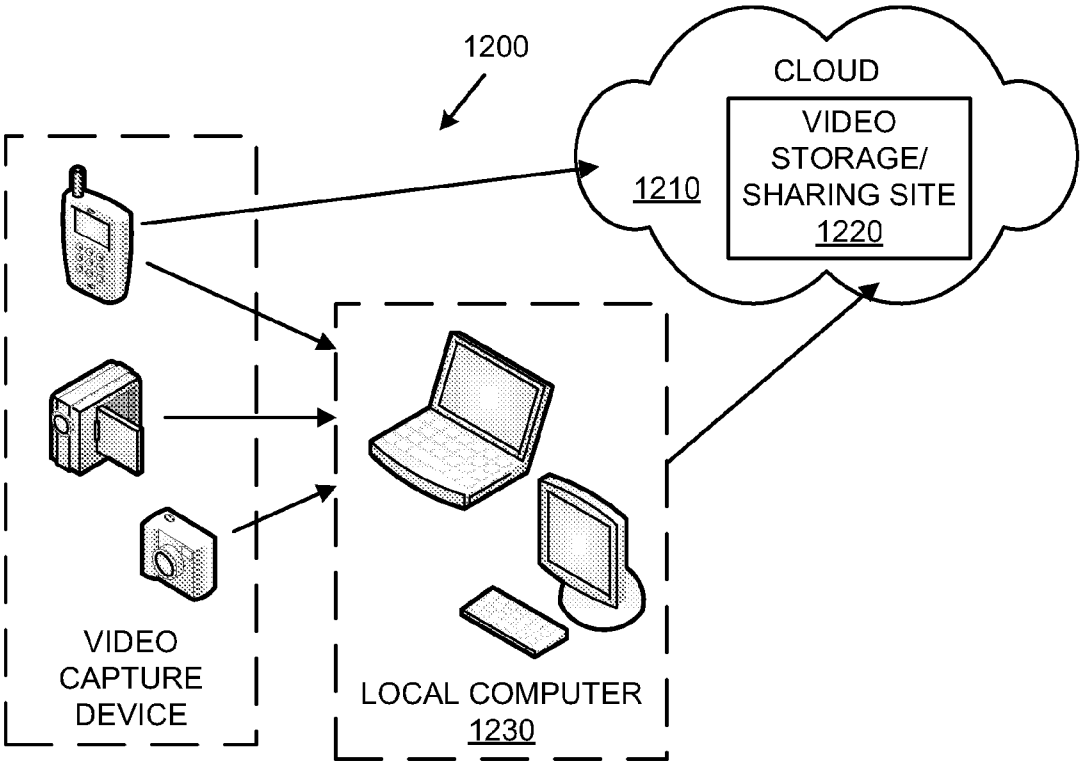


Figure 12

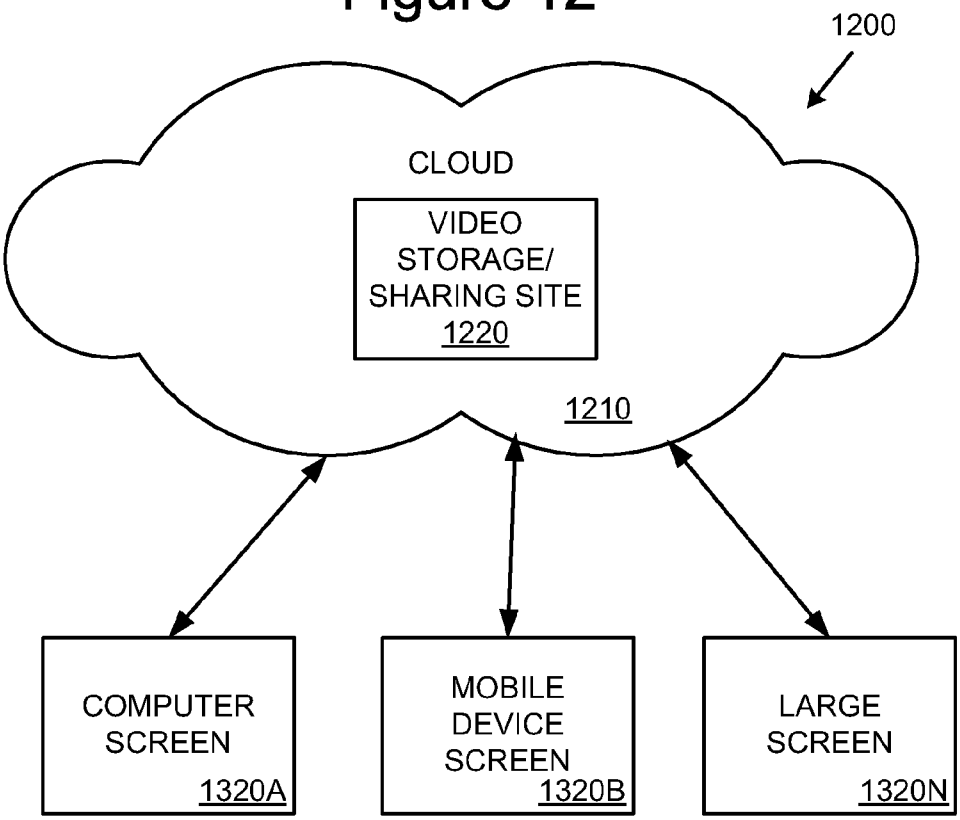
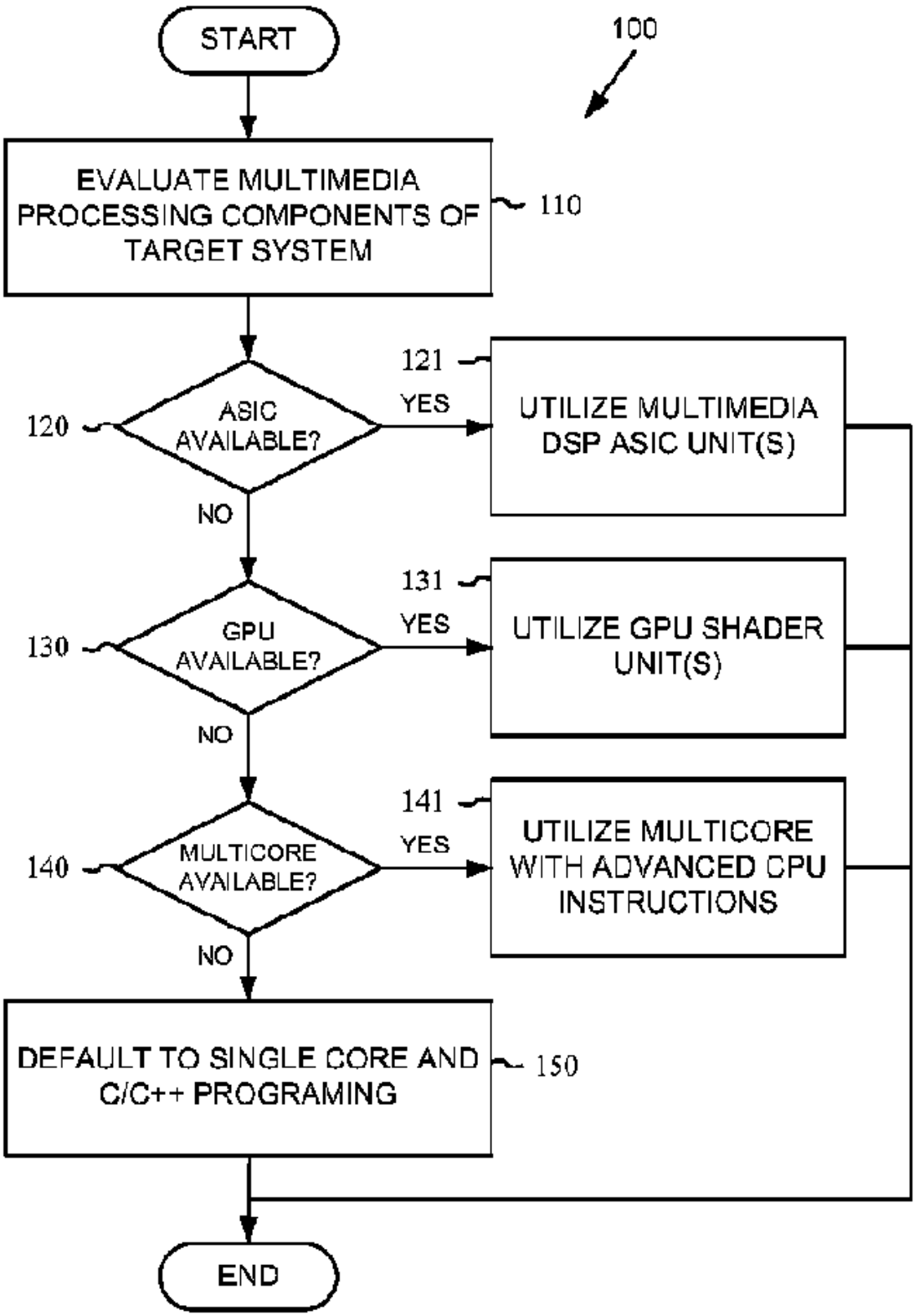


Figure 13


START
100
EVALUATE MULTIMEDIA PROCESSING COMPONENTS OF TARGET SYSTEM
110
120
ASIC AVAILABLE?
YES
121
UTILIZE MULTIMEDIA DSP ASIC UNIT(S)
NO
130
GPU AVAILABLE?
YES
131
UTILIZE GPU SHADER UNIT(S)
NO
140
MULTICORE AVAILABLE?
YES
141
UTILIZE MULTICORE WITH ADVANCED CPU INSTRUCTIONS
NO
DEFAULT TO SINGLE CORE AND C/C++ PROGRAMING
150
END