(12) STANDARD PATENT APPLICATION (11) Application No. **AU 2008200965 A1**
(19) AUSTRALIAN PATENT OFFICE

(54) Title
Network Surveillance Systems

(51) International Patent Classification(s)
H04N 7/01 (2006.01) ***G08B 23/00*** (2006.01)

(21) Application No: **2008200965** (22) Date of Filing: **2008.02.28**

(43) Publication Date: **2009.09.17**
(43) Publication Journal Date: **2009.09.17**

(71) Applicant(s)
Canon Kabushiki Kaisha

(72) Inventor(s)
Patrick, Lachlan James;McLeish, David Grant;Simpson-Young, William

(74) Agent / Attorney
Spruson & Ferguson, Level 35 St Martins Tower 31 Market Street, Sydney, NSW, 2000

ABSTRACT

NETWORK SURVEILLANCE SYSTEM

Disclosed is a networked surveillance system (100) comprising at least one networked camera (110, 160) imaging a scene. A primary application (150, 190) is configured for communication with the camera via a network(115,135,175) according to a primary protocol (140,180) for viewing and recording data associated with the imaged scene. At least one instance of a secondary application or corresponding hardware (150), each instance being interposed in a communication path (115,135) of the network between the camera and the primary application such that the primary and secondary applications communicate using the primary protocol, and the secondary applications and the camera communicate using a secondary protocol (120). The secondary applications each interpret data associated with the imaged scene and transform the data in accordance with content (322, 332) identified from the data of the imaged scene, the transformed data (312) being communicated with the primary application via the primary protocol.

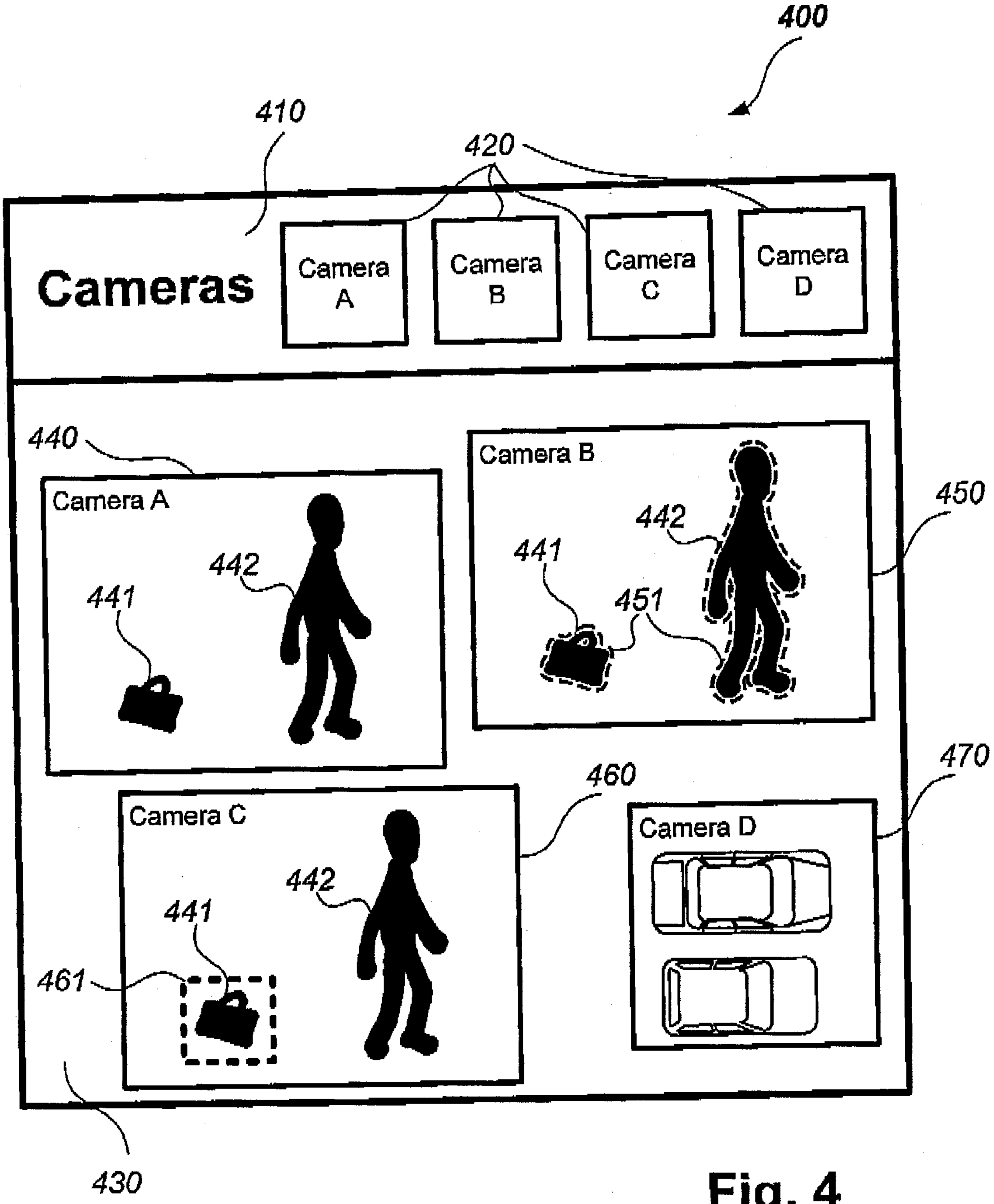

400
410
420
Cameras
Camera A
Camera B
Camera C
Camera D
440
Camera A
441
442
Camera B
442
441
451
450
460
470
Camera C
442
441
461
Camera D
430

Fig. 4

S&F Ref: 847751

AUSTRALIA

PATENTS ACT 1990

COMPLETE SPECIFICATION

FOR A STANDARD PATENT

Name and Address of Applicant :	Canon Kabushiki Kaisha, of 30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo, 146, Japan
Actual Inventor(s):	David Grant McLeish Lachlan James Patrick William Simpson-Young
Address for Service:	Spruson & Ferguson St Martins Tower Level 35 31 Market Street Sydney NSW 2000 (CCN 3710000177)
Invention Title:	Network Surveillance Systems

The following statement is a full description of this invention, including the best method of performing it known to me/us:

5845c(1148464_1):ROL

NETWORK SURVEILLANCE SYSTEM

Technical Field

The current invention relates generally to networked video surveillance systems and, in particular to object detection using a video camera using the emulation of a video camera network protocol to add functionality to an existing surveillance system.

Background

Video surveillance systems are commonly used for security and statistical analysis purposes. This is particularly the case in airports and shopping malls, where both traffic measurements and detection of lost or abandoned objects are important problems. Surveillance in manufacturing facilities may also be used for quality control purposes. Whilst security applications remain important, video surveillance systems are now being used for increasingly diverse applications.

Such video surveillance systems are typically implemented using a number of cameras coupled via a communication network to analysis computers, video storage databases and video viewing applications operable from client terminals. Often such systems grow over time, and so when upgrading or expanding such systems there are issues of scalability, interoperability, cost and a variety of functional video processing needs to be considered.

Scalability is a consideration because as cameras are added to the system, or lower-resolution cameras are replaced with newer models which output higher-resolution images, more video imagery must be analysed than before. Consequently, any object detection or tracking systems used must also be upgraded or augmented so as to process the additional volume of image data in a timely fashion. Furthermore, the storage databases may require new storage capacity to handle the additional footage. Viewing applications may have only been configured to cope with a fixed number of cameras, such as 16 or 64, and increasing the number of cameras beyond such a number can introduce user interface problems. Network bandwidth can also be strained by such addition or replacement of cameras.

Interoperability is also a consideration, since new models of existing cameras, or new brands of cameras, may be introduced into the system. Each such camera may communicate using a different protocol from that by which the system already interfaces. Correctly integrating such a camera may involve expensive upgrades to the viewing application software, storage database interface software, drivers, and object detection and processing software or hardware.

Functional needs of video processing systems may widely differ between different installations and customers. For example, an airport may need abandoned object detection for security purposes, a car park may need car counting for revenue accumulation, whereas a shop may need customer (person) entrance and exit information for market analysis. Integrating such diverse functions into existing surveillance systems is a difficult challenge because changes need to be made throughout the system.

Dealing with each of the above issues can incur substantial costs, including the need to test such new installations and fix problems on what could be critical security systems relied upon daily.

Summary

The arrangements disclosed herein in intended to work within surveillance systems, such as those described above, in particular with the presumption that object detection can be achieved using additional processors and the results sent to observers via a network such that some of the costs and scalability issues of upgrading the system are mitigated.

Disclosed is a method to expand video camera based surveillance system. A video surveillance system including an object detection module with an object tracker component is used to detect objects within a viewed scene. The object detection module with object tracker component operates within a processing unit which is physically external to the camera and any video client unit. The processing unit performs both video analytics operations and translation of some of the communication protocol messages between the camera and any video storage unit. The processing unit uses a pre-existing camera protocol which the client can already use. In particular, the processing unit passes object detection and tracking results to the client using the protocol, in addition to video imagery. Furthermore, the processing unit's operational parameters can be adjusted using that protocol, even if the protocol does not explicitly allow for such parameter adjustment to object detection operation. By mapping the existing messages and controls which the protocol does provide onto object detection results and parameters, it is possible to enable the expansion of such surveillance systems through additional cameras and processing units, with little need for upgrading the software or hardware of other components of the system.

Also disclosed is a networked surveillance system comprising:

at least one networked camera imaging a scene;

a primary application configured for communication with said camera via a network according to a primary protocol for viewing and recording data associated with the imaged scene; and

at least one instance of a secondary application, each said instance being interposed in a communication path of said network between the camera and the primary application such that the primary and secondary applications communicate using the primary protocol, and the secondary applications and the camera communicate using a secondary protocol;

the secondary applications each interpreting data associated with the imaged scene and to transform the data in accordance with content identified from the data of the imaged scene, the transformed data being communicated with the primary application via the primary protocol.

Other aspects are also disclosed.

Brief Description of the Drawings

At least one embodiment of the present invention will now be described with reference to the drawings in which:

Fig. 1 is a schematic block diagram representation of a video surveillance system according to the present disclosure;

Fig. 2 is a more detailed schematic block diagram of the processing unit of Fig. 1;

Fig. 3 illustrates an example user interface which may be used in the system of Fig. 1;

Fig. 4 illustrates another example interface for use with the system of Fig. 1; and

Fig. 5 is a schematic block diagram of a general purpose computer which may be used to implement the system of Fig. 1

Detailed Description including Best Mode

The arrangements to be described provide for adding object detection capabilities to an existing video surveillance system in an essentially user-transparent fashion. Particularly, it is desired for a user interface application (viewer) to not require alteration in order to operate in the increasingly capable system

Fig. 1 shows a networked surveillance system **100**, including a camera **110** arranged to capture which captures images of a scene and to produce a stream of video data. A communications channel **115** carries the video data from the camera **110** to a processing unit **130**. The communications channel **115** forms part of a communications network which may be implemented using wired or wireless technologies. Imagery conveyed within the video data stream is transmitted from the camera **110** to the processing unit **130** via the channel **115** by means of a network camera protocol **120**. Similarly, commands may also be issued to control the operation of the camera **110** over the channel **115** via the same protocol **120**.

The processing unit **130** receives the video data stream and is configured to perform object detection and tracking on the stream. The results of this processing, together with a transformed video stream, are transmitted to a client via a network **135**. The network **135** may be either wired or wireless and operates using a communications protocol **140** specific to a type of the camera **110**. The network 135 may be realised over a local network loopback device such that client and processing unit 130 can be executed on a single computer. The protocol 140 as such may be layered above and is distinct from standard communication formats such as HTTP and TCP/IP which are used to convey the protocol, its associated commands and the video streams. Commands from the client may also be received via the protocol **140**, and may affect, manipulate or otherwise adjust one or more processing parameters of the processing unit **130**. The protocols **120** and **140** may be identical, and so too may be the protocols **140** and **180**.

The client, which accepts the transformed video stream and object detection and tracking results, may store some or all of those data for later viewing. In one implementation, the client is a storage unit **150**. In another implementation, the client may be a viewer unit **190**, or a combination of the storage unit **150** and the viewer unit **190**. Typically the viewer unit **190** is implemented as a viewer application executable upon a client computer device, as will be described.

The viewer unit **190** is connected to the storage unit **150** via a network **170** operating under a protocol **180** or, by was of removable storage media **180**, either of which allows a user to view the contents of the storage unit **150**. In the latter respect, the storage unit **150** may be configured to operate by recording information onto a removable medium such as a hard disk or a DVD such that any viewer unit **190** can then be physically distinct from the storage unit **150**. In another implementation, the storage unit **150** and the viewer unit **190** may be integrated into one client unit, such as a desktop computer, such that there is no network needed to connect those two components. In yet another embodiment, the storage unit **150** and the processing unit **130** are integrated and the primary camera protocol **140** is only used to communicate between the viewer unit **190** (the client, representing a primary application) and the combined processing unit/server unit.

Fig. 1 also shows a second camera **160** connected to the storage unit **150** (or client) via a network **137** which operates according to the same communications protocol **140** described earlier. The camera **160** serves to illustrate that the communication protocol **140** used by the storage unit (client) **150** to communicate with the processing unit **130** need not be the same as the protocol **120** (a secondary protocol) used by the processing unit **130** (representing a secondary application) to communicate with the camera **110**.

The arrangements described above may be implemented using a computer system **500**, such as that shown in Fig. 5, wherein the processes of Figs. 1 to 4 may be implemented as a combination of hardware and software, including one or more application programs executable within the computer system **500**. The method of adding object detection capabilities to a surveillance system may be effected by instructions in the software that are carried out within the computer system **500**. The instructions may be formed as one or more code modules, each for performing one or more particular tasks. The software may include separate parts, in which a first part and the corresponding code modules performs the additional capabilities methods and a second part and the corresponding code modules manage protocol adaptation between the first part and any existing user interface, such as the viewer **190**. The software may be stored in a computer readable medium, including the storage devices described below, for example. The software is loaded into the computer system **500** from the computer readable medium, and then executed by the computer system **500**. A computer readable medium having such software or computer program recorded on it is a computer program product. The use of the computer program product in the computer system **500** preferably effects an advantageous apparatus for adding capability to an existing video surveillance system.

As seen in Fig. 5, the computer system **500** is formed by a computer module **501**, input devices such as a keyboard **502** and a mouse pointer device **503**, and output devices including a printer **515**, a display device **514** and loudspeakers **517**. An external Modulator-Demodulator (Modem) transceiver device **516** may be used by the computer module **501** for communicating to and from a communications network **520** via a connection **521**. The network **520** may be a wide-area network (WAN), such as the Internet or a private WAN operating using wired and/or wireless technologies. Where the connection **521** is a telephone line, the modem **516** may be a traditional "dial-up" modem. Alternatively, where the connection **521** is a high capacity (eg: cable) connection, the modem **516** may be a broadband modem. A wireless modem may also be used for wireless connection to the network **520**.

The computer module **501** typically includes at least one processor unit **505**, and a memory unit **506** for example formed from semiconductor random access memory (RAM) and read only memory (ROM). The module **501** also includes an number of input/output (I/O) interfaces including an audio-video interface **507** that couples to the video display **514** and loudspeakers **517**, an I/O interface **513** for the keyboard **502** and mouse **503** and optionally a joystick (not illustrated), and an interface 508 for the external modem **516** and printer **515**. In some implementations, the modem 516 may be incorporated within the computer module 501, for example within the interface 508. The computer module 501 also has a local network interface 511 which, via a connection 523, permits coupling of the computer system 500 to a local computer network 522, known as a Local Area Network (LAN). As also illustrated, the local network 522 may also couple to the wide network 520 via a connection 524, which would typically include a so-called "firewall" device or similar functionality. The interface 511 may be formed by an Ethernet^{TM} circuit card, a wireless Bluetooth^{TM} or an IEEE 802.11 wireless arrangement. As seen in Fig. 5, the cameras 110 and 160 are connected to respective ones of the networks 520 and 522.

The interfaces 508 and 513 may afford both serial and parallel connectivity, the former typically being implemented according to the Universal Serial Bus (USB) standards and having corresponding USB connectors (not illustrated). Storage devices 509 are provided and typically include a hard disk drive (HDD) 510. Other devices such as a floppy disk drive and a magnetic tape drive (not illustrated) may also be used. An optical disk drive 512 is typically provided to act as a non-volatile source of data. Portable memory devices, such optical disks (eg: CD-ROM, DVD), USB-RAM, and floppy disks for example may then be used as appropriate sources of data to the system 500.

The components 505 to 513 of the computer module 501 typically communicate via an interconnected bus 504 and in a manner which results in a conventional mode of operation of the computer system 500 known to those in the relevant art. Examples of computers on which the described arrangements can be practised include IBM-PC's and compatibles, Sun Sparcstations, Apple Mac^{TM} or alike computer systems evolved therefrom.

Typically, the application programs discussed above are resident on the hard disk drive 510 and read and controlled in execution by the processor 505. Intermediate storage of such programs and any data fetched from the networks 520 and 522 may be accomplished using the semiconductor memory 506, possibly in concert with the hard disk drive 510. In some instances, the application programs may be supplied to the user encoded on one or more CD-ROM and read via the corresponding drive 512, or alternatively may be read by the user from the networks 520 or 522. Still further, the software can also be loaded into the computer system 500 from other computer readable media. Computer readable media refers to any storage medium that participates in providing instructions and/or data to the computer system 500 for execution and/or processing. Examples of such media include floppy disks, magnetic tape, CD-ROM, a hard disk drive, a ROM or integrated circuit, a magneto-optical disk, or a computer readable card such as a PCMCIA card and the like, whether or not such devices are internal or external of the computer module 501. Examples of computer readable transmission media that may also participate in the provision of instructions and/or data include radio or infra-red transmission channels as well as a network connection to another computer or networked device, and the Internet or Intranets including e-mail transmissions and information recorded on Websites and the like.

In order to implement the surveillance system 100 of Fig.1, the computer system 500 of Fig. 5 may be configured such that the HDD 510 operates as the storage unit 150, and the viewer 190 operates as viewer application program having corresponding code modules to implement one or more graphical user interfaces (GUIs) to be rendered or otherwise represented upon the display 514 to permit viewing of a live or stored video stream.

Further, the processing unit 130 may be configured to operate as a further application program executable within the computer 501, as an application program executable within a server computer forming part of either of the networks 520 and 522, as an application program executable as part of the camera 110, or as a hardware device configurable in the communication path between the camera 110 and the computer 501. Through manipulation of the keyboard 502 and the mouse 503, a user of the computer system 500 and the viewer application may manipulate the interface to provide controlling commands and/or input the camera 160 or, processing unit 130.

It is noted that, with respect to Fig. 1, that the chain of components formed by the camera **160**, the network **137**, the protocol **140**, and the client (eg. the storage unit **150**, network **175**, protocol **180**, viewer **190**) represent an existing surveillance system formed with one camera. The present disclosure is concerned with adding capability to such a system without altering the protocols, thus permitting existing cameras, storage units and viewers to continue to be used. This additional capability is afforded by the incorporation of the processing unit **130** (in the path associated with camera **110**) which is configured to operate using the existing protocols and network connections.

The processing unit **130** performs object detection and tracking on a video stream input supplied by the camera **110**. The processing unit **130** passes the results of that processing to the surveillance system using the camera communication protocol **140** which the client already understands. By converting the communications protocols between the input camera **110** and the existing surveillance system, the processing unit **130** performs the dual functions of analysing the input imagery and allowing low-cost expansion of the system by addition of new cameras.

The camera communication protocol which the processing unit **130** produces must be a protocol which the surveillance system already understands. For example, the surveillance system **100** may have the existing camera **160** connected and which uses that protocol. Although the camera communication protocol **140** must be the same protocol as one that is already supported by the system **100**, it is not necessary for an existing camera **160** to be connected to the system **100** for the processing unit **130** to operate. It may be possible to reproduce a camera protocol which the surveillance system **100** understands but for which no corresponding cameras are currently attached to the system.

The reproduced protocol may allow for more than just the transmission of images or video streams. This permits the results of processing to be conveyed using the protocol **140**, and also allows the camera commands issued to the processing unit **130** from the viewer **190** via that protocol to be mapped to commands which change the operating parameters of the processing unit **130**. Thus the object detection results can be conveyed, but also the object detection parameters can be modified, for example by using simple camera controls that would be otherwise used to modify operation of the camera.

A preferred implementation uses camera motion controls such as pan, tilt or zoom commands to modify the operation of the processing unit **130**. Another implementation may pass the pan, tilt and zoom commands to the camera **110**, or may choose between passing the commands to the camera and modifying the operation of the processing unit 130 based on some condition, such as the state of an administrative setting. Other implementations may use frame rate, camera pre-set parameters, white balance controls, or other camera-specific commands to adjust the parameters of the processing unit 130.

The processing unit 130 may be implemented as software operating on computer hardware, or it may be implemented as hardware alone, or firmware, or some other mixture of hardware and software such as multiple processing units connected via a network. The important input and output characteristics of the processing unit 130 are that it receives an input stream of images from a video camera, and optionally controls that camera by steering it or adjusting the operational parameters of the camera, and that the output of the processing unit 130 is a camera protocol, and optionally the parameters of the processing unit 130 are controlled using that protocol.

Fig. 2 shows internal operational components of one implementation of the processing unit 130. The camera 110, which captures images of a scene and produces a stream of video data, transmits the video data using a network camera protocol 120 to the processing unit 130. Within the processing unit 130, the video stream is received by a camera interface 210.

The video stream is made available from the camera interface 210 to an object detection module 220. The object detection module 220 performs object detection on the received video stream. Many object detection methods are possible and are well known in the art. The precise method(s) used is not important to the present description. What matters is that analysis of the video stream produces higher level object data, for example, object silhouettes. The results of this object detection are made available, along with the video data, to an object processing module 230.

The object processing module **230** performs processing operations on the detected objects. This may include, but is not limited to, object tracking, entry/exit event determination, abandoned object detection, background modelling processing, and so on. The results of this object processing are made available, along with the video and object data, to a video transformation module **240**.

The video transformation module **240** transforms the received input video stream from the camera **110**, for example by drawing or compositing object silhouettes onto video frames, or by providing more abstract high-level symbolic representations of the object motions in accordance with detections made within the video content. The module **240** constructs or forms a transformed video stream, along with the object data and object processing results, which are passed to a transmitter **250**.

The transmitter **250** encodes both the transformed video stream and the object detection and processing results into the camera protocol **140**. The camera protocol **140** may differ from the communication protocol **120** used by the camera **110** or it may be the same. The encoded video stream representation is then transmitted to the client, in this example represented by the storage unit **150**, which records the video data to a storage medium. Alternatively, the client is may be the viewer 190 which displays the video data on a screen, such as the display 514.

The client receives the transformed video stream and object detection and tracking results and if the client is the storage unit **150**, the storage unit **150** stores all or a subset of them, for example in a database. Again, video storage methods are well known in the art.

The client can also issue commands to the processing unit **130** via the same communication channel and protocol **140**. These commands may be camera-specific commands such as pan, tilt and zoom commands, or they may be camera-specific eventrelated commands, such as a command to activate the motion detection infra-red sensor. Note, the camera **110** connected to the processing unit **130** might not have any infra-red detector at all, in which case the processing unit **130** may be configured to map such a command onto changes in its own internal processing parameters.

The commands are received from the client into a receiver **260** which transmits them to the object detection module **220**, which uses those commands to modify its operational parameters. For example, the command to pan, tilt, zoom, or activate an infra-red sensor, may be used to activate object detection for the scene, or to begin transmitting object detection results in some way, for example to start drawing (compositing) silhouettes around objects in the transformed video stream. Any mapping is possible. It is even possible for the object detection module **220** to cause the camera interface **210** to control the camera **110** to support some of these activities, for example by increasing the frame rate of the video stream to perform more accurate object tracking.

The video stream may be transformed by the processing unit **130** in transit. For example, the results of object detection may be partially indicated within that transformed video stream by drawing borders or silhouettes around objects using a bright colour which differs from the scene's background. Similarly, timestamps, processing parameters, and other context information may be drawn onto or otherwise embedded within the video stream itself. Such data may also be conveyed via the protocol **140** in other ways. For example, a camera might have a method of indicating motion within a scene using a motion indicator data field in its protocol, or it may be able to report door or window opening events using an external tripwire hardware attachment for the camera, or it might be able to report infra-red signals in a sub-channel alongside the visual spectrum imagery.

Such other components of the protocol **140** can be used by the processing unit **130** to convey the results of object detection and tracking.

It is important that the scope of the term 'protocol' is clarified. The presently disclosed arrangement relies on the emulation of a communication protocol which a camera uses. A camera of the particular camera model thus emulated may in fact be present, for example the camera **160**, in the surveillance system, or it might not. The important criterion is that the protocol is not a low-level communications protocol such as HTTP or TCP/IP, but rather a protocol that contains the highest-level camera-specific commands and data fields. The protocol must do more than simply stream images, since the system **100** requires that object detection results can be transmitted in other data fields besides merely the (transformed) video stream, such as fields for transmitting pan, tilt or zoom settings, camera pre-set positions, events, times, alarms, notifications, sensor alerts or settings. Similarly, the commands which can be issued to the processing unit **130** must be camera-specific commands, i.e. commands designed within the protocol for controlling a camera or modifying the settings or attached peripherals of the camera, such as pan, tilt, zoom, activate or deactivate motion detection, activate or deactivate an infra-red sensor, activate or deactivate a tripwire sensor.

The commands issued by the client (eg. storage unit **150**, viewer **190**) can modify the operation of the processing unit **130** in many ways. A preferred implementation allows the following mappings of camera commands to operational changes:

(i) When the client requests a higher frame rate from the processing unit **130**, the processing unit **130** responds by producing a higher frame rate within the transformed video stream, and also supplies object detection and tracking results at that higher rate.

(ii) When the storage server **150** requests a lower frame rate, the transformed video stream's frame rate is also reduced and the object detection results may be supplied at that lower rate. However, the camera **110** might still be transmitting a video stream to the processing unit **130** at a higher frame rate, in order to support more accurate object tracking.

(iii) When the client requests a zoom-in operation, the processing unit **130** might fulfil that request by a digital zoom facility, rather than asking its camera **110** to physically zoom into the scene. This is because zooming the camera lens might disrupt object detection or tracking by invalidating parts of the object detection module's model of the scene.

(iv) When the client requests a zoom-out operation, the processing unit **130** might fulfil that request by reversing any digital zoom already performed by it earlier, as mentioned in the previous paragraph. This has the same benefits as discussed earlier. If reversing the digital zoom is insufficient to fully satisfy the zoom-out request, the processing unit **130** may send commands to the camera to cause it to physically zoom out. The processing unit **130** may also produce lower-quality estimates of object positions and motions during the time that its model of the scene is refreshing to account for the newly visible scene.

(v) Similar mappings are possible with pan and tilt commands, wherein physical camera zoom may be augmented with digital zoom features within the processing unit **130** to ensure that some pan and tilt requests can be satisfied without physical motion of the camera and the consequential disruption to the scene model of the processing unit **130**.

(vi) Alternatively, zoom, pan and tilt operations which seek to follow the motion of an object in the scene may be interpreted by the processing unit **130** as requests to highlight or track that object, and instead of changing the field of view of the scene, may cause graphics to be drawn on the transformed video stream. For example, there could be a scene in which multiple moving objects are visible. Initially one object in the centre of the scene may be considered a primary object which is highlighted or selected (e.g. drawn with a silhouette around it). When a pan command is received by the processing unit **130** this could change the parameters of the processing unit **130** such that a different object is now considered the primary object. Panning to the left may thus cause a change to which object is highlighted or selected. The primary object could then be physically followed by motions of the camera, or this selection of a primary object may be used to provide additional data about that object, such as its time in the scene or its average speed or other data, either by drawing those data on the transformed video stream, or by sending additional information within the protocol data fields.

Many other mappings of camera operational controls to processing unit parameters are possible.

Fig. 3 illustrates an example of a user interface **300** displayed on the viewer unit **190** or client, such as via the display 514. The transformed video stream is displayed in a video playback region **310**, for which a single frame **312** is seen in Fig. 3. The frame **312** is a transformed video frame produced by the video transformation module **240** and is shown together with content objects (a suitcase or bag **322**, a person **332**) moving within the scene, by outlining the detected objects with a border or silhouette **320**, **330** respectively. Such borders may be bounding boxes, approximate boxes, ellipses or silhouettes, pixel-perfect silhouettes, polygons, block-based silhouettes, multi-blob approximations, or many other possible representations.

The user interface 300 is desirably an interactive graphical user interface (GUI) which also includes a video object time-line 340. The current time of playback or recording is indicated by a marker 390. This point may move across the timeline 340 or the times might move and the current time display may remain stationary, or some combination thereof. The user may even manipulate the current time marker by selecting components thereof, using a mouse pointing device 503 or keyboard 502, to replay video and object data recorded earlier or change the speed of viewing. In this fashion, components of the GUI 300 are preferably formed as user selectable icons or "buttons", as they are commonly known.

Indications of object lifetimes 350, 360, 370 are also shown. Each may have a visible "first appearance" marker 365, or "last appearance" marker 355. The lifetime may show the length of time during which the object has been detected in the scene 312.

Different colours, shading, textures, dashes, patterns, or other visual representations 350, 365 may be used on or within the object borders 320, 330 to various effects, such as distinguishing objects from each other or associating objects with other data on the display.

The interface 300 may also use visual markers or icons 380 that appear in the video playback region and which correspond to a point in time 385 on the timeline 340. These may be associated with detected events such as object crossings, enter or exit events, object appearances or disappearances, object removals or deposits in the scene, object abandonment, or merging or splitting of objects (such as a person taking or dropping a briefcase), lighting changes, or other detection events such as infra-red or motion or tripwire events from external sensors. Such events may additionally or instead be listed in a scrollable list, possibly described by words instead of graphical markers. In the present example, the marker **380** and point in time **385** may corresponding with the moment at which the suitcase or bag **322** ceased moving, particularly upon abandonment by the person **332**. The detection of that cessation of movement and continued lack of movement for a pre-determined duration may then trigger an abandoned object alarm in the particular implementation.

In order to display some of these data, data fields are provided within the emulated protocol which allow the transmission of such data. If the protocol cannot transmit some of these data, then fewer parts of this user interface will be available. For instance, if object appearance and disappearance events cannot be transmitted, then the object lifetime indicators **350, 360, 370** might not be visible. Similarly, if triggered events can't be represented within the protocol, then visual markers **380, 385** might not be shown. The protocol being discussed, therefore, must have at least one of these features in addition to carrying a video stream:

(i) Transmit event notifications, including at least one of object entrance, exit, deposit, removal, abandonment, merge, split, occlusion, or scene lighting change, camera motion, external sensor event;

(ii) Transmit object data, including at least one of object position, object age, object time-in-scene, object trajectory, object stillness, object shape, object silhouette, object bounding box, object centroid, object relationship to other objects in the same or previous frames, object signature (including colour, size or frequency information);

(iii) Control at least one of pan, tilt or zoom of the camera;

(iv) Control at least one of a sensor, light, infra-red lamp or detector attached to the camera;

(v) Control camera settings of at least one of frame rate, motion detection threshold values, event notification settings, regions of interest, remembered pan-tilt-or-zoom tuples.

The video display region **310** or timeline **340** or various buttons within the user interface **300** may allow certain camera controls to be activated. For example, selecting the video, by mouse clicking within the region **310** might ordinarily cause a camera to zoom into the scene, by transmitting a protocol message which indicates 'Zoom In'. This would be the case where the camera directly couples to the client (eg. the camera **160** of Fig. 1). However, where the processing unit **130** is interposed between the camera and client (ie. for the camera **110** of Fig. 1), the processing unit **130** receives this command and acts upon the command by modifying its own operational parameters, for example as described above. This is achieved within the processing unit **130** by a mapping of the camera controls available to the user via the user interface of the viewer 190 to a similar but distinct purpose within the processing unit **130**. In this fashion the user may manipulate the viewer **190** in a traditional fashion, whilst the processing unit **130** receives those traditional commands and interprets them differently to add capability to the system **100**, in this example abandoned object detection. Alternatively, those controlling commands might not be available for user instigation, but rather may be computer controlled from within the storage unit **150**, in which case the effect is similar except the user does not personally control the behaviour of the system **100**, but merely views the results of automatic decisions. A consequence of these modes of operation is that the processing unit 130 becomes, what the present inventors have termed, a "virtual" or "emulated" camera, in the sense that additional functionality is afforded over the traditional camera function of streaming images of a scene arising from a "real" camera instance, such as the camera **160**. Such functionality is achieved, at least in part, through the use of an emulated protocol that appears traditional to the client (viewer **190**, storage unit **150**) but which is interpreted differently by the processing unit **130**. Further, by selectively enabling and disabling the operation of the processing unit **130**, the camera **110** in the example of Fig. 1 may be returned to its traditional role and function within the system **100**.

Many user interface options exist in such a representation of object detection results. Those options include:

(a) Boxes can be drawn around objects of particular interest. For example, those objects which move near or cross into a particular part of the scene can be marked for the attention of the user, and optionally an event marker can be used. Regions of interest may be selected interactively by the user through the use of pan and tilt controls in the emulated protocol, which could expand or shrink or move a visual display of a region indicator (eg. **380**) of the processing unit **130** in order to establish or modify its shape or position.

(b) Objects can be detected as abandoned. There may be many criteria upon which this decision is made, including the duration of stillness of that object within the scene, the fact that it split from another object, the fact the other object has now been absent from the scene for longer than a certain time, and so on. Once the processing unit **130** decides to ascribe to an object a status of 'abandoned', that fact can be indicated in several ways, including placing a special border around it or pointing to the object or objects which were last in contact with it or changing its border colour gradually or dramatically or changing the playback such that earlier appearances of the object in relation to other objects are shown.

(c) Events can be listed on the timeline or elsewhere, and may be responsive to user interaction. For example, clicking on an event marker icon could cause the video playback to jump to that point in time or just before or after that point in time.

(d) The icons or object borders shown within the video display may be responsive to user interaction. For example, the user could double-click the mouse pointer **503** on a border **330** of a content object **332** and the video timeline could rewind to the first observance **365** in a frame of that object in the scene, or just before or after it.

(e) Areas around the objects of interest could be dimmed or their colours changed via image processing techniques well-known in the art, both to draw attention to the objects of interest, but also to draw attention away from background scenery which may be less important. Again, this may be automatic or as the result of user interaction with the object borders via pan, tilt or zoom operations.

(f) Object lifetimes within the scene can be indicated visually on the timeline and optionally using numeric descriptions. Clicking on an object timeline could replay its behaviour from that time or from its entrance in the scene.

(g) Object motion trails or time-lapse views of objects can be shown on the transformed video display. These may use dots or translucency effects to reveal how objects are moving through time and space.

(h) Persons or object occurrences can be counted by the processing unit and displayed on the transformed video stream. For example, in the context of a shopping centre surveillance system, it may be useful to count the number of shoppers walking through or pausing near a particular avenue. In this context, a request to zoom in to the scene could cause the processing unit **130** to transform the video stream to show additional more detailed information about the scene and the object counts.

(i) When a new event is detected by the processing unit **130**, such as if an object is left stationary longer than a predetermined duration, new event information may be transmitted to the storage unit **150**. This new event information may contain retroactive events from earlier times, for example, the first appearance within the scene of that object, or of the person carrying that object, may be known to the processing unit **130** and conveyed via the emulated protocol to the storage unit **150**. This presupposes that the emulated protocol has a data field which is capable of conveying such retroactive event information. The viewer unit **190** can display such events by adding new visual markers **385, 360, 365** to its timeline.

(j) The timeline displayed in the user interface **300** may instead be displayed within the transformed video stream, and interacted with entirely using pan, tilt and zoom commands generated by the client via simulation from a pointing device. Thumbnail views of objects may be extracted from the video and shown on such a timeline to indicate which object timelines are associated with which objects or people in the scene.

(k) Panning left or right could modify the playback speed of the transformed video stream. This presupposes that the processing unit **130** has its own internal video stream storage database which can be controlled via the emulated protocol.

The above examples show how performing object detection within a processing unit which analyses and modifies the video stream issuing from a camera can add new functionality to a surveillance system without requiring any additional upgrading of significant existing components of that system, thus providing a low-cost way to add advanced object detection facilities and cameras into the system.

Another aspect of operation of the system 100 is shown with respect to a user interface **400** of Fig. 4, which may also be rendered upon the display **514**. The user interface 400 contains a region 410 that shows icon representations 420 of each of the "cameras" available to the user of the viewer 190. Each of these represents a device with which the user can communicate using the network camera protocol 140. Each "camera" can be either a physical camera 110, 160, or a processing unit 130 that supports and thus is able to interpret the network camera protocol, thus forming a "virtual" camera, as discussed above. A list of available cameras is typically preconfigured by a user or system administrator. In the preferred implementation, each such camera is identified using an IP address or host name, and a TCP port.

The user interface 400 also contains a region 430 that displays video streams from any of the represented cameras 420. The user can interact with the camera icon region 410 to specify the video streams of certain cameras to display in the video region 430. In one implementation, the user may achieve this by dragging a camera icon 420 to the video region 430. Additionally, the user can change the position or displayed size of any of the corresponding video displays 440, 450, 460, 470 in the video region 430.

Because each physical camera 110, 160, and each processing unit 130 is identified by an IP address or host name and TCP port, the user interface can simultaneously display a direct video stream 440 and 470 from a physical camera 110 and 160 respectively, and video streams 450, 460 from multiple ones or instances of the processing units 130 that receive video from the same physical camera 110. Alternatively, the user interface could display a video stream from any number of different physical camera 160 or different processing units.

In the example shown in Fig. 4, and by relating the same to Fig. 1, "Camera D" may represent a traditional surveillance camera 160 which couples directly to the client for the provision of a simple video stream. In this case, the camera 160 is imaging a car park in which two vehicles are seen from above (plan view), as represented in the stream 470 seen in Fig. 4. However, according to the present disclosure, the video region 430 is also configured to show a video stream 440 from one physical camera 110 of Fig. 1, and two video streams 450, 460 from different processing units 130 each associated with the same source video obtained from the physical camera 110. The first video stream 440 shows a scene captured by the physical camera 110, designated as Camera A in Fig. 4. The scene contains a person 4421 and a suitcase 441. The second video stream 450 shows the output of a first processing unit 130, operating as "virtual" Camera B, that is configured to receive a video stream from the same physical camera 110, and is additionally configured to transform that video stream by overlaying object outlines 451 over each of the detected objects in the scene 441, 442, as detected by the object detection module 230. The third video stream 460 shows the output of a second processing unit 130, operating as "virtual" Camera C in Fig. 4, that is again configured to receive a video stream from the same physical camera 110, but in this case is configured to transform the video stream by drawing an indicator 461 around abandoned objects, as determined by the object processing module 240. This example therefore depicts multiple virtual camera instances.

It will be appreciated from Fig. 4 that instances of the processing unit 130 provide for the addition of capability to the system 100. Further, where the processing unit 130 is implemented in software, the number of instances of such may be varied from zero to any practical number limited by the alternate processes available. Where the processing unit is implemented in specialised hardware, such hardware may be configured to implement multiple instances of applications replicating the processing unit.

Industrial Applicability

The arrangements described are applicable to networked video surveillance systems and particularly where additional functionality is desired to be added to an existing system with reduced expense and increased convenience.

The foregoing describes only some embodiments of the present invention, and modifications and/or changes can be made thereto without departing from the scope and spirit of the invention, the embodiments being illustrative and not restrictive. For example, whilst the functionality of the processing unit **130** is illustrated and described separate from the camera **110**, such may be incorporated into the camera **110** by means of software or hardware implementation. In this fashion system upgrade may be facilitate merely by supply of new, perhaps additional cameras, without a need to alter the client or any server application. Incorporation of the processing unit 130 as a software component (eg. a plug-in) of the client or server application may also be desirable in some implementations.

(Australia Only) In the context of this specification, the word “comprising” means “including principally but not necessarily solely” or “having” or “including”, and not “consisting only of”. Variations of the word "comprising", such as “comprise” and “comprises” have correspondingly varied meanings.

The claims defining the invention are as follows:

1. A networked surveillance system comprising:

at least one networked camera imaging a scene;

a primary application configured for communication with said camera via a network according to a primary protocol for viewing and recording data associated with the imaged scene; and

at least one instance of a secondary application, each said instance being interposed in a communication path of said network between the camera and the primary application such that the primary and secondary applications communicate using the primary protocol, and the secondary applications and the camera communicate using a secondary protocol;

the secondary applications each interpreting data associated with the imaged scene and to transform the data in accordance with content identified from the data of the imaged scene, the transformed data being communicated with the primary application via the primary protocol.

2. A system according to claim 1 wherein the secondary protocol corresponds to the primary protocol.

3. A system according to claim 1 or 2 wherein said primary application is a client application and, with said camera, provides for at least one of recording and display of the scene according to a real instance of the camera, and said client application and each instance of the secondary application provide for at least one of recording and display of the scene transformed according to a corresponding virtual instance of the camera.

4. A system according to claim 3 wherein the client application generates commands according to the primary protocol which are:

(i) interpretable by the camera according to the real instance to control operating parameters of the camera, and

(ii) interpretable by an instance of the secondary application to control transformation of the data according to a virtual instance of the camera

5. A system for emulating a network video camera, said system comprising a processing unit, a client, a video camera and a communications network, in which the processing unit is configured to:

(a) obtain a video stream from the video camera using a first network video camera communication protocol;

(b) interpret commands from the client sent using a second network video camera communication protocol, and to use the commands to adjust at least one of a set of processing parameters;

(c) perform object based processing of the content of the video stream according to the processing parameters, said processing including detecting events;

(d) construct a second video stream based on a result of the processing; and

(e) use the second network video camera communication protocol to transmit the second video stream to the client via the network, including data of the detected events, such that the processing unit emulates the second network video camera protocol.

6. A system according to claim 5 wherein the first and second network camera communication protocols are identical.

7. A system according to claim 5 wherein the second network camera communication protocol is a subset of the first network camera communication protocol.

8. A system according to claim 5 wherein the commands from the client are interpreted differently by the processing unit than from how a camera which natively uses the second protocol would interpret them.

9. A processing unit for a networked video surveillance system, said processing unit comprising:

a first connection by which said processing unit couples to a client arrangement of said system via a communications network according to a first communications protocol;

a second network connection by which said processing unit couples to a camera of said system via said communications network according to a second communication protocol;

an enablement receiver by which processing by said processing unit is selectively enabled and disabled,

such that when disabled, commands received from the client arrangement according to the first protocol are transferred to the camera according to the second protocol to modify operating parameters of the camera to adjust capture of a scene by the camera from which a video data stream of the scene is conveyed to the client arrangement via said protocols, and

when enabled, commands received from the client arrangement are mapped to processing commands for said processing to transform the video data stream by means of scene content processing and to convey the transformed video data stream to the client arrangement via the first protocol.

10. A processing unit according to claim 9 wherein the processing unit is formed as an application executable by a computer apparatus such that when executed, said processing unit is enabled and is interposed in said communications network between the camera and the client arrangement.

11. A processing unit according to claim 9 wherein the processing unit is formed of hardware.

12. A network video camera for a surveillance system comprising a processing unit according to claim 9, 10 or 11.

13. A server forming part of surveillance network, said server being coupled in a communication network between at least one camera and a client, said server comprising a processing unit according to claim 9, 10 or 11.

14. A client application for using in a networked surveillance system, said client application being executable upon a computer apparatus to reproduce surveillance video streams obtained from one or more networked surveillance cameras coupled to a communications network to which said computer apparatus is connectable, said client application comprising a processing unit according to claim 10.

15. A networked surveillance system substantially as described herein with reference to any one of the embodiments as that embodiment is illustrated in the drawings.

16. A processing unit substantially as described herein with reference to any one of the embodiments as that embodiment is illustrated in the drawings.

17. A client device including a processing unit according to claim 16.

18. A method of adding capability to a surveillance system comprising at least one client unit coupled to at least one video camera over a communications network according to a protocol, said method comprising the step of interposing a processing unit according to claim 9, 10 or 11 into said system between the at least one camera and the at least one client unit.

Dated this 28th day of February 2008

CANON KABUSHIKI KAISHA

Patent Attorneys for the Applicant

Spruson&Ferguson

2008200965 28 Feb 2008

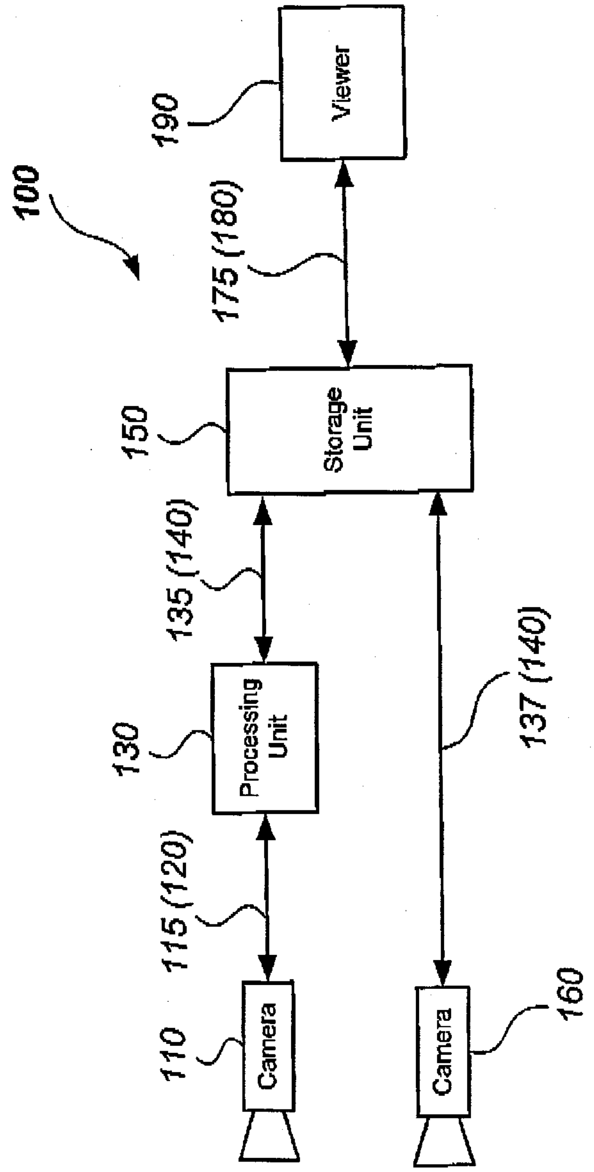
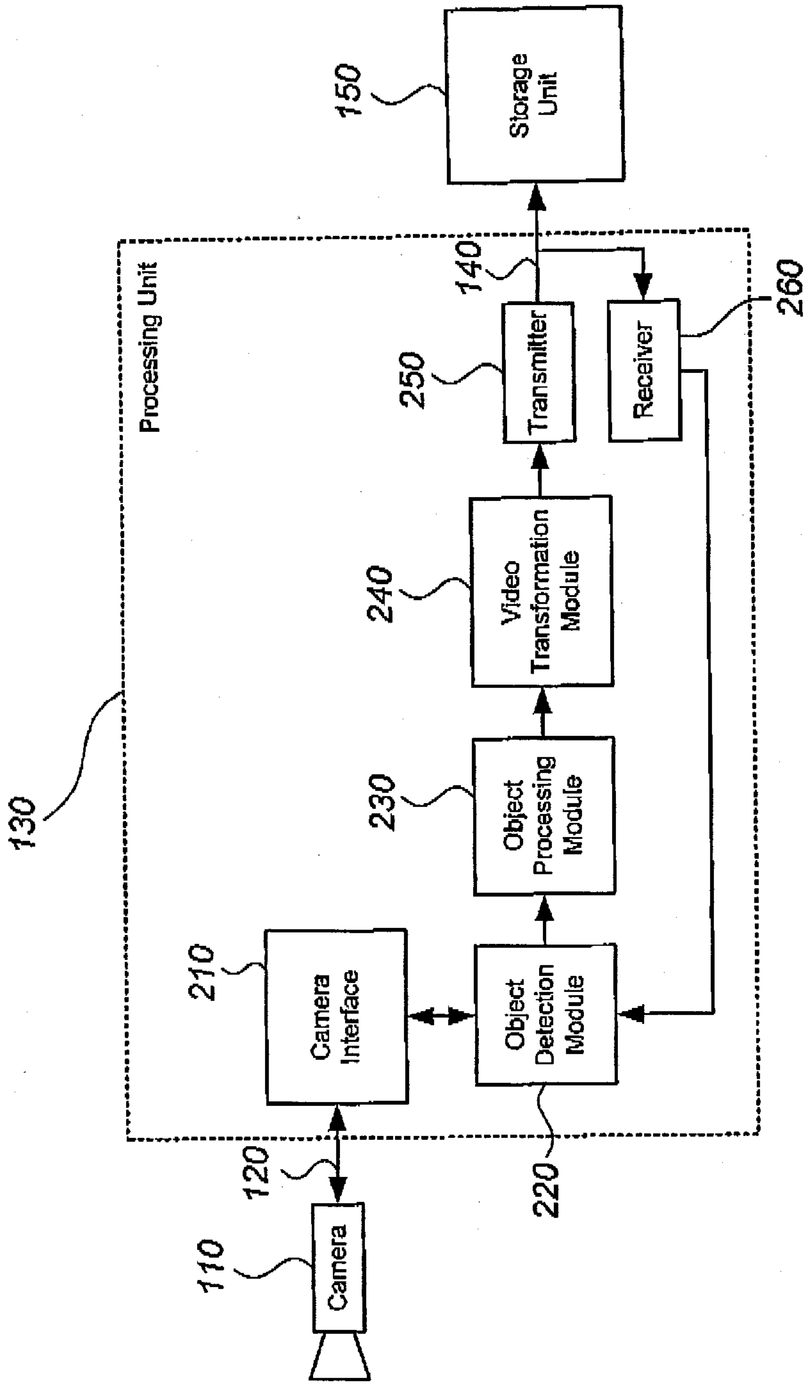


Fig. 1



Storage Unit
150
Processing Unit
140
260
Transmitter
250
Receiver
Video Transformation Module
240
Object Processing Module
230
130
Camera Interface
210
Object Detection Module
220
120
Camera
110

Fig. 2

2008200965 28 Feb 2008

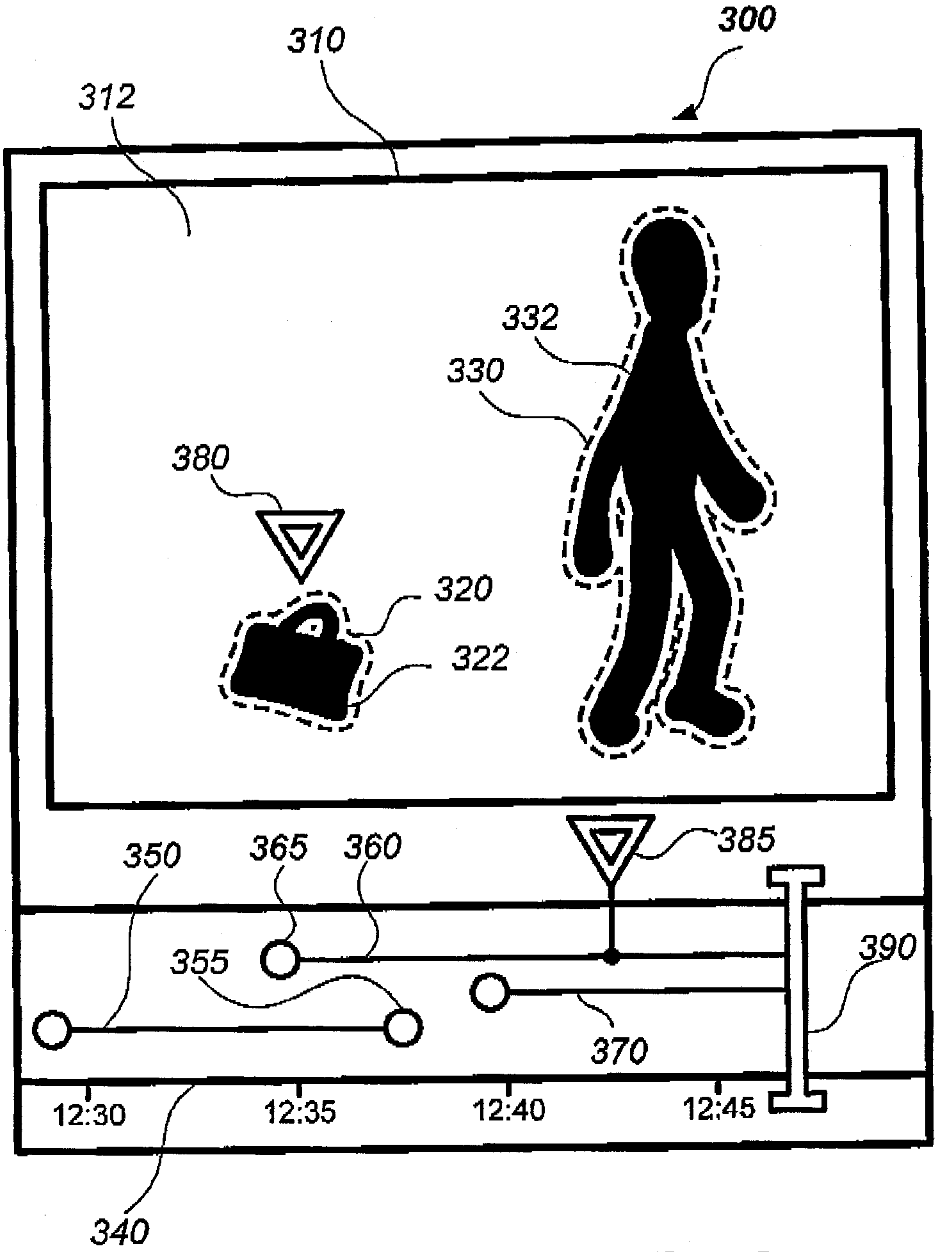


Fig. 3

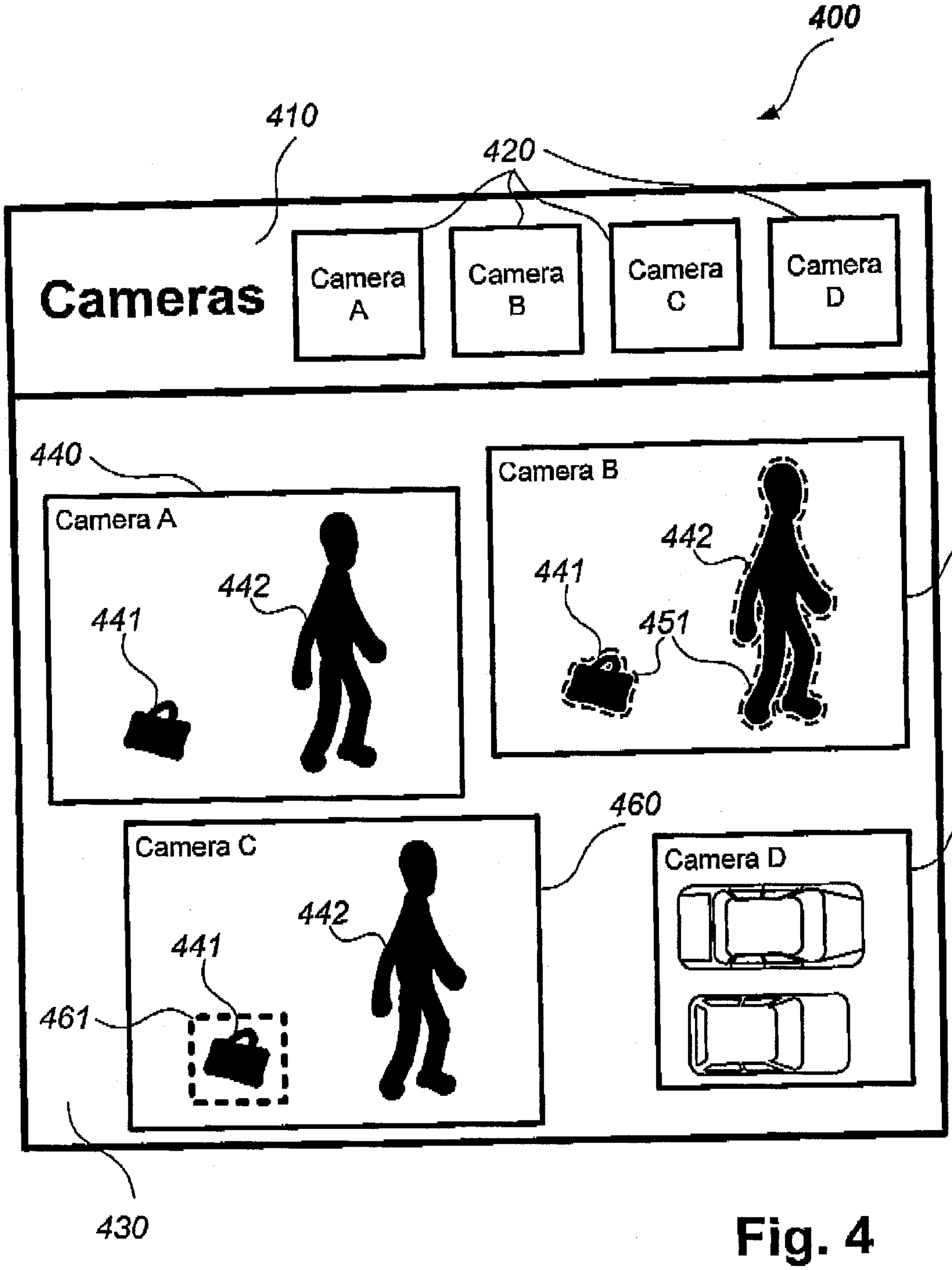
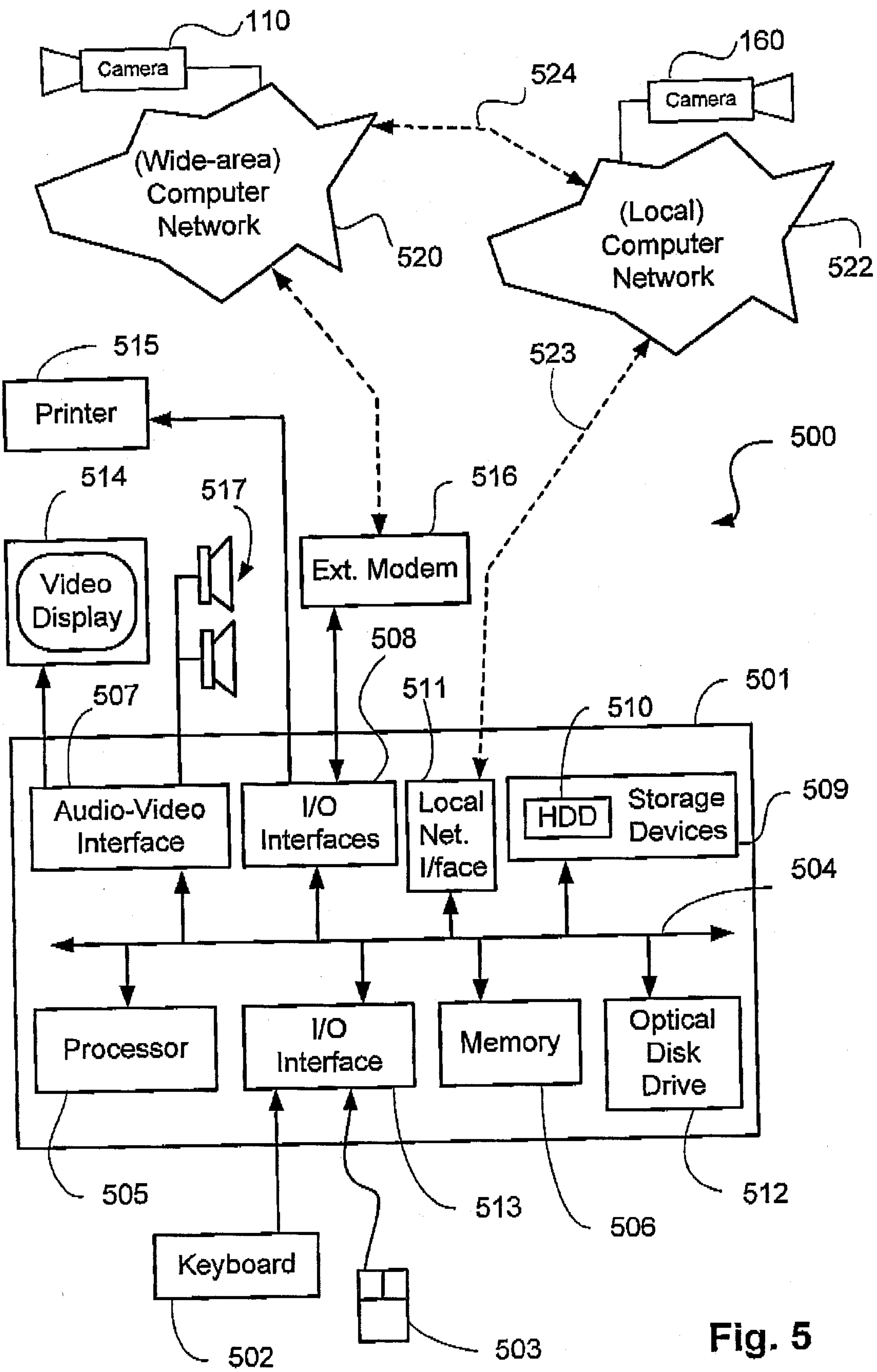

400
410
420
Cameras
Camera A
Camera B
Camera C
Camera D
440
Camera A
441
442
Camera B
442
441
451
450
460
470
Camera C
442
441
461
Camera D
430

Fig. 4



110
Camera
160
Camera
524
(Wide-area) Computer Network
520
(Local) Computer Network
522
515
Printer
523
500
514
517
516
Video Display
Ext. Modem
508
507
511
510
501
Audio-Video Interface
I/O Interfaces
Local Net. I/face
HDD
Storage Devices
509
504
Processor
I/O Interface
Memory
Optical Disk Drive
505
513
506
512
Keyboard
502
503

Fig. 5