

(43) International Publication Date  
23 September 2010 (23.09.2010)(10) International Publication Number  
**WO 2010/106060 A2**(51) International Patent Classification:  
**G06K 9/00** (2006.01)(21) International Application Number:  
PCT/EP2010/053373(22) International Filing Date:  
16 March 2010 (16.03.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
12/407,520 19 March 2009 (19.03.2009) US(71) Applicant (for all designated States except US): **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, New York 10504 (US).(71) Applicant (for MG only): **IBM UNITED KINGDOM LIMITED** [GB/GB]; PO Box 41, North Harbour, Portsmouth Hampshire PO6 3AU (GB).

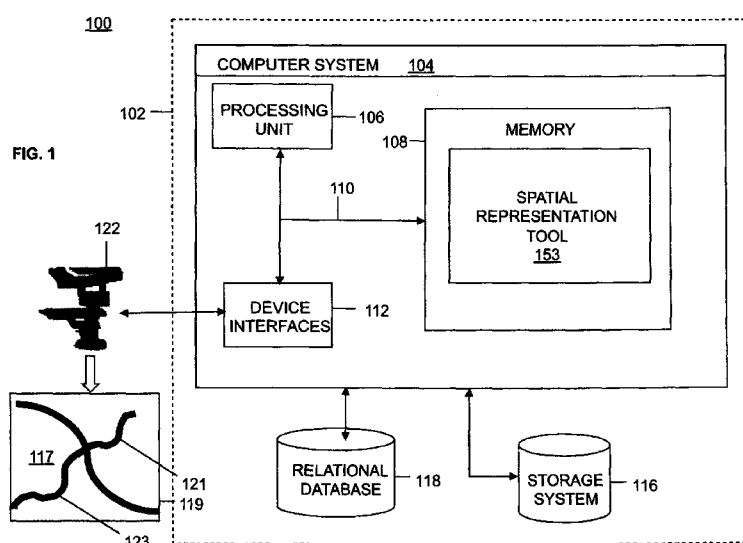
(72) Inventors; and

(75) Inventors/Applicants (for US only): **HAMPAPUR, Arun** [US/US]; IBM Corporation, 1101, Kitchawan Road, Route 134, PO Box 218, Yorktown Heights, New York 10598 (US). **LU, Zuoxuan** [CN/US]; IBM Corporation, 19, Skyline Drive, Hawthorne, New York 10532-1596 (US). **SHU, Chiao-Fe** [US/US]; IBM Corporation, 19, Skyline Drive, Hawthorne, New York10532-1596 (US). **MILITE, Christopher Scott** [US/US]; IBM Corporation, Mail Drop # 343, 150, Kettledown Road, Southbury, Connecticut 06488-2600 (US). **DESIMONE, Michael Joseph** [US/US]; IBM Corporation, Mail Drop 342, 150, Kettledown Road, Southbury, Connecticut 06488-2600 (US). **TAN, Chek Keong** [US/US]; IBM Corporation, Mail Drop 223, 150, Kettletown Road, Southbury, Connecticut 06488-2600 (US). **RUS-SO, Stephen** [US/US]; IBM Corporation, 150 Kettletown Road, Southbury, Connecticut 06488-2600 (US). **MERCIER, Carl** [US/US]; IBM Corporation, Mail Drop 272, 150, Kettledown Road, Southbury, Connecticut 06488-2600 (US).(74) Agent: **ROBERTS, Scott**; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester Hampshire SO21 2JN (GB).

(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.

[Continued on next page]

(54) Title: CODING SCHEME FOR IDENTIFYING SPATIAL LOCATIONS OF EVENTS WITHIN VIDEO IMAGE DATA



(57) Abstract: An invention for generating a coding schema for identifying a spatial location of an event within video image data is provided. In one embodiment, there is a spatial representation tool, including a compression component configured to receive trajectory data of an event within video image data, generate a lossless compressed contour-coded blob to encode the trajectory data of the event within video image data, and generate a lossy searchable code to enable searching of a relational database based on the trajectory data of the event within the video image data.



(84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (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, SE, SI, SK, SM,

TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

**Published:**

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

## **CODING SCHEME FOR IDENTIFYING SPATIAL LOCATIONS OF EVENTS WITHIN VIDEO IMAGE DATA**

### **FIELD OF THE INVENTION**

5

The present invention generally relates to video surveillance, and more specifically to coding for spatial surveillance event searching.

### **BACKGROUND OF THE INVENTION**

10

Large surveillance networks that are deployed on buildings, highways, trains, metro stations, etc., integrate a large number of cameras, sensors, and information. Human operators typically cannot adequately control and monitor all the cameras within a large surveillance system. As such, many prior art approaches involve object detection and tracking techniques to identify and analyze events occurring within a camera field of view. However, when it comes to searching through large amounts of video data in an effort to identify an event within video image data, it is difficult to obtain reliable results.

15

20

25

30

For example, consider a surveillance camera that is monitoring a long-term parking lot. The parking lot attendant receives a complaint that a car has been vandalized at some point in the past month. The prior art requires either a manual review of tapes/files from the video camera for the entire month, or the use of a query box drawn around the particular parking spot with the surveillance system retrieving all movement that occurred in the query box. The first approach is typically ineffective because an operator or group of operators must review hundreds of hours of video to observe an event that may have lasted a few seconds. The second approach uses automatic video object tracking and meta-data indexing using a standard relational database to support spatial queries. However, the drawback of this approach is that the representation of the meta-data is very voluminous and makes the indexing of large numbers of cameras impractical due to the heavy volume of network traffic and the size of database tables created.

## SUMMARY OF THE INVENTION

In one embodiment, there is a method for providing a coding scheme for identifying a spatial location of an event within video image data. In this embodiment, the method comprises:  
5 receiving trajectory data of an event within video image data; generating a lossless compressed contour-coded binary large object (blob) to encode the trajectory data of the event within video image data; and generating a lossy searchable code to enable searching of a relational database based on the trajectory data of the event within the video image data.

10 In a second embodiment, there is a system for providing a coding scheme for identifying a spatial location of an event within video image data. In this embodiment, the system comprises at least one processing unit, and memory operably associated with the at least one processing unit. A spatial representation tool is storable in memory and executable by the at least one processing unit. The spatial representation tool comprises: a compression  
15 component configured to receive trajectory data of an event within video image data; generate a lossless compressed contour-coded blob to encode the trajectory data of the event within video image data; and generate a lossy searchable code to enable searching of a relational database based on the trajectory data of the event within the video image data.

20 In a third embodiment, there is a computer-readable medium storing computer instructions, which when executed, enables a computer system to provide a coding scheme for identifying a spatial location of an event within video image data, the computer instructions comprising: receiving trajectory data of an event within video image data; generating a lossless compressed contour-coded blob to encode the trajectory data of the event within video image  
25 data; and generating a lossy searchable code to enable searching of a relational database based on the trajectory data of the event within the video image data.

In a fourth embodiment, there is a method for deploying a spatial representation tool for use in a computer system that provides a coding scheme for identifying a spatial location of an  
30 event within video image data. In this embodiment, a computer infrastructure is provided and is operable to: receive trajectory data of an event within video image data; generate a lossless compressed contour-coded blob to encode the trajectory data of the event within

video image data; and generate a lossy searchable code to enable searching of a relational database based on the trajectory data of the event within the video image data.

## BRIEF DESCRIPTION OF THE DRAWINGS

5

Preferred embodiments of the present invention will now be described, by way of example only, with reference to the following drawings in which:

FIG. 1 shows a schematic of an exemplary computing environment in which elements of the present invention may operate;

10

FIG. 2 shows a spatial representation tool that operates in the environment shown in FIG. 1;

FIG. 3 shows a system for searching within video image data according to embodiments of the invention;

15

FIG. 4 shows an approach for lossless contour coding generation according to embodiments of the invention;

FIG. 5 shows an approach for lossy search code generation according to embodiments of the invention;

FIG. 6 shows an approach for identifying an event within the video image data according to embodiments of the invention; and

20

FIG. 7 shows a flow diagram of a method for searching within the video image data according to embodiments of the invention.

The drawings are not necessarily to scale. The drawings are merely schematic representations, not intended to portray specific parameters of the invention. The drawings are intended to depict only typical embodiments of the invention, and therefore should not be considered as limiting the scope of the invention. In the drawings, like numbering represents like elements.

25

## DETAILED DESCRIPTION OF THE INVENTION

30

Embodiments of this invention are directed to a coding scheme that enables searching large numbers of surveillance camera events using relational database tables based on the location

of an event within a camera field of view. In these embodiments, a spatial representation tool provides this capability. Specifically, the spatial representation tool comprises a compression component configured to receive trajectory data of an event within video image data; generate a lossless compressed contour-coded binary large object (blob) to encode the trajectory data of the event within video image data; and generate a lossy searchable code to enable searching of a relational database based on the trajectory data of the event within the video image data.

FIG. 1 illustrates a computerized implementation 100 of the present invention. As depicted, implementation 100 includes computer system 104 deployed within a computer infrastructure 102. This is intended to demonstrate, among other things, that embodiments of the present invention could be implemented within a network environment (e.g., the Internet, a wide area network (WAN), a local area network (LAN), a virtual private network (VPN), etc.), or on a stand-alone computer system. In the case of the former, communication throughout the network can occur via any combination of various types of communications links. For example, the communication links can comprise addressable connections that may utilize any combination of wired and/or wireless transmission methods. Where communications occur via the Internet, connectivity could be provided by conventional TCP/IP sockets-based protocol, and an Internet service provider could be used to establish connectivity to the Internet. Still yet, computer infrastructure 102 is intended to demonstrate that some or all of the components of implementation 100 could be deployed, managed, serviced, etc., by a service provider who offers to implement, deploy, and/or perform the functions of embodiments of the present invention for others.

Computer system 104 is intended to represent any type of computer system that may be implemented in deploying/realizing the teachings recited herein. In this particular example, computer system 104 represents an illustrative system for generating a coding scheme for identifying a spatial location of an event in video image data. It should be understood that any other computers implemented in accordance with embodiments of the present invention may have different components/software, but will perform similar functions. As shown, computer system 104 includes a processing unit 106 capable of analyzing sensor data, and producing a usable output, e.g., compressed video and video meta-data. Also shown is

memory 108 for storing a spatial representation tool 153, a bus 110, and device interfaces 112.

Computer system 104 is shown communicating with a sensor device 122 that communicates with bus 110 via device interfaces 112. Sensor device 122 (or multiple sensor devices) includes sensor devices for capturing image data representing objects and visual attributes of moving objects (e.g., people, cars, animals, products, etc.) within a camera view 119 from sensor device 122, including trajectory data 121 and 123 (i.e., paths of events/objects within video image data 119). Sensor device 122 can include virtually any type of sensor capable of capturing visual attributes of objects, such as, but not limited to: optical sensors, infrared detectors, thermal cameras, still cameras, analog video cameras, digital video cameras, or any other similar device that can generate sensor data of sufficient quality to support the methods in accordance with embodiments of of invention as described herein.

Processing unit 106 collects and routes signals representing outputs from sensor devices 122 to spatial representation tool 153. The signals can be transmitted over a LAN and/or a WAN (e.g., T1, T3, 56 kb, X.25), broadband connections (ISDN, Frame Relay, ATM), wireless links (802.11, Bluetooth, etc.), and so on. In some embodiments, the video signals may be encrypted using, for example, trusted key-pair encryption. Different sensor systems may transmit information using different communication pathways, such as Ethernet or wireless networks, direct serial or parallel connections, USB, Firewire®, Bluetooth®, or other proprietary interfaces. (Firewire is a registered trademark of Apple Computer, Inc. Bluetooth is a registered trademark of Bluetooth Special Interest Group (SIG)). In some embodiments, sensor device 122 is capable of two-way communication, and thus can receive signals (to power up, to sound an alert, etc.) from spatial representation tool 153.

In general, processing unit 106 executes computer program code, such as program code for operating spatial representation tool 153, which is stored in memory 108 and/or storage system 116. While executing computer program code, processing unit 106 can read and/or write data to/from memory 108 and storage system 116 and a relational database 118.

Relational database 118 stores sensor data, including video metadata generated by processing unit 106, as well as rules against which the metadata is compared to identify objects and trajectories of objects present within video image data 119. As will be further

described herein, relational database 118 stores trajectory data 117 as both a lossy searchable code and lossless compressed contour-coded blob, as well as information for efficient querying. It will be appreciated that storage system 116 and relational database 118 can include VCRs, DVRs, RAID arrays, USB hard drives, optical disk recorders, flash storage devices, image analysis devices, general purpose computers, video enhancement devices, de-interlacers, scalers, and/or other video or data processing and storage elements for storing and/or processing video. The video signals can be captured and stored in various analog and/or digital formats, including, but not limited to, Nation Television System Committee (NTSC), Phase Alternating Line (PAL), and Sequential Color with Memory (SECAM), uncompressed digital signals using DVI or HDMI connections, and/or compressed digital signals based on a common codec format (e.g., MPEG, MPEG2, MPEG4, or H.264).

FIG. 2 shows a more detailed view of spatial representation tool 153 according to embodiments of the invention. As shown, spatial representation tool 153 comprises a compression component 155 configured to receive trajectory data 117 of an event within video image data 119 (e.g., object and track data from sensor device 122). Compression component 155 processes trajectory data 117 from sensor device 122 in real-time, identifying objects and trajectories of objects that are detected in video image data 119. Compression component 155 provides the software framework for hosting a wide range of video analytics to accomplish this. The video analytics are intended to detect and track objects moving across a field of view and perform an analysis of tracking data associated with each object. The set of moving objects can be detected using a number of approaches, including but not limited to: background modeling, object detection and tracking, spatial intensity field gradient analysis, diamond search block-based (DSBB) gradient descent motion estimation, or any other method for detecting and identifying objects captured by a sensor device.

As shown in FIGS. 2-3, compression component 155 is configured to receive trajectory data 117 of video image data 119 and generate a lossless compressed contour-coded blob 134 to encode trajectory data 117 of the event within video image data 119. Compression component 155 is also configured to generate a lossy searchable code 134 to enable

searching of relational database 118 based on the trajectory data 117 of the event within the video image data 119.

5 Next, both lossy searchable code 132 and lossless compressed contour-coded blob 134 are stored within relational database 118, along with the corresponding track ID, for subsequent retrieval. As shown in FIG. 2-3, spatial representational tool 153 comprises a database component 160 configured to input lossless compressed contour-coded blob 134, lossy searchable code 132, and a corresponding trajectory identifier (e.g., track ID) into relational database 118. In one embodiment, database component 160 generates and uploads messages  
10 in extensible mark-up language (XML) to relational database 118 including Track ID, search code represented as a CHAR String, and contour code packaged as a proprietary file with binary representation.

15 During operation, retrieval may occur when a user that is monitoring video image data 119 wishes to investigate an event (e.g., a person, a security breach, a criminal act, suspicious activity, etc.). As shown in FIGS. 2-3, spatial representation tool 153 comprises a search component 165 configured to search relational database 118 to identify a spatial location of the event within video image data 119. Specifically, search component 165 is configured to specify a region of interest 140 (FIG. 3) within video image data 119. This selection may be  
20 performed by the user monitoring video image data 119, e.g., via a pointing device (not shown). Search component 165 then converts region of interest 140 to a lossy query code 136 and performs a database search of relational database 118. Specifically, search component 165 compares lossy query code 136 to lossy searchable code 132 of trajectory data 117 of the event within video image data 119. In one embodiment, each row of  
25 relational database 118 is evaluated using a 'UDF→C Function' for performing 'BITWISE AND' between lossy query code 136 and lossy searchable code 132 corresponding to each track in the table. All rows that intersect region of interest 140 are returned as part of the result set to identify the spatial location of the event.

30 The result set is then typically returned to the user as a display 148 (e.g., via a graphical user interface). To accomplish this, spatial representation tool 153 comprises a display component 170 (FIG. 2) configured to decompress contour-coded blob 134 corresponding to

lossy query code 136 based on the comparison of lossy query code 136 to lossy searchable code 132 of trajectory data 117 of the event within video image data 119. Contour-coded blob 134 is converted back to the original version of trajectory data 117 and displayed on display 148. Display component 170 plots a trajectory (147, 149) of the event within video image data 119 to identify the spatial location of the event.

Referring now to FIGS. 3-6, a coding scheme for identifying a spatial location of an event within video image data 119 will be described in further detail. As mentioned above, compression component 155 (FIG. 2) is configured to generate a lossy searchable code 132 of trajectory data 117 of the event within video image data 119, and a lossless compressed contour-coded blob 134 of trajectory data 117 of the event within video image data 119. As shown in FIG. 4, in the first case, compression component 150 is configured to receive trajectory data 117 of event "X" (e.g., a person, a security breach, a criminal act, suspicious activity, etc.) within video image data 119, and generate a contour-coded blob 134 from lossless contour code 131 (FIG. 3) to encode trajectory 121 of event "X". To accomplish this, compression component 155 is configured to divide video image data 119 into a plurality of pixel regions 23A, 23B, 23C,...23N, determine whether each of plurality of pixel regions 23A-23N contains trajectory data 117. That is, each pixel is analyzed to determine if trajectory 121 intersects the pixel. If yes, a '1' is entered into 36 bit contour-coded blob 134. If trajectory 121 does not intersect the pixel, '0' is entered. This process is repeated until contour-coded blob 134 is complete and is entered into relational database 118.

Next, as shown in FIG. 5, a lossy searchable code 132 of trajectory data 117 of the event within video image data 119 is generated. To accomplish this, compression component 155 is configured to divide video image data 119 into a second plurality of pixel regions 25A, 25B, 25C,...25N. As shown, second plurality of pixel regions 25A-25N comprises less pixel regions than plurality of pixel regions 23A-23N for contour coded blob 134. In this case, the 6x6 representation of video image data 119 is quantized into a 3x3 image, thus generating 9 bit lossy searchable code 132. Once again, to encode trajectory data 117, it is determined whether each of second plurality of pixel regions 25A-25N contains trajectory data 117. That is, each pixel is analyzed to determine if trajectory 121 intersects the pixel.

If trajectory 121 intersects, a '1' is entered to form 9 bit lossy searchable code 132. If trajectory 121 does not intersect the pixel, a '0' is entered. This process is repeated until lossy searchable code 132 is formed, and lossy searchable code 132 is then entered into relational database 118 to enable subsequent searching based on trajectory data 117 of event "X" within video image data 119.

Next, as shown in FIG. 6, trajectory data 117 of trajectory 121 is more precisely analyzed. In this embodiment, video image data 119 is analyzed using an 8-point neighborhood scan 180 to generate the transition chain code. As shown, event "X" starts at point (0,1), and the direction of trajectory 121 is plotted according to 8-point neighborhood scan 180. This embodiment allows increased specificity over the 6x6 image shown in FIG. 4. Rather than simply identifying whether trajectory 121 is present within each pixel, 8-point neighborhood scan provides information on a direction of trajectory 121 within each pixel. It will be appreciated that the precision may be adjusted by increasing or decreasing the number of points in the neighborhood scan.

It can be appreciated that the methodologies disclosed herein can be used within a computer system to identify a spatial location of an event within video image data, as shown in FIG. 1. In this case, spatial representation tool 153 can be provided, and one or more systems for performing the processes described in accordance with embodiments of the invention can be obtained and deployed to computer infrastructure 102. To this extent, the deployment can comprise one or more of (1) installing program code on a computing device, such as a computer system, from a computer-readable medium; (2) adding one or more computing devices to the infrastructure; and (3) incorporating and/or modifying one or more existing systems of the infrastructure to enable the infrastructure to perform the process actions in accordance with embodiments of the invention.

The exemplary computer system 104 may be described in the general context of computer-executable instructions, such as program modules, being executed by a computer.

Generally, program modules include routines, programs, people, components, logic, data structures, and so on that perform particular tasks or implements particular abstract data types. Exemplary computer system 104 may be practiced in distributed computing

environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices.

5

The program modules carry out the methodologies disclosed herein, as shown in FIG. 7. According to one embodiment for generating a coding scheme, at 202, trajectory data of an event within video image data is received. At 204, a lossy searchable code of the trajectory data of the event within the video image is generated. At 204B, a lossless compressed contour-coded blob of the trajectory data of the event within the video image data is generated. At 206, lossless compressed contour-coded blob and the lossy searchable code are entered into the relational database.

10

15

20

The flowchart of FIG. 7 illustrates the architecture, functionality, and operation of possible implementations of systems, methods and computer program products according to various embodiments of the present invention. In this regard, each block in the flowchart may represent a module, segment, or portion of code, which comprises one or more executable instructions for implementing the specified logical function(s). It should also be noted that, in some alternative implementations, the functions noted in the blocks may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently. It will also be noted that each block of flowchart illustration can be implemented by special purpose hardware-based systems that perform the specified functions or acts, or combinations of special purpose hardware and computer instructions.

25

Furthermore, an implementation of exemplary computer system 104 (FIG. 1) may be stored on or transmitted across some form of computer readable media. Computer readable media can be any available media that can be accessed by a computer. By way of example, and not limitation, computer readable media may comprise "computer storage media" and "communications media."

30

"Computer storage media" include volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules, or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by a computer.

"Communication media" typically embodies computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as carrier wave or other transport mechanism. Communication media also includes any information delivery media.

The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared, and other wireless media. Combinations of any of the above are also included within the scope of computer readable media.

It is apparent that there has been provided, in accordance with embodiments of the present invention, an approach for identifying a spatial location of an event within video image data.

## CLAIMS

1. A method for generating a coding scheme for identifying a spatial location of an event within video image data comprising:

5 receiving trajectory data of an event within video image data;  
generating a lossless compressed contour-coded binary large object (blob) to encode the trajectory data of the event within video image data; and  
generating a lossy searchable code to enable searching of a relational database based on the trajectory data of the event within the video image data

10 2. The method according to claim 1, the generating the lossless compressed contour-coded blob comprising:  
dividing the video image data into a plurality of pixel regions; and  
determining whether each of the plurality of pixel regions contains trajectory data of  
15 the event within the video image data.

3. The method according to claim 2, the generating the lossy searchable code comprising:  
dividing the video image data into a second plurality of pixel regions, the second  
20 plurality of pixel regions comprising less pixel regions than the plurality of pixel regions for the lossless compressed contour-coded blob; and  
determining whether each of the second plurality of pixel regions contains trajectory data of the event within the video image data.

25 4. The method according to claim 1, further comprising inputting the lossless compressed contour-coded blob, the lossy searchable code, and a trajectory identifier into the relational database.

30 5. The method according to claim 4, further comprising:  
specifying a region of interest within the video image data;  
converting the region of interest within the video image data to a lossy query code;  
and

comparing the lossy query code to the lossy searchable code within the relational database.

6. A system for generating a coding scheme for identifying a spatial location of an event within video image data comprising:

at least one processing unit;

memory operably associated with the at least one processing unit; and

a spatial representation tool storable in memory and executable by the at least one processing unit, the spatial representation tool comprising a compression component configured to:

receive trajectory data of an event within video image data; and

generate a lossless compressed contour-coded binary large object (blob) to encode the trajectory data of the event within video image data; and

generate a lossy searchable code to enable searching of a relational database based on the trajectory data of the event within the video image data.

7. The spatial representation tool according to claim 6, the compression component further configured to:

divide the video image data into a plurality of pixel regions; and

determine whether each of the plurality of pixel regions contains trajectory data of the event within the video image data.

8. The spatial representation tool according to claim 7, the compression component further configured to:

divide the video image data into a second plurality of pixel regions, the second plurality of pixel regions comprising less pixel regions than the plurality of pixel regions for the lossless compressed contour-coded blob; and

determine whether each of the second plurality of pixel regions contains trajectory data of the event within the video image data.

9. The spatial representation tool according to claim 6, further comprising an input component configured to input the lossless compressed contour-coded blob, the lossy

searchable code, and a trajectory identifier into the relational database.

10. The spatial representation tool according to claim 9, further comprising a search component configured to:

5 specify a region of interest within the video image data;  
convert the region of interest within the video image data to a lossy query code; and  
compare the lossy query code to the lossy searchable code within the relational database.

10 11. A computer-readable medium storing computer instructions, which when executed, enables a computer system to generate a coding scheme for identifying a spatial location of an event within video image data, the computer instructions comprising:

receiving trajectory data of an event within video image data;  
generating a lossless compressed contour-coded binary large object (blob) to encode  
15 the trajectory data of the event within video image data; and  
generating a lossy searchable code to enable searching of a relational database based on the trajectory data of the event within the video image data.

20 12. The computer-readable medium according to claim 11, the computer instructions for generating the lossless compressed contour-coded blob further comprising:

dividing the video image data into a plurality of pixel regions; and  
determining whether each of the plurality of pixel regions contains trajectory data of the event within the video image data.

25 13. The computer-readable medium according to claim 12, the computer instructions for generating the lossy searchable code further comprising:

dividing the video image data into a second plurality of pixel regions, the second plurality of pixel regions comprising less pixel regions than the plurality of pixel regions for the lossless compressed contour-coded blob; and

30 determining whether each of the second plurality of pixel regions contains trajectory data of the event within the video image data.

14. The computer-readable medium according to claim 11 further computer instructions for inputting the lossless compressed contour-coded blob, the lossy searchable code, and a trajectory identifier into the relational database.

5 15. The computer-readable medium according to claim 14, the computer instructions further comprising:

specifying a region of interest within the video image data;

converting the region of interest within the video image data to a lossy query code;

and

10 comparing the lossy query code to the lossy searchable code within the relational database.

16. A method for deploying a spatial representation tool for use in a computer system that generates a coding scheme for identifying a spatial location of an event within video  
15 image data, the method comprising:

providing a computer infrastructure operable to:

receive trajectory data of an event within video image data; and

generate a lossless compressed contour-coded binary large object (blob) to encode  
the trajectory data of the event within video image data; and

20 generate a lossy searchable code to enable searching of a relational database based on the trajectory data of the event within the video image data.

17. The method according to claim 16, the computer infrastructure operable to generate the lossless compressed contour-coded blob further operable to:

25 divide the video image data into a plurality of pixel regions; and

determine whether each of the plurality of pixel regions contains trajectory data of the event within the video image data.

18. The method according to claim 17, the computer infrastructure operable to generate  
30 the lossy searchable code further operable to:

divide the video image data into a second plurality of pixel regions, the second plurality of pixel regions comprising less pixel regions than the plurality of pixel regions for

the lossless compressed contour-coded blob; and

determine whether each of the second plurality of pixel regions contains trajectory data of the event within the video image data.

5        19.     The method according to claim 16, the computer infrastructure further operable to input the lossless compressed contour-coded blob, the lossy searchable code, and a trajectory identifier into the relational database.

10        20.     The method according to claim 19, the computer infrastructure further operable to:  
specify a region of interest within the video image data;  
convert the region of interest within the video image data to a lossy query code; and  
compare the lossy query code to the lossy searchable code within the relational database.

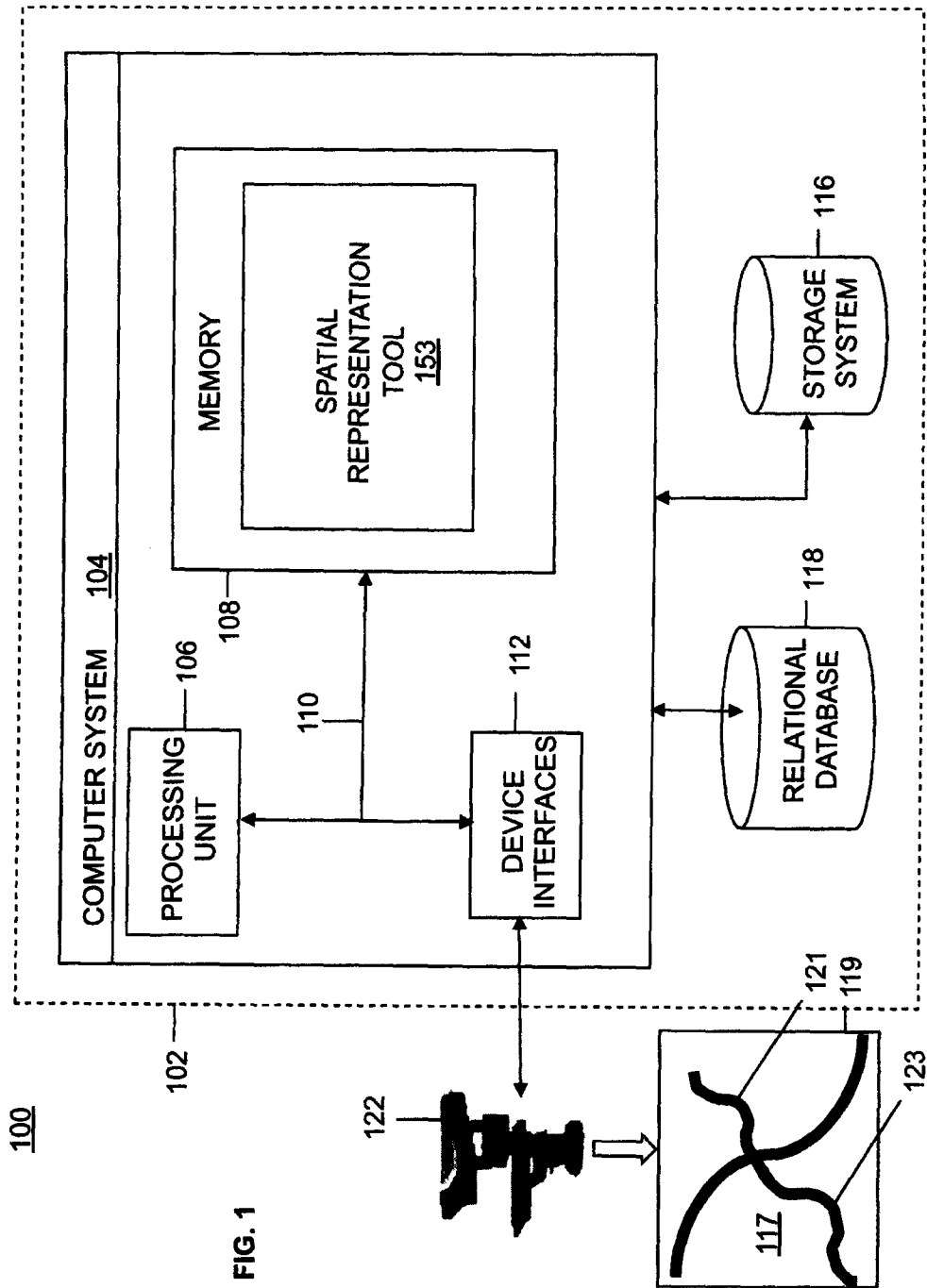
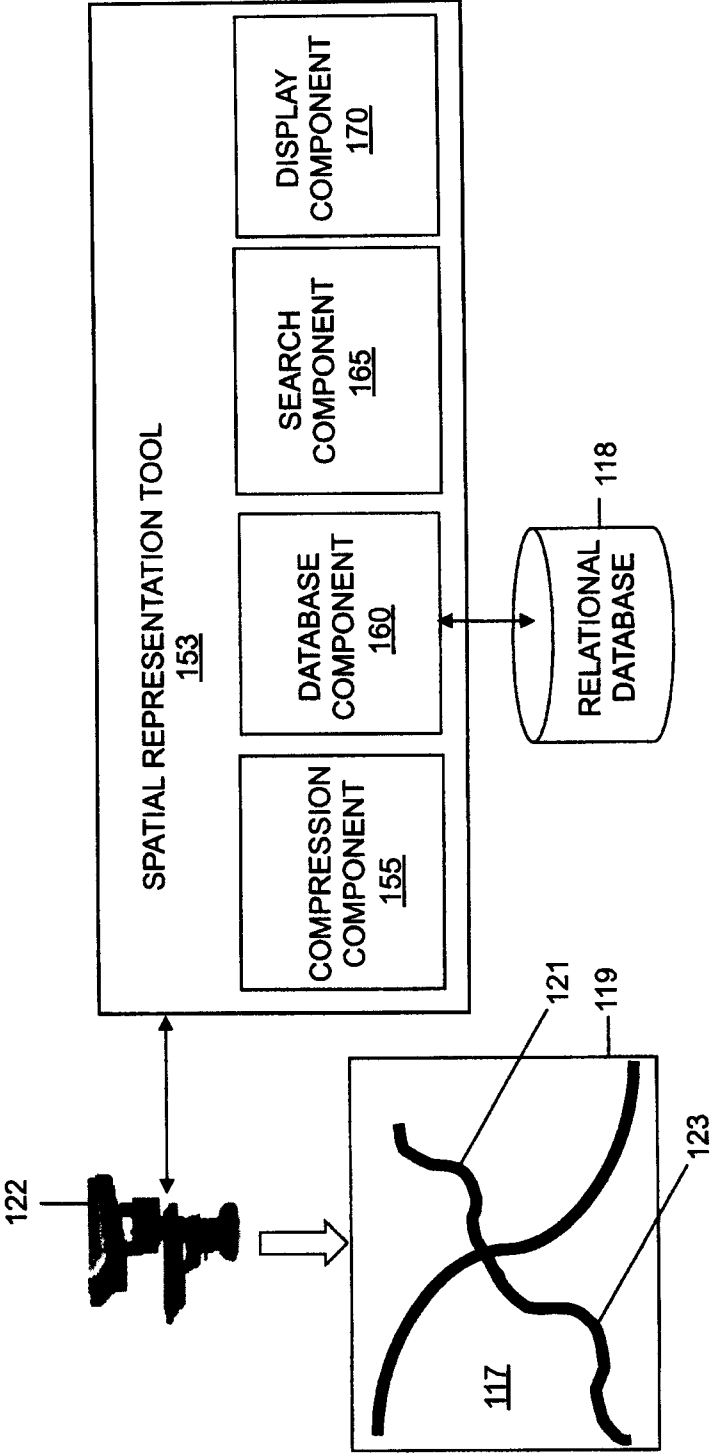


FIG. 2



3/7

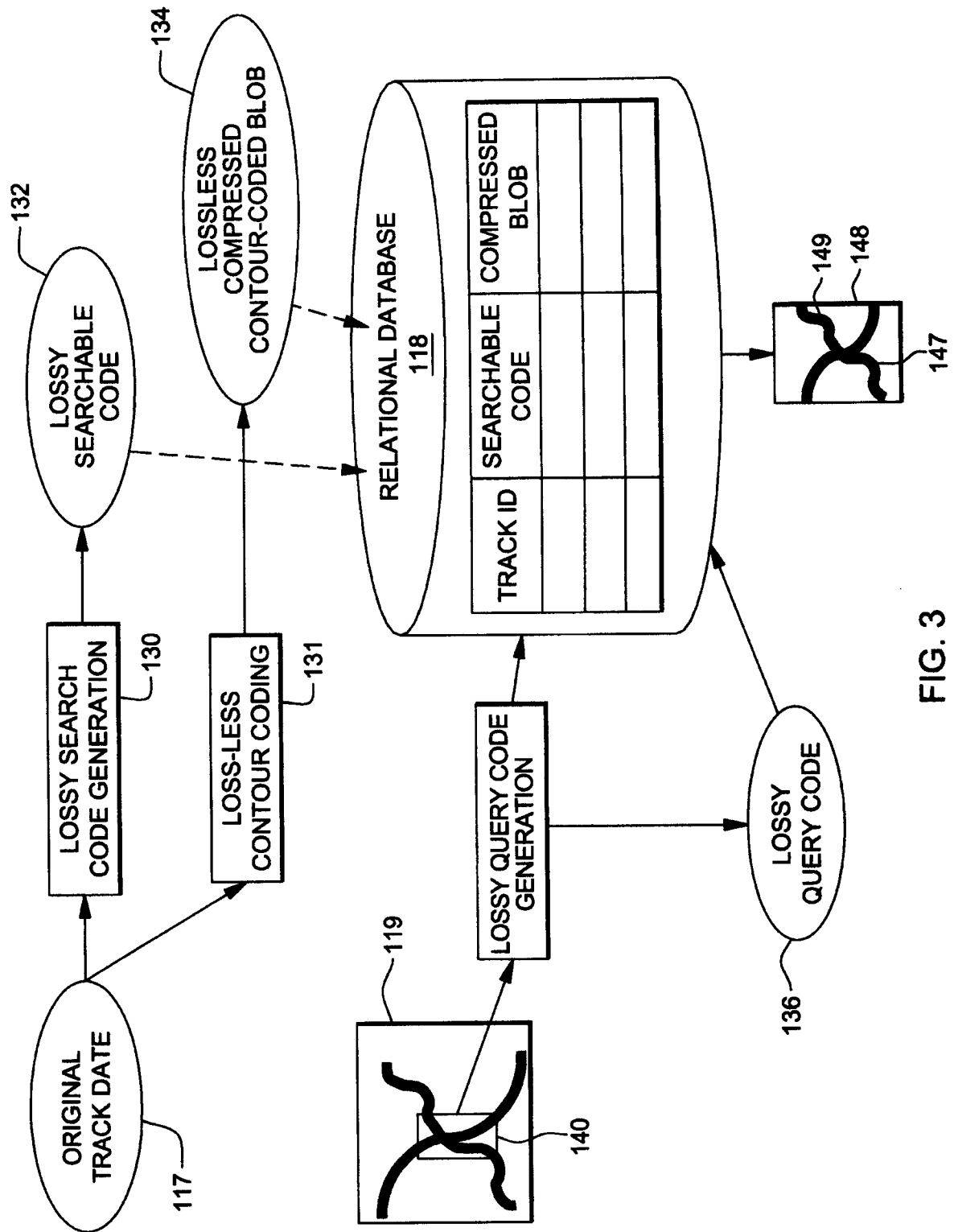
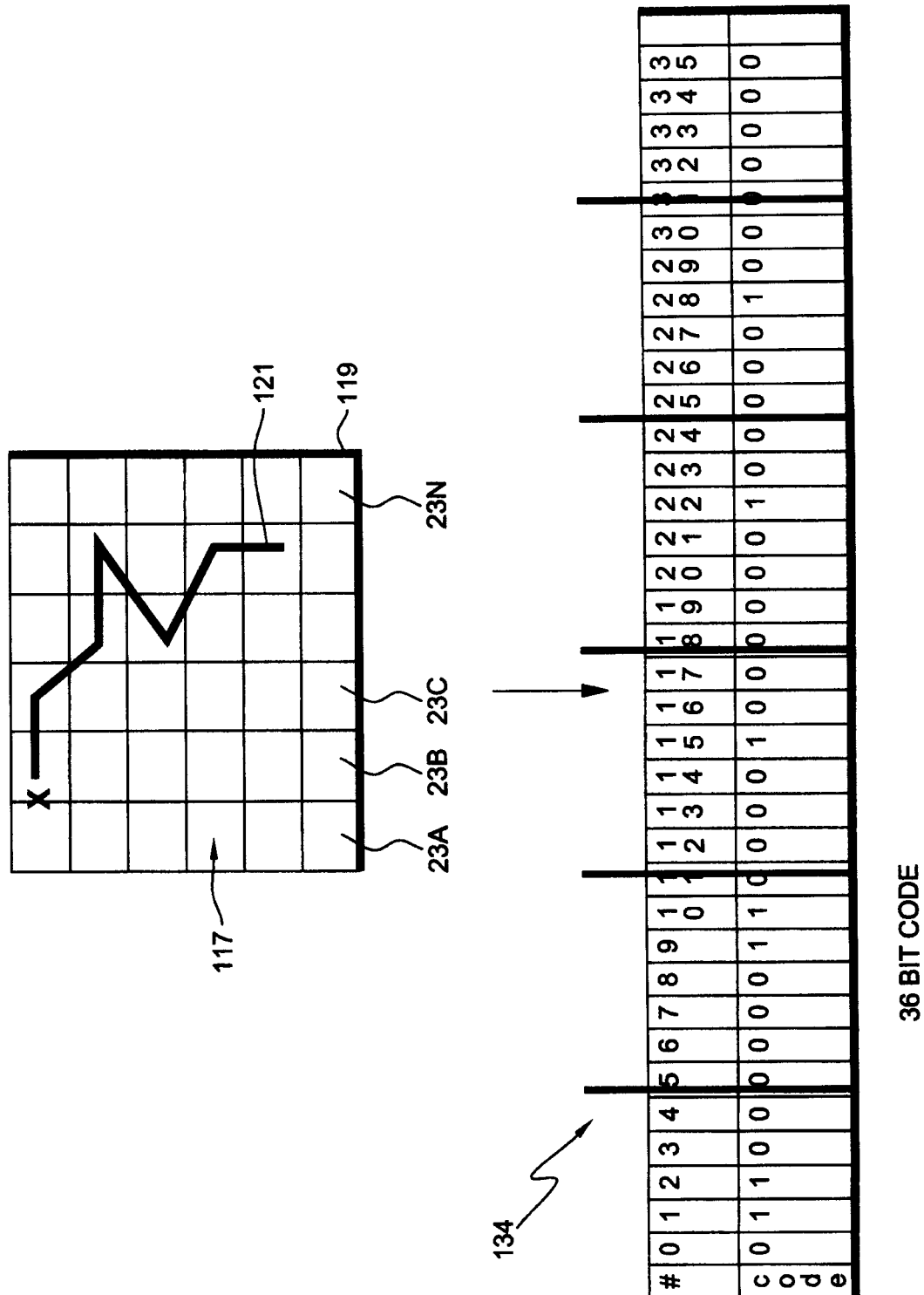


FIG. 3



**FIG. 4**

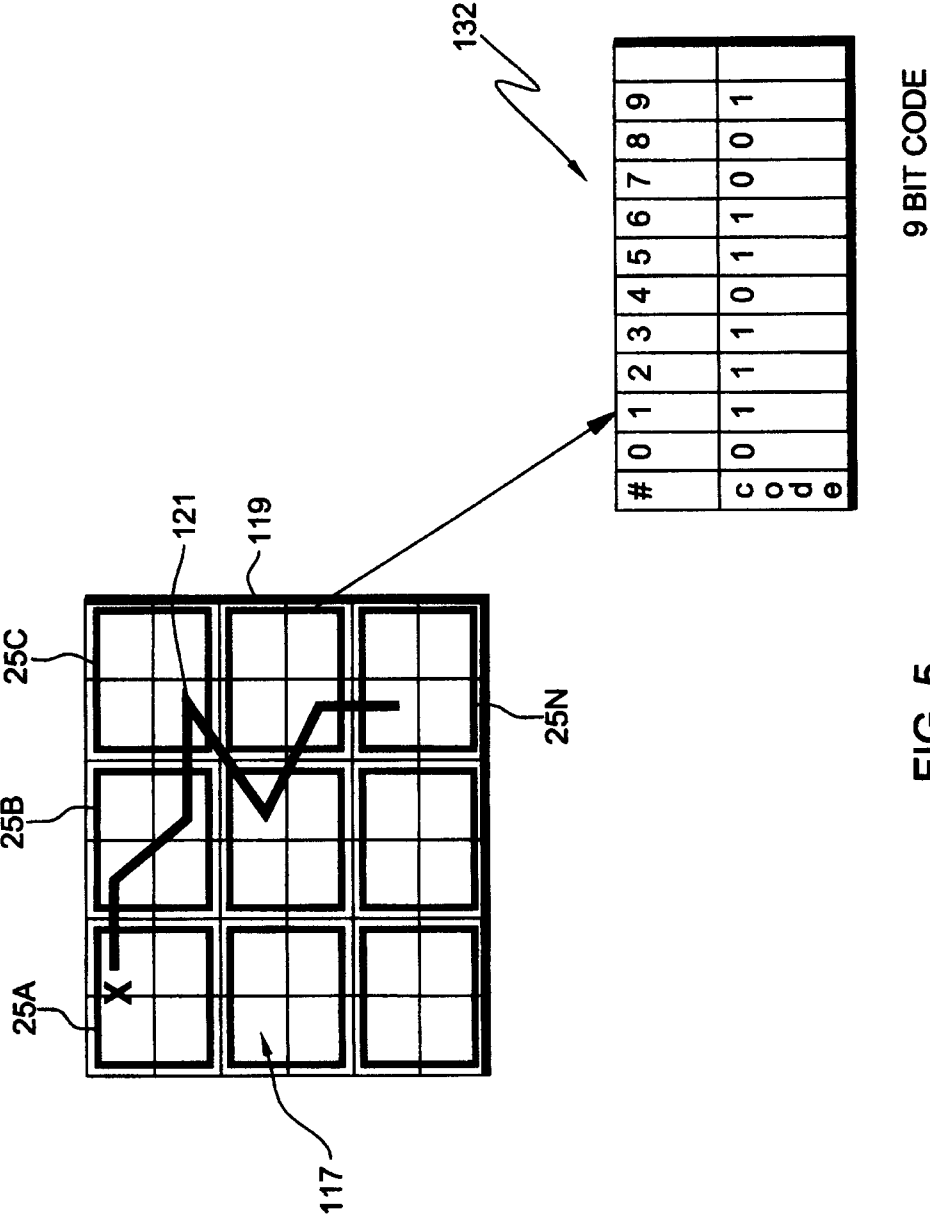
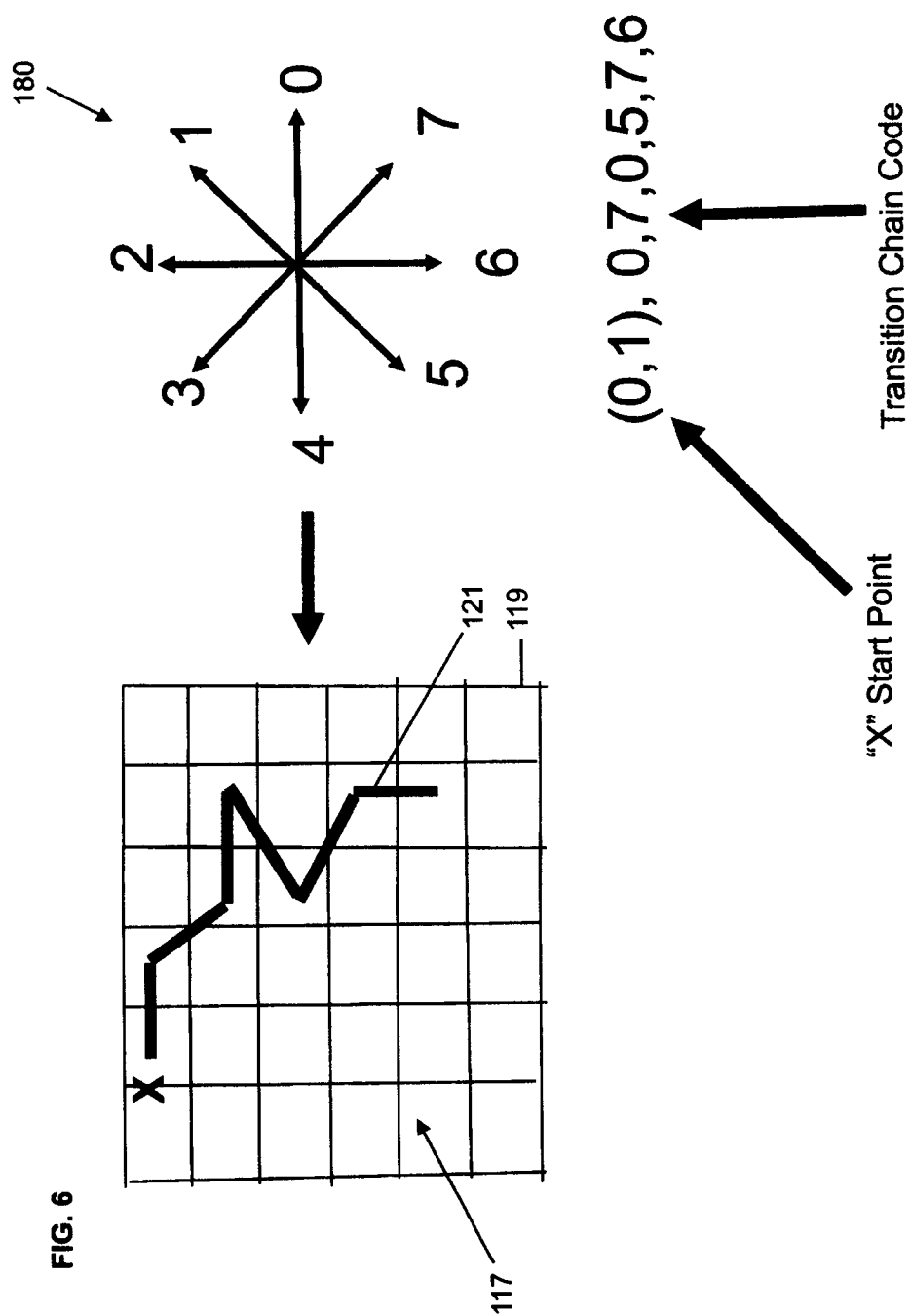


FIG. 5

6/7



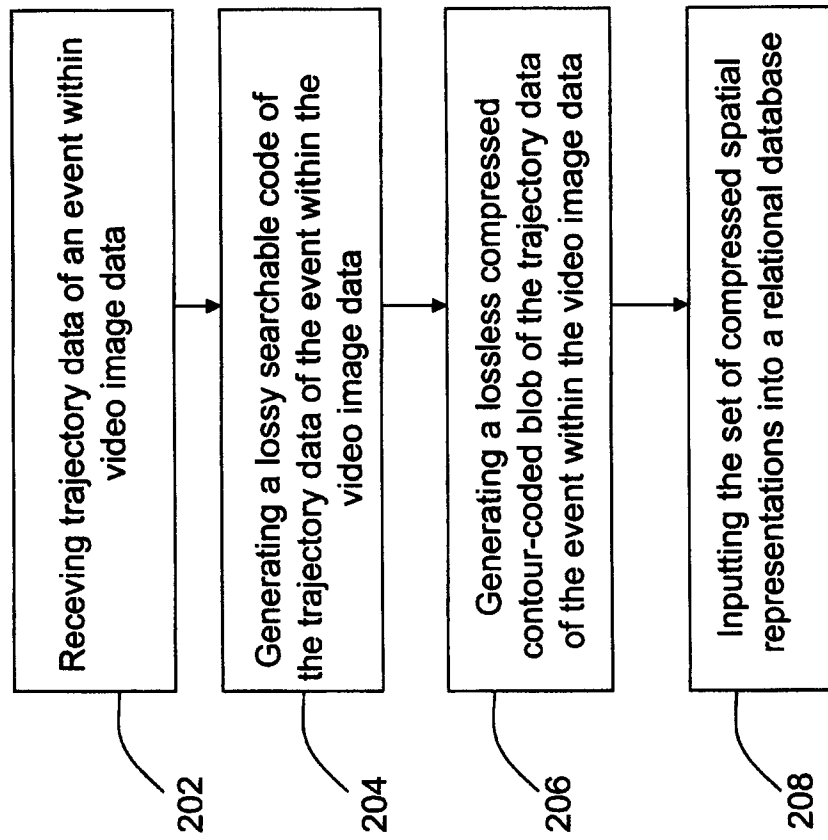


FIG. 7