

(51) International Patent Classification:
H04N 7/18 (2006.01) *G06T 7/20* (2006.01)(21) International Application Number:
PCT/MY2010/000241(22) International Filing Date:
29 October 2010 (29.10.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PI20095475 21 December 2009 (21.12.2009) MY(71) Applicant (for all designated States except US): **MIMOS BERHAD** [MY/MY]; Technology Park Malaysia, 57000 Kuala Lumpur (MY).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **LIANG, Kim Meng** [MY/MY]; Mimos Berhad, Technology Park Malaysia, 57000 Kuala Lumpur (MY). **TANG, Sze Ling** [MY/MY]; Mimos Berhad, Technology Park Malaysia, 57000 Kuala Lumpur (MY). **LIM, Mei Kuan** [MY/MY]; Mimos Berhad, Technology Park Malaysia, 57000 Kuala Lumpur (MY). **ZULAIKHA, Kadim** [MY/MY]; Mimos Berhad, Technology Park Malaysia, 57000 Kuala Lumpur (MY). **LAI, Weng Kin** [MY/MY]; Mimos Berhad, Technology Park Malaysia, 57000 Kuala Lumpur (MY). **NORSHUHUDA, Samudin** [MY/MY]; Mimos Berhad, Technology Park Malaysia, 57000 Kuala Lumpur (MY). **CHAN, Chee Seng** [MY/MY]; Mimos Berhad, Technology Park Malaysia, 57000 Kuala Lumpur (MY). **MALEEHA, Kiran** [PK/MY]; MimosBerhad, Technology Park Malaysia, 57000 Kuala Lumpur (MY). **ABD BAH'A ALDEEN, Ahmed** [IQ/MY]; Mimos Berhad, Technology Park Malaysia, 57000 Kuala Lumpur (MY).(74) Agent: **YAP, Kah, Hong**; Pyprus Sdn Bhd, Suite 8.02, 8Th Floor, Plaza First Nationwide, 161, Jalan Tun H.S. Lee 50000 Kuala Lumpur (MY).

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

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

Published:

— without international search report and to be republished upon receipt of that report (Rule 48.2(g))

(54) Title: METHOD OF DETERMINING LOITERING EVENT

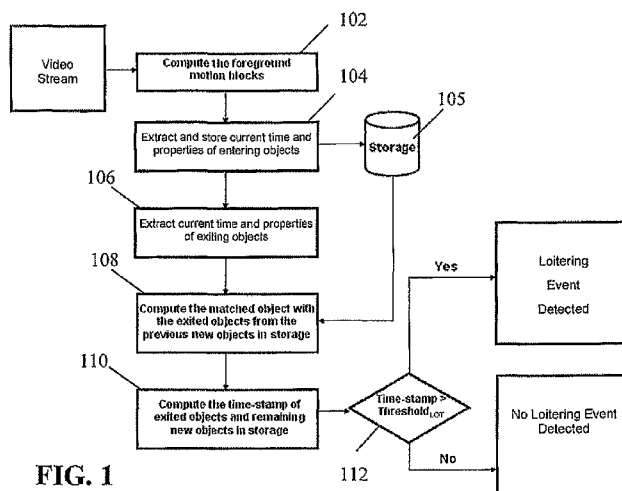


FIG. 1

(57) Abstract: The present invention provides a method for indentifying loitering event occurred on an object within an area of interest from a video stream. The method comprises detecting one or more the objects entering the area-of-interest; extracting (104) an entering time and properties of each of the objects; storing (104) the entering times ad properties of the objects; computing (110) a time-stamp for each object based on a difference between a current time and the entering time of the corresponding object; and identifying (112) a loitering event when the time-stamp is longer than a predetermined period.

METHOD OF DETERMINING LOITERING EVENT

Field of the Invention

The present invention relates to surveillance method. In particular, the present invention relates to a method of determining loitering event.

5

Background

Rising of security concerns has led to the increase of the installation of imaging means such as optical camera in the rooms, buildings, airports, cities and etc for surveillance task. However, it is labour intensive to monitor all the live events or captured videos. Although it is well recognized the monitoring these event manually would be most effective and accurate, but the level of effectiveness is decreases as the attention span and the number of monitor screen increases. Thus, automating the monitoring system would allow the security personnel to carry out the video monitoring task more effectively.

One of the demanding monitoring tasks is to detect loitering event. Detection of loitering event is highly crucial as the loitering behavior is related to harmful activities such as drug-dealing activity, scene investigation for robbery and also unhealthy social problem of teenagers wasting their time in the public area.

State-of-the-art systems and methods require tracking and tagging the object of interest or the area of interest to carry out the loitering detection. The accuracy of the calculated time is highly dependent on the performance of the tracking and tagging methodology.

US6,985,172B1 (US'172) issued to Southwest Research Institute on 10 Jan 2006 discloses a surveillance apparatus and method for detecting incidents. A reference image is generated by removing objects from an image of a region of interest. Accordingly, acquired images are compared with the reference image to detect motion objects.

WO 00/73996 (WO'996) by John Reid et al. published on 7 December 2000 discloses a method of tracking an object moving relative to a scene. It generates a background by acquiring images. Potential objects are matched by comparing spatial

characteristics of potential objects of a following image of the comparing images, and the matched potential object are being taken as a moving object.

As many other systems and methods, loitering object can also be detected through tracking the motion of the detected objects in US'172 and WO'996.

5

Summary

In one aspect of the present invention, there is provided a method for indentifying loitering event occurred on an object within an area of interest from a video stream. Method comprises detecting one or more the objects entering the area-of-interest; extracting (104) an entering time and properties of each of the objects; storing (104) the entering times and properties of the objects; computing (110) a time-stamp for each object based on a difference between a current time and the entering time of the corresponding object; and identifying (112) a loitering event when the time-stamp is longer than a predetermined period.

In one embodiment, the method further comprises detecting an exiting object exiting the area of interest; extracting (106) an exiting time and properties of the exiting object; and matching (108) the exiting object with the exited object against the stored objects based on the properties to identify a corresponding entering object. The current time for computing (110) the time-stamp is the exiting time. The area of interest may cover the entire area of the video stream.

It is also possible that the area of interest includes one or more partial areas within the video stream.

In another embodiment, detection of the one or more objects, including the exiting object, may further comprise computing an overlapping value based on a plurality of motion blocks extracted from the area of interest and a plurality of motion blocks extracted from a previous frame of the video stream. The motion blocks may be computed by any foreground detection technique.

Further, the identification (108) of the corresponding entering object may comprise computing a matching value through matching the properties of the exited

object with the properties of each object stored, wherein the object with the highest matching value is the corresponding entering object.

Brief Description of Drawings

Preferred embodiments according to the present invention and the existing art
5 will now be described with reference to the figures accompanied herein, in which like reference numerals denote like elements.

FIG. 1 is a flowchart illustrating a method of determining loitering event from a video stream in accordance with one embodiment of the present invention;

FIG. 2 is a flow chart illustrating a process for extracting and storing entering
10 time and properties of new entering object in accordance with one embodiment of the present invention;

FIG. 3 is a flow chart illustrating a process for extracting exiting time and properties of a detected exiting object in accordance with one embodiment of the present invention;

15 **FIG. 4** is a flowchart illustrating a process of obtaining a matched object of a exited object from previously stored objects in accordance with another embodiment of the present invention; and

FIG. 5 is a flowchart illustrating a process that computes a time-stamp of an object in question, the exiting object, and remaining entering objects in accordance
20 with one embodiment of the present invention.

Detailed Description

Embodiments of the present invention shall now be described in detail, with reference to the attached drawings. It is to be understood that no limitation of the scope of the invention is thereby intended, such alterations and further modifications
25 in the illustrated device, and such further applications of the principles of the invention as illustrated therein being contemplated as would normally occur to one skilled in the art to which the invention relates.

In one embodiment of the present invention, a method for determining loitering event is presented. The method includes a technique that calculates time-stamps of objects from a video stream to determine if loitering event occurs. The time-stamp is duration of a particular object entering and exiting a region-of-interest or scene of the video stream. Each particular object is referenced with the two specific timings: Entering Time, t_{enter} , and Exiting Time, t_{exit} . The entering time is a present time of an object entering the region-of-interest or scene, while the exiting time is a present time of the object in question exiting the region-of-interest or scene. When the time-stamp exceeding a prescribed period limit, the object in question is considered loitering. It is understood that the present method does not require tracking and/or tagging the object in question throughout the stream of video. The tracking/tagging herein referred to mean any motion detection method that requires labeling or the like.

It is understood that the present invention is applicable to detect loitering event based on features extracted/detected from a video stream, generally life video stream. As mentioned, the method may be used to extract features from one specific area-of-interest or scene from the video stream. It is also possible that the features are detected from more than one area-of-interest or scene or the entire screen of the video stream. In an alternative embodiment, the area of interest includes the entire area of the video stream and one or more partial areas within the video stream. The entire area of the video stream may cover the one or more partial areas within the video stream. Further, when desire, the present invention is also applicable on a (or more) dynamic area-of-interest within a video stream.

In this embodiment, when an entering time is recorded, i.e. an object is entering the area-of-interest or scene of the video stream, properties and the current time of the object in question entering the scene are extracted and stored together with a reference index. Without any limitations, the properties are visual features that include color, texture, shape and etc., which are extractable from a video stream via any possible methods. The reference index is attached to the time and properties captured for indexing. Similarly, when an exiting time is recorded, i.e. any object (whether or not corresponds with the earlier detected object) exiting the region-of-interest or scene, properties and the current time of that object are also extracted. The

properties of the object exiting the region-of-interest or scene are taken to match with the list of properties that are previously stored for objects that entered the region-of-interest or scene. When a matched is found, the entering time of the most matched object (based on the properties) is extracted and computed with the exiting time of the detected object exiting the region-of-interest or scene. For the matched reference index, the properties and present time information is removed from the storage after the extraction process of the corresponding present time. The difference between these two presents times represents a time-stamp of the object in question enters and exit the region-of-interest or scene. For each remaining new objects in the storage, the time-stamp is calculated by finding the difference between the current time of the new object in the storage and the running time of the system. When the difference value, i.e. the time stamp, is larger/longer than a predefined threshold, a loitering event is detected. Otherwise, no loitering event is detected.

FIG. 1 is a flowchart illustrating a method of determining loitering event from a video stream in accordance with one embodiment of the present invention. Briefly, the method comprises computing foreground motion blocks in the region-of-interest or scene at step **102**; extracting and storing a present time and properties of new objects entering the region-of-interest or scene at step **104**; extracting a present time and properties of an exiting object at step **106**; computing a matched object by matching the exited object with stored entering objects at step **108**; computing a time-stamp at step **110**; and determining if the time-stamp is larger/longer than a predefined threshold at step **112**.

Referring back to the step **102**, foreground motion blocks are computed in the region-of-interest or scene to detect an object entering the area-of-interest or scene. In one non-limiting embodiment, the foreground motion blocks are determined by finding intensity value differences between a current frame and a previous frame of the video stream. Feature extractions from a digital video stream through intensity are widely known in the art, and thus will not be described herein in details. When a new motion block is determined, a new object is considered detected entering the area-of-interest or scene of the video stream. In the step **104** the present time and properties of motion block that are identified as a new object are extracted and recorded in a storage **105** with reference index. The storage **105** is provided to store all the entering

time and properties of each entering object. At the step **106**, when the new object is detected exiting the area-of-interest or scene, the present time and properties of exited object are extracted. Both entering time and exiting time, and properties of the entering and exiting objects may be extracted with any techniques well known in the art. At the step **108**, properties of the exiting object are compared and matched with those of the entering object(s) stored in the storage to determine a corresponding entering object. Once a matched entering object is found, the corresponding entering time and properties are extracted from the storage. At the step **110**, exiting time of the exiting object and the entering time of the corresponding entering object are computed to obtain a time-stamp. In addition, the process may further calculate a time stamp of the remaining stored object (i.e. those that are not matched) based on their entering time against a present time. At the step **112**, the time-stamp is then evaluated against the predefined threshold, $\text{Threshold}_{\text{LOT}}$, when the time-stamp is larger/longer than the predefined threshold, a loitering event is detected. When the time-stamp is smaller/shorter than the predefined threshold, no loitering event is detected with respect to the exiting object within the area-of-interest or scene.

FIG. 2 is a flow chart illustrating a process for extracting and storing entering time and properties of new entering object in accordance with one embodiment of the present invention. The process comprises computing overlapping values of a current motion block with all previously stored motion blocks at step **202**; determining if the overlapping value is smaller than a predetermined threshold, $\text{Threshold}_{\text{enter}}$, at step **204**; extracting an entering time of a new entering object at step **206**; extracting properties of the new entering object at step **208**; storing the entering time and properties of the new entering object at the storage **105** at step **210**; and determining if any further new entering motion blocks exist at step **212**.

Referring back to the step **202**, the process computes the overlapping value of the currently detected motion block with all the previously stored motion blocks. The previously stored motion blocks may be stored at a separate storage (not shown in the **FIG. 2**). The previously stored motion blocks are the motion blocks that are computed in the previous frame of the video stream. The overlapping value is calculated by determining a total number of overlapping pixels from two motion blocks, i.e. the current motion block and previously stored motion block that are

overlapping with each others, then dividing the total number of overlapping pixels with a largest area from the current motion block or previous motion block. If overlapping value is lesser than the predetermined threshold, $\text{Threshold}_{\text{enter}}$, the steps 206, 208 and 210 are executed. Otherwise, at the step 212, the process checks if further motion block(s) are to be checked against the previously stored motion blocks, if so, the process loops back to the step 202 to determine if the further motion block has a overlapping value that smaller than the predetermined threshold, $\text{Threshold}_{\text{enter}}$. If not, the process for extracting and storing the time stamp and properties ends. It is to be noted that when the overlapping value is larger than (i.e. not smaller than) the predetermine threshold value, it represents that there is no new object entering the area-of-interest or scene.

Referring back to the step 204, when the overlapping value is smaller than that of the predetermined threshold, $\text{Threshold}_{\text{enter}}$, a new entering object is detected. Then the process extracts an Entering Time of the entering object at the step 206. In the following step 208, the process further extracts properties of the entering object, and annotated as Entering Properties. In the step 210, the Entering Time and Entering Properties corresponds to the new motion block is stored in the storage 105 with a unique reference index in the storage. After the Entering Time and Properties are stored, the process proceeds to the step 212 to check if any further new motion block to be processed.

FIG. 3 is a flow chart illustrating a process for extracting exiting time and properties of a detected exiting object in accordance with one embodiment of the present invention. This process is carried out when a new exiting object is detected. The process comprises computing overlapping values of all previous motion blocks with all current motion blocks at step 302; determining if the overlapping value is smaller than a predetermined threshold, $\text{Threshold}_{\text{exit}}$, at step 304; extracting an exiting time of a detected exiting object at step 306; extracting properties of the detected exiting object at step 308; determining a matched object of the detected exiting object among the entering objects stored at the storage 105 at step 310; and determining if further previous motion blocks are to be compared at step 312.

Referring back to the step 302, the process computes the overlapping value of the previous motion block that is stored separately with all the detected motion

blocks. When the computed overlapping value is lesser than the predetermined threshold, $\text{Threshold}_{\text{exit}}$, the steps 306, 308 and 310 are executed. Otherwise, at the step 312, the process checks whether further previous motion block(s) in the storage are to be processed, if so, the process loops back to the step 302 to determine if the further previous motion block has a overlapping value that smaller than the predetermined threshold, $\text{Threshold}_{\text{exit}}$. If not, the process for determining a corresponding previous motion block ends.

Referring back to the step 304, when the overlapping value is smaller than that of the predetermined threshold, $\text{Threshold}_{\text{exit}}$, an exiting object is detected. Then an Existing Time of the detected exiting object at the step 306 is extracted. In the following step 308, properties of the exiting object is also extracted, and annotated as Exiting Properties. Once the Exiting Time and Exiting Properties corresponds to the detected existing motion block obtained, the process computes a matched object of detected existing object among the entering objects stored at the storage 105 at step 310. Then, the step 312 is executed to check if any further previous motion block to be processed.

FIG. 4 is a flowchart illustrating a process of obtaining a matched object of a exited object from previously stored (entering) objects in accordance with another embodiment of the present invention. This process is meant to identify a corresponding entering object for the exited object. The process comprises computing a matching value of properties of exiting object with those stored in the storage 105 at step 402; storing the matching values with the corresponding properties in a storage 105 at step 404; determining if any properties entries exist at step 406; extracting the current time with the highest matching value from storage at step 408; storing the corresponding entering time and exiting time in a temporary storage 450 at step 410; and removing the property entries with the highest matching value at step 412 from the storage 105.

Referring back to the step 402, the matching value is derived from the properties of the exiting object with each property entry of the entering objects previously stored in the storage 105. Then the corresponding matching values are stored with their corresponding entry in the storage 105 at the step 404. In the step 406, when any of the property entries in the storage 105 still exist for processing, the

step of computing the matching value in step 402 is repeated. The loops are carried out until all the property entries are processed. At the step 408, when all the property entries in the storage 105 are processed, the corresponding entering time with the highest matching value is extracted from the storage 105. At the step 410, the exiting time of the object in question and the corresponding entering time are stored temporary in the memory storage 450, and subsequently, at the step 412, the property entry with the highest matching value is removed from the storage 105. It signifies that the information that belongs to the object in question is removed as the object in question exits the region-of-interest or scene.

FIG. 5 is a flowchart illustrating a process that computes a time-stamp of an object in question, the exiting object, and remaining entering objects in accordance with one embodiment of the present invention. The process comprises computing time-stamp at step 502; extracting the current time of the running system at step 504; and computing time-stamps for the remaining objects stored in the storage 105. At the step 502, the time-stamp for the exiting object is computed through the Exiting Time and the corresponding Entering Time stored in the temporary memory storage 450 at step 502. The time-stamp of each exited object is calculated with Equation (1) as follows:

$$\text{Time-stamp} = \text{Exiting Time} - \text{Entering Time} \quad (1)$$

Subsequently, at step 504, the current time of the running system is extracted there from and annotated as Running Time. At step 506, the time-stamps of each of the remaining objects in the storage 105 is calculated with Equation (2) as follows:

$$\text{Time-stamp}_{\text{remaining}} = \text{Running Time} - \text{Entering Time} \quad (2)$$

The time-stamp for the remaining entering objects can be used to determine if the entering objects has been loitering even if the objects never exit the area-of-interest or scene of the video stream.

In the present application, a method of identifying loitering event provided. The time-stamp of an object under surveillance that represent the duration of the object that continuously appears in the region-of-interest or scene is used as a factor to determine loitering event. The computation of the time-stamp in this invention is

determined by calculating the duration between the time of the object appears in the region-of-interest or scene and the time of the same object disappears from the region-of-interest or scene.

5 While specific embodiments have been described and illustrated, it is understood that many changes, modifications, variations and combinations thereof could be made to the present invention without departing from the scope of the invention.

CLAIMS

1. A method for identifying loitering event within an area of interest from a video stream, the method comprising:
 - detecting one or more objects entering the area-of-interest;
 - 5 extracting (104) an entering time and properties of each of the objects;
 - storing (104) the entering times and properties of the objects;
 - computing (110) a time-stamp for each object based on a difference between a current time and the entering time of the corresponding object; and
 - identifying (112) a loitering event when the time-stamp is longer than a
 - 10 predetermined period,
 - wherein the method is carried out without tracking and/or tagging each of the object entered the area-of-interest.
2. The method according to Claim 1, further comprising:
 - 15 detecting an exiting object exiting the area of interest;
 - extracting (106) an exiting time and properties of the exiting object; and
 - matching (108) the exiting object with the exited object against the stored objects based on the properties to identify a corresponding entering object,
 - wherein the current time for computing (110) the time-stamp is the exiting
 - 20 time.
3. The method according to Claim 1, wherein the area of interest covers the entire area of the video stream.
- 25 4. The method according to Claim 1, wherein the area of interest includes one or more partial areas within the video stream.
5. The method according to Claim 2, wherein the detecting of the one or more objects, including the exiting object, further comprises computing an overlapping
- 30 value based on a plurality of motion blocks extracted from the area of interest and a plurality of motion blocks extracted from a previous frame of the video stream.

6. The method according to Claim 5, wherein the motion blocks are computed by a foreground detection technique.

5 7. The method according to Claim 2, wherein identifying (108) the corresponding entering object comprises computing a matching value through matching the properties of the exited object with the properties of each object stored, wherein the object with the highest matching value is the corresponding entering object.

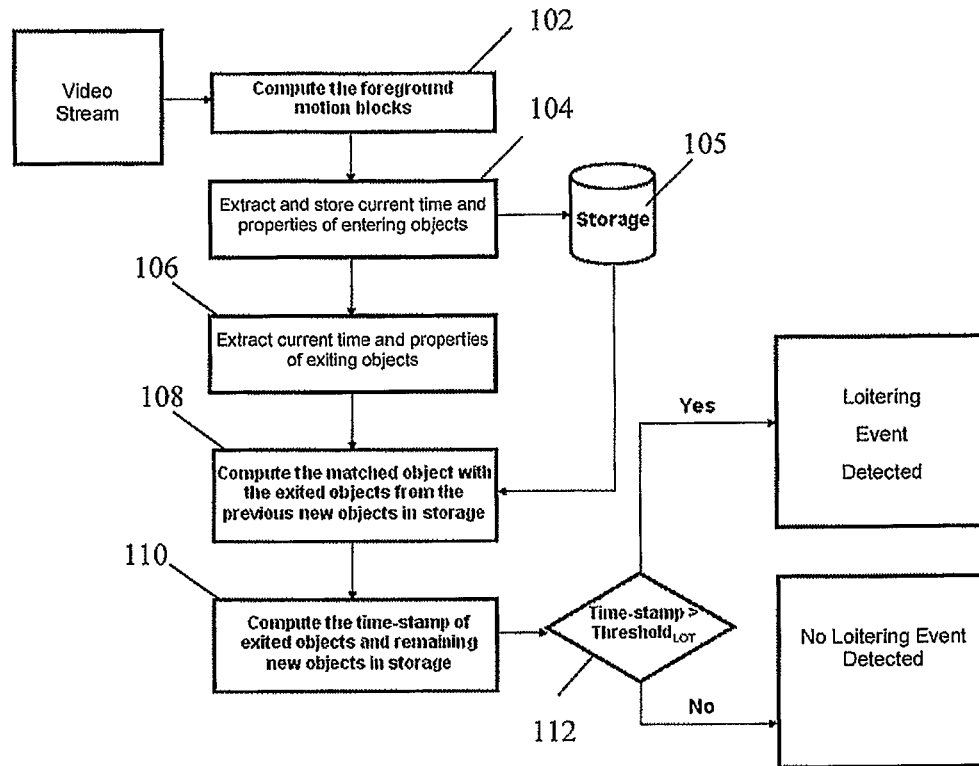


FIG. 1

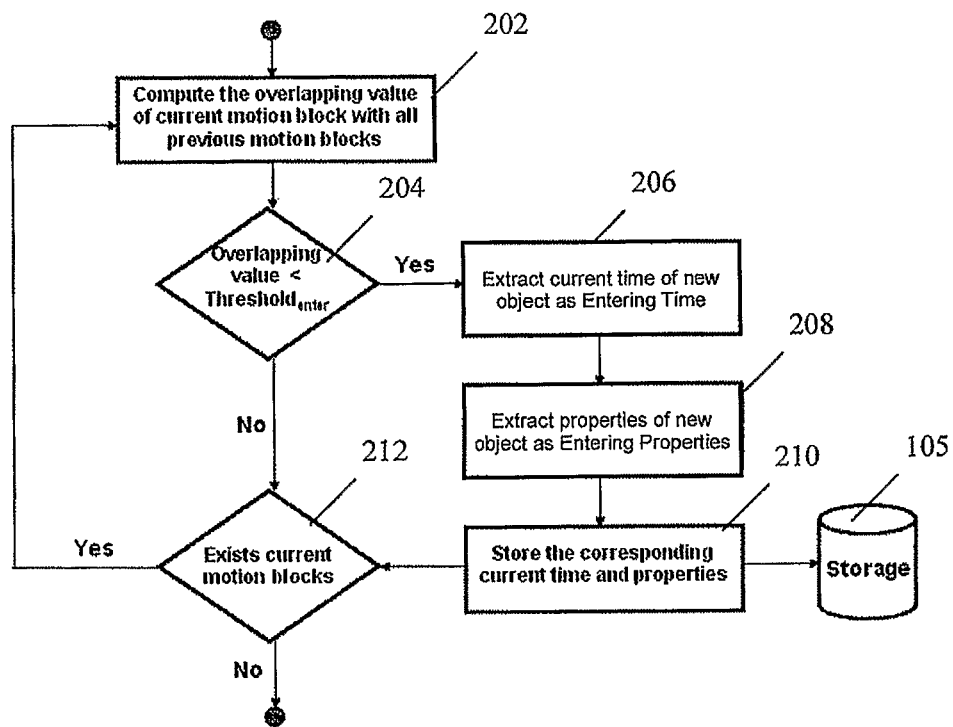


FIG. 2

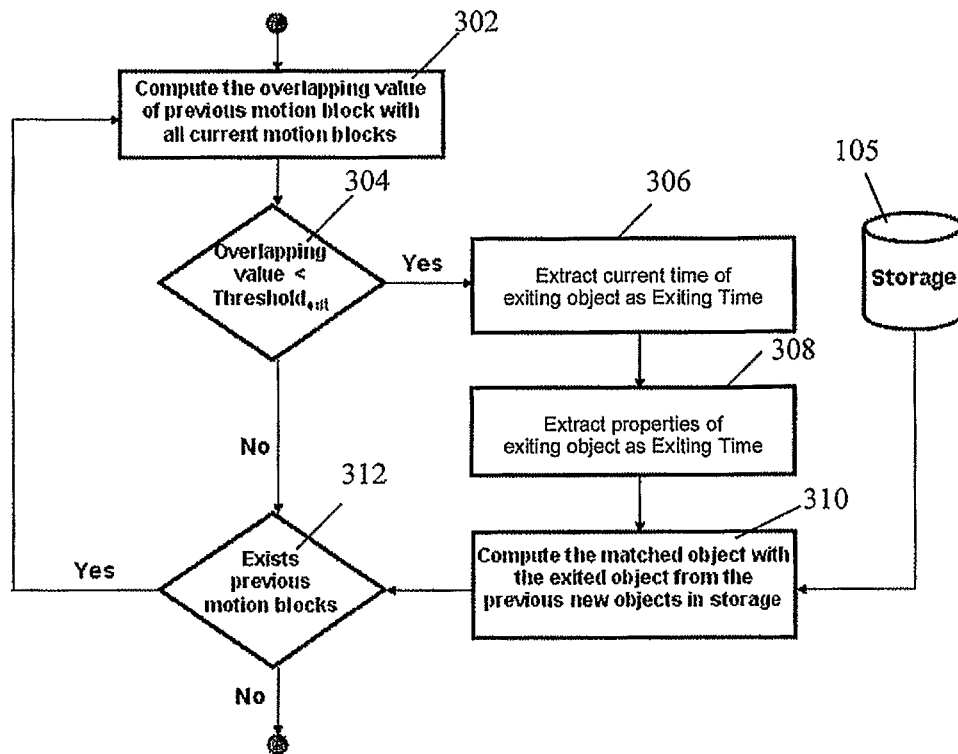


FIG. 3

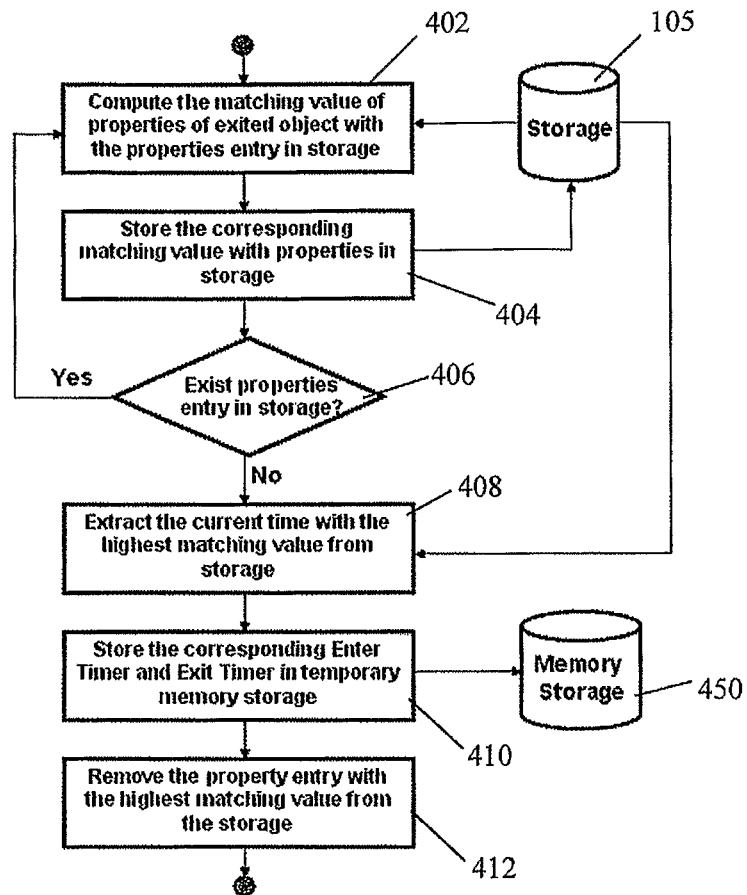


FIG. 4

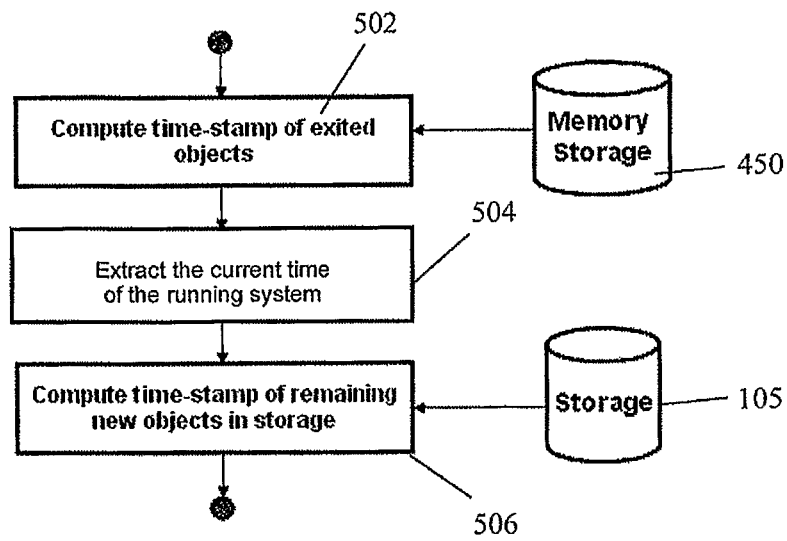


FIG. 5