

VIDEO ANALYZER

5

FIELD OF THE INVENTION

The invention generally relates to video analysis.

BACKGROUND

In general, analog camera surveillance systems produce a time division multiplexed output that is stored to tape. The cameras are coupled to a
10 multiplexor that switches at predetermined intervals between the cameras. Such systems have a first camera that outputs an image at T0 and a second camera that outputs an image at T1 etc. As such, the output tape contains images from each of the cameras in sequence. The time division multiplexing allows large amounts of information to be stored. However, if an event occurs that requires the video
15 tape to be reviewed, finding a particular incident and demultiplexing the tape becomes a difficult and time consuming task.

Even if digital cameras are used and a digital output is produced, the multiplexed digital stream requires a great deal of human intervention in order to identify an event. Thus there is a need for an automated system for
20 demultiplexing a video stream.

SUMMARY OF THE INVENTION

In accordance with one aspect of the invention, a multiplexed motion picture stream (e.g., video) is demultiplexed in real time using image recognition
25 techniques. In some embodiments, stored motion picture images of different channels can be viewed simultaneously at the same image by advancing to a given image of only one stored channel. The motion picture stream contains one or more images from each of a plurality of camera sources.

Illustrative embodiments of the invention are implemented as a computer program product having a computer usable medium with computer readable program code thereon. The computer readable code may be read and utilized by a computer system in accordance with conventional processes.

5 The method includes receiving at least a portion of the video stream composed of a plurality of video images in a processor. A first video image and a second video image are parsed from the video stream. An amount of mismatch is determined between the first and second video image. Determining an amount of mismatch can be accomplished using any one of a number of comparison
10 techniques. Pixel by pixel comparisons may be made. Region by region comparisons can be made in the alternative. If the mismatch is below a mismatch threshold, the first and second video images are assigned to a first channel. The first channel may be a memory storage location that is associated with a first camera. If the mismatch is above a mismatch threshold, the first video image is
15 assigned to the first channel and the second video image is assigned to a second channel. The second channel is a second memory storage location that is associated with a second camera.

 The process continues wherein a current image is parsed from the video stream and compared to each of the images that are assigned to channels. After a
20 number of images, each camera within the system has an image associated with it. The latest image that is associated with a channel (camera) is the reference image. The reference image is used for all subsequent comparisons with the current image in the image stream. After all of the channels have assigned images mismatch levels are determined for each channel. If any of the mismatch
25 values are below a mismatch threshold, the current image is assigned to a channel having the lowest mismatch. If all of the mismatch values are above the mismatch threshold, then the mismatch values are compared to a discard threshold and if all of the mismatch values are above the discard threshold, then

the current image is discarded. When the term discarded is used herein, the term means that the discarded images are not associated with a camera/channel, however the images may be stored and kept for later review. Images are only discarded once all of the channels have a representative image. If all of the
5 channels/cameras do not have an associated video image, then the discard threshold is not used. The system can automatically determine both the number of cameras and also when to begin using the discard threshold. The system will obtain reference images for the cameras and when a particular camera (for example camera 1) has been assigned a predetermined number of images, the
10 system will recognize that all of the cameras have been accounted for and that the discard threshold may then be used.

The methodology happens within a computer system within a processor and the processor can operate on a video stream automatically without user intervention.

15 In certain embodiments, a user can interact with a graphical user interface to change various parameters. For example, the user may decide to change the mismatch threshold or change the discard threshold. A user can also cause all of the reference images to be discarded and for the system to begin again acquiring images from the video stream and assigning the images to each of the channels.
20 A user may also assign any image to any channel, even though the image is above a mismatch threshold for the channel. Either during the demultiplexing process or after the demultiplexing process, a user can recall images from a particular channel and can both proceed forward and in reverse through the sequence of images. At the same time all of the other camera images can be
25 simultaneously displayed and sequenced through by just moving through the images associated with a single camera.

The methodology may operate in real-time parsing images from a stream either from memory or that is received from the multiplexor. To assist in real-

time comparisons, the reference images and the current image may be sub-sampled. In other embodiments, a sub-section of the video display may be selected and then the data within the sub-section of the reference images and the current image is compared. In such a fashion, less data is used for comparison
5 than the entire video image.

BRIEF DESCRIPTION OF THE DRAWINGS

The features of the invention will be more readily understood by reference to the following detailed description, taken with reference to the accompanying
10 drawings, in which:

Fig. 1 shows one environment for implementing a first embodiment for demultiplexing video images;

Fig. 1A is a flow diagram illustrating demultiplexing video images from a plurality of cameras in a stream of video images;

15 Fig. 1B is a flow diagram illustrating further aspects of demultiplexing video images wherein the images are sub-sampled;

Fig. 1C is a flow diagram illustrating further aspects of demultiplexing video images wherein sub-sections of the images are compared;

Fig. 1D is a screen shot showing a graphical display of views from multiple
20 cameras;

Fig. 2A represents a reference image as displayed on a display device and the lined box indicates the user defined area;

Fig. 2B represents the current image as displayed on a display device and shows that the same user defined area as that in Fig. 2A is used for comparison;

25 Fig. 3 shows three cameras (camera 1, camera 2, and camera 3) producing a sequence of multiplexed video images in which an image from each camera is added to the sequence over time;

Fig. 4 shows a user selected region of interest in which the region contains a displayed date and time; and

Fig. 5 is an embodiment of the invention in which modules are constructed in hardware to perform the functions of the disclosed method.

5

Detailed Description of Specific Embodiments

As used in the following text, the term "digital video" implies a digital representation of a sequence of images which may be temporally displayed on a display device. Typically, digital video contains a plurality of frames wherein each frame represents a separate image. The frames may be further subdivided such that the frames are made up of a series of pixels. As used in the following description the term "pixel" shall mean a single point of an image. The greater the number of pixels that are contained in an image, the greater the resolution of the video. Resolutions are conventionally referenced by length and width measurements of the number of pixels, for example, in a resolution of 800x600, there are 800 pixels along the length of an image by 600 pixels along the width of the image.

A "channel" as defined herein is either a physical or virtual separation of information that allows the information to be stored and retrieved from memory. For example, a channel can be a designated storage location in memory that is associated with a particular camera. Similarly, a channel may simply be the association of information with a delineator, such as a camera name or number.

In an embodiment of the present invention, video images in a stream of images formed from a plurality of video sources are demultiplexed so that the images from a given source can be collected, analyzed and viewed.

Fig. 1 shows one environment for implementing an embodiment of the invention. In Fig. 1 a plurality of cameras (Camera A, Camera B, and Camera C)

are shown. Each camera produces a video signal containing video information which is multiplexed together and stored either in a digital format in memory associated with a processor 20 or in digital or analog format on a tape 25. The multiplexed information may contain header information to identify the source, but the present invention does not need header information to demultiplex the information. In fact, if header information exists, the present invention as described will either remove the header information or skip the header information and only process information that forms an image. The multiplexed information is subsequently demultiplexed by the processor. If the video information is in an analog format, the information is first digitized in an analog to digital conversion process which is well known to those of ordinary skill in the art. After the video information is digitized, the sequence of digital data is transferred to a processor. The images may be compressed or uncompressed or in field or frame format. If the information is compressed the processor will decompress the images prior to demultiplexing the video image stream. Similarly, the digital video may be in any one of a number of color space representations including RGB and YUV. The processor may demultiplex the information in real-time, such that as the information is captured from the cameras and multiplexed together, the information can be streamed to the processor and demultiplexed. Similarly, the stream of information can first be stored in memory and then provided to the processor.

The processor performs the operation as described with respect to the flow charts of Figs. 1A for demultiplexing the video image stream. The video image stream can be demultiplexed without user intervention and under total control of the processor. In certain embodiments, a user may change parameters or reset the system or manually select images to be associated with a separate channel. However, even in this type of an embodiment, the system can automatically process the image stream.

It should be understood that within this disclosure the term frame will be used to refer to a video image which is to be displayed for a designated time period. The disclosure is equally applicable to video fields which are normally produced for display devices displaying interlaced images.

5 As shown in Figs. 1A, a first video image is identified in a video stream 110. As previously stated the video stream may come directly from a plurality of cameras after the images from the cameras have been multiplexed or the video stream may come from memory that is brought into the processor. The first video image in the video stream is parsed and stored in a memory location
10 associated with the processor. The processor associates the first video image with a first camera and the image is marked as a reference image for the first camera 120. As a reference image for a camera, the image is used for comparisons with images in the video stream as discussed below.

 The next image ("current image") is identified in the video stream 130.
15 The image data is parsed and stored to memory associated with the processor. The processor then recalls all of the reference images and the current image and compares the image data 140. During this comparison a mismatch value is calculated. The mismatch value can be the percentage of error between pixels of a reference image and the current image. The mismatch value can be calculated
20 in any number of ways so long as it is a comparison of the amount of shared information between the reference image and the current image. For example, difference ratios can be calculated between the current image and each reference image of an identified source (camera). The sum of the absolute value of the difference between the current image's pixels and the reference image's pixels is
25 calculated and then divided by the sum of the intensity values of the current image's pixels. This ratio represents the difference between the current image and a given camera reference image.

A separate mismatch value is calculated for each reference image and the current image. In the flow chart, at this stage, there is only a single reference image, since only one camera has been identified, but as more images are identified as coming from different reference cameras, a reference frame will be associated with each camera.

The processor then automatically identifies if any of the determined mismatch values are below a mismatch threshold 145. Although the term below is used, the term equal to or below may be substituted without deviating from the scope of the invention. Further, the comparison process assumes that the mismatch values are provided on a scale in which a lower value indicates that the representative image and the current image share more information in common than if the mismatch value was higher. It should be understood that a reverse scale could be implemented wherein the greater the mismatch value, the more information that is shared between the current image and the representative image. In such a situation all of the comparisons would be reversed.

If the mismatch values are below a mismatch threshold, the current image is associated with the camera (reference image) having the lowest mismatch value 157. The current image is then made the reference image for the camera and replaces the previous reference image 160. If none of the mismatch values are below the mismatch threshold then, the processor compares the mismatch values to a discard threshold 147. The discard threshold is only used for comparison purposes once an image has been associated with all of the channels/cameras.

If all of the mismatch values are not above a discard threshold, the processor assigns the current video image to a new camera 150. For example, the current image would be associated with camera number two as the first reference image was associated with camera number one. In the process of assigning the

current video image to a new camera, the current image may be provided with a header which indicates that the current image is from the second camera. The processor may also store the current image to a memory location associated with the second image. The memory location may be shared memory with the other
5 cameras or may be separate memory which is exclusively used for camera two. If the memory is shared memory, the processor will construct a (look-up-table) LUT for associating the image data with a particular camera so that the images for the camera can be retrieved and displayed. If all of the mismatch values are above the discard threshold, then the system can automatically discard the
10 current video image 155. The image may be stored and saved to a separate memory location for further review. In other embodiments, the image is discarded and is not stored.

If the current video image is assigned to a new camera, the processor identifies the current image as the reference image for the new camera 160.

15 The processor then checks to see if there are any more images and if there are none, the process ends 165. If there are additional images, a new image is retrieved and the process continues. As the process continues, more current images are assigned to new cameras, until all of the cameras in the system have an associated reference image. The processor can either be programmed with the
20 total number of camera or the processor can automatically determine the total number of cameras. In the surveillance industry, multiplexed video is multiplexed such that an image or multiple images from each camera are stored in a sequential manner (camera 1, camera 2, etc.) within the video stream. For example if there are three cameras, one image from each camera may be stored in
25 the stream, such that the first image is taken at T0 and the second image is at T1 and the third at T2. In other embodiments, certain cameras may add more than one image to the video stream before other cameras add images. For example, camera 1 may provide 3 images taken at T0, T1 and T2 to the video stream and

then camera 2 provides an image at T3 and camera 3 at T4. As a result, after the processor processes the first few images using the method as described above for demultiplexing the video stream, all of the cameras are normally identified. If a camera, such as the first camera has a predetermined number of images

5 associated with it, for example, 10 images, the processor can then count the total number of cameras having assigned images. Since multiple images have been associated with a camera (the first camera), it is very likely that images for all of the cameras have been processed and that the multiplexed video stream has looped around and back to the first camera.

10 In order to increase the speed of processing the video stream, the images may be sub-sampled. The images may be sub-sampled in both the horizontal and vertical directions prior to the images being compared. In one embodiment, the image which may be an NTSC frame is reduced in size, for example to 1/8 of the original size prior to storage. Reduction of the image size may be simply
15 achieved by removing pixels. For example, to reduce the number of pixels by a factor of 4 for an 800 x 600 pixel image, every other pixel could be removed such that there were 400 x 300 pixels. In a further example, the image may be transform coded, using a Fourier or other transform and then sub-sampled in the frequency domain. In another embodiment, the data may be reduced through
20 averaging. For example, every eight rows of eight pixels could be averaged such that 64 pixels are reduced to one pixel.

This sub-sampling is shown in Fig. 1B. Both the reference image and the current image are sub-sampled prior to comparison 125B, 135B. The video images may be stored in their entirety with all of their pixel information for
25 display purposes, but the images are sub-sampled to increase the speed of the comparison process. Further Fig. 1C shows that a section of the image data may be used for the comparison process rather than use all of the displayable image data 125C, 135C. A user of the system, can pre-select or the system may pre-

select a section of the image data to compare. When the term section is used in this application, the term implies that the image data is displayed on a display device and a block or other shape is selected such that less than all of the image data is selected. If the section is predetermined, the image data does not need to
5 be displayed on a display device. The image data will be selected based on the address location that is associated with the section of the display device that the user desires to have compared.

It should be understood by one of ordinary skill in the art that both sub-sampling and also selection of a sub-section of an image may be used in
10 combination to further increase the speed for comparing images and demultiplexing the image stream.

In certain embodiments, prior to comparison of the reference images and current image, the selected area of the images are normalized so as to remove any change in brightness/intensity from the comparison. For example, the
15 average intensity value for the pixels in the user selected area in the reference image and the same area in the current image are calculated. The average intensity value is subtracted from each of the pixel intensity values. This step normalizes the values accounting for any changes in brightness, such as sudden flashes of light. Thus, only the absolute value of the variation about the median
20 of the two images is compared. For example, if a camera is monitoring an automated teller machine and a car approaches the machine in which the headlights of the car suddenly illuminate the viewing area, this lighting change will be accounted for. This normalization may also be performed in any one of a number of ways known in the art, including using the RMS value as opposed to
25 the average intensity for the user selected area.

During the course of demultiplexing the images, the demultiplexing process can be displayed on a display device 800. The images that are associated with a camera are displayed in separate windows 810. The reference image is

displayed in the window for each of the cameras as shown in Fig. 1D. By continually displaying and resaving the reference image for each camera, temporal synchronization between each of the various cameras is maintained. For example for each cycle through a four camera time division surveillance system where camera one captures three images before camera two takes an image, and both cameras three and four each take one image each, each channel associated with a camera will have six images saved. Camera one will have the images displayed at T0, T1, and T2 followed by the T2 image repeated three additional times. Camera 2 will have three blank images and then its image at T3 which is repeated two additional times. Camera three would have four blank images stored followed by the image at T4 which is repeated twice and camera four would have five blank images followed by the image taken at T5. This accounts for the first cycle through the cameras. On subsequent cycles, the reference frame would be repeated until a new current frame is associated with the camera and the reference frame is updated. As such, images would appear for all of the cameras on the display device.

Since each channel for a camera stores an equal number of images, the images can be viewed in a synchronized fashion. As such, a user of the system either during the demultiplexing process or after all of the images have been demultiplexed, can access a point in time and then view all of the images captured by all of the cameras during that time period. For, example a user of the system can select to rewind or forward through a single set of images for a particular camera, the system can then display the images for all of the cameras that occur at the same time as that for the selected camera.

The system and method as defined above will operate to demultiplex a video stream without user intervention. However, a user may override the system and change settings within the system. For example, a user may change the disregard threshold 820 or the mismatch threshold 830 as shown in Fig. 1D.

A user may assign an image to a camera even though the system indicates that the image should be discarded or assigned to a different camera. As such, the system and method provides the user with a graphical user interface which provides controls for changing the various parameters of the system and
5 overriding the system.

The process of replacing the reference image with the current image assists in source identification, since the effect of changes which gradually occur, such as lighting changes between day and night and camera panning are minimized.

10 In one embodiment of the invention cameras film an area by sweeping across the area. The images captured by the sweeping cameras which are multiplexed together are subsequently demultiplexed as shown in Fig.3. In this embodiment, when the mismatch threshold is set for associating an image with a camera, the temporal change from image to image as the camera sweeps is taken
15 into account. For example, Fig. 3 shows three cameras (camera 1, camera 2, and camera 3) producing a sequence of multiplexed video images 320 in which an image from each camera is added to the sequence over time. After an image from all three cameras has been saved to the sequence, the process repeats. In Fig. 3A images from camera 1 occur at T0 and T3. As shown in Fig. 3B if the image from
20 T0 is shifted by three pixels which accounts for the movement of the camera between T0 and T3 and the images are overlaid the majority of the pixels from the image at T0 will overlap with the image at T3. By replacing the reference image with the current image and using the updated reference image during comparison, the amount of movement between the reference image and the
25 current image is minimized such that more elaborate tracking and comparison techniques need not be employed. If the movement of the camera is slow in comparison to the time between saved images, the threshold can be adjusted to accommodate movement.

In other embodiments, the pixels of the reference image are shifted in position prior to comparison with the current image to account for the sweep of the camera. Additionally, in the comparison process, the current image can be shifted several times in several directions for calculating a difference ratio and
5 comparing the ratio to the threshold. Further, the direction of movement can be anticipated through trend analysis wherein the direction of movement is based upon previous movement for the same source. Thus, by tracking the direction of movement, the number of shifts of the data locations and comparisons are reduced.

10 In yet another embodiment, after all of the cameras within the surveillance system have an associated image within the system, the processor has enough information to identify the order in which images are associated with the cameras. Once the order is known, either the system or the user can raise the mismatch threshold and assign a separate mismatch threshold for each of the
15 camera such that only significant changes in the images are identified. For example if there is movement or substantial changes in lighting, then the system can identify these images and assign a tag to the digital data so that they can be found by a user at a subsequent time when reviewing the images. In another embodiment, in order to conserve memory space, only the video images that do
20 have movement may be recorded and saved to the channel associated with the camera.

Under certain circumstances, the video images may be separated by time and/or date. In certain multiplexing systems in which cameras are digital cameras such as those that include CCD (charged-coupled devices), the digital
25 data stream containing the digital images may include a date and time stamp preceding the digital video image. In such a case, the images can be stored and separated out according to date and time. In other multiplexing systems, in which the cameras produce an analog signal with a superimposed date and time

or in which digital cameras include date and time within the digital image, the video images may be subdivided out by time and date. In this embodiment, after the video signal is digitized, the user selected region of interest is selected as the displayed time or date. If the video images are to be separated out by time, the user selected region could be selected to be the hour 400A as shown in Fig. 4A and any time there is a change in that area of the video image the data of the video image could be saved to a different memory location which is associated with that hour. Similarly in Fig. 4B a video display 410 of digital image data is shown in which the day of the month is the selected user-defined region 400B.

10 The system can then continuously look for changes in data within the selected user-defined area that are above a threshold in order to identify images that come from different days of the month.

Fig. 5 is another embodiment of the invention in which the invention is in a device form 500. The device 500 may be an electronic device such as a programmed FPGA module or an ASIC chip. The device 500 includes a retrieval module 501 which retrieves or receives digital image data 550. The digital image data includes images of video data from multiple sources. The retrieval module 501 retrieves a first set of digital video data representative of a first image and also retrieves a current set of digital video data representative of a current image.

15 The retrieval module 501 passes the first set of digital data to a storage module 502. The storage module 502 stores to memory 560 the first set of digital video data to a memory location associated with a first source 561. The data is also identified as the representative video data of the first source. The retrieval module 501 passes the first set of digital video data to a calculation module 503

20 along with the current set of digital video data.

The calculation module 503 calculates a difference ratio using the first set of digital video data and the current set of digital video data as explained above. If more than one source has already been identified, the digital video data of

each representative image for a source is retrieved and a difference ratio is calculated for each. The difference ratio(s) ("mismatch values") are forwarded to a query module 504. The query module 504 compares the difference ratio(s) to a mismatch threshold. If a difference ratio for a representative image of a camera
5 source is below the threshold then the current image is associated with that camera.

The digital video data is then stored in memory associated with the identified camera 562. For example if the mismatch value for the representative image of the 4th camera is below the mismatch threshold, the current digital video
10 data will be associated with that camera. Further, the current image will become the representative image of the 4th camera. The memory that is associated with a camera may be either shared memory, wherein an identifier is associated with the image data to identify the image data as being associated with the camera or the memory may be separate or segmented memory. If the memory is completely
15 separate the processor will send the associated image data to a particular physical piece of memory such as a RAM chip. If the memory is segmented, various memory addresses will be associated with a particular camera and the processor will have a LUT and will assign the digital image data to the memory segment that is associated with the camera.

20 If all of the difference ratios are above the mismatch threshold the query module 504 compares the mismatch values to a discard threshold. The system can then automatically discard the image data without a user's intervention if the mismatch values are all above the discard threshold. If the mismatch values are not above the discard threshold, then the query module will assign the image
25 data to a new camera source automatically. In other embodiments, a user receives a message on the display device 510 that the current image of video data cannot readily be associated with a particular source since the mismatch values are above a threshold. The user can then send a response through an input

device 511, such as a keyboard or a mouse which is connectively coupled to the device which indicates that the current image is either from the first source or other previously identified source, should be associated with a new source or should be discarded. The query module 504 receives this information and

5 instructs the storage module 502 to store the current digital video data for the current image of video in a memory location with the identified source. The user of the system is provided with a display of the current image, as well as, each representative image for each camera. This allows the user to make a visual comparison aiding in the identification process.

10 It should be understood that if the user identifies the current digital video data as coming from a new source the current digital video data will be associated with a memory location associated with a new camera source. A normalization module 505 may also be included in the device. The normalization module normalizes the digital video data of the current video image as

15 compared to the representative video images. As expressed above, normalization of the digital video data can account for changes in lighting such as the changes that occur from day to night. Normalization occurs prior to the calculation of the difference ratio.

It should be noted that the flow diagrams are used herein to demonstrate

20 various aspects of the invention, and should not be construed to limit the present invention to any particular logic flow or logic implementation. The described logic may be partitioned into different logic blocks (e.g., programs, modules, functions, or subroutines) without changing the overall results or otherwise departing from the true scope of the invention. Often times, logic elements may

25 be added, modified, omitted, performed in a different order, or implemented using different logic constructs (e.g., logic gates, looping primitives, conditional logic, and other logic constructs) without changing the overall results or otherwise departing from the true scope of the invention.

The present invention may be embodied in many different forms, including, but in no way limited to, computer program logic for use with a processor (*e.g.*, a microprocessor, microcontroller, digital signal processor, or general purpose computer), programmable logic for use with a programmable
5 logic device (*e.g.*, a Field Programmable Gate Array (FPGA) or other PLD), discrete components, integrated circuitry (*e.g.*, an Application Specific Integrated Circuit (ASIC)), or any other means including any combination thereof.

Computer program logic implementing all or part of the functionality previously described herein may be embodied in various forms, including, but in
10 no way limited to, a source code form, a computer executable form, and various intermediate forms (*e.g.*, forms generated by an assembler, compiler, linker, or locator.) Source code may include a series of computer program instructions implemented in any of various programming languages (*e.g.*, an object code, an assembly language, or a high-level language such as Fortran, C, C++, JAVA, or
15 HTML) for use with various operating systems or operating environments. The source code may define and use various data structures and communication messages. The source code may be in a computer executable form (*e.g.*, via an interpreter), or the source code may be converted (*e.g.*, via a translator, assembler, or compiler) into a computer executable form.

20 The computer program may be fixed in any form (*e.g.*, source code form, computer executable form, or an intermediate form) either permanently or transitorily in a tangible storage medium, such as a semiconductor memory device (*e.g.*, a RAM, ROM, PROM, EEPROM, or Flash-Programmable RAM), a magnetic memory device (*e.g.*, a diskette or fixed disk), an optical memory device
25 (*e.g.*, a CD-ROM), a PC card (*e.g.*, PCMCIA card), or other memory device. The computer program may be fixed in any form in a signal that is transmittable to a computer using any of various communication technologies, including, but in no way limited to, analog technologies, digital technologies, optical technologies,

wireless technologies, networking technologies, and internetworking technologies. The computer program may be distributed in any form as a removable storage medium with accompanying printed or electronic documentation (*e.g.*, shrink wrapped software or a magnetic tape), preloaded
5 with a computer system (*e.g.*, on system ROM or fixed disk), or distributed from a server or electronic bulletin board over the communication system (*e.g.*, the Internet or World Wide Web.)

Hardware logic (including programmable logic for use with a programmable logic device) implementing all or part of the functionality
10 previously described herein may be designed using traditional manual methods, or may be designed, captured, simulated, or documented electronically using various tools, such as Computer Aided Design (CAD), a hardware description language (*e.g.*, VHDL or AHDL), or a PLD programming language (*e.g.*, PALASM, ABEL, or CUPL.)

15 The present invention may be embodied in other specific forms without departing from the true scope of the invention. The described embodiments are to be considered in all respects only as illustrative and not restrictive.

5

10

15 We claim:

1. A method for demultiplexing video images in a video stream, the video images produced by a plurality of cameras, the method comprising:
 - receiving at least a portion of the video stream composed of a plurality of video images in a processor;
 - 20 parsing a first video image from the video stream;
 - parsing a second video image from the video stream;
 - determining an amount of mismatch between the first and second video image;
 - if the mismatch is below a mismatch threshold, assigning the first
 - 25 and second video images to a first channel.
2. The method according to claim 1, wherein the first channel includes a memory storage location for storing video images.

3. The method according to claim 1, wherein the first channel is associated with a first camera.
4. The method according to claim 1, wherein if the mismatch is above a mismatch threshold, assigning the first video image to a first channel and
5 the second video image to a second channel.
5. The method according to claim 4, wherein the second channel is associated with a second camera.
6. The method according to claim 4, wherein after each channel associated with a camera has been assigned a video image,
10 comparing a new video image from the video stream with at least one video image from each channel;
determining an amount of mismatch between the new video image and the at least one video image from each channel; and
assigning the new video image to a channel having the lowest
15 mismatch.
7. The method according to claim 6, wherein if the lowest mismatch is above a discard threshold, then discarding the new video image rather than assigning the new video image to a channel.
8. The method according to claim 4, further comprising:
20 comparing a third video image to the first and the second video images to determine a first and a second mismatch;
if the first and second mismatches are above a mismatch threshold
assigning the third video image to a third channel.
9. The method according to claim 4, further comprising:
25 comparing a third video image to the first and the second video images to determine a first and a second mismatch;

if the first mismatch and second mismatches are below a mismatch threshold, assigning the third video image to the channel associated with the lower mismatch.

10. The method according to claim 4, further comprising:

5 comparing a third video image to the first and the second video images to determine a first and second mismatch;

 if both the first and the second mismatches are above a discard threshold, discarding the third video image.

11. The method according to claim 4, wherein each channel is associated with a
10 camera.

12. The method according to claim 1, further comprising:

 providing the number of cameras to the processor.

13. A method for demultiplexing an image stream having a plurality of
15 images, the method comprising:

 comparing each new image within the image stream to a representative image associated with a camera to determine a mismatch value;

 if each of the mismatch values are above a mismatch threshold, associating the new image with a new camera.

20 14. A method according to claim 13, wherein

 if any of the mismatch values are below the mismatch threshold, selecting the lowest mismatch value and associating the new video image with the camera associated with the lowest mismatch value.

25 15. A method according to claim 14, wherein if the mismatch values are all above a discard level, discarding the new video image.

16. A method according to claim 14 wherein after a predetermined number of video images are assigned to a particular camera, the total number of cameras can be determined.

17. The method according to claim 1, wherein channel assignment is performed in real-time.
18. The method according to claim 1, wherein the digital image stream is parsed and the video images are assigned to a channel in real-time
- 5 19. The method according to claim 1, wherein the first video image and the second video image are sub-sampled prior to comparison.
20. The method according to claim 1, wherein if the video image stream contains header information, discarding the header information.
21. The method according to claim 1, further comprising:
10 providing a user interface for setting the mismatch level;
receiving user input of the mismatch level.
22. The method according to claim 1, further comprising:
providing a user interface for setting the discard level;
receiving user input of the discard level.
- 15 23. The method according to claim 15, further comprising:
allowing a user to assign a video image to a camera even though the mismatch is above the discard error level.
24. The method according to claim 15, wherein a user may select video images to discard prior to comparison.
- 20 25. The method according to claim 15, wherein if a video image is associated with a camera and a new video image is also associated with the camera, the new video image becomes the reference image for all further comparisons.
26. The method according to claim 13, wherein a user may clear all reference
25 images and begin the process of assigning a video image to each camera.
27. The method according to claim 26, further comprising:

automatically adjusting brightness within the reference video images and the new video image so that brightness levels are substantially similar prior to comparison.

28. The method according to claim 13, further comprising:

5 selecting a portion of the reference images to be used for comparison to a same portion of the new video image.

29. A computer program product containing computer code thereon readable by a computer for demultiplexing video images in a video stream, the
10 video images produced by a plurality of cameras, the computer code comprising:

 computer code for sequentially receiving the video stream
composed of a plurality of video images in a processor;
 computer code for parsing a first video image from the video
15 stream;

 computer code parsing a second video image from the video
stream;

 computer code for determining an amount of mismatch between
the first and second video image;

20 computer code for assigning the first and second video images to a first channel if the mismatch is below a mismatch threshold.

30. The computer program product according to claim 29, wherein in the
computer code for assigning, the computer code assigns the first and
25 second video images to a memory storage location.

31. The computer program product according to claim 29, wherein the first
channel is associated with a first camera.

32. The computer program product according to claim 29, further comprising:

computer code for assigning the first video image to a first channel and the second video image to a second channel if the mismatch is above a mismatch threshold.

33. The computer program product according to claim 32, wherein the second
5 channel is associated with a second camera.

34. The computer program product according to claim 32,
computer code for comparing a new video image from the video
stream with at least one video image from each channel after each channel
associated with a camera has been assigned a video image;

10 computer code for determining an amount of mismatch between
the new video image and the at least one video image from each channel; and
computer code for assigning the new video image to a channel
having the lowest mismatch.

35. The computer program product according to claim 34, further comprising:
15 computer code for discarding the new video image if the lowest mismatch
is above a discard threshold rather than assigning the new video image to a
channel.

36. The computer program product according to claim 32, further comprising:

20 computer code for comparing a third video image to the first and
the second video images to determine a first and a second mismatch;

computer code for assigning the third video image to a third
channel if the first and second mismatches are above a mismatch threshold.

37. The computer program product according to claim 32, further comprising:

25 computer code for comparing a third video image to the first and
the second video images to determine a first and a second mismatch;

computer code for assigning the third video image to the channel
associated with the lower mismatch if the first mismatch and second
mismatches are below a mismatch threshold.

38. The computer program product according to claim 32, further comprising:
computer code for comparing a third video image to the first and
the second video images to determine a first and second mismatch;
computer code for discarding the third video image if both the first
5 and the second mismatches are above a discard threshold.
39. The computer program product according to claim 32, wherein each
channel is associated with a camera.
40. The computer program product according to claim 29, further comprising:
computer code providing a user interface for entering the number of
10 cameras.
41. A computer program product having computer code thereon for
demultiplexing an image stream having a plurality of images, the computer
code for use with a computer, the computer code comprising:
15 computer code for comparing each new image within the image stream to
a representative image associated with a camera to determine a mismatch
value;
computer code for associating the new image with a new camera if each of
the mismatch values are above a mismatch threshold.
- 20 42. A computer program product according to claim 41, wherein
computer code for selecting the lowest mismatch value and associating
the new video image with the camera associated with the lowest mismatch
value if any of the mismatch values are below the mismatch threshold.
43. A computer program product according to claim 42, further comprising:
25 computer code for discarding the new video image if the mismatch values
are all above a discard level.

44. A computer program product according to claim 42, computer code for determining the total number of cameras after a predetermined number of video images are assigned to a particular camera.
45. The computer program product according to claim 29, wherein channel
5 assignment is performed in real-time.
46. The computer program product according to claim 29, wherein the computer program parses the digital image stream and the video images are assigned to a channel in real-time.
47. The computer program product according to claim 29, further comprising
10 computer code for sub-sampling the first video image and the second video image are prior to comparison.
48. The computer program product according to claim 29, further comprising computer code for identifying header information associated with video images and discarding the header information.
49. The computer program product according to claim 29, further comprising:
15 computer code for providing a user interface for setting the mismatch level;
computer code for receiving user input of the mismatch level.
50. The computer program product according to claim 29, further comprising:
computer code for providing a user interface for setting the discard level;
20 computer code for receiving user input of the discard level.
51. The computer program product according to claim 43, further comprising:
computer code allowing a user to assign a video image to a camera even
though the mismatch is above the discard error level.
52. The computer program product according to claim 43, further comprising
25 computer for allowing a user to select video images to discard prior to comparison.
53. The computer program product according to claim 41, wherein if a video image is associated with a camera and a new video image is also associated

with the camera, the new video image becomes the reference image for all further comparisons.

54. The computer program product according to claim 41, further comprising computer code allowing a user to clear all reference images and to begin the process of assigning a reference video image to each camera.
55. The computer program product according to claim 54, further comprising: computer code for automatically adjusting brightness within the reference video images and the new video image so that brightness levels are substantially similar prior to comparison.
56. The computer program product according to claim 41, further comprising: computer code for selecting a portion of the reference images to be used for comparison to a same portion of the new video image.