

(19) World Intellectual Property Organization
International Bureau



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

PCT

(10) International Publication Number
WO 2008/091484 A2

(51) International Patent Classification: **Not classified**

(21) International Application Number:
PCT/US2008/000091

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

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/881,982 23 January 2007 (23.01.2007) US

(71) Applicant (for all designated States except US): **EUCLID DISCOVERIES, LLC** [US/US]; Suite 212, 60 Monument Square, Concord, MA 01742 (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: OBJECT ARCHIVAL SYSTEMS AND METHODS

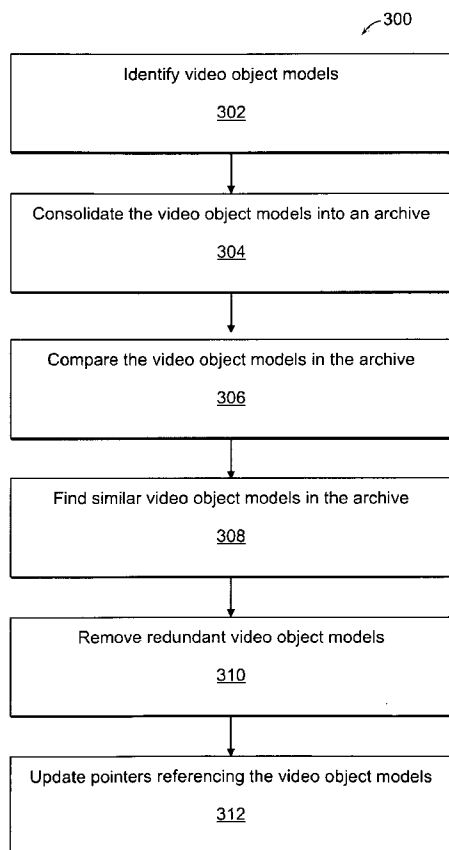


FIG. 3

(57) Abstract: Personal object based archival systems and methods are provided for processing and compressing video. By analyzing features unique to a user, such as face, family, and pet attributes associated with the user, an invariant model can be determined to create object model adapters personal to each user. These personalized video object models can be created using geometric and appearance modeling techniques, and they can be stored in an object model library. The object models can be reused for processing other video streams. The object models can be shared in a peer-to-peer network among many users, or the object models can be stored in an object model library on a server. When the compressed (encoded) video is reconstructed, the video object models can be accessed and used to produce quality video with nearly lossless compression.

WO 2008/091484 A2

OBJECT ARCHIVAL SYSTEMS AND METHODS

RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional Application No. 60/881,982, 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.

15 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.

Unfortunately, conventional video compression schemes suffer from a number of inefficiencies, which manifest 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, while retaining 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. A video signal can be processed to create object models from one or more objects represented in the video signal. The object models can be archived. The archived object models can be used as a library of object models for structure, deformation, appearance, and illumination modeling. One or more of the archived object models can be used when processing a compressed video file. The one or more archived object models and a codec can be used to reconstruct the compressed video file. The object models can be used to create an implicit representation of one or more of the objects represented in the video signal.

The object models in the archive can be compared to determine whether there are substantially equivalent object models stored in the archive. The size of the archive can be reduced by eliminating redundant object models that are substantially equivalent to each other. Object models in the archive that are similar can be combined.

A video codec can be used to reconstruct the compressed video file. The object models can be stored separately from the video codec. The object models can be included or bundled with the video codec. A customized codec can be created by grouping several of the object models. The customized codec can be optimized to reconstruct the compressed video file.

The compressed video file can be associated with a group of other compressed video files having similar features. The customized codec can be optimized to reconstruct any of the compressed video files in this group. The group of compressed video files can be determined based on personal information about a user. The personal information about a user can be determined by analyzing uncompressed video files provided by the user. When the uncompressed video files provided by the user are analyzed, reoccurring objects depicted in the uncompressed video files provided by the user can be identified. The reoccurring objects, for example, can be particular human faces or animals identified in the uncompressed video files provided by the user. Customized object models can be created that are trained to reconstruct those reoccurring objects. The customized objects can be used to create a customized codec for reconstructing the compressed video file.

The compressed video file can be sent from one user computer to another. While this compressed video file is being reconstructed, the archived object models can be accessed from a server. The server can be used to maintain and mine the archived object models for a plurality of users. The server can create an object model library. In this way, a video processing service can be provided, where members of the service can store their object models on the server, and access the object models remotely from the server to reconstruct their compressed video files.

The archived object models can be shared among a plurality of user computers in a peer-to-peer network. A request for the compressed video file from one computer in the peer-to-peer network can be received. In response to the request, one of the archived object models can be sent from a different user computer in the peer-to-peer network. Also in response to the request, another one of the archived object models can be sent from yet another computer in the peer-to-peer network. Further in response to the request, another one of the archived object models, or a sub-partitioning of those models can be sent from yet another user computer in the peer-to-peer network. In this way, the archived object models can be maintained and disseminated using a distributed approach.

One or more of object models can be used to control access to the compressed video stream. The object models can be used with a codec to reconstruct the compressed video file. The video file may not be reconstructed or

rendered on a user's computer without using one or more of the object models. By controlling access to the object models, access (e.g. playback access) of the compressed video file can be controlled. The object models can be used as a key to access the video data. The playback operation of the coded video data can depend
5 on the object models. This approach makes the compressed video data unreadable without access to the object models. In this way, the object models can be used as a form of encryption and digital rights management. Different quality object models can be used to provide different quality levels of the decompressed video from the same video file. This allows for a differential decoding of a common video file.
10 (e.g. a Standard Definition and High Definition version of the video based on the object model used and a common video file).

One or more of the object models can include advertisements that cause ads to be inserted into the reconstructed video stream upon playback. For example, during reconstruction (e.g. playback) of the encoded video, the models can cause
15 frames that provide advertisement to be generated into the playback video stream.

A software system for processing video can be provided. An encoder can process a video signal to create object models for one or more objects represented in the video signal. An object library can store the object models. A decoder can use a codec and one or more of the archived object models from the object library when
20 reconstructing a coded video file.

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
25 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.

30 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 archiving object models in a preferred embodiment.

FIG. 4 is a schematic diagram illustrating an example of the architecture of a personal video processing service of the present invention using a client-server
5 framework.

FIG. 5 is a block diagram illustrating the present invention sharing of object models.

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

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

DETAILED DESCRIPTION

15 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
5 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
10 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.

15 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
20 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
25 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
30 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
5 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
10 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
15 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
20 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
25 disproportionate bandwidth relative to other portions of the video signal 101. One method related to the 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 purposes of coherence monitoring, parts of the video signal that correspond to each other across sequential
30 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:

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

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 discussed above;

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
 5 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
 10 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
 15 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
 20 accounts for cardinal deformation of a cardinal pose (e.g., eyes opened/closed, mouth opened/closed, etc.) Thus appearance model 108 $AM(\sigma)$ is represented by cardinal pose P_c and cardinal deformation Δ_c in cardinal pose P_c ,

$$AM(\sigma) = \sum_i (P_c + \Delta_c P_c) \quad \text{Equation 2}$$

The pels in the
 appearance model 108 are preferably biased based on their distance and angle of
 25 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,
 30 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.

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

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. [cpp: might want to change "Normalized Object" to be "Normalized Object 1."] 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.

Data Mining Object Models

By archiving these object models (e.g. deformation, structure, motion, illumination, and appearance models), persistent forms of these object models can be determined and reused for processing other video streams. For example, when digital video is imported from a camera, the digital video can be transcoded and the video object archive can be accessed to determine whether any of the object models match. Although this can be done on a frame by frame basis, preferably the portions

of the video stream or the entire video stream can be analyzed using batch processing by grouping together similar items. The frames can be analyzed in a non-sequential manner, and a statistical analysis can be performed to determine which object models provide the best fit for coding.

5 FIG. 3 is a flow diagram illustrating the process 300 of archiving object models. At step 302, the object models are identified as discussed above. At step 304, the object models are consolidated into an archive or object model library. At step 306, the object models are compared and, at step 308 similar object models are identified. At step 310, the redundant object models can be removed, and similar
10 models can be consolidated. At step 312, pointers/identifiers to the video object models can be updated. Pointers to object models used in an encoded video stream, for example, can be updated to reference the relevant, updated object model in the library.

In this way, the present archival system 300 can mine these object models in
15 the object library and analyze object models to identify similar object models. Once the similar object models are identified, the system 300 can capitalize on the redundancy by creating generic object models that can be used over and over again for processing other video. The similarity tends to be based on similar structure, deformation, motion, illumination, and/or appearance.

20 The object models can be used for subsequent video processing in any number of ways. As discussed in more detail below, the models can be used in a client/server framework, the object models can be bundled into a package with the video codec for use when decoding encoded video file, the models can be used in connection with a personal video service, and the models can be distributed and
25 made available to many users using a distributed system, such as a peer-to-peer network. Also, the processing of the models can occur in a distributed computing network.

Personal Video Processing Service

30 In the example where the object models are stored on a server, a personal video processing service can be provided. FIG. 4 is a diagram illustrating an example of the architecture of a personal video processing service 400 using a client

414 server 410 framework. In this example, a user or member of the personal video service can use the present invention software to transcode all of their video files 418 using object based video compression. During the transcoding process, object models 416 are generated. The object models can be uploaded to an object model library 404 as part of the personal video service. When a member of the service transmits an encoded video file 418 to another member, the file size can be reduced substantially. During playback on the other member's system, the relevant object models 404 can be accessed from the server 410 to process and render the encoded video stream.

10 The system 400 can analyze the object models uploaded from a particular member and determine whether there are redundant object models. If, for example, the member continually transcodes digital video that depicts the same subjects, e.g. the same faces, same pets, etc., it is likely that the same object models will be created over and over again. The system 400 can capitalize on this redundancy by creating a cache of object models that are personal to the user (e.g. a cache of face object models, pet object models, etc.). The system can further capitalize on this redundancy by creating a codec 417 that is customized and personal to that user. The codec 417 can be bundled with the object models 416 that are particular to that user.

20 By having a substantial amount of members uploaded their models 416 to the server 410, the models can be analyzed to identify common or similar models. The most commonly used or generated models can be tracked. In this way, the system 400 can learn and determine what models 416 are the most likely to be needed, and a codec can be designed to include only the most important object models.

25 If a user tries to process an encoded video with the codec and the particular model has not been bundled with that codec, the system can access the server 410 to obtain the necessary models from archive 404. The codec may also access the server 410 periodically to update itself with new and updated object models.

30 As a further embodiment, the encoded videos could be such that the original "conventional" encoding of the video file is accessible on the client node 414. In this case, the advantage of the processing is used for transmitting the video, while more "conventional" compression is used to store the video on the hard disk to

facilitate more conventional processing of the video. For instance, if a video editing application wishes to use a different format, then the present inventive method can primarily be utilized during transmission of the video file.

5 Tuning the Codec

The codec 417 can be tuned to particular types of encoded video data. For example, if the video stream has a reoccurrence of certain objects, a common theme or particular style throughout, then the object models can be reused when reconstructing the entire encoded video file. Similarly, the codec 417 can be optimized to handle these reoccurring objects, such as faces. Likewise, if the video stream is a movie that has certain characteristics, such as a film of a particular genre, such as action film, then it may use similar object models 416 throughout the film. Even where the digital video is a film noir, for example, which is often characteristic of a low-key black-and-white visual style, then particular lighting and illumination object models may be applicable and used when reconstructing the entire encoded version of the movie. As such, there may be common object models (e.g. structure and illumination models) that are applicable to a substantial portion of the encoded movie. These models can be bundled together to create a customized codec.

Sharing Object Models

The object models could also be shared among any number of users. The object models can be stored on a server or in a database so they can be easily accessed when decoding video files. The object models may be accessed from one user computer to another user computer. FIG. 5 is a block diagram illustrating the sharing of object models. The object models can be accessed from the object model library 502 on the server 504, or they can be accessed from other client systems 510, 520. A respective object model manager 512, 522 can manage the object models 514, 524 that are needed on each client 510, 520 to process the encoded video files. The object model manager is similar to a version control system or source control management system, where the system software manages the ongoing development of the object models 514, 524. Changes to the object models can be identified by incrementing an associated number or letter code (e.g. a revision number or revision level) and associated historically with the change. In this way, the object models 514, 524 can be tracked, as well as any changes to the object models. This electronic tracking of the object models enables the system 500 to control and manage the various copies, versions, of the object models.

In addition to using a client-server framework, object models can be shared and distributed using a peer-to-peer network or other framework. In this way, users can download compressed video files and object models from other users in the peer-to-peer network. For example, if an encoded version of the movie Harry Potter were being downloaded from one system in the peer-to-peer network, to facilitate efficiency the relevant models, or partitions of those models, could be downloaded from other systems in the network.

25

Digital Rights Management

The process of deploying security schemes to protect access to digital video is long, involved and expensive. Content users want unfettered access to digital content without being required to undergo a burdensome authentication process. One of the most complicated aspects of developing a security model for deploying content is finding a scheme in which the cost benefit analysis accommodates all participants, i.e. the content user, content provider and software developer. At this

30

time, the currently available schemes do not provide a user-friendly, developer-friendly and financially effective solution to restrict access to digital content.

The object models of the present invention can be used as a way to control access to the encoded digital video. For example, without the relevant object models, a user would not be able to playback the video file. The object models can
5 be used as a key to access the video data. The playback operation the coded video data can depend on a piece of auxiliary information, the object models. This approach makes the encoded video data unreadable without access to the object models.

10 By controlling access to the object models, access to playback of the content can be controlled. This scheme can provide a user-friendly, developer-friendly solution, and efficient solution to restricting access to video content.

Additionally, the object models can progressively unlock the content. With a certain version of the object models, an encoding might only decode to a certain
15 level, then with progressively more complete object models, the whole video would be unlocked. Initial unlocking might enable thumbnails of the video to be unlocked, giving the user the capability of determining if they want the full video. A user that wants a standard definition version would procure the next incremental version of the object models. Further, the user needing high definition or cinema quality would
20 download yet more complete versions of the object model. Both the encoding and the object models are coded in such a way as to facilitate a progressive realization of the video quality commensurate with encoding size and quality, without redundancy.

Processing Environment

25 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
30 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., object models, codec and object model library discussed above). 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, hard drives, 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
5 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
10 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.

15 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
20 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 network of FIGs. 4-7 are for purposes of
illustration and not limitation of the present invention.

25 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.

30 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.

- 5 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.
- 10 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
15 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
20 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
25 network adapters.

 Further, in some embodiments, there may be the following advertisement feature.

Embedding Advertisements in the Video Using the Object Models

 The object models can be used to cause frames that include advertisements to
30 be inserted into the video stream during playback. In this way, the actual encoded video content would not need to be modified by the advertisements. However,

during reconstruction (e.g. playback) of the encoded video, the models can cause frames that provide advertisement to be generated into the playback video stream.

CLAIMS

What is claimed is:

1. A method of processing video, the method comprising the computer implemented steps of:
 - 5 processing a plurality of video signals to create object models for one or more objects represented in the video signals;
 - archiving the object models; and
 - using one or more of the archived object models when processing a compressed video file.
- 10 2. A method as in Claim 1 wherein archiving the object models further includes the computer implemented steps of:
 - 15 comparing one or more of the archived object models to determine whether there are substantially equivalent object models stored in the archive; and
 - reducing the size of the archive by eliminating the redundant, substantially equivalent object models.
- 20 3. A method as in Claim 2 wherein reducing the size of the archiving includes the computer implemented steps of:
 - 25 reducing the size of the archive by combining one or more object models into a common object; and
 - using the common object, generating one or more reduced original object models.
4. A method as in Claim 1 wherein archiving the object models further includes the computer implemented step of combining one or more of the archived object models that are similar.
- 30 5. A method as in Claim 1 wherein archiving the object models further includes the computer implemented step of grouping object models to

create a customized codec that is optimized to reconstruct objects in the compressed video file.

- 5 6. A method as in Claim 1 wherein the compressed video file is associated with a group of other compressed video files having similar features, and a codec is optimized to reconstruct the compressed video files in the group.
- 10 7. A method as in Claim 6 wherein the group of compressed video files is determined based on personal information about a user.
- 15 8. A method as in Claim 7 wherein the personal information about a user is determined by analyzing other video files provided by the user.
- 20 9. A method as in Claim 8 wherein analyzing the video files provided by the user further includes the computer implemented step of identifying reoccurring objects depicted in the video files provided by the user, the reoccurring objects including human faces or animals identified in the video files provided by the user.
- 25 10. A method as in Claim 8 wherein the video files provided by the user are encoded using a conventional encoding process.
- 30 11. A method as in Claim 10 wherein information contained in the conventionally encoded videos is used to determine how to process and encode the video.
12. A method as in Claim 10 wherein additional constraints are imposed on processing of the video files provided by the user to allow the conventional encoding of the video files to be recreated.

13. A method as in Claim 6 wherein the group of compressed video files is determined based on reoccurring objects depicted in the compressed video files.
- 5 14. A method as in Claim 5 wherein the customized codec is used to reconstruct the compressed video file.
- 10 15. A method as in Claim 1 wherein using one or more of the archived object models when processing a compressed video file further includes the computer implemented step of using the one or more archived object models and a codec to reconstruct the compressed video file.
- 15 16. A method as in Claim 1 wherein using one or more of the archived object models when processing a compressed video file further includes the computer implemented steps of:
receiving, at a user computer, the compressed video file from another user computer; and
accessing one or more of the archived object models from a
20 server.
- 25 17. A method as in Claim 16 wherein the server is used in connection with video processing service that maintains the archived object models on a server for a user, where the video processing service maintains other archived object models for other users.
- 30 18. A method as in Claim 1 wherein using one or more of the archived object models when processing a compressed video file further includes the computer implemented step of sharing the archived object models amongst a plurality of user computers in a peer-to-peer network.

19. A method as in Claim 18 wherein sharing the archived object models amongst a plurality of user computers in a peer-to-peer network further includes the computer implemented steps of:
- 5 receiving the compressed video file from a first user computer in the peer-to-peer network;
- in response to the request, sending one of the archived object models from a second user computer in the peer-to-peer network; and
- in response to the request, sending another one of the archived object models from a third user computer in the peer-to-peer network.
- 10
20. A method as in Claim 1 wherein one or more of the archived object models are used to control access to the compressed video file.
21. A method as in Claim 1 wherein one or more of the archived object models further include advertisements such that when the compressed video file is processed and reconstructed using one or more of the archived object models, one or more of the advertisements are inserted into the reconstructed the video file.
- 15
22. A method as in Claim 1 wherein using one or more of the archived object models when processing a compressed video file further includes creating an implicit representation of one or more objects using based on one or more of the archived object models.
- 20
23. A method as in Claim 1 wherein the object models include deformation models, structure models, and appearance models.
- 25
24. A computer system for processing video comprising:
- an encoder processing a video signal to create object models
- 30 for one or more objects represented in the video signal;
- an object library storing the object models; and

a decoder accessing a codec and the object library to use one or more of the archived object models when reconstructing a coded video file.

- 5 25. A computer system for processing video comprising:
- means for processing a plurality of video signals to create object models for one or more objects represented in the video signals;
 - means for archiving the object models; and
 - 10 means for using one or more of the archived object models when processing a compressed video file.

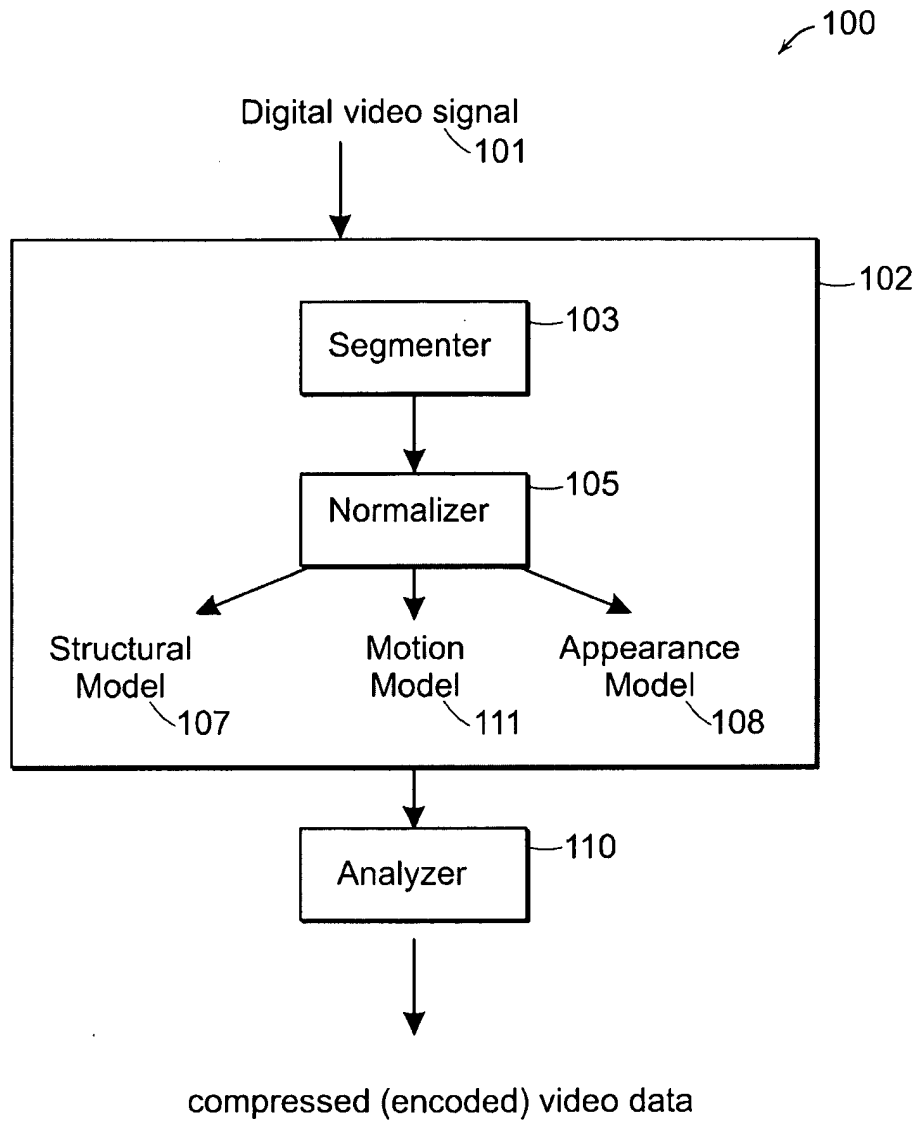


FIG. 1

2/7

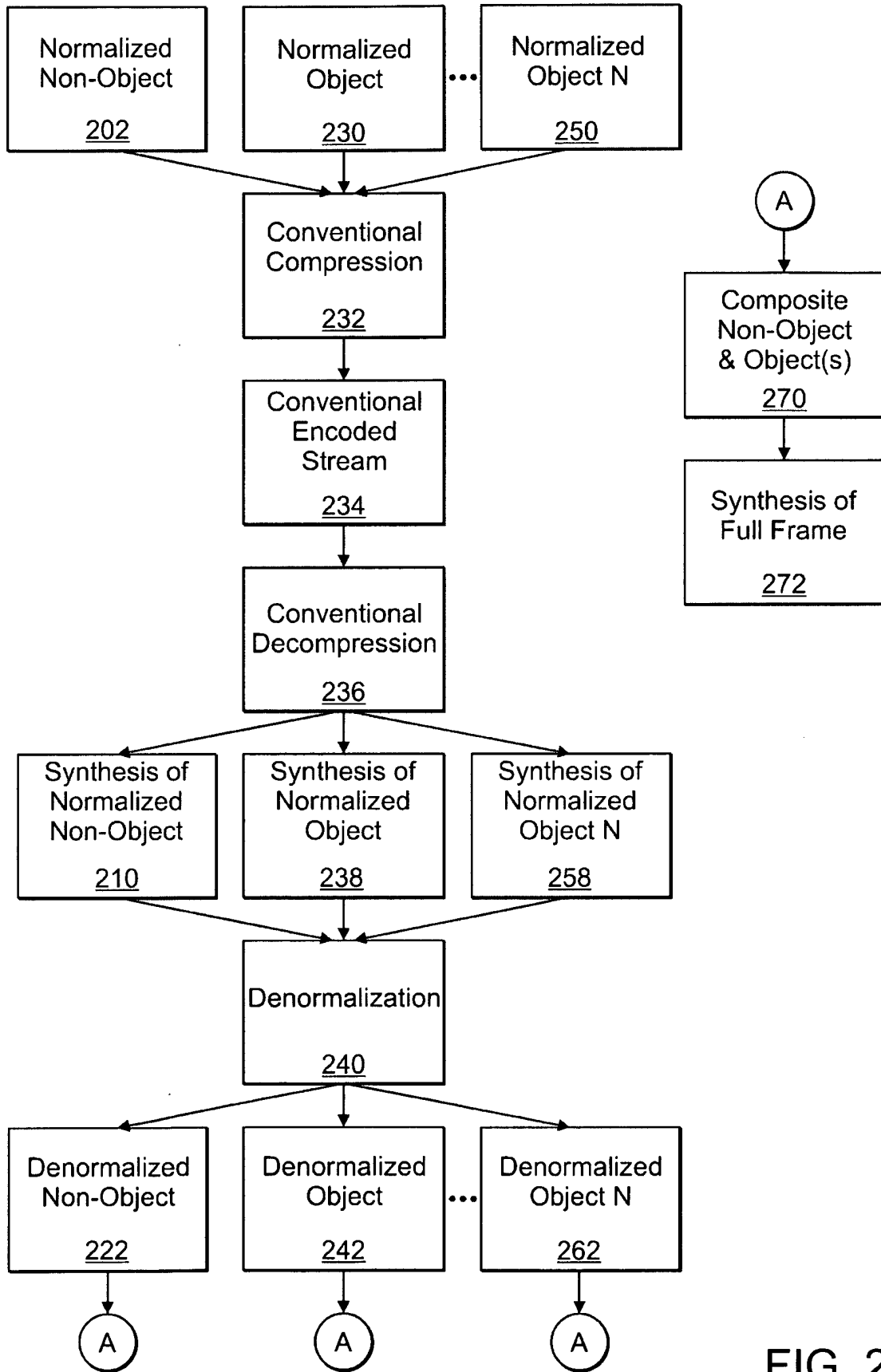


FIG. 2

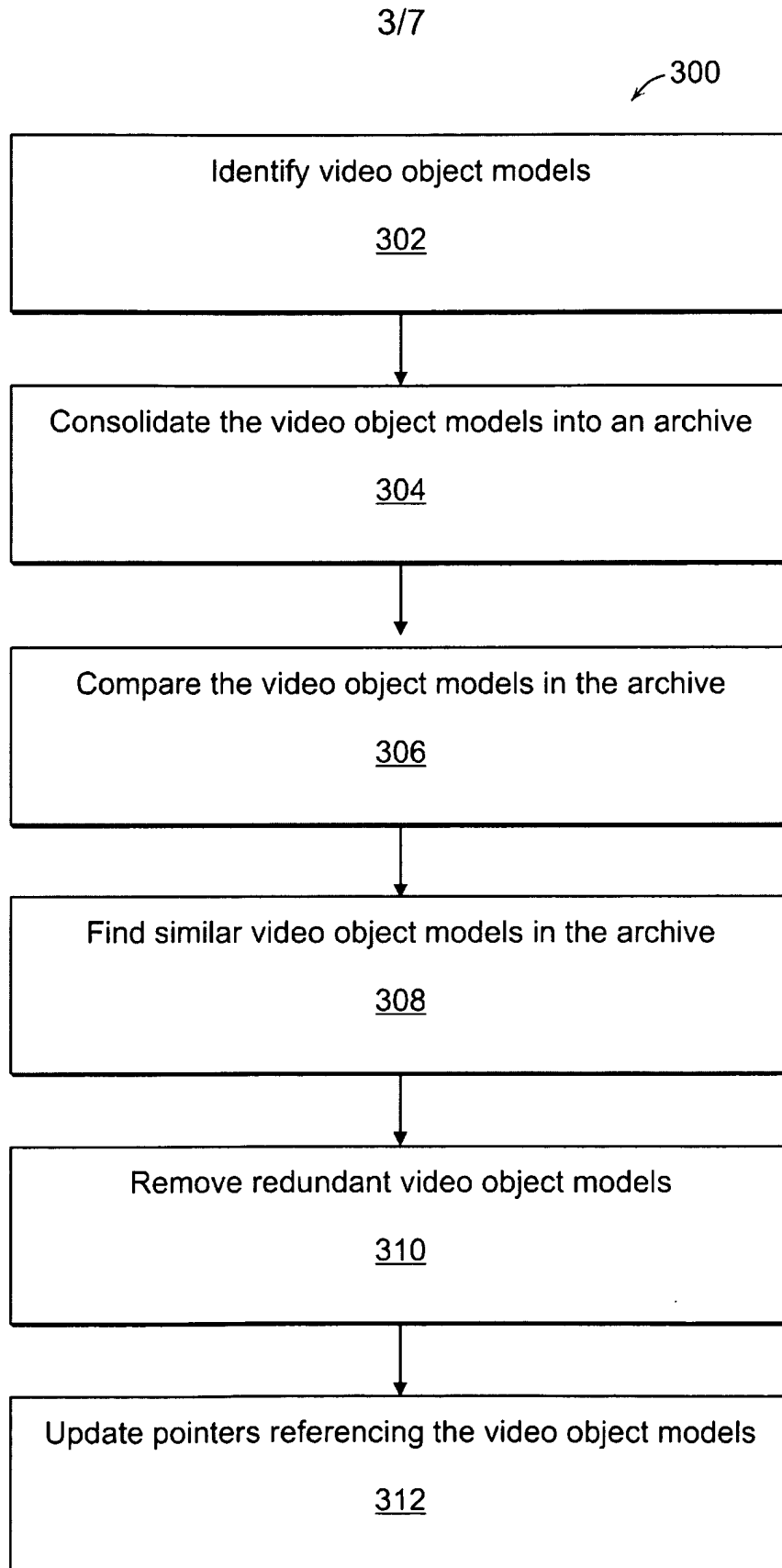


FIG. 3

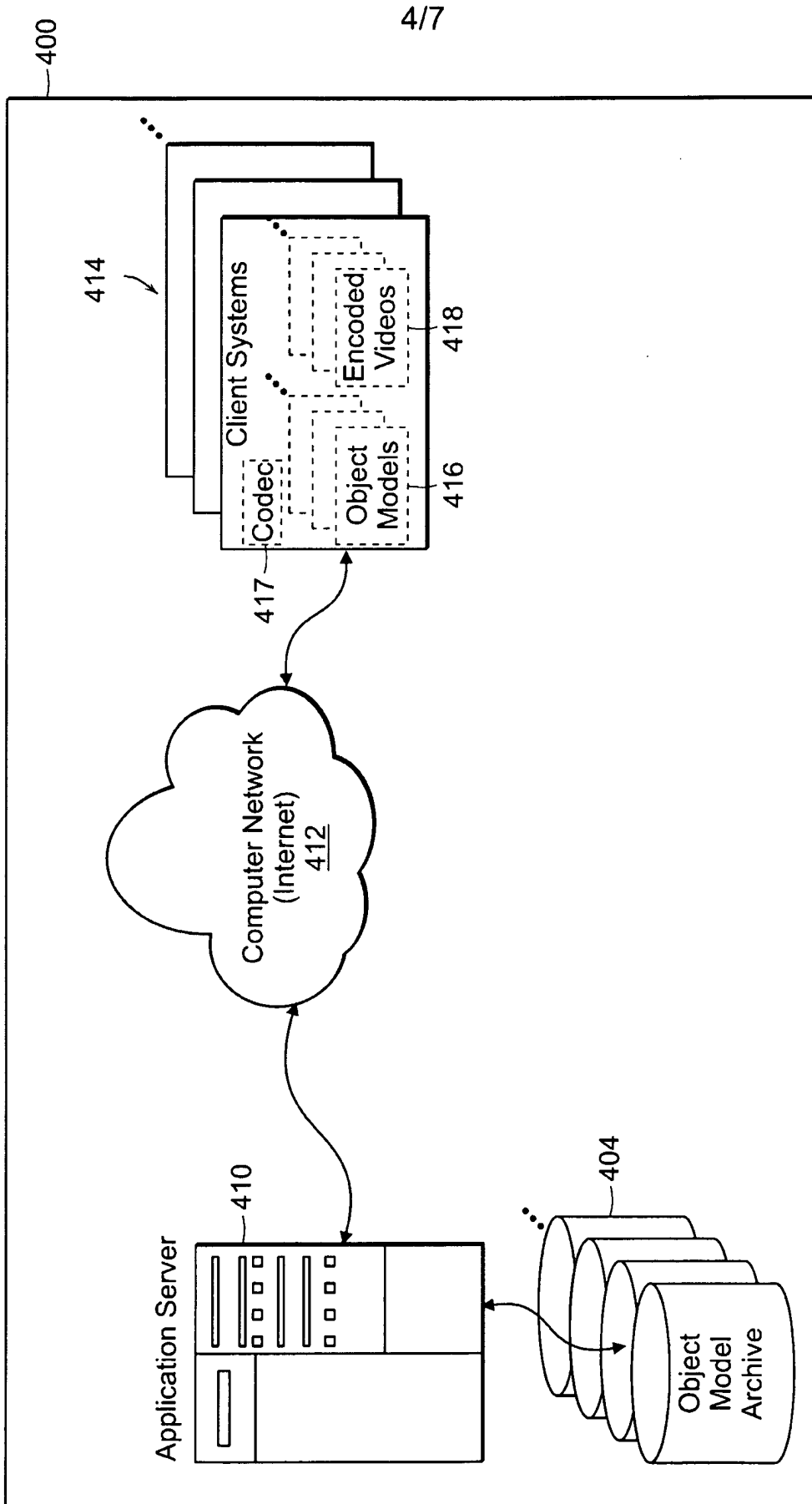


FIG. 4

500 ↙

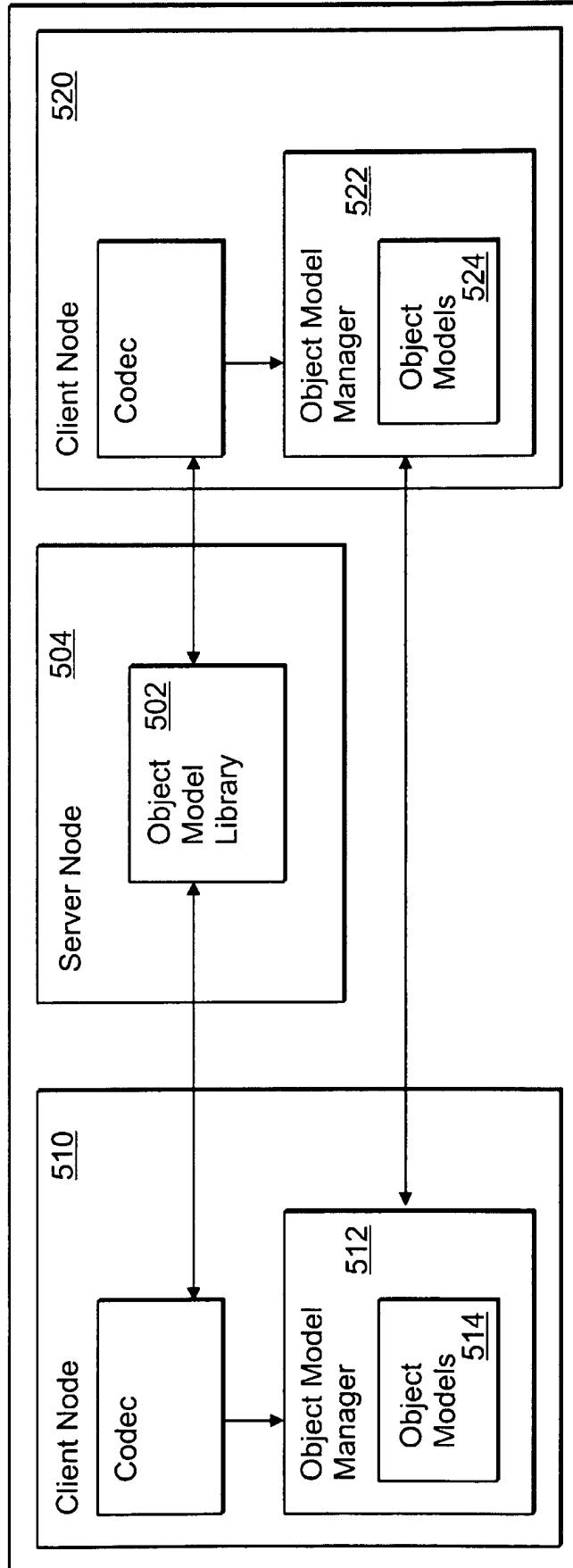


FIG. 5

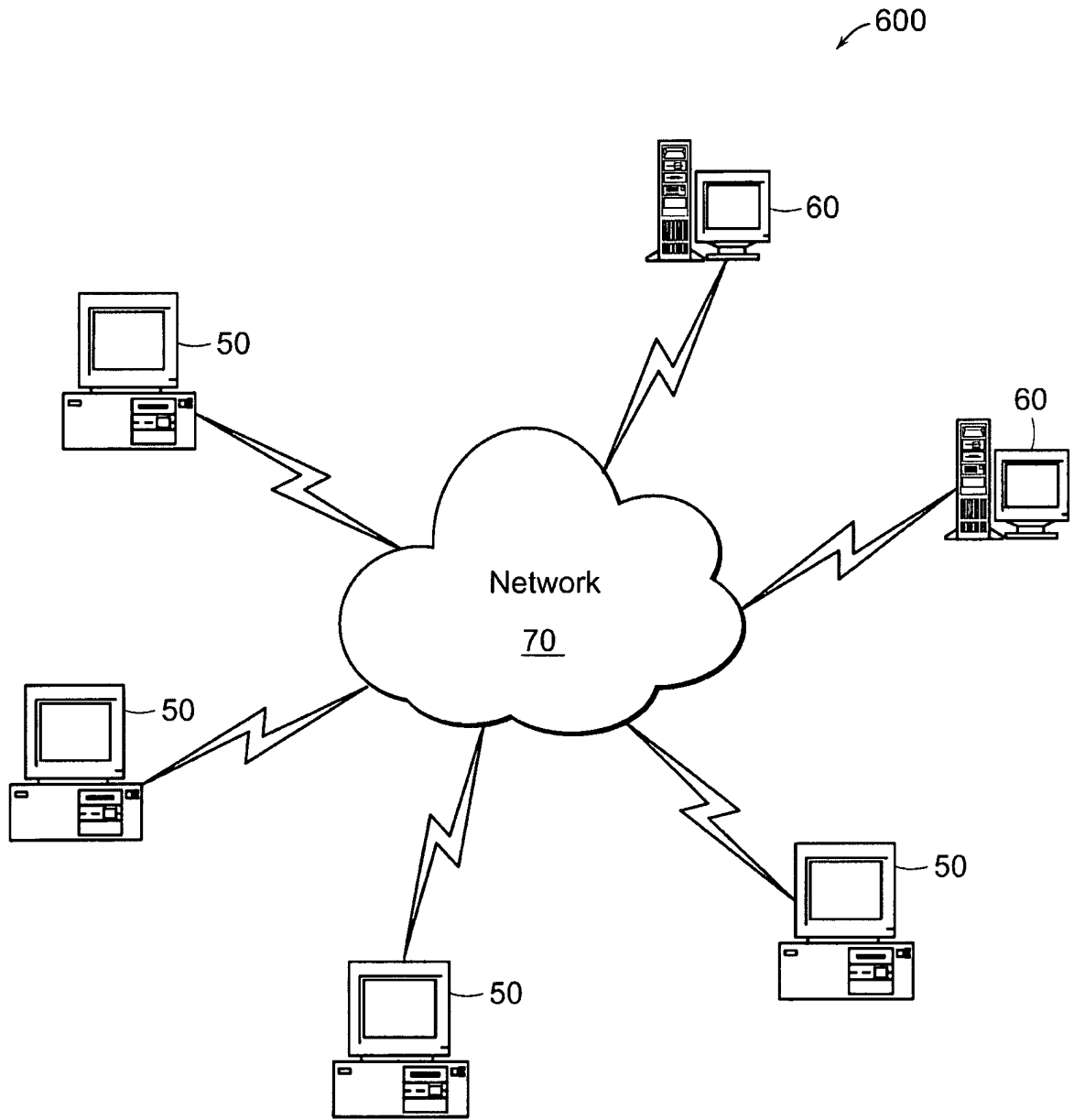


FIG. 6

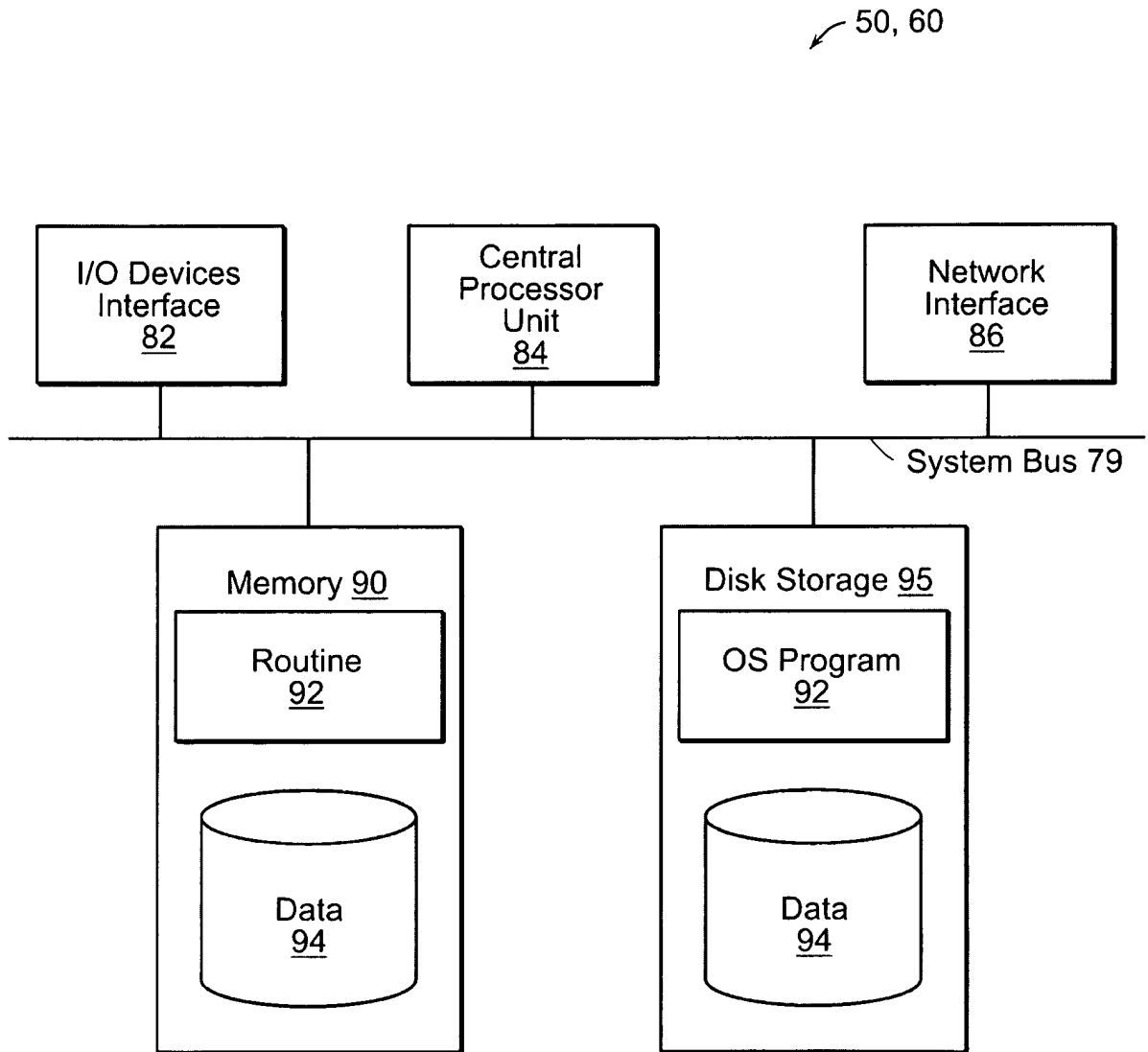


FIG. 7