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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 June 2012 (07.06.2012)

WIPO | PCT

(10) International Publication Number
WO 2012/074366 A2

(51) International Patent Classification:
H04N 7/18 (2006.01)

(21) International Application Number:
PCT/MY2011/000147

(22) International Filing Date:
24 June 2011 (24.06.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PI2010005760 2 December 2010 (02.12.2010) MY

(71) Applicant *(for all designated States except US)*: **MIMOS BHD.** [MY/MY]; Technology Park Malaysia, 57000 Kuala Lumpur (MY).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **LIM, Mei, Kuan** [MY/MY]; c/o Mimos Berhad, Technology Park Malaysia, 57000 Kuala Lumpur (MY). **LIANG, Kim, Meng** [MY/MY]; c/o Mimos Berhad, Technology Park Malaysia, 57000 Kuala Lumpur (MY). **YONG, Yen, San** [MY/MY]; c/o Mimos Berhad, Technology Park Malaysia, 57000 Kuala Lumpur (MY). **CHAU, Sheau, Wei** [MY/MY]; c/o Mimos Berhad, Technology Park Malaysia, 57000 Kuala Lumpur (MY).

(74) Agent: **CHUAH, Jern, Ern**; Advanz Fidelis Sdn. Bhd., Suite 609, Block D, Phileo Damansara 1, No 9, Jalan 16/11, 46350 Petaling Jaya, Selangor (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).

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

[Continued on next page]

(54) Title: A SYSTEM AND A METHOD FOR DETECTING A LOITERING EVENT

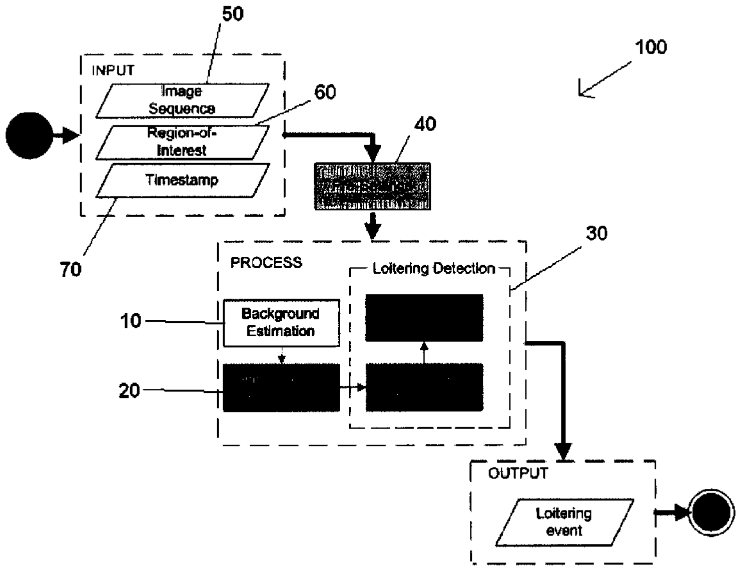


FIG. 1

(57) Abstract: The present invention relates to a system (100) and a method for detecting a loitering event. The system (100) comprises a motion detector (10) for delineating motion pixels or foreground from a background of an image sequence (50); an object detector (20) for detecting a new object in at least one region-of-interest (ROI) (60) from the image sequence (50); and an event detector (30) for detecting the loitering event. In another aspect of the present invention, the method comprises the steps of delineating motion pixels or foreground from a background of an image sequence by means of a motion detector (10), detecting a new object in at least one region-of-interest from the image sequence (50) by means of a object detector (20), and detecting the loitering event by means of an event detector (30).

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

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

A SYSTEM AND A METHOD FOR DETECTING A LOITERING EVENT

FIELD OF THE INVENTION

The present invention relates to a system and a method for determining a loitering event.

BACKGROUND ART

Nowadays, video surveillance systems are rapidly being deployed in public spaces to help strengthen the public safety. This is motivated by the need for better public safety in the society, availability of powerful computing hardware at lower costs and advancement of technology and tools. Generally, surveillance cameras are usually monitored sparingly or not at all; often use merely as a historical archive or in post-mortem, to refer back to an incident is known to have taken place once. Video surveillance can be a far more useful tool, instead of passively recording the footage. They can be used to detect events that demand attention, as they happen to alert the authorities accordingly.

Loitering is generally defined as an act of remaining in a particular public place for a protracted or prolonged period of time. More often than not, the act of loitering is related to harmful activities such as vandalism, prostitution, begging or drug dealing activity. This gives rise to the right of the police officers to disperse such persons. As such, it is crucial to automatically detect a loitering event and to provide more effective video surveillance.

In the state-of-the-art systems and methods to detect a loitering event, tracking mechanisms are often invoked to keep track of the timestamp of each individual appearing in the scene. The timestamp is then compared against the predefined loitering threshold to determine whether an individual is exhibiting loitering behavior or not. Common problems arising from these approaches include:

1. The accuracy to detect a loitering event is highly dependent on tracking

mechanisms, thus the detection is often limited to non-crowded scene in which the rate of occlusion is low.

2. The timestamp associated to each individual is refreshed once the individual is out of the scene.
3. The unavailability of the triggering event location
4. Most conventional system and method of detecting a loitering event triggers an alert when there is any object-of-interest that the timestamp exceeds the predefined threshold, resulting in false alarms.
5. False detection of object that in actual has left the scene due to the existence of object with similar properties in the scene, which is of high possibilities in crowded scene.
6. False negative in which loiterer is not detected when at actual, the loiterer still appears in the scene, which may result from the different perspective of the object in the current scenes as compared to the model.

In the patent document no. US 7,088,846 B2, the time span of each object-of-interest is computed based on the correlation between their trajectory pattern and time. Rule-based reasoning and multiple hypothesis scoring are then applied to detect a loitering event. This method is highly dependent on the time stamp computation for each object-of-interest, thus requires a robust tracker.

In another related patent document, US 6,985,172 B1, images captured from infrared and visible light spectrum sensors are used to allow the method works well during daylight, transition and dark conditions. Their invention incorporates advanced algorithm to include temporal processing and model-based analysis to achieve machine perception and recognition of normal scene motions. Subsequent discrepancies between the scene and the model are then detected and compared to classification criteria for disposition. The model-based approach requires the scene to be monitored over a period of time first to construct the model of the scene. Similar to other prior arts, this method is highly dependent on the results from tracking to detect a predefined event.

In the patent document no. US 2009/002155 A1, an electronic tracking device and a transmitter is introduced so that more accurate tracking information may be obtained. Each of the transmitters can be configured to be associated with a particular object-of-interest or group of objects-of-interest. The location information and trajectory pattern of the object-of-interest obtained from the transmitter is then analyzed to determine the behavior patterns or loitering event. This invention requires the tracking device and transmitter to be attached to the object-of-interest and is not suitable for most surveillance deployment (i.e. public area or uncontrolled environment) in which monitoring is done non-intrusively, without object-of-interest knowing. In addition, this method requires every known object-of-interest to be tracked as it assumes that every object-of-interest has the potential to exhibit loitering behavior.

In another related state-of-the art system, tracking is removed totally to eliminate the dependency on tracking results and thus improving the accuracy of detecting loitering events. However, common problems arising from these approaches include:

1. The unavailability of the triggering event location upon detecting an event.

In one related prior art, the computation of timestamp is determined by computing the difference in duration between the current time and the first appearance of the individual in the Region Of Interest (ROI) or within the current time and the first time when a similar individual disappears from the ROI. In the context of the present invention, the ROI is definable in relation to imaging as a section or a segment to be analyzed defined on a captured image or image sequence.

the boundaries of a tumor may be defined on an image or in a volume, for the purpose of measuring its size.

In light of the above, there appears a need for a system and a method for detecting a loitering event that is able to overcome the drawbacks of the prior art.

SUMMARY OF THE INVENTION

Accordingly, there is provided a system and the method for detecting a loitering event.

In one aspect of the present invention, the system comprises a motion detector for delineating at least one motion pixel from an image sequence, an object detector (20) for detecting at least one object in at least one region-of-interest (ROI) from the image sequence based on the results of the motion detector, and an event detector for detecting the loitering event.

The event detector detects at least one object-of-interest from the at least one objects by using a non-tracking approach and validates the at least one object-of-interest as a loiterer by using a tracking approach. Said tracking approach tracks the object-of-interest including the position of the object-of-interest. Said non-tracking approach tracks the object-of-interest except in terms of the position of the object-of-interest.

In another aspect of the present invention, the method comprises the steps of delineating at least one motion pixel from an image sequence by means of a motion detector of a system for detecting a loitering event, detecting at least one object in at least one region-of-interest from the image sequence based on the results of the motion detector by means of an object detector of the system, and detecting the loitering event by means of an event detector of the system.

At least one object-of-interest is detected from the objects by using a non-tracking approach. The at least one object-of-interest is validated as a loiterer by using a tracking approach. Said tracking approach tracks the object-of-interest including the position of the object-of-interest. Said non-tracking approach tracks the object-of-interest except in terms of the position of the object-of-interest.

It is an object of the present invention to provide a system and a method that allows accurate and reliable loitering event detection not limited to non-crowded scene. This effect is achieved by integrating the non-tracking approach for detecting the loitering event and the simple tracking approach after an event is detected.

It is also an object of the present invention to provide a system and a method that are capable of highlighting triggering event location. This is also achieved by the integration of the non-tracking and the tracking approaches.

It is further object of the present invention to provide a more intuitive set of results in which human operators or authority can quickly understand. The output of this system is an indicator to mark any loitering event if the system detects any object that exhibits loitering behavior or vice versa.

It is also an object of the present invention to affirm that the object-of-interest that has not left the scene and is loitering by introducing a multi layer tracking and analysis after an event is detected to affirm that the object-of-interest has not left the scene and is loitering.

It is also an object of the present invention to reduce false alarms and addresses the issue of false detection of object that in actual has left the scene due to the existence of object with similar properties in the scene, which is of high possibilities in a crowded scene. The system and the method of the present invention achieves this through performing an analysis on all potential loiterer first before deciding on whether an object is exhibiting loitering behavior or not. In addition, a false counter is introduced for a more robust analysis that is not depending on one frame basis.

It is also an object of the present invention to this is to eliminate false negative (the loiterer is not detected when at actual, the loiterer still appears in the scene), which may result from the different perspective of the object in the current scene as compared to the model. The system and the method achieve this by not assuming that the loiterer is no longer in the scene, when the results show otherwise.

It is further an object of the present invention to remove a potential loiterer or loitering from a database once any of the removal condition is met.

It is final an object of the present invention to provide a system and a method that are allowably applied in any video analytics solution to identify abnormal behavior due to loitering to help prevent crime.

The present invention consists of certain novel features and a combination of parts hereinafter fully described and illustrated in the accompanying drawings and particularly pointed out in the appended claims; it being understood that various changes in the details may be made without departing from the scope of the invention or sacrificing any of the advantages of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

For the purpose of facilitating an understanding of the invention, there is illustrated in the accompanying drawings the preferred embodiments from an inspection of which when considered in connection with the following description, the invention, its construction and operation and many of its advantages would be readily understood and appreciated.

FIG. 1 shows the overall architecture of the system for detecting a loitering event of the present invention, which integrates both the non-tracking and tracking approaches.

FIG. 2 shows the process flow of the Pre-Settings sub-component of the present invention.

FIG. 3 shows the overall process flow of the New Object Detection sub-component of the present invention.

FIG. 4 shows the overall process flow of the Loitering Detection sub-component of the present invention.

FIG. 5 shows the process flow of the Assign Tracker step of the present invention.

FIG. 6 shows the overall process flow of the Identity Potential Loiterer step of the present invention.

FIG. 7 shows the overall process flow of the Track Potential Loiterer step of the present invention.

FIG. 8 shows the overall process flow of the Identify Loiterer step of the present invention.

FIG. 9 shows the overall process flow of the track loiterer step of the present invention.

FIG. 10 shows the overall process flow of the update timer step of the present invention.

FIG. 11 shows the examples of scenario, which illustrates loitering behavior.

FIG. 12 shows the overall process flow of the remove loiterer step.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention relates to a system (100) and a method for detecting a loitering event. Hereinafter, the system (100) and the method shall be described according to the preferred embodiments of the present invention and by referring to the accompanying description and drawings. However, it is to be understood that limiting the description to the preferred embodiments of the invention and to the drawings is merely to facilitate discussion of the present invention and it is envisioned that those skilled in the art may devise various modifications without departing from the scope of the appended claim.

As represented in the drawings, the system (100) and the method according to the present invention is described as follows:

Referring to FIG. 1, according to one aspect of the present invention, the system (100) comprises a motion detector (10) for delineating at least one motion pixel from an image sequence (50), an object detector (20) for detecting at least one object in at least one region-of-interest (ROI) (60) from the image sequence (50) based on the results of the motion

detector (10), and an event detector (30) for detecting the loitering event.
The event detector (30) detects at least one object-of-interest from the at least one objects by using a non-tracking approach and validates the at least one object-of-interest as a loiterer by using a tracking approach. Said tracking approach tracks the object-of-interest including the position of the object-of-interest. Said non-tracking approach tracks the object-of-interest except in terms of the position of the object-of-interest.

The non-tracking approach keeps track of a timestamp (70) of each object-of-interest from its first appearance in a scene.

The system (100) filters said loiterer from said object-of-interest to said loiterer in a multi-stages manner. It should be noted that the loiterer is an object that has not left the scene and is loitering after the loitering event is detected.

Referring to FIG. 1 and 2, the systems further comprises a pre-settings component for extracting parameters from the system (100). The parameters include user input parameters that are inputted by a user and stored into the system (100), and default parameters stored into the system (100). The system (100) associates the parameters to the at least one region-of-interest (60) to an extent that each region-of-interest (60) has at least one predetermined threshold. The threshold is predetermined in relation to a timestamp limit of which the loitering event is allowed to occur.

Referring still to FIG. 1, the motion detector (10) is a background estimation sub-component (10). The background estimation sub-component (10) processes an input image, delineates the motion pixels or the foreground from the background in the image sequence (50), and groups together the motion pixels with similar properties as motion blobs to allow the blobs to be emphasized corresponding to the object-of-interest and the noise to be eliminated. Also, an object classification method for discriminating human and non-human blobs is allowed to be used dependent on the functionality of the system (100).

Referring to FIG. 1 and 3, the object detector (20) is a new object detection sub-component (20). The new object detection sub-component (20) is adapted to perform a blob-overlapping between previous and current frames to identify the relationship between the previous and the current frames, perform a similarity matching between the blobs in which features including color, area, directions, and other features are allowed to be used, generates a relationship table to represent the association of each blob in the current frame with the blobs from the previous frame, perform a region-of-interest masking on each blob for validation, and storing the blobs that are detected as the new object into a respective ROI's (60) new object database (190).

Referring to FIG. 1 and 4, the event detector (30) is a loitering detection sub-component (30). The loitering detection sub-component (30) comprises a tracker assigning unit (120) for registering the properties and an entering time of each detected new object into a tracker database (80), a potential loiterer identification unit (130) for identifying and storing the object-of-interest that has the potential to exhibit loitering behavior into a potential loiterer database (90), a potential loiterer tracking unit (140) for performing tracking on each identified potential loiterer in the potential loiterer database (90) and for analyzing the consistency of each tracking trial, a loiterer identification unit (150) for identifying and storing the loiterer into a loiterer database (110) by analyzing correlation scores and the consistency of tracking each potential loiterer, a loiterer tracking unit (160) for continuously tracking and locating the loiterer throughout the lifespan of the scene and analyzing the consistency of each tracking trial, a timer updating unit (170) for consistently accumulating the timestamp (70) of each object-of-interest in each database (80, 90, 110) and estimating the accumulated timestamp (70) of each object-of-interest in the tracker, the potential loiterer, and the loiterer databases (80, 90, 110), and finally, a loiterer removal unit (180) for performing at least one removal condition to each object-of-interest in each database (80, 90,110) and to each invalid object from the database (80, 90, 110).

Referring to FIG. 1 and 4, the event detector (30) is a loitering detection sub-component (30).

The loitering detection sub-component (30) comprises a track assigning unit (120) for registering the properties and current time or entering time of each detected new object into a tracker database (80), a potential loiterer identification unit (130) for identifying and storing the object-of-interest that has the potential to exhibit loitering behavior into a potential loiterer database (90), a potential loiterer tracking unit (140) for performing tracking on each identified potential loiterer in the potential loiterer database (90) and for analyzing the consistency of each tracking trial, a loiterer identification unit (150) for identifying and storing the loiterer into a loiterer database (110) by analyzing correlation scores and the consistency of tracking each potential loiterer, a loiterer tracking unit (160) for continuously tracking and locating the loiterer throughout the lifespan of the scene and analyzing the consistency of each tracking trial, a timer updating unit (170) for consistently accumulating the timestamp (70) of each object-of-interest in each database (80, 90, 110) and estimating the accumulated timestamp (70) of each object-of-interest in the tracker, the potential loiterer, and the loiterer databases (80, 90, 110), and a loiterer removal unit (180) for performing at least one removal condition to each object-of-interest in each database (80, 90, 110) and invalid object from the database (80, 90, 110).

Referring to FIG. 1, 4 and 5, in the track assigning unit (120), the properties and the current time are stored into the tracker database (80) of the respective ROI (60). The extracted properties comprise color properties, blob sizes, speed, and velocity; a timer at each frame increases the time of appearance of each object-of-interest in the scene. The timer associated to each blob is added to the time difference between the current and the previous frames; and the timestamp (70) of each object is accumulated at each image sequence (50).

Referring now to FIG. 1, 4 and 6, with regards to the potential loiterer identification unit (130), the timestamp (70) of each object-of-interest is analyzed in the tracker database (90). If the timestamp (70) of the object-of-interest exceeds the timestamp threshold, a locality prediction analysis is performed on the object-of-interest. Similarity and correlation searches are then performed between the object-of-interest that the timestamp (70) has exceeded the potential loiterer threshold and to each blob in the scene. Particles are randomly distributed around all motion blobs detected within the respective ROI (60) to search for similar object properties. If similar property is detected, the system (100) assumes that there is a possibility that the loiterer has not left the scene. The information of the potential loiterer is then identified and stored into the potential loiterer database (90).

Referring now to FIG. 1, 4 and 7, the potential loiterer tracking unit (140), refined particle filter approach is used to track each potential loiterer. An approximation of the probability density function (pdf) is initially developed. A measure of confidence level of a prediction is then obtained from the pdf. A tracking confidence for each prediction is checked against the tracking confidence threshold to allow the identification of the object that no longer appears in the scene. If the prediction exceeds the tracking confidence threshold, a true counter of the object is increased by one and the locality of the object is updated into the potential loiterer database (90) using the resultant prediction from the particle filter tracking. If the prediction does not exceed the tracking confidence threshold, a false counter is reset.

With reference no to FIG. 1, 4 and 8, in the loiterer identification unit (150), the information of the potential loiterer whom its similarity or correlation is found in the current scene is stored into the potential loiterer database (90). A confidence analysis is initially performed for each potential loiterer. The confidence analysis is done based on the true counter associated to each potential loiterer. If the true counter exceeds the confidence threshold, the system (100) confirms the potential loiterer to be exhibiting loitering behavior.

Referring now to FIG. 1, 4 and 9, in the loiterer tracking unit (160), the particle filter tracking is used to track each loiterer as defined in the loiterer database (110); the similarity matching is able to be represented by approximating the pdf. The tracking confidence for each prediction is checked against the tracking confidence threshold to allow identification and removal of the loiterer that has left the scene. If no similar object properties are found, the false counter is increased by one.

Referring now to FIG. 1, 4, and 10, in the timer-updating unit (170), the timer associated to each blob is added to the time difference between the current and the previous frames. The timer information of each object is incremented from its first appearance in the scene until the object has been confirmed as leaving the scene using a set of coherent rules.

Referring to FIG. 1, 4, and 11, in the loiterer removal unit, the potential loiterer or loiterer is removed from the database (80, 90, 110) once at least one of the removal condition is met. The removal condition is the false counter associated to each potential loiterer in the potential loiterer database (90) and the false counter associated to each loiterer in the loiterer database (110). The removal condition assumes that the particle filter tracking estimation is of lower confidence consistently throughout a number of frames if the object is no longer in the scene. The system (100) removes the potential loiterer or loiterer from the database if the associated false counter exceeds the empirically defined or predefined removal threshold.

In another aspect of the present invention, with reference to FIG. 1, the method comprises the steps of:

- delineating at least one motion pixel from an image sequence (50) by means of a motion detector (10) of a system (100) for detecting a loitering event;
- detecting at least one object in at least one region-of-interest (60) from the image sequence (50) based on the results of the motion detector (10) by means of an object detector (20) of the system (100); and
- detecting the loitering event by means of an event detector (30) of the system (100);

At least one object-of-interest is detected from the objects by using a non-tracking approach. The at least one object-of-interest is validated as a loiterer by using a tracking approach.

Said tracking approach tracks the object-of-interest including the position of the object-of-interest. Said non-tracking approach tracks the object-of-interest except in terms of the position of the object-of-interest.

Said object is filtered from said object-of-interest to said loiterer in a multi-stages manner. The loiterer is an object that has not left the scene and is loitering after the loitering event is detected. The non-tracking approach keeps track of a timestamp (70) of each object from its first appearance in a scene. The non-tracking approach performs similarity and correlation searches on the object-of-interest that the timestamp (70) has exceeded at least one predetermined threshold and on each blobs in the scene in order to identify a potential loiterer. The tracking approach tracks each potential loiterer and analyze the consistency of each tracking trial in order to validate the potential loiterer.

Referring to FIG. 1 and 2, prior to the step of delineating at least one motion pixel, there is a step of pre-setting the system (100). The step of pre-setting the system (100) comprises the steps of extracting parameters from the system (100), of which the parameters include user input parameters that are inputted by a user and stored into the system (100), and default parameters that are stored into the system (100). The system (100) associates the parameters to the at least one region-of-interest (60) to an extent that each region-of-interest (60) has said at least one predetermined threshold. Said at least one threshold is predetermined in relation to a timestamp limit of which the loitering event is allowed to occur.

Referring to FIG. 1, the motion detector (10) is a background estimation sub-component (10). The step of delineating at least one motion pixel comprises the steps of processing an input image, delineating the motion pixels from the background in the image sequence (50), and grouping together the motion pixels with similar properties as motion blobs to allow the blobs to be emphasized corresponding to the object-of-interest and the noise to be eliminated. Also, the step of delineating at least one motion pixel further comprises the step of using an object classification method for discriminating human and non-human blobs dependently on the functionality of the system (100).

The object detector (20) is a new object detection sub-component (20). The step of detecting at least one object comprises the steps of performing a blob-overlapping between the previous and current frames to identify the relationship between the previous and the current frames; performing a similarity matching between the blobs of which features including color, area, directions, and other features that are allowed to be used, generating a relationship table to represent the association of each blob in the current frame with the corresponding blob from the previous frame; performing a region-of-interest masking on each blob for validation, and storing the blobs that are detected as the new object into a respective ROI's (60) new object database (190).

The step of detecting the loitering event comprises the steps of

- registering and entering time of each detected new object into a tracker database (80);
- analyzing the timestamp (70) of each object-of-interest in the tracker database (80);
- performing the similarity and correlation searches between the object-of-interest that the timestamp (70) has exceeded the loitering threshold and to each blob in the scene;
- identifying and storing the information of the potential loiterer whom similarity or correlation is found in the current scene into a potential loiterer database (90);
- tracking each potential loiterer and analyzing the consistency of each tracking trail;
- identifying and storing the information of loiterer whom tracking consistency is more than a defined threshold into a loiterer database (110);
- tracking the loiterer and analyzing the consistency of each tracking trial,
- accumulating the timestamp (70) of each object-of-interest in the each database (80, 90, 110) and estimating the accumulated timestamp (70) of each object-of-interest in the tracker, the potential loiterer, and the loiterer databases (80, 90, 110); and
- performing at least one removal condition to each object-of-interest in each database (80, 90, 110) and invalid object from the database (80, 90, 110).

The threshold further comprises a predetermined timestamp threshold. Referring to FIG. 1, 4 and 5, the step of registering and entering time comprises the steps of storing the properties and the current time into the tracker database of the respective ROI (60), of which the extracted properties comprise color properties, blob sizes, speed, and velocity; increasing the time of appearance of each object-of-interest in the scene by a timer at each frame; adding the timer associated to each blob to the time difference between the current and the previous frames; and accumulating the timestamp (70) of each object at each image sequence (50).

The step of randomly distributing a plurality of particles comprises the step of performing the similarity and correlation searches by distributing particles randomly around all detected motion pixels and calculating the similarity value between the particle and object-of-interest.

Referring still to FIG. 1, 4 and 5, the step of analyzing the timestamp comprises the step of performing a locality prediction analysis on the object-of-interest if the timestamp (70) of the object-of-interest exceeds the predetermined timestamp threshold. The step of performing the similarity and correlation searches comprises the step of randomly distributing a plurality of particles around all motion blobs detected within the respective ROI (60) to search for similar object properties; and assuming that there is a possibility that the loiterer has not left the scene.

The threshold comprises a predetermined tracking confidence threshold; in the potential loiterer tracking unit (140). Referring to FIG. 1, 4, 6, and 7, the step of tracking each potential loiterer comprises the steps of tracking each potential loiterer using a refined particle filter approach; developing an approximation of the probability density function (pdf), of which a measure of confidence level of a prediction is obtainable from the pdf; checking a tracking confidence for each prediction against the predetermined tracking confidence threshold to allow identification of the object that no longer appears in the scene; increasing a true counter of the object by one and updating the locality of the object into the potential loiterer database (90) using the resultant prediction from the particle filter tracking if the prediction exceeds the predetermined tracking confidence threshold; and resetting a false counter if the prediction does not exceed the predetermined tracking confidence threshold.

Referring to FIG. 1, 4, 8, ad 9, the step of identifying and storing the information of the loiterer comprises the steps of storing the information of the potential loiterer whom its similarity or correlation is found in the current scene into the potential loiterer database (90); performing a confidence analysis for each potential loiterer, of which the confidence analysis is done based on the true counter associated to each potential loiterer; and confirming that the potential loiterer is exhibiting loitering behavior if the true counter exceeds the predetermined true counter threshold. The step of tracking the loiterer comprises the steps of tracking each loiterer as defined in the loiterer database (110) using the particle filter tracking, approximating the pdf in order to ably represent the similarity matching, checking the tracking confidence for each prediction against the predetermined tracking confidence threshold to allow identification and removal of the loiterer that has left the scene, and increasing the false counter by one if no similar object properties are found.

The step of accumulating the timestamp (70) comprises the steps of adding the timer associated to each blob to the time difference between the current and the previous frames; and incrementing the timer information of each object from its first appearance in the scene until the object has been confirmed to have left the scene using a plurality of coherent rules.

The threshold further comprises a empirically defined or predefined according to the parameters predetermined removal threshold. The step of performing at least one removal condition comprises the steps of removing the potential loiterer or loiterer from the database (80, 90, 110) once at least one of the removal condition is met. The removal condition is the false counter associated to each potential loiterer in the potential loiterer database (90) and the false counter is associated to each loiterer in the loiterer database (110). The removal condition assumes that the particle filter tracking estimation is of a lower confidence consistently throughout a number of frames if the object is no longer in the scene. The step of performing at least one removal condition comprises the step of removing the potential loiterer or loiterer from the database if the associated false counter exceeds the predetermined removal threshold.

The overall architecture or concept of the present invention comprises of 3 stages; i) input, ii) process and iii) output. The pre-settings sub-component is only required to be performed only once, during the initialization process. This sub-component is adapted to read the user input parameters and associate each parameter to the at least one region-of-interest. The hybrid loitering detection system consists of 3 main sub-components in the *Process stage*

1. *background estimation* sub-component (10) to delineate the motion pixels or foreground from the background in the image sequence (50);
2. *new object detection* sub-component (20) to identify the new object appearing in each ROI (60); and
3. *loitering detection* sub-component (30) to identify loiterer in each ROI (60) using the non-tracking approach and to perform tracking on each identified loiterer thereafter.

The loitering detection sub-component (30) is adapted to perform 7 major steps as follows:

1. *assign tracker* step (120) to register the properties and current time (entering time) of each new object detected into the tracker database (80);
2. *identify potential loiterer* step (130) to identify and store object-of-interest that has the potential to exhibit loitering behavior into the potential loiterer database (90) by analyzing on the timestamp (70) of each object-of-interest registered in the tracker database (80);
3. *track potential loiterer* step (140) to perform tracking on each identified potential loiterer in the potential loiterer database (90);
4. *identify loiterer* step (150) to identify and store loiterer into the loiterer database (110) by analyzing on the correlation scores and the consistency of tracking for each potential loiterer;

5. *track loiterer* step to continuously track and locate loiterer throughout their lifespan in the scene;
6. *update timer* step to consistently accumulate the timestamp of each object-of-interest in each database (80, 90, 110); and
7. *remove loiterer* step to perform the at least one removal condition to each object-of-interest in each database (80, 90, 110) and invalid object from the database (80, 90, 110).

The tracking approach invoked would provide a more intuitive set of results in which human operators or authority can quickly understand. The output of this system (100) is an indicator to mark any loitering event if the system (100) detects any object that exhibits loitering behavior and vice versa.

The present invention introduces an integration of both the tracking and non-tracking approaches to fully utilize the advantages of both. The non-tracking method is for detecting loitering event for a more accurate detection in crowded scene of which the occlusion rate is usually high. Once a loitering event is detected, the tracking on the detected loiterer is performed to estimate the locality of the triggering event. The present invention overcomes the challenge of inaccurate tracking in crowded scenes as well as the unavailability to provide the triggering event location.

The loitering event is detected if an object appears in the predefined ROI (60) within an interval when the associated timestamp (70) exceeds the predefined timestamp threshold.

As mentioned above, the overall architecture of this invention which comprises of 3 stages namely i) input, ii) process and iii) output is as shown in FIG. 1. There are 3 inputs to the system; i) the sequence of images or video streams captured from at least one sensor, ii) the region-of-interest (ROI) in the scene to be monitored and iii) at least one loitering threshold that differentiates between the normal and loitering behavior (longer than would be necessary). Each ROI should be associated to one loitering threshold. The number of input parameters is allowably varied to allow flexibility to a user to configure the system accordingly. Prior to performing the processes involved to detect the loitering event, the hybrid system is adapted to perform a pre-settings step to associate each input parameters to each ROI (60).

The tracking approach invoked would provide a more intuitive set of results in which human operators or authority can quickly understand. The output of this system is an indicator to mark any loitering event if the system detects any object that exhibits loitering behavior and vice versa. The indicator can be of audio or visual triggers to alert the authority and may include a text indicating *'Loiterer Detected'* which is overlaid on the screen or bounding boxes to highlight the location of the triggering events.

Pre-Settings

The *Pre-Settings* sub-component (40) needs to be performed once only, during the installation of the hybrid system (100) to any sensor, also known as the initialization process. In this sub-component (40), the hybrid system (100) extracts the parameters input by the user. If the input parameters are valid, the parameters will be associated to each predefined ROI (60) as defined by the user. Otherwise, the system (100) associates the at least one predefined ROI (60) with the default parameters. In scenarios in which the ROI is not defined, the system (100) considers the entire image as the ROI (60). The *Pre-Settings* step is crucial to allow different ROIs (60) in the scene to have different loitering threshold. 2. For example, the region near the entrance to a highly secured room should have a stricter loitering threshold (shorter timestamp or duration) as compared to the region near the entrance to a lift. This alludes to the fact that people tend to spend a longer time near the entrance to a lift rather than the entrance to a secured room.

Process

Referring to FIG. 2, once the *Pre-Settings* step has been performed, each image sequence is fed into the *Process* stage which consists of 3 main sub-components; the i) *background estimation* sub-component (10), ii) the *new object detection* sub-component (20) and iii) the *loitering detection* component (30). The novelty of the hybrid method to detect and track loitering event using the integration of non-tracking and tracking approaches is described in this stage.

Background Estimation Sub-component (10)

The first sub-component, the *Background Estimation* sub-component (10) processes the input image and extracts the motion pixels or foreground from the background in the image sequence (50). This sub-component is allowed to be optional. This outputs a binary map (not shown) of which the zero pixels indicate the background while the non-zero pixels represent the motion pixels. A connecting component criterion may be applied on the binary map to group together motion pixels with similar properties as motion blobs. The key idea of grouping motion pixels into motion blobs is to emphasize blobs corresponding to object-of-interest and eliminate noise as much as possible. An object classification method may also be applied beforehand to discriminate between human and non-human blobs so that only preferred object-of-interest is analyzed further by the subsequent processes. This step is dependent on the aim or functionality of the system (100). For example, if the system (100) is only interested in analyzing human subject, then a human classification method is allowably applied. The resultant binary map is then fed into the subsequent sub-component for further processing.

New Object Detection Sub-component (20)

The *New Object Detection* sub-component (20) is adapted to perform blob overlapping between current and previous frames to identify the relationship between the current and previous frames. This sub-component is allowed to be optional. Amongst the features that can be used for similarity matching between blobs in the current and previous frames includes colour, area, directions, and any other features in which the similarity matching is able to be performed. Then, a relationship table to represent the association of each blob in the current frame with blobs from the previous frame is constructed. Subsequently, region-of-interest masking is performed on each blob for validation. Only blobs which are detected as new object and falls within the specified ROI (60) is stored into the respective ROI's new object database (190) for further processing. An object that has been deemed as new object in one ROI (60) is also allowably a new object in another ROI (60) if it is detected to be within the latter at any frame. Otherwise, the blob is ignored. The overall process flow is as shown in FIG. 3.

Loitering Detection Sub-component (30)

The final sub-component is the *Loitering Detection* sub-component (30). It comprises of 7 steps which will be hereinafter described in greater detail. This sub-component (30) introduces a novel approach that do not require an object to be tracked or tagged using any tracking mechanism to determine loitering event. Thus, this hybrid system (100) is not limited to detect loitering event in non-crowded scene but is also capable to detect loitering event in scenarios where the rate of occlusion is high. Unlike most of the state-of-the art method that tracks each object-of-interest to determine the timestamp (70) of each object appearing in the scene. The present invention also contains a non-tracking approach that keeps track of the timestamp (70) of each object from its first appearance (new object) in the scene. For each new object identified, the properties of the object-of-interest and the entering time associated is stored in a database called the tracker database (80). At each frame, the timer increases the time of appearance of each object-of-interest in the scene and their timestamp (70) are accumulated. Only object that has exceeded the predefined threshold associated to the predefined ROI (60) invokes the locality prediction analysis step; in which a similarity search is performed to find identical object appearing in the current frame. If there is a similar object appearing in the scene at current frame, the hybrid system (100) determines that there is a potential that the object-of-interest is still in the scene (have not exited the scene). Thus, a tracking mechanism using the particle filter approach is triggered to continue tracking the object-of-interest for a number of frames. Once an object-of-interest is tracked successfully for a predefined number of frames or time (can be represented by using a threshold to denote the sensitivity or how responsive the system is to detect loitering event) loitering event is confirmed and an alert is triggered. The present invention only invokes tracking after detecting loitering event, allowing the hybrid system (100) to highlight the location of the triggering event. Thus, the novel hybrid method that integrates the non-tracking and tracking approach provides an accurate and reliable detection which is not limited to a non-crowded scene. An additional checking layer to reset the system when there are no moving objects appearing in the scene for a predefined number of frames or duration is also presented. The overall process flow of the proposed loitering detection sub-component (30) is as illustrated in FIG. 4. Further details of the steps are as described below:

Step 1: Assign Tracker

Referring to FIG. 1, and 4, in the *Assign Tracker* step (120), the properties and current time of each new object detected by the *New Object Detection* sub-component (20) is extracted and stored in database (80) of the respective ROI (60). For example, if a new object is detected in ROI (60) labeled 1, the properties and current time of the new object is stored in the Tracker database (80) of ROI (60) 1. The properties extracted consist of temporal features that include colour properties, blob size, speed and velocity. The timestamp (70) of each object is then accumulated at each image sequence accordingly by adding the timer associated to each blob to the time difference between the current and previous frames in Step 6. The overall process flow of the *Assign Tracker* step is as shown in FIG. 5.

Step 2: Identify Potential Loiterer

Referring to FIG. 1, 4, and 6, in the *Identify Potential Loiterer* step, every object stored in the tracker database (80) for each ROI (60) is processed and analyzed. The accumulated timestamp (70) associated to each object-of-interest in the database (80) will be compared against the loitering threshold associated to the ROI (60). If the timestamp (70) of the object-of-interest exceeds the predefined threshold, a locality prediction analysis on the object-of-interest is performed.

A method that imitates the particle distribution is introduced in particle filter tracking approach. This method distributes an army of particles randomly around all motion blobs detected within the respective ROI (60) to search for similar object properties. This modified approach does not perform tracking yet as in the conventional particle filter approach. Thus, the present method of finding similarity between properties of object-of-interest in the database (80) that has exceeded the loitering threshold and the properties of objects in the current frame is simpler and faster. If no similar object is found in the current frame, the system continues to another object in the database (80). Otherwise, if a similar property is found, the hybrid system assumes that there is a possibility that the loiterer has not left the scene. Thus, the object-of-interest and its properties are moved from the tracker database (80) into the potential loiterer database (90). Most conventional method of detecting loitering event triggers an alert when there is any object-of-interest that timestamp exceeds the predefined threshold, resulting in false alarms. Thus, the present system (100) and method perform further analysis on all potential loiterer first before deciding whether an object is exhibiting loitering behavior or not. The present system (100) and method also addresses the issue of false detection of object that in actual has left the scene due to the existence of object with similar properties in the scene, which is of high possibilities in crowded scene.

Step 3: Track Potential Loiterer

Referring to FIG. 1, 4, and 7, in the *Track Potential Loiterer* step, every potential loiterer stored in the tracker database (80) for each ROI (60) is tracked to confirm its appearance in the scene before deciding that the object-of-interest is indeed exhibiting loitering behavior. Refined particle filter approach is used to track every potential loiterer, given the properties extracted during the earlier process in the new object detection sub-component (20). Particle Filter Tracking is another statistical method for correspondence in the presence of dynamics. The basic principle of the particle filter method is to develop an approximation of the probability density function, pdf of the state vector as a set of weighted samples by random particles, which is recursively, updated based on available measurements. Since this pdf embodies all available statistical information, the pdf is substantially the complete solution of the estimation problem. Next, the particle's filter confidence level or tracking confidence for each prediction is checked against a predefined threshold to allow identification of object that no longer appears in the scene. In principle, a measure of confidence level of the prediction is obtained from the pdf (correlation score). If the prediction exceeds the predefined threshold (usually is of value between 0 to 1), the true counter of the object is increased by one and the locality of the object is updated into the database (90) using the resultant prediction from particle filter tracking. Otherwise, the false counter is increased by one. Each time the true counter is increased, the false counter is reset. The false counter is introduced for a more robust analysis that is not depending on one frame basis. Instead the analysis is done on a sequence of frame or blocks to reduce false alarms.

Step 4: Identify Loiterer

Referring to FIG. 1, 4, and 8, subsequently, the Identify Loiterer step is performed. In this step, confidence analysis is performed for each potential loiterer stored in the database (90). The confidence analysis is done based on the true counter associated to each potential loiterer. If the true counter exceeds a predefined confidence threshold, the system (100) confirms the potential loiterer to be exhibiting loitering behavior. The true counter represents the confidence of the particle filter tracking approach to track the loiterer for a specific duration which is characterized by the predefined confidence threshold. The present hybrid method does not depend on tracking information to increase the timestamp (70) of individual object. Thus, this step is crucial to ensure that the potential loiterer has yet to leave the scene or is still loitering in the scene before making a conclusion that loitering event is detected.

Step 5: Track Loiterer

Referring to FIG. 1, 4, and 9, now that the at least one loiterer has been detected, again the present invention uses the particle filter tracking method to track each loiterer as defined in the loitering database (110). Similar to step 3 which tracks potential loiterer, the particle's filter confidence level or tracking confidence for each prediction is checked against a predefined threshold to allow identification and removal of loiterer that has left the scene. If the locality estimated by the particle filter method has very similar properties as the model loiterer, the true counter of the loiterer is increased by one and the false counter is reset. The similarity matching is allowably represented by approximating the probability density function or pdf. Otherwise, if no similar object properties are found, the false counter is increased by one. The present method does not assume that the loiterer is no longer in the scene at once, when the pdf is below a defined confidence level. This is to eliminate false negative (the loiterer not detected when at actual, the loiterer still appears in the scene), which may result from the different perspective of the object in the current as compared to the model. For example, an object may have slight difference in its color properties when captured from its frontal view as compared to its back view.

Step 6: Update Timer

Referring now to FIG. 1, 4 and 10, a major drawback of state-of-the art method is the dependency on tracking results to determine the timestamp (70) of each object in the scene. The present method of non-tracking increases the timestamp of each object appearing in either the tracker database (80), potential loiterer database (90) or loiterer database (110) for each ROI using simple logic; by adding the timer associated to each blob to the time difference between the current and previous frames. The present approach will continue incrementing the timer information of each object from its first appearance in the scene until the object has been confirmed to have left the scene using a set of coherent rules as explained in Step 7. Since the present invention does not require tracking of objects to estimate the timestamp (70). FIG. 7 illustrates the 2 possible scenarios in which loitering event is detected using the present invention.

Step 7: Remove Loiterer

Referring to FIG. 1 and 10, this step is crucial as there is tendency of a loiterer to leave the scene after sometime. Also, once the authority personnel has been alerted that a loitering event has taken place it is not necessary to continue triggering an alert for the same event. Thus, the present invention uses a method to remove potential loiterer or loiterer from the database (80, 90, 110) once any of the removal condition is met. In FIG. 12, the removal condition is shown as the false counter associated to each potential loiterer in the potential loiterer database (90) and the false counter associated to each loiterer in the loiterer database (110). This removal condition assumes that the particle filter tracking's estimation is of the lower confidence consistently throughout a number of frames if the object is no longer in the scene. Therefore, if the associated false counter exceeds an empirically defined or predefined threshold, the system (100) removes the potential loiterer or loiterer from the database (80, 90, 110). The detected loiterer is assumed to have left the scene and is no longer exhibiting potential threat. Other removal condition using the same logic is allowed to be adopted. Amongst the possible removal conditions include but are not limited to overlapping regions between more than one loiterers or potential loiterers.

While in the foregoing specification this invention has been described in relation to certain preferred embodiments thereof and many details have been set forth for purpose of illustration, it will be apparent to those skilled in the art that the invention is susceptible to additional embodiments and that certain of the details described herein can be varied considerably without departing from the basic principles of the invention.

CLAIMS

1. A system (100) for detecting a loitering event; characterized in that the system (100) comprises:

a motion detector (10) for delineating at least one motion pixel from an image sequence (50);

an object detector (20) for detecting at least one object in at least one region-of-interest (ROI) (60) from the image sequence (50) based on the results of the motion detector (10); and

an event detector (30) for detecting the loitering event, the event detector (30) detects at least one object-of-interest from the at least one objects by using a non-tracking approach and validates the at least one object-of-interest as a loiterer by using a tracking approach; wherein said tracking approach tracks the object-of-interest including the position of the object-of-interest; and said non-tracking approach tracks the object-of-interest except in terms of the position of the object-of-interest.

2. A system (100) as claimed in Claim 1 wherein the non-tracking approach keeps track of a timestamp (70) of each object-of-interest from its first appearance in a scene; and the system (100) filters said loiterer from said object-of-interest to said loiterer in a multi-stages manner; wherein the loiterer is an object that has not left the scene and is loitering after the loitering event is detected.

3. A system (100) as claimed in Claim 2 further comprising a pre-settings component for extracting parameters from the system (100), and the parameters are selected from a group consisting of user input parameters inputted by a user and stored into the system (100); and default parameters stored into the system (100); wherein the system (100) associates the parameters to the at least one region-of-interest (60) to an extent that each region-of-interest (60) has at least one predetermined threshold; and the threshold is predetermined in relation to a timestamp limit of which the loitering event is allowed to occur.

4. A system (100) as claimed in Claim 3 wherein the event detector (30) is a loitering detection sub-component (30); and the loitering detection sub-component (30) comprises a tracker assigning unit (120) for registering the properties and an entering time of each detected new object into a tracker database (80); a potential loiterer identification unit (130) for identifying and storing the object-of-interest that has the potential to exhibit loitering behavior into a potential loiterer database (90); a potential loiterer tracking unit (140) for performing tracking on each identified potential loiterer in the potential loiterer database (90) and for analyzing the consistency of each tracking trial; a loiterer identification unit (150) for identifying and storing the loiterer into a loiterer database (110) by analyzing correlation scores and the consistency of tracking each potential loiterer; a loiterer tracking unit (160) for continuously tracking and locating the loiterer throughout the lifespan of the scene and analyzing the consistency of each tracking trial, a timer updating unit (170) for consistently accumulating the timestamp (70) of each object-of-interest in each database (80, 90, 110) and estimating the accumulated timestamp (70) of each object-of-interest in the tracker, the potential loiterer, and the loiterer databases (80, 90, 110); and a loiterer removal unit (180) for performing at least one removal condition to each object-of-interest in each database (80, 90,110) and to each invalid object from the database (80, 90, 110).

5. A method for detecting a loitering event; characterized in that the method comprises the steps of:

delineating at least one motion pixel from an image sequence (50) by means of a motion detector (10) of a system (100) for detecting a loitering event;

detecting at least one object in at least one region-of-interest (60) from the image sequence (50) based on the results of the motion detector (10) by means of an object detector (20) of the system (100);

detecting the loitering event by means of an event detector (30) of the system (100); wherein at least one object-of-interest is detected from the objects by using a non-tracking approach, the at least one object-of-interest is validated as a loiterer by using a tracking approach; wherein said tracking approach tracks the object-of-interest including the position of the object-of-interest; and said non-tracking approach tracks the object-of-interest except in terms of the position of the object-of-interest.

6. A method as claimed in Claim 5 wherein said object is filtered from said object-of-interest to said loiterer in a multi-stages manner, the loiterer is an object that has not left the scene and is loitering after the loitering event is detected; wherein the non-tracking approach keeps track of a timestamp (70) of each object from its first appearance in a scene; the non-tracking approach performs similarity and correlation searches on the object-of-interest that the timestamp (70) has exceeded at least one predetermined threshold and on each blobs in the scene in order to identify a potential loiterer; and the tracking approach tracks each potential loiterer and analyze the consistency of each tracking trial in order to validate the potential loiterer.

7. A method as claimed in Claim 6 further comprising, prior to the step of delineating at least one motion pixel, the step of pre-setting the system (100); wherein the step of pre-setting the system (100) comprises the steps of extracting parameters from the system (100); wherein the parameters are selected from a group consisting of user input parameters inputted by a user and stored into the system (100), and default parameters stored into the system (100); the system (100) associates the parameters to the at least one region-of-interest (60) to an extent that each region-of-interest (60) has said at least one predetermined threshold; and said at least one threshold is predetermined in relation to a timestamp limit of which the loitering event is allowed to occur.

8. A method as claimed in any one of Claims 5 to 7 wherein the motion detector (10) is a background estimation sub-component (10); and the step of delineating at least one motion pixel comprises the steps of processing an input image, delineating the motion pixels from the background in the image sequence (50), and grouping together the motion pixels with similar properties as motion blobs to allow the blobs to be emphasized corresponding to the object-

of-interest and the noise to be eliminated; and the step of delineating at least one motion pixel further comprises the step of using an object classification method for discriminating human and non-human blobs dependently on the functionality of the system (100).

9. A method as claimed in Claim 8 wherein the object detector (20) is a new object detection sub-component (20); and the step of detecting at least one object comprises the steps of performing a blob-overlapping between the previous and current frames to identify the relationship between the previous and the current frames; performing a similarity matching between the blobs of which features including color, area, directions, and other features that are allowed to be used, generating a relationship table to represent the association of each blob in the current frame with the corresponding blob from the previous frame; performing a region-of-interest masking on each blob for validation; and storing the blobs that are detected as the new object into a respective ROI's (60) new object database (190).

10. A method as claimed in Claim 9 wherein the threshold comprises a predetermined potential loiterer threshold and a predetermined true counter threshold; and the step of detecting the loitering event comprises the steps of:
registering and entering time of each detected new object into a tracker database (80);
analyzing the timestamp (70) of each object-of-interest in the tracker database (80);
performing the similarity and correlation searches between the object-of-interest that the timestamp (70) has exceeded a predetermined potential loiterer threshold and each blob in the scene;

identifying and storing the information of the potential loiterer whom similarity or correlation is found in the current scene into a potential loiterer database (90);
tracking each potential loiterer and analyzing the consistency of each tracking trail;
identifying and storing the information of the loiterer whom tracking consistency is more than the predetermined true counter threshold into a loiterer database (110);
tracking the loiterer and analyzing the consistency of each tracking trial,

accumulating the timestamp (70) of each object-of-interest in the each database (80, 90, 110) and estimating the accumulated timestamp (70) of each object-of-interest in the tracker, the potential loiterer, and the loiterer databases (80, 90, 110); and

performing at least one removal condition to each object-of-interest in each database (80, 90, 110) and an invalid object from the database (80, 90, 110).

11. A method as claimed in Claim 10 wherein the threshold further comprises a predetermined timestamp threshold; and the step of registering and entering time comprises the steps of storing the properties and the current time into the tracker database of the respective ROI (60), wherein the extracted properties comprise color properties, blob sizes, speed, and velocity; increasing the time of appearance of each object-of-interest in the scene by a timer at each frame; adding the timer associated to each blob to the time difference between the current and the previous frames; and accumulating the timestamp (70) of each object at each image sequence (50); the step of analyzing the timestamp comprises the step of performing a locality prediction analysis on the object-of-interest if the timestamp (70) of the object-of-interest exceeds the predetermined timestamp threshold; and the step of performing the similarity and correlation searches comprises the step of randomly distributing a plurality of particles around all motion blobs detected within the respective ROI (60) to search for similar object properties; and assuming that there is a possibility that the loiterer has not left the scene.

12. A method as claimed in Claim 11 wherein the threshold comprises a predetermined tracking confidence threshold; in the potential loiterer tracking unit (140); the step of tracking each potential loiterer comprises the steps of tracking each potential loiterer using a refined particle filter approach; developing an approximation of the probability density function (pdf), wherein a measure of confidence level of a prediction is obtainable from the pdf; checking a tracking confidence for each prediction against the predetermined tracking confidence threshold to allow identification of the object that no longer appears in the scene; increasing a true counter of the object by one and updating the locality of the object into the

potential loiterer database (90) using the resultant prediction from the particle filter tracking if the prediction exceeds the predetermined tracking confidence threshold; and resetting a false counter if the prediction does not exceed the predetermined tracking confidence threshold.

13. A method as claimed in Claim 12 wherein the step of identifying and storing the information of the loiterer comprises the steps of storing the information of the potential loiterer whom its similarity or correlation is found in the current scene into the potential loiterer database (90); performing a confidence analysis for each potential loiterer, wherein the confidence analysis is done based on the true counter associated to each potential loiterer; and confirming that the potential loiterer is exhibiting loitering behavior if the true counter exceeds the predetermined true counter threshold; and the step of tracking the loiterer comprises the steps of tracking each loiterer as defined in the loiterer database (110) using the particle filter tracking; approximating the pdf in order to ably represent the similarity matching; checking the tracking confidence for each prediction against the predetermined tracking confidence threshold to allow identification and removal of the loiterer that has left the scene; and increasing the false counter by one if no similar object properties are found.

14. A method as claimed in Claim 13 wherein the step of accumulating the timestamp (70) comprises the steps of adding the timer associated to each blob to the time difference between the current and the previous frames; and incrementing the timer information of each object from its first appearance in the scene until the object has been confirmed to have left the scene using a plurality of coherent rules; the threshold further comprises a empirically defined or predefined according to the parameters predetermined removal threshold; and the step of performing at least one removal condition comprises the steps of removing the potential loiterer or loiterer from the database (80, 90, 110) once at least one of the removal condition is met, wherein the removal condition is the false counter associated to each potential loiterer in the potential loiterer database (90) and the false counter is associated to each loiterer in the loiterer database (110), and the removal condition assumes that the particle filter tracking estimation is of lower confidence consistently throughout a number of frames if the object is

no longer in the scene; and removing the potential loiterer or loiterer from the database if the associated false counter exceeds the predetermined removal threshold.

15. A method as claimed in Claim 14 wherein the step of randomly distributing a plurality of particles comprises the step of performing the similarity and correlation searches by distributing particles randomly around all detected motion pixels and calculating the similarity value between the particle and object-of-interest.

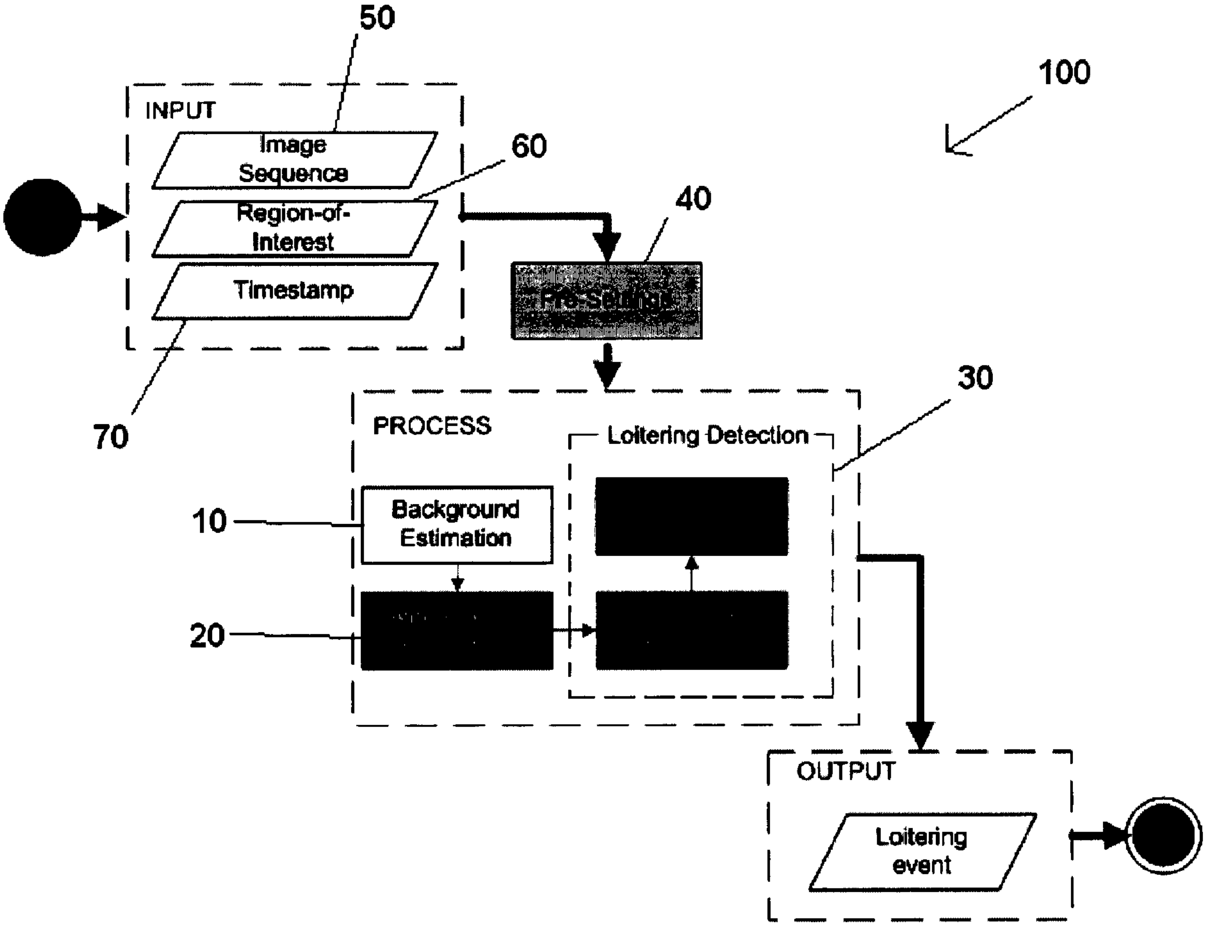

50
100
INPUT
Image Sequence
60
Region-of-Interest
40
Timestamp
Pre-Settings
70
30
PROCESS
Loitering Detection
10
Background Estimation
20
OUTPUT
Loitering event

FIG. 1

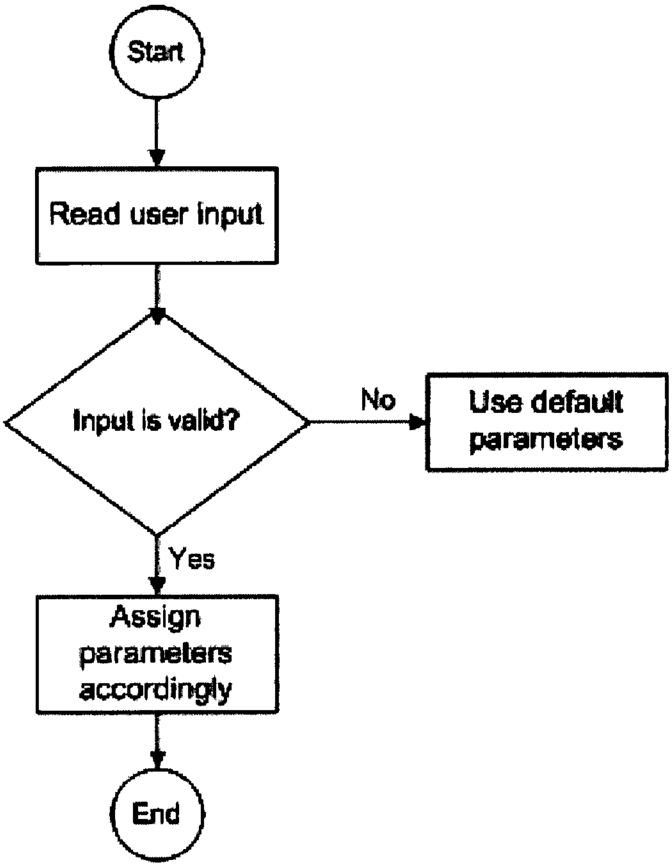

Start
Read user input
Input is valid?
No
Use default parameters
Yes
Assign parameters accordingly
End

FIG. 2

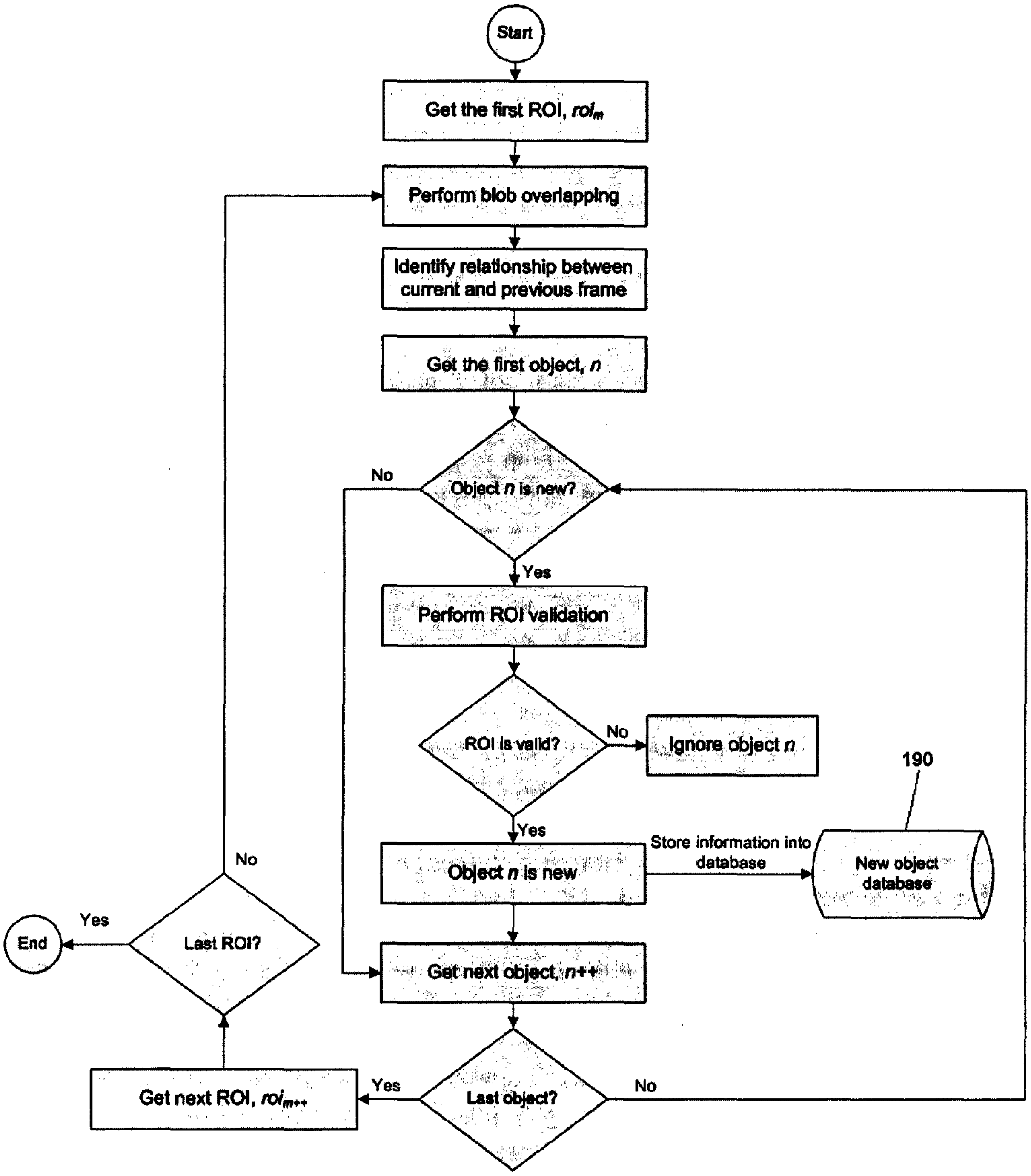

Start
Get the first ROI, roi_m
Perform blob overlapping
Identify relationship between current and previous frame
Get the first object, n
Object n is new?
No
Yes
Perform ROI validation
ROI is valid?
No
Ignore object n
190
Yes
Object n is new
Store information into database
New object database
Get next object, n++
Last object?
Yes
No
Get next ROI, roi_{m++}
Last ROI?
No
Yes
End

FIG. 3

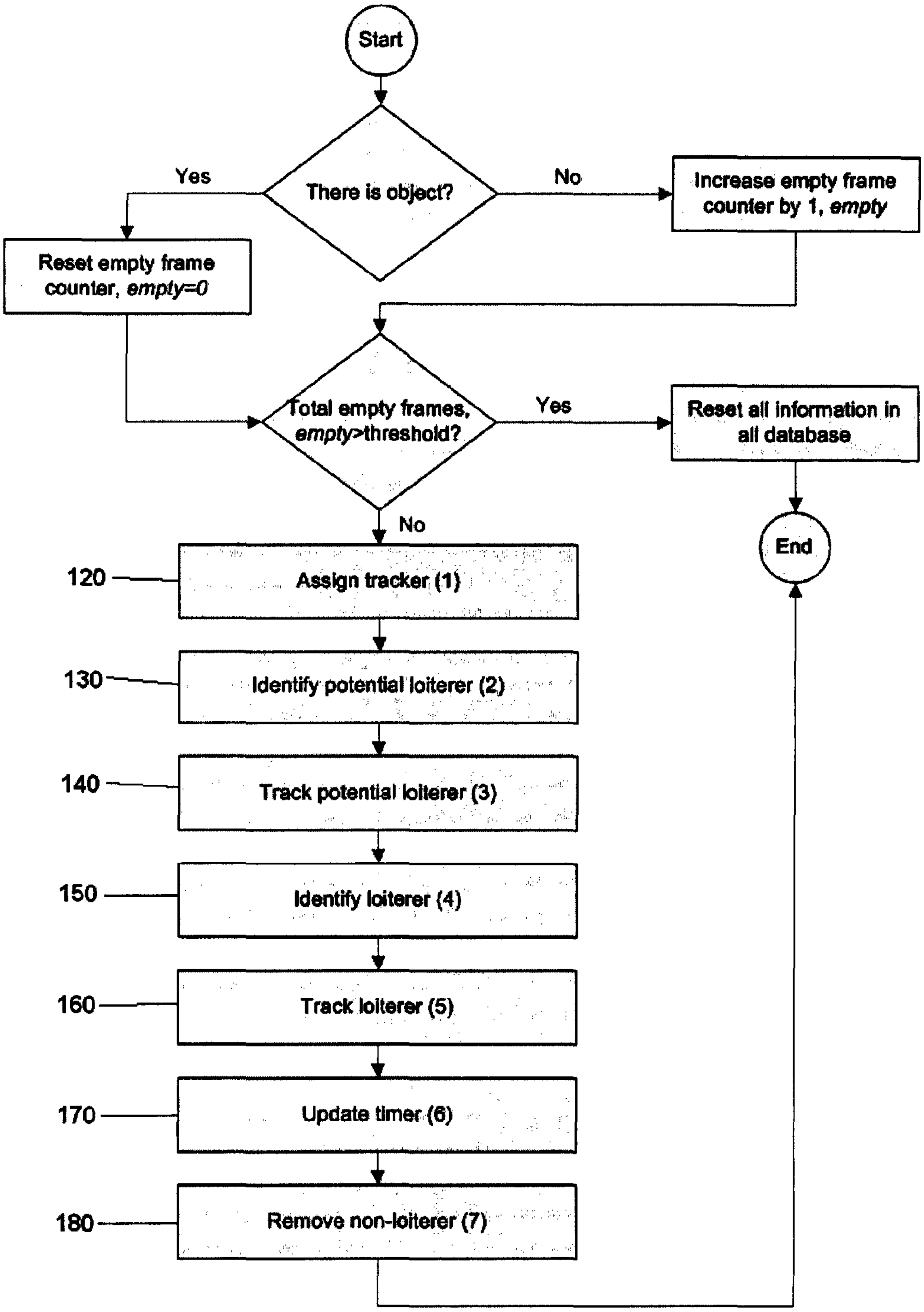

Start
There is object?
Yes
No
Reset empty frame counter, *empty=0*
Increase empty frame counter by 1, *empty*
Total empty frames, *empty*>threshold?
Yes
Reset all information in all database
End
No
120
Assign tracker (1)
130
Identify potential loiterer (2)
140
Track potential loiterer (3)
150
Identify loiterer (4)
160
Track loiterer (5)
170
Update timer (6)
180
Remove non-loiterer (7)

FIG. 4

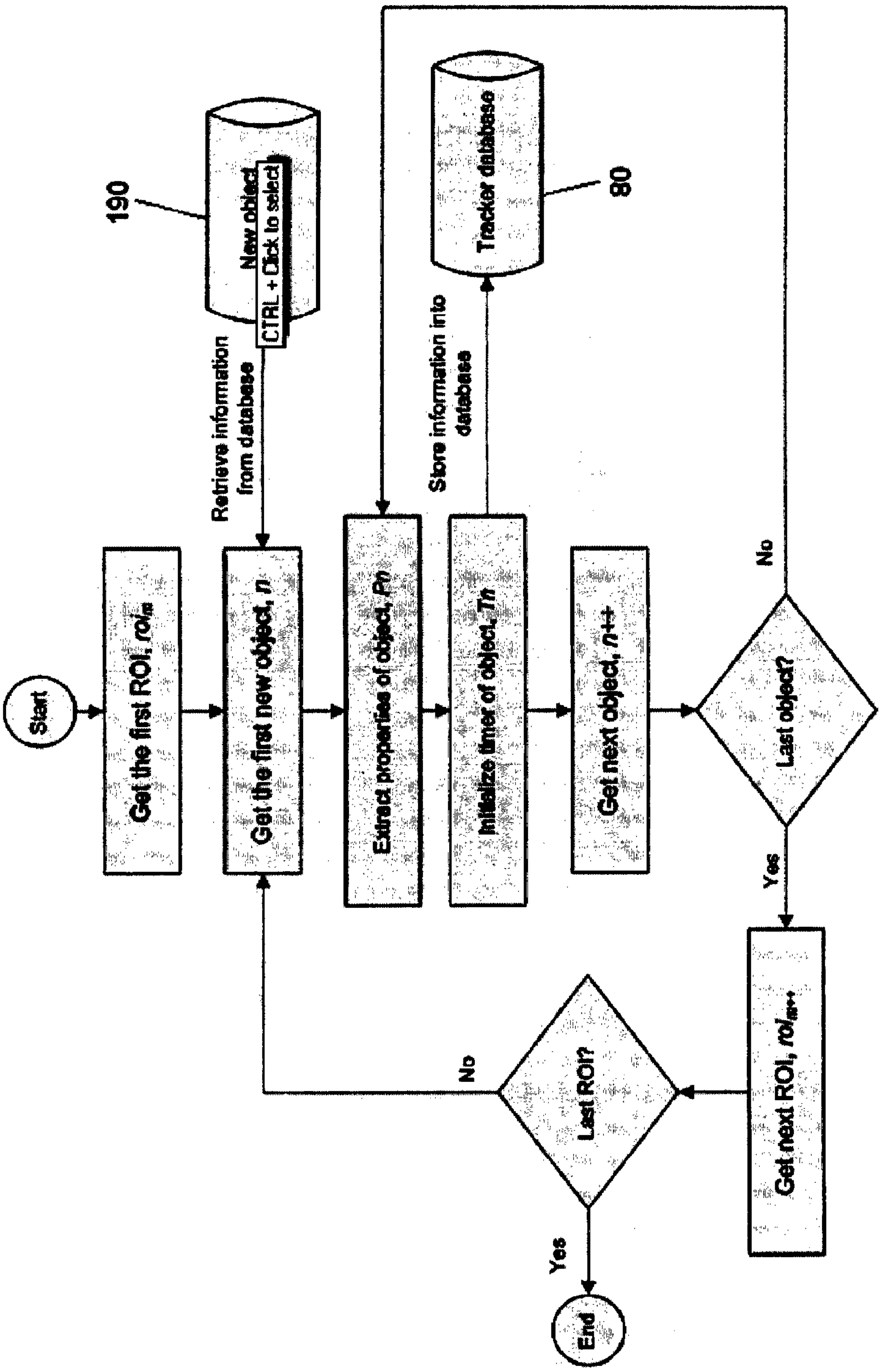

Start
Get the first ROI, roi_m
Get the first new object, n
Retrieve information from database
190
New object
CTRL + Click to select
Extract properties of object, Pn
Initialize timer of object, Tn
Store information into database
Tracker database
80
Get next object, n++
Last object?
No
Yes
Get next ROI, roi_m++
Last ROI?
No
Yes
End

FIG. 5

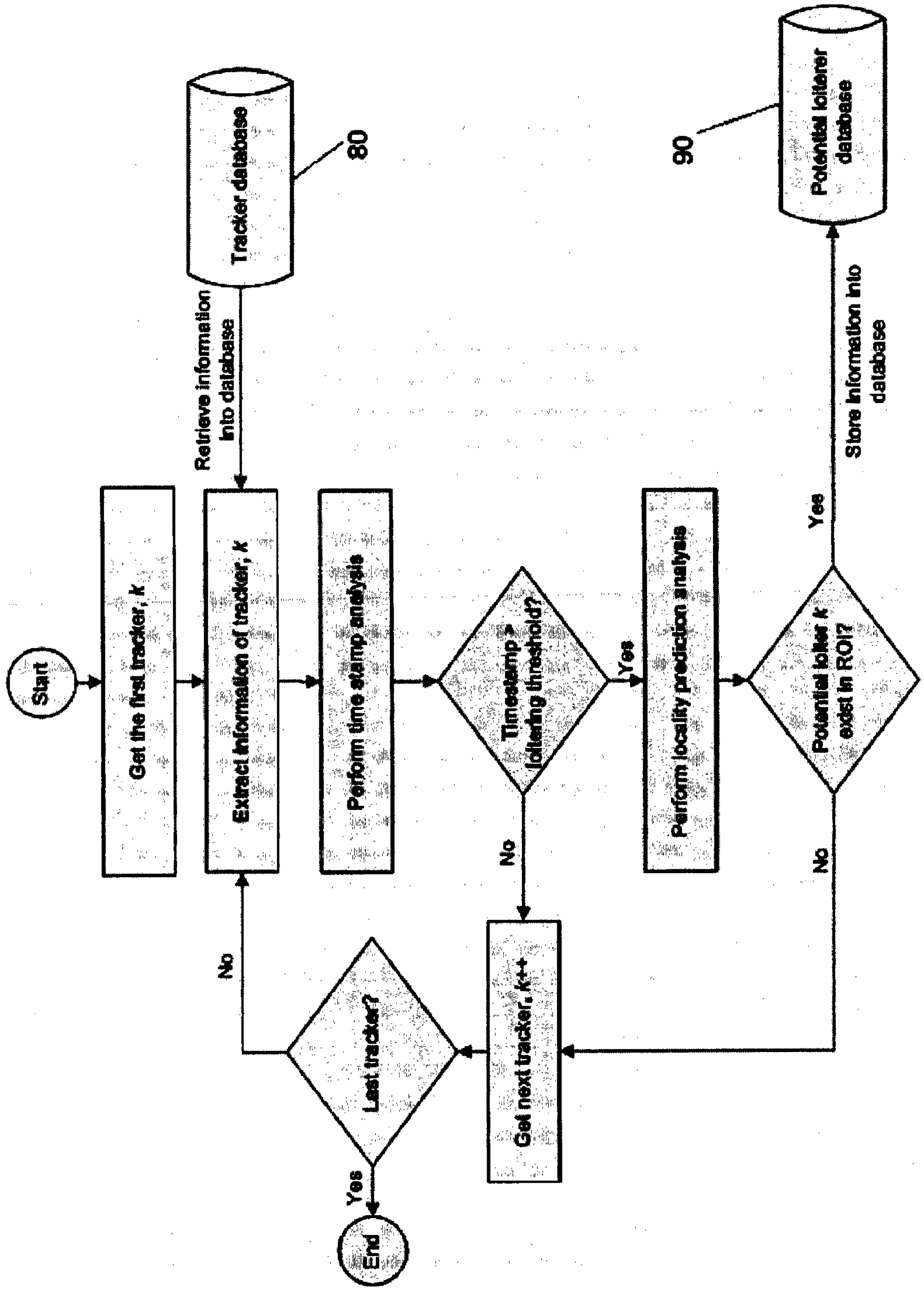

Start
Get the first tracker, k
Retrieve information into database
Tracker database
80
Extract information of tracker, k
Perform time stamp analysis
Timestamp > loitering threshold?
Yes
No
Perform locality prediction analysis
Potential loiter k exist in ROI?
Yes
No
Store information into database
Potential loiterer database
90
Get next tracker, k++
Last tracker?
No
Yes
End

FIG. 6

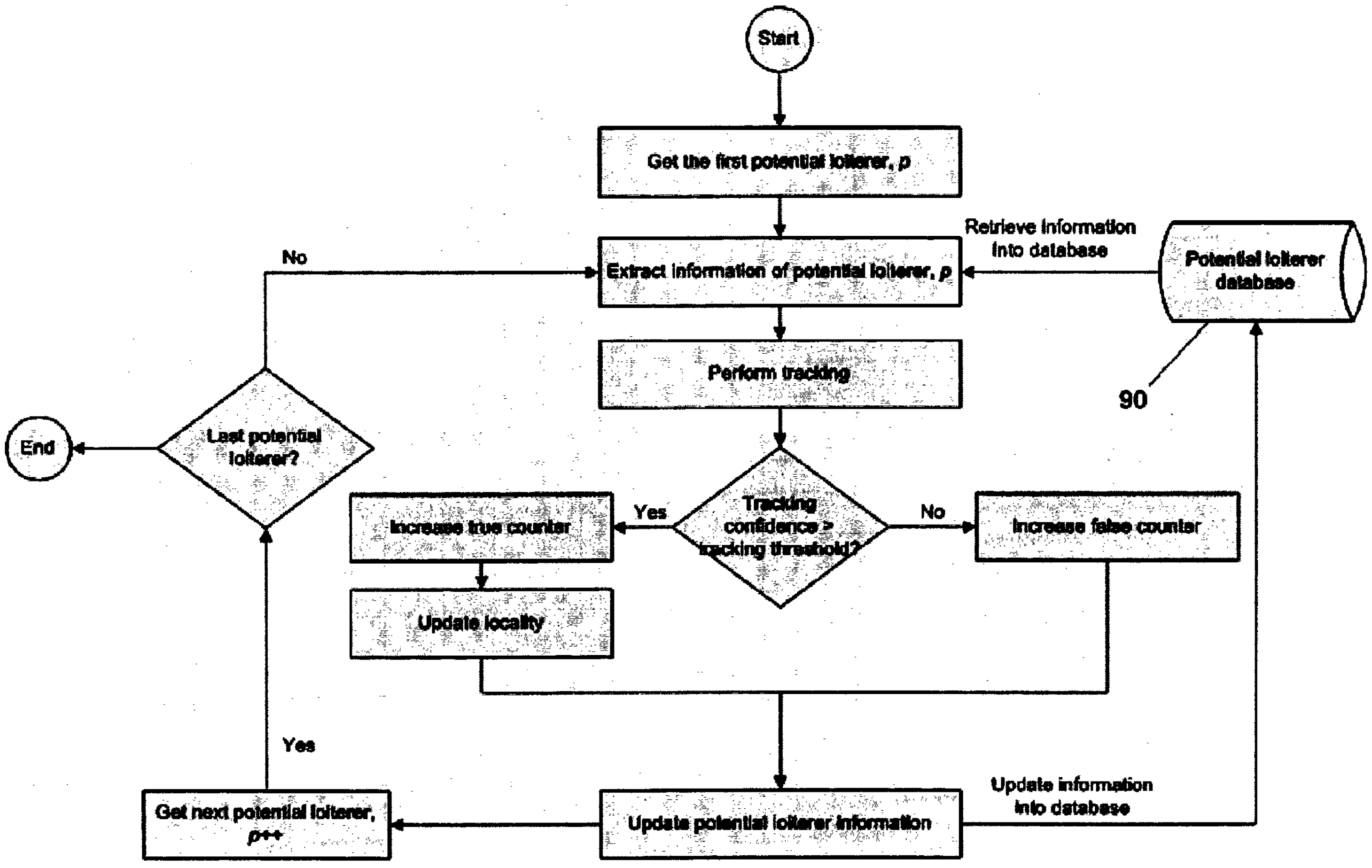

Start
Get the first potential loiterer, p
Extract information of potential loiterer, p
Retrieve information into database
Potential loiterer database
90
Perform tracking
Tracking confidence > tracking threshold?
Yes
No
Increase true counter
Update locality
Increase false counter
Update potential loiterer information
Update information into database
Get next potential loiterer, p++
Yes
Last potential loiterer?
No
End

FIG. 7

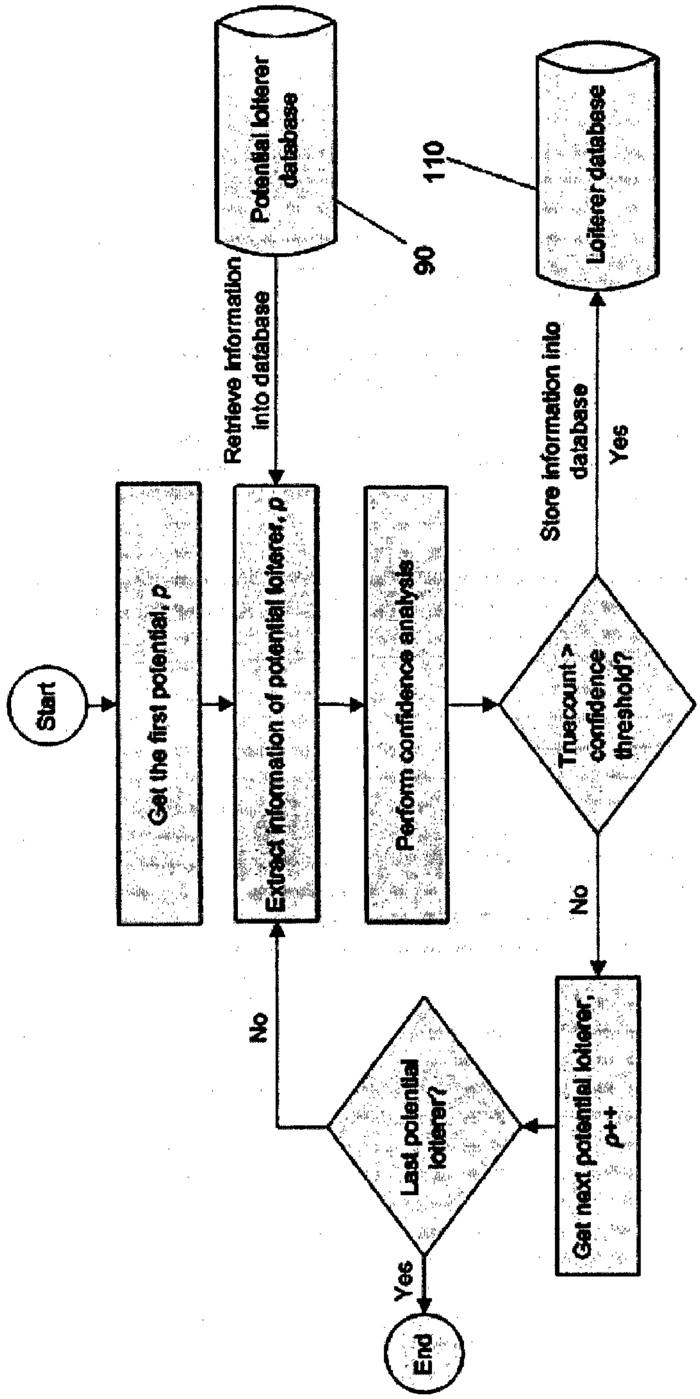

Start
Get the first potential, p
Retrieve information into database
Potential loiterer database
90
Extract information of potential loiterer, p
Perform confidence analysis
Truecount > confidence threshold?
Yes
Store information into database
110
Loiterer database
No
Get next potential loiterer, p++
Last potential loiterer?
No
Yes
End

FIG. 8

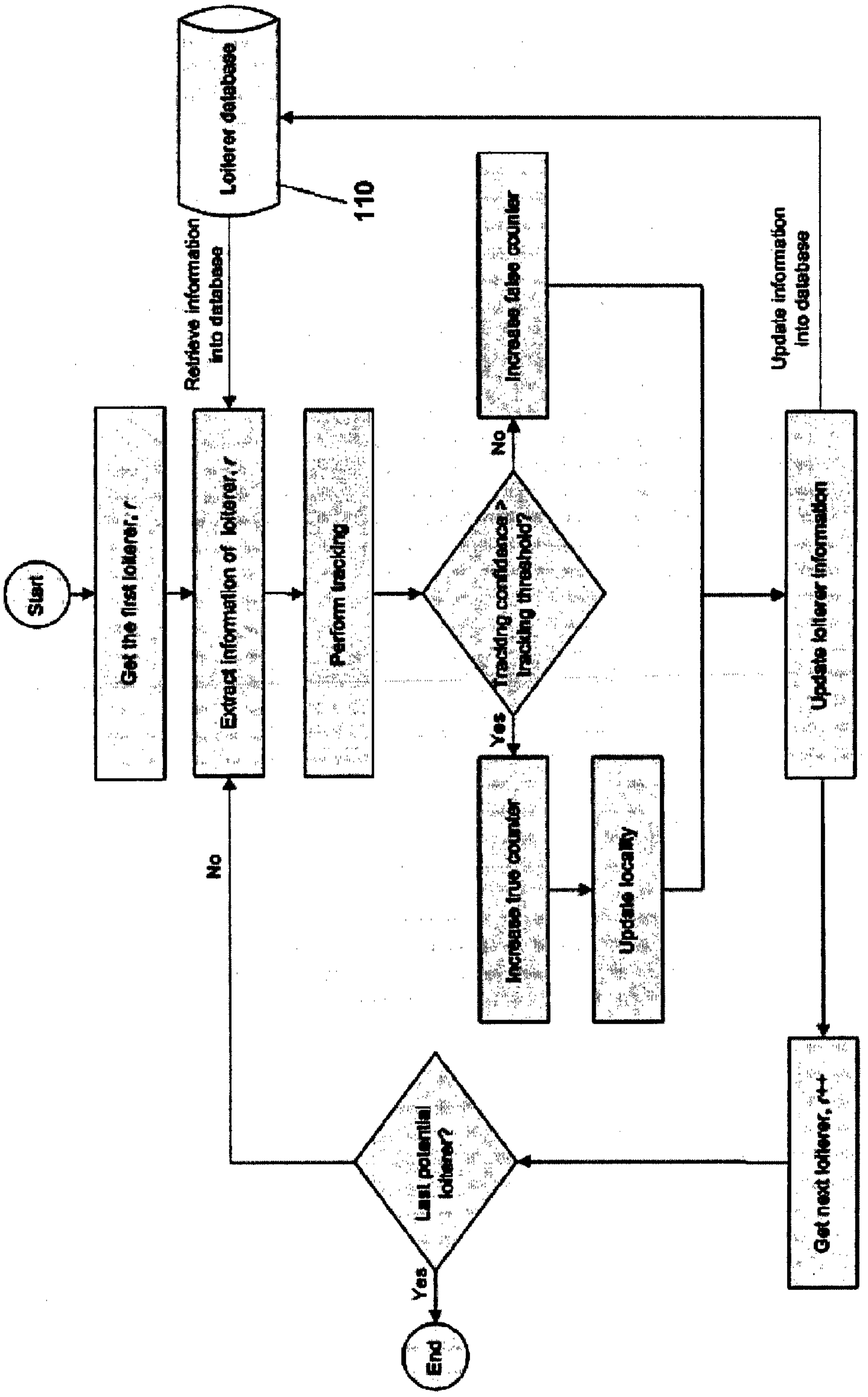

Start
Get the first loiterer, r
Retrieve information into database
Loiterer database
110
Extract information of loiterer, r
Perform tracking
Tracking confidence > tracking threshold?
Yes
No
Increase true counter
Update locality
Increase false counter
Update loiterer information
Update information into database
Get next loiterer, r++
Last potential loiterer?
No
Yes
End

FIG. 9

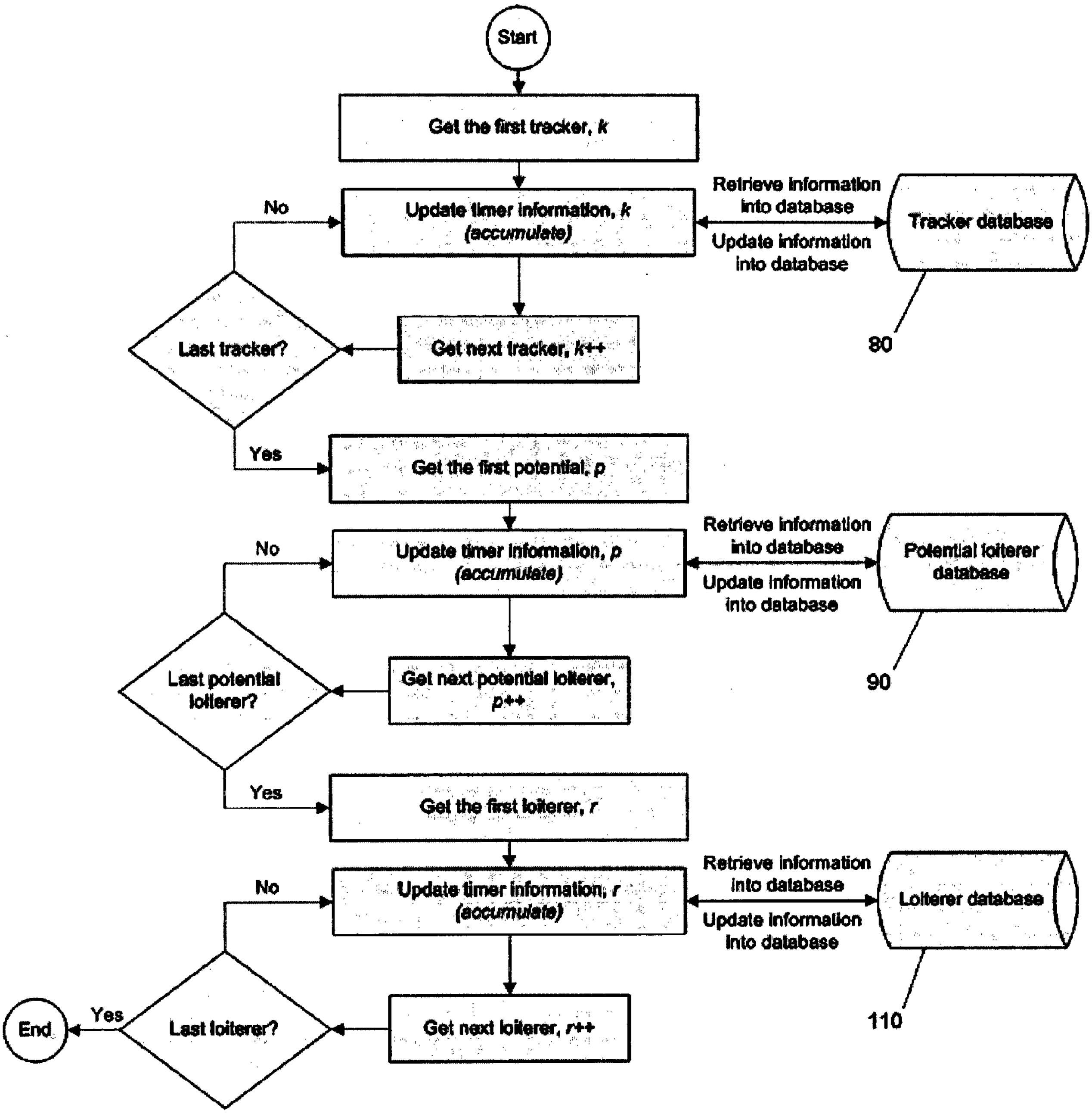
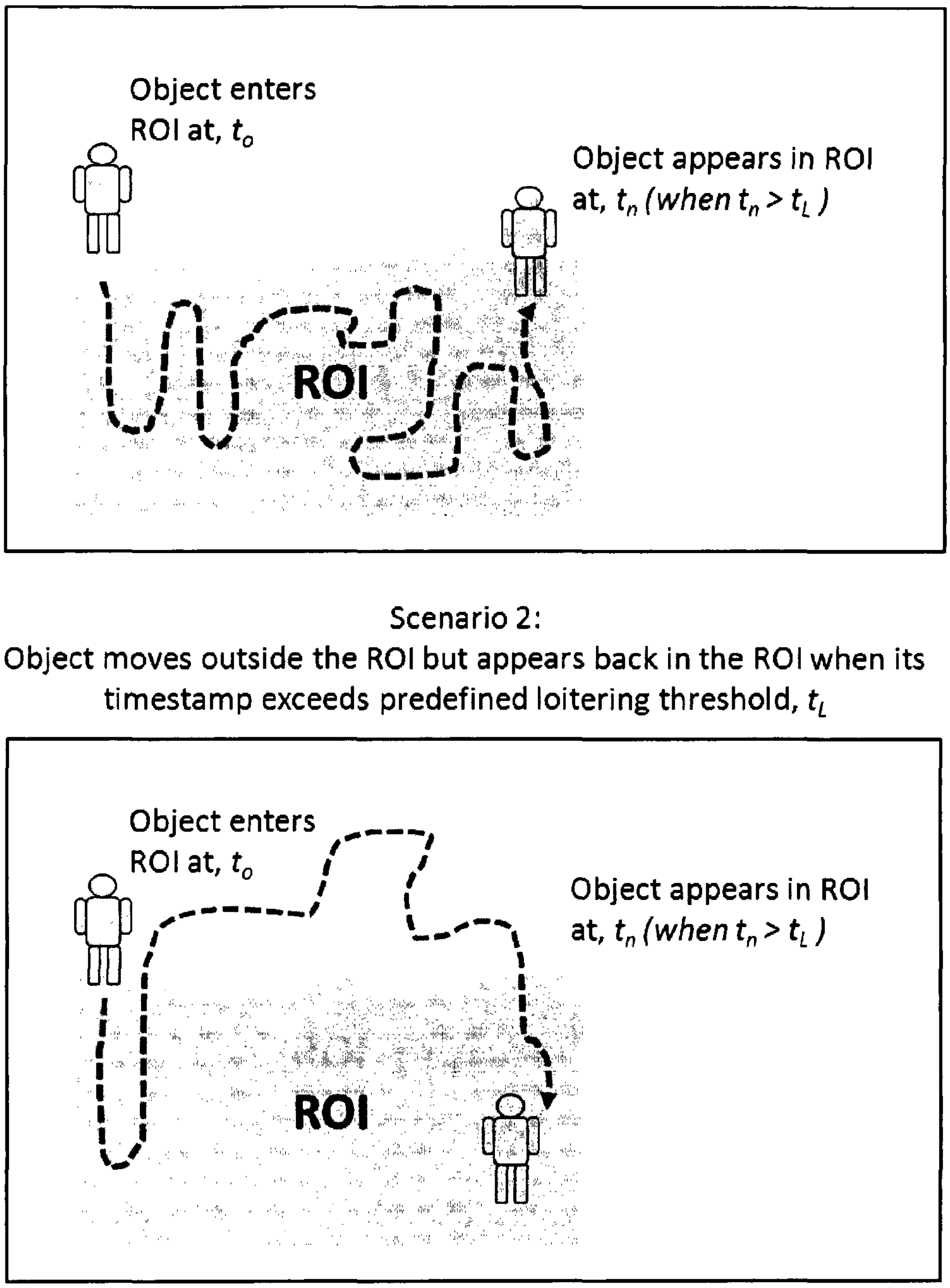

Start
Get the first tracker, k
Update timer information, k
(accumulate)
Retrieve information into database
Update information into database
Tracker database
80
No
Last tracker?
Get next tracker, k++
Yes
Get the first potential, p
Update timer information, p
(accumulate)
Retrieve information into database
Update information into database
Potential loiterer database
90
No
Last potential loiterer?
Get next potential loiterer, p++
Yes
Get the first loiterer, r
Update timer information, r
(accumulate)
Retrieve information into database
Update information into database
Loiterer database
110
No
End
Yes
Last loiterer?
Get next loiterer, r++

FIG. 10



Scenario 1:
Object moves within the ROI and exceeds predefined loitering threshold, t_L

Object enters ROI at, t_o
Object appears in ROI at, t_n *(when* $t_n > t_L$ *)*
ROI

Scenario 2:
Object moves outside the ROI but appears back in the ROI when its timestamp exceeds predefined loitering threshold, t_L

Object enters ROI at, t_o
Object appears in ROI at, t_n *(when* $t_n > t_L$ *)*
ROI

FIG. 11

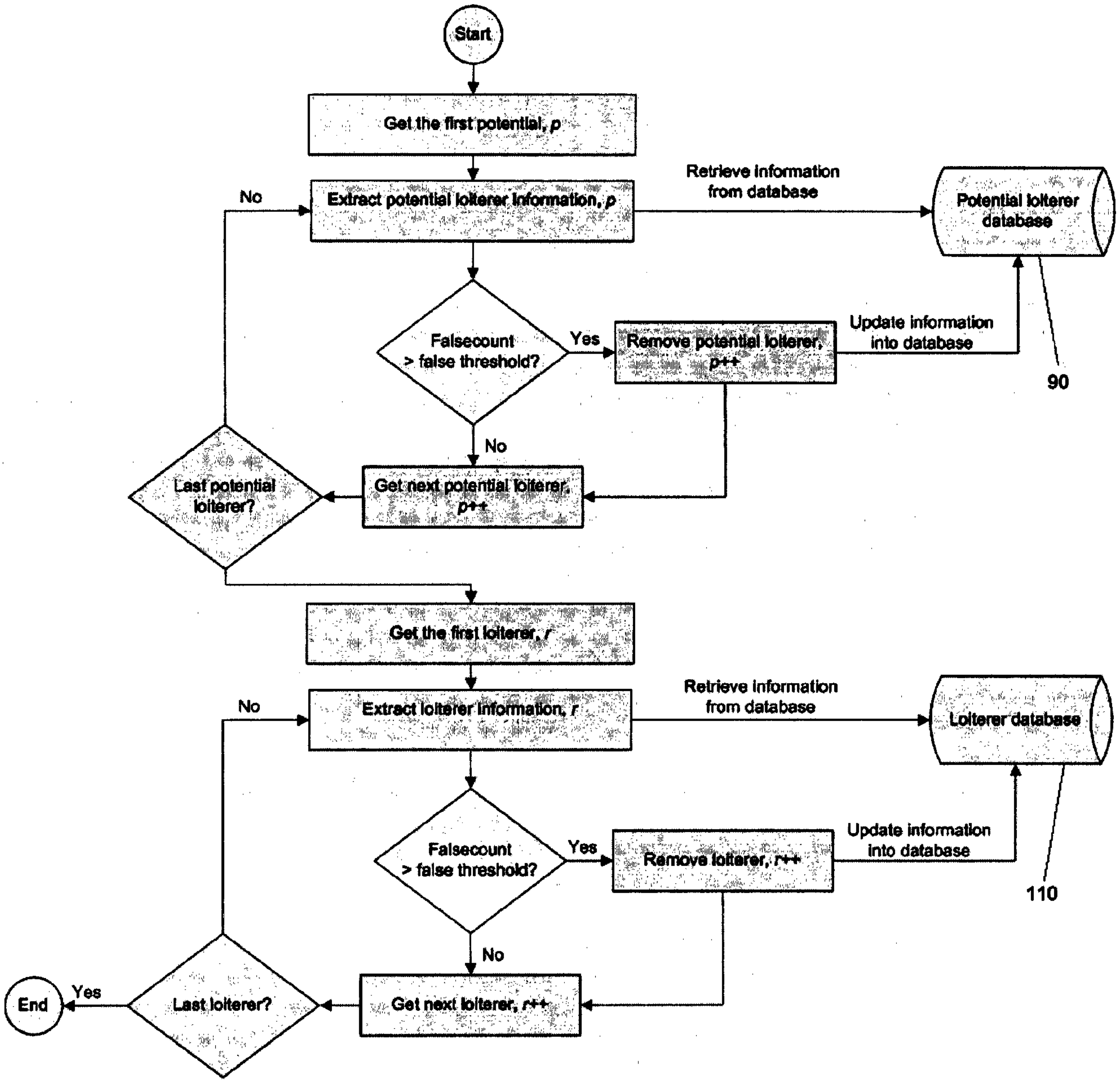

Start
Get the first potential, p
Extract potential loiterer information, p
Retrieve information from database
Potential loiterer database
No
Falsecount > false threshold?
Yes
Remove potential loiterer, p++
Update information into database
90
No
Last potential loiterer?
Get next potential loiterer, p++
Get the first loiterer, r
Extract loiterer information, r
Retrieve information from database
Loiterer database
No
Falsecount > false threshold?
Yes
Remove loiterer, r++
Update information into database
110
No
End
Yes
Last loiterer?
Get next loiterer, r++

FIG. 12