US 20100178966A1

(19) **United States**

(12) **Patent Application Publication**

Seydoux

(10) **Pub. No.: US 2010/0178966 A1**

(43) **Pub. Date: Jul. 15, 2010**

(54) **A METHOD OF RECOGNIZING OBJECTS IN A SHOOTER GAME FOR REMOTE-CONTROLLED TOYS**

(75) Inventor: **Henri Seydoux**, Paris (FR)

Correspondence Address:
HAVERSTOCK & OWENS LLP
162 N WOLFE ROAD
SUNNYVALE, CA 94086 (US)

(73) Assignee: **PARROT**, Paris, FR (FR)

(21) Appl. No.: **12/526,933**

(22) PCT Filed: **Feb. 13, 2008**

(86) PCT No.: **PCT/FR08/00180**

§ 371 (c)(1),
(2), (4) Date: **Mar. 2, 2010**

(30) **Foreign Application Priority Data**

Feb. 13, 2007 (FR) 07 00998

Publication Classification

(51) **Int. Cl.**
A63F 9/24 (2006.01)

(52) **U.S. Cl.** **463/2**; 463/30

(57) **ABSTRACT**

The invention relates to a method of recognizing objects for a video shooter game, the system comprising a first remote-controlled vehicle (**51**) having an on-board video camera (**25**), a second remote-controlled vehicle (**53**), and an electronic video display entity serving to remote control the first vehicle (**51**).

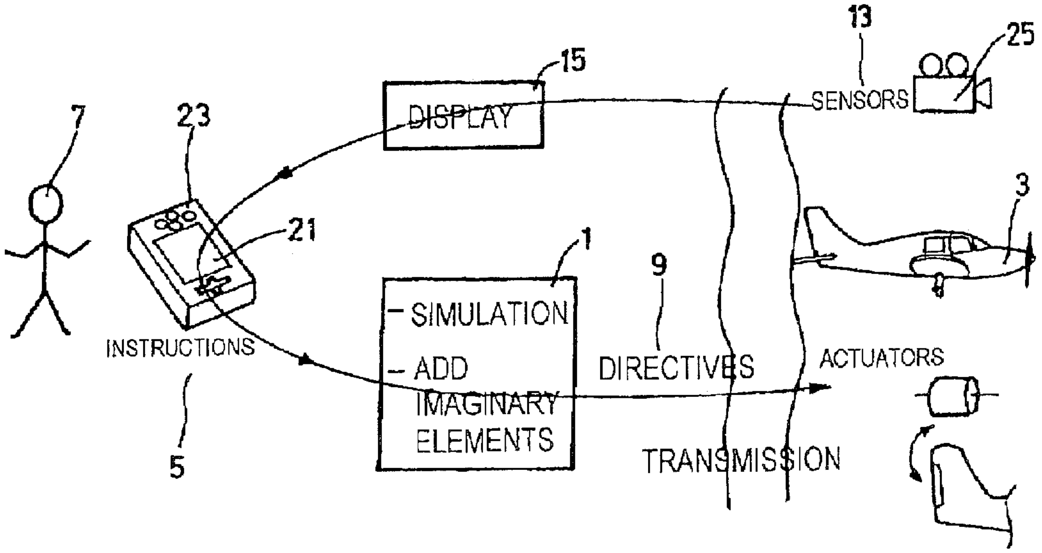


FIG_1a

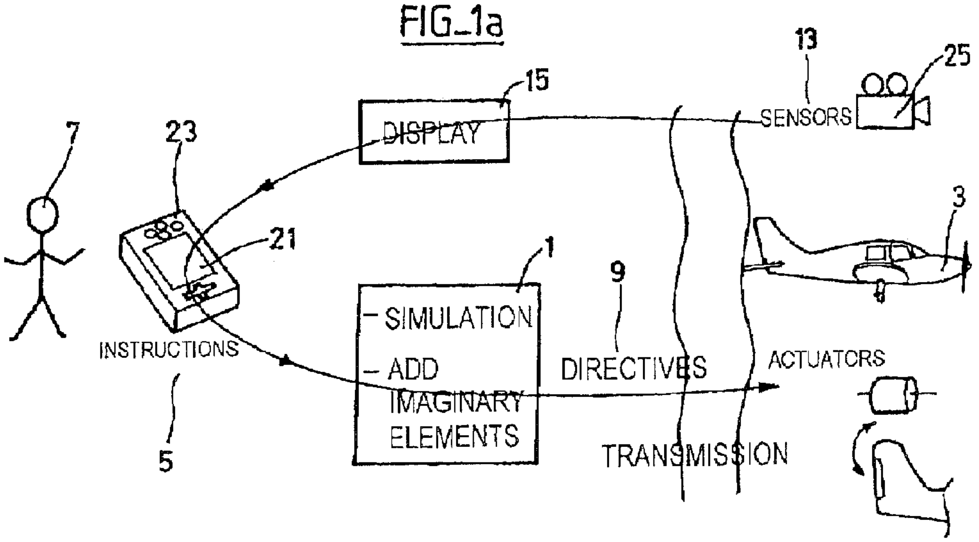
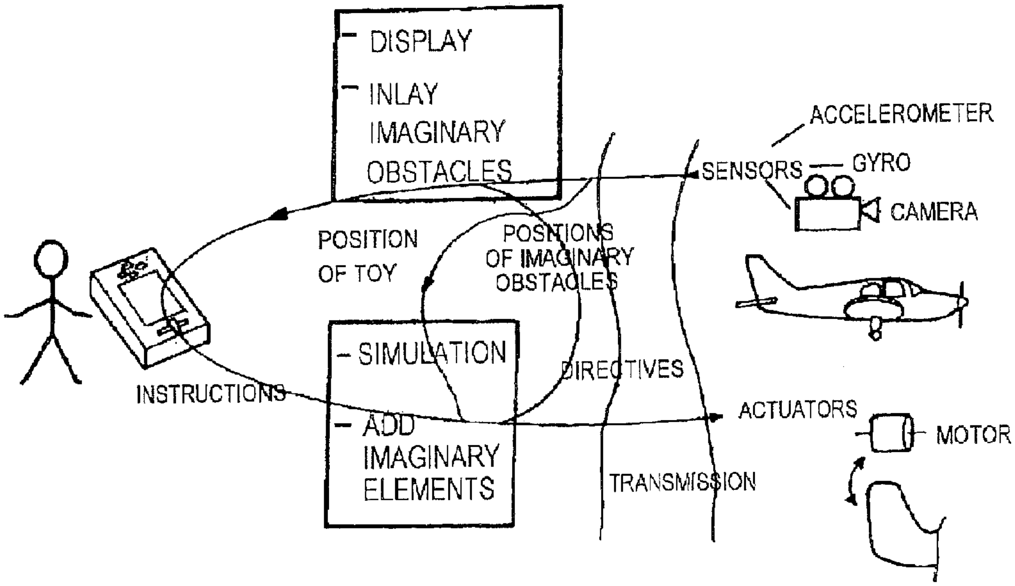

13
25
15
SENSORS
DISPLAY
7
23
3
21
1
SIMULATION
ADD IMAGINARY ELEMENTS
9
DIRECTIVES
ACTUATORS
INSTRUCTIONS
5
TRANSMISSION

FIG_1b


DISPLAY
INLAY IMAGINARY OBSTACLES
ACCELEROMETER
SENSORS
GYRO
CAMERA
POSITION OF TOY
POSITIONS OF IMAGINARY OBSTACLES
SIMULATION
DIRECTIVES
INSTRUCTIONS
ADD IMAGINARY ELEMENTS
ACTUATORS
MOTOR
TRANSMISSION

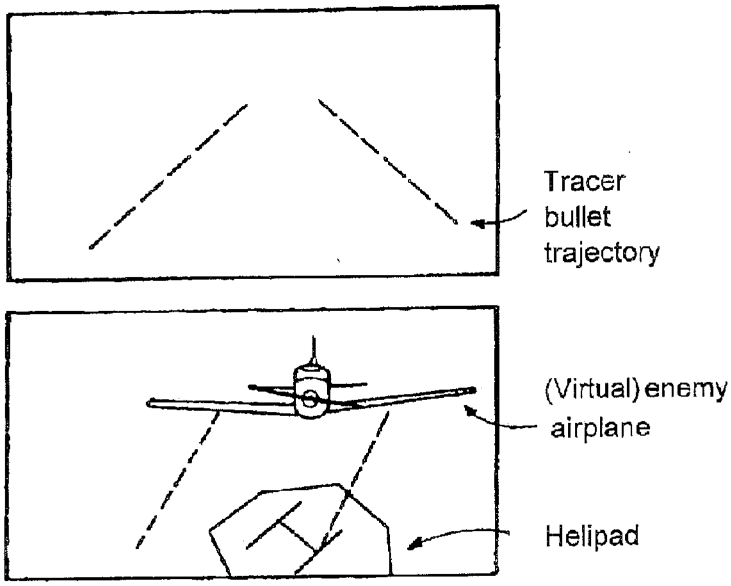

Tracer
bullet
trajectory
(Virtual) enemy
airplane
Helipad

FIG_1C

FIG_1d

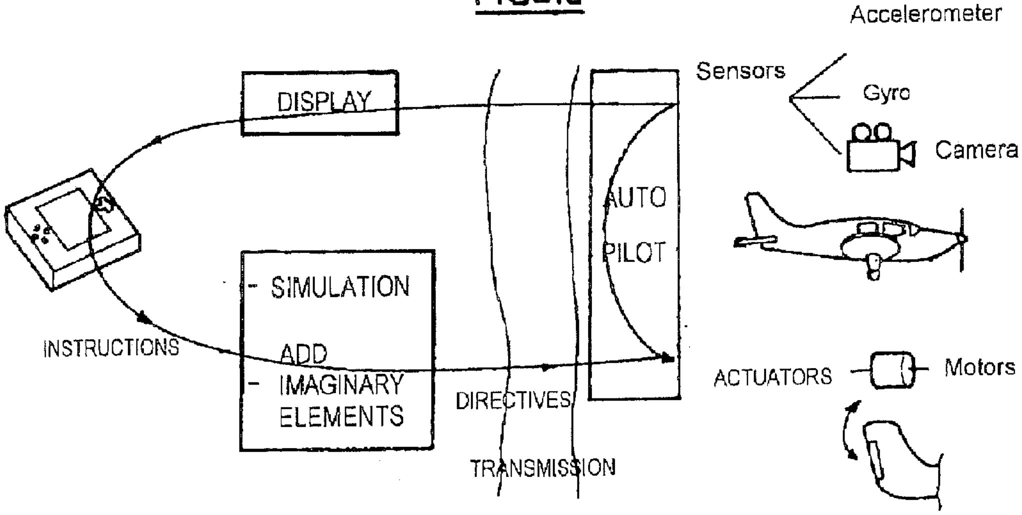

DISPLAY
- SIMULATION
ADD
- IMAGINARY
ELEMENTS
INSTRUCTIONS
DIRECTIVES
TRANSMISSION
AUTO
PILOT
Sensors
Accelerometer
Gyro
Camera
ACTUATORS
Motors

FIG_2a

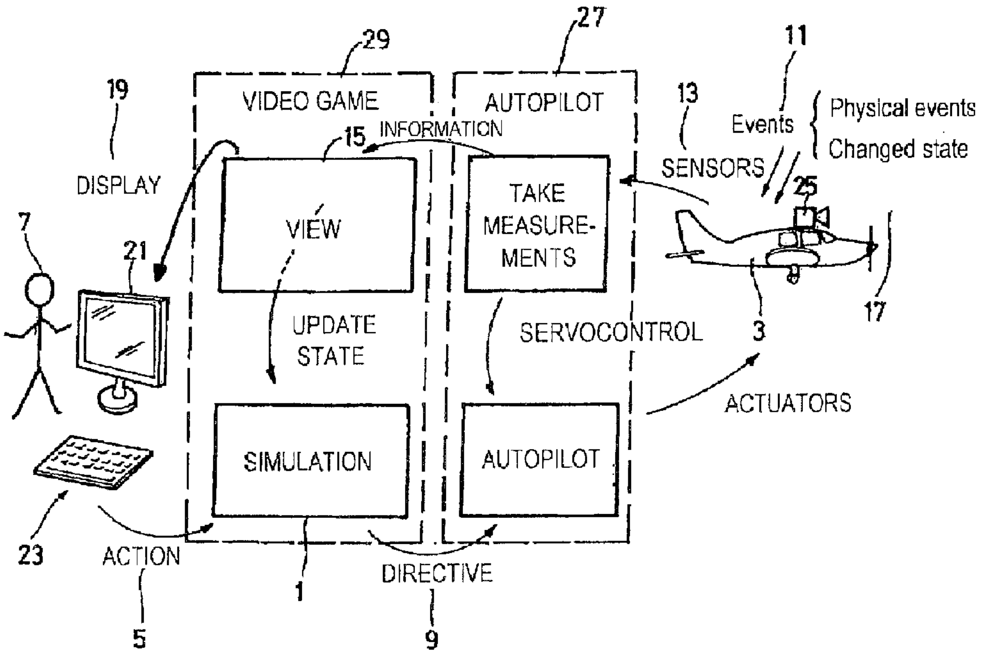
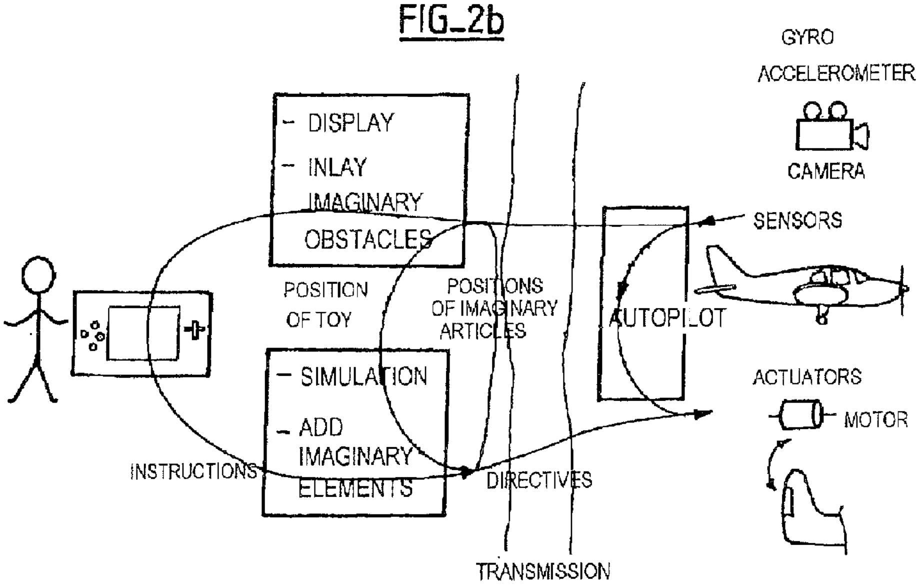

29
27
VIDEO GAME
AUTOPILOT
19
13
11
Events
Physical events
Changed state
INFORMATION
15
DISPLAY
SENSORS
TAKE MEASURE-MENTS
25
VIEW
7
21
17
UPDATE STATE
SERVOCONTROL
3
ACTUATORS
SIMULATION
AUTOPILOT
23
ACTION
1
DIRECTIVE
5
9

FIG_2b


GYRO
ACCELEROMETER
CAMERA
- DISPLAY
- INLAY IMAGINARY OBSTACLES
SENSORS
POSITION OF TOY
POSITIONS OF IMAGINARY ARTICLES
AUTOPILOT
ACTUATORS
- SIMULATION
- ADD IMAGINARY ELEMENTS
MOTOR
INSTRUCTIONS
DIRECTIVES
TRANSMISSION

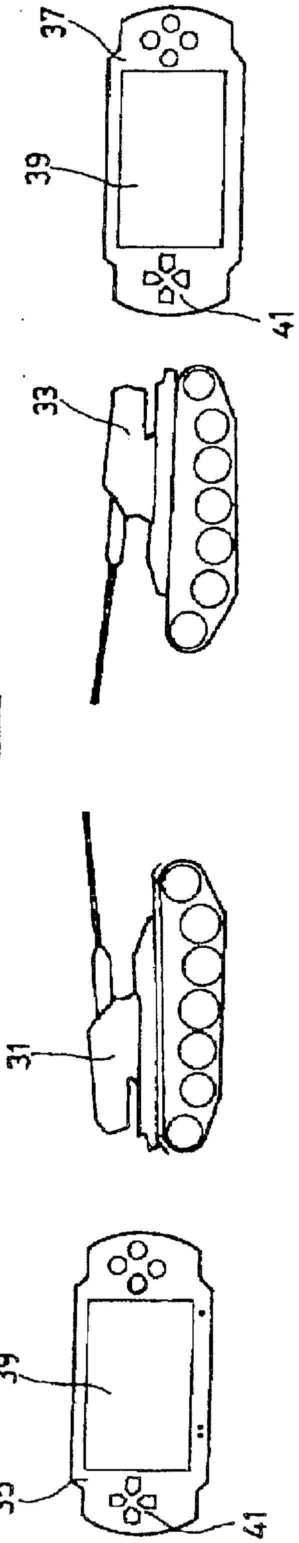


FIG_3

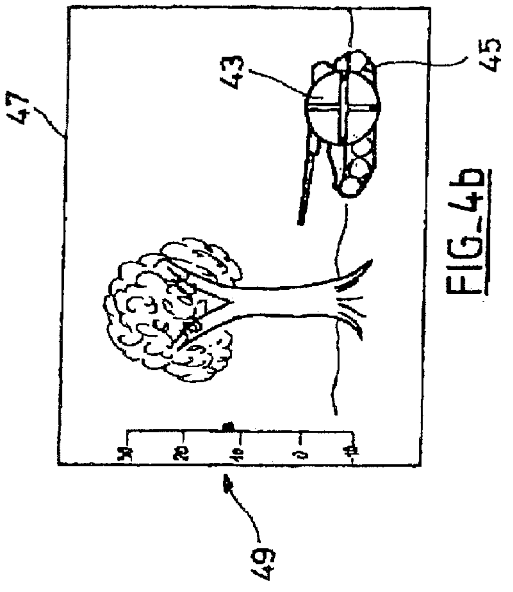


FIG_4b

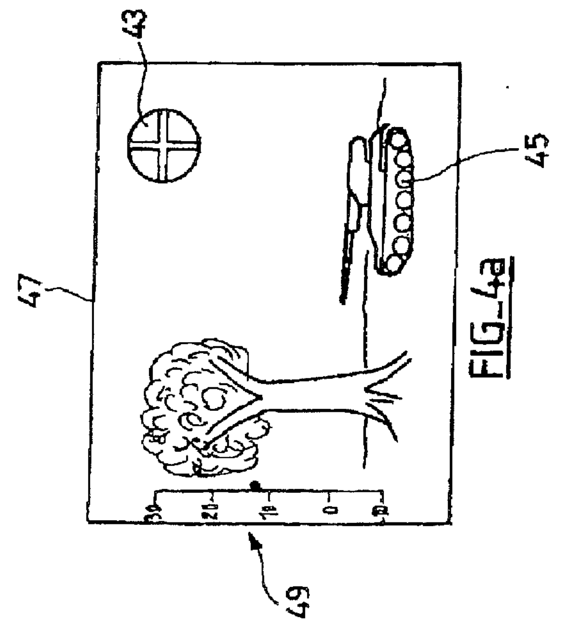


FIG_4a

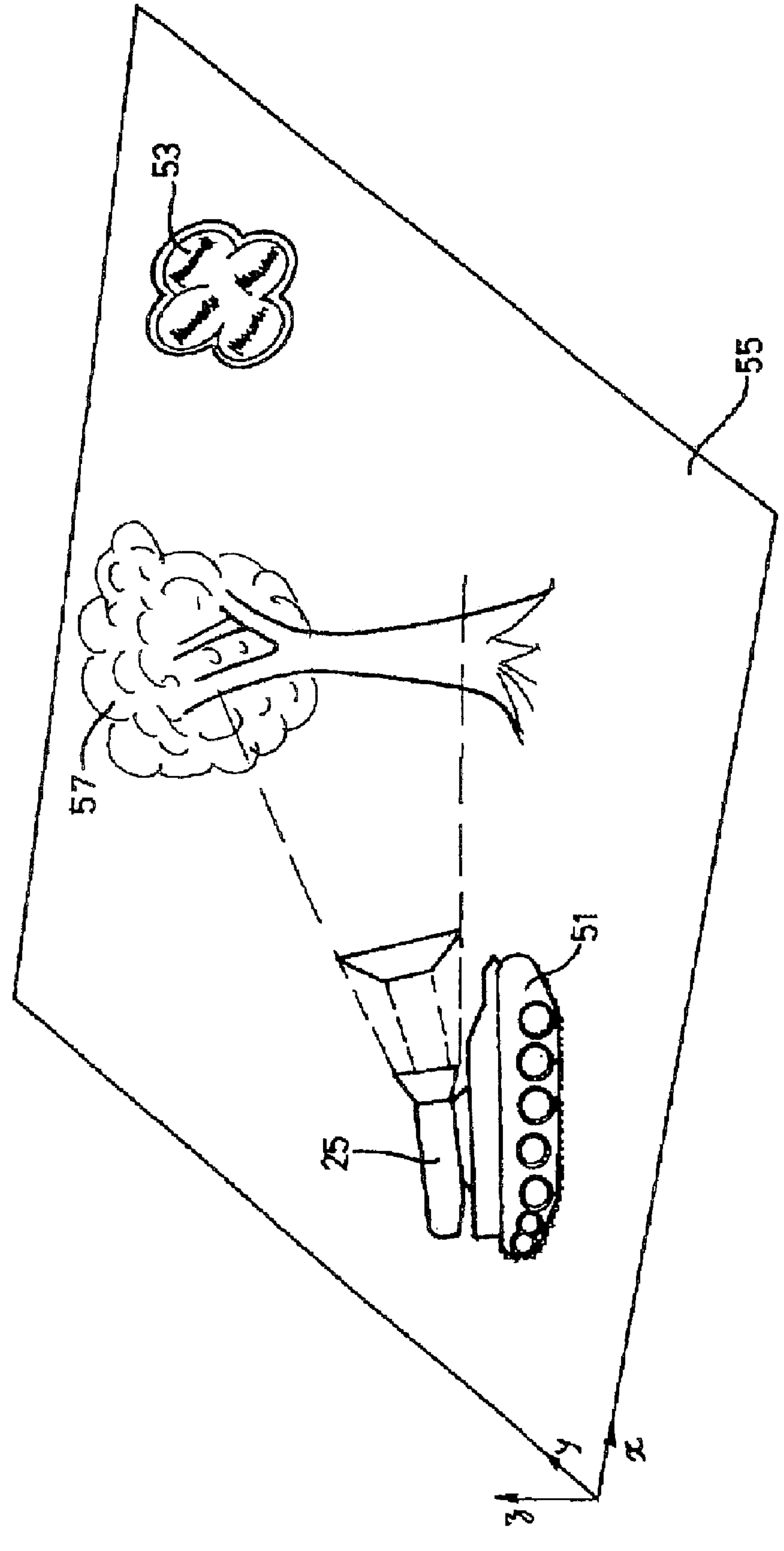


FIG_5

FIG_6a

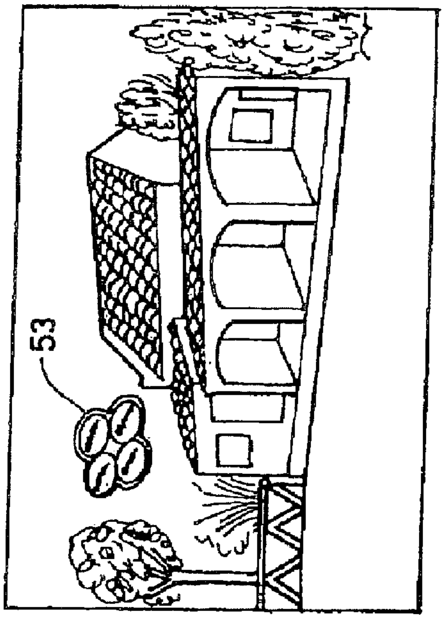


FIG_6b

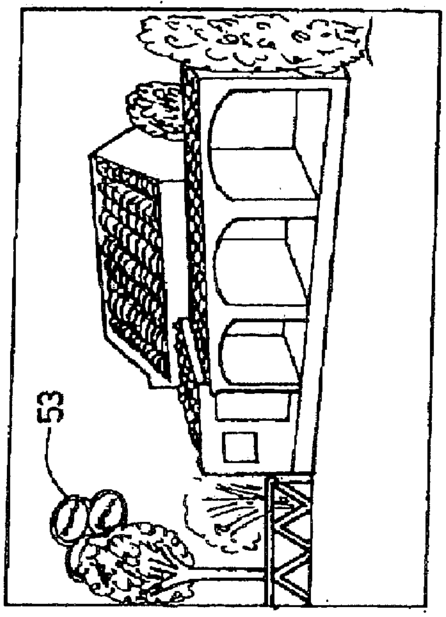
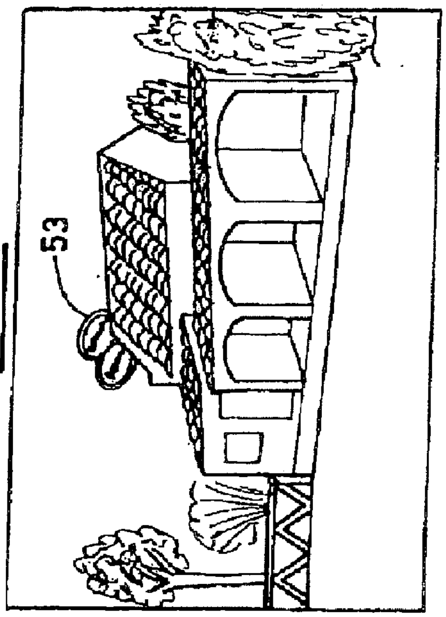


FIG_6c



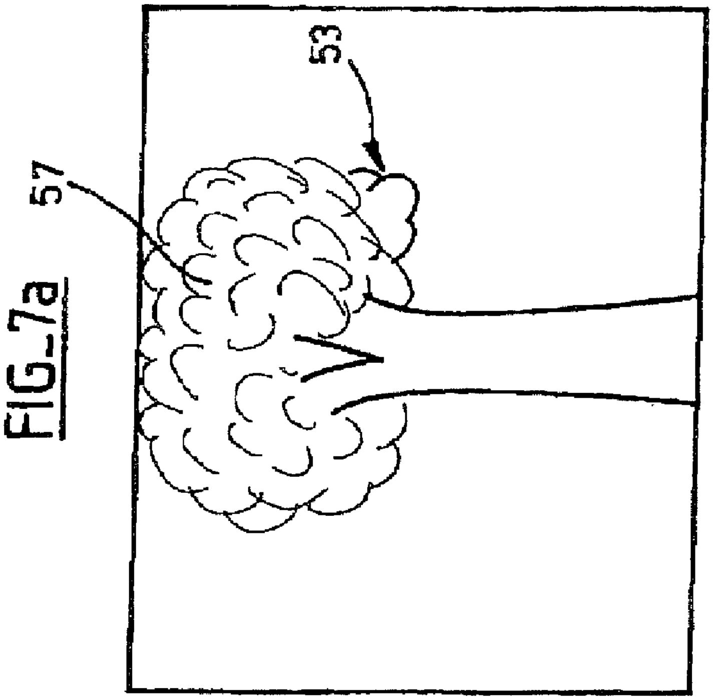


FIG_7a

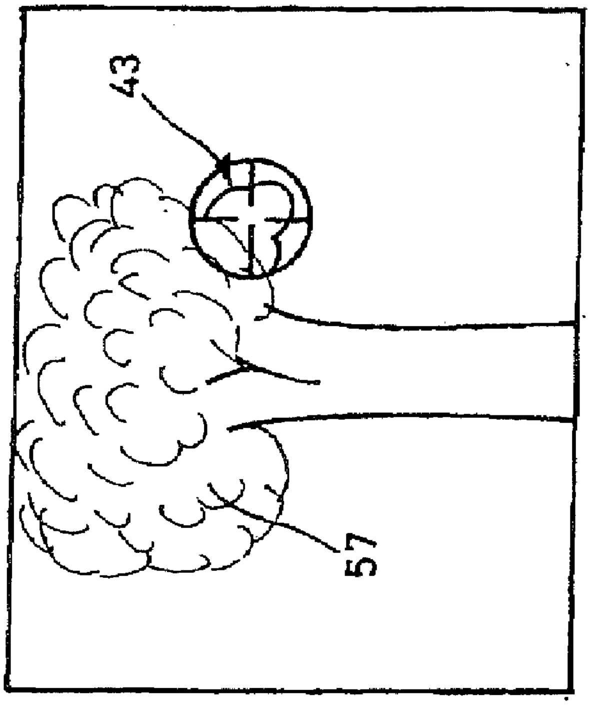


FIG_7b

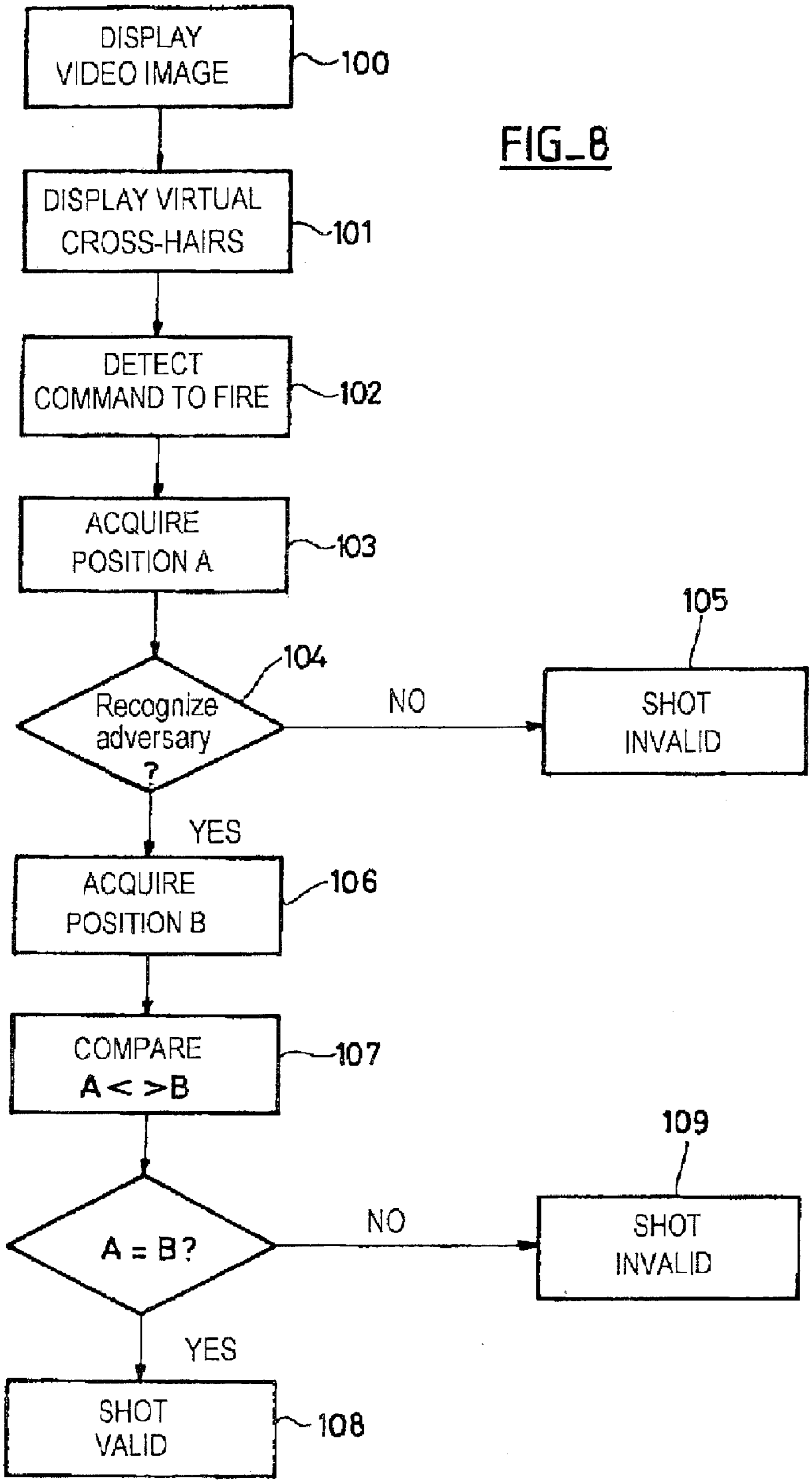

DISPLAY VIDEO IMAGE
100
DISPLAY VIRTUAL CROSS-HAIRS
101
DETECT COMMAND TO FIRE
102
ACQUIRE POSITION A
103
104
Recognize adversary ?
NO
105
SHOT INVALID
YES
ACQUIRE POSITION B
106
COMPARE A< >B
107
A = B?
NO
109
SHOT INVALID
YES
SHOT VALID
108

FIG_8

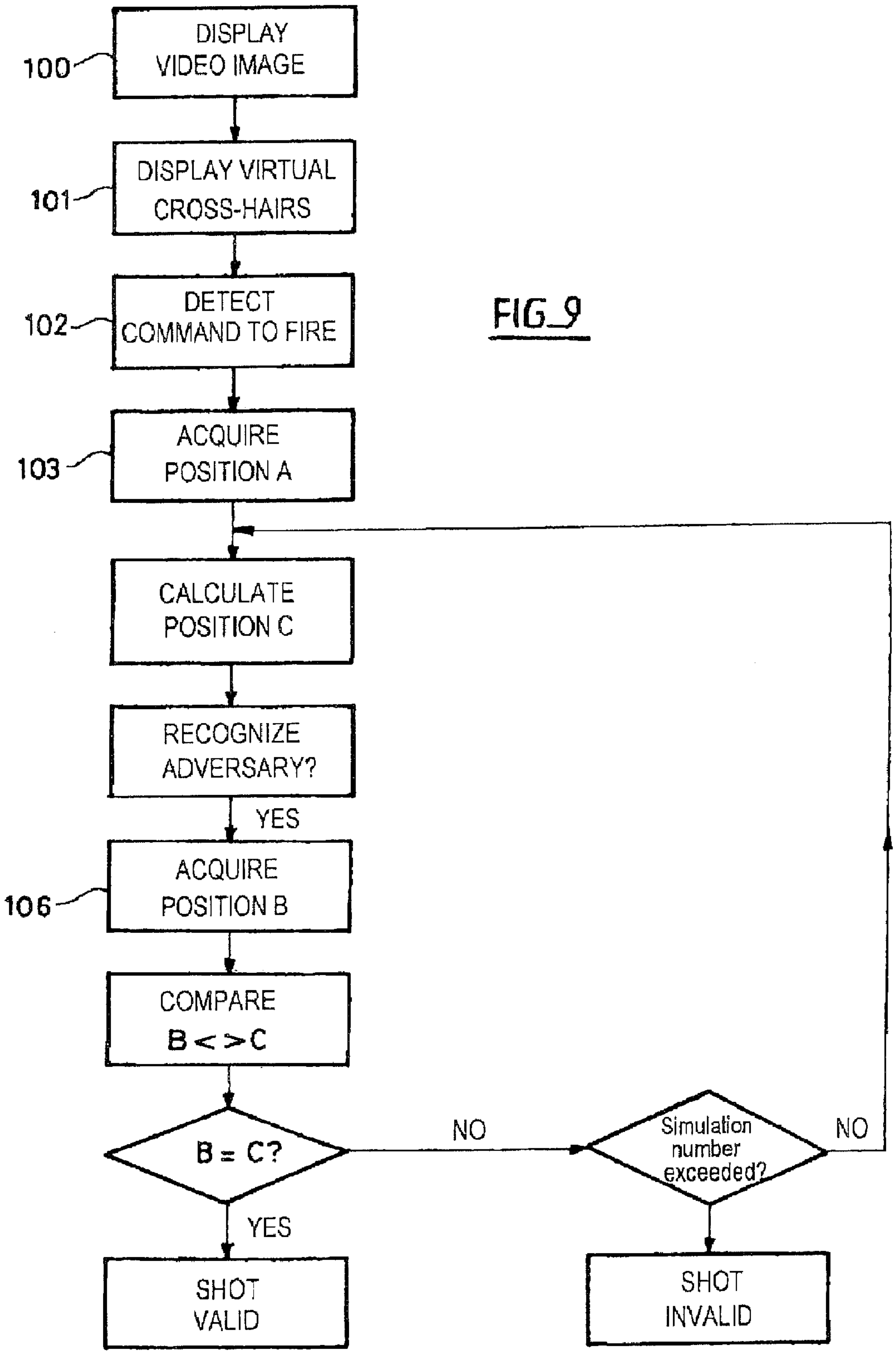

100
DISPLAY VIDEO IMAGE
101
DISPLAY VIRTUAL CROSS-HAIRS
102
DETECT COMMAND TO FIRE
103
ACQUIRE POSITION A
CALCULATE POSITION C
RECOGNIZE ADVERSARY?
YES
106
ACQUIRE POSITION B
COMPARE B<>C
B = C?
NO
YES
Simulation number exceeded?
NO
SHOT VALID
SHOT INVALID

FIG_9

A METHOD OF RECOGNIZING OBJECTS IN A SHOOTER GAME FOR REMOTE-CONTROLLED TOYS

[0001] The invention relates to a method of recognizing objects for a video game system.

[0002] Such a system is known from document WO 01/95568 A1. That document describes a video hunter game involving two remotely-guided vehicles with on-board video cameras. One of the two remotely-guided vehicles is the hunter and the other is the prey. The video images from the video camera of the hunter vehicle are transmitted to a control unit where they are displayed. The video images delivered by the hunter vehicle are scanned in order to detect the image of the adversary vehicle. If the adversary vehicle is detected in the video image, the adversary vehicle in the image is replaced by a character of the virtual game. Thus, the player using the control unit to drive the hunter vehicle sees on the video image, not an image of the adversary vehicle but a virtual image of a game character that the player is chasing with the vehicle.

[0003] The object recognition method disclosed in document WO 01/95988 A1 is nevertheless not applicable to a shooter game. The aim of the present invention is thus to propose an object recognition method for a video shooter game.

[0004] According to the invention, this aim is achieved by a method of recognizing objects for a video shooter game system, the system comprising:

[0005] a first remote-controlled vehicle including an on-board video camera;

[0006] a second remote-controlled vehicle; and

[0007] an electronic video display entity serving to remotely control the first vehicle;

[0008] the method comprising the following steps:

[0009] displaying the image delivered by the on-board camera on the video display of the electronic entity;

[0010] displaying virtual cross-hairs that are movable in the video image;

[0011] detecting a command to fire a virtual shot being input into the electronic entity;

[0012] acquiring the position A of the virtual cross-hairs in the video image;

[0013] recognizing the second remote-controlled vehicle in the video image; and

[0014] if recognition fails, invalidating the virtual shot; or

[0015] if recognition succeeds:

- **[0016]** a) acquiring the position B of the second vehicle in the video image;
- **[0017]** b) comparing the position A with the position B;
- **[0018]** c) in the event of A and B being identical, validating the virtual shot; or else
- **[0019]** d) in the event of A and B being different, invalidating the virtual shot.

[0020] The electronic entity of the system is preferably a portable console with a video screen.

[0021] By means of the method of the invention, it is possible to provide a video shooter game with a plurality of real remote-controlled toys. Each participant drives a remote-controlled vehicle and can make use of a real environment or setting in which the remote-controlled vehicles move.

[0022] In particular, the players may make use of real obstacles in order to attempt to protect their remote-controlled vehicles from shots fired by other players. In the invention, the shots fired by the vehicles are fictitious only, and they are simulated by the game system. The invention thus provides a novel combination between the aspect of a conventional video shooter game that is entirely virtual, and a conventional remote-controlled vehicle game that is entirely real.

[0023] By means of the invention, the players driving the remote-controlled vehicles can make use of elements of the real setting as elements of the game.

[0024] Preferably, the second vehicle is recognized by recognizing distinctive elements arranged on the second vehicle, namely light-emitting diodes (LEDs).

[0025] In a preferred application, the LEDs are arranged in the form of a specific shape. The LEDs may also flash at a determined frequency, have specific colors, and/or have colors that vary over time, so as to make it easier to recognize the second vehicle.

[0026] Recognizing the LEDs may also include measuring their total brightness in order to make it possible to estimate the fraction of the second vehicle that is visible as a function of the measured brightness value.

[0027] By measuring brightness in this way, it is possible to estimate whether a remote-controlled vehicle is partially hidden by an obstacle of the real setting. Such recognition of part of the vehicle can then influence the decision on whether or not a fictitious shot should be validated. In particular, the quantity of fictitious damage caused by the fictitious shot may be a function of the size of the detected fraction of the second vehicle.

[0028] Preferably, the method of the invention also makes use of speed and/or acceleration and/or position sensors on the target vehicle and/or on the shooter vehicle. The sensors enable the two vehicles to determine their own three-dimensional coordinates in real time. These coordinates may then be transmitted by radio means.

[0029] Preferably, the method of the invention further comprises a step of predicting the movement of the second vehicle in the video image, on the basis:

[0030] of a measurement of the movement of the first vehicle; and/or

[0031] of the earlier movement of the second vehicle in the video image.

[0032] The two above characteristics, i.e. using sensors on board the vehicles and predicting the movement of the second vehicle in the video image, may be coupled together. Software combines position information so as to obtain pertinent information that is unaffected by the drift to which the inertial units on board each of the two vehicles may be subject.

[0033] In the preferred version of the invention, the vehicles obtain fine estimates of their positions. The fact that an enemy vehicle disappears suddenly from the estimated position indicates that it is very probably hidden behind a real obstacle.

[0034] The feature whereby a virtual shooter game takes account of obstacles that are real is the preferred aspect of the invention.

[0035] Implementations of the invention are described below with reference to the accompanying drawings.

[0036] FIG. **1***a* shows the operation of a first system for simulating fictitious events in accordance with the invention;

[0037] FIG. **1***b* shows a second system enabling imaginary objects to be added, e.g. obstacles;

[0038] FIG. **1***c* shows imaginary elements added by the game on the display of the game console;

[0039] FIG. **1***d* shows a third system of the invention with an on-board autopilot;

[0040] FIG. **2***a* shows the operation of a second simulation system of the invention associated with an autopilot;

[0041] FIG. **2***b* shows the complete system: fictional events, imaginary objects, and an autopilot interacting in a very complete game;

[0042] FIG. **3** shows two remotely-controlled vehicles and their associated remote controls;

[0043] FIGS. **4***a* and **4***b* show the video display of a first shooter game of the invention;

[0044] FIG. **5** shows the real environment of a second shooter game of the invention;

[0045] FIGS. **6***a*, **6***b*, and **6***c* show various different video images taken from the second shooter game of the invention;

[0046] FIGS. **7***a* and **7***b* show the video image of the second shooter game while a player is firing a fictitious shot;

[0047] FIG. **8** is a flow chart of the object recognition method for validating or invalidating fictitious shots fired during shooter games; and

[0048] FIG. **9** is a flow chart for parabolic shot validation or invalidation when the target is hidden by a real obstacle.

[0049] FIG. **1***a* shows the concept of the invention which is that of a "open-loop" video game.

[0050] A simulator **1** supervises the operation of a radio-controlled toy **3**.

[0051] The simulator **1** of the video game modifies the instructions **5** for driving the toy **3** and coming from the player **7**. The toy **3** receives driving directives **9** from the simulator **1**. These directives **9** are generated by the simulator **1** while taking account of the driving instructions **5**. The toy **3** depends not only on the received directives **9**, but also on physical events that are external to the game.

[0052] Sensors **13** arranged on the toy **3** send information about the environment of the toy **3** to display means **15** of the video game. The information coming from the sensors **13** enables the video game system to estimate the changes of state of the toy **3** in its real environment. The display means **15** use the information from the sensors **13** to generate a display on a screen **21** of a control unit **23** handled by the player **7**.

[0053] The sensors **13** comprise in particular a video camera **25** on board the remote-controlled toy **3**. This video camera **25** delivers video images that are displayed by the display means **15** on the screen **21** used by the player **7**. The video camera **25** thus gives the player **7** a perspective as "perceived" by the remote-controlled vehicle **3**.

[0054] The toy may also be provided with other additional sensors. These may be sensors that are very simple such as accelerometers, or sensors that are extremely sophisticated, such as for example an inertial unit. By means of the sensors **13**, the video game constitutes a display. For example, with a gyro and/or accelerometers and display software, the video game can reconstitute an artificial horizon if the remote-controlled vehicle is a remotely-guided airplane.

[0055] There follows a description in detail of the role and the operation of the simulator **1**. The simulator **1** is situated between the player **7** and the radio-controlled toy **3**. It receives driving instructions **5** from the player **7**. These driving commands or actions **5** represent changes that the player **7** seeks to impart to the propulsion elements (such as the engine of the toy **3**) and/or guidance elements (such as the control surface of the toy **3**), e.g. for the purpose of directing the toy **3** in a certain direction.

[0056] These driving actions **5** are not transmitted directly, as is, to the remote-controlled toy. The toy **3** is decoupled from driving by the player **7** by the simulator **1**. It is the simulator **1** that directly controls the toy **3** by sending it directives **9**. These directives **9** are created by the simulator **1** while taking account of the instructions **5**.

[0057] The particularly advantageous aspect is that the simulator **1** generates the directives **9** not only on the basis of the instructions **5**, but also, and above all, on the basis of "handling characteristics" that are automatically generated by the simulator **1**. These handling characteristics are created as a function of the video game that has been selected by the player **7**. Depending on the selected video game, the simulator **1** simulates novel events that are absent from the physical world and that have an impact on the toy **3**. These fictional events are "translated" into handling characteristics that modify the directives **9** sent to the toy **3**, so that the behavior of the toy **3** is modified thereby.

[0058] For example, the simulator **1** may simulate a breakdown of the engine of the toy **3**. If the toy **3** is an airplane, the simulator **1** may make the airplane artificially "heavier" by creating handling characteristics that give the player **7** the impression that the airplane **3** is responding more slowly to commands **5** than usual. The simulator **1** may also create complete fictitious scenarios such as the airplane **3** flying through a storm. In this example, the simulator **1** generates handling characteristics that give rise to directives **9** that have the effects of causing the remote-controlled airplane **3** to shake as though it were being subjected to gusts of wind.

[0059] Thus, the simulator **1** acts in complex manner and in real time in the driving of the toy **3** so as to give the player **7** a game experience that is very rich. This is not merely a question of monitoring the player's driving in order to take action in the event of errors or danger. The simulator **1** is exerting an active and deliberate influence on the driving of the toy **3** so as to give it behavior that is more varied and more entertaining.

[0060] The main difference with a conventional video game taking place entirely on a computer without any real remote-controlled vehicle is that the change in state is not operated solely by the simulation, but is the result of the change of the open loop and this change of state is measured by sensors.

[0061] FIG. **1***b* is a more complete version of the system of the invention. To further enrich the game scenarios, the simulator adds imaginary elements to the instructions in addition to combined events. These imaginary elements may for example be obstacles, or, more interestingly, virtual objects that present behavior, for example virtual enemies. The imaginary elements may also be virtual elements of the radio-controlled toy itself, such as a weapons system having a sight and a virtual weapon that fires virtual projectiles.

[0062] Under such circumstances, two additional feedback loops are added to the system. In the first loop, information from the sensors of the toy is used by the simulator so as to estimate the position of the radio-controlled toy, e.g. to determine whether a shot from a virtual enemy has hit the toy.

[0063] The second feedback loop is defined between the simulator and the display software. The display software is informed about the movement of the virtual objects so as to be able to produce a composite display. For example, it adds virtual elements to the video image: obstacles; virtual enemies; or indeed elements of the shooter system such as the elements **43** in FIG. **4***a*.

[0064] FIG. **1***c* shows an example of augmented reality. Virtual elements are added, namely an enemy, projectiles, and a landing zone. The image is a composite of the real world plus imaginary objects.

[0065] FIG. **1***d* shows a loop when there is an on-board autopilot. A feedback loop takes place within the drone itself. The same sensors are used as those used for the display loop.

[0066] FIG. **2***a* shows an even more complete version of the system of the invention.

[0067] The system has an autopilot **27** added thereto. This serves to servo-control the operation of the airplane **3**. By virtue of the autopilot **27**, the airplane **3** can be made more stable and predictable in its behavior. Thus, the sources of interaction between the video game **29** and the toy **3** are more numerous. In a system with an autopilot **27**, the toy **3** may be said to be a drone since it has the ability to move autonomously in the real setting without needing to be driven by the player **7**.

[0068] The autopilot **27** has a command envelope. If the vehicle **3** is a tank, then the envelope may for example define its maximum speed, its maximum acceleration, its turning speed, etc.

[0069] If the vehicle **3** is a quadricopter, the command envelope of the autopilot **27** may define its maximum rate of climb, its maximum angular speed, and a description of the main transition between hovering flight and forward flight.

[0070] The command envelope of the autopilot **27** is thus a data set defining constraints on the movement that the vehicle **3** is capable of performing. The command envelope of the autopilot **27** thus limits the capabilities of the vehicle **3**.

[0071] By manipulating the envelope of the autopilot, the video game can simulate several physical magnitudes. For example, by changing the data of the envelope, it is possible to simulate a vehicle that is heavier, by simulating a larger amount of inertia by limiting acceleration of the vehicle. Such an envelope has the effect of delivering less power to the engine of the vehicle **3**. This is under the control of the simulator **1**.

[0072] Thus, the simulator **1** can create a wide variety of fictitious scenarios. For example, at the beginning of the game sequence, the simulator **1** may simulate a vehicle **3** that is heavier since it has a full load of fuel. As the game progresses, the simulator **1** then simulates a lightening in the weight of the vehicle **3**. Or else, the simulator **1** may simulate a mission that consists in unloading fictitious equipment that is being transported by the vehicle **3**. Once more, the simulator **1** generates handling characteristics that cause the directives **9** to give the player **7** the impression that the weight of the vehicle **3** changes during the game as the fictitious equipment is unloaded. The directive between the simulator and the driver in this example serves to modify the virtual weight of the drone. In this example, at the beginning of the game, the handling characteristics may impose a maximum speed and a maximum acceleration on the toy **3** that are relatively low. The player **7** will therefore not be able to cause the toy **3** to go at high speed, even when the driving commands **5** request that. As the game progresses, the handling characteristics change, thereby progressively raising the speed and acceleration limits that the toy **3** is capable of achieving. At a late stage in the game, the player can thus reach higher speeds with the toy **3**. In this way, the player genuinely has the impression of driving a toy that becomes lighter as time advances, even if this is achieved solely by simulation.

[0073] In theory, the controls of the autopilot could be implemented directly by the simulator in a “remote autopilot” mode. It is much more efficient in terms of system design for the autopilot to be on board the toy **3**. Because of the control of the autopilot **27**, the amount of information and the critical nature thereof in real-time terms are reduced. The simulator sends higher-level directives to the autopilot, such as for example fictitious damage, a change in the weight of the vehicle, or instructions to carry out an emergency landing as a result of virtual events resulting from the video game.

[0074] The simulator **1** may modify the instructions **5** from the player **7** to a certain extent, in particular by superposing thereon novel events having the same class as the instructions. This superposition may be addition, subtraction, division, multiplication, setting limits, etc. The superposition may be carried out using any combination of arithmetic and/or logical operations.

[0075] Such superposition, e.g. by adding or subtracting signals generated by the simulator **1** to or from the command signals **5** given by the player **7** is very useful for simulating events that seek to bias the driving of the vehicle **3**.

[0076] For example, if the vehicle **3** is a tank, superposition makes it possible to simulate the tank receiving a hit from an adversary. The hit is simulated as being non-fatal but as damaging the tank. The simulator **1** thus superposes signals on the actions **5** of the player that simulate a tendency for the tank to revolve slowly. The player **7** then needs to adapt the way the tank is driven in order to compensate for this superposed bias. Under such circumstances, the player **7** needs to give opposite-direction commands to the tank so as to compensate for the bias created by the simulator **1**.

[0077] With a toy **3** in the form of a quadricopter, driving can be more complex by simulating wind by adding a drift component. It is possible to simulate in even more complex manner, e.g. simulating gusts of wind. The player then needs to compensate these events that follow on from one another.

[0078] In another mode of interaction, the video game takes complete control of the toy **3**, either temporarily, or permanently. For example, if vehicle **3** is a tank, it is possible to simulate that it has been hit. The simulator **1** then takes exclusive control and causes the vehicle to spin round and shake. Thereafter control is returned to the player.

[0079] Another example: the vehicle receives a fatal hit, which is simulated in such a manner that the vehicle performs a final lurch before stopping completely. The game is then over.

[0080] FIG. **2***b* shows a complete simulation system in which the feedback loops combine at three different levels to create a complete augmented reality game.

[0081] The first device is the device that transforms the “instructions” from the player into directives, e.g. for the purpose of adding imaginary events.

[0082] The first feedback loop is the loop in which virtual objects are added to the system by the simulator. The display software then combines the measurements from the sensors and the information from the simulator to produce a composite display comprising the real image plus the virtual objects.

[0083] The second feedback loop is that in which the measurements from the sensors of the radio-controlled toy are used by the simulator. They enable it to simulate virtual objects, for example to verify whether the toy is colliding with a virtual obstacle, or to inform a virtual enemy seeking to chase the radio-controlled toy.

[0084] The third loop is that of the autopilot, which serves to match the behavior envelope of the vehicle to high-level directives from the simulator, for example to make the vehicle virtually heavier so that it appears different.

[0085] FIG. **3** shows an implementation of the system of FIG. **1** or **2** in the form of a first remote-controlled vehicle **31** and a second remote-controlled vehicle **33** together with their associated remote controls **35** and **37**. The remote-controlled vehicles **31** and **33** are toys in the form of tanks. The remote controls **35** and **37** are in the form of portable consoles with video displays **39** and control buttons **41**. Each of the tanks **31** and **33** has an on-board video camera. The video image delivered by the on-board camera is sent to the corresponding portable console **35**, **37** and displayed on its screen **39**. A user controls one of the tanks **31**, **33** with the help of the controls **41** on the corresponding game console **35**, **37**. Communication between the game consoles and the tanks, and communication between the tanks themselves preferably takes place using a Bluetooth or WiFi (registered trademarks) protocol.

[0086] The tanks **31** and **33** and the consoles **35** and **37** can be used by two players to engage in a shooter game, with representations thereof appearing in FIGS. **4***a* and **4***b*.

[0087] The players play against each other, each via a game console **35**, **37** and a video toy **31**, **33**. The game consists mainly in causing the tanks **31**, **33** to move, in ordering movements of the tanks' turrets and guns, in moving virtual cross-hairs **43** over the screen **39** of the game console **35**, **37**, in superposing the virtual cross-hairs **43** on the image **45** of the adversary, and in firing a virtual shot at the adversary.

[0088] These various steps are shown in FIGS. **4***a* and **4***b*. FIG. **4***a* is an example of a video image delivered by one of the two video cameras on one of the tanks. The video image **47** is transmitted from the tank and shown to the driver of the tank on the screen **39** of the game console. Various virtual elements are inlaid on the video image, such as the virtual cross-hairs **43** of the virtual sight and a virtual scale **49** indicating the elevation angle of the tank's gun.

[0089] With the controls **41** on the console, the player moves the virtual cross-hairs **43** over the video image **47** until they are superposed on the image **45** of the adversary, as shown in FIG. **4***b*. Once the player has thus aimed at the adversary, it suffices to fire a shot. The video game then simulates the path of a fictitious shell, in particular the speed, parabolic trajectory, an impact angle thereof, which parameters are all estimated by the video game. It is also possible to simulate damage to the adversary tank. Under such circumstances, information is forwarded to the adversary tank (or to the adversary game console) representative of the extent of the damage. This information is used for simulating damage of the adversary tank. For this purpose, the ability of the tank to move is modified artificially by the simulator **1** (cf. FIGS. **1** and **2**). If the hit is not considered as being fatal, the video game system artificially limits the speed of the tank or modifies the way it responds to driving commands in such a manner that the tank's turret can turn in only one direction. If the hit is considered as being fatal, then the system stops the tank from moving and the game is over.

[0090] FIGS. **5** to **7** show a second variant of a shooter game of the invention.

[0091] In this variant, the game takes place not between two remotely-controlled tanks, but between a toy in the form of an anti-aircraft vehicle **51** having an on-board video camera **25**, and a toy in the form of a quadricopter **53**. FIG. **5** gives an overall view of a real environment **55**, e.g. including a real tree **57** as an obstacle. The vehicle **51** and the quadricopter **53** move in the real setting **55** as a function of driving directives. One player drives the anti-aircraft vehicle **51** and another player drives the quadricopter **53**. In the context of the game, the driver of the anti-aircraft vehicle **51** seeks to shoot down, virtually, the quadricopter **53** controlled by the other player.

[0092] FIGS. **6***a* to **6***c* show the viewpoint delivered to the driver of the vehicle **51** by the video camera **25** on board that vehicle. FIGS. **6***a* to **6***c* show a real setting that is different from that of FIG. **5**. Here the real environment has two houses and several trees. In the first video image in FIG. **6***a*, the quadricopter **53** can clearly be seen. In the second video image in FIG. **6***b*, subsequent to the image of FIG. **6***a*, the quadricopter **53** has moved and is masked in part by the vegetation. In the third video image of FIG. **6***c*, subsequent to the image of FIG. **6***b*, the quadricopter has gone back the other way and is in the process of hiding behind one of the houses.

[0093] The game system of the invention is capable of estimating whether the quadricopter **53** is visible, visible in part, or hidden in the video images as shown in FIGS. **6***a* to **6***c*. This recognition procedure is shown in detail by the flow chart of FIG. **8**.

[0094] By such recognition of objects in the image, the game system is capable of validating a fictitious shot at the quadricopter **53** if the quadricopter is recognized as being visible in the video image (and is properly aimed at). If the quadricopter **53** is recognized as being hidden, the shot in its direction is invalidated.

[0095] In this way, the players of the video game can make use of elements in the real setting as elements of the game.

[0096] FIGS. **7***a* and **7***b* show two video images as delivered by the camera **25** on board the anti-aircraft vehicle **51** of FIG. **5**. In particular, FIGS. **7***a* and **7***b* show the procedure for recognizing the quadricopter **53** in the video image and they show virtual cross-hairs **43** being positioned to fire a fictitious shot. The driver of the vehicle **51** moves virtual cross-hairs **43** representing its weapon over the video image in order to aim at the adversary, i.e. the quadricopter **53**. The player moves the virtual cross-hairs **43** until they are superposed on the image of the quadricopter **53**, as shown in FIG. **7***b*. The player then fires a shot. However, since the quadricopter **53** is hidden by the tree **57**, even if the cross-hairs are properly placed by the shooter, the game will invalidate the shot.

[0097] FIG. **8** shows the detail of the object recognition and shot validation procedure used by the video game system. In step **100**, the procedure begins by displaying the video image delivered by the video camera **25** on the screen of the game console of the shooter. In step **101**, the video game system inlays virtual cross-hairs **43** in the displayed video image. The shooter can now move the virtual cross-hairs **43** to take aim and subsequently to fire a shot. Firing of the shot is detected in step **102**. As soon as a shot is detected, the system acquires the instantaneous position A of the virtual cross-hairs **43** on the video image, in a step **103**.

[0098] Once the position A of the virtual cross-hairs during firing is known, image recognition software scans the video image in order to recognize the presence of the adversary vehicle **53** in the video image, in a step **104**.

[0099] The adversary may be recognized in the image in various ways. For example, the recognition software may merely scan the video image while attempting to find the known shape of the adversary vehicle **53**. Nevertheless, in order to facilitate recognition, it is preferable for the adversary vehicle **53** to have recognition elements placed on its surface. By way of example, these may be reflecting elements.

[0100] In the preferred implementation of the invention, the adversary vehicle **53** has numerous flashing LEDs all around it. These LEDs are in a known configuration, they flash at a known frequency, and they are of known colors. The recognition software can thus detect the adversary vehicle **53** more easily in the video image.

[0101] It is also possible to envisage using multi-color LEDs that pass from green to red to orange so as to facilitate recognition. The LEDs may also emit in the infrared. In this way, there are no longer any elements that are visible to the human eye in the game system.

[0102] The image recognition software may be located either in the remote-controlled vehicle **51**, or in the control unit, i.e. the game console.

[0103] In order to further improve recognition, the recognition software may include an algorithm for tracking the recognition elements of the adversary vehicle **53**. Image after image, this algorithm measures the movements of the recognition elements. For this purpose, it is possible to rely on the movement of the shooter vehicle **51**, thus enabling it to predict where the adversary vehicle **53** ought to be found in the image, by adding the previously-observed movement of the adversary vehicle **53** so as to enable the software to predict its position.

[0104] In order to further improve recognition of position between radio-controlled toys, the sensors **13** on each radio-controlled toy are used. The information is sent not only to the game console used for driving the vehicle, but also to all the other game consoles. In this way, the simulator knows the estimated position of the enemy vehicle. The video recognition means no longer serves to find the enemy vehicle in the image, but to verify whether or not there is a real object between two vehicles. This principle is of great interest in the context of a video game. It makes it easy to take account of real obstacles. In this way, as shown in FIG. 7*b*, players may hide behind a tree.

[0105] Returning to FIG. **8**, after the recognition step **104** has been performed as described above, the procedure continues as a function of the results of the recognition.

[0106] If recognition is negative, the shot is invalidated in a step **105**. In contrast, if the recognition software has recognized the adversary vehicle **53** in the video image, the next step **106** consists in acquiring the instantaneous position B of the adversary vehicle **53** in the video image. Thereafter, the position B of the adversary vehicle **53** is compared in a step **107** with the position A of the virtual cross-hairs **43** when the shot was fired. If the positions are identical, the shot is validated in a step **108**, i.e. the fictitious shot has reached its target. If the positions are different, the shot is invalidated, in a step **109**.

[0107] To make the recognition of objects even more effective, the adversary vehicle **53** may also transmit its instantaneous position to the shooter vehicle **51**.

[0108] As described above, the vehicles have sensors, and in particular inertial units. Throughout the duration of the game the vehicles transmit their positions. When a vehicle comes within the field of view of the video camera, its position as transmitted by radio is known. This is used as an initial position by the detection algorithms. This enables them to converge more easily and faster.

[0109] The two methods of sensing and predicting video movements can be coupled together. Software makes use of the position information to obtain pertinent information that is unaffected by drift in the inertial unit.

[0110] Thus, in the preferred version of the invention, the vehicles produce fine estimates of their positions. The fact that the detection algorithm indicates that a vehicle has suddenly disappeared from the estimated position indicates that there is very probably a real obstacle between the two vehicles.

[0111] To summarize, by using software for analyzing images and position sensors, the game system knows the position of the adversary vehicle **53** in the image. It can determine whether the adversary vehicle **53** is masked or out of sight. It waits until it detects the LEDs at a precise position in the image, and otherwise it detects nothing and concludes that the adversary vehicle **53** is masked.

[0112] It is also possible to imagine a game stage in which the shooter fires indirectly at a hidden adversary vehicle. The fictitious shot is nevertheless validated because of the position known from the inertial unit and assuming that the game involves firing a parabolic shot around the obstacle behind which the adversary vehicle is hiding.

[0113] It is also possible to imagine a game stage that is even more realistic in which firing is indirect and the shooter aims the cross-hairs in front of the adversary vehicle. The simulation software then defines the position of the virtual projectile step by step. The software for recognition and movement prediction tracks the movement of the adversary to the point of virtual impact. The simulation software can simulate complex firing parameters: for example, the initial speed of a shell may be very fast on firing and may decrease rapidly. The software can also simulate complex behaviors of munitions such as the projectile exploding close to the target, or guided missiles, or homing missiles. This type of projectile simulation makes the game even more interesting, an adversary may see an adversary projectile being launched and then maneuver very quickly in order to avoid it. This also makes it possible to simulate situations that are extremely complex such as a shooter game between fighter planes in which the trajectory of the projectiles depend on the trajectories of the firing airplane, and the position of a point of impact depends on the position of the adversary airplane several seconds after the shot was fired.

[0114] FIG. **9** is a flow chart for indirect shooting. The position C is the virtual position calculated by the projectile simulator. The purpose of the algorithm is to verify whether the position of the virtual projectile is identical to the position of the adversary vehicle at position B. Position A is the position of the cross-hairs and serves only to define the initial conditions under which the projectile is fired. The position C of the projectile varies over time, and the duration of the simulation is limited by the software.

[0115] Finally, the procedure for recognizing objects may also proceed with recognition that is partial, i.e. it may estimate whether or not the object is partially hidden. Such partial recognition may be performed by measuring variation in the brightness of the LEDs from one image to another. In order for this partial recognition to be effective, it is preferable to place numerous LEDs on the surfaces of the remote-controlled vehicle in question. By way of example, the system may assume that when brightness is halved, then half of the LEDs are masked, which means that the adversary vehicle is half hidden.

[0116] Naturally, even if the video games described herein involve only two remote-controlled vehicles, the invention is not limited to a game having only one or two of them. The games described may be played with more than two players and more than two remote-controlled toys.

1. A shot validation method for a video game system, the system comprising:

a first remote-controlled vehicle (**51**) including an onboard video camera (**25**);

a second remote-controlled vehicle (**53**); and

an electronic video display entity (**35**, **37**) serving to remotely control the first vehicle (**51**);

the method being characterized in that it comprises the following steps:

displaying the image delivered by the on-board camera on the video display of the electronic entity (**100**);

displaying virtual cross-hairs (**43**) that are movable in the video image (**101**);

detecting a command to fire a virtual shot being input into the electronic entity (**102**);

acquiring the position A of the virtual cross-hairs (**43**) in the video image (**103**);

recognizing the second remote-controlled vehicle (**53**) in the video image (**104**); and

if recognition fails, invalidating the virtual shot (**105**); or

if recognition succeeds:

a) acquiring the position B of the second vehicle (**53**) in the video image (**106**);

b) comparing the position A with the position B (**107**);

c) in the event of A and B being identical, validating the virtual shot (**108**); or else

d) in the event of A and B being different, invalidating the virtual shot (**109**).

2. A method according to claim **1**, the second vehicle (**53**) being recognized by recognizing distinctive elements arranged on the second vehicle, namely LEDs.

3. A method according to claim **2**, the LEDs:

being arranged to form a specific pattern;

flashing at a determined frequency;

having specific colors; and/or

having colors that vary over time;

in order to facilitate recognizing the second vehicle (**53**).

4. A method according to claim **2**, recognition of the LEDs including measuring their total brightness in order to enable a fraction of the second vehicle (**53**) that is visible to be recognized as a function of the measured brightness value.

5. A method according to claim **1**, further including a step of predicting the movement of the second vehicle (**53**) in the video image, on the basis:

of a measurement of the movement of the first vehicle (**51**); and/or

of the earlier movement of the second vehicle (**53**) in the video image.

6. A method according to claim **1**, the vehicles (**51**, **53**) having sensors, in particular position sensors, to track their movements, and telecommunications means for enabling said movements to be transmitted, position B of the second vehicle (**53**) in the video image (**106**) being recognized on the basis of information from the sensors as transmitted by the vehicles (**51**, **53**).

7. A method according to claim **1**, the positions measured by the sensors being used as initial conditions for recognition algorithms in the video image (**106**).

8. A method according to claim **2**, including an analysis step consisting in considering that the second vehicle (**53**) passes behind a real obstacle when the recognition of distinctive elements in the video image (**106**) indicates a sudden disappearance of the second vehicle (**53**).

9. A method according to claim **1**, including an alternative step of validating a shot, in the event of indirect shooting, the shot being validated by simulating the trajectory of a virtual projectile and its impact at the position B of the target vehicle (**53**) in the video image (**106**).

10. A method according to claim **3**, recognition of the LEDs including measuring their total brightness in order to enable a fraction of the second vehicle (**53**) that is visible to be recognized as a function of the measured brightness value.

11. A method according to claim **3**, including an analysis step consisting in considering that the second vehicle (**53**) passes behind a real obstacle when the recognition of distinctive elements in the video image (**106**) indicates a sudden disappearance of the second vehicle (**53**).

12. A method according to claim **4**, including an analysis step consisting in considering that the second vehicle (**53**) passes behind a real obstacle when the recognition of distinctive elements in the video image (**106**) indicates a sudden disappearance of the second vehicle (**53**).

* * * * *