CrossMark

Trajectory based database management for intelligent surveillance system with heterogeneous sensors

Jung-Min Oh[1] · Nammee Moon[2] · Sangjin Hong[1]

Abstract In this paper, we present a database management scheme for an intelligent surveillance system utilizing multiple visual sensors and RFID readers. The objects are tracked and identified by multiple visual sensors and RFID readers. We define three different types of data structure to consistently store data for effective data storage. They contain global object number and identification as the common information of the same object. The global object number is uniquely assigned for each track object. The previously stored data without the common information is back-annotated when it is available in the system. Moreover, when the global object number changes because of imperfect detection and tracking, the system maintains consistency information between global object numbers for the same object by comparing their local target information or positions. The fragmented information for an object is also stitched through map information. The simulation results demonstrate that the database information for objects is successfully recovered with consistency.

Keywords Intelligent surveillance · Consistency management · Iterative decision process · Heterogeneous sensor network · Database

1 Introduction

Recently, many researchers have much interest in an intelligent surveillance system. An intelligent surveillance system consists of many different processes such as object detection, object tracking, object classification, object identification, information management and information visualization [8, 14, 17, 18]. In these surveillance systems, many sensors are deployed to collaboratively process information [1, 2]. Visual sensors are used to track the positions of

✉ Nammee Moon
   mnm@hoseo.edu

[1] Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794, USA

[2] Department of Computer Engineering, Hoseo University, Asan 336-795, South Korea

🍂 Springer

objects and identification sensors are used to uniquely identify the objects [3, 4, 6]. Also, visual sensors with PTZ functionality zoom in objects to obtain the high resolution information [5]. One of the key functionalities in an intelligent surveillance system is to search the information related to a surveillance event from the database [12, 16]. Information management and searching process have also been considered important approach for other research area, for example, data recommendation field [9, 11, 15].

However, the raw data that the surveillance system collects has a common problem of its incompleteness [13]. It is mainly due to sensor's imperfect detection capability [10, 20, 22]. In many surveillance systems, the object tracking is not trivial when the number of objects is large and densely populated. Because of the density of the object, uniquely identifying and associating the objects under heterogeneous sensor network environment is far from ideal. Often times, the object identifications are falsely associated. As result, a very limited true result is searched from the existing database [7, 19, 21]. The aggregated raw data must be conditioned properly for consistency. Moreover, the amount of aggregated data is significantly large so that accessing the database is very inefficient in terms of computation such that an effective information caching mechanism is necessary to reduce the information access time. In this paper, we present a database management scheme for intelligent surveillance system with heterogeneous sensors. In order to store the raw data, we define two different data types for local target information and global object information. When no tag is associated in a data, we infer the relevancy of the data among objects, which have existing data. As result, the data has probability of objects, which are related to the data. Similarly, when multiple tag IDs exist, we measure the probability. In order to minimize the information access time, we proposed to use subcommand process mechanism so that minimal set of data is needed to provide accurate trajectory information.

The remainder of this paper has 5 sections. In Section 2, we present the overview of the application model and the problem description. Section 3 present data store and management for an intelligent surveillance system with heterogeneous sensors. Section 4 presents information retrieval methodology with database consistency. Section 5 presents system evaluation and our contribution is summarized in Section 6.

2 System description and design overview

In the traditional surveillance system, human operators monitor objects by relying on the visual information through surveillance cameras and may simply record the video frames into storage systems. It may be sufficient to confirm suspects with the past information at specific time and place. However, the intelligent surveillance should be able to retrieve any accident automatically even with limited clue information. In order to support such capability, the surveillance systems must incorporate the database system with effective information management and retrieval. In order to maintain the database system, the application first requires the system modules, which gather sensory information, associates heterogeneous sensory information and creates/updates database information.

In this paper, we consider the surveillance system with heterogeneous sensors such as visual sensors and RFID sensors. While each visual sensor locally detects objects, the visual association module plays a role of tracking objects by associating objects local detection information from pairwise visual sensors. The RFID association module plays a role of

associating objects tracked by the visual association module with their RFID tag ID when they enter or leave the coverage of a RFID reader. RFID tag IDs are used for identifying objects tracked by visual sensors. The visual association module and the RFID association module create/update information in the database system. The active visual sensors with PTZ functionality operate to obtain the high resolution image of objects. The objective of this paper is to make a decision about people's type of actions related to surveillance accidents through the database system.

Figure 1 shows the proposed platform to handle user's information retrieval. As the surveillance system senses objects, it gives the sensed data to the raw data management. It structures the data in a format to be stored in the raw data database. In the meantime, information search requests from users take place.

The raw data in the surveillance system stores large amount of time and space information about objects. As objects continues to moves around, the raw data keeps growing toward the time and space dimension. Differently from the data in the traditional database, the data inherently has incompleteness. Due to sensor's detection errors, there exists the fragmented trajectory data in the raw data level. The group and miss-association problems of heterogeneous sensors cause an object to have multiple identification information or no identification information respectively in the raw data. The back-annotation mechanism is incorporated to correct such identification in the raw data. However, there is limitation to handle incomplete data in the raw data level, especially when an object is not sensed and no raw data exists.

In order to search decision result from the raw data, the query processing has to be structured. We define user's interesting decision as a query. To handle a query, we design a language for raw data access and decision search. For efficient data access, a complex query is decomposed into combination of simple queries. To stitch the incomplete data, the estimation/prediction using map information is incorporated. Based on the stitched data, the decision for objects is inferred. The estimation becomes impossible when there is no clue data at all. Thus, the decision is made through incremental data access.
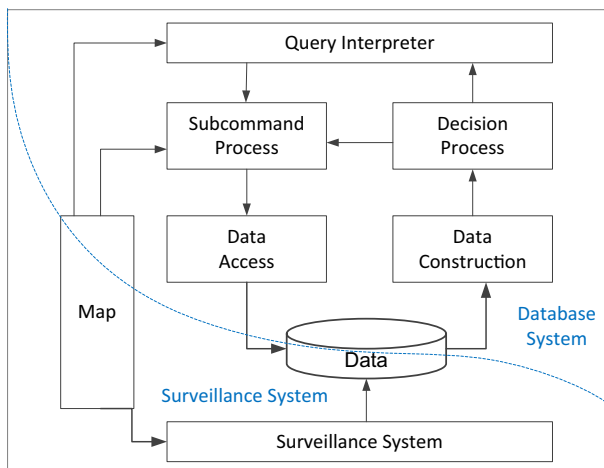

Fig. 1 The block diagram of the system model

3 Raw data management

3.1 Basic data store mechanism

Figure 2 illustrates the target surveillance system consists of visual sensors and RFID sensors. Each visual sensor locally detects an object within its field of view. An object becomes localized by two visual sensors through the association of their local detection information about the object. A RFID sensor detects object's enter or leave within its sensor coverage. In order to recognize if an object enters or leaves the sensor coverage through visual sensors, there is a homographic line approximating the coverage. When an object is found as crossing the line and is detected by a RFID sensor, the object is identified through its tag ID as well as localized.

Each visual association module consists of the set of visual sensors and each visual sensor detects objects in the surveillance region. The visual association module finds correspondences between detected targets and localizes objects. Every time new data for detected targets and objects are received in the visual association scheduler module from visual association modules, the data is stored by the predefined data structures shown in Fig. 3. The first data structure is created for each locally detected target. The location of the local target on the image is stored with detected time t, camera number C^l, local target number TN^l in the created data structure. The second data type is created for each global object. The global location is stored with the second associated camera time t, global object number ON^i, a pair of camera numbers C^l and C^m, a pair of local target numbers TN^l and TN^m. The third data type is created for each zoomed target by active tracking
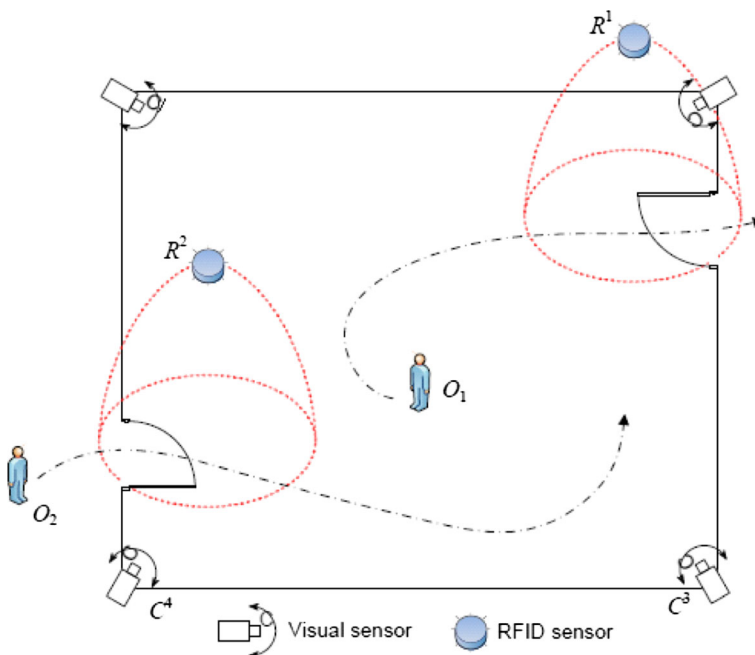

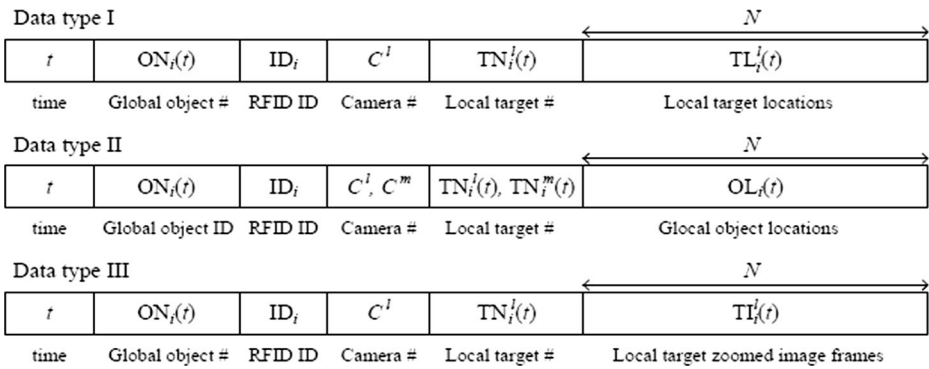**Fig. 2** Illustration of the surveillance system with heterogeneous sensors

Data type I					N
t	$ON_i(t)$	ID_i	C^l	$TN_i^l(t)$	$TL_i^l(t)$
time	Global object #	RFID ID	Camera #	Local target #	Local target locations

Data type II					N
t	$ON_i(t)$	ID_i	C^l, C^m	$TN_i^l(t), TN_i^m(t)$	$OL_i(t)$
time	Global object ID	RFID ID	Camera #	Local target #	Glocal object locations

Data type III					N
t	$ON_i(t)$	ID_i	C^l	$TN_i^l(t)$	$TI_i^l(t)$
time	Global object #	RFID ID	Camera #	Local target #	Local target zoomed image frames

Fig. 3 Illustration of three different data types in the database

module. Once the visual sensor begins the zooming process, the data structure of the third data type is created. The zoomed image frame is stored with the detected time t, camera number C^l, local target number TN^l in the created data structure.

3.2 Back-annotation method

When an object moves around the surveillance region, each visual sensor detects the object. The system begins storing the local target coordinate in the data structure of the first type. The local target number does not change while the target is continuously tracked by the system. If the system fails to locally track the target, the system considers it as a new target and a new local target number is assigned to the detected target. Because the visual association scheduler does not receive the data for the old local target number, the data structure for the old local target number is transferred in the database and the data structure for the new local target number is created with the new detection time to store the local target coordinate. When the object enters the overlapped field of views by more than two cameras, the visual association scheduler also receives the data for the global object. Then, the global object number is filled in the data structure of the first data type for local targets and the data structure of the second data type for the global object is created to store the global coordinate. However, the visual association scheduler updates the field of the global object number for the data that already transferred in the database.

In order to update the global object number for the old data in the database, the visual association scheduler creates a back-annotation event whenever it receives new global objects. A back-annotation event for the global object number is defined with its local target number. Once the back-annotation event is received in the back-annotation event handler module, the back-annotation event handler module checks if the database is not written by any other modules. If the database is available for the write access, the back-annotation event handler module updates the global object number by searching the data structure having its local target numbers.

When the surveillance system has RFID readers for object identification, the system has another back-annotation event to fill in the identification in the database. The identification of the global object is updated in the global information by the RFID association event handler module and the visual association scheduler module uses the identification to fill in the data structure. However, before the identification is updated in the global information table, some

data without identification can be transferred in the database because of the maximum size of the data. In order to have the complete information in the database, the RFID association event handler module updates the global information table for the identification as well as creates a back-annotation event for the identification. A back-annotation event for the identification is defined with the global object number and the corresponding identification. The back-annotation event handler module searches the data structure with the global object number and fills in the identification in the database.

Figure 4 illustrates the process of back annotation mechanism. When the object enters the coverage of RFID reader R1. It is assumed that the RFID association event handler module associates global object number ON_1 (t_{125}) with its identification ID_1 and updates the global information table at time t_{250}. The visual association scheduler module reads ID_1 for ON_1 (t_{125}) in the global information table and fills in ID_1 in the data structure. At the same time, the RFID association event handler module creates a back-annotation event with ON_1 (t_{125}) and ID_1. The back- annotation event handler module searches the old data without identification by ON_1 (t_{125}) and fills in ID_1 in the database.
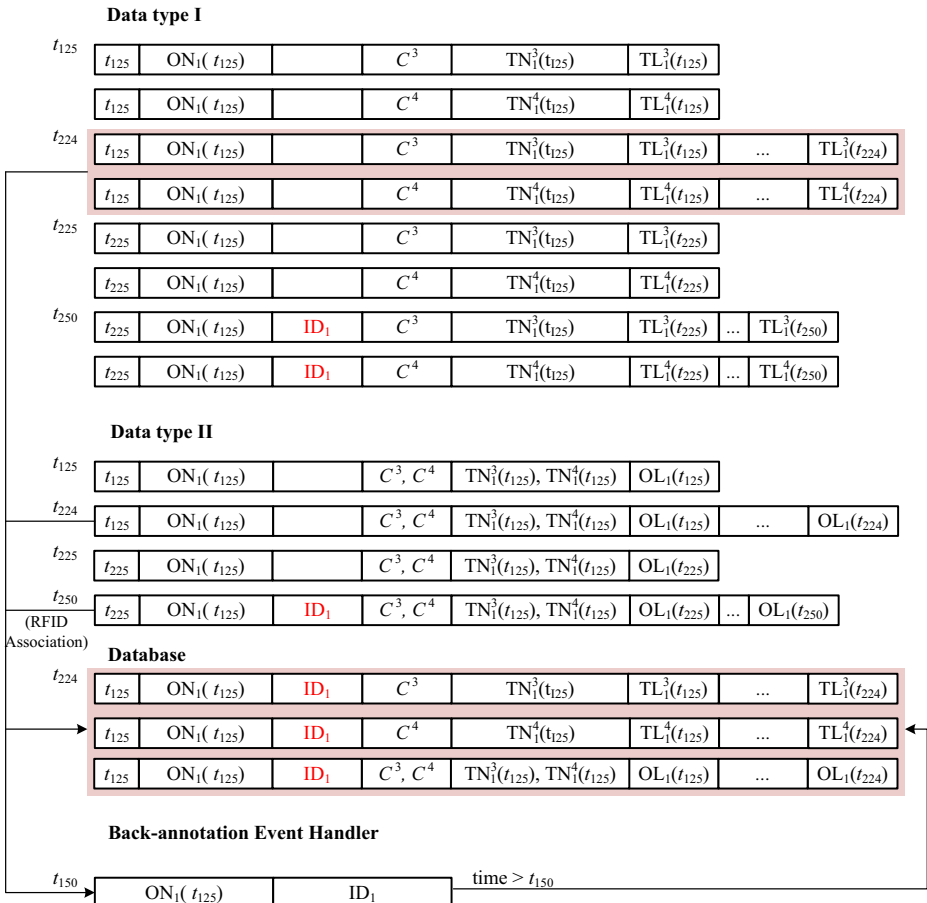

Fig. 4 Illustration of the back-annotation for single association with identification

3.3 Low level inconsistency database handling

We consider two non-ideal sources to manage the database. One is detection failure and another is local tracking failure. Both the failures cause the change of the global object number for the same object. Although the targets may be detected by two cameras at the next time, they are considered to be new targets and the new local target numbers are assigned to them. Also, the global object generated by them is considered to be a new global object with a new global object number. The local tracking failure means that targets are continuously detected but they are not successfully associated with previously detected targets. Because they are considered to be new targets, they have new local target numbers. The global object generated by them also has a new global object number. Because the system executes the back-annotation operation with the global object number in the back-annotation event, it does not fill in the identification in the data with different global object numbers. Then, when the system tries to search the object trajectory with the identification, only partial object trajectory can be found.

The system keeps the global object information to consistently maintain the database against the detection and local tracking failures. The global object information is updated by the visual association scheduler every sampling time. If the global object disappears due to the local tracking failure in one camera or the detection failure in one camera, they have the same local target number. Hence, the system compares the local target numbers of the disappeared global object with the local target numbers of each candidate global object. If the same local target number is found, the system keeps the list of the previous global object number and the global object number with the same local target number. On the other hand, when the global object disappears because of the local tracking failure in two cameras, the system cannot find the changed global object number with the local target number. In this case, the system checks the distance between the locations of the disappeared global objects with the location of each candidate global objects. When only one distance is within the estimated distance by the constant velocity, the candidate global object is considered to be the disappeared global object. For the detection failures in two cameras, the system cannot link the disappeared global object to one of any current global objects. Thus, the system uses the estimation window to find the link between the disappeared global object and the current global object.

4 Query handling and database access

4.1 Operation overview and query characterization

A query is a high-level representation to be used for obtaining the information that one wants to know about objects from the database. Figure 5 illustrates the subcommand-based query processing for simple query and compound query. A subcommand retrieves and constructs data necessary for deciding a query answer. Given a query, it is decomposed into a sequence of subcommands in the first pass. The query answer may be searched in the data prepared by those commands only. We consider such query as a simple query. On the other hand, if a query iteratively requires subcommands, we consider it as a complex query.

The entire query processing operates through subcommands. A query interpreter creates a sequence of subcommands by using map information for a given query. It also decides if it is necessary to make additional subcommands, especially for a complex query, when it receives a query answer from the decision process. A subcommand process receives a subcommand the
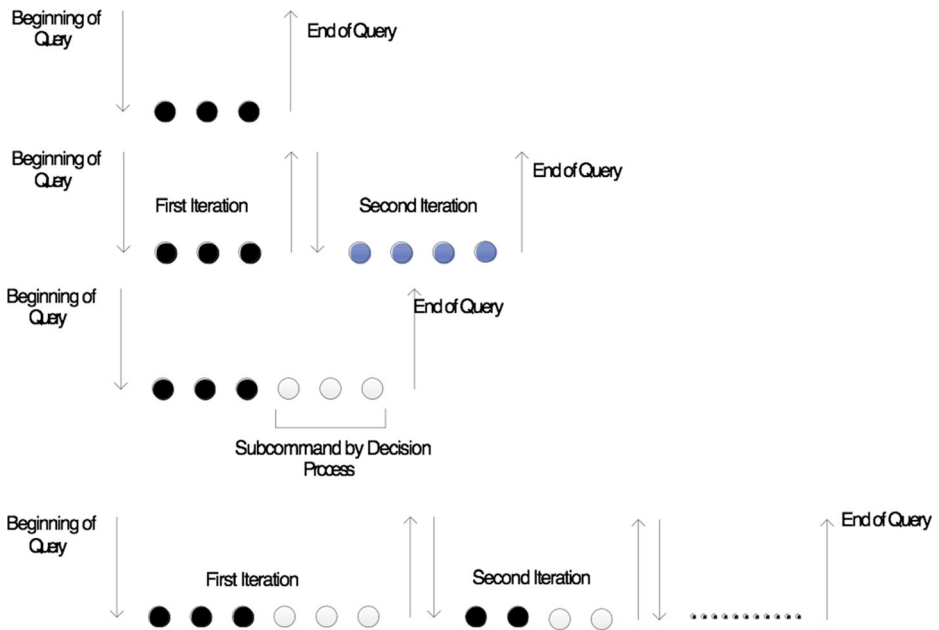
Fig. 5 Illustration of subcommand-based query processing for simple query and compound query

decision process as well as the query interpreter. Then, the subcommand process gives a set of commands to the data access. A command enables data reading from the database based on the minimum resolution. Whenever the data access process reads raw data, the data construction process formats it in the structure defined for easily getting a query answer. Since it is time-consuming, they share the state information about the previously collected data, to avoid redundant data access and construction.

4.2 Decision process and interpreter interaction

The decision process operates in the following steps. Suppose that the subcommand process constructs the data through a list of subcommands from the interpreter. The decision process firstly checks if the data necessary for decision exists in the constructed data. Then, it performs the data searching based on subcommands from the interpreter and returns the search results to the interpreter. Before searching the data, it is necessary to check whether or not the necessary data for searching is indeed in the constructed data. Suppose that the data is constructed for a query about checking if two objects A and B meet. Even though the data for two objects exists in the raw data, the data about object A is only stored in the retrieved data. Thus, the decision process is not able to obtain a proper search result.

The decision process stores a list of objects in the data table. It is for the interpreter to easily generate a final query answer. The table is created for a query. A list of objects obtained through subcommands for the query is stored in a table. For each object, not only single tag ID, but also multiple tag IDs and unknown tag ID are stored. Thus, the time range and place range of the table is determined by the time and place given in a query. The place range in the table is bounded by the map size. However, when a query has no specified time, the time range may infinitely increases. In order to practically maintain the table, the time range for the given query

is bounded to a certain range (e.g., a day) by the interpreter. The table is flushed by the interpreter, once it finds the final answer.

4.3 Iterative decision process

The interpreter mainly has three roles, to generate subcommands for data retrieval and to generate subcommands for decision (searching), and to generate combining methods of searched results to make a query answer. According to queries, the interpreter has different behaviors with respect to the above roles.

Given a query, it may produce a list of subcommands for data retrieval and decision at once. Then, it decides a final answer with all searched data. Another behavior is that an interpreter produces multiple subcommands but it may determine the query answer with search results from the first few subcommand. Then, it flushes the constructed data and stops the remaining subcommands. The third behavior is that an interpreter may generate subcommands for decision not at once, but iteratively. Thus, after analyzing the search result of a subcommand, it generates next subcommand. After all subcommands for decision, it determines a query answer like others.

When no place information is specified in a given query, the interpreter bounds the place by using the reference place. Specifically, the interpreter searches a list of logical places under the reference place through the map hierarchy. The searched places are in the level of spatial resolution of a subcommand. For each place, it formation is specified in a given query, the interpreter bounds the time by using the reference time range. It divides the maximum time range into smaller ones according to the time resolution of a subcommand. Then, it generates a list of subcommands for each time range. This mechanism is naturally applied to the case that a query has time information larger than the time resolution. A query may have no both time and place. Then, the interpreter generates a list of subcommands about the reference time range with respect to each place in the reference place range.

4.4 Handling insufficient data for decision

Since the data that the subcommand process retrieves is insufficient inherently, the interpreter may make an incorrect query answer. It is because the raw data in the database used as source data, stores incomplete information about objects due to the imperfect detection of sensors. As result, even though an object passed by a certain place indeed, the data misses such record. Figure 6 illustrates the information flow of the overall stitching process.

The decision process generates subcommands for a query, and updates the object list from the constructed data to the data table. From the data table, the interpreter makes a query answer. However, since the constructed data has incomplete information, the decision process tests if the data is insufficient or not, before getting the object list. When insufficient, it notifies the stitching process to estimate the missing data. Then, the stitching process stores the stitched data separately from the original constructed data. Practically, they are overlaid in the same format to allow the decision process to obtain the object list in the uniform manner. For stitching, the decision process may determine that it needs more data beyond the spatial and temporal range that a query specifies. Then, it gives subcommands to the subcommand process and the additional data is added to the constructed data. Based on the data, the decision process repeats the stitching process.
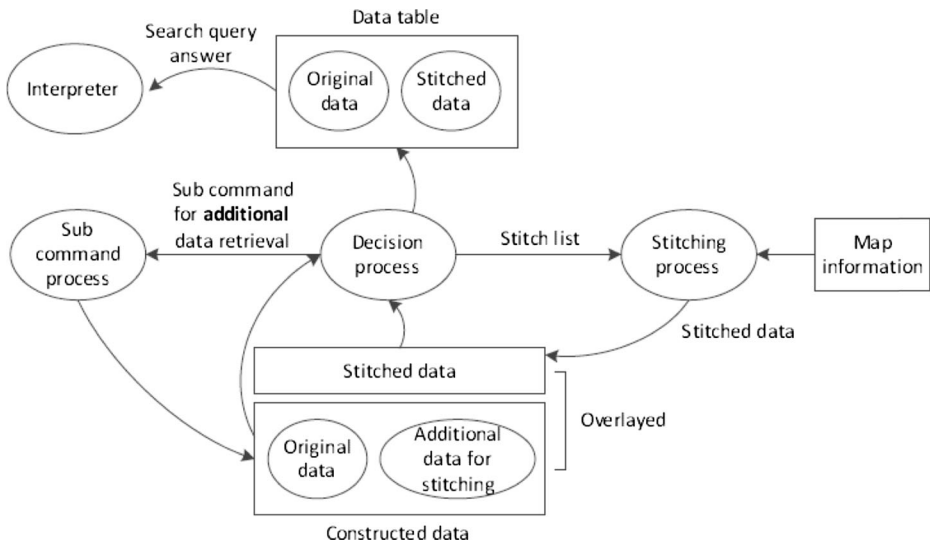
Fig. 6 Illustration of the information flow of the overall system for the stitching process

The subcommand for decision is extended to start the stitching process as well as to obtain object list from the constructed data. It is because the constructed data contains the missing data for objects. Thus, naturally, the decision process starts the stitching process before obtaining the object list. There are a variety of insufficient data cases according to query types. First, the constructed data may have two sequential existing paths where the missing data exists between them. If we generate probable paths from the end of a path to the end of another, the missing data is possibly estimated based on those paths. However, second, the constructed data may have an existing path such that it is not certain how to estimate. Thus, it is necessary to retrieve additional data, to search another path closest to the existing path temporally.

The basic missing data case is when two data units for an object are sequential, but no data exists between those data units for some time ranges. The decision process searches objects from the retrieved data in all existing place ranges in the order of the time range. This is called object scanning. At each time range, it stores place and time about the searched objects. Suppose that the data about a stored object is searched next.

If the difference between the stored time and the searched data is more than the time resolution, the decision process determines the missing data exists between the previous place and time and the currently searched place and time. Specifically, Fig. 7 illustrates the procedure that the decision process determines that the missing data exists between two regions. The decision process gives the path connectivity information including an object, the probable paths about the missing data and the time range surrounding the missing data, to the stitching process. The missing data between sequential data units may spread out the multiple places. In order to help the stitching process to generate the missing data, it is necessary to give the path connectivity information for the missing data. Thus, the decision process uses the region-connectivity to obtain a path from the place of the first data unit to the place of the second data unit. Along with the path, the decision process gives the time information of the first and second data units. This is for the stitching process to give the time information to each place of stitched data.
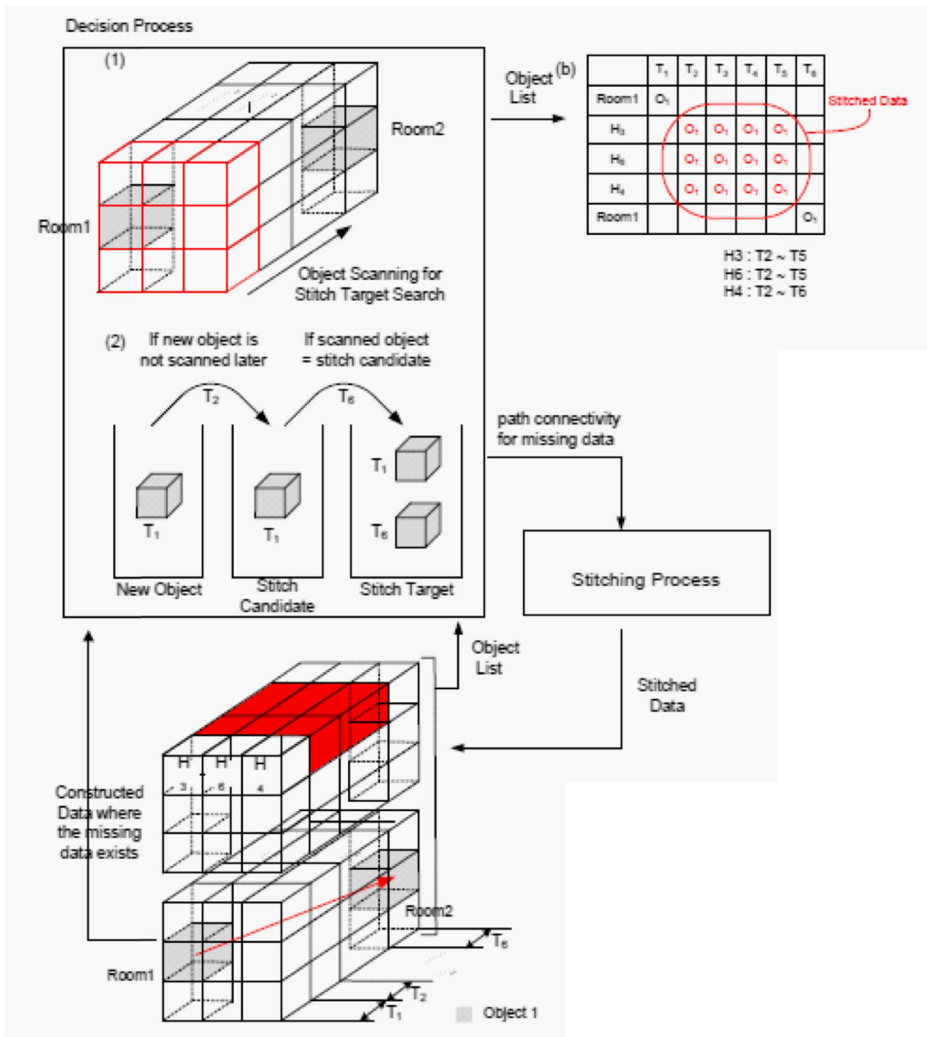
Fig. 7 The decision process is illustrated for initiating the stitching process to connect the data corresponding to two available regions. The data is connected through probable paths

5 Evaluation and verification

In this evaluation, we use CEWIT building at Stony Brook University as the surveillance environment. 40 objects with tag ID and 60 objects with no tag ID exist in the environment from 8 AM to 5 PM. In each room and hallway, two cameras and RFID sensor are considered. Its location is associated with tag ID detected by RFID sensor. We generate three types of data according to the degree of data inconsistency. The inconsistency is mainly affected by two factors such as tracking performance and association performance. The tracking failure takes place as randomly according to uniform distribution. For association performance, we generate three cases, group association, miss-association and incomplete association by 30, 20 and 10 % for three types of data.

Figure 8 shows an example of the hierarchy map. To get effectiveness of search, we define a spatial element, partition. The number of rooms and hallways are evenly distributed within each partition. The data retrieval is done hierarchically from partition level to unit level. In order to estimate object's future trajectory, it is necessary to know the connectivity between spatial elements. Such connectivity map information is represented by a graph as shown in Fig. 8b–d. The edge indicates bidirectional connectivity between two nodes indicating spatial element.

As shown in Fig. 9, all objects can be visualized. Moreover, the trajectories before and after the consistency maintenance can be visualized. Figure 10 demonstrates the results of the query for determining the possible meeting events of two objects. The result indicates there are two possible regions that two objects met. One illustrates that the inferred region is small due to sufficient data available. The other shows the entire room (shown in bottom right) due to insufficient data available.
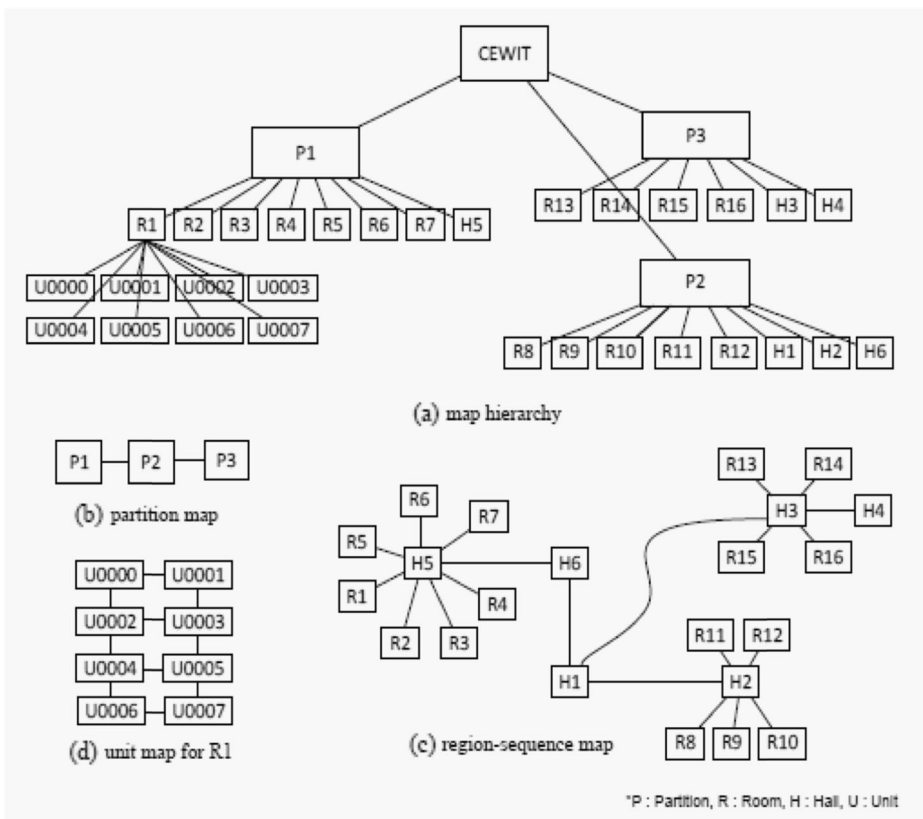

Fig. 8 The map information generated for the environment: **a** map hierarchy, **b** partition map, **c** region map, **d** unit map for R1
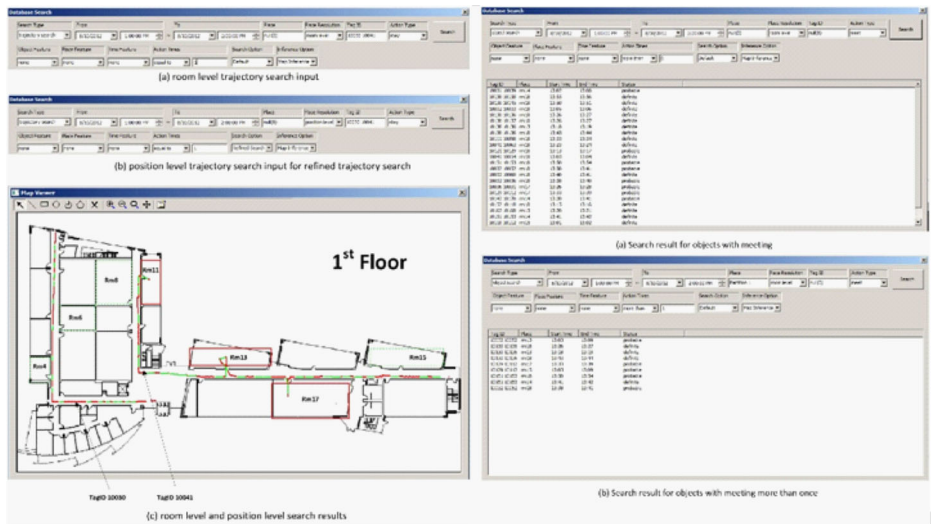
Fig. 9 Illustration of visualized trajectories. Trajectories from the raw data as well as stitched data are shown
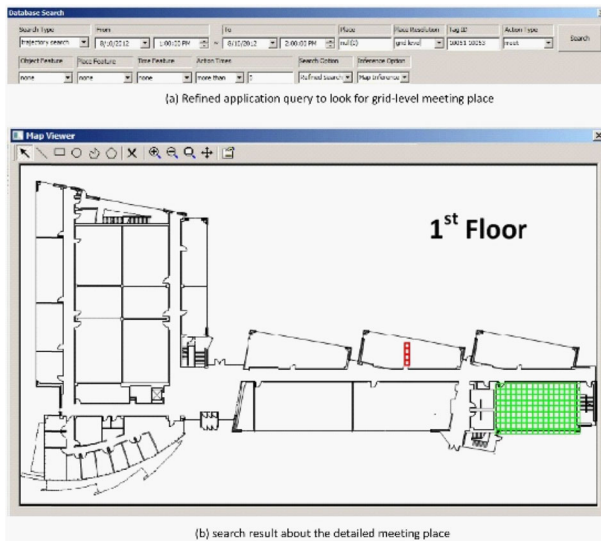

Fig. 10 Inferred meeting region of two objects. Larger region indicates that insufficient data were available

6 Conclusion

This paper presents a database management scheme for an intelligent surveillance system with heterogeneous sensors. One of the key problems for designing intelligent surveillance system database is accurate information access efficiently. The size of data grows in time and the database inconsistency makes the information access very difficult. In order to increase the accuracy and to minimize the access time, the inconsistency of the database is minimized with real-time back-annotation. The information query process is simplified through subcommand process method. The subcommand process generates minimal amount information as a cache to improve the access time. To verify the proposed system, we have constructed simulation environment with heterogeneous sensors (visual cameras and RFID readers). TO create the inconsistency, the dense object trajectories are incorporated to recreate group associations and false associations. The simulation results demonstrate that the database information for objects is successfully recovered with consistency. For the future work, we will investigate the probabilistic approach with the history trajectories of objects to consistently maintain the global object numbers for closely neighboring objects.

References

 1. Cho SH, Hong S, Moon N, Park P, Oh S-J (2010) Object association and identification in heterogeneous sensors environment. EURASIP J. Adv. Sig. Proc. 2010
 2. Cho SH, Hong S, Nam Y (2009) Association and identification in heterogeneous sensors environment with coverage uncertainty. AVSS 553–558
 3. Cho S, Kim KH, Hong S (2014) Effective object identification and association by varying coverage through RFID power control. J Comput Sci Technol 29(1):4–20
 4. Cho SH, Nam Y, Hong S, Cho W (2010) Locally initiating line-based object association in large scale multiple cameras environment. TIIS 4(3):358–379
 5. Cho SH, Nam Y, Hong S, Cho W (2011) Sector based scanning and adaptive active tracking of multiple objects. TIIS 5(6):1166–1191
 6. Cho SH, Nam Y, Hong A (2010) Multiple camera collaboration strategies for dynamic object association. TIIS 4(6):1169–1192
 7. Goswami K, Hong G-S, Kim B-G (2013) A novel mesh-based moving object detection technique in video sequence. J Converg 4(3):20–24
 8. Hu W, Tan T, Wang L, Maybank S (2004) A survey on visual surveillance of object motion and behaviors. IEEE Trans Syst Man Cybern Part C Appl Rev 34(3):334–-352. doi:10.1109/TSMCC.2004.829274
 9. Kim KR, Moon N (2012) Designing a social learning content management system based on learning objects. Multimed Tools Appl 51(3):1–15

10. Ma J, Liu W, Miller P, Yan W (2009) Event composition with imperfect information for bus surveillance. IEEE Conf Adv Video Signal Based Surveill 382–387
11. Oh J-M, Moon N (2011) User-selectable interactive recommendation system in mobile environment. Multimed Tools Appl 52(1):1–19
12. Okadome T, Hattori T, Hiramatsu K, Yanagisawa Y (2006) A real-world event search system in sensor network environments. IEEE Int Conf Mob Data Manag 62
13. Porikli F, Bremond F, Dockstader SL, Ferryman J, Hoogs A, Lovell BC, Pankanti S, Rinner B, Tu P, Venetianer PL (2013) Video surveillance: past, present, and now the future. IEEE Signal Process Mag 30(3): 190–198
14. Ra Andty T (2010) Survey on contemporary remote surveillance systems for public safety. IEEE Trans Syst Man Cybern Part C Appl Rev 40(5):493–515. doi:10.1109/TSMCC.2010.2042446
15. Salim K, Hafida B, Ahmed RS (2014) Probabilistic models for local patterns analysis. J Inf Process Syst 10(1):145–161
16. Shahabi C, Kim SH, Nocera L, Constantinou G, Lu Y, Cai Y-H, Medioni G, Nevatia R, Banaei-Kashani F (2014) Janus - multi source event detection and collection system for effective surveillance of criminal activity. J Inf Process Syst 10(1):1–22
17. Shu CF, Hampapur A, Lu M, Brown L, Connell J, Senior A, Tian Y (2005) IBM smart surveillance system (s3): an open and extensible framework for event based surveillance. IEEE Conf AVSS 2005 318–323. doi:10.1109/AVSS.2005.1577288
18. Valera M, Velastin S (2005) Intelligent distributed surveillance systems: a review. IEEE Proc Vis Image Signal Process 152(2):192–204
19. Wang X (2013) Intelligent multi-camera video surveillance: a review. Pattern Recogn Lett 34:319
20. Wasserkrug S, Gal A, Etzion O (2008) Inference of security hazards from event composition based on incomplete or uncertain information. IEEE Trans Knowl Data Eng 20(8):1111–1114
21. Yang X, Peng G, Cai Z, Zeng K (2013) Occluded and low resolution face detection with hierarchical deformable model. J Converg 4(2):11–14
22. Yap FGH, Yen H-H (2014) A survey on sensor coverage and visual data capturing/processing/transmission in wireless visual sensor networks. Sensors 14(2):3506–3527

Jung-Min Oh Jung-Min Oh received the B.S. degree in Business Administration from Sookmyung University and the M.S. degree in Digital Media from Seoul University of Venture & Information, in 1999 and 2008, respectively. Also she received the degree in Ph.D in IT Application Technology from the University of Hoseo Graduate School of Venture. She is currently working as the post doctor at the Stony Brook University. Her current research interests include user behavior analysis, metadata, and interactive service model.

Nammee Moon Nammee Moon received the B.S. degree in Computer Science from Ewha Womans University and the M.S. degree in Data Base from Ewha Womans University, in 1985 and 1987, respectively. Also she received the Ph.D. degree in Computer Technology in 1998 from Ewha Womans University. From 2000 to 2003, she served as a lecturer in the Ewha Womans University. Then, she joined the Department of Digital Media at Seoul University of Venture & Information as an associate professor from 2003 to 2008. After that, she joined the Department of Mobile Software at Hoseo University as a professor. Her current research interests include interactive digital business model, HCI, service oriented interactive video application, and metadata.

Sangjin Hong Sangjin Hong received the B.S and M.S degrees in Electrical Engineering \& Computer Science from the University of California, Berkeley. He received his Ph.D in Electrical Engineering \& Computer Science from the University of Michigan, Ann Arbor. He is currently with the department of Electrical and Computer Engineering at Stony Brook University - SUNY. Before joining SUNY, he has worked at Ford Aerospace Corp. Computer Systems Division as a systems engineer. He also worked at Samsung Electronics in Korea as a technical consultant. His current research interests are in the areas of low power system design of multimedia wireless communications and digital signal processing systems, reconfigurable Systems on Chip design and optimization, and VLSI signal processing. Prof. Hong served on numerous Technical Program Committees for IEEE conferences. He is currently serving as an editorial board member in EURASIP Journal of Advances in Signal Processing, Journal of Signal Processing Systems, and Scientific World Journal. Prof. Hong is a Senior Member of IEEE. He is also a member of Eta Kappa Nu and Tau Beta Pi Honor societies.