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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
31 July 2008 (31.07.2008)

PCT

(10) International Publication Number
WO 2008/091485 A2

(51) International Patent Classification: Not classified

(21) International Application Number:
PCT/US2008/000092

(22) International Filing Date: 4 January 2008 (04.01.2008)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/881,979 23 January 2007 (23.01.2007) US

(71) Applicant *(for all designated States except US)*: **EUCLID DISCOVERIES, LLC** [US/US]; 60 Monument Square, Suite 212, Concord, MA 01460 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **PACE, Charles, P.** [US/US]; 60 Smith Farm Road, North Chittenden, VT 23128 (US).

(74) Agents: **WAKIMURA, Mary Lou** et al.; Hamilton, Brook, Smith & Reynolds, P.c., 530 Virginia Road, P.o. Box 9133, Concord, MA 01742-9133 (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, 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, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

(54) Title: SYSTEMS AND METHODS FOR PROVIDING PERSONAL VIDEO SERVICES

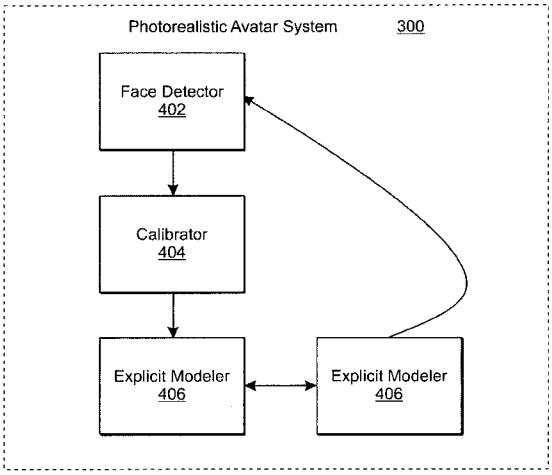

FIG. 4

(57) Abstract: Systems and methods for processing video are provided. Video compression schemes are provided to reduce the number of bits required to store and transmit digital media in video conferencing or videoblogging applications. A photorealistic avatar representation of a video conference participant is created. The avatar representation can be based on portions of a video stream that depict the conference participant. A face detector is used to identify, track and classify the face. Object models including density, structure, deformation, appearance and illumination models are created based on the detected face. An object based video compression algorithm, which uses machine learning face detection techniques, creates the photorealistic avatar representation from parameters derived from the density, structure, deformation, appearance and illumination models.

WO 2008/091485 A2

SYSTEMS AND METHODS FOR PROVIDING PERSONAL VIDEO SERVICES

RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional Application No. 60/881,979, filed January 23, 2007. This application is related to U.S. Provisional Application No. 60/881,966, titled "Computer Method and Apparatus for Processing Image Data," filed January 23, 2007, U.S. Provisional Application No. 60/811,890, titled "Apparatus And Method For Processing Video Data," filed June 8, 2006. This application is related to U.S. Application No. 11/396,010 filed March 31, 2006, which is a continuation-in-part of U.S. Application No. 11/336,366 filed Jan. 20, 2006, which is a continuation-in-part of U.S. Application No. 11/280,625 filed Nov. 16, 2005, which is a continuation-in-part of U.S. Application No. 11/230,686, filed Sep. 20, 2005, which is a continuation-in-part of U.S. Application No. 11/191,562, filed Jul. 28, 2005, now U.S. Patent No. 7,158,680. Each of the foregoing applications is incorporated herein by reference in its entirety.

BACKGROUND

With the recent surge in popularity of digital video, the demand for video compression has increased dramatically. Video compression reduces the number of bits required to store and transmit digital media. Video data contains spatial and temporal redundancy, and these spatial and temporal similarities can be encoded by registering differences within a frame (spatial) and between frames (temporal). The hardware or software that performs compression is called a codec (coder/decoder). The codec is a device or software capable of performing encoding and decoding on a digital signal. As data-intensive digital video applications have become ubiquitous, so has the need for more efficient ways to encode signals. Thus, video compression has now become a central component in storage and communication technology.

Codecs are often used in many different technologies, such as videoconferencing, videoblogging and other streaming media applications, e.g. video podcasts. Typically, a videoconferencing or videoblogging system provides digital compression of audio and video streams in real-time. One of the problems

with videoconferencing and videoblogging is that many participants suffer from appearance consciousness. The burden of presenting an acceptable on-screen appearance, however, is not an issue in audio-only communication.

Another problem videoconferencing and video blogging presents is that the compression of information can result in decreased video quality. The compression ratio is one of the most important factors in video conferencing because the higher the compression ratio, the faster the video conferencing information is transmitted. Unfortunately, with conventional video compression schemes, the higher the compression ratio, the lower the video quality. Often, compressed video streams result in poor images and poor sound quality.

In general, conventional video compression schemes suffer from a number of inefficiencies, which are manifested in the form of slow data communication speeds, large storage requirements, and disturbing perceptual effects. These impediments can impose serious problems to a variety of users who need to manipulate video data easily, efficiently, and without sacrificing quality, which is particularly important in light of the innate sensitivity people have to some forms of visual information.

In video compression, a number of critical factors are typically considered including: video quality and the bit rate, the computational complexity of the encoding and decoding algorithms, robustness to data losses and errors, and latency. As an increasing amount of video data surges across the Internet, not just to computers but also televisions, cell phones and other handheld devices, a technology that could significantly relieve congestion or improve quality represents a significant breakthrough.

SUMMARY

Systems and methods for processing video are provided to create computational and analytical advantages over existing state-of-the-art methods. Video compression schemes are provided to reduce the number of bits required to store and transmit digital media in video conferencing or videoblogging applications. A photorealistic avatar representation of a video conference participant is created. The avatar representation can be based on portions of a video stream that depict the conference participant. An object based video compression algorithm, can use a face detector, such as a Violla-Jones face detector, to detect, track and

classify the face of the conference participant. Object models for structure, deformation, appearance and illumination are created based on the detected face in conjunction with registration of pre-defined object models for general faces. These object models are used to create an implicit representation, and thus, generate the photorealistic avatar representation of the video conference participant.

This depiction can be a lifelike version of the face of the video conference participant. It can be accurate in terms of the user's appearance and expression. Other parts of the originally captured frame can be depicted, possibly with lower accuracy. A short calibration session, executed once per unique user, can take place. This would enable the system to initialize the compression algorithms and create the object models. Preferably, subsequent video conferencing sessions would not need additional calibration.

Should the user require a video representation that is as faithful as a conventional video depiction, the system might require an additional calibration period to adjust the stored models to better match the user's appearance. Otherwise, the user may prefer to use a preferred object model rather than a current object model. The preferred model may be some advantageous representation of the user, for example a calibration session with best lighting and a neater appearance of the user. Another preferred object model would be a calibration model that has been "re-lit" and with "smoothing" applied to the face – both processing steps to achieve a "higher quality" representation of the subject.

A video conferencing/blogging system can be provided using client server framework. A user at a client node can initiate a video conferencing session, communicating through the use of a video camera and headset. The photorealistic avatar representation of each user's face can be generated. The photorealistic avatar representation created can be an implicit representation of the face of the video conference participant.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing will be apparent from the following more particular description of example embodiments of the invention, as illustrated in the accompanying drawings in which like reference characters refer to the same parts

throughout the different views. The drawings are not necessarily to scale, emphasis instead being placed upon illustrating embodiments of the present invention.

FIG. 1 is a block diagram of a video compression (image processing, generally) system employed in embodiments of the present invention.

FIG. 2 is a block diagram illustrating the hybrid spatial normalization compression method employed in embodiments of the present invention.

FIG. 3 is a flow diagram illustrating the process for creating a photorealistic avatar representation of a conference participant in a preferred embodiment.

FIG. 4 is a block diagram illustrating an example of the system components used in connection with generating photorealistic avatar.

FIG. 5A is a schematic diagram illustrating an example of a video conferencing system of the present invention using an instant messaging server.

FIG. 5B is a schematic diagram illustrating an example of a video conferencing/blogging system of the present invention.

FIG. 6 is a schematic illustration of a computer network or similar digital processing environment in which embodiments of the present invention may be implemented.

FIG. 7 is a block diagram of the internal structure of a computer of the network of FIG. 6.

DETAILED DESCRIPTION

A description of example embodiments of the invention follows.

Creating Object Models

In video signal data, frames of video are assembled into a sequence of images. The subject of the video is usually a three-dimensional scene projected onto the camera's two-dimensional imaging surface. In the case of synthetically generated video, a "virtual" camera is used for rendering; and in the case of animation, the animator performs the role of managing this camera frame of reference. Each frame, or image, is composed of picture elements (pels) that represent an imaging sensor response to the sampled signal. Often, the sampled signal corresponds to some reflected, refracted, or emitted energy, (e.g. electromagnetic, acoustic, etc.) sampled through the camera's components on a two

dimensional sensor array. A successive sequential sampling results in a spatiotemporal data stream with two spatial dimensions per frame and a temporal dimension corresponding to the frame's order in the video sequence. This process is commonly referred to as the "imaging" process.

The invention provides a means by which video signal data can be efficiently processed into one or more beneficial representations. The present invention is efficient at processing many commonly occurring data sets in the video signal. The video signal is analyzed, and one or more concise representations of that data are provided to facilitate its processing and encoding. Each new, more concise data representation allows reduction in computational processing, transmission bandwidth, and storage requirements for many applications, including, but not limited to: encoding, compression, transmission, analysis, storage, and display of the video signal. Noise and other unwanted parts of the signal are identified as lower priority so that further processing can be focused on analyzing and representing the higher priority parts of the video signal. As a result, the video signal can be represented more concisely than was previously possible. And the loss in accuracy is concentrated in the parts of the video signal that are perceptually unimportant.

As described in U.S. Application No. 11/336,366 filed Jan. 20, 2006 and U.S. Application No. (Attorney Docket No. 4060.1009-000), titled "Computer Method and Apparatus for Processing Image Data," filed January 23, 2007, the entire teachings of which are incorporated by reference, video signal data is analyzed and salient components are identified. The analysis of the spatiotemporal stream reveals salient components that are often specific objects, such as faces. The identification process qualifies the existence and significance of the salient components, and chooses one or more of the most significant of those qualified salient components. This does not limit the identification and processing of other less salient components after or concurrently with the presently described processing. The aforementioned salient components are then further analyzed, identifying the variant and invariant subcomponents. The identification of invariant subcomponents is the process of modeling some aspect of the component, thereby revealing a parameterization of the model that allows the component to be synthesized to a desired level of accuracy

In one embodiment, the PCA/wavelet encoding techniques are applied to a preprocessed video signal to form a desired compressed video signal. The preprocessing reduces complexity of the video signal in a manner that enables principal component analysis (PCA)/wavelet encoding (compression) to be applied with increased effect. PCA/wavelet encoding is discussed at length in co-pending application, U.S. Application No. 11/336,366 filed Jan. 20, 2006 and U.S. Application No. (Attorney Docket No. 4060.1009-000), titled "Computer Method and Apparatus for Processing Image Data," filed January 23, 2007.

FIG. 1 is a block diagram of an example image processing system 100 embodying principles of the present invention. A source video signal 101 is input to or otherwise received by a preprocessor 102. The preprocessor 102 uses bandwidth consumption or other criteria, such as a face/object detector to determine components of interest (salient objects) in the source video signal 101. In particular, the preprocessor 102 determines portions of the video signal which use disproportionate bandwidth relative to other portions of the video signal 101. One method of segmenter 103 for making this determination is as follows.

Segmenter 103 analyzes an image gradient over time and/or space using temporal and/or spatial differences in derivatives of pels. For the purposes of coherence monitoring, parts of the video signal that correspond to each other across sequential frames of the video signal are tracked and noted. The finite differences of the derivative fields associated with those coherent signal components are integrated to produce the determined portions of the video signal which use disproportionate bandwidth relative to other portions (i.e., determines the components of interest). In a preferred embodiment, if a spatial discontinuity in one frame is found to correspond to a spatial discontinuity in a succeeding frame, then the abruptness or smoothness of the image gradient is analyzed to yield a unique correspondence (temporal coherency). Further, collections of such correspondences are also employed in the same manner to uniquely attribute temporal coherency of discrete components of the video frames. For an abrupt image gradient, an edge is determined to exist. If two such edge defining spatial discontinuities exist then a corner is defined. These identified spatial discontinuities are combined with the gradient flow, which produces motion vectors between corresponding pels across

frames of the video data. When a motion vector is coincident with an identified spatial discontinuity, then the invention segmenter 103 determines that a component of interest (salient object) exists.

Other segmentation techniques are suitable for implementing segmenter 103.

Returning to Fig. 1, once the preprocessor 102 (segmenter 103) has determined the components of interest (salient objects) or otherwise segmented the same from the source video signal 101, a normalizer 105 reduces the complexity of the determined components of interest. Preferably, the normalizer 105 removes variance of global motion and pose, global structure, local deformation, appearance, and illumination from the determined components of interest. The normalization techniques previously described in the related patent applications stated herein are utilized toward this end. This results in the normalizer 105 establishing object models, such as a structural model 107 and an appearance model 108 of the components of interest.

The structural object model 107 may be mathematically represented as:

$$SM(\sigma) = \sum_{x,y}[(v_{x,y} + \Delta_t) + Z] \qquad \text{Equation 1}$$

where σ is the salient object (determined component of interest) and SM ( ) is the structural model of that object;

$v_{x,y}$ are the 2D mesh vertices of a piece-wise linear regularized mesh over the object σ registered over time;

Δ_t are the changes in the vertices over time t representing scaling (or local deformation), rotation and translation of the object between video frames; and

Z is global motion.

From Equation 1, a global rigid structural model, global motion, pose, and locally derived deformation of the model can be derived. Known techniques for estimating structure from motion are employed and are combined with motion estimation to determine candidate structures for the structural parts (component of interest of the video frame over time). This results in defining the position and orientation of the salient object in space and hence provides a structural model 107 and a motion model 111.

The appearance model 108 then represents characteristics and aspects of the salient object which are not collectively modeled by the structural model 107 and the

motion model 111. In one embodiment, the appearance model 108 is a linear decomposition of structural changes over time and is defined by removing global motion and local deformation from the structural model 107. Applicant takes object appearance at each video frame and using the structural model 107 and reprojects to a "normalized pose." The "normalized pose" will also be referred to as one or more "cardinal" poses. The reprojection represents a normalized version of the object and produces any variation in appearance. As the given object rotates or is spatially translated between video frames, the appearance is positioned in a single cardinal pose (i.e., the average normalized representation). The appearance model 108 also accounts for cardinal deformation of a cardinal pose (e.g., eyes opened/closed, mouth opened/closed, etc.) Thus appearance model 108 AM(σ) is represented by cardinal pose P_c and cardinal deformation Δ_c in cardinal pose P_c,

$$AM(\sigma) = \sum_{i} (P_c + \Delta_c P_c) \qquad \text{Equation 2}$$

The pels in the appearance model 108 are preferably biased based on their distance and angle of incidence to camera projection axis. Biasing determines the relative weight of the contribution of an individual pel to the final formulation of a model. Therefore, preferably, this "sampling bias" can factor into all processing of all models. Tracking of the candidate structure (from the structural model 107) over time can form or enable a prediction of the motion of all pels by implication from a pose, motion, and deformation estimates.

Further, with regard to appearance and illumination modeling, one of the persistent challenges in image processing has been tracking objects under varying lighting conditions. In image processing, contrast normalization is a process that models the changes of pixel intensity values as attributable to changes in lighting/illumination rather than it being attributable to other factors. The preferred embodiment estimates a salient object's arbitrary changes in illumination conditions under which the video was captured (i.e., modeling, illumination incident on the object). This is achieved by combining principles from Lambertian Reflectance Linear Subspace (LRLS) theory with optical flow. According to the LRLS theory, when an object is fixed – preferably, only allowing for illumination changes, the set of the reflectance images can be approximated by a linear combination of the first nine spherical harmonics; thus the image lies close to a 9D linear subspace in an

ambient "image" vector space. In addition, the reflectance intensity for an image pixel (x,y) can be approximated as follows.

$$I(x,y) = \sum_{i=0,1,2} \sum_{j=-i,\,-i+1\ldots i-1,i} l_{ij} b_{ij}(n),$$

Using LRLS and optical flow, expectations are computed to determine how lighting interacts with the object. These expectations serve to constrain the possible object motion that can explain changes in the optical flow field. When using LRLS to describe the appearance of the object using illumination modeling, it is still necessary to allow an appearance model to handle any appearance changes that may fall outside of the illumination model's predictions.

Other mathematical representations of the appearance model 108 and structural model 107 are suitable as long as the complexity of the components of interest is substantially reduced from the corresponding original video signal but saliency of the components of interest is maintained.

Returning to FIG. 1, PCA/wavelet encoding is then applied to the structural object model 107 and appearance object model 108 by the analyzer 110. More generally, analyzer 110 employs a geometric data analysis to compress (encode) the video data corresponding to the components of interest. The resulting compressed (encoded) video data is usable in the FIG. 2 image processing system. In particular, these object models 107, 108 can be stored at the encoding and decoding sides 232, 236 of FIG. 2. From the structural model 107 and appearance model 108, a finite state machine can be generated. The conventional coding 232 and decoding 236 can also be implemented as a conventional Wavelet video coding decoding scheme.

PCA encoding is applied to the normalized pel data on both sides 232 and 236, which builds the same set of basis vectors on each side 232, 236. In a preferred embodiment, PCA/wavelet is applied on the basis function during image processing to produce the desired compressed video data. Wavelet techniques (DWT) transform the entire image and sub-image and linearly decompose the appearance model 108 and structural model 107 then this decomposed model is truncated gracefully to meet desired threshold goals (ala EZT or SPIHT). This enables scalable video data processing unlike systems/methods of the prior art due to the "normalize" nature of video data.

As shown in FIG. 2, the previously detected object instances in the uncompressed video streams for one or more objects 230, 250, are each processed with a separate instance of a conventional video compression method 232. Additionally, the non-object 202 resulting from the segmentation of the objects 230, 250, is also compressed using conventional video compression 232. The result of each of these separate compression encodings 232 are separate conventional encoded streams for each 234 corresponding to each video stream separately. At some point, possibly after transmission, these intermediate encoded streams 234 can be decompressed (reconstructed) at the decoder 236 into a synthesis of the normalized non-object 210 and a multitude of objects 238, 258. These synthesized pels can be de-normalized 240 into their de-normalized versions 222, 242, 262 to correctly position the pels spatially relative to each other so that a compositing process 270 can combine the object and non-object pels into a synthesis of the full frame 272.

Creating a Photorealistic Avatar Representation

FIG. 3 is a flow diagram illustrating the steps taken by the video conferencing photorealistic avatar generation system 300. This system 300 creates a photorealistic avatar representation of a video conference or video blog participant. As shown in FIG. 3, at 302, a face of one of the video conference participants is detected from one or more video frames of the video conference data stream. The face is detected using the Viola-Jones face detector (or any other face detector).

At 304, the system 100 determines whether the face has been calibrated before. If there is no existing calibration, then at 306 the face is calibrated. Calibration information can include information about face orientation (x, y positions specifying where the face is centered), scale information, and structure, deformation, appearance and illumination information. These parameters can be derived using a hybrid three-dimensional morphable model and LRLS algorithm and the structure, deformation, appearance and illumination models. These models are discussed in U.S. Application No. 11/336,366 filed Jan. 20, 2006 and U.S. Application No. (Attorney Docket No. 4060.1009-000), titled "Computer Method and Apparatus for Processing Image Data," filed January 23, 2007, the entire

teachings of which are incorporated by reference. Other known modeling technologies may also be used to determine these parameters, such as three-dimensional morphable modeling, active appearance models, etc. These approximations can be used to estimate the pose and structure of the face, and the illumination conditions for each frame in the video. Once the structure, deformation, appearance and illumination basis (e.g. calibration information) for the individual's face has been resolved, then at 308, these explicit models can be used to detect, track and model the individual's face.

At 310, these parameters (e.g. structure, deformation, appearance and illumination basis) can be used to initialize the implicit modeling. The implicit modeling builds its model relative to the information obtained from the explicit modeling and provides a compact encoding of the individual's face. The parameters obtained from the explicit modeling are used as a ground truth for estimating the implicit model. For example, the explicit modeling parameters are used to build expectations about how lighting interacts with the structure of the face and then the face is sampled, these constraints provide a means of limiting the search space for the implicit algorithm. At 312, the individual's face is detected, tracked and classified using the implicit model, and a photorealistic avatar representation is generated. The frames generated using the implicit modeling use less encoding per frame and require fewer parameters than the explicit model. The photorealistic avatar representation is a synthetic representation of the face (e.g. a proxy avatar) of the conference participant. The synthetic representation fidelity can range from a faithful representation of the participant in the original video capture all the way to a representation supported by a previous calibration session.

The system 300 performs periodic checking to ensure that it is basing its modeling on realistic approximations. Thus, at step 314, the system 300 checks to confirm that its implicit object modeling is working properly. The system may determine that the implicit object modeling is working if the reprojection error is low for a certain amount of time. If the reprojection error is low and there is significant amount of motion, then it is likely that the implicit object modeling is working properly. If, however, the reprojection error is high, then the system 300 may determine that the implicit modeling is not working optimally. Similarly, if the

system 300 detects a disproportional amount of bandwidth, the system may determine that the implicit modeling is not working optimally.

If it is determined that the implicit modeling is not working, then at step 316, the system 300 checks to determine whether a face can be detected. If a face can be detected, then at step 304, the system 300 finds the existing calibration information for the face and proceeds accordingly. If a face cannot be detected, then the system proceeds to step 302 to detect the face using the Viola-Jones face detector.

In another preferred embodiment, the present invention uses the explicit modeling to re-establish the implicit modeling. The explicit modeling re-establishes the model parameters necessary to re-initialize the implicit model. The full re-establishment involving running the face detector is performed if the explicit modeling cannot re-establish modeling of the participant.

It should be noted that face detection leads can use implicit modeling for calibration. In this case, the implicit model is used to "calibrate" the explicit model. Then, the explicit model starts it's processing, which then leads to an initialization of the implicit model as well.

This periodic checking enables the system 300 to reconfirm that it is in fact modeling a real object, a human face, and causes the system 300 to reset its settings periodically. This arrangement provides a tight coupling between the face detector 402, the calibrator 404, the explicit modeler 406 and the implicit modeler 408. In this way, periodically, the feedback from the explicit modeler 406 is used to reinitialize the implicit modeler 408. A block diagram illustrating an example implementation of this system 300 is shown in FIG. 4.

Photorealistic Avatar Preferences

The photorealistic avatar generation system 300 can provide a host of preferences to conference participants to make their video conference experience more enjoyable. For example, a conference participant can select a preference to require that their photorealistic avatar representation always look directly into camera, such that it appears that the avatar representation is looking directly at the other conference participant. Since the modeling employed allows for the re-posing of any model relative to a virtual camera, the gaze adjustment required for non-co-located cameras and monitors can be compensated for. The conference participant can also select a specific background model. By selecting a consistent background model, the system 300 is able to provide an even more efficient compressed version the video stream. The model may be a predefined background or a low-resolution of the actual background, for example. During face detection and calibration, the conference participant can also customize features associated with their personal attributes in their photorealistic avatar representation, such as removal of wrinkles, selection of hair style/effects, selection of clothing, etc.

By providing a photorealistic avatar representation of the conference participant, the system 300 provides an added layer of security that is not typically available in conventional video conference systems. In particular, because the photorealistic avatar representation is a synthetic representation, the conference participant does not need to worry about the other conference participant knowing potentially confidential information, such as confidential documents that the conference participant is looking at during the video conference, or other confidential information that might be derived by being able to view the specific environment in which video conference is being recorded.

Video Conferencing System

FIG. 5A is a diagram illustrating an example of a asynchronous or near-synchronous video conferencing system 500 using an asynchronous or near-synchronous video conferencing server, referred to hereafter as an instant messaging server 502. In this example, a three node network is shown with the instant messaging server 502 and two client machines 504, 506. A user sitting at each

machine 504, 506 would be able to initiate a video conferencing session, communicating through the use of a video camera and headset. A photorealistic avatar representation of each user's face would appear in front of each user. This depiction is intended to be accurate in terms of the user's appearance and expression. Other parts of the originally captured frame will be depicted, preferably at a lower accuracy. A short calibration session, executed once per unique user, would take place. This would enable the system to initialize the compression algorithms and create the object models. Subsequent video conferencing sessions would most likely not require additional calibration. Each user can "play" the sequence of asynchronous communication in the order of interchange. In this way, each user can cue the session recording based on user input, detected speech, or some other cue. Additionally, this interaction allows for many simultaneous "conversations" to occur without the "interruptions" that might occur with a fully synchronous scenario.

The asynchronous or semi-synchronous messaging system environment 500 provides a means by which multiple participants are able to interact with each other. This is an important element of usability. The instant messaging session aspect allows the users to "edit" their own video, and review it prior to "sending" it to the other side. There is an aspect of control and also bandwidth reduction that is critical. The editing and control aspects may also be used to generate "higher" quality video segments that can then later be used for other purposes (e.g. by associating the phonemes, or audio phrase patterns, in the video, a video session can be provided without a camera, by using "previous" segments stitched together.)

FIG. 5B is a diagram illustrating an example of a video conferencing/blogging system 540. In this example, client systems 551 connect to the application server 556, which hosts the photorealistic avatar generation system 300 referenced in FIGs. 3 and 4. The application server 556 can store previously generated object (density, structure, appearance, illumination, etc.) models 552 in the object model archive 554. These object models 552 are created to generate the photorealistic avatar representation for users of the system 540 as discussed above in Figures 3 and 4. The photorealistic avatar representation can be streamed for video blogging (vlogs) 558 to the client systems 551.

Processing Environment

FIG. 6 illustrates a computer network or similar digital processing environment 600 in which the present invention may be implemented. Client computer(s)/devices 50 and server computer(s) 60 provide processing, storage, and input/output devices executing application programs and the like. Client computer(s)/devices 50 can also be linked through communications network 70 to other computing devices, including other client devices/processes 50 and server computer(s) 60. Communications network 70 can be part of a remote access network, a global network (e.g., the Internet), a worldwide collection of computers, Local area or Wide area networks, and gateways that currently use respective protocols (TCP/IP, Bluetooth, etc.) to communicate with one another. Other electronic device/computer network architectures are suitable.

FIG. 7 is a diagram of the internal structure of a computer (e.g., client processor/device 50 or server computers 60) in the computer system of FIG. 6. Each computer 50, 60 contains system bus 79, where a bus is a set of hardware lines used for data transfer among the components of a computer or processing system. Bus 79 is essentially a shared conduit that connects different elements of a computer system (e.g., processor, disk storage, memory, input/output ports, network ports, etc.) that enables the transfer of information between the elements. Attached to system bus 79 is an Input/Output (I/O) device interface 82 for connecting various input and output devices (e.g., keyboard, mouse, displays, printers, speakers, etc.) to the computer 50, 60. Network interface 86 allows the computer to connect to various other devices attached to a network (e.g., network 70 of FIG. 6). Memory 90 provides volatile storage for computer software instructions 92 and data 94 used to implement an embodiment of the present invention (e.g., personal video service). Disk storage 95 provides non-volatile storage for computer software instructions 92 and data 94 used to implement an embodiment of the present invention. Central processor unit 84 is also attached to system bus 79 and provides for the execution of computer instructions.

In one embodiment, the processor routines 92 and data 94 are a computer program product, including a computer readable medium (e.g., a removable storage

medium, such as one or more DVD-ROM's, CD-ROM's, diskettes, tapes, etc.) that provides at least a portion of the software instructions for the invention system. Computer program product can be installed by any suitable software installation procedure, as is well known in the art. In another embodiment, at least a portion of the software instructions may also be downloaded over a cable, communication and/or wireless connection. In other embodiments, the invention programs are a computer program propagated signal product embodied on a propagated signal on a propagation medium (e.g., a radio wave, an infrared wave, a laser wave, a sound wave, or an electrical wave propagated over a global network, such as the Internet, or other network(s)). Such carrier medium or signals provide at least a portion of the software instructions for the present invention routines/program 92.

In alternate embodiments, the propagated signal is an analog carrier wave or digital signal carried on the propagated medium. For example, the propagated signal may be a digitized signal propagated over a global network (e.g., the Internet), a telecommunications network, or other network. In one embodiment, the propagated signal is a signal that is transmitted over the propagation medium over a period of time, such as the instructions for a software application sent in packets over a network over a period of milliseconds, seconds, minutes, or longer. In another embodiment, the computer readable medium of computer program product is a propagation medium that the computer system may receive and read, such as by receiving the propagation medium and identifying a propagated signal embodied in the propagation medium, as described above for computer program propagated signal product.

Generally speaking, the term "carrier medium" or transient carrier encompasses the foregoing transient signals, propagated signals, propagated medium, storage medium and the like.

While this invention has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the scope of the invention encompassed by the appended claims.

For example, the present invention may be implemented in a variety of computer architectures. The computer networks illustrated in FIGs. 5A, 5B, 6 and 7 are for purposes of illustration and not limitation of the present invention.

The invention can take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment containing both hardware and software elements. In a preferred embodiment, the invention is implemented in software, which includes but is not limited to firmware, resident software, microcode, etc.

Furthermore, the invention can take the form of a computer program product accessible from a computer-usable or computer-readable medium providing program code for use by or in connection with a computer or any instruction execution system. For the purposes of this description, a computer-usable or computer readable medium can be any apparatus that can contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device.

The medium can be an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system (or apparatus or device) or a propagation medium. Examples of a computer-readable medium include a semiconductor or solid state memory, magnetic tape, a removable computer diskette, a random access memory (RAM), a read-only memory (ROM), a rigid magnetic disk and an optical disk. Some examples of optical disks include compact disk – read only memory (CD-ROM), compact disk – read/write (CD-R/W) and DVD.

A data processing system suitable for storing and/or executing program code will include at least one processor coupled directly or indirectly to memory elements through a system bus. The memory elements can include local memory employed during actual execution of the program code, bulk storage, and cache memories, which provide temporary storage of at least some program code in order to reduce the number of times code are retrieved from bulk storage during execution.

Input/output or I/O devices (including but not limited to keyboards, displays, pointing devices, etc.) can be coupled to the system either directly or through intervening I/O controllers.

Network adapters may also be coupled to the system to enable the data processing system to become coupled to other data processing systems or remote printers or storage devices through intervening private or public networks. Modems, cable modem and Ethernet cards are just a few of the currently available types of network adapters.

CLAIMS

What is claimed is:

1. A method of video conferencing, the method comprising the steps of:

detecting a human face of a video conference participant depicted in portions of a video stream;

creating one or more object models to model the face of the video conference participant; and

using the object models, creating a photorealistic avatar representation of the video conference participant.

2. A method for providing video conferencing as in Claim 1 wherein the face of the video conference participant is detected and tracked using a Viola/Jones face detection algorithm.

3. A method for providing video conferencing as in Claim 1 wherein the photorealistic avatar representation object models are created as an implicit representation of the face of the video conference participant.

4. A method for providing video conferencing as in Claim 3 wherein the implicit representation of the video conference participant is a simulated representation of the face of the video conference participant.

5. A method for providing video conferencing as in Claim 3 wherein the detecting and tracking comprise using a Viola/Jones face detection algorithm further includes the steps of:

identifying corresponding elements of at least one object associated with the face in two or more video frames from the video stream; and

tracking and classifying the corresponding elements to identify relationships between the corresponding elements based on previously calibrated and modeled faces.

6. A method for providing video conferencing as in Claim 1 wherein the object models include object models for structure, deformation, pose, motion, illumination, and appearance.

7. A video conferencing system comprising:

a face detector detecting a face of a video conference participant in a video stream;'

a calibrator generating a calibration model calibrating the face of the video conference participant;

object models, in combination with the calibrator and face detector, the object models modeling portions of the video stream depicting the face of the video conference participant based on the calibration model; and

a photorealistic avatar representation of the video conference participant, the photorealistic avatar representation generated from the face detector, calibrator and object models.

8. A system for video conferencing comprising:

means for providing object models to model portions of a video stream depicting at least one participant of a video conference; and

means for using the object models to create a photorealistic avatar representation of the video conference participant.

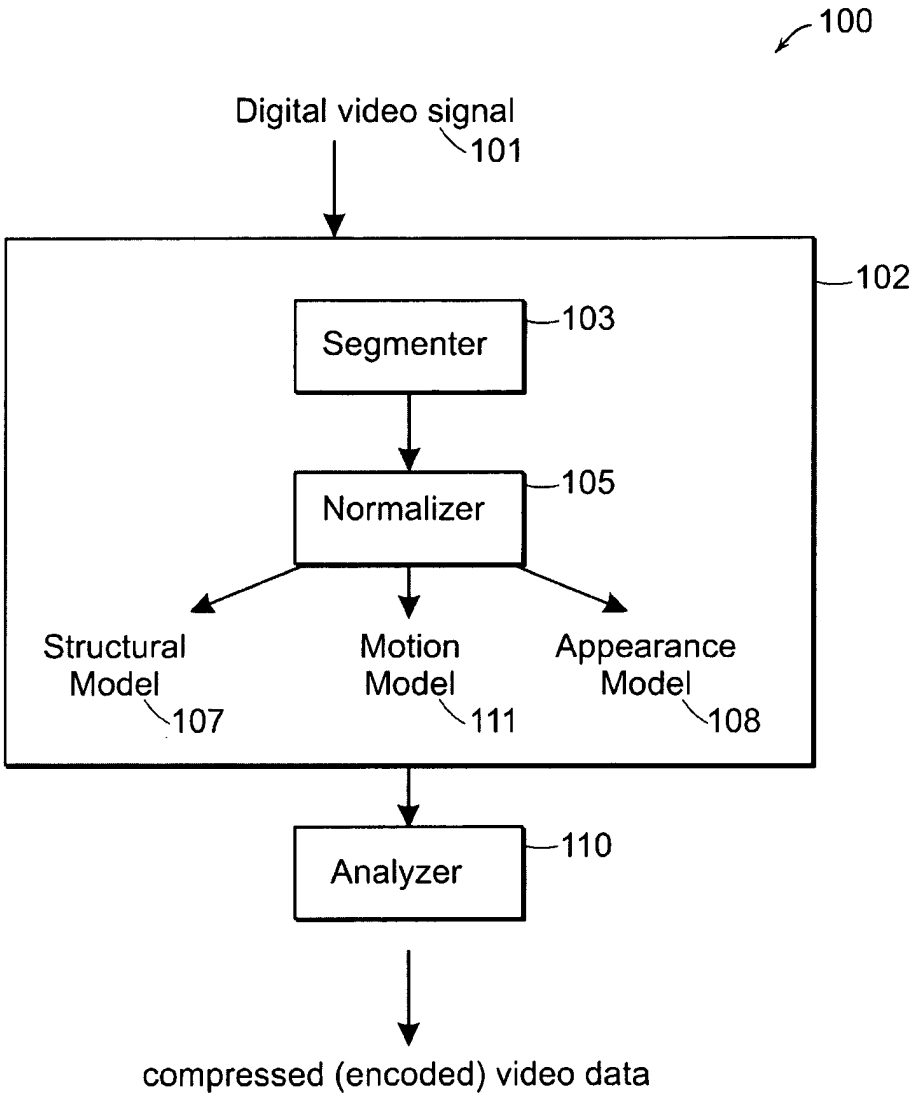
100
Digital video signal
101
102
Segmenter
103
Normalizer
105
Structural Model
107
Motion Model
111
Appearance Model
108
Analyzer
110
compressed (encoded) video data

FIG. 1

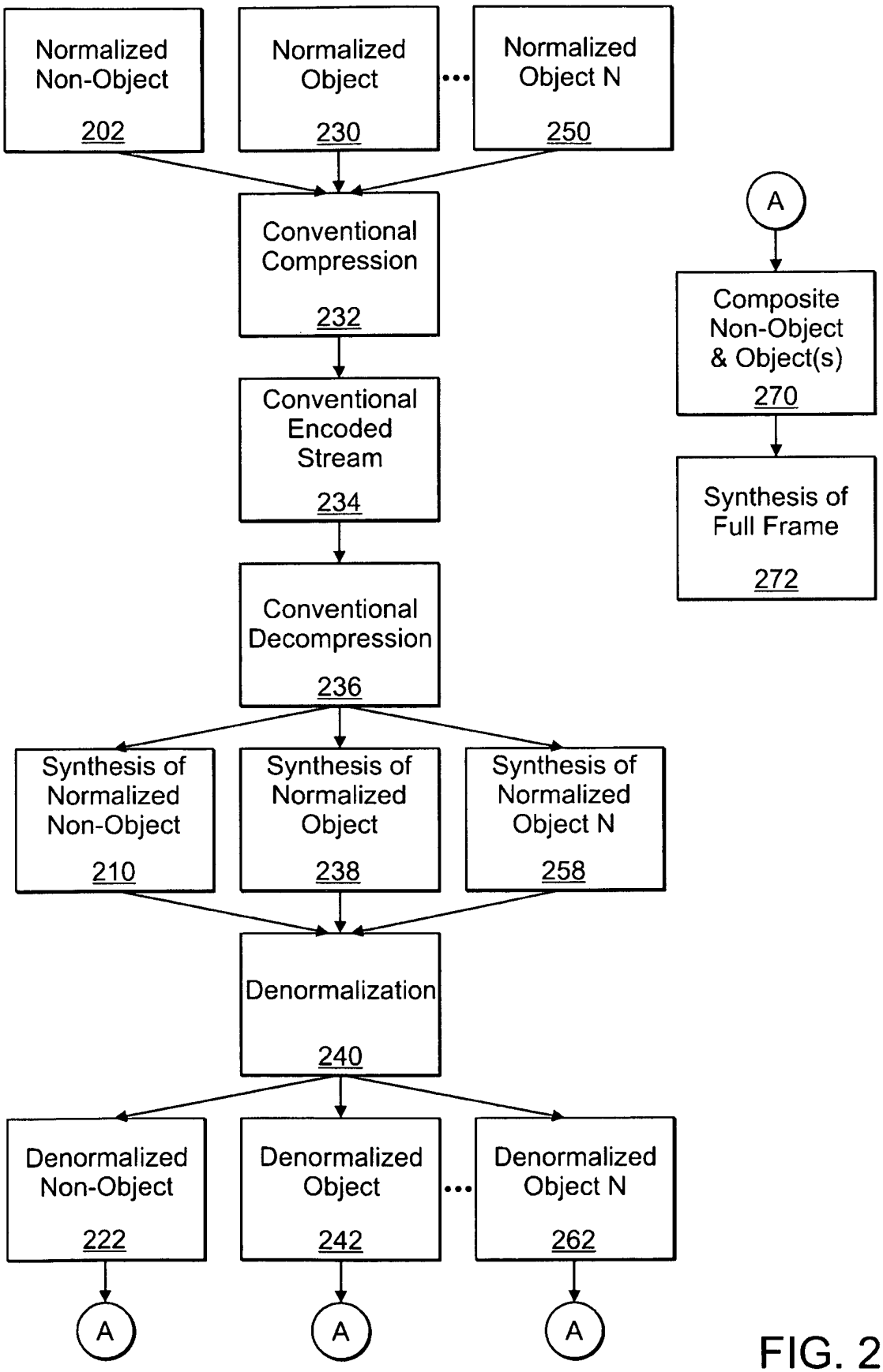
Normalized Non-Object 202
Normalized Object 230
Normalized Object N 250
Conventional Compression 232
Conventional Encoded Stream 234
Conventional Decompression 236
Synthesis of Normalized Non-Object 210
Synthesis of Normalized Object 238
Synthesis of Normalized Object N 258
Denormalization 240
Denormalized Non-Object 222
Denormalized Object 242
Denormalized Object N 262
A
A
A
A
Composite Non-Object & Object(s) 270
Synthesis of Full Frame 272

FIG. 2

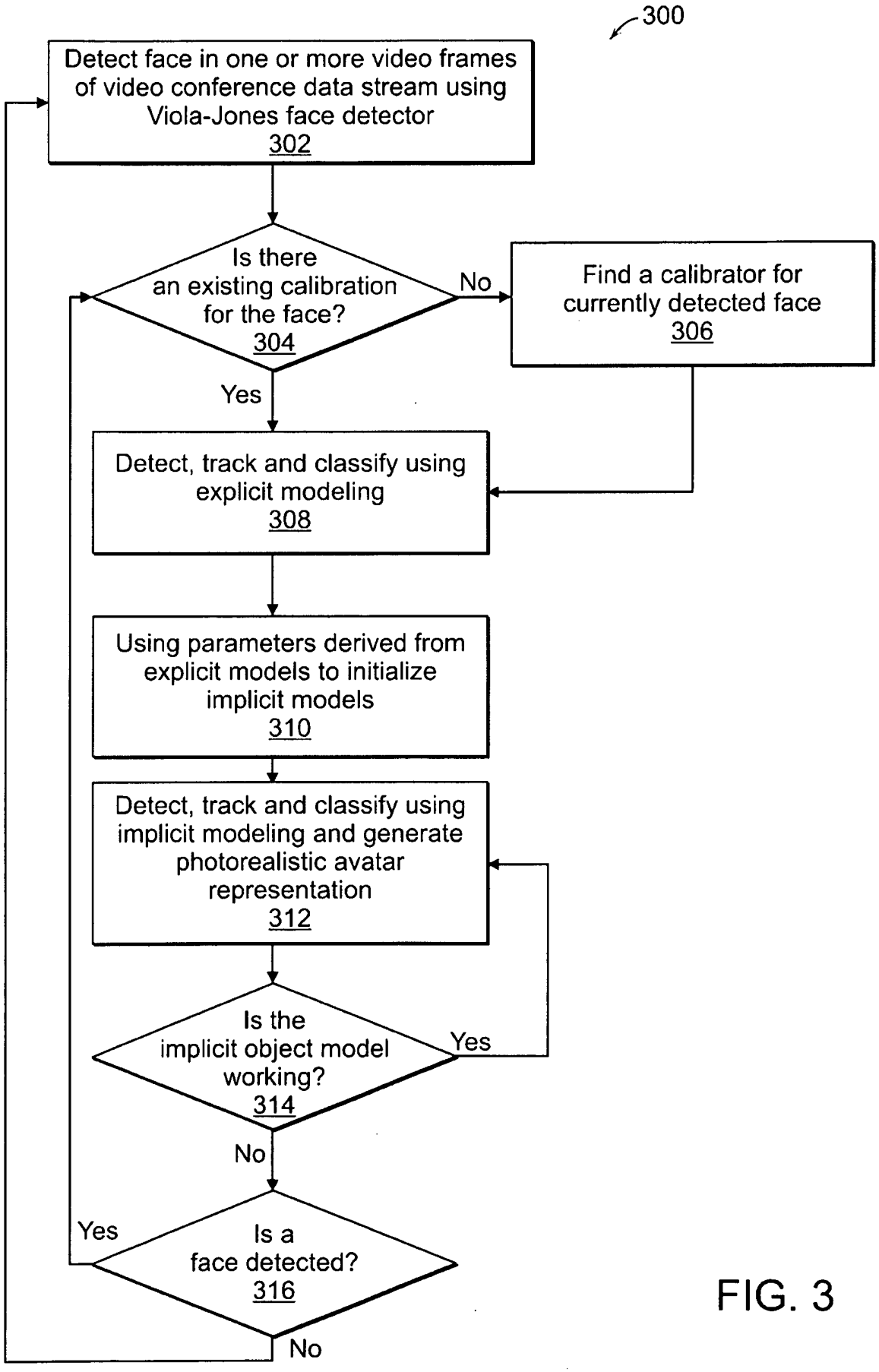
300
Detect face in one or more video frames of video conference data stream using Viola-Jones face detector
302
Is there an existing calibration for the face?
304
No
Find a calibrator for currently detected face
306
Yes
Detect, track and classify using explicit modeling
308
Using parameters derived from explicit models to initialize implicit models
310
Detect, track and classify using implicit modeling and generate photorealistic avatar representation
312
Is the implicit object model working?
314
Yes
No
Is a face detected?
316
Yes
No

FIG. 3

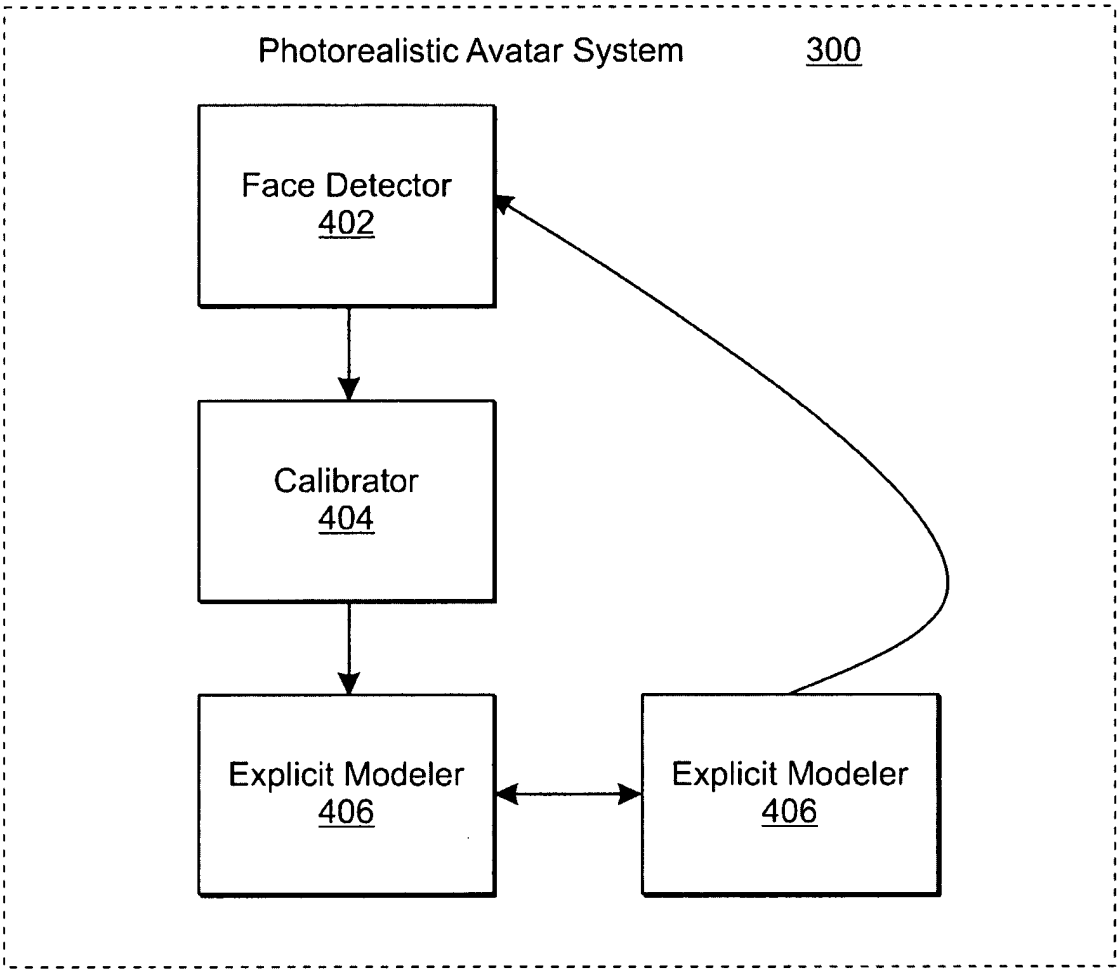
Photorealistic Avatar System
300
Face Detector
402
Calibrator
404
Explicit Modeler
406
Explicit Modeler
406

FIG. 4

500
Client Node
504
Instant Messaging Application
Video Conferencing Application
300
Codec
Server Node
502
Instant Messaging Server
Client Node
506
Instant Messaging Application
Video Conferencing Application
300
Codec

FIG. 5A

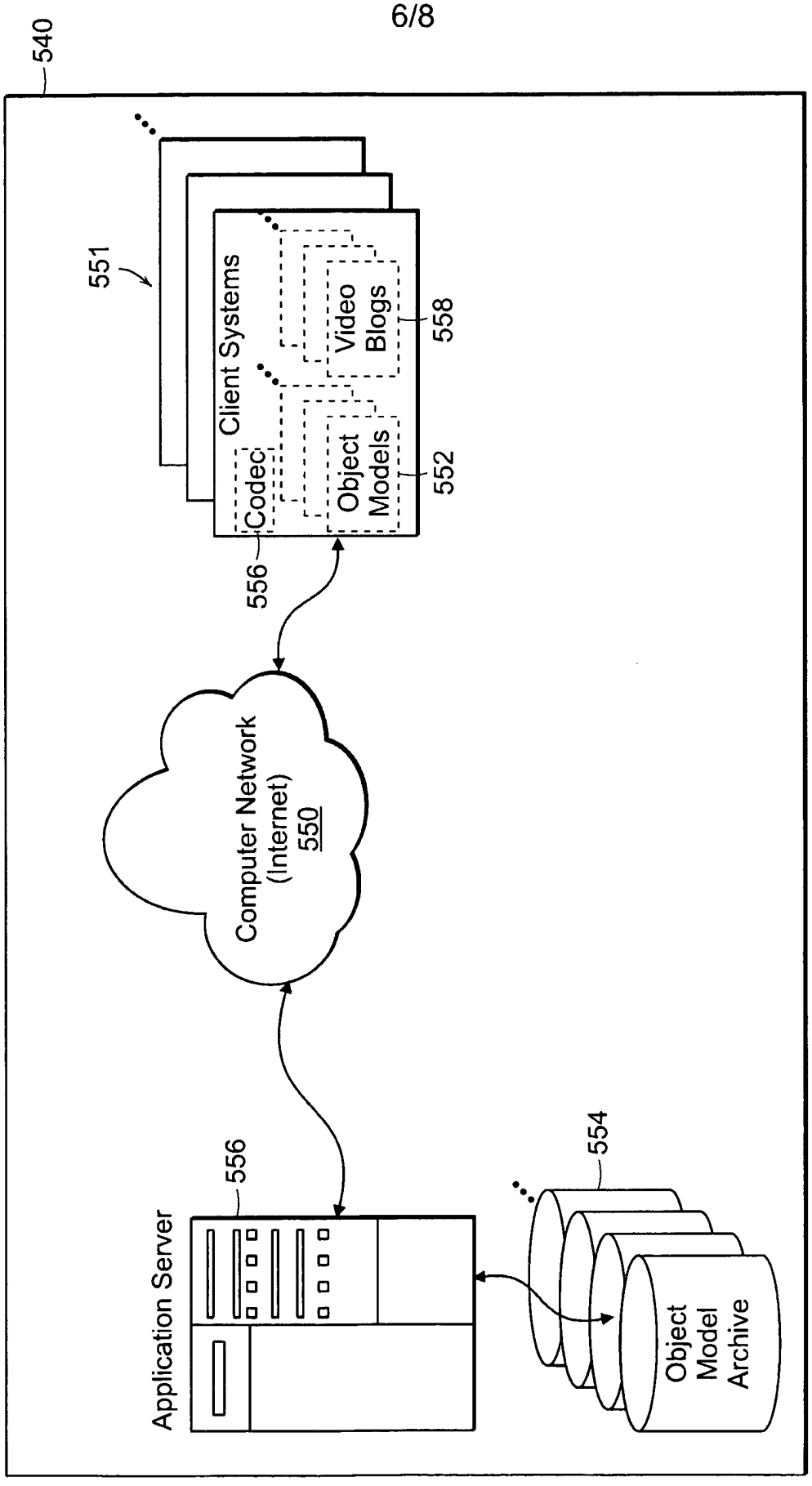
540
551
Client Systems
Codec
Object Models
Video Blogs
552
556
558
Computer Network (Internet) 550
Application Server
556
554
Object Model Archive

FIG. 5B

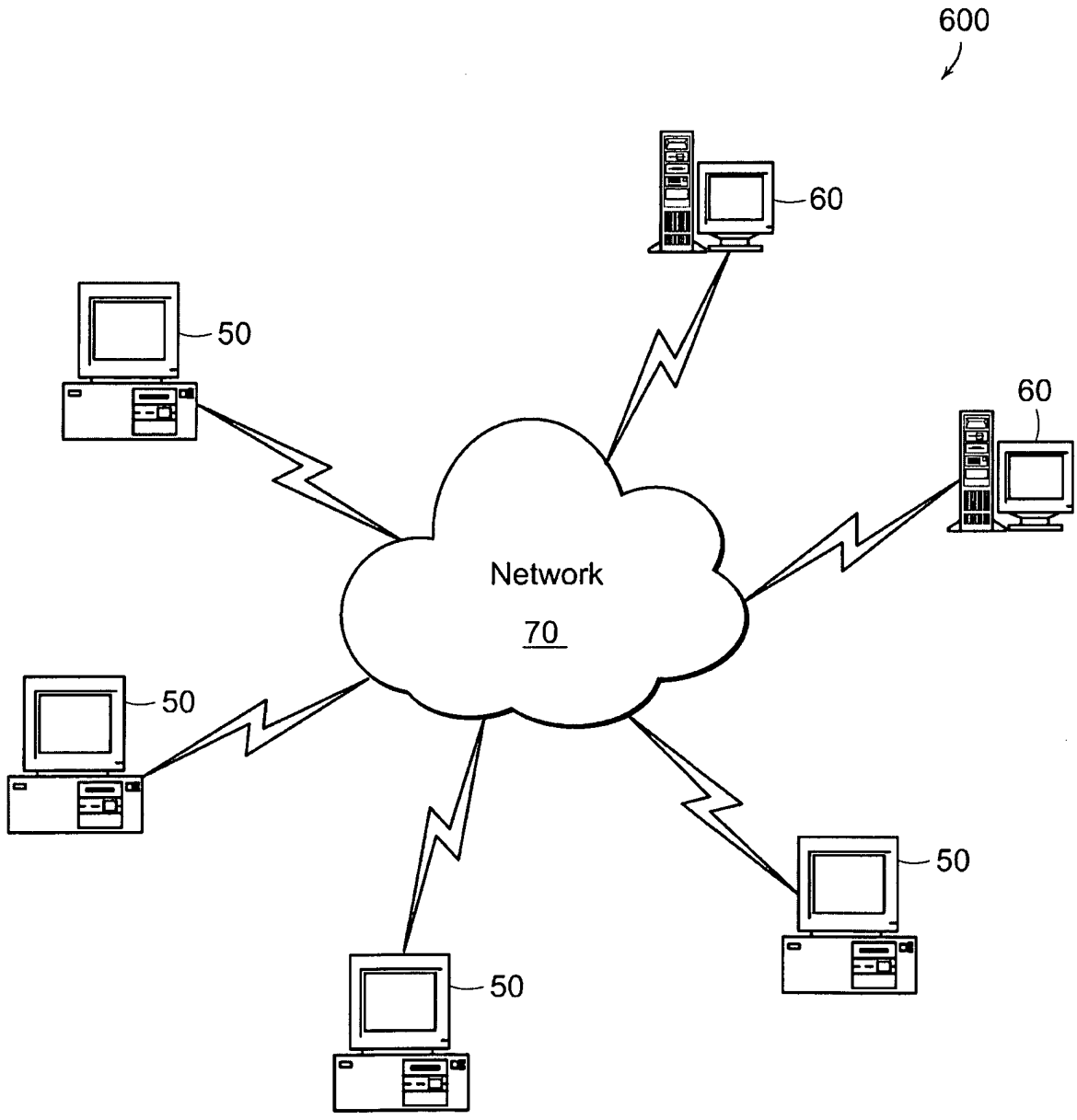

FIG. 6

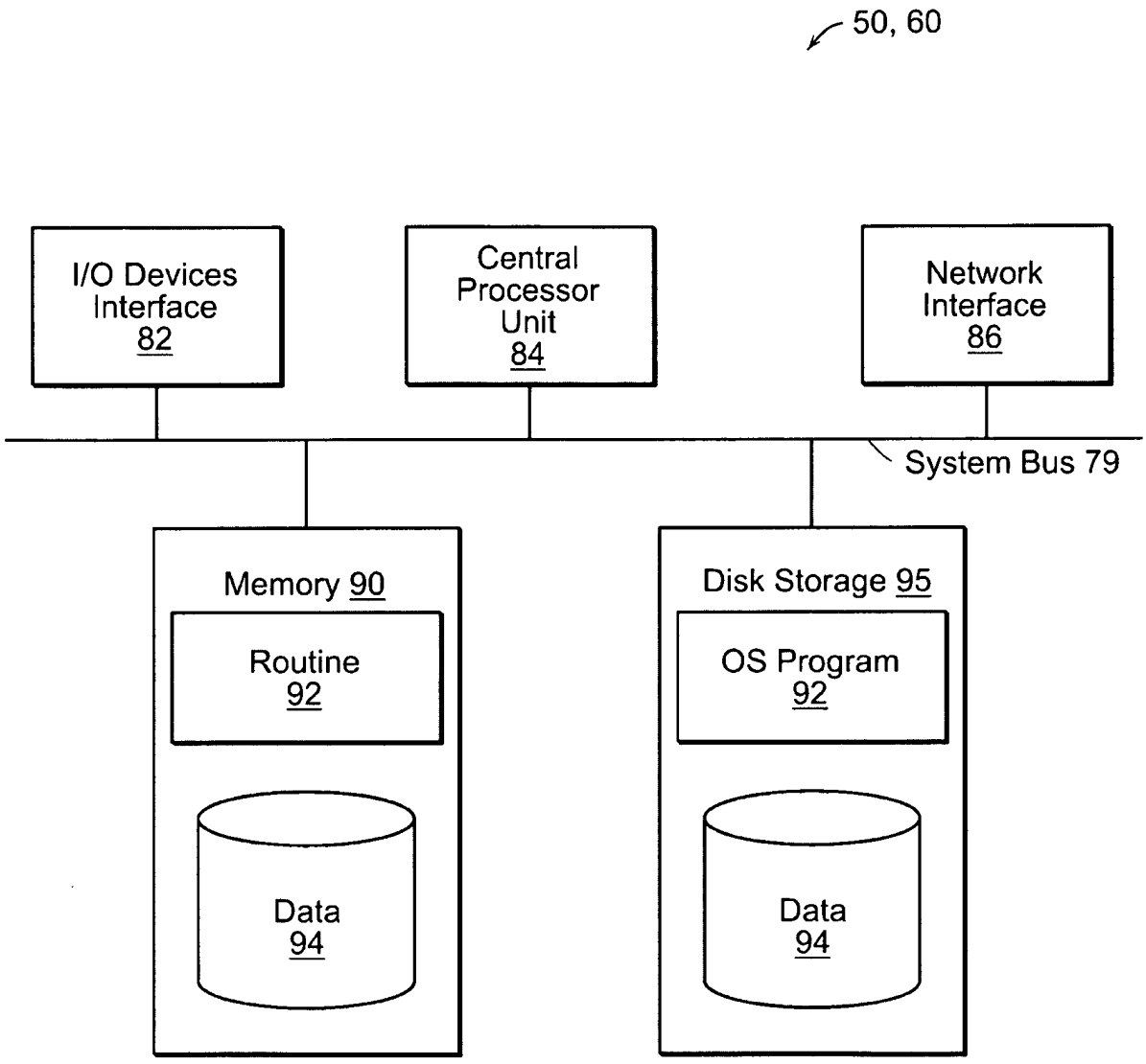
50, 60
I/O Devices Interface 82
Central Processor Unit 84
Network Interface 86
System Bus 79
Memory 90
Routine 92
Data 94
Disk Storage 95
OS Program 92
Data 94

FIG. 7