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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
9 September 2016 (09.09.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/140680 A1

(51) **International Patent Classification**:
G06F 17/00 (2006.01)

(21) **International Application Number**:
PCT/US2015/019014

(22) **International Filing Date**:
5 March 2015 (05.03.2015)

(25) **Filing Language**: English

(26) **Publication Language**: English

(71) **Applicant**: **HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP** [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).

(72) **Inventors**: **KAFAI, Mehran**; 1501 Page Mill Rd., Palo Alto, California 94304-1100 (US). **AN, Le**; 1501 Page Mill Rd., Palo Alto, California 94304-1100 (US). **MITCHELL, April Slayden**; 1501 Page Mill Rd., Palo Alto, California 94304-1100 (US).

(74) **Agents**: **KIRCHEV, Ivan T** et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, Fort Collins, CO 80528 (US).

(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, BN, 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, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, 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, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, 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, KM, 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**: MULTI-LEVEL OBJECT RE-IDENTIFICATION

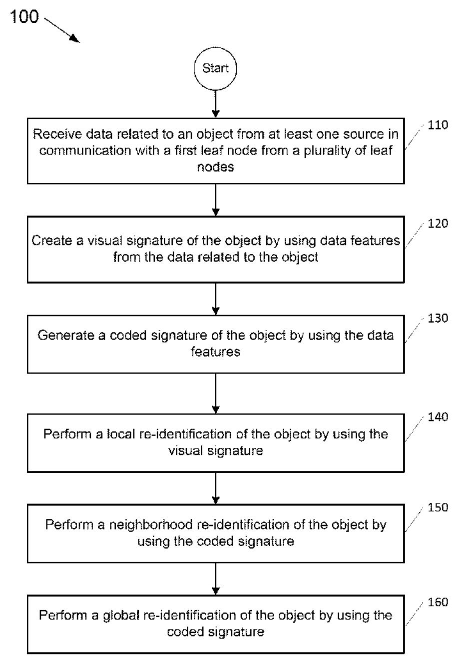

Fig. 2

(57) **Abstract**: An example system is provided in according with one implementation of the present disclosure. The system includes a data engine to receive data related to an object from at least one source in communication with a first leaf node from a plurality of leaf nodes. The system also includes a processing engine to extract data features from the data related to the object to create a visual signature of the object and to transform the data features to a coded signature of the object. The system also includes a re-identification analysis engine to perform a multi-level re-identification of the object, wherein the visual signature of the object is used in a first level of re-identification and the coded signature of the object is used in at least one different level of re-identification.

WO 2016/140680 A1

Published:

— *with international search report (Art. 21(3))*

MULTI-LEVEL OBJECT RE-IDENTIFICATION

BACKGROUND

[0001] Generally, object re-identification aims to recognize an object when its presence occurs more than once. With the recent improvements in technology, accurate and efficient object re-identification has been a desirable goal for many organizations. The collected re-identification information may be used for different purposes and may be of a high value to these organizations.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Figure 1 is a schematic illustration of an example system for object re-identification in accordance with an implementation of the present disclosure.

[0003] Figure 2 illustrates a flow chart showing an example of a method for object re-identification in accordance with an implementation of the present disclosure.

[0004] Figure 3 is an example block diagram illustrating a computer-readable medium in accordance with an implementation of the present disclosure.

DETAILED DESCRIPTION OF SPECIFIC EXAMPLES

[0005] As noted above, with the recent improvements in technology, the desire for accurate and efficient object re-identification is becoming increasingly popular in all industries. The goal of re-identification is to detect an object and determine whether that object has appeared before in an environment by matching the identity of the recognized object as compared against other known identities. As used herein, the term “object” may refer to any type of object that can be re-identified by using data related to that object. Examples of objects may include people, images, audio signals, vehicles, documents, etc.

[0006] The information collected from the object re-identification process may be used in many different ways. For example, person re-identification is an important application for many industries including surveillance, security, and retail. The goal of person re-identification is to match individuals recognized in images or videos across different cameras at different time and locations.

Security entities may benefit from the ability to track and locate known suspects in a particular area. Business intelligent analysts can use person re-identification to analyze customer shopping behavior to estimate customer attention and dwell-time in a retail store. These statistics may be used by companies to develop targeted marketing and personalized recommendations. Other entities, which utilize individual matching or broad pattern matching across large data, may also benefit from the proposed framework.

[0007] Generally, object re-identification may be performed by comparing the detected object (e.g., the data features of the object) to a gallery of previously detected and/or stored objects. In other words, comparison and matching of the object is performed with the features of the object. Many times, no data scaling or normalization is performed when the object is initially detected and compared to existing objects. Thus, in many situations, large amounts of data need to pass through traffic in order for the re-identification to occur. This is generally a slow and inefficient way to process and compare data. Further, the accuracy of the re-identification may be compromised due to the large amounts of data passing through the network, which may cause various different problems for the network.

[0008] Due to the encumbrances described above, improved systems and techniques for object re-identification are desired. The present description is directed to an enhanced solution for object re-identification that performs a distributed multi-level re-identification of an object. Specifically, implementations of the disclosed technology provide a system that executes efficient three-level matching of a detected object by utilizing different data feature spaces at the different levels.

[0009] In one example, local re-identification (i.e., first level) is performed locally on a leaf node that is in communication with a source (e.g., a sensor) that detected the object and provided the data related to the object. Local re-identification may use the original data feature descriptors of the object to perform the re-identification. Then, neighborhood re-identification (i.e., second level) is performed on a group of neighboring leaf nodes. Finally, if required, global re-identification (i.e., third level) is executed on a central node connected to the plurality of leaf nodes. The original data features of the object may be transformed to a coded signature (e.g., orthogonal transform based hash, etc.) of the object and only the coded signature of the object (as opposed to the original data features) may be used in the neighborhood and the global re-identification.

[0010] Thus, the proposed technology allows for efficient and accurate object re-identification. Scalability is ensured via local matching in a reduced search space, in addition to neighborhood and global matching with coded signatures. Unlike conventional object re-identification frameworks, both the data acquisition and data processing are distributed throughout a mesh network. By utilizing a three-level distributed matching protocol, the proposed solution only uses original feature descriptors of the object at the local re-identification level. That removes the need to transmit raw data (e.g., images, etc.) across the network. Distributing the matching process by using the coded signature in the second and third re-identification levels improves the overall efficiency especially in large-scale sensor networks. That way, through usage of coded signatures information leakage is minimized.

[0011] In the following detailed description, reference is made to the accompanying drawings, which form a part hereof, and in which is shown by way of illustration specific examples in which the disclosed subject matter may be practiced. It is to be understood that other examples may be utilized and structural or logical changes may be made without departing from the scope of the present disclosure. The following detailed description, therefore, is not to be taken in a limiting sense, and the scope of the present disclosure is defined by the appended claims. Also, it is to be understood that the phraseology and terminology used herein is for the purpose of description ad should not be regarded as limiting. The use of “including,” “comprising” or “having” and variations thereof herein is meant to encompass the items listed thereafter and equivalents thereof as well as additional items. Furthermore, the term “based on,” as used herein, means “based at least in part on.” It should also be noted that a plurality of hardware and software based devices, as well as a plurality of different structural components may be used to implement the disclosed methods and devices.

[0012] Referring now to the figures, Figure 1 is a schematic illustration of an example system 5 for object re-identification. The illustrated system 5 is capable of carrying out the techniques described below. As shown in Figure 1, the system 5 is depicted as including at least one source 10 (e.g., a sensor, etc.), the source in communication with at least one leaf node 15 from a plurality of leaf nodes, at least one central node 20, and a computing device 25 including engines 39-41. Although only computing device 25 is described in details below, the techniques described herein may be performed with several computing devices or with engines distributed on different devices.

[0013] In one example, sources 10 may be sensors (e.g., cameras, etc.) detecting objects (not shown) and providing data related to these objects. A leaf node may 15 may be any type of a processing unit, a computing device, or a workstation with moderate processing power. In one implementation, the central node 20 may be a single processing unit with large memory to store a global set of data from all leaf nodes 15. However, the central node 20 may also include be a cluster of machines or processing units.

[0014] The computing device 25 may be any type of a computing device and may include at least engines 39-41. In one example, the computing device 25 may be any of the leaf nodes 15 or the central node 20. In another example, the computing device 25 may be an independent computing device. In yet another alternative example, engines 39-41 may be distributed on the leaf nodes 15, the central node 20, or any other computing device.

[0015] The computing device 25 may include additional components (e.g., processor, machine-readable storage medium, etc.) and some of the components depicted therein may be removed and/or modified without departing from a scope of the system that allows for carrying out the functionality described herein. It is to be understood that the operations described as being performed by the engines 39-41 of the computing device 25 that are related to this description may, in some implementations, be performed by external engines (not shown) or distributed between the engines of the computing device 25 and other electronic/computing devices.

[0016] The computing device 25 may include communication interfaces (e.g., a Wi-Fi® interface, a Bluetooth® interface, a 3G interface, a 4G interface, a near field communication (NFC) interface, etc.) that are used to connect with other devices/systems and/or to a network. In one example, the network may be a mesh sensor network (not shown). The network may include any suitable type or configuration of network to allow for communication between the computing device 25, sources 10, leaf nodes 15, central node 20, and any other devices/systems (e.g., other computing devices, displays, etc.).

[0017] Each of the engines 39-41 may include, for example, at least one hardware device including electronic circuitry for implementing the functionality described below, such as control logic and/or memory. In addition or as an alternative, the engines 39-41 may be implemented as any combination of hardware and software to implement the functionalities of the engines. For example, the hardware may be a processor and the software may be a series of instructions or microcode encoded on a machine-readable storage medium and executable by the processor. Therefore, as used herein, an engine may include program code, e.g., computer executable instructions, hardware, firmware, and/or logic, or combination thereof to perform particular actions, tasks, and functions described in more detail herein in reference to Figures 2-3.

[0018] In one example, the data engine 39 may receive data related to an object (not shown) from at least one source 10 in communication with a first leaf node 15 from a plurality of leaf nodes. For instance, sources 10 may be internet protocol ("IP") cameras that may take images of a specific field of view at a certain rate and send the images to at least one leaf node 15. In other instances, the data related to the object may be any other type of data that can be retrieved by a source and sent to a leaf node. Each source 10 from the plurality of sources may be assigned to a leaf node 15, such that the node 15 may be in charge of the source 10. Thus, a source may deliver the data it has collected to the node, the node may control how the source performs any given task (e.g., determine parameters, time, etc.), the node may configure the source, the node may monitor the source (e.g., for fault, errors, etc.), etc. In some examples, each source 10 is assigned to the node 15 that is closest to that source.

[0019] The processing engine 40 may extract data features from the data related to the object to create or generate a visual signature of the object and to transform the data features to a coded signature of the object. In one implementation, that process may be performed on a leaf node 15 that is in charge of the source 10 providing the data (i.e., the source 10 is assigned to the node 15). For example, the processing engine 40 may extract a plurality of original data features (e.g., texture, color, etc.) from the received data related to the object to create a visual signature of the object. As used herein, the term "visual signature" refers to a combination of original data features extracted from data associated with an object that may be used to identify the object. In one example, the data features and the visual signatures of the objects may be stored at the leaf node 15 in charge of the source 10 providing the data. The different techniques for creating visual signatures of the objects the data features will be described in more details below.

[0020] Further, the processing engine 40 may transform the data features of the object to generate to a coded signature of the object. As used herein, the term "coded signature" refers to a combination of transformed and/or digested data features for an object (e.g., a compilation or summary of material or information of the object) that preserve the similarity between objects. In one example, the coded signature of the object may be in the form of a hash (e.g., orthogonal transform based hash, Walsh Hadamard Transform ("WHT") hash, etc.) or any other type of data digest generated from a data string, where the coded signature is substantially smaller than the original data. The hashing process (i.e., generating a coded signature) may be performed at a leaf node 15 and the coded signature may be stored at a local gallery of the node 15. Then, the coded signature of the object may be sent to a group of neighboring leaf nodes 15 and the central node 20 to be stored at a global gallery. Before sending the coded signature, the processing engine 40 may filter the coded signatures to ensure no duplicates are sent to the leaf nodes. The different techniques for generating a coded signature of the data features of an object will be described in more details below.

[0021] The re-identification analysis engine 41 may perform a distributed multi-level re-identification of the object initially detected by the source 10. In one implementation, the engine 41 may use the visual signature of the object in a first level of re-identification and the engine 41 may use the coded signature of the object in at least one different level of re-identification.

[0022] For example, the re-identification analysis engine 41 may perform the first level of re-identification (i.e., local matching) on a first leaf node 15 (i.e., the node that received the object data from a source) by comparing the visual signature of the object to visual signatures stored at the first leaf node (e.g., at a local gallery). Thus, the original data features of the identified object may be compared to a plurality of visual signatures. By comparing the original features in object, the first level of re-identification achieves a high accuracy, but it may be slow to process because it may use a linear scan of the local gallery (e.g., one to one comparison). As explained in additional details below, the engine 41 may perform clustering before the linear scan in order to speed the re-identification.

[0023] In addition, the re-identification analysis engine 41 may perform a second level of re-identification (i.e., neighborhood matching) on each of a group of leaf nodes 15 that are neighboring the first leaf node (i.e., the node that received the object data from a source). The second level of re-identification may be performed by comparing the coded signature of the object to existing coded signatures stored at each of the neighboring leaf nodes. In one example, each leaf node 15 stores a plurality of coded signatures from sources 10 in communication with the leaf node in a local gallery, and the central node 20 stores coded signatures collected from the plurality of leaf nodes 10 in a global gallery. These galleries may include records of known objects that were previously detected and identified by the system.

[0024] Thus, if local matching of the object is not successful, the leaf node 15 may send the created coded signature to its neighboring nodes for further processing/matching. In some examples, each leaf node 15 may be aware of its neighboring leaf nodes. As explained in additional details below, different techniques may be used to define the neighboring nodes of a node 15.

[0025] Generally, there is a higher probability that an object has been previously detected within a close vicinity of its current location. Therefore, the neighboring leaf nodes are examined to determine if the object can be re-identified. In that case, data needs to be sent over the network to perform the matching. However, sending a coded signature (e.g. a hash) makes the transfer of data fast and the consequent searching on the plurality of neighboring leaf nodes very efficient.

[0026] Further, the re-identification analysis engine 41 may perform a third level of re-identification (e.g., global matching) on a central node 20 connected to the plurality of leaf nodes 15 by comparing the coded signature of the object to existing coded signatures stored at the central node. The third level of re-identification may be performed by comparing the coded signature of the object to existing coded signatures stored at the central node (e.g., at a global gallery). In other words, if the neighborhood re-identification of the object is not successful, the leaf node 15 (e.g., the engine 41) sends the created coded signature to the central node for further processing/matching. A global gallery in the central node may store coded signatures (e.g., hashed items) of every object ever detected within the network. Storing and comparing only coded signatures of objects takes less space and is much faster.

[0027] The re-identification analysis 41 engine may further add the coded signature of the object to the global gallery of the central node 20 as a new record of a known object when at least one of the first level of re-identification and the second level of re-identification is successful. In addition, the engine 41 may add the coded signature of the object to the global gallery of the central node 20 as a new record of a known object and may add the visual signature of the object to a local gallery of the first leaf node 15 as a new record of a known object when the third level of re-identification is successful. Further, the engine 41 may add the coded signature of the object to the global gallery of the central node as a new record of a new object and add the visual signature of the object to a local gallery of the first leaf node as a new record of a new object when the multi-level re-identification is unsuccessful. That way, the galleries are continuously updated with data for new records.

[0028] Figure 2 illustrates a flow chart showing an example of a method 100 for object re-identification. Although execution of the method 100 is described below with reference to the system 5, the components for executing the method 100 may be spread among multiple devices/systems. The method 100 may be implemented in the form of executable instructions stored on a machine-readable storage medium, and/or in the form of electronic circuitry.

[0029] In one example, the method 100 can be executed by at least one processor of a computing device (e.g., device 25). In other examples, the method may be executed by another processor in communication with the system 5. Various elements or blocks described herein with respect to the method 100 are capable of being executed simultaneously, in parallel, or in an order that differs from the illustrated serial manner of execution. The method 100 is also capable of being executed using additional or fewer elements than are shown in the illustrated examples.

[0030] The method 100 begins at 110, where a processor may receive data related to an object from at least one source in communication with a first leaf node from a plurality of leaf nodes. As mentioned above, the source may be any type of source that may provide any type of data related to an object.

[0031] At 120, the processor may create a visual signature of the object by using data features from the data related to the object. In one implementation, the processor may extract data features from the data related to the object to generate the visual signature of the object. In one example, the visual signature of an object can be computed or generated by using various color or texture descriptors of the object. Hue-saturation-value ("HSV") and LAB color histograms may be used to describe the color feature of an object. Local binary patterns (LBP) may be used to define the texture feature of an object. In one example, to extract the visual signature of an object, the image of the object may be divided into equal sized blocks and the color and LBP features from each block may be concatenated to form the final visual signature of the entire object.

[0032] Other data features of an object may also be used for creating a visual signature and for re-identification. For example, semantic color names may be used to describe an object instead of using simple color histograms. The color names are high level color descriptors that can be well interpreted by humans and are more robust as compared to color histograms. In another example, the processor may use an ensemble of color and texture features including RGB, YCbCr, HSV color histograms, and two types of texture features extracted by Schmid and Gabor filters computed across different radiuses and scales to create a visual signature. In these methods, the features may be extracted from image blocks and the final visual signature may consist of features from all blocks.

[0033] In another proposed approach, a neural network may be used for creating a visual signature. For example, misalignment, pose difference, occlusions and background clutter data features may be jointly handled with abundant data. Instead of extracting hand crafted features such as color histogram or LBP, the deep neural network is able to learn the visual signature directly from image intensities. In one implementation, that visual signature generation process may be performed on a leaf node 15 in charge of the source 10 providing the data (i.e., the source 10 is assigned to the leaf node 15).

[0034] With continued reference to Figure 2, the processor may generate a coded signature of the object by using the data features (at 130). In one implementation, the coded signature may be generated as a hash or any other type of data digest based on the received data features. For example, a data object may be represented as an N-dimensional numerical vector A and it may be mapped to a set of H keys. Each key may be an integer chosen from a universe of keys from 1…U. The mapping process may include multiple steps: (a) an extended version of A (called A') is generated by concatenating A d times where d = floor(U/N); (b) an orthogonal transform of a randomly permutated copy of the extended vector A' is computed; and (c) the indices of the H largest members of the transform represent the coded signature for A. In one example, the process of generating a coded signature may be performed at a leaf node 15 (e.g., the leaf node that received the object data from the source 10). When generated, the coded signature of an object may be stored at a local gallery of the leaf node 15 (i.e., the node that received the object data from the source 10), and may also be sent to the central node 20 to be stored at a global gallery.

[0035] At 140, a processor may perform a local re-identification of the object by using the visual signature. In one implementation, the local re-identification may be performed on a leaf node 15 (e.g., the leaf node 15 that received the object data from the source 10) by comparing the visual signature of the object to visual signatures stored at a local gallery of the leaf node 15. The local gallery may include a list of objects that have corresponding visual signatures and associated IDs. Thus, the original data features of the identified object (in the form of a visual signature) may be compared to a plurality of visual signatures of objects previously detected by any sources 10 that provides data to that leaf node 15. If the visual signature of the object is matched to an existing signature in the local gallery, the ID of the stored object is assigned to the new object. If the visual signature of the object is not matched, neighborhood re-identification is performed next.

[0036] In some implementations, the processor may perform clustering of the received data related to the object before local re-identification is performed. For example, the objects in the local gallery of the node 15 may be first partitioned into multiple clusters. Various clustering methods (e.g., affinity propagation algorithm ("APA") clustering, K-means clustering, spectral clustering, etc.) may be used. In one example, APA clustering may perform message passing among the data points and cluster centers may be automatically selected based on two types of messages: responsibility and availability. As used herein, a cluster center is so-called exemplar which is representative of the data in this cluster. Once the local gallery is clustered, given a query (e.g., a visual signature), it is first compared with the exemplars from all the clusters. Then, a linear search may be performed in the cluster whose exemplar is closest to the query. In this manner, the search space during local re-identification is reduced.

[0037] With continued reference to Figure 2, the processor may then perform a neighborhood re-identification of the object by using the coded signature (at 150). In one implementation, the neighborhood re-identification is performed on each of a group of leaf nodes 15 that are neighboring a leaf node (e.g., the leaf node 15 that received the object data from the source 10) by comparing the coded signature of the object to existing coded signatures stored at each of the neighboring leaf nodes. Various matching techniques may be used to match the coded signature to the plurality of coded signatures stored at the neighboring nodes 15.

[0038] Thus, after an unsuccessful local re-identification, neighborhood re-identification is performed at the neighboring nodes 15 of the node that initially received the object data. As noted above, each leaf node 15 includes a local gallery with coded signatures related to different objects. In some implementations, a group of neighboring leaf nodes 15 may be defined geographically (e.g., within 500 feet from a node 15), may be defined based on a numerical value (e.g., the three closest nodes), may be based on a field of sensing (e.g., when the leaf nodes have overlapping field of view/sensing they), may be defined in any other applicable way. Each leaf node 15 may have a predefined or flexible number of neighboring leaf nodes 15. However, each leaf nodes 15 may be aware of its neighboring leaf nodes at all times.

[0039] If the coded signature of the object is matched to an existing coded signature in the local gallery of a neighboring leaf node 15, the ID of the stored object is returned to the leaf node 15 that initially detected the object. The processor may then assign an ID of the matched object to the new object and may store that data accordingly (e.g., in the local gallery, the global gallery of the central node, etc.). Also, the processor may add the coded signature of the object to the global gallery of the central node 20 as a new record of a known object when at least one of the local re-identification and the neighborhood re-identification is successful. If the coded signature of the object is not matched to any of neighboring leaf node 15, global re-identification is performed next.

[0040] At 160, the processor may perform a global re-identification of the object by using the coded signature. In one implementation, the global re-identification is performed on a central node 20 connected to the plurality of leaf nodes 15 by comparing the coded signature of the object to existing coded signatures stored at the central node. Thus, the different levels of re-identifications of the detected object are performed by utilizing different data feature spaces (e.g., visual signatures vs. coded signatures) at the different levels.

[0041] After an unsuccessful neighborhood re-identification, global re-identification is performed at the central node 20. As noted above, the central node 20 includes a global gallery with coded signatures from all leaf nodes 15 in communication with the central node 20 (i.e., potentially capturing objects detected from all sources 10 within the network). Various matching techniques may be used to match the coded signature to the plurality of coded signatures stored at the central node.

[0042] If the coded signature of the object is matched to an existing coded signature in the global gallery of a central node 20, the processor may add the coded signature of the object to the global gallery of the central node 20 as a new record of a known object and may also add the visual signature of the object to a local gallery of the leaf node 15 (i.e., the node that first received the object data) as a new record of a known object. Further, if all three levels of re-identification are not successful, the processor may add the coded signature of the object to the global gallery of the central node 20 as a new record of a new object. In addition, the processor may add the visual signature of the object to a local gallery of the leaf node (i.e., that first received the object data) as a new record of a new object. That way, the galleries are continuously updated.

[0043] One example of how multi-level re-identification and utilizing coded signatures of object helps to increase the efficiency of the re-identification process is described below. In one implementation, a typical feature descriptor (for the purpose of person re-identification) for a single image of a person consists of about 70,000 features (e.g., real-valued numbers representing texture plus color histogram from densely sampled image regions). Such feature descriptor may be transformed into a WHT hash coded signature of 100 dimensions (i.e., 100 unsigned integers/ 400 bytes). Sending the coded signature between leaf nodes 15 reduces network traffic in the system 5. On the other hand, using the WHT has coded signatures on the central node increases retrieval efficiency.

[0044] Figure 3 illustrates a computer 301 and a non-transitory machine-readable medium 305 according to an example. In one example, the computer 301 maybe similar to the computing device 25 of the system 5 or may include a plurality of computers. For example, the computer may be a server computer, a workstation computer, a desktop computer, a laptop, a mobile device, or the like, and may be part of a distributed system. The computer may include one or more processors and one or more machine-readable storage media. In one example, the computer may include a user interface (e.g., touch interface, mouse, keyboard, gesture input device, etc.).

[0045] Computer 301 may perform method 100 and variations thereof. Additionally, the functionality implemented by computer 301 may be part of a larger software platform, system, application, or the like. Computer 301 may be connected to a database (not shown) via a network. The network may be any type of communications network, including, but not limited to, wire-based networks (e.g., cable), wireless networks (e.g., cellular, satellite), cellular telecommunications network(s), and IP-based telecommunications network(s) (e.g., Voice over Internet Protocol networks). The network may also include traditional landline or a public switched telephone network (PSTN), or combinations of the foregoing.

[0046] The computer 301 may include a processor 303 and non-transitory machine-readable storage medium 305. The processor 303 (e.g., a central processing unit, a group of distributed processors, a microprocessor, a microcontroller, an application-specific integrated circuit (ASIC), a graphics processor, a multiprocessor, a virtual processor, a cloud processing system, or another suitable controller or programmable device) and the storage medium 305 may be operatively coupled to a bus. Processor 303 can include single or multiple cores on a chip, multiple cores across multiple chips, multiple cores across multiple devices, or combinations thereof.

[0047] The storage medium 305 may include any suitable type, number, and configuration of volatile or non-transitory machine-readable storage media to store instructions and data. Examples of machine-readable storage media in include read-only memory ("ROM"), random access memory ("RAM") (e.g., dynamic RAM ["DRAM"], synchronous DRAM ["SDRAM"], etc.), electrically erasable programmable read-only memory ("EEPROM"), magnetoresistive random access memory (MRAM), memristor, flash memory, SD card, floppy disk, compact disc read only memory (CD-ROM), digital video disc read only memory (DVD-ROM), and other suitable magnetic, optical, physical, or electronic memory on which software may be stored.

[0048] Software stored on the non-transitory machine-readable storage media 305 and executed by the processor 303 includes, for example, firmware, applications, program data, filters, rules, program modules, and other executable instructions. The processor 303 retrieves from the machine-readable storage media 305 and executes, among other things, instructions related to the control processes and methods described herein.

[0049] The processor 303 may fetch, decode, and execute instructions 307-311 among others, to implement various processing. As an alternative or in addition to retrieving and executing instructions, processor 303 may include at least one integrated circuit (IC), other control logic, other electronic circuits, or combinations thereof that include a number of electronic components for performing the functionality of instructions 307-311. Accordingly, processor 303 may be implemented across multiple processing units and instructions 307-311 may be implemented by different processing units in different areas of computer 301.

[0050] The instructions 307-311 when executed by processor 303 (e.g., via one processing element or multiple processing elements of the processor) can cause processor 303 to perform processes, for example, method 100, and/or variations and portions thereof. In other examples, the execution of these and other methods may be distributed between the processor 303 and other processors in communication with the processors 303.

[0051] For example, data receiving instructions 307 may cause processor 303 to receive data related to an object from at least one source 10 in communication with a first leaf node 15 from a plurality of leaf nodes. These instructions may function similarly to the techniques described in block 110 of method 100.

[0052] Data processing instructions 309 may cause the processor 303 to extract data features from the data related to the object to generate a visual signature of the object and to generate a coded signature of the object from the extracted data features. These instructions may function similarly to the techniques described blocks 120 and 130 of method 100. For example, the processing instructions 309 may cause the processor 303 to transform the extracted data features of the object to generate to a visual signature and a coded signature of the object that is used in the subsequent re-identification process.

[0053] Re-identification instructions 311 may cause the processor 303 to perform a three-level re-identification analysis of the object by using the visual signature of the object in the first level of re-identification and the coded signature of the object in at least one different level of re-identification (e.g., the second or the third level of re-identification). These instructions may function similarly to the techniques described blocks 140-160 of method 100. For example, re-identification instructions 311 may cause the processor 303 to: perform the first level of re-identification on the first leaf node 15 by comparing the visual signature of the object to visual signatures stored at the first leaf node 15, perform a second level of re-identification on each of a group of leaf nodes 15 that are neighboring the first leaf node 15 by comparing the coded signature of the object to existing coded signatures stored at each of the neighboring leaf nodes, and perform a third level of re-identification on a central node 20 connected to the plurality of leaf nodes by comparing the coded signature of the object to existing coded signatures stored at the central node.

[0054] In the foregoing description, numerous details are set forth to provide an understanding of the subject matter disclosed herein. However, implementations may be practiced without some or all of these details. Other implementations may include modifications and variations from the details discussed above. It is intended that the appended claims cover such modifications and variations.

CLAIMS

What is claimed is:

1. A system comprising:

a data engine to receive data related to an object from at least one source in communication with a first leaf node from a plurality of leaf nodes;

a processing engine to extract data features from the data related to the object to create a visual signature of the object and to transform the data features to a coded signature of the object; and

a re-identification analysis engine to perform a multi-level re-identification of the object, wherein the visual signature of the object is used in a first level of re-identification and the coded signature of the object is used in at least one different level of re-identification.

2. The system of claim 1, wherein the re-identification analysis engine is to perform the first level of re-identification on the first leaf node by comparing the visual signature of the object to visual signatures stored at the first leaf node, wherein the source is assigned to the first leaf node.

3. The system of claim 2, wherein the re-identification analysis engine is to perform a second level of re-identification on each of a group of leaf nodes that are neighboring the first leaf node by comparing the coded signature of the object to existing coded signatures stored at each of the neighboring leaf nodes.

4. The system of claim 3, wherein the re-identification analysis engine is to perform a third level of re-identification on a central node connected to the plurality of leaf nodes by comparing the coded signature of the object to existing coded signatures stored at the central node.

5. The system of claim 4, wherein each leaf node stores a plurality of coded signatures from sources in communication with the leaf node in a local

gallery, and wherein the central node stores coded signatures collected from the plurality of leaf nodes in a global gallery.

6. The system of claim 5, wherein the re-identification analysis engine is further to: add the coded signature of the object to the global gallery of the central node as a new record of a known object when at least one of the first level of re-identification and the second level of re-identification is successful.

7. The system of claim 5, wherein the re-identification analysis engine is further to:

add the coded signature of the object to the global gallery of the central node as a new record of a known object; and

add the visual signature of the object to a local gallery of the first leaf node as a new record of a known object when the third level of re-identification is successful.

8. The system of claim 5, wherein the re-identification analysis engine is further to:

add the coded signature of the object to the global gallery of the central node as a new record of a new object; and

add the visual signature of the object to a local gallery of the first leaf node as a new record of a new object when the multi-level re-identification is unsuccessful.

9. A method comprising, by at least one processor:

receiving data related to an object from at least one source in communication with a first leaf node from a plurality of leaf nodes;

creating a visual signature of the object by using data features from the data related to the object;

generating a coded signature of the object by using the data features;

performing a local re-identification of the object by using the visual signature;

performing a neighborhood re-identification of the object by using the coded signature; and

performing a global re-identification of the object by using the coded signature.

10. The method of claim 9, wherein the local re-identification is performed on the first leaf node by comparing the visual signature of the object to visual signatures stored at the first leaf node.

11. The method of claim 9, wherein the neighborhood re-identification is performed on each of a group of leaf nodes that are neighboring the first leaf node by comparing the coded signature of the object to existing coded signatures stored at each of the neighboring leaf nodes.

12. The method of claim 9, wherein the global re-identification is performed on a central node connected to the plurality of leaf nodes by comparing the coded signature of the object to existing coded signatures stored at the central node.

13. A non-transitory machine-readable storage medium encoded with instructions executable by at least one processor, the machine-readable storage medium comprising instructions to:

receive data related to an object from at least one source in communication with a first leaf node from a plurality of leaf nodes;

extract data features from the data related to the object to generate a visual signature of the object;

generate a coded signature of the object from the extracted data features; and

perform a three-level re-identification analysis of the object, by using the visual signature of the object in a first level of re-identification and the coded signature of the object in at least one different level of re-identification.

14. The non-transitory machine-readable storage medium of claim 13, further comprising instructions to: add the coded signature of the object to a global gallery of the central node as a new record of a known object when at least one of a first level of re-identification and a second level of re-identification is successful.

15. The non-transitory machine-readable storage medium of claim 14, further comprising instructions to:

add the coded signature of the object to the global gallery of the central node as a new record of a known object; and

add the visual signature of the object to a local gallery of the first leaf node as a new record of a known object when a third level of re-identification is successful.

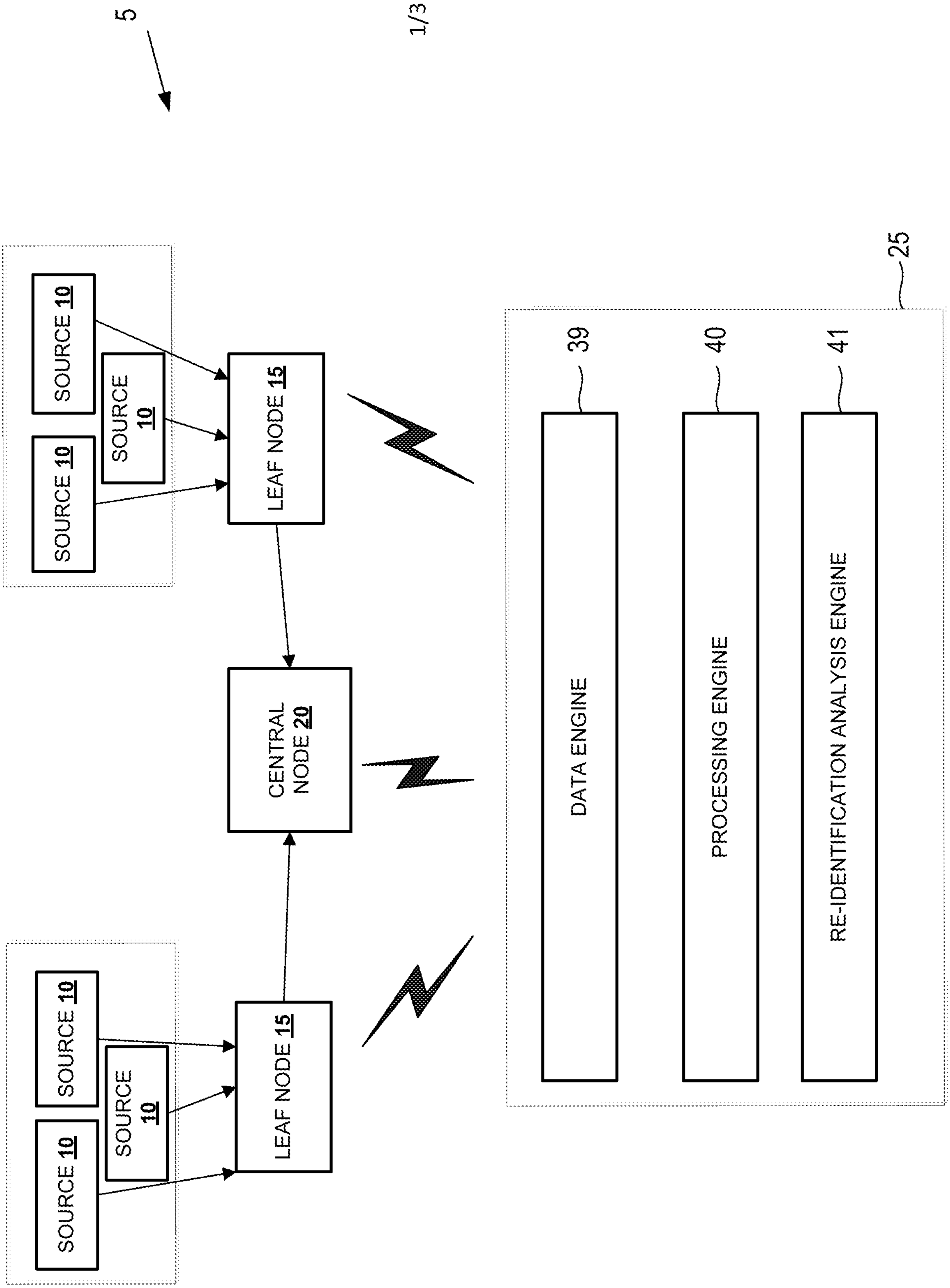
5
SOURCE 10
SOURCE 10
SOURCE 10
LEAF NODE 15
CENTRAL NODE 20
SOURCE 10
SOURCE 10
SOURCE 10
LEAF NODE 15
25
39
DATA ENGINE
40
PROCESSING ENGINE
41
RE-IDENTIFICATION ANALYSIS ENGINE

FIG. 1

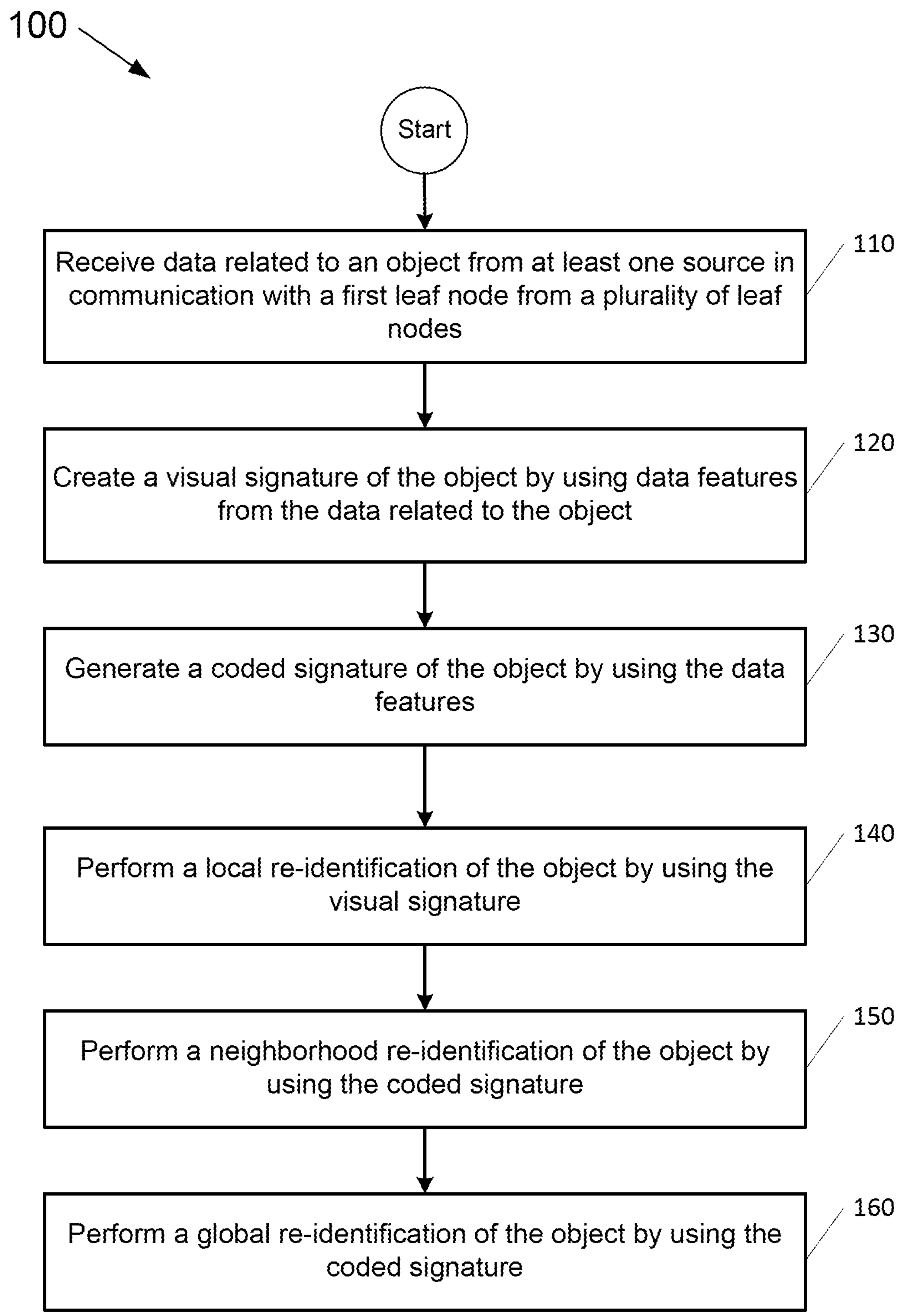
100
Start
Receive data related to an object from at least one source in communication with a first leaf node from a plurality of leaf nodes
110
Create a visual signature of the object by using data features from the data related to the object
120
Generate a coded signature of the object by using the data features
130
Perform a local re-identification of the object by using the visual signature
140
Perform a neighborhood re-identification of the object by using the coded signature
150
Perform a global re-identification of the object by using the coded signature
160

Fig. 2

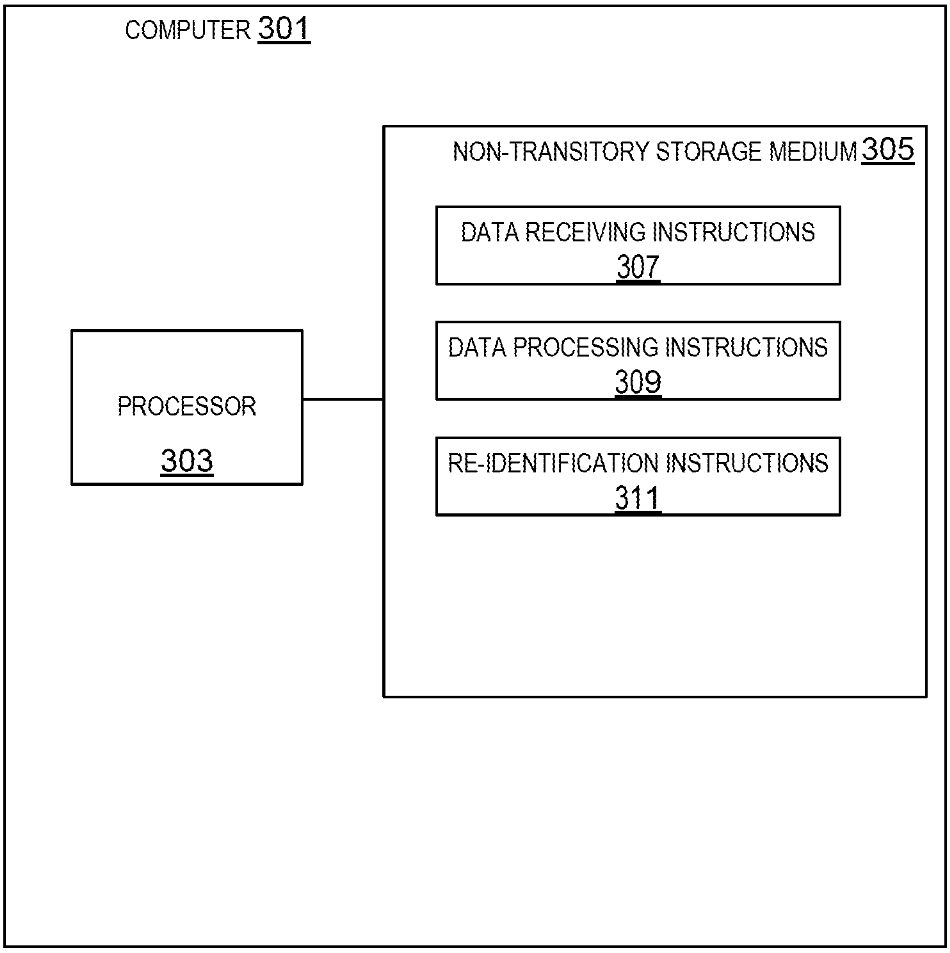
COMPUTER 301
NON-TRANSITORY STORAGE MEDIUM 305
DATA RECEIVING INSTRUCTIONS 307
DATA PROCESSING INSTRUCTIONS 309
RE-IDENTIFICATION INSTRUCTIONS 311
PROCESSOR 303

Fig. 3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/019014

A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/00(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F 17/00; H04N 5/225; H04N 7/18; G06K 9/46; H04M 11/04; G06K 9/00; G08B 1/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords:object re-identification, visual signature, coded signature, neighboring node, central node

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2007-0237357 A1 (COLIN ANDREW LOW) 11 October 2007 See paragraphs [0003], [0006], [0009], [0025], [0034], [0040]-[0041], [0047], [0054], [0143], and [0161]; and figure 1.	1-15
Y	US 2012-0249802 A1 (CAMILLO JOSE TAYLOR) 04 October 2012 See paragraphs [0013]-[0014], [0036], [0052], and [0056]; and figure 4B.	1-15
A	US 2014-0354821 A1 (DAVID A. MONROE) 04 December 2014 See paragraphs [0002], [0004], [0053], and [0055]-[0059].	1-15
A	US 2010-0195872 A1 (CARMELO VELARDO et al.) 05 August 2010 See paragraphs [0002], [0018], [0028]-[0029], and [0051].	1-15
A	US 2008-0258880 A1 (RICHARD A. SMITH et al.) 23 October 2008 See paragraphs [0002], [0008], [0047], [0054], and [0098].	1-15

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
20 April 2016 (20.04.2016)	**22 April 2016 (22.04.2016)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 35208, Republic of Korea
Facsimile No. +82-42-481-8578

Authorized officer
KIM, KI HO
Telephone No. +82-42-481-8691

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/019014

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2007-0237357 A1	11/10/2007	GB 0420778 D0	20/10/2004
		GB 2418310 A	22/03/2006
		GB 2418310 B	27/06/2007
		US 7697720 B2	13/04/2010
US 2012-0249802 A1	04/10/2012	US 2012-250978 A1	04/10/2012
		US 2012-250984 A1	04/10/2012
		US 8867793 B2	21/10/2014
US 2014-0354821 A1	04/12/2014	None	
US 2010-0195872 A1	05/08/2010	JP 2010-539623 A	16/12/2010
		WO 2009-039288 A1	26/03/2009
US 2008-0258880 A1	23/10/2008	US 8274377 B2	25/09/2012
		WO 2008-086507 A1	17/07/2008