



US 20200263957A1

(19) **United States**

(12) **Patent Application Publication**  
**Northrup**

(10) **Pub. No.: US 2020/0263957 A1**

(43) **Pub. Date: Aug. 20, 2020**

(54) **SYSTEM AND METHOD FOR MARKSMANSHIP TRAINING**

now Pat. No. 10,030,937, which is a continuation-in-part of application No. 14/149,418, filed on Jan. 7, 2014, now Pat. No. 9,261,332, which is a continuation-in-part of application No. 13/890,997, filed on May 9, 2013, now Pat. No. 9,267,762.

(71) Applicant: **Shooting Simulator, LLC**, Dallas, TX (US)

(72) Inventor: **James L. Northrup**, Dallas, TX (US)

(73) Assignee: **Shooting Simulator, LLC**, Dallas, TX (US)

(21) Appl. No.: **16/814,860**

(22) Filed: **Mar. 10, 2020**

**Publication Classification**

(51) **Int. Cl.**  
**F41G 3/26** (2006.01)

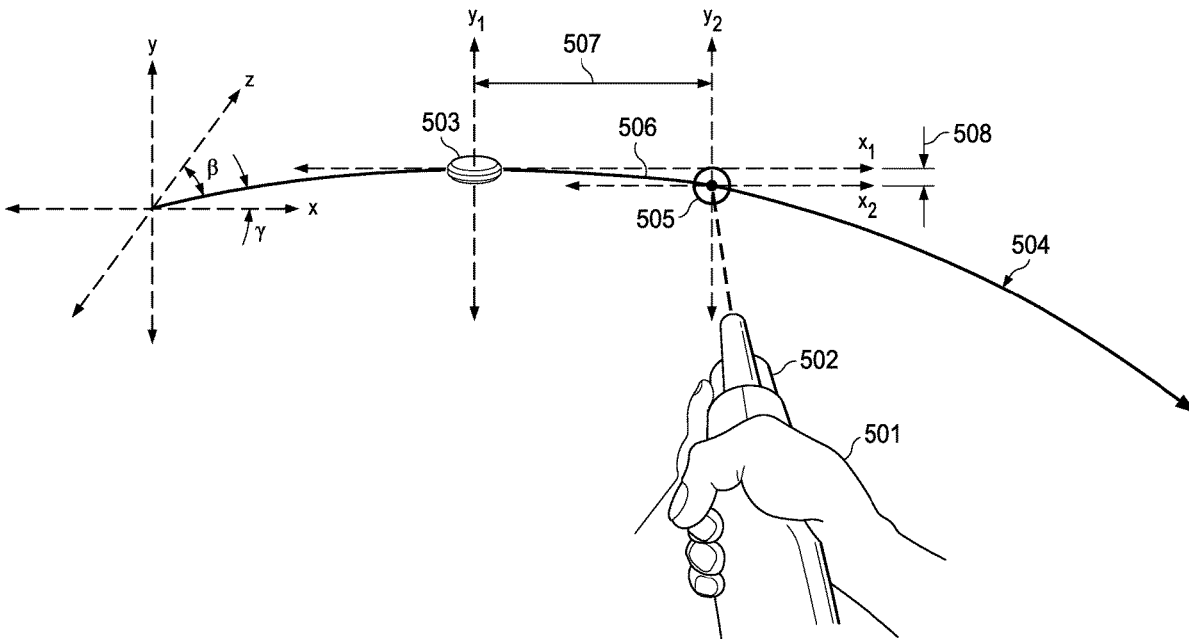
(52) **U.S. Cl.**  
CPC ..... **F41G 3/2633** (2013.01)

(57) **ABSTRACT**

A system for and method of use of an augmented reality display is provided. The preferred system is implemented by one or more tactical units calculating a target path from a weapon position and a range. A lead is calculated. A phantom target is displayed at the lead. A virtual laser and virtual tracer are provided to assist in target tracking. A spotter unit is also provided to supplement target path and range data. A neural network is provided to learn from tracking and successful lead data and to predict lead in the tactical theatre.

**Related U.S. Application Data**

(63) Continuation-in-part of application No. 16/397,983, filed on Apr. 29, 2019, now Pat. No. 10,584,940, which is a continuation-in-part of application No. 15/589,603, filed on May 8, 2017, now Pat. No. 10,274,287, which is a continuation-in-part of application No. 14/969,302, filed on Dec. 15, 2015, now Pat. No. 10,234,240, which is a continuation-in-part of application No. 14/686,398, filed on Apr. 14, 2015,



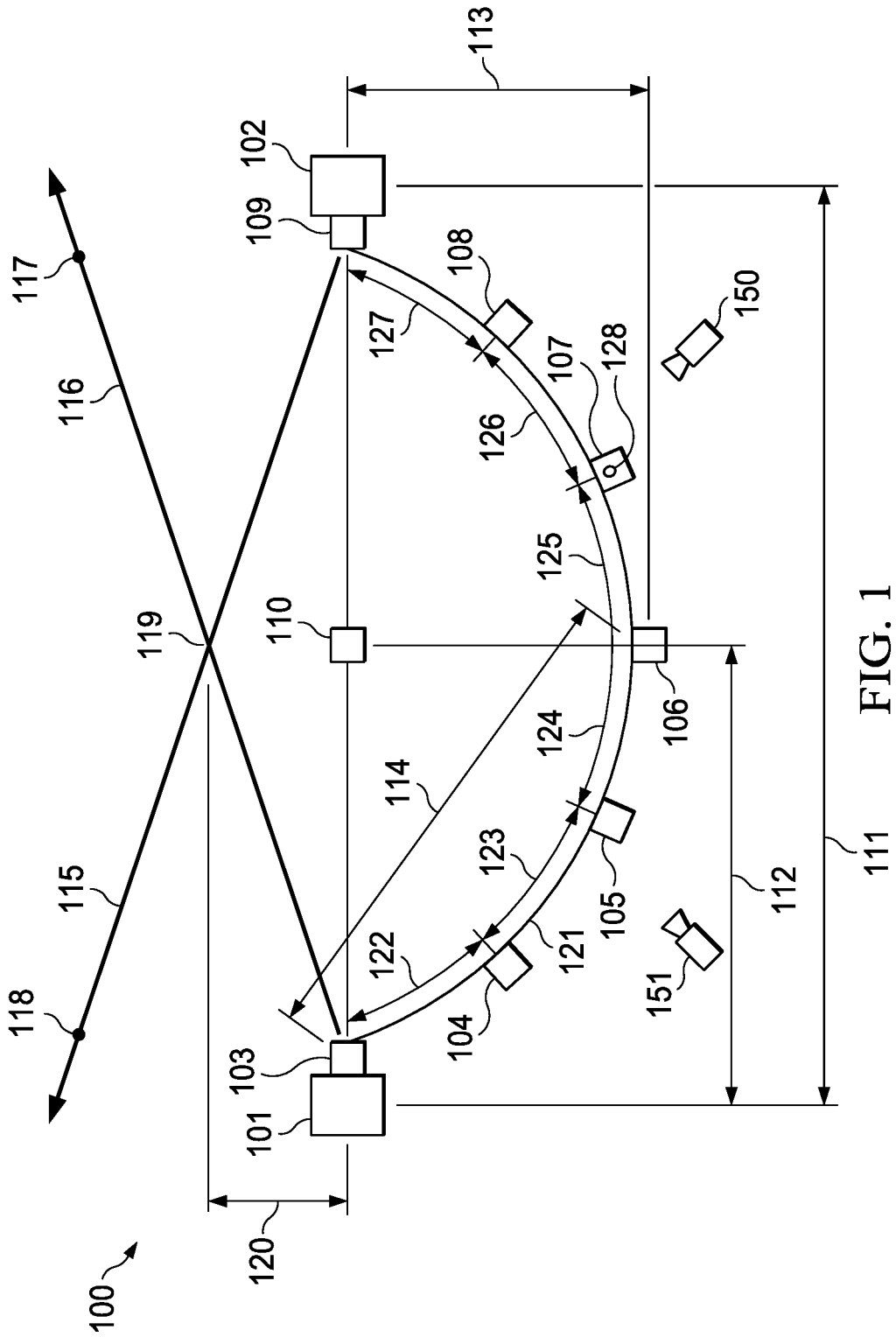
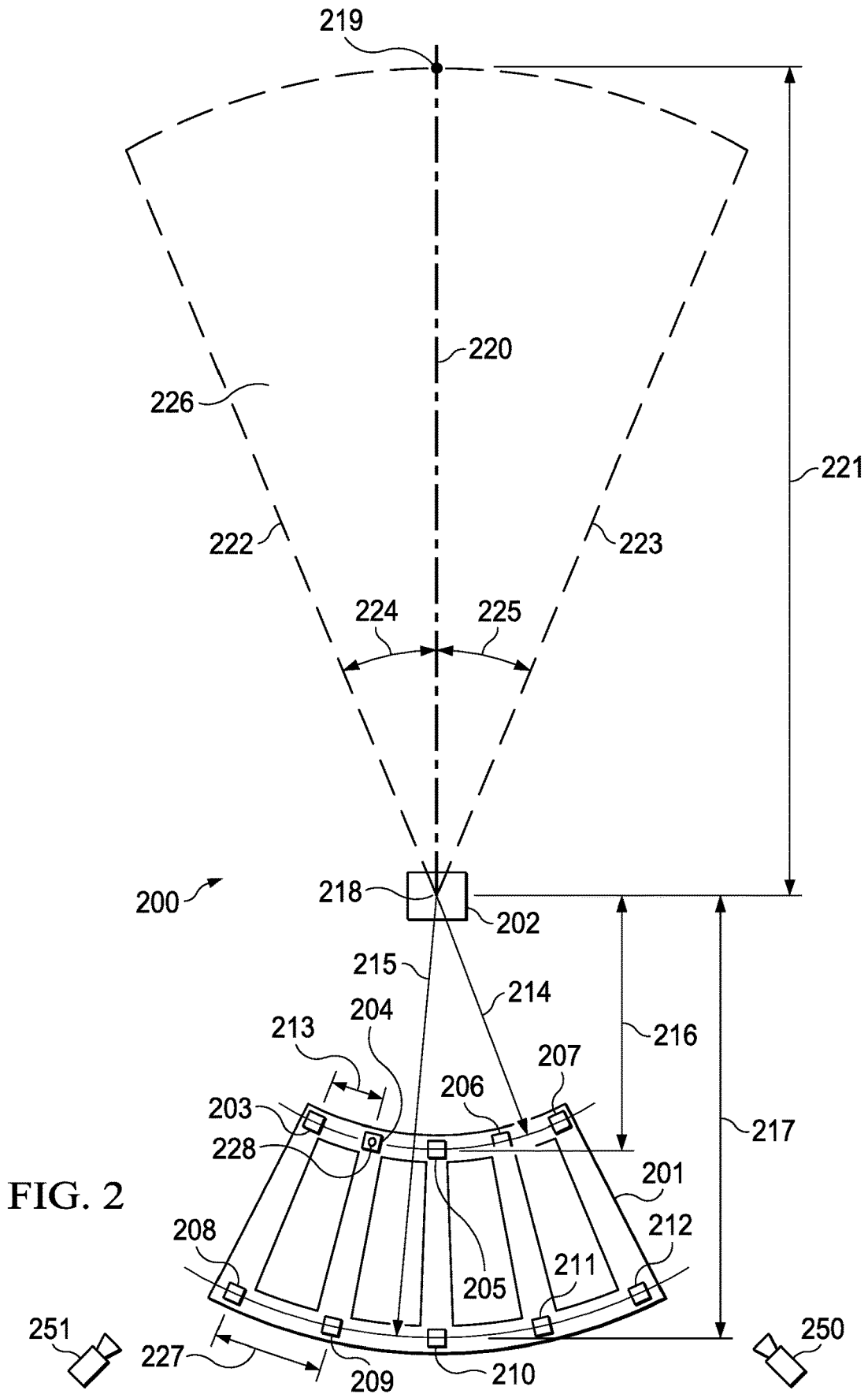
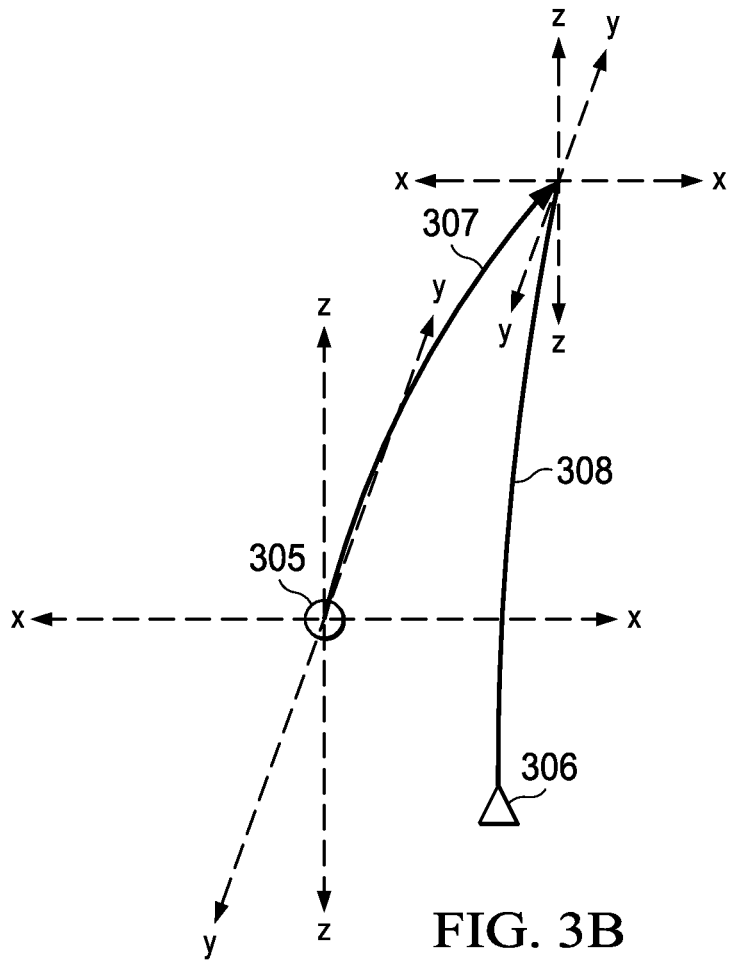
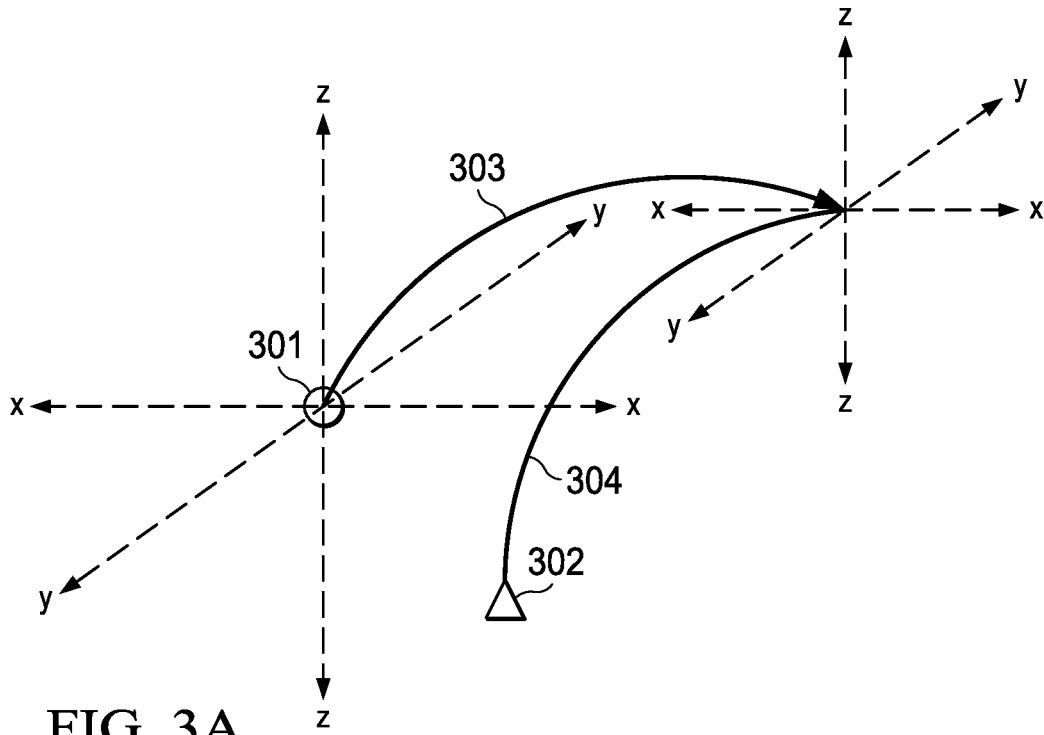


FIG. 1





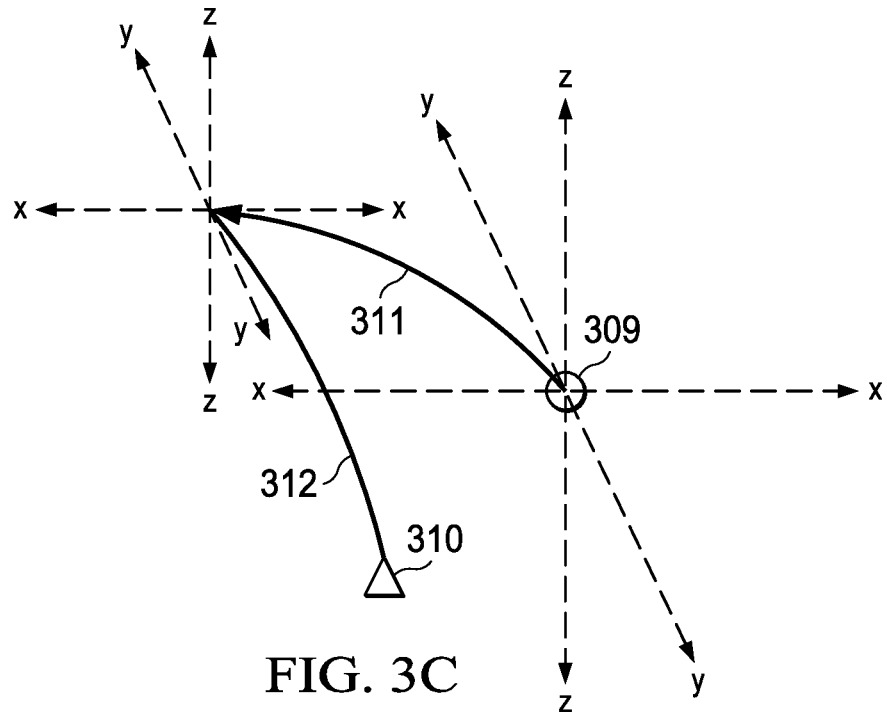


FIG. 3C

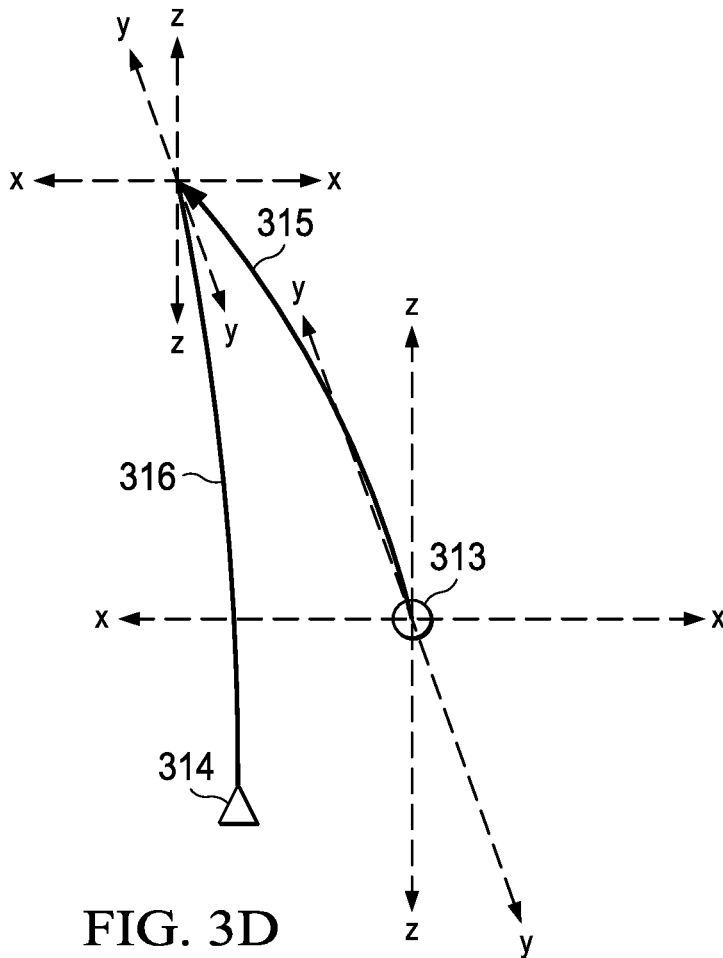


FIG. 3D

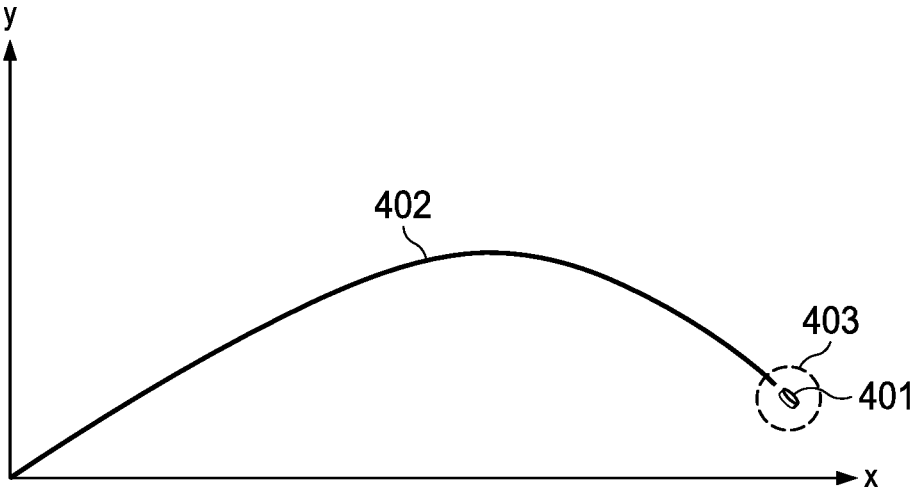


FIG. 4A

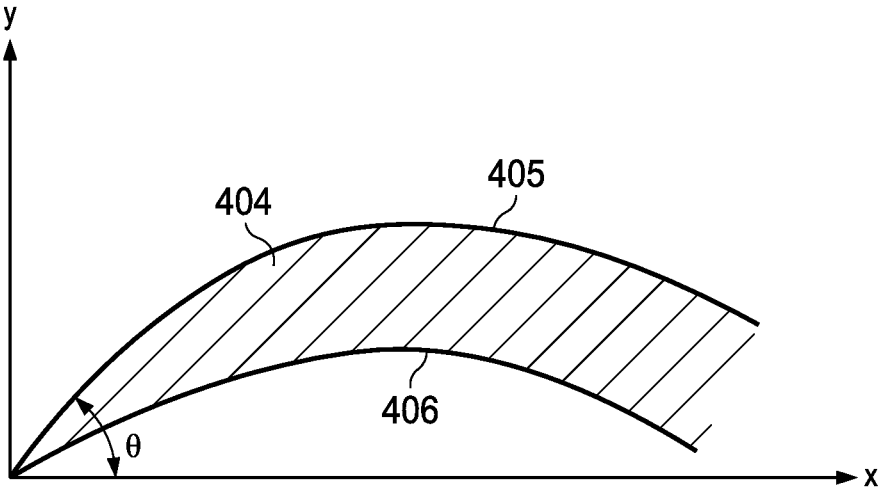


FIG. 4B

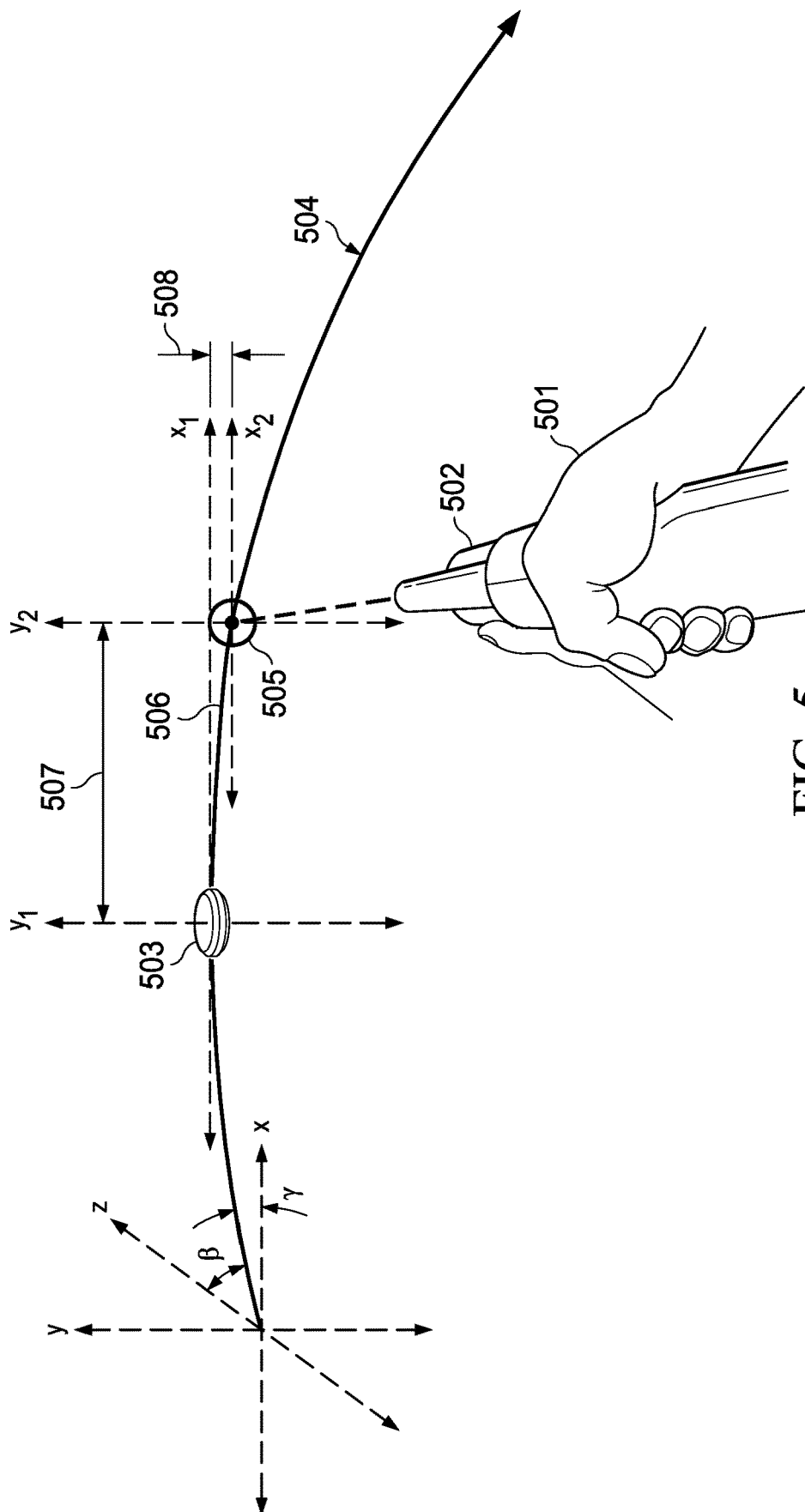


FIG. 5

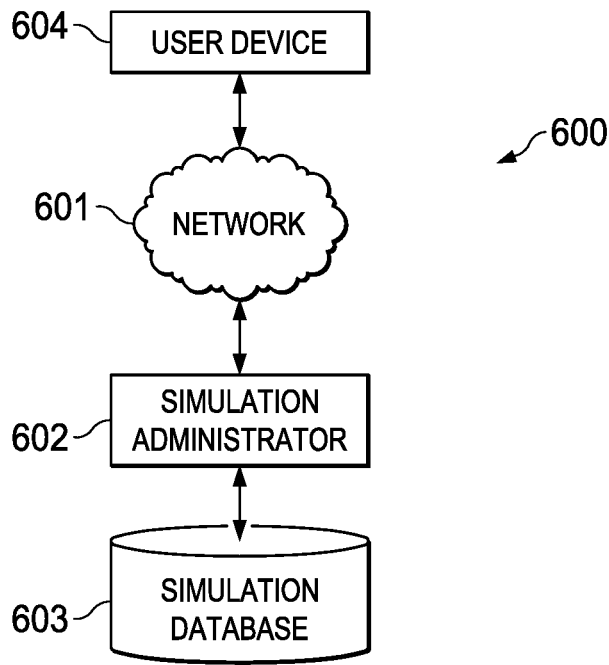


FIG. 6

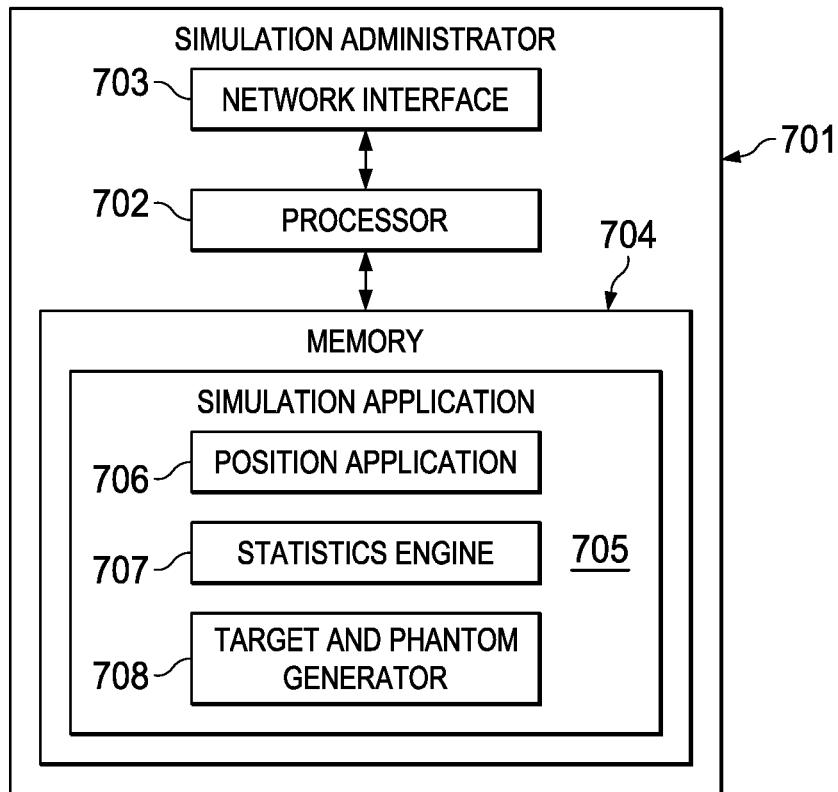


FIG. 7



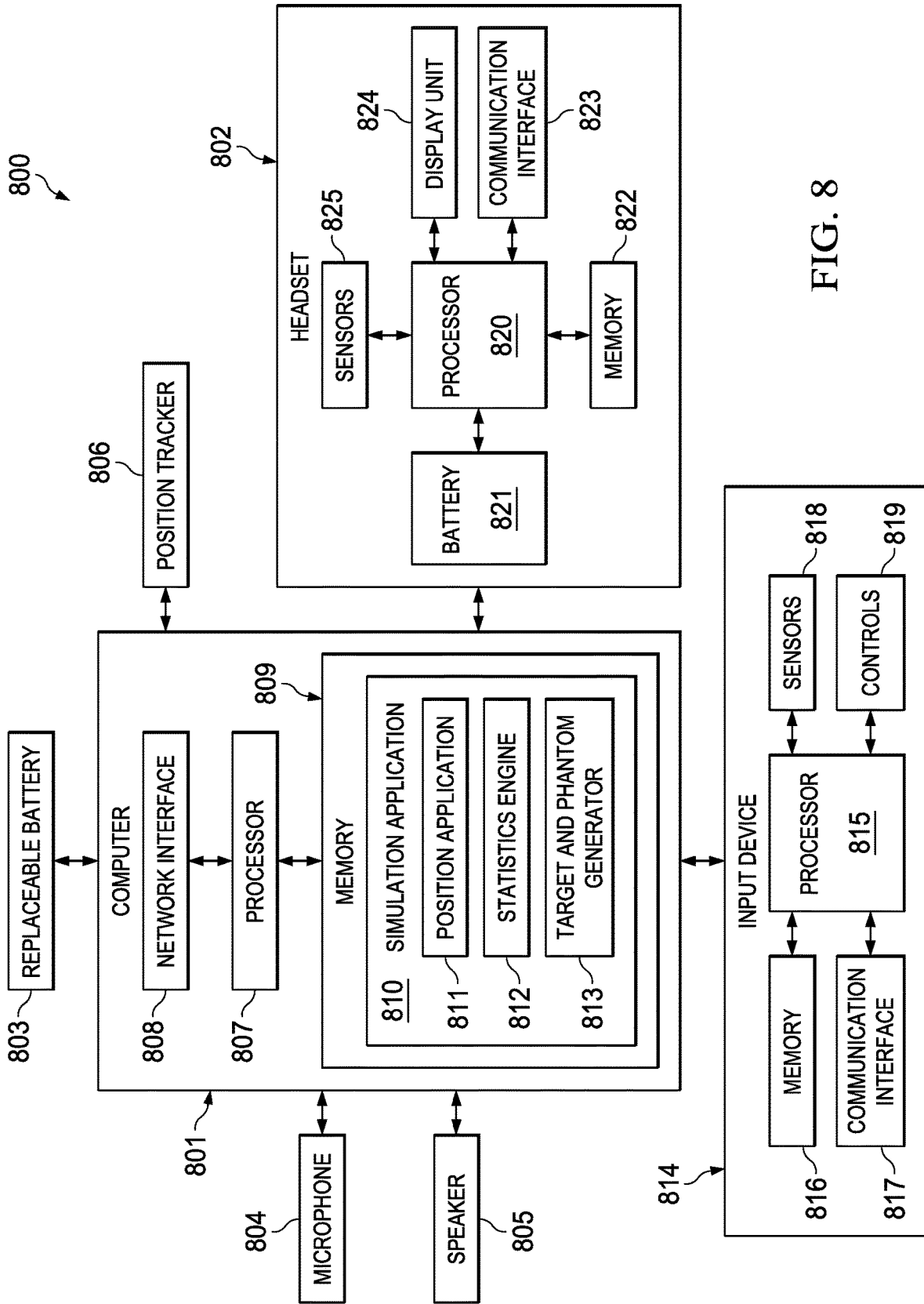


FIG. 8

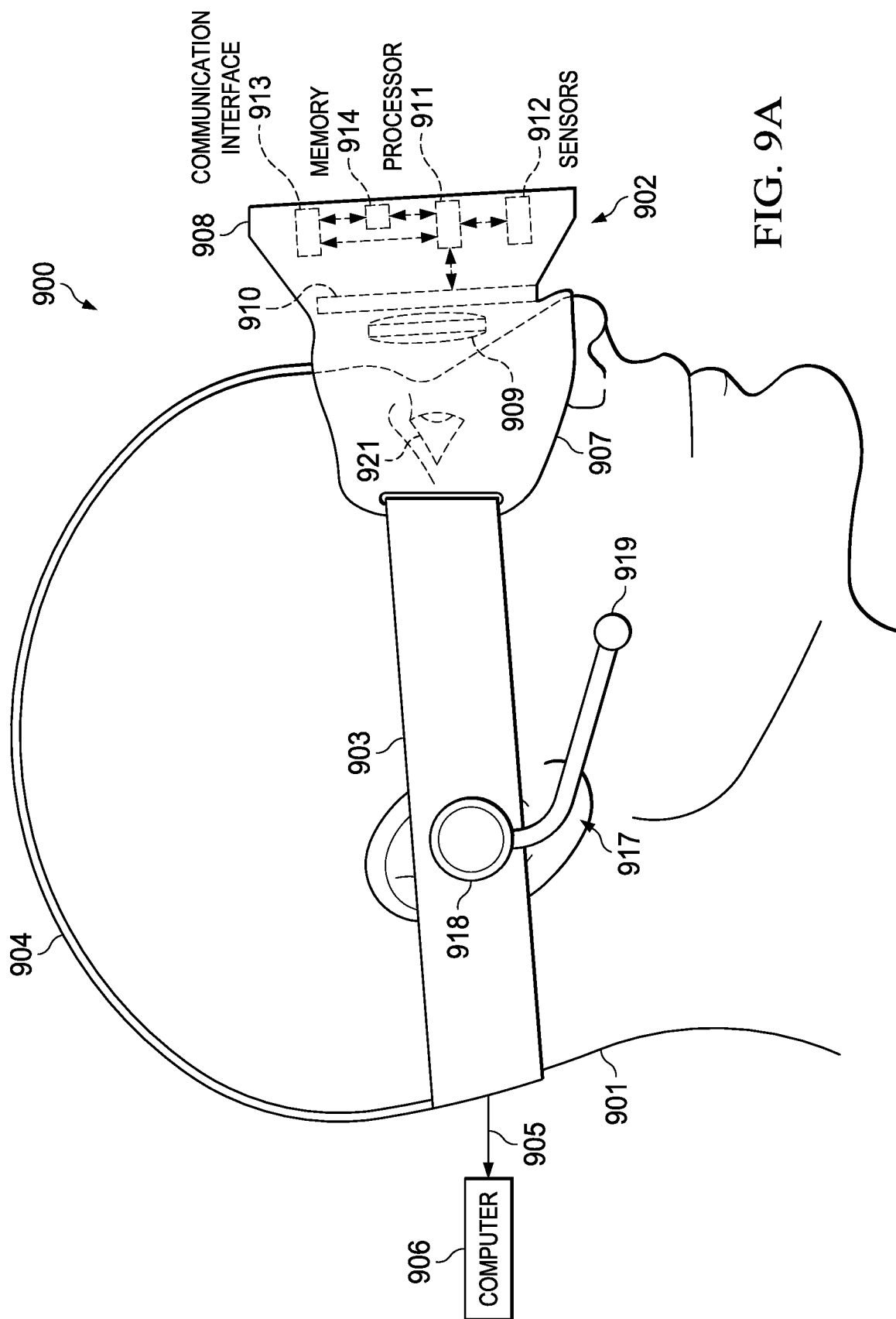


FIG. 9A

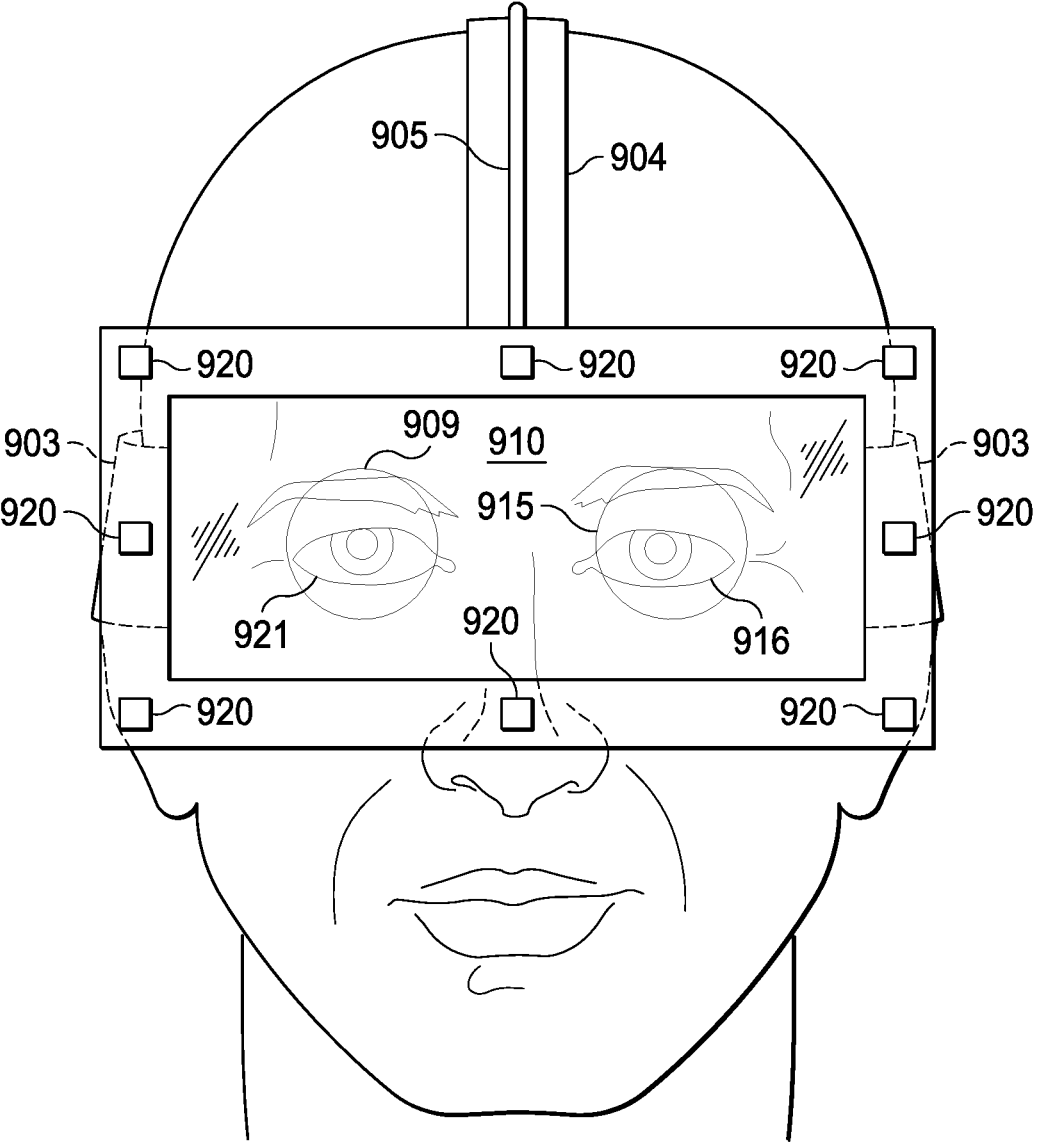
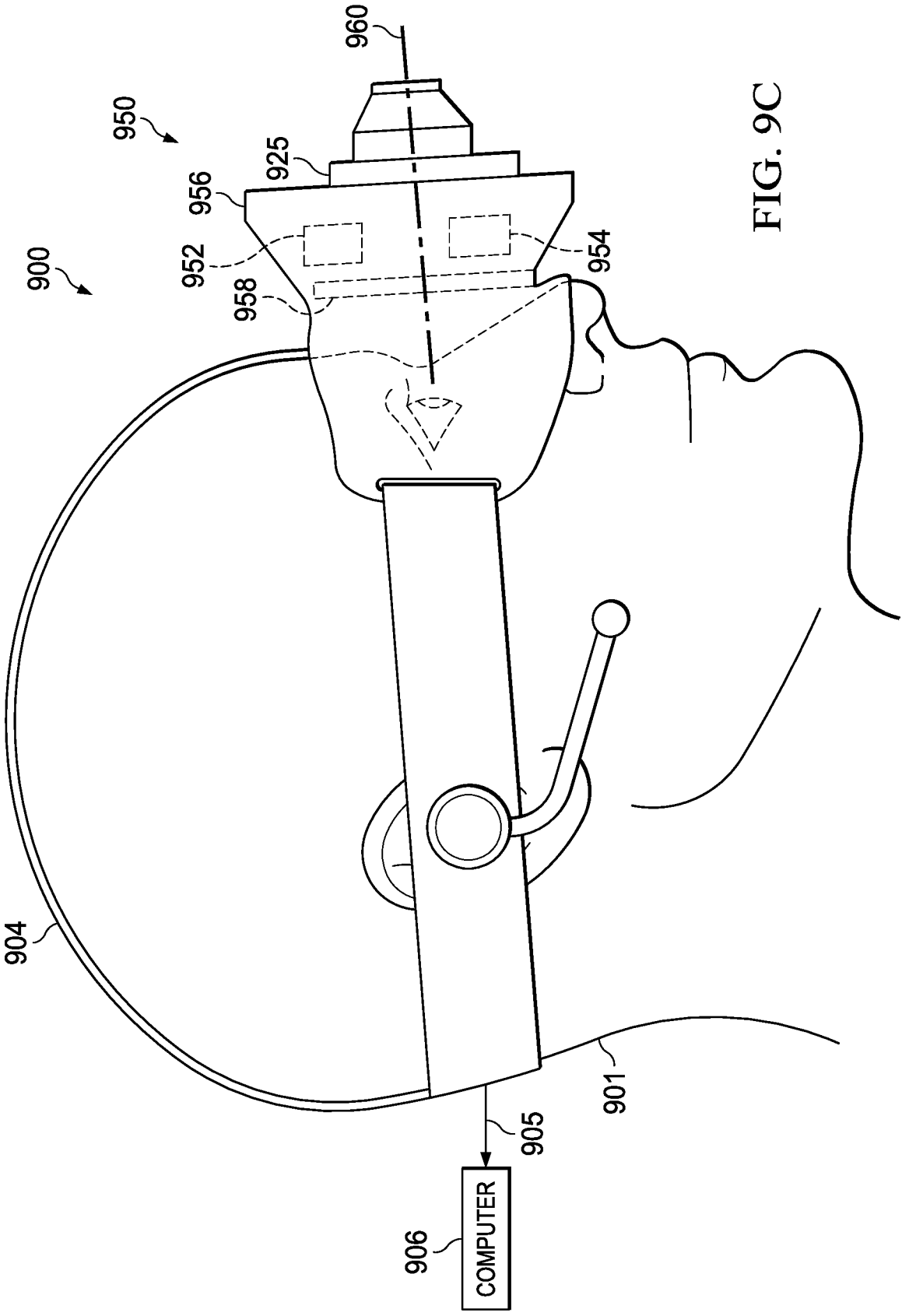


FIG. 9B



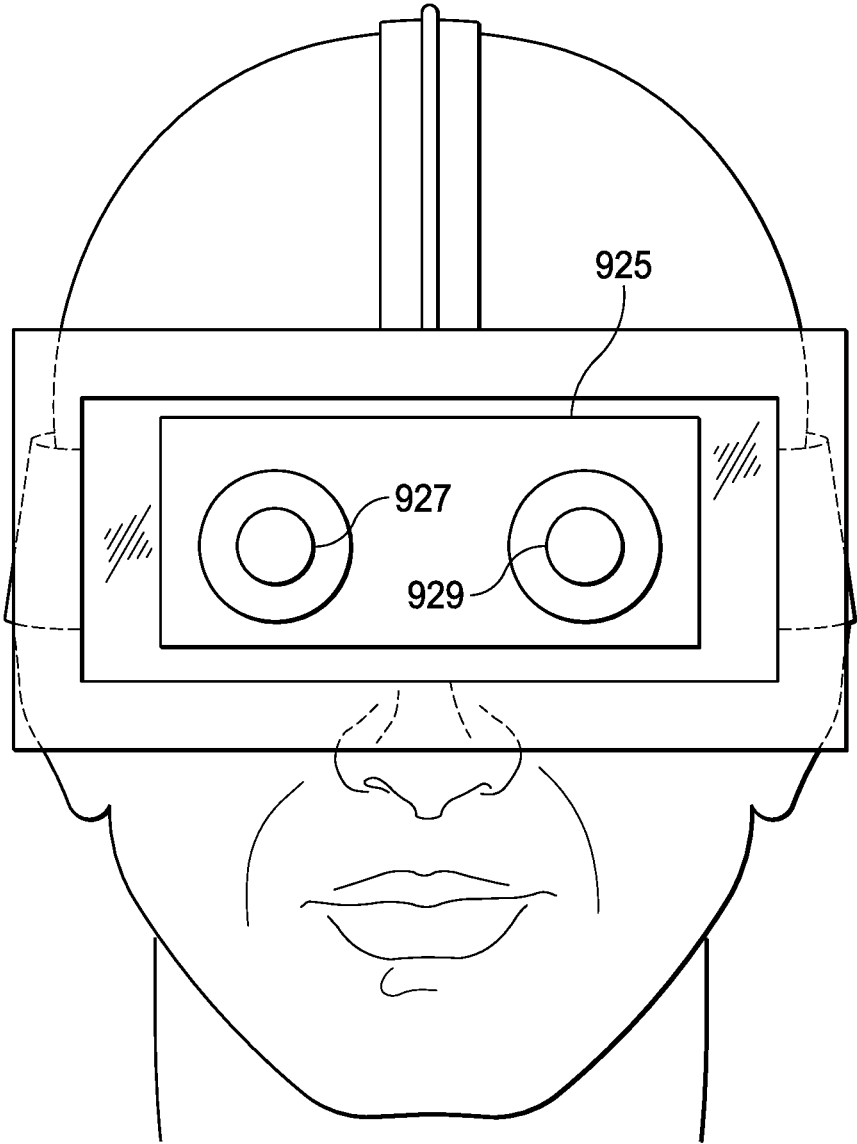
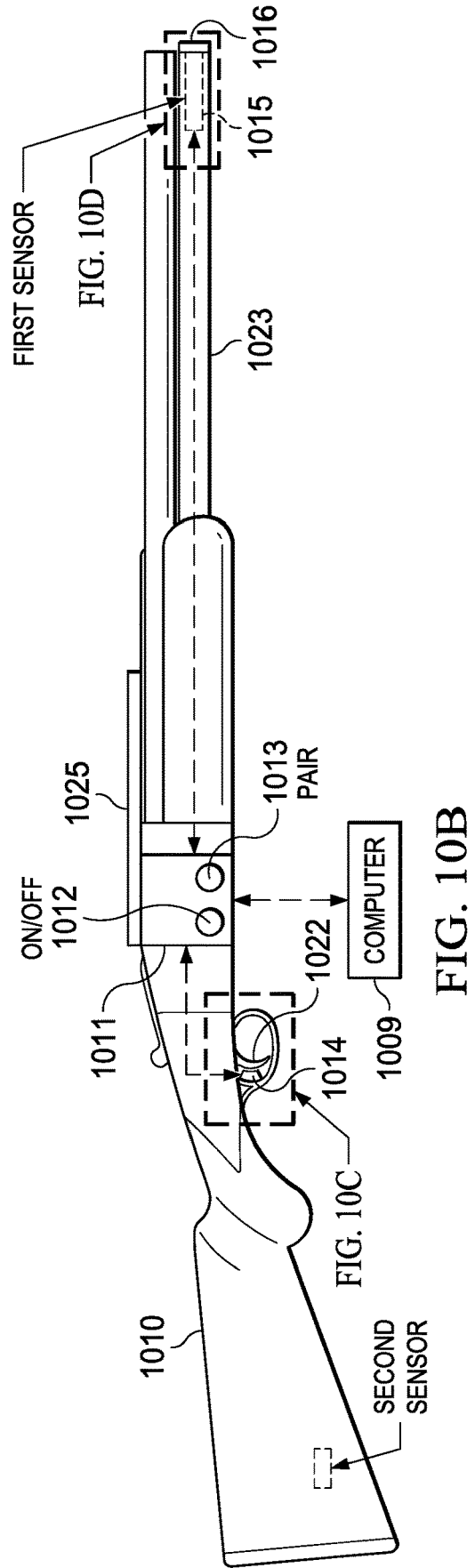
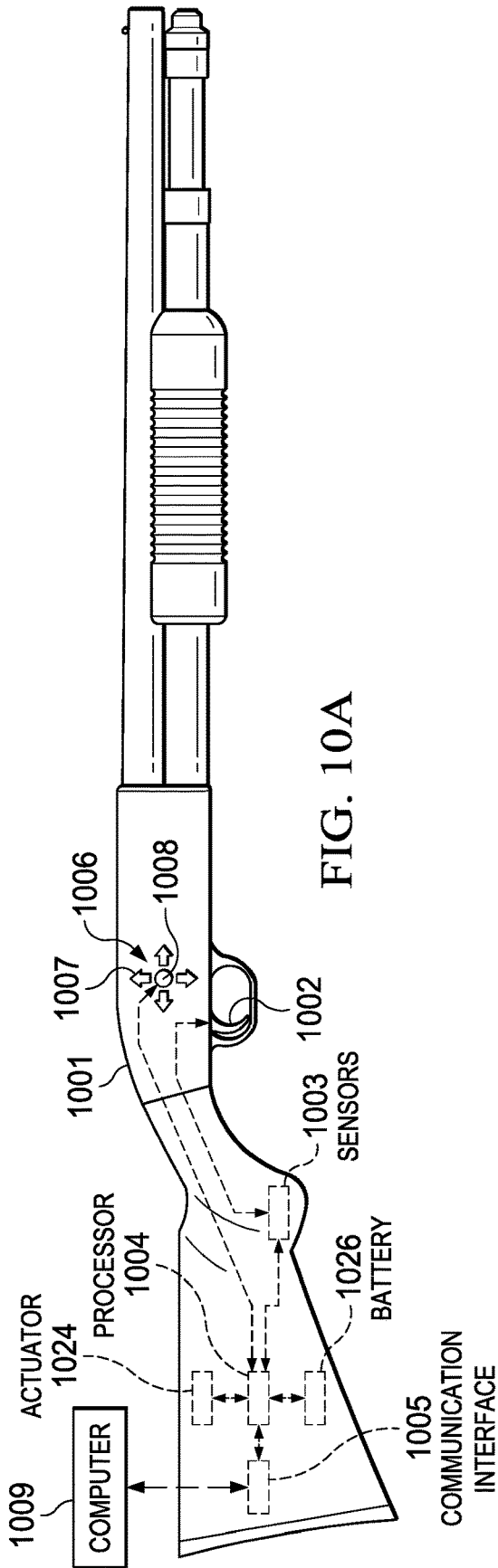


FIG. 9D



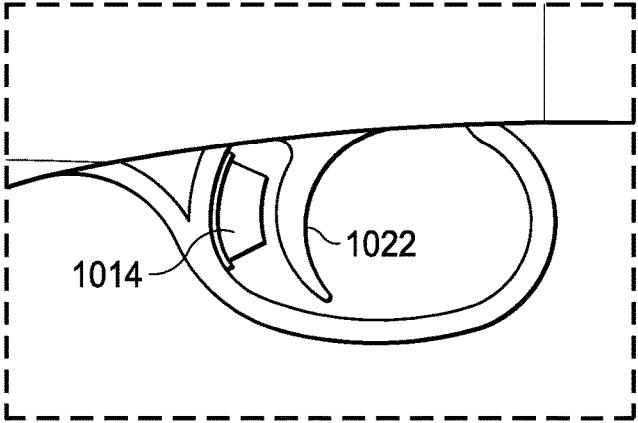


FIG. 10C

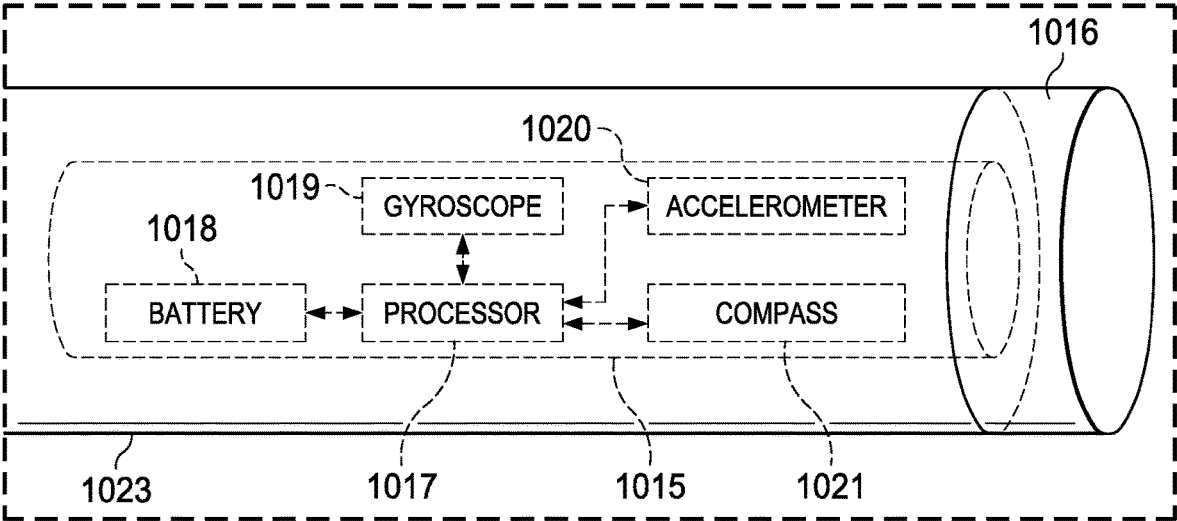


FIG. 10D

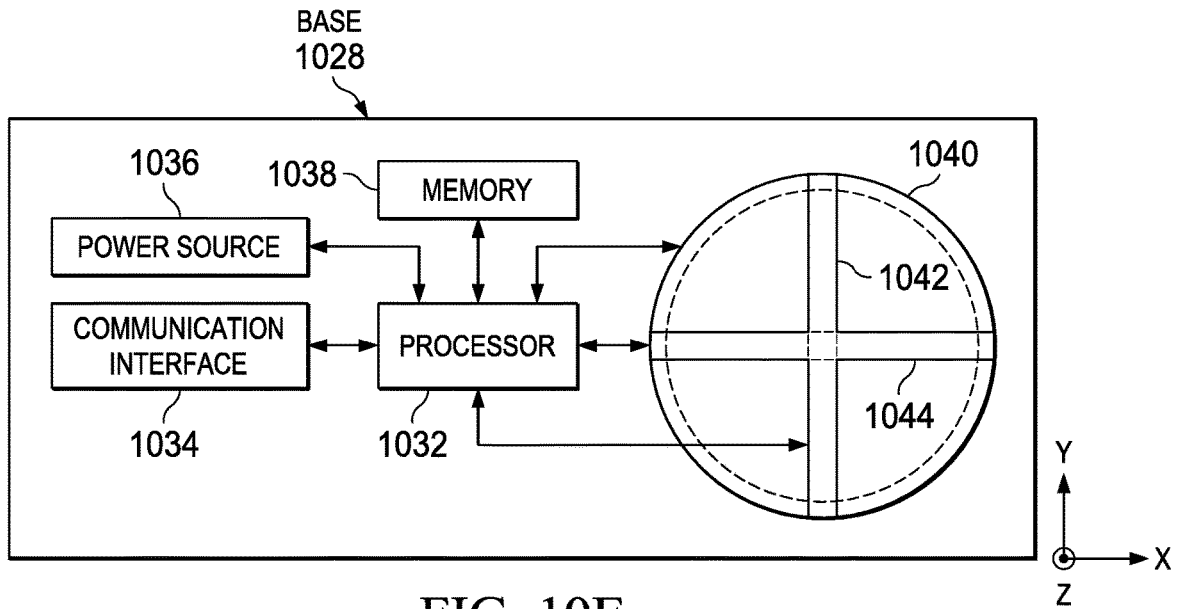


FIG. 10E

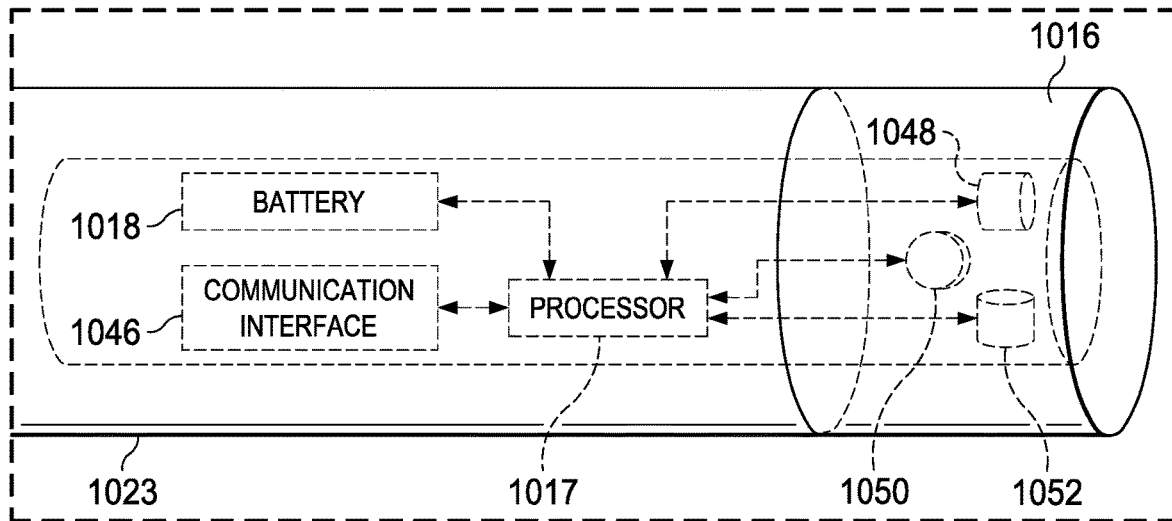


FIG. 10F



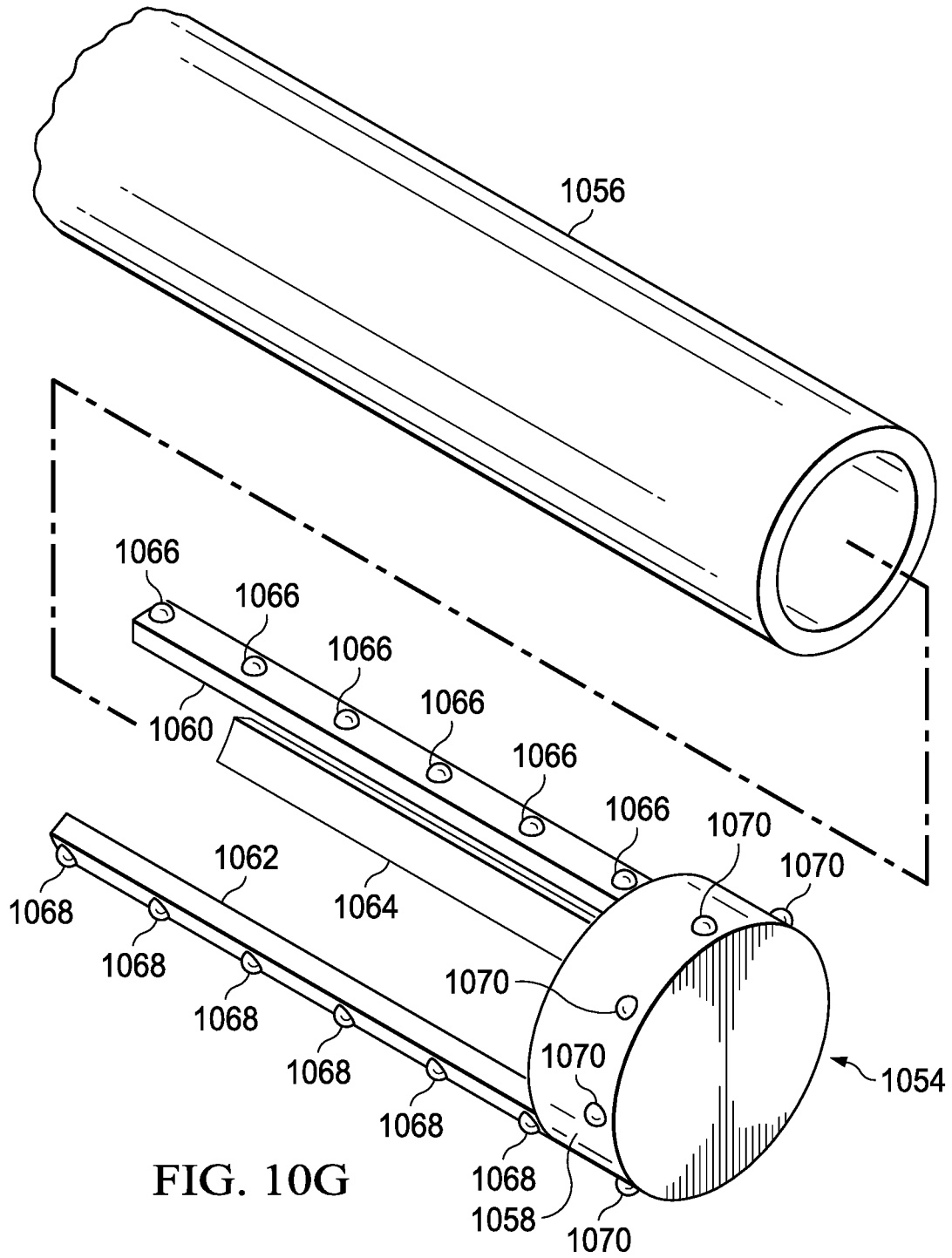


FIG. 10G

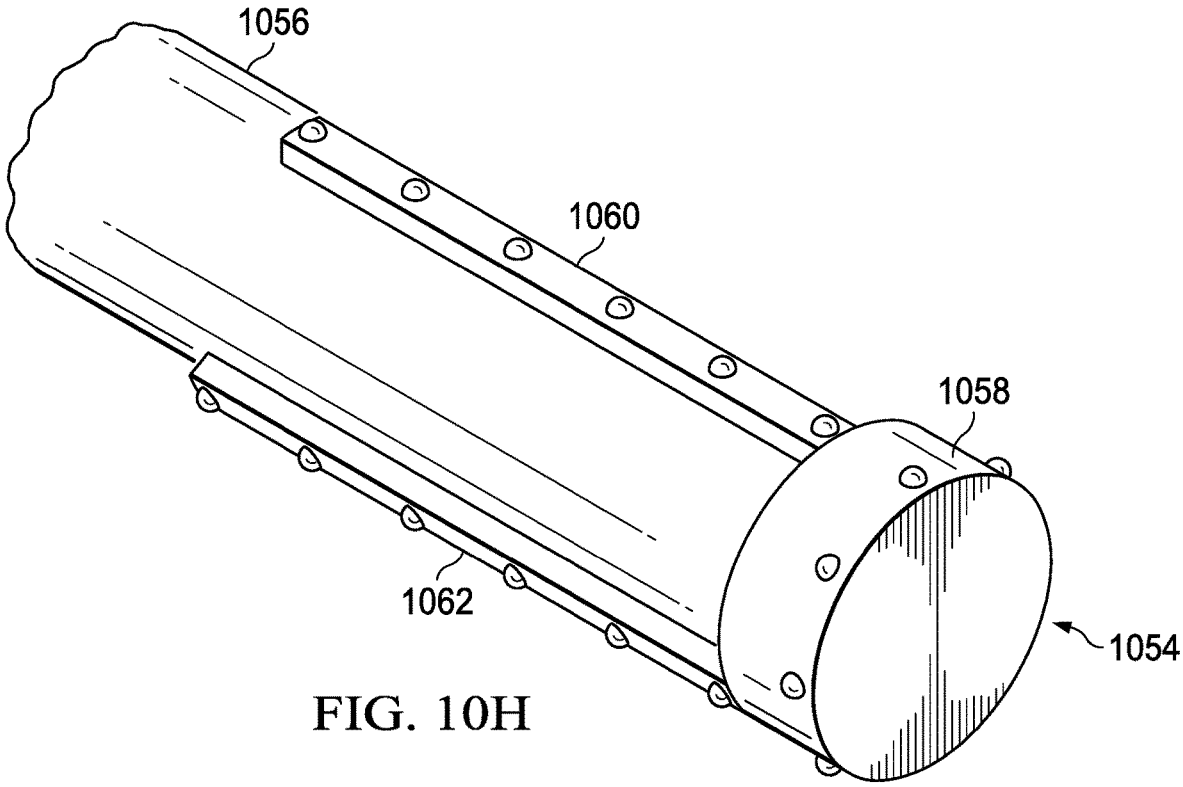


FIG. 10H

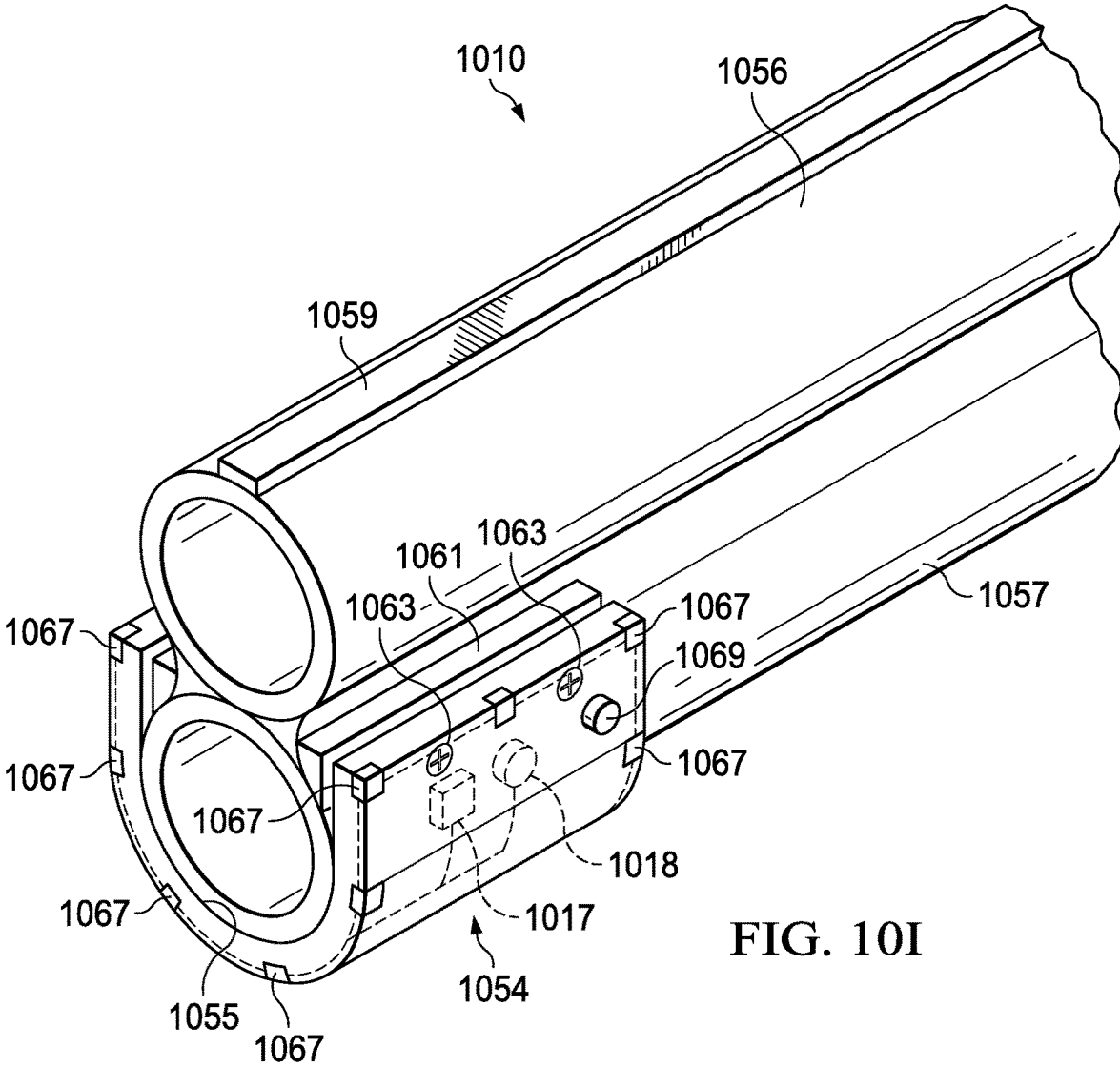


FIG. 10I

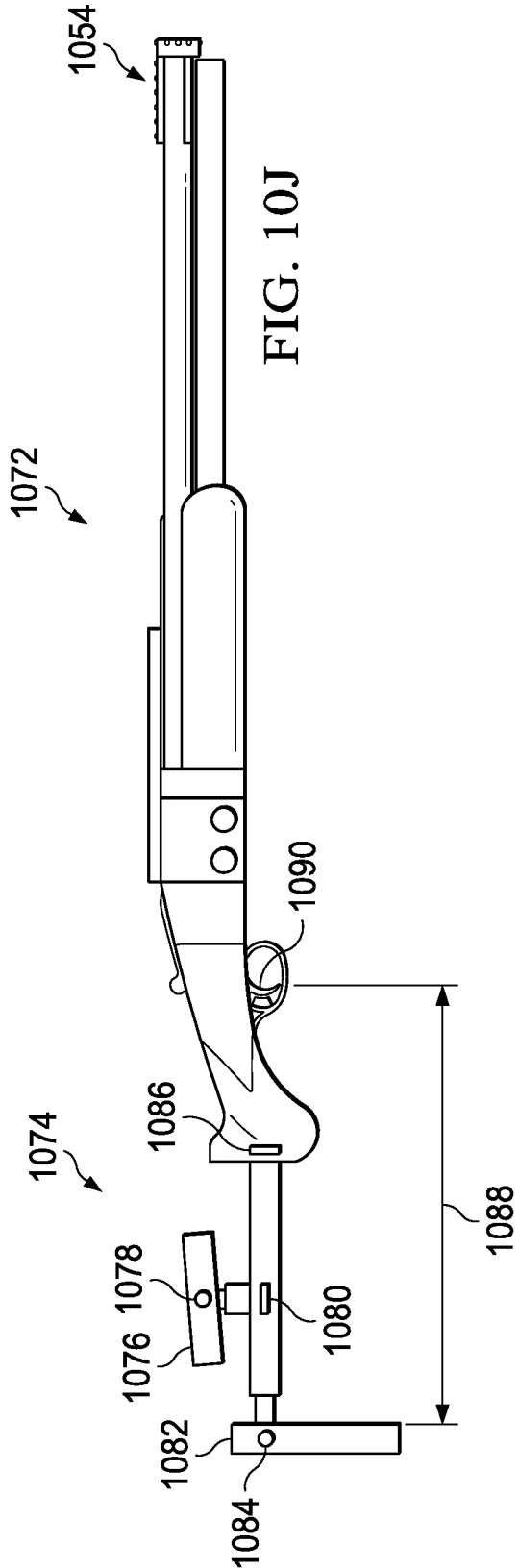


FIG. 10J

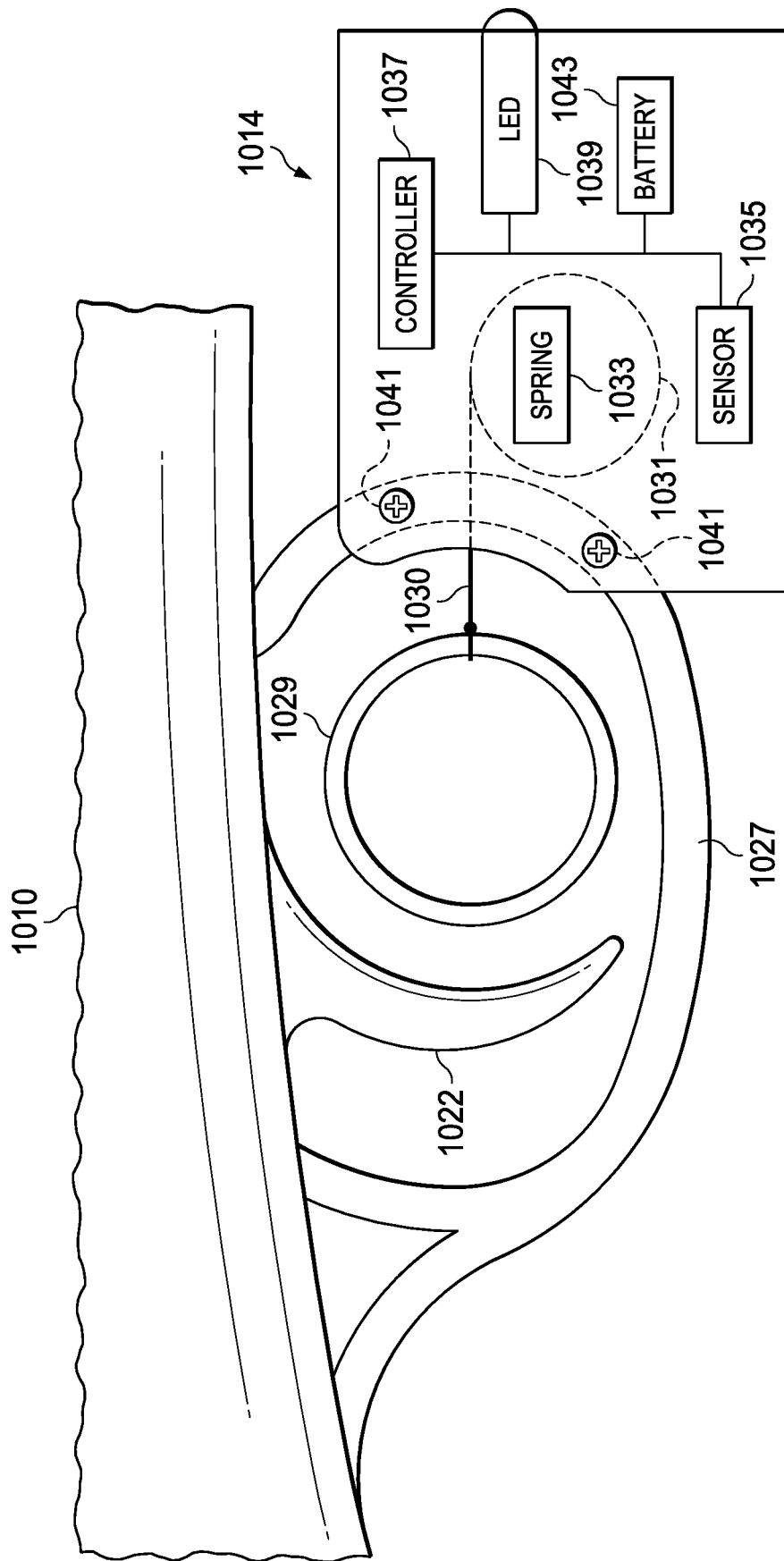


FIG. 10K

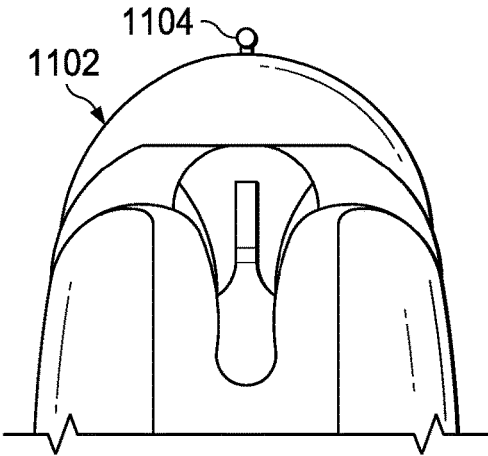


FIG. 11A

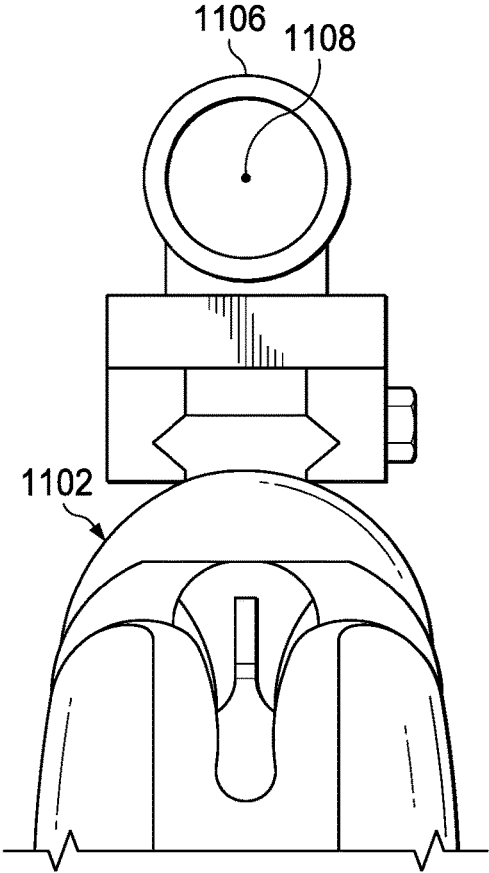


FIG. 11B

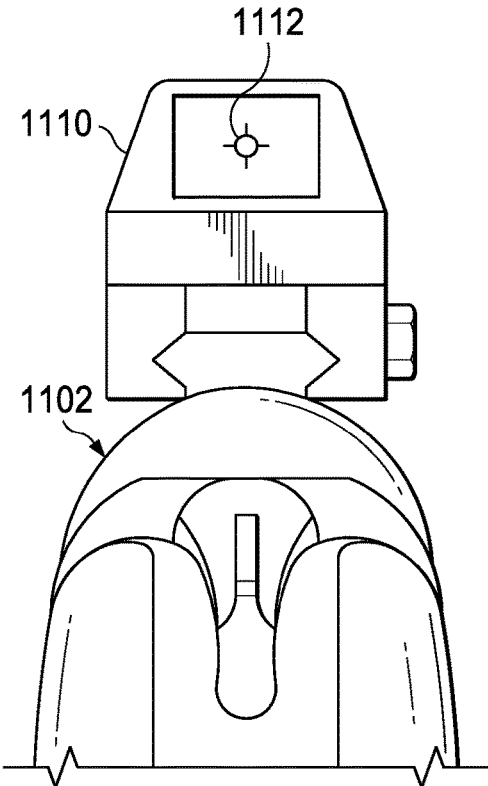


FIG. 11C

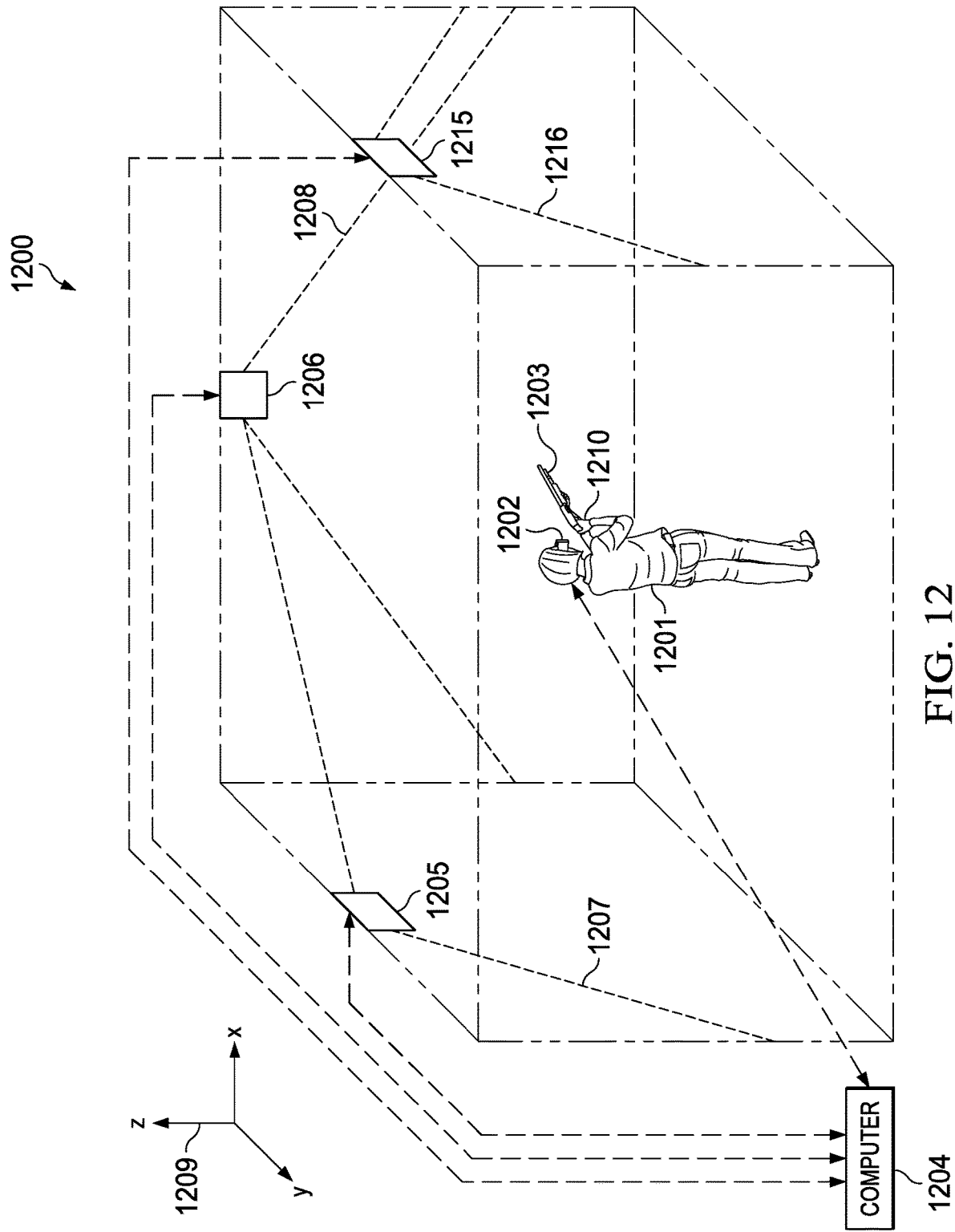


FIG. 12

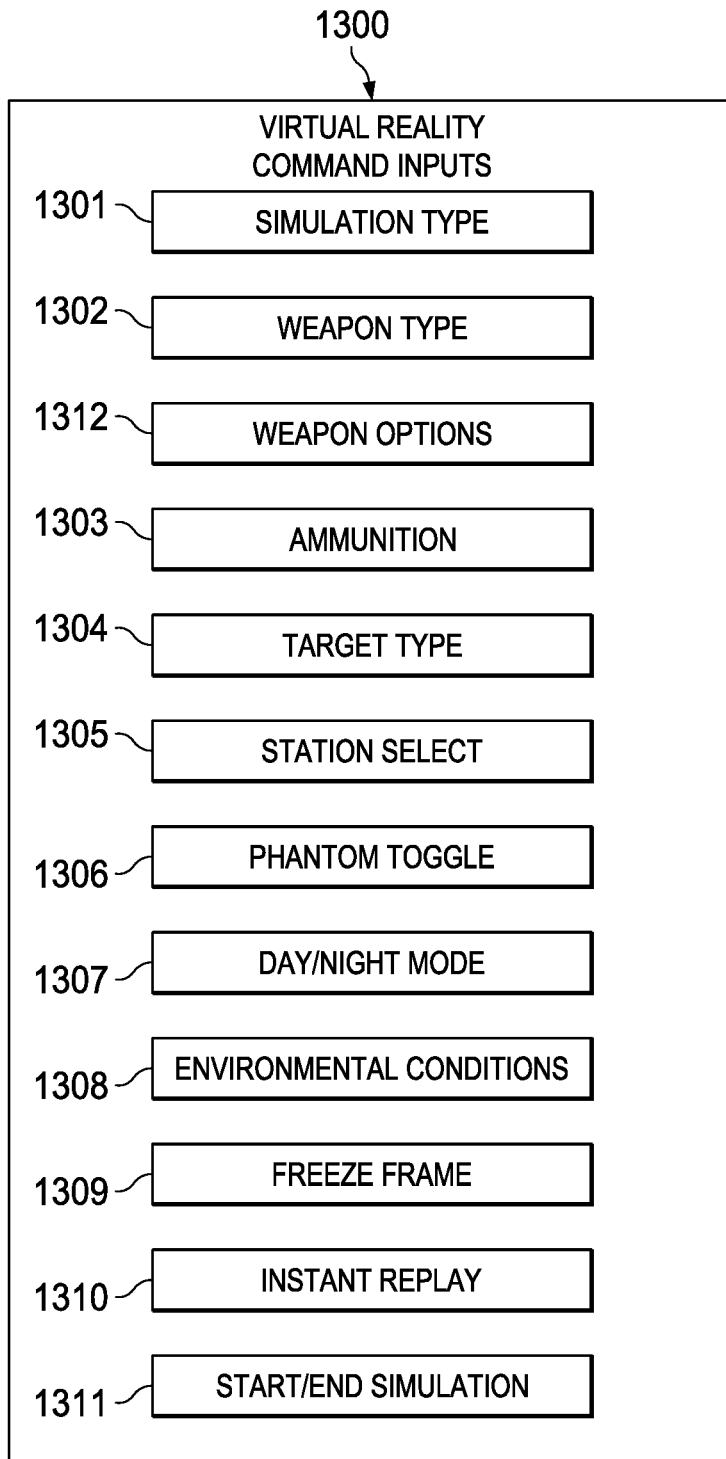


FIG. 13



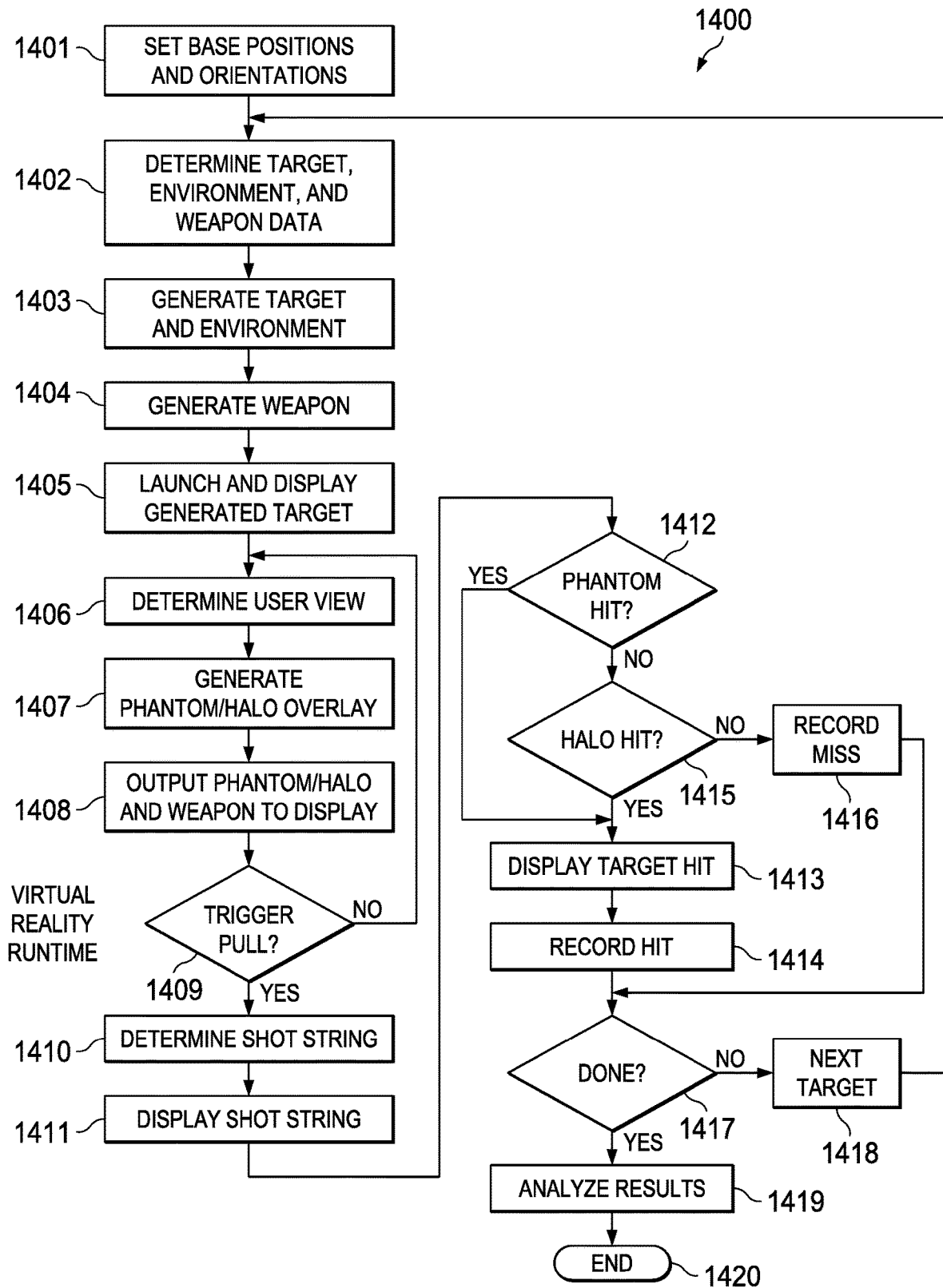


FIG. 14

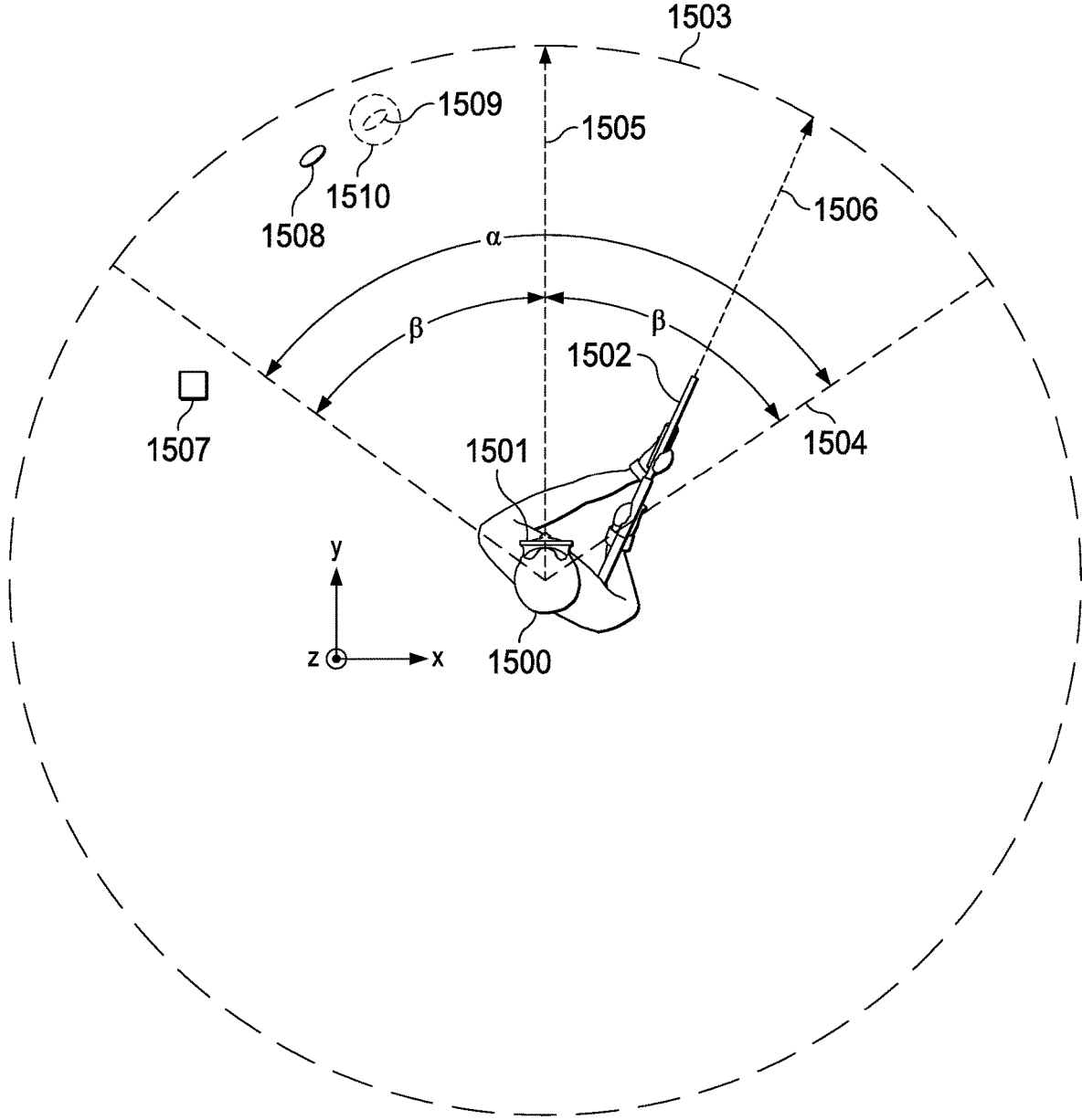


FIG. 15A

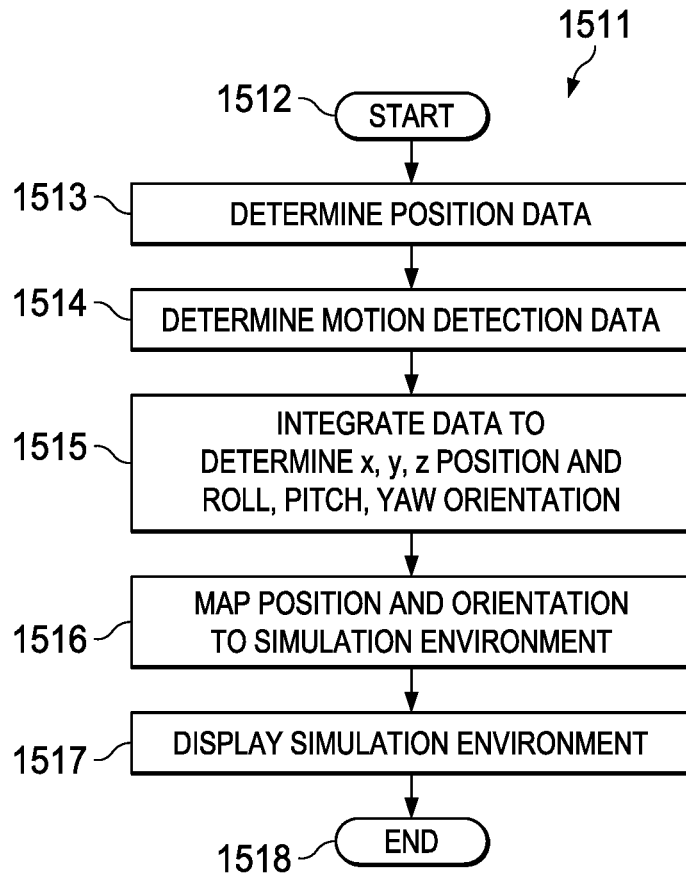


FIG. 15B

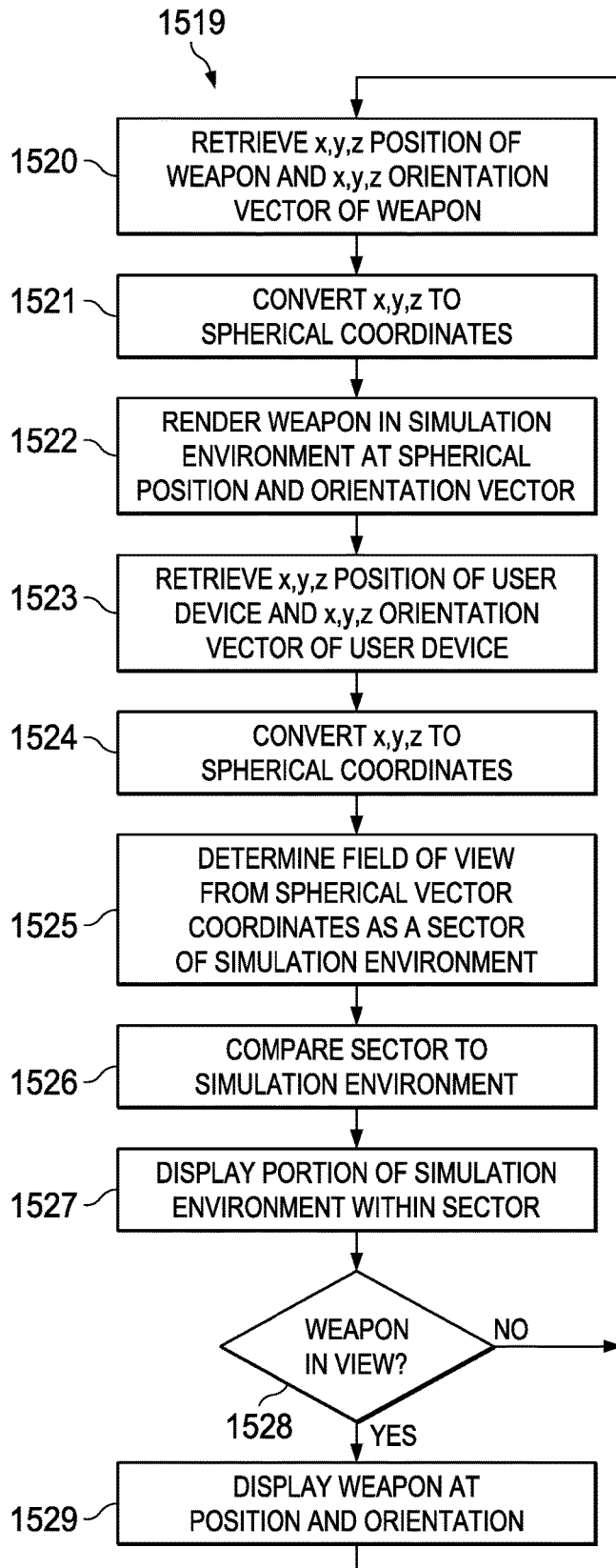


FIG. 15C

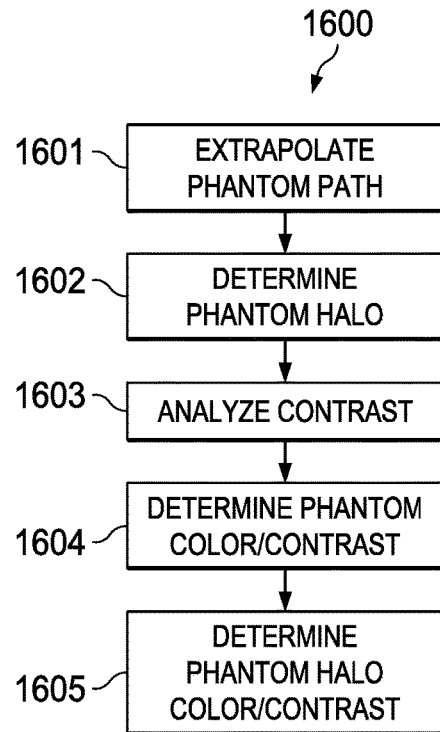


FIG. 16A

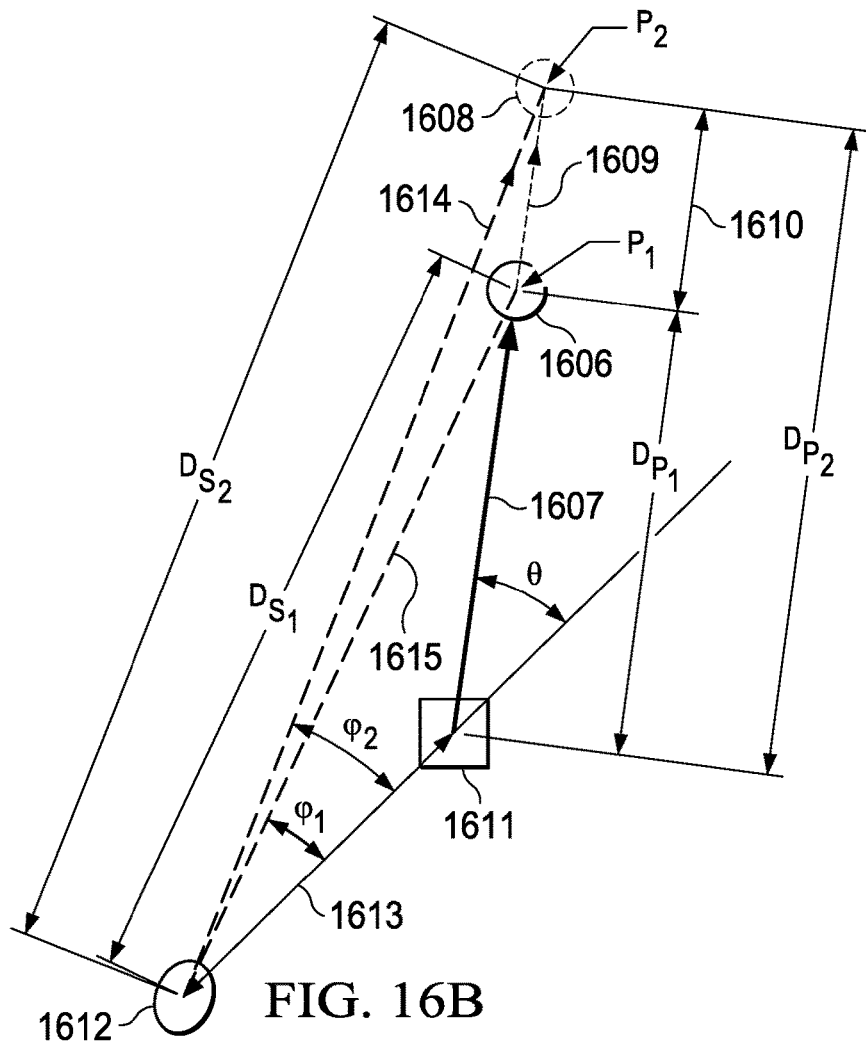


FIG. 16B

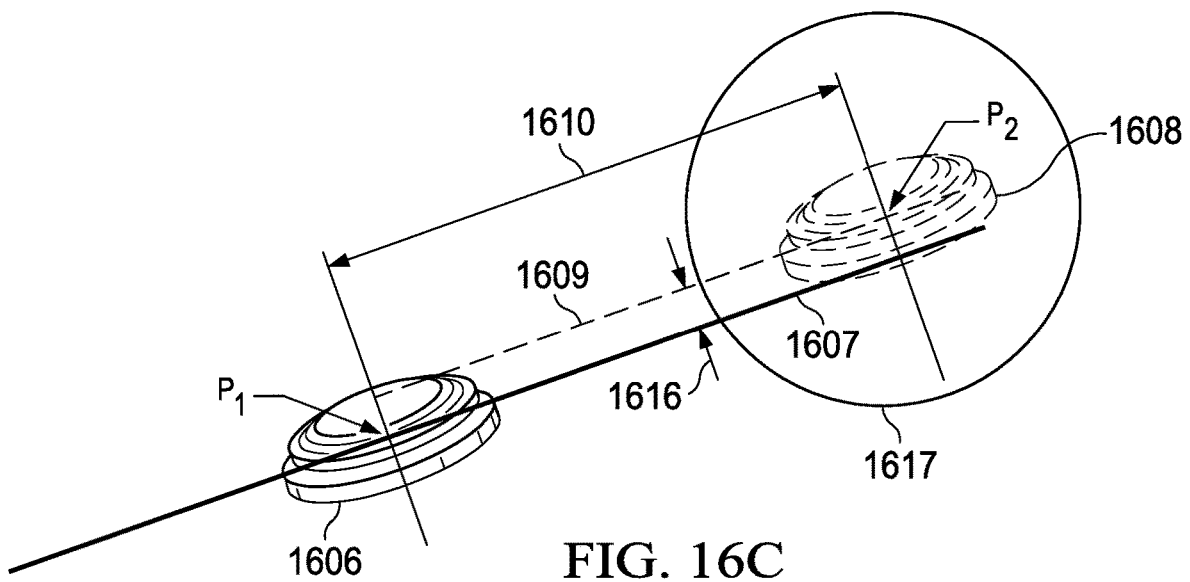


FIG. 16C

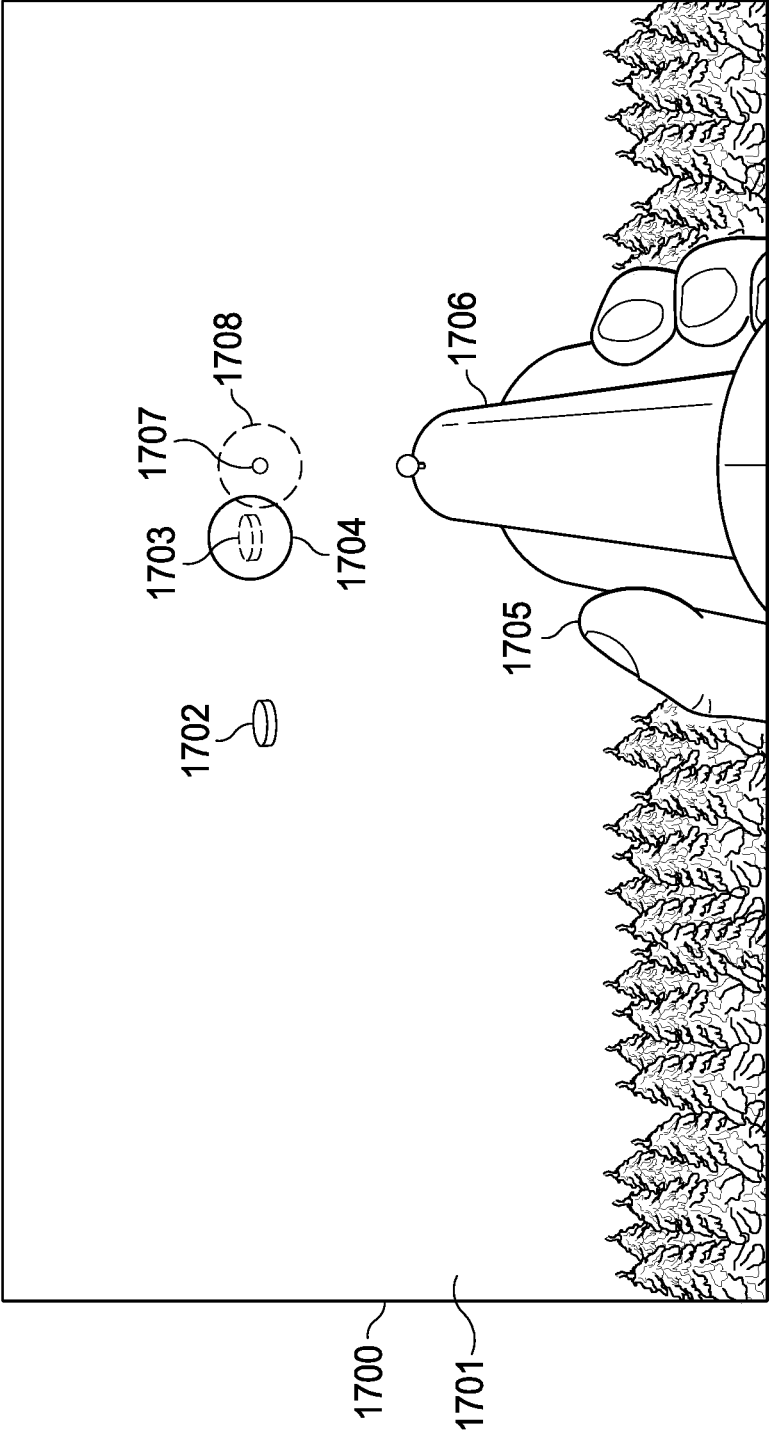


FIG. 17

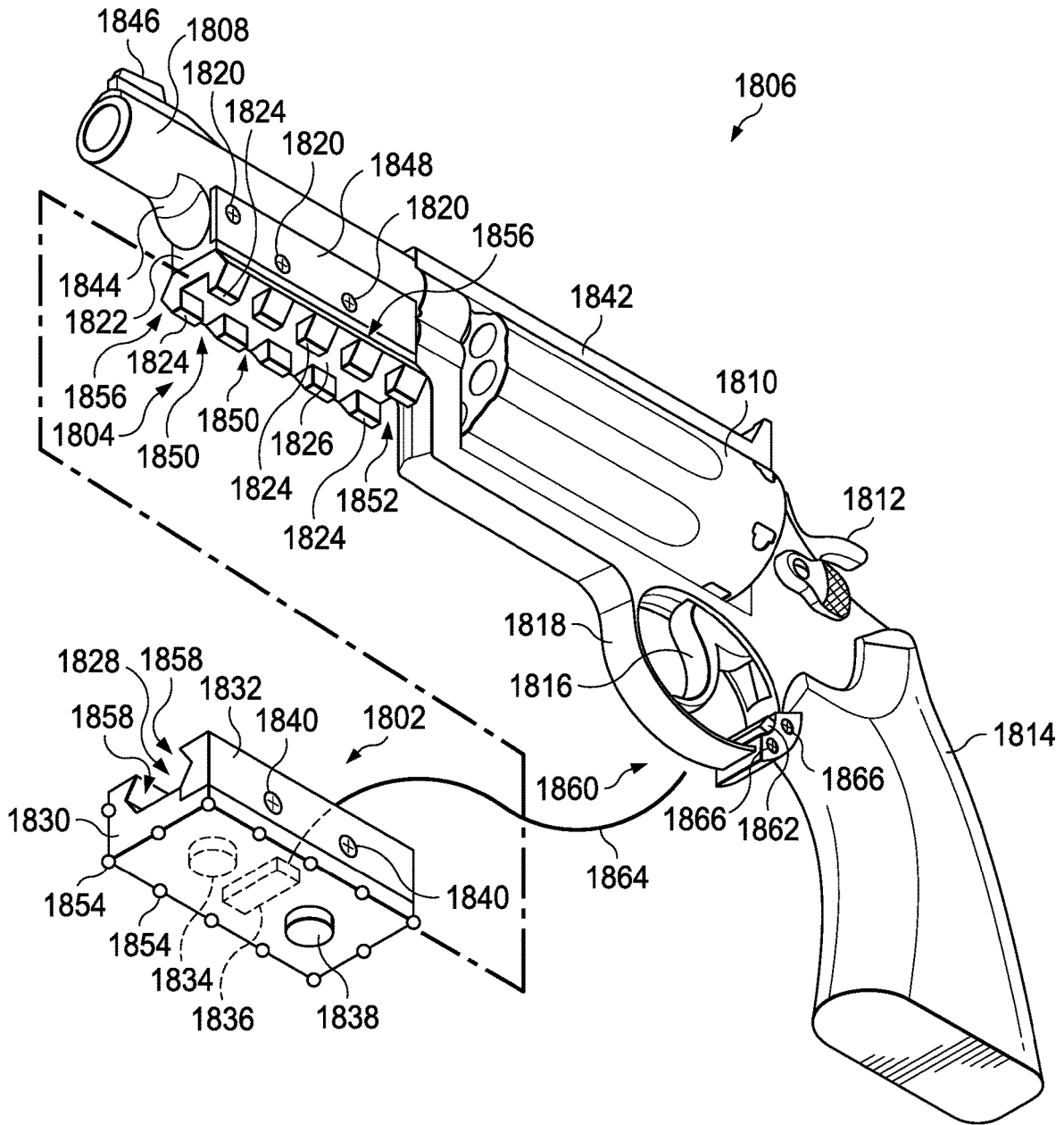


FIG. 18

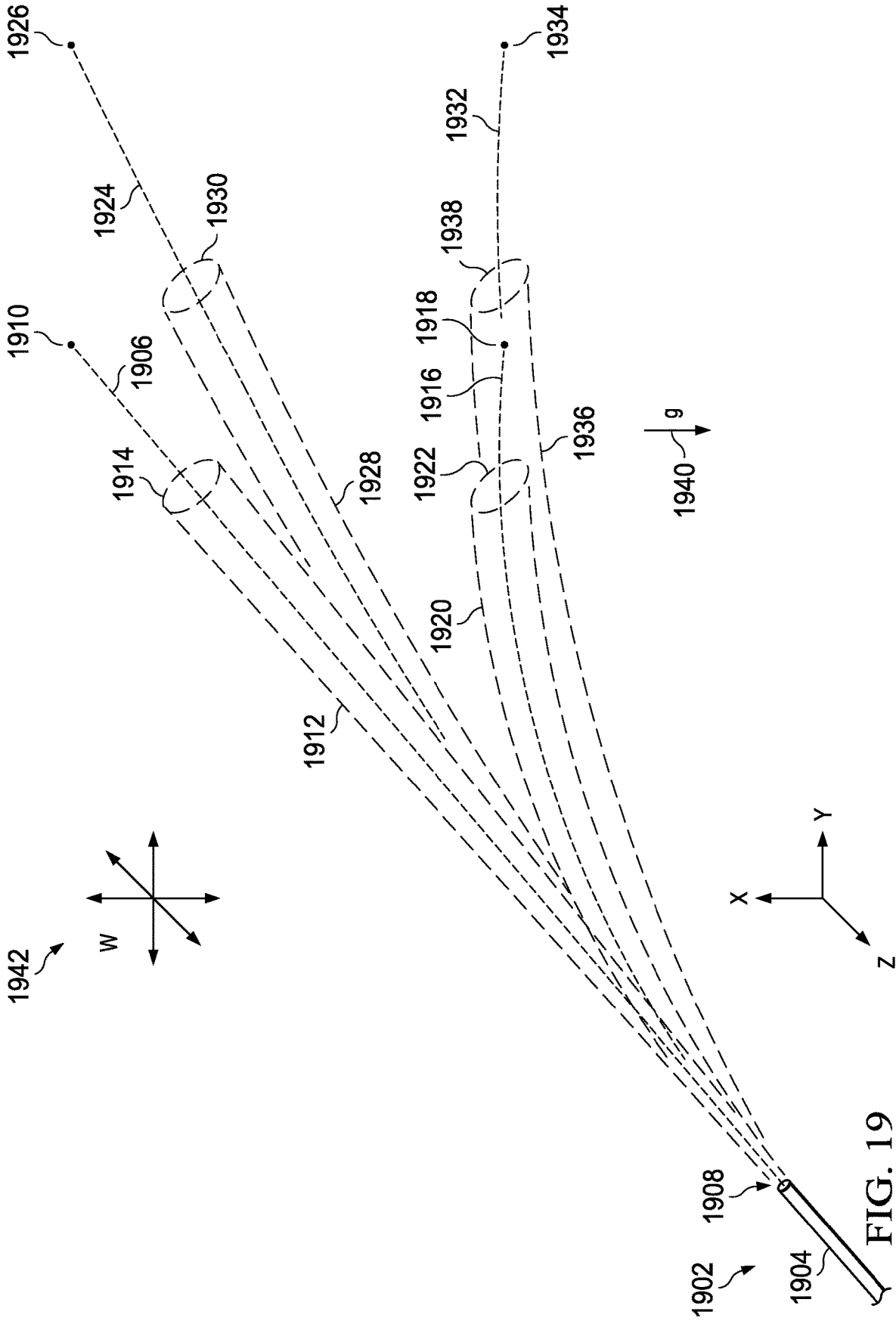


FIG. 19



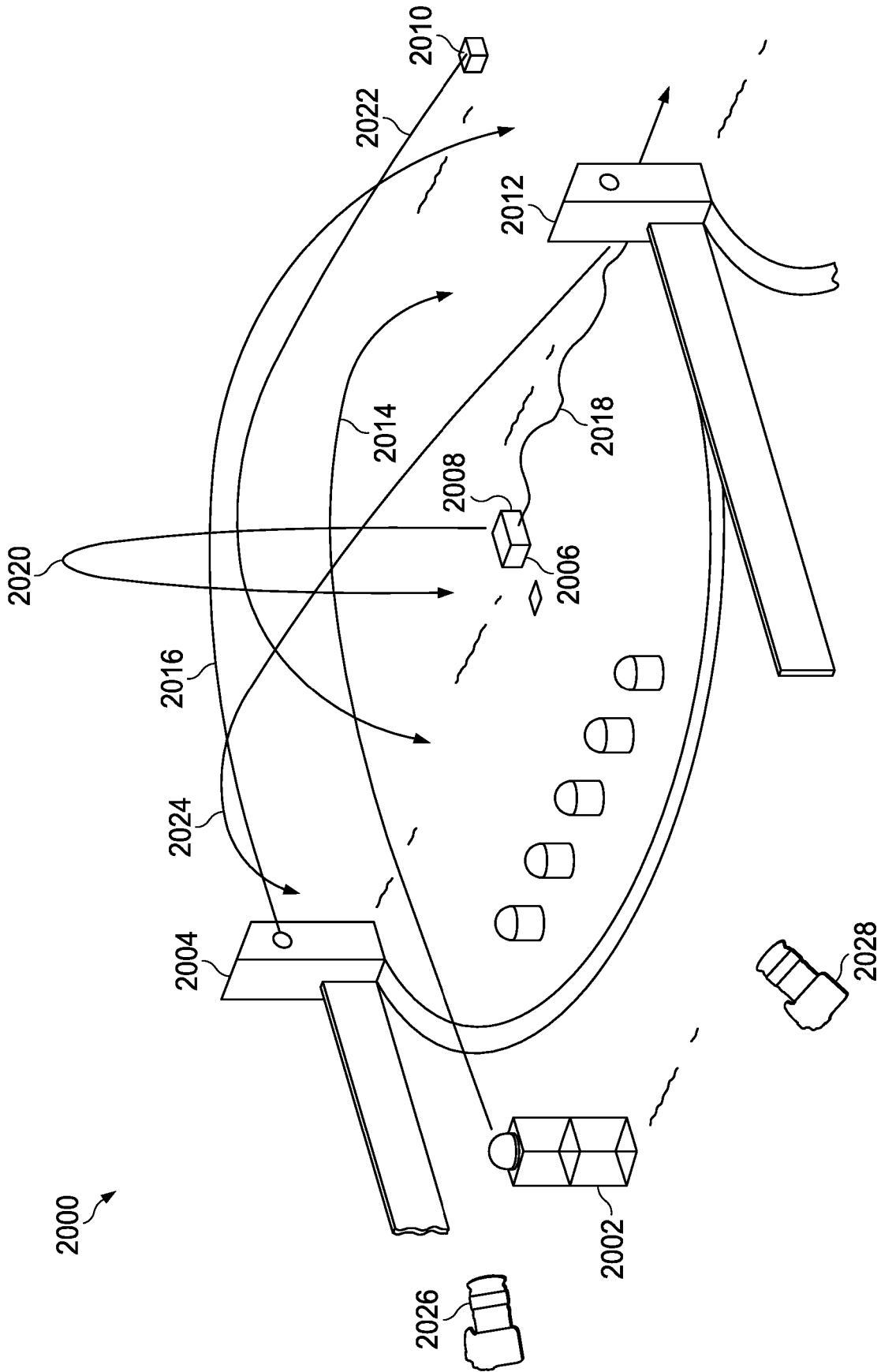


FIG. 20A

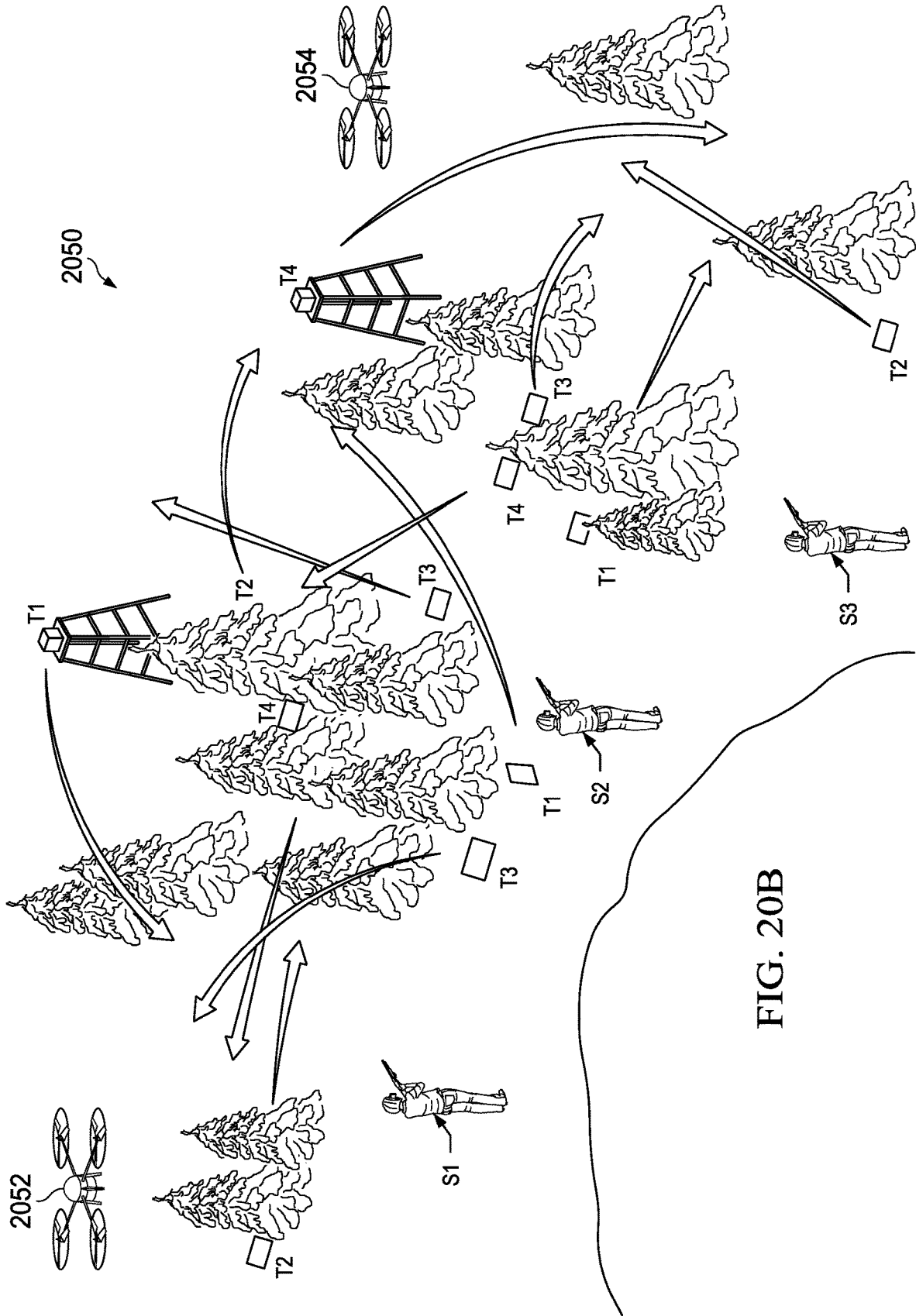


FIG. 20B

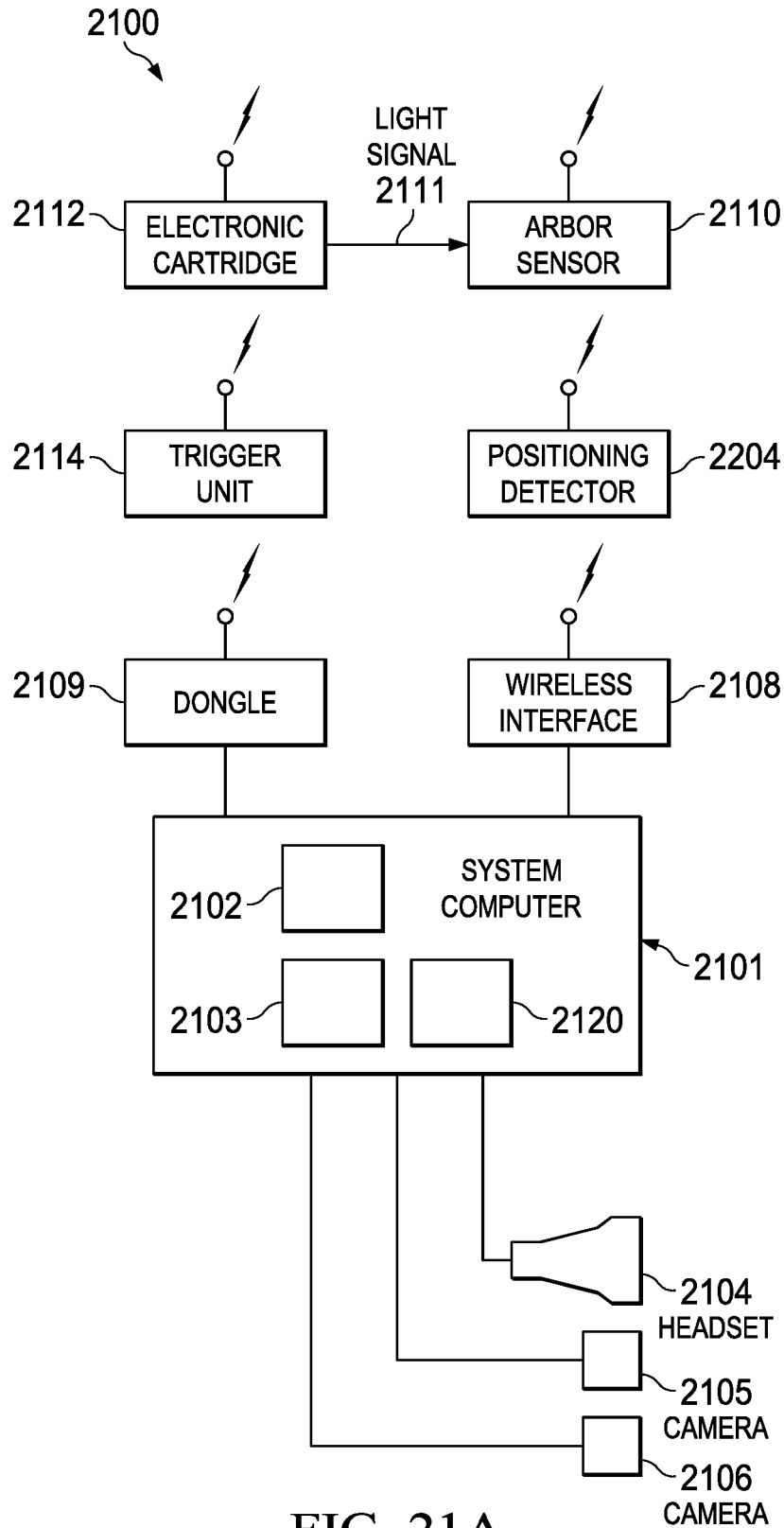


FIG. 21A

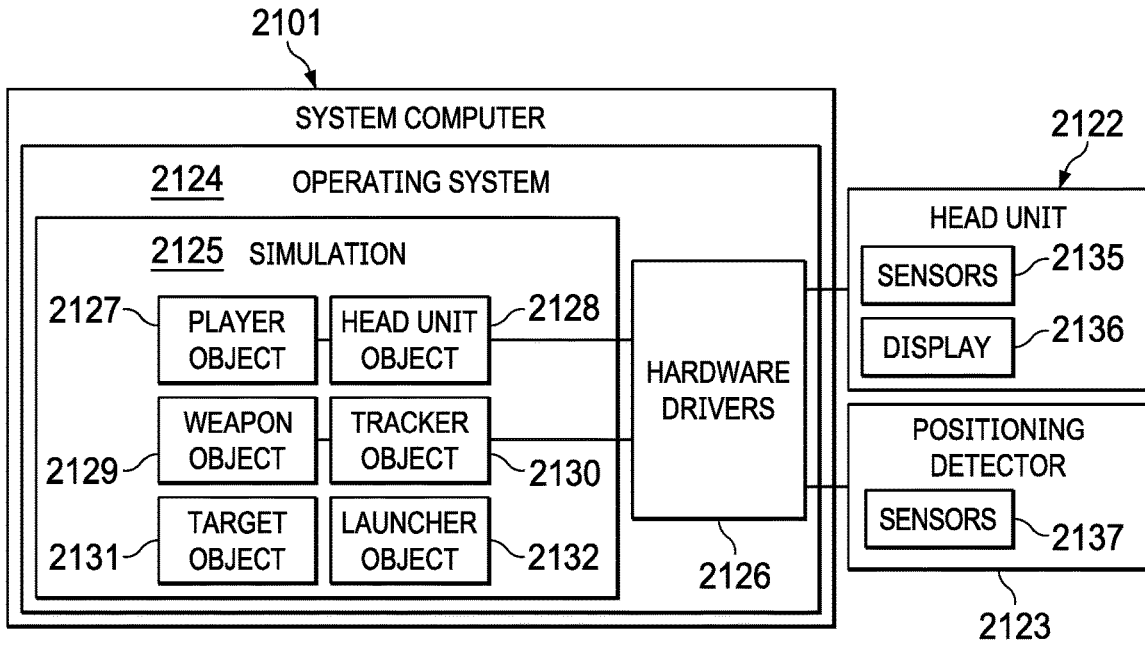


FIG. 21B

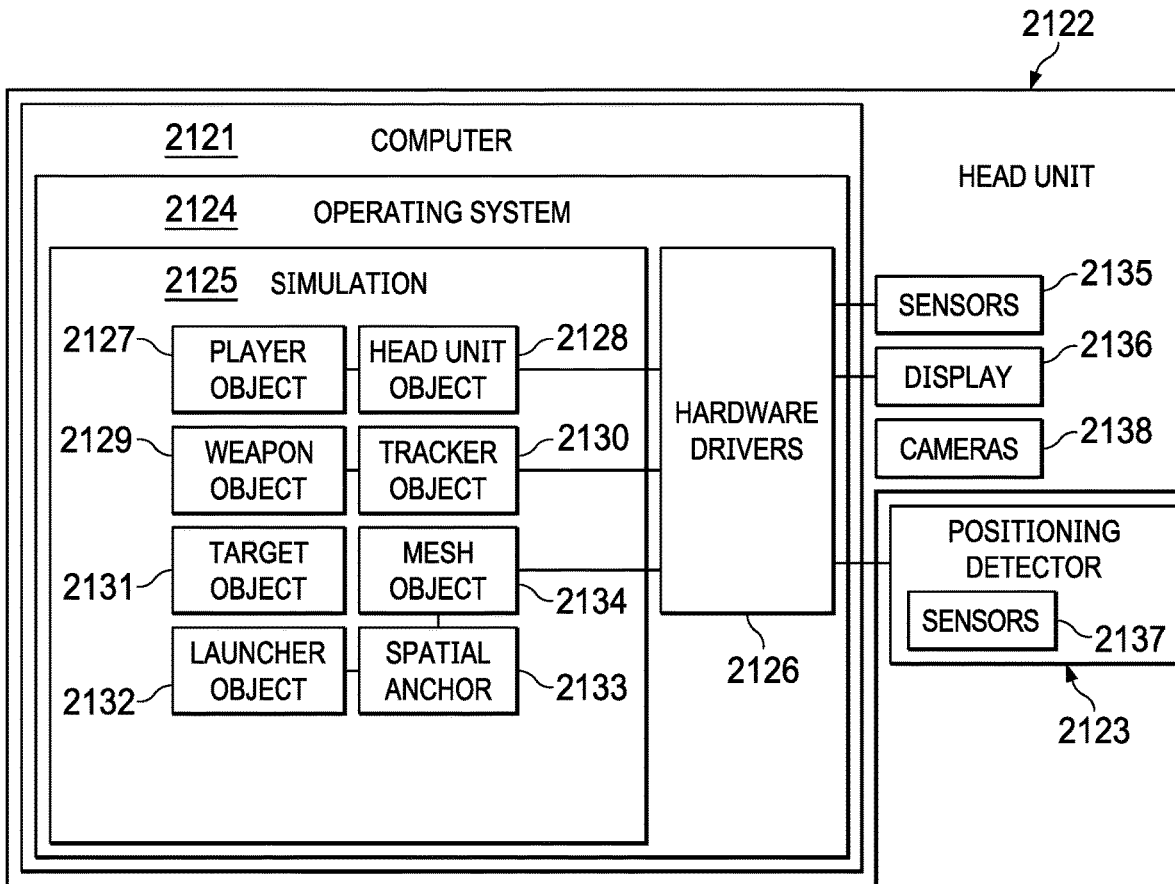
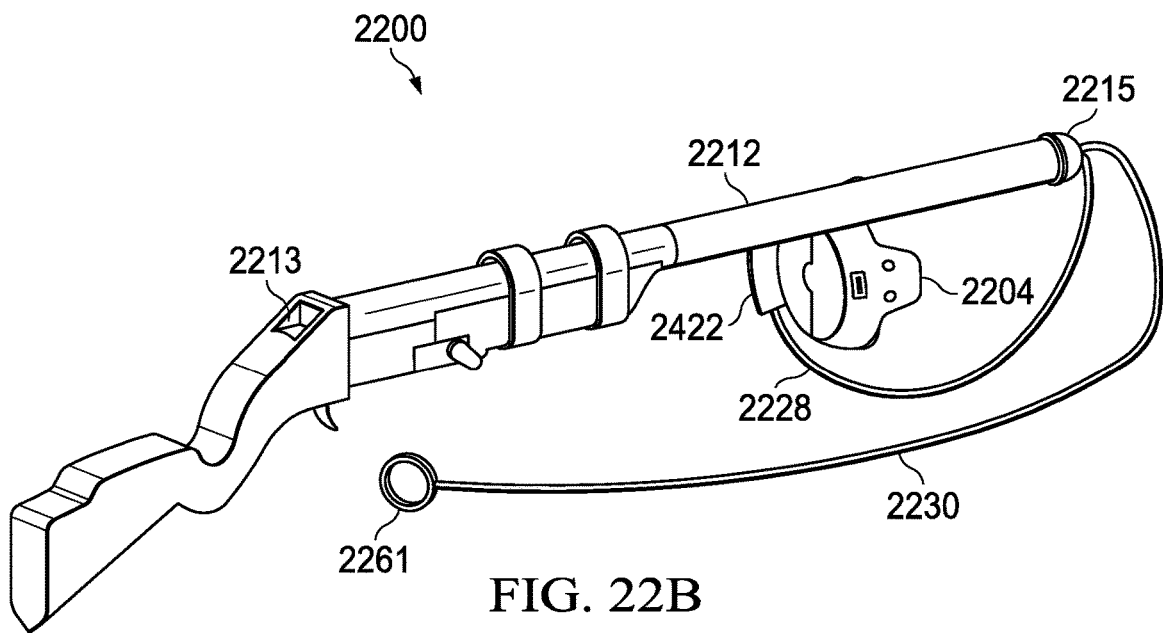
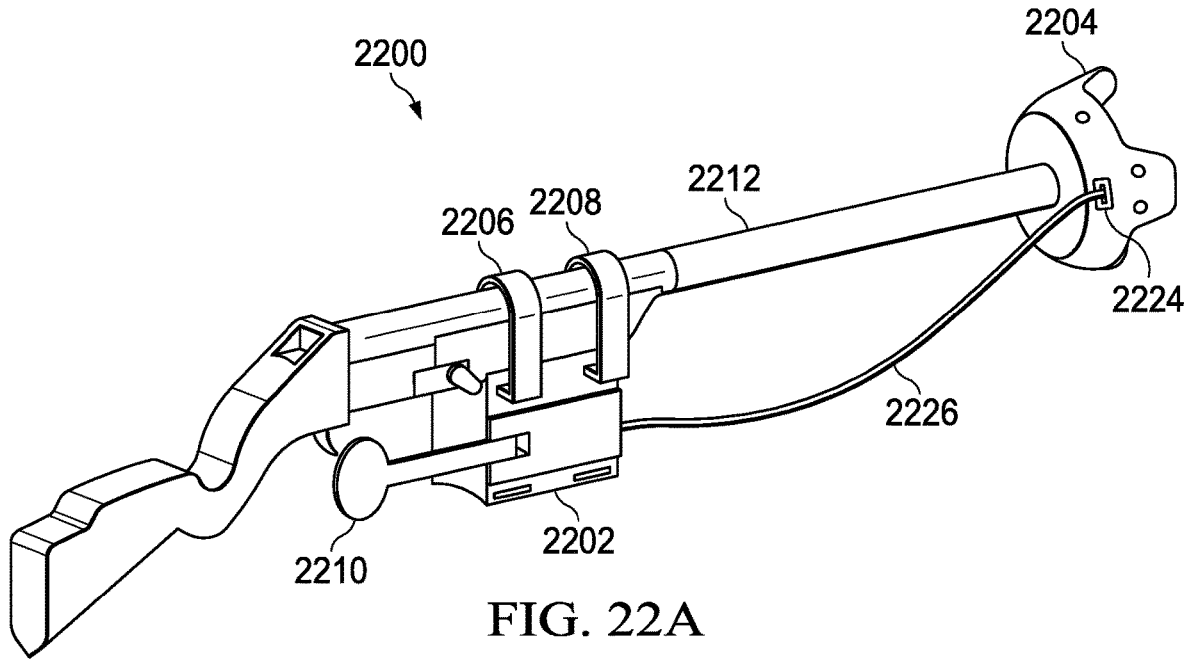
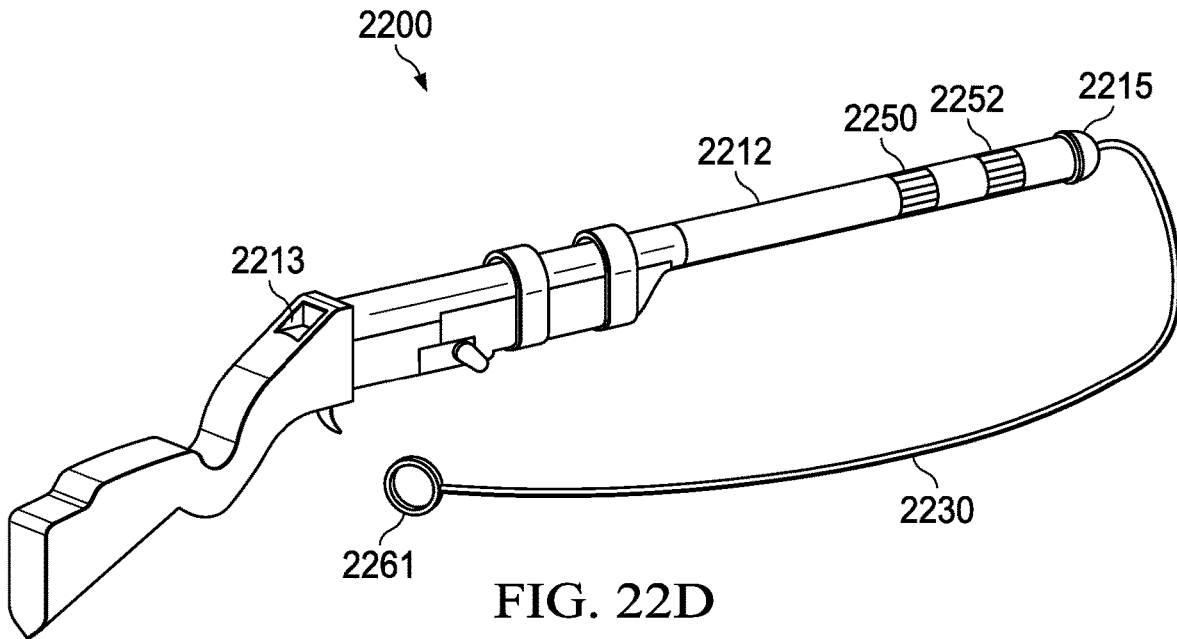
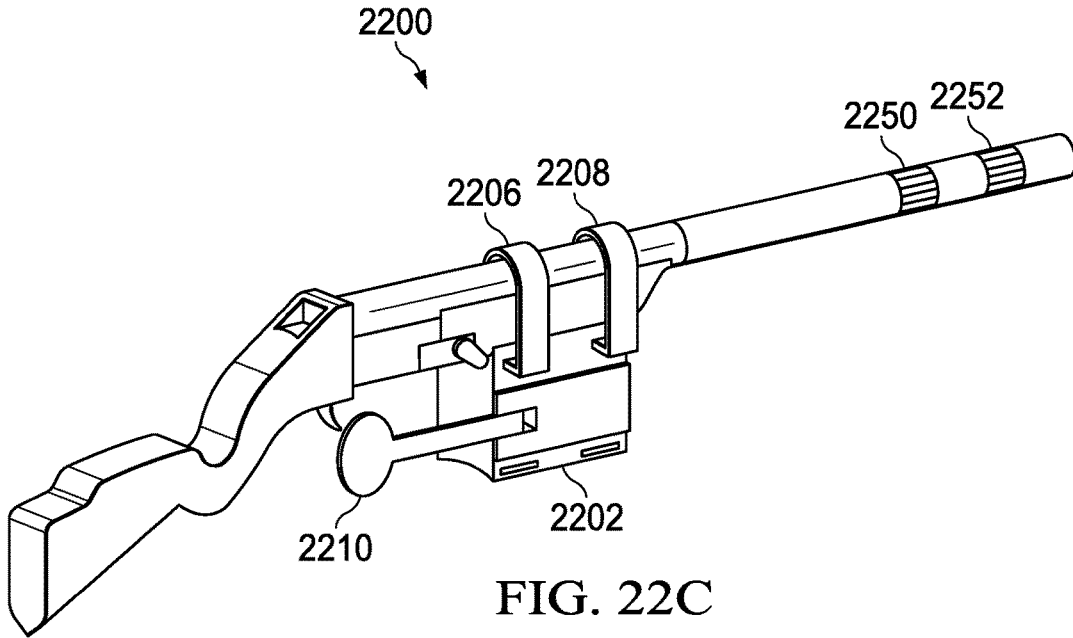


FIG. 21C





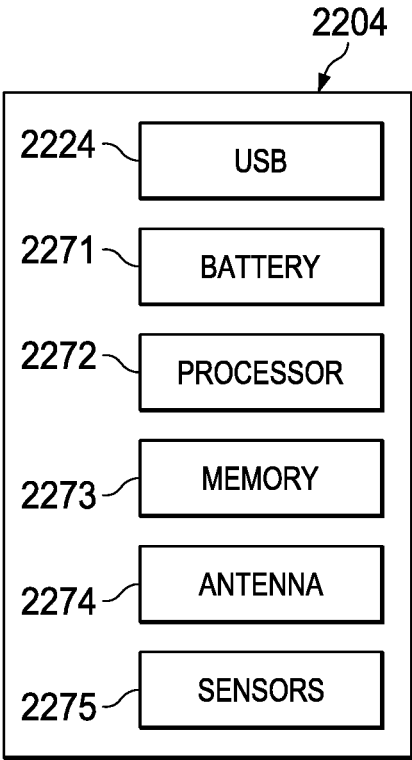


FIG. 22E

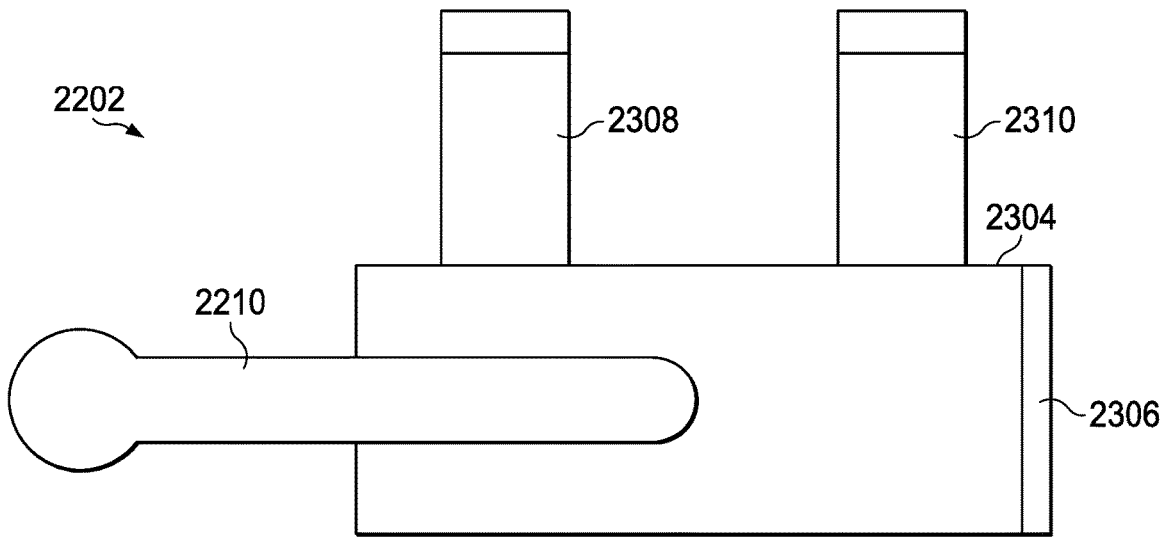


FIG. 23A

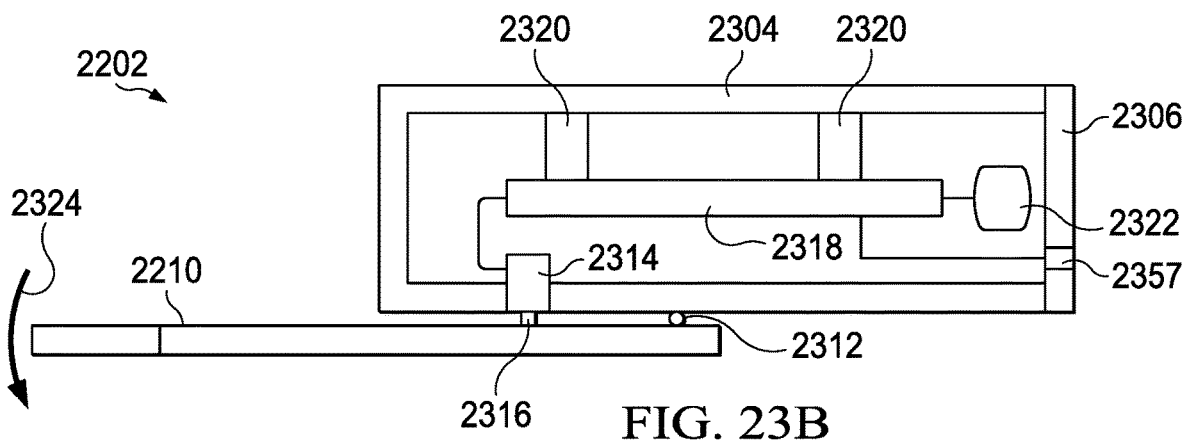


FIG. 23B



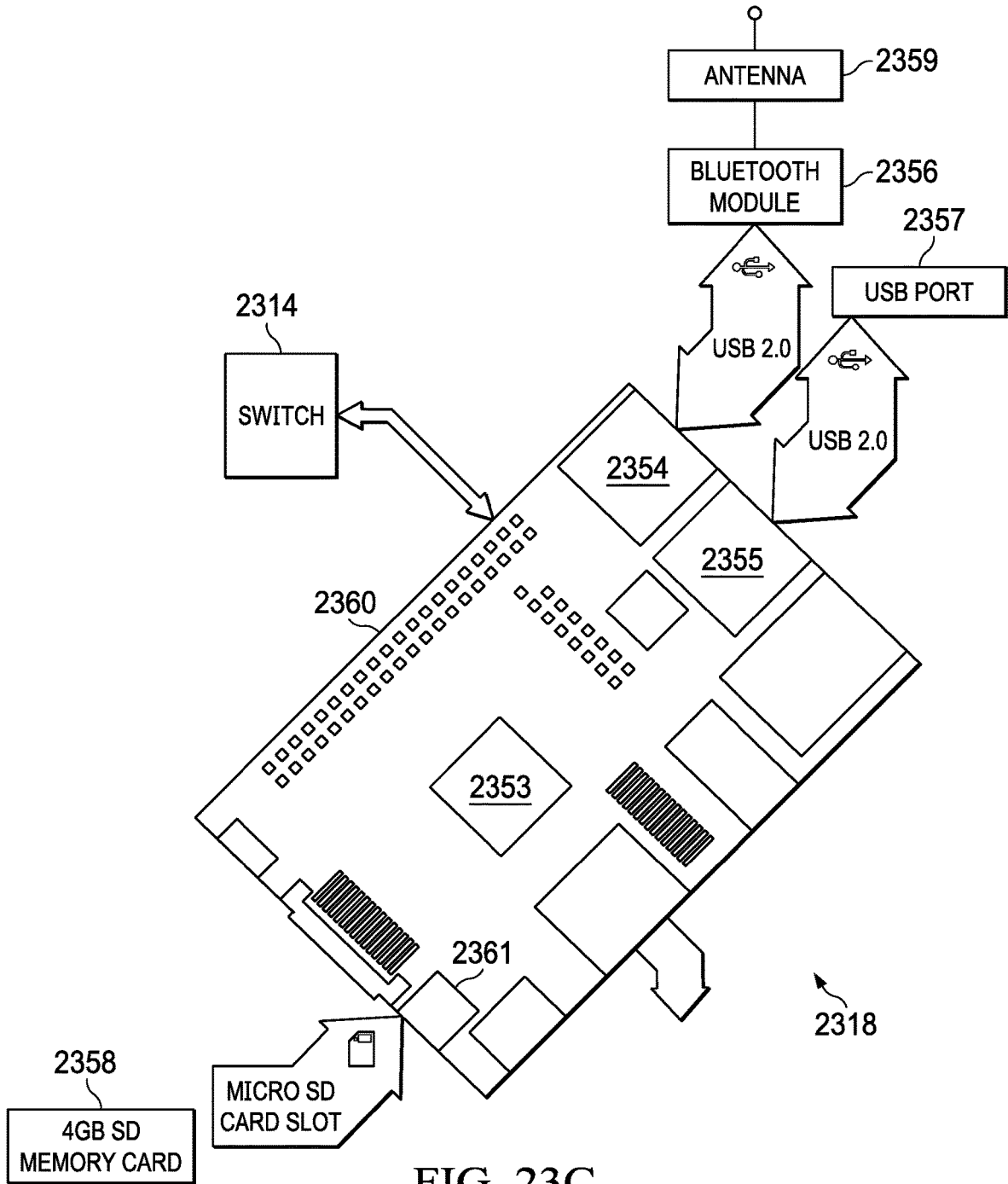


FIG. 23C

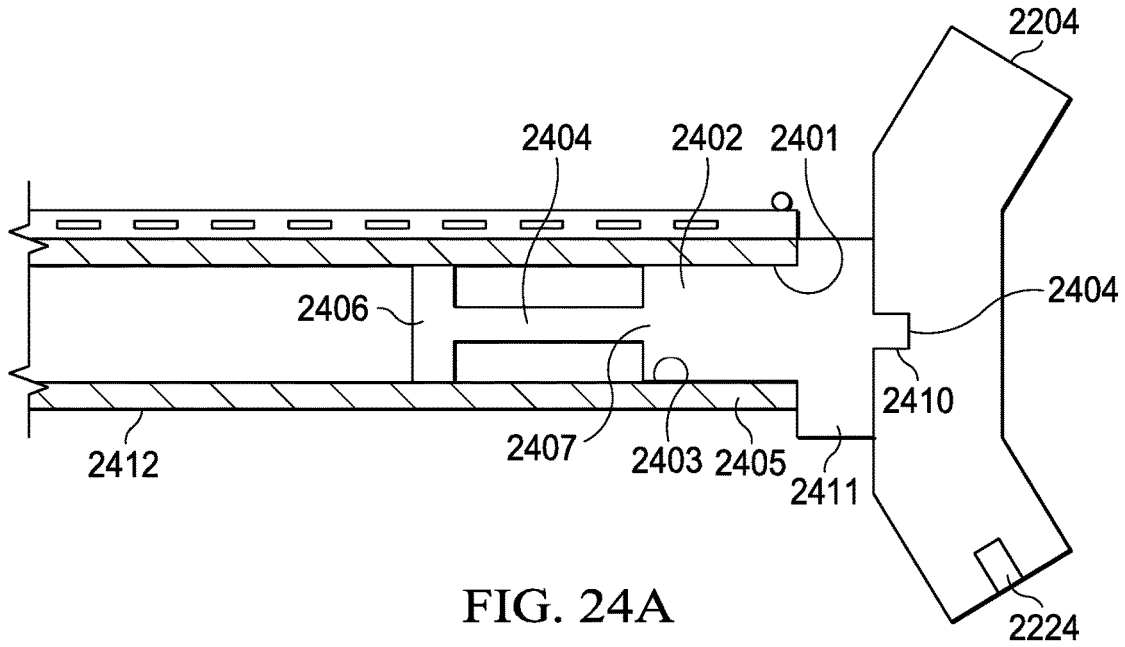


FIG. 24A

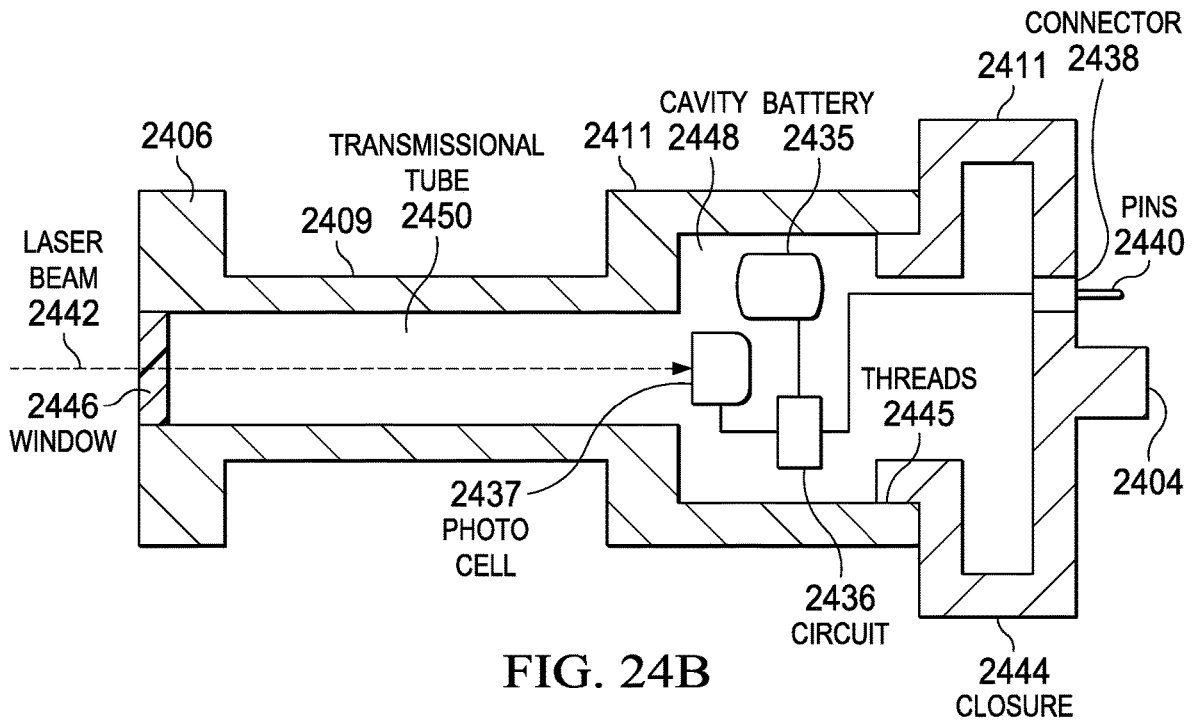


FIG. 24B

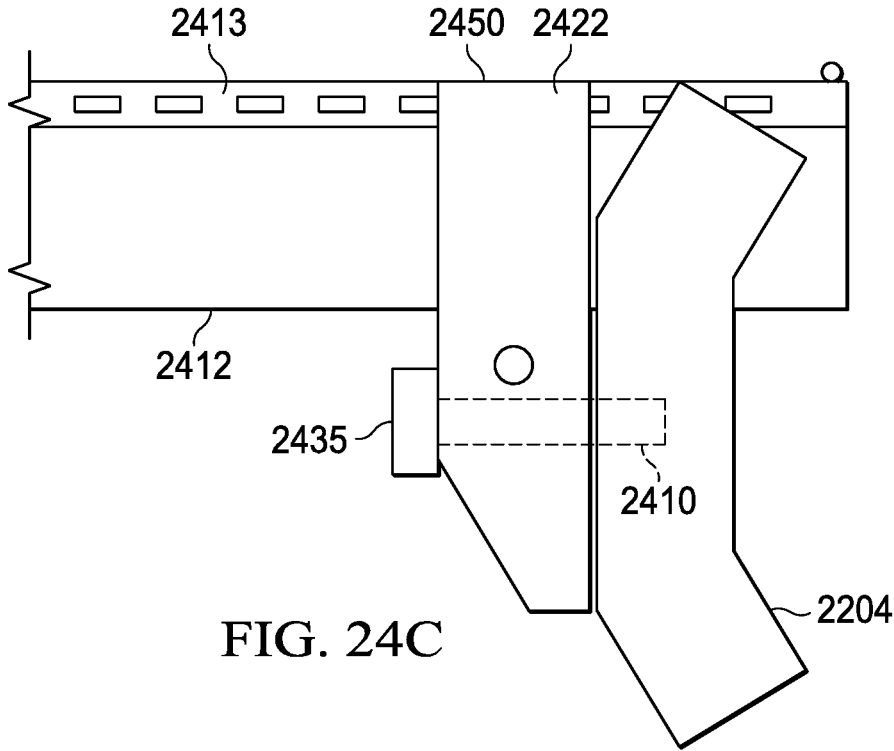


FIG. 24C

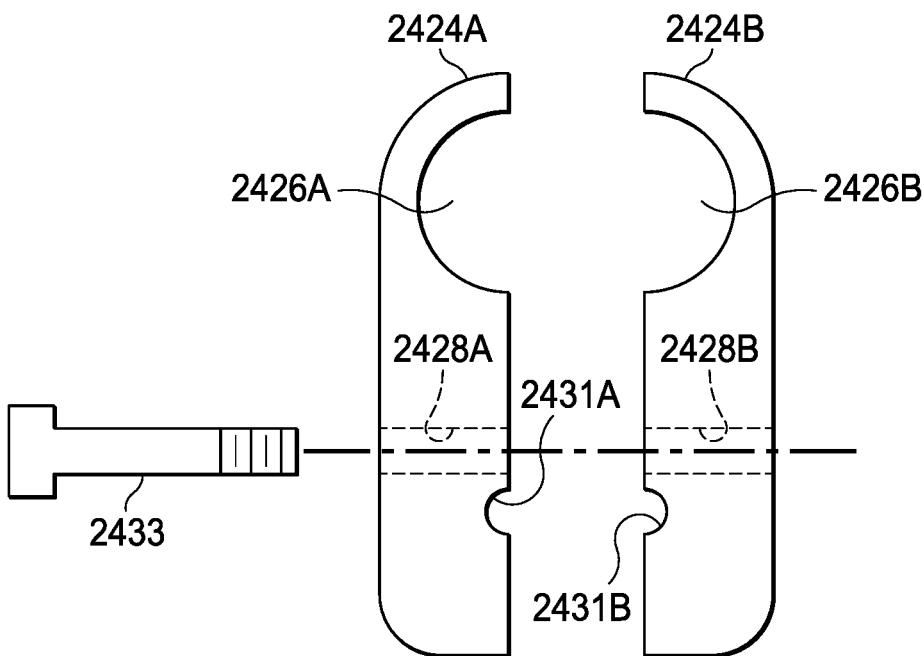
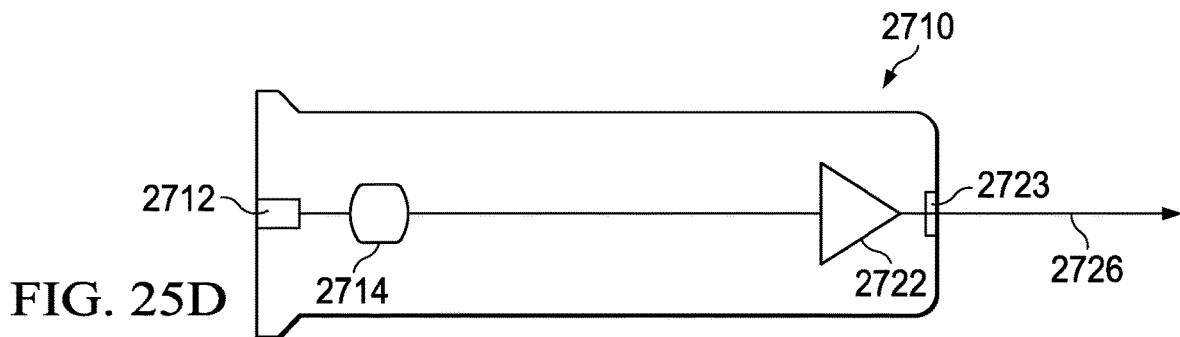
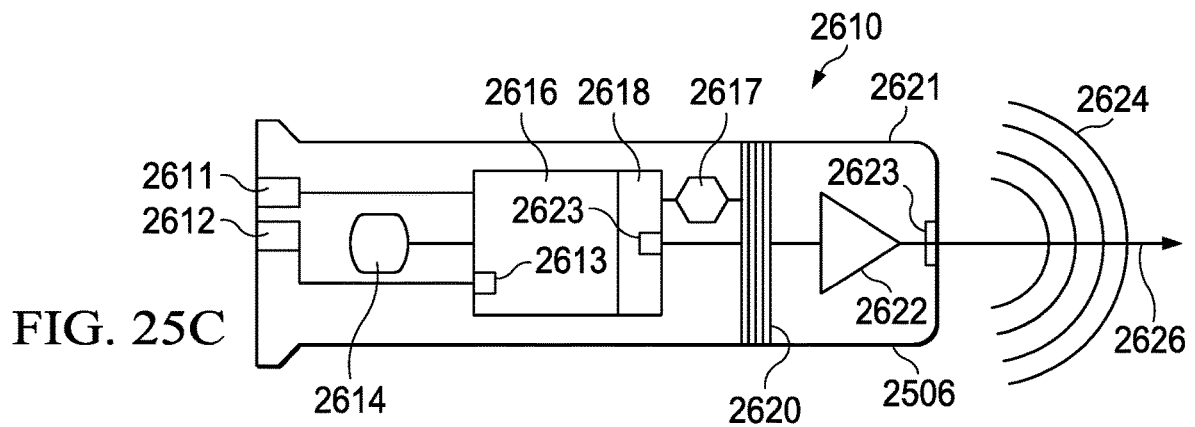
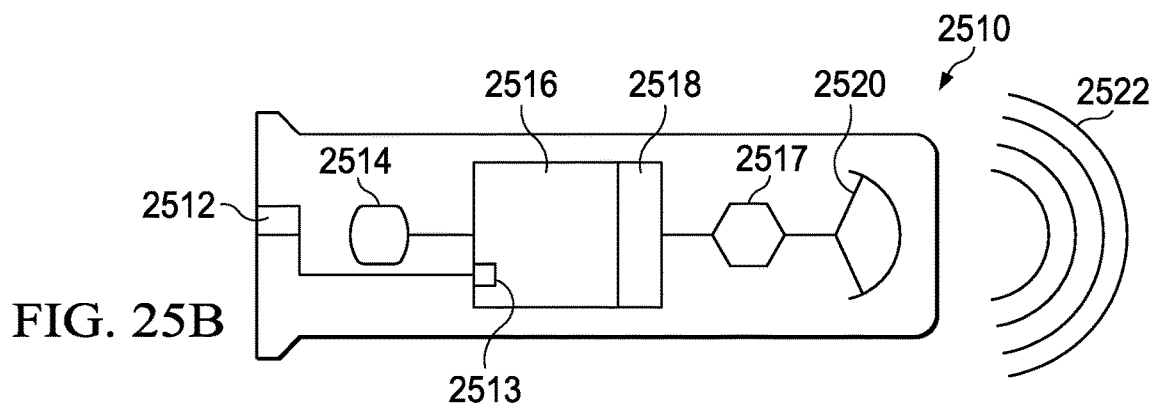
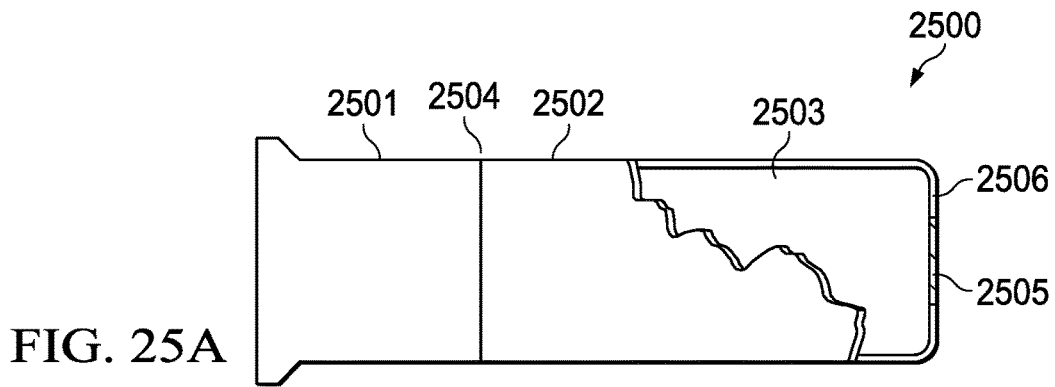
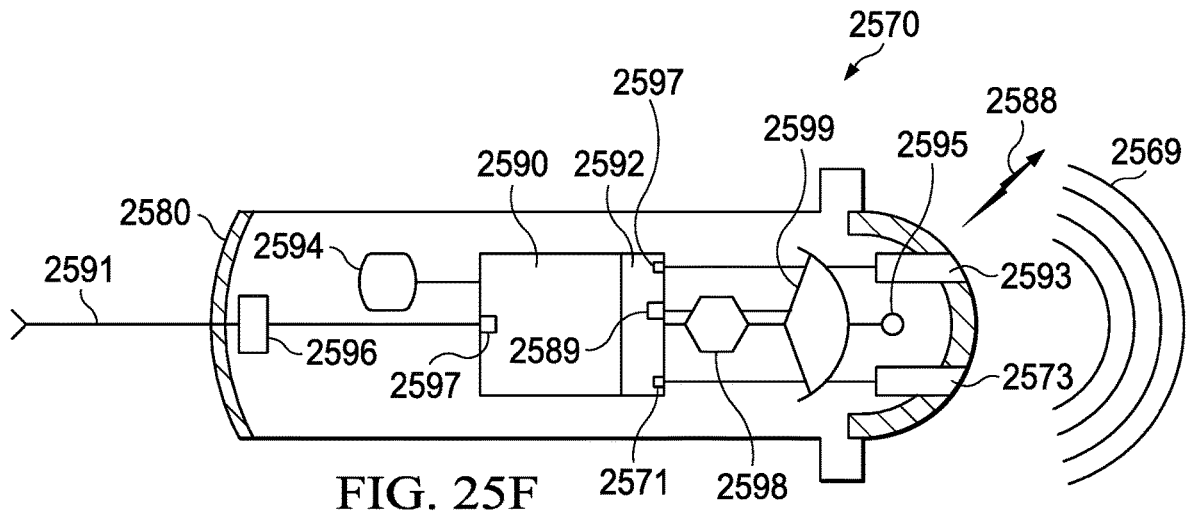
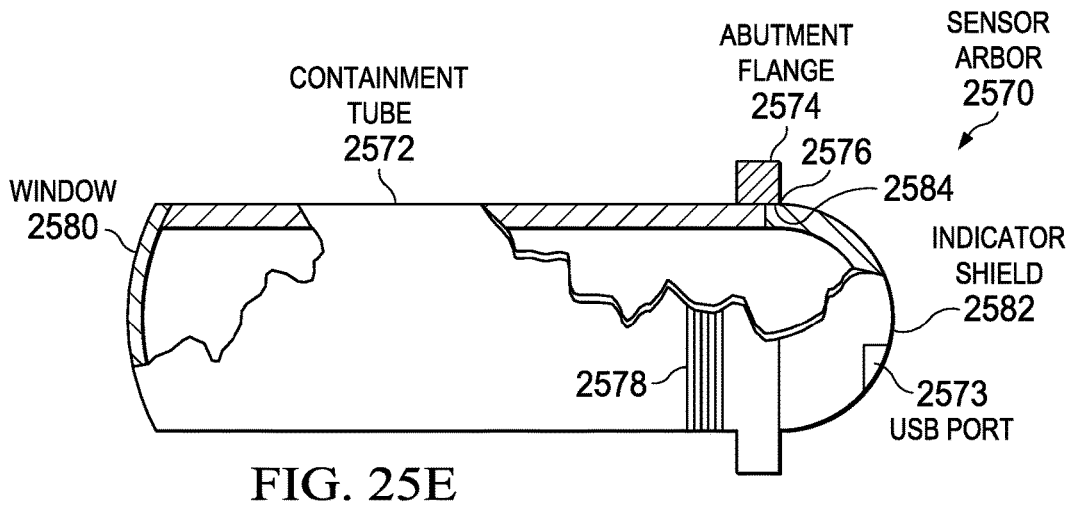


FIG. 24D





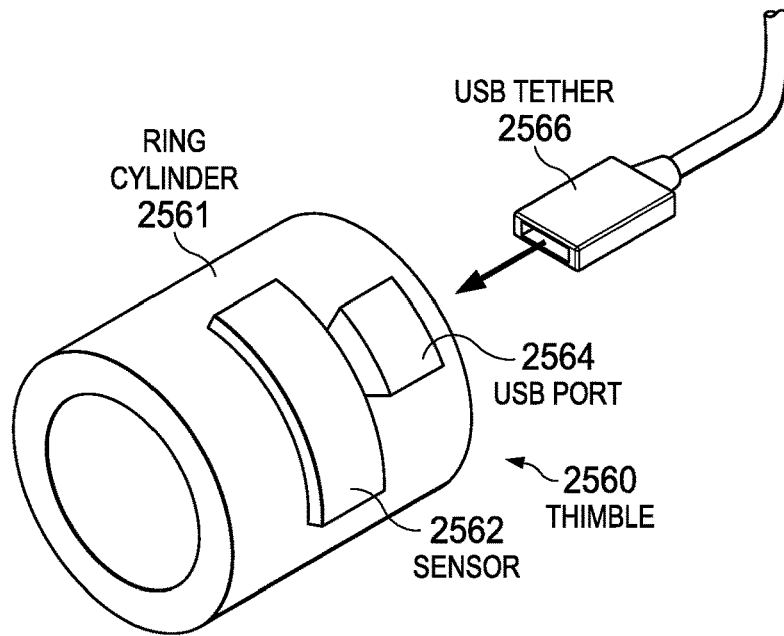


FIG. 25G

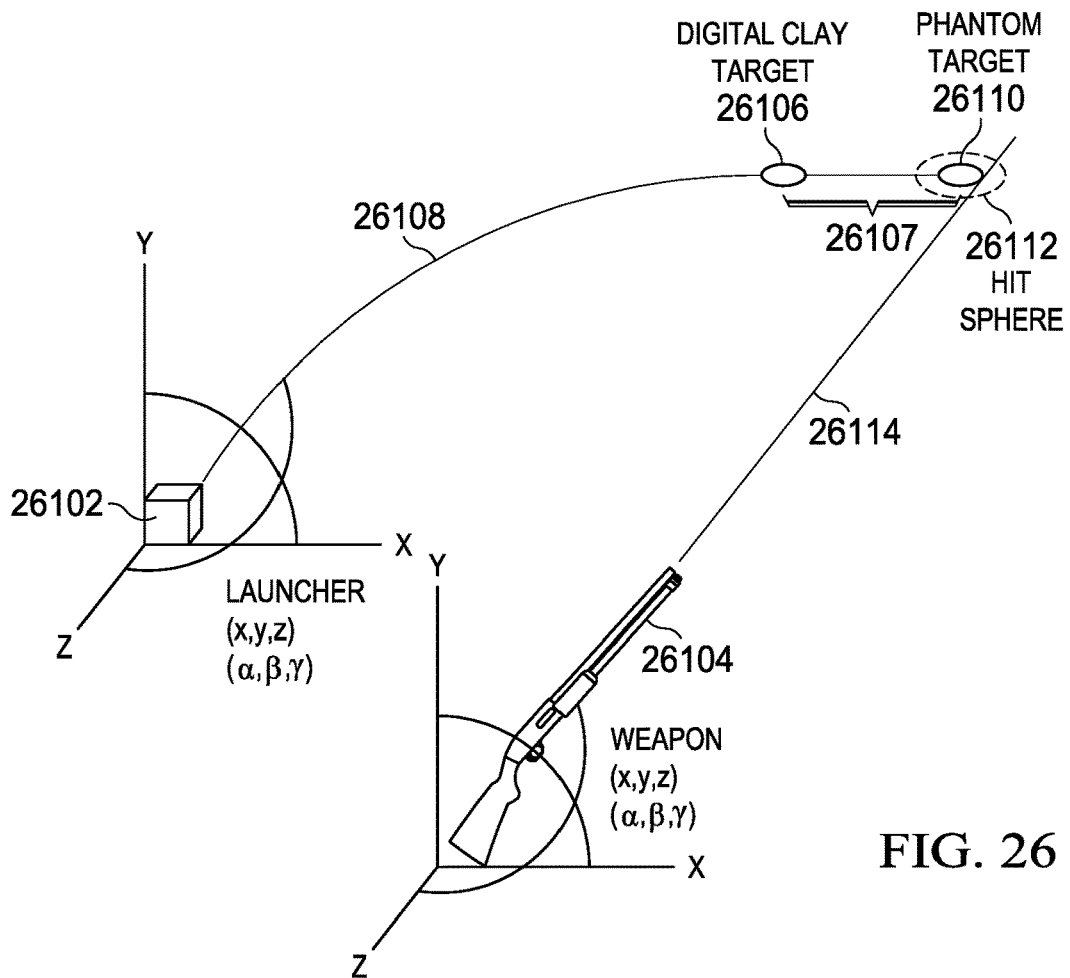


FIG. 26

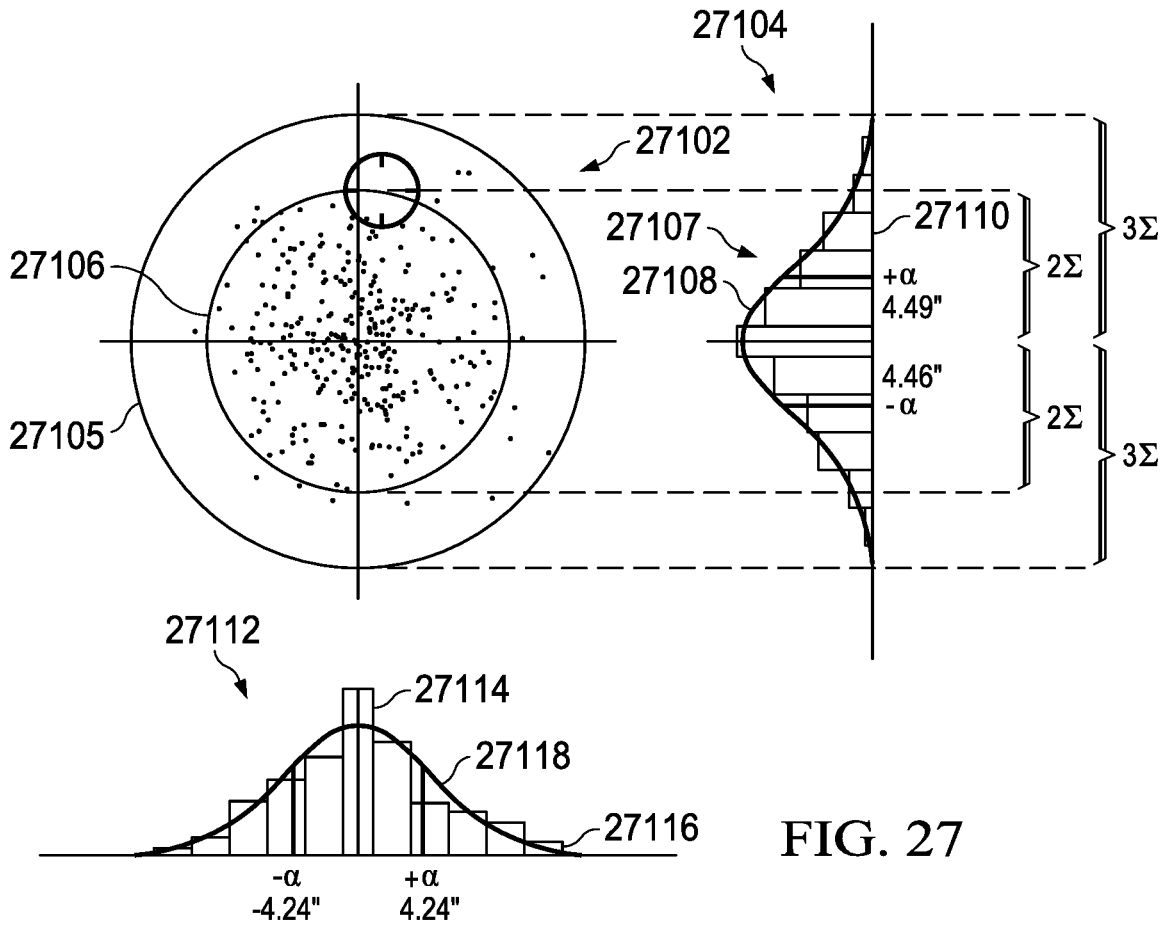


FIG. 27

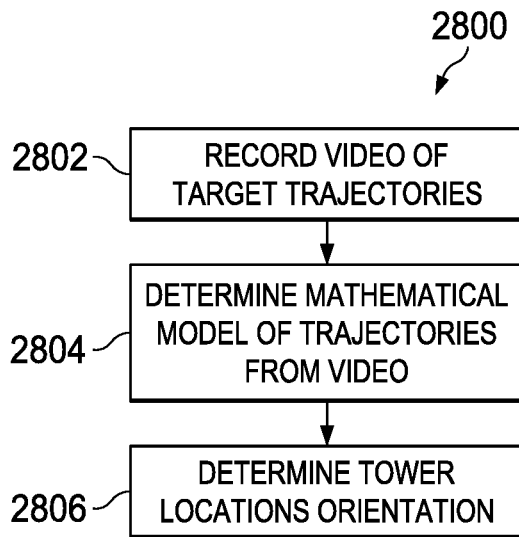


FIG. 28A

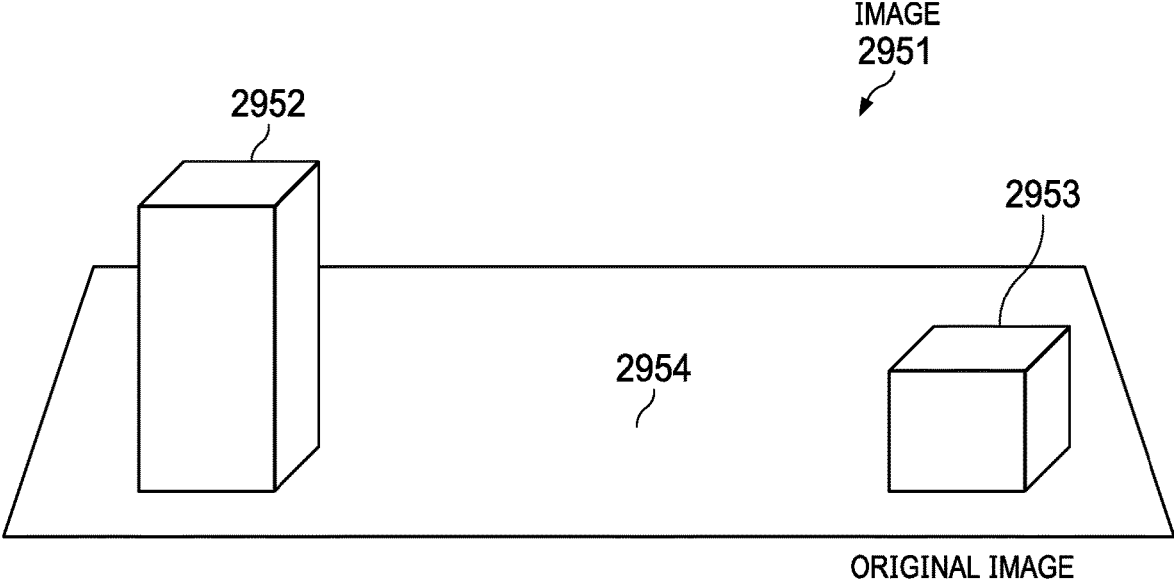


FIG. 28B

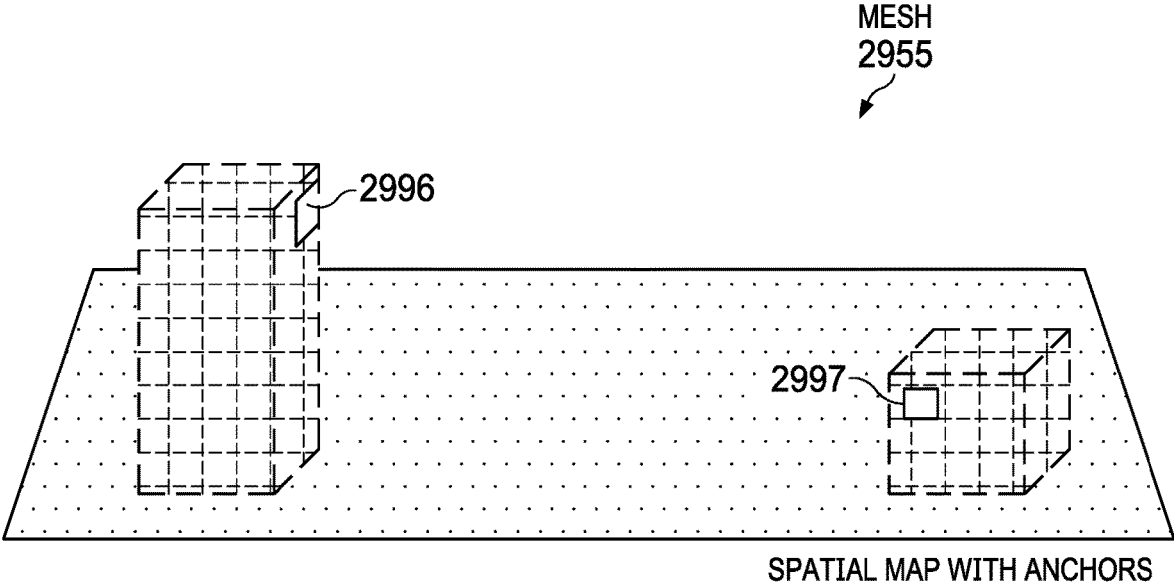


FIG. 28C



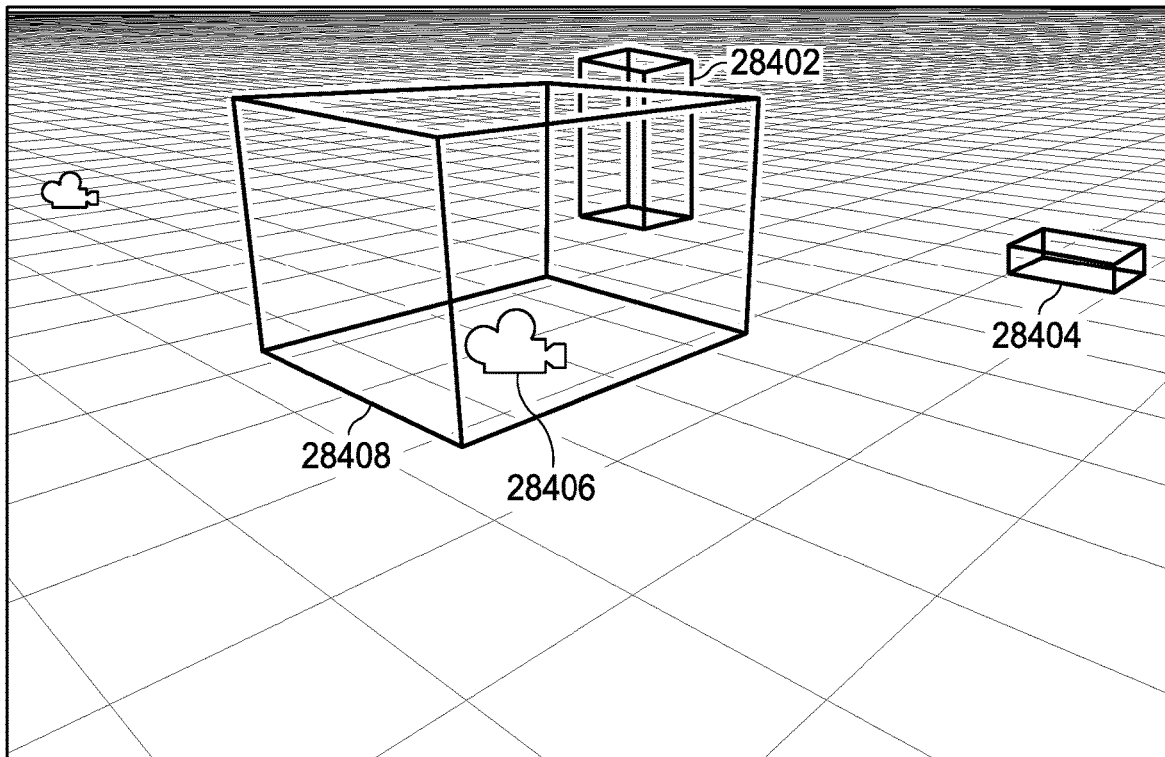


FIG. 28D

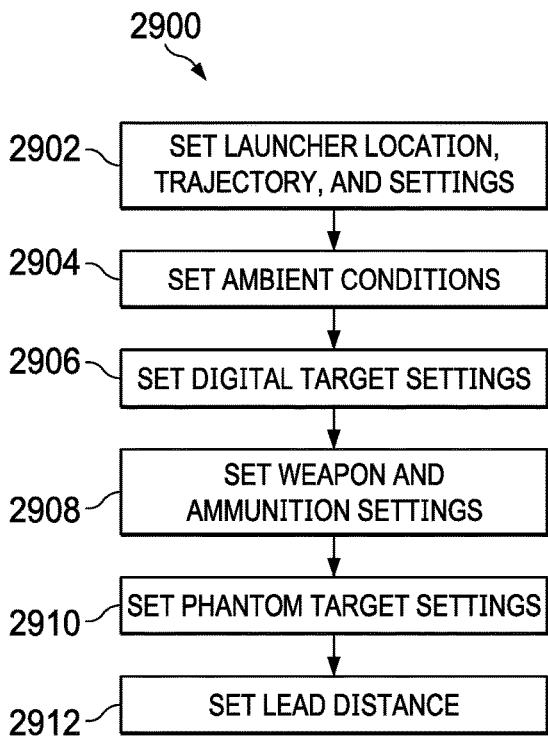


FIG. 29A

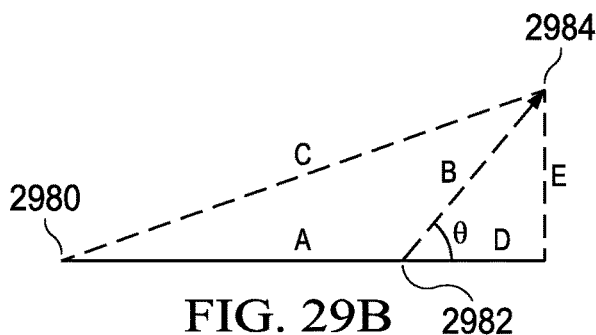


FIG. 29B

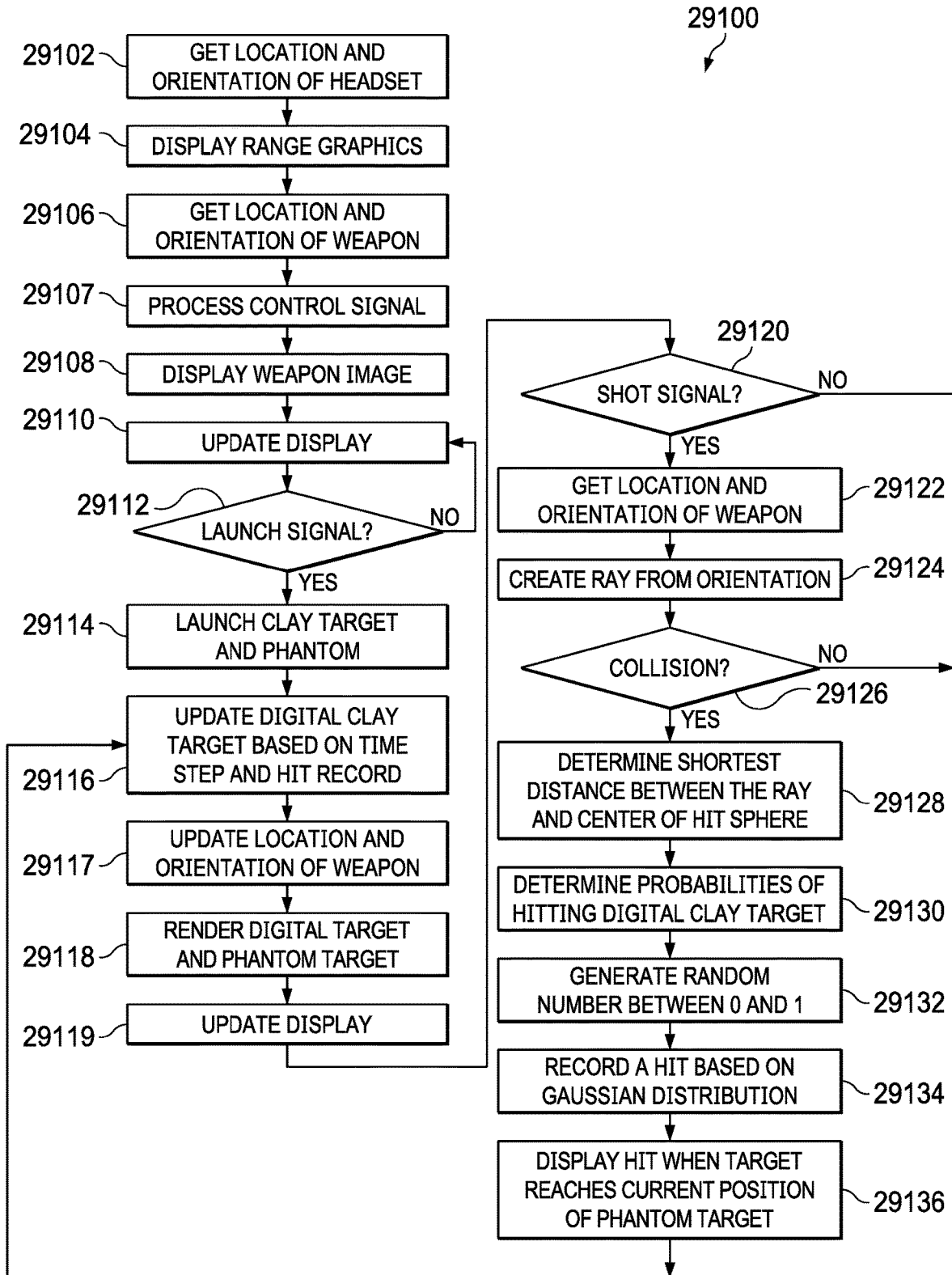


FIG. 29C

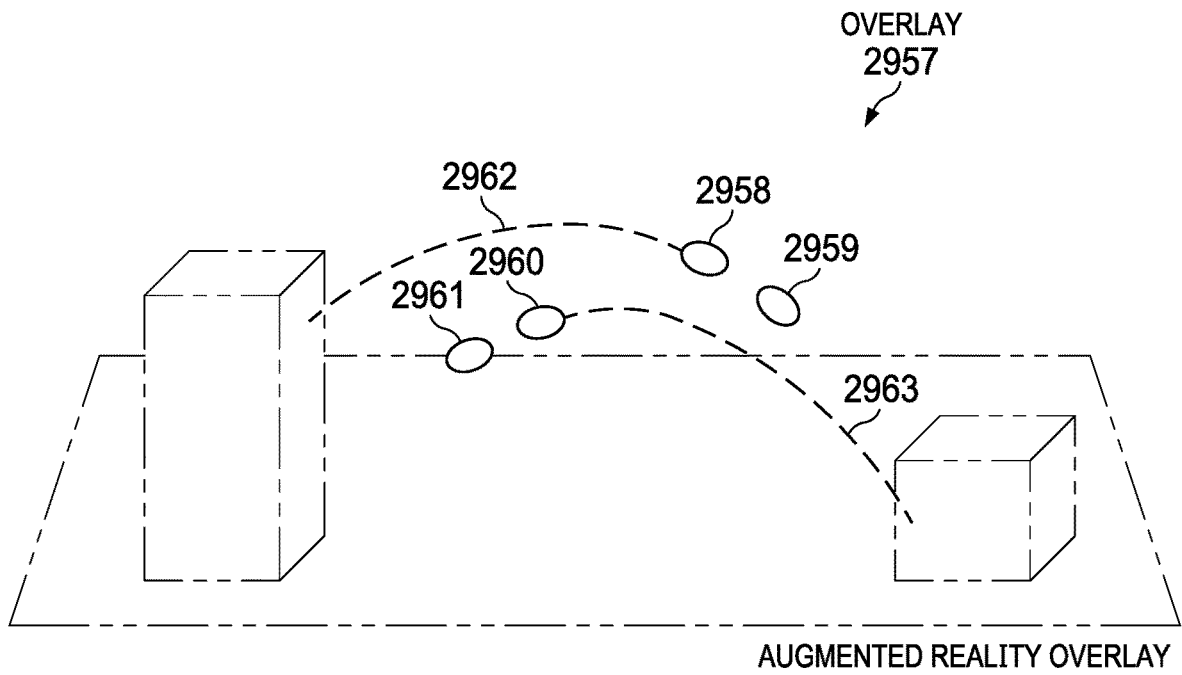


FIG. 29D

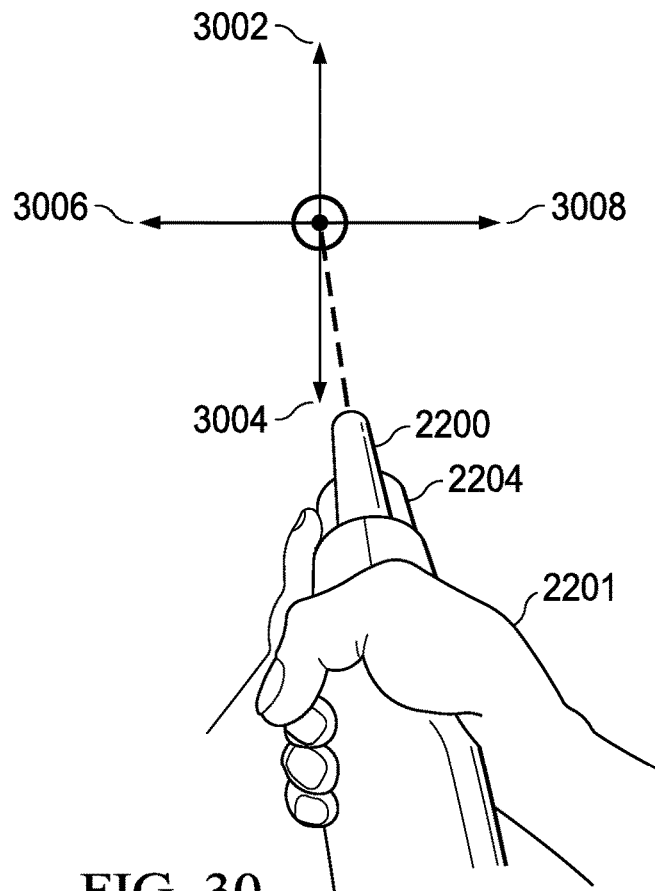


FIG. 30

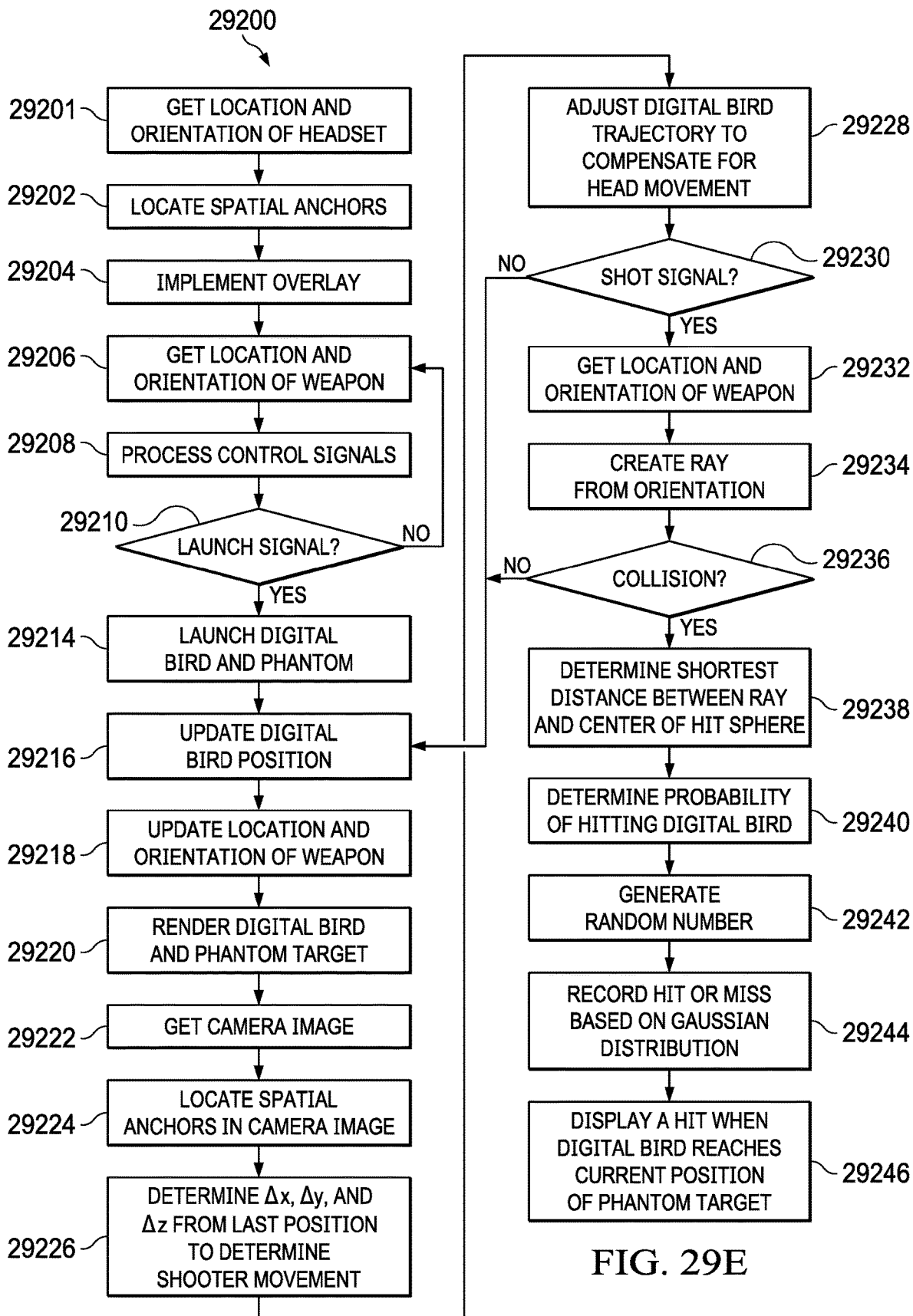


FIG. 29E

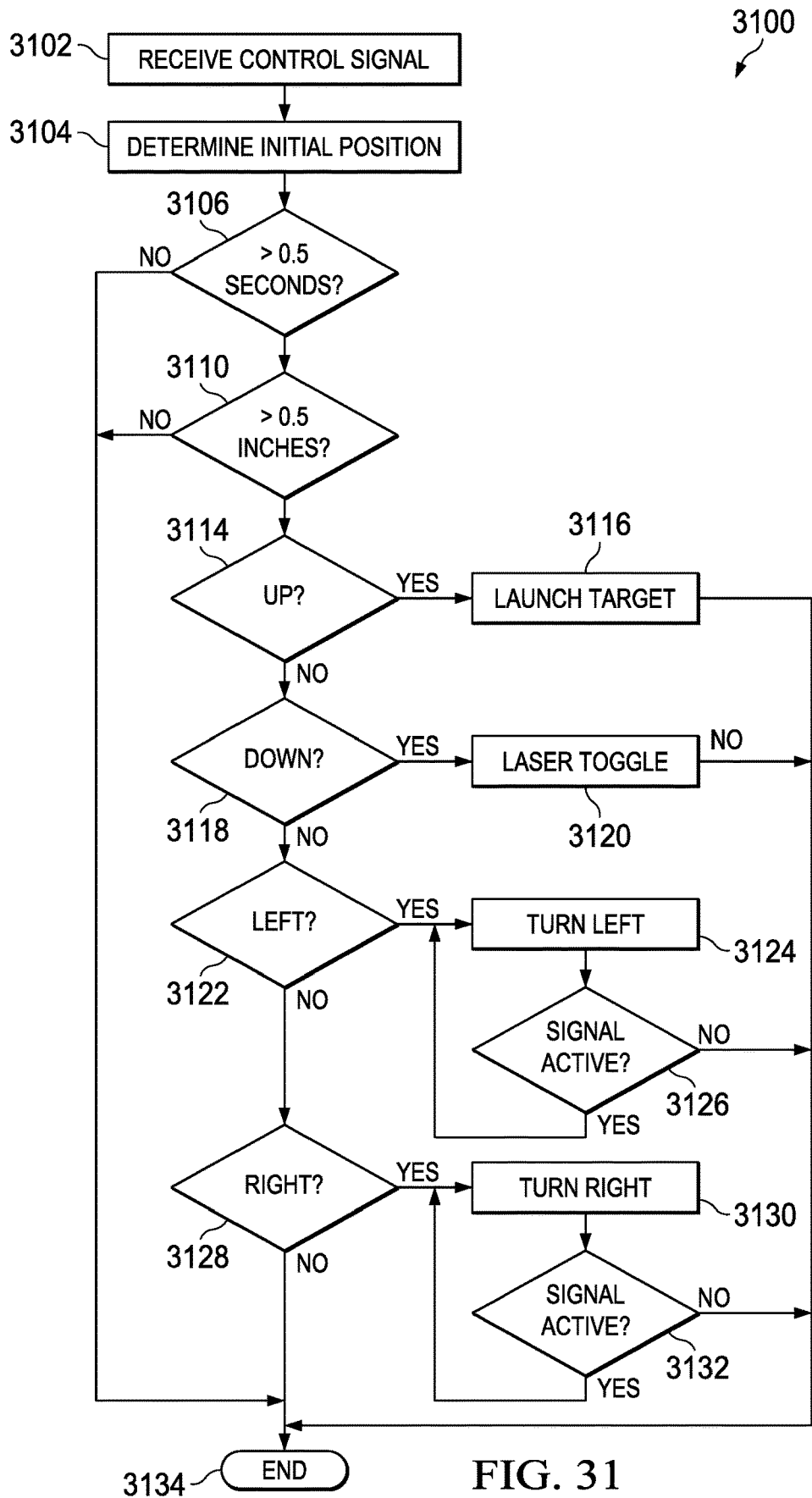


FIG. 31

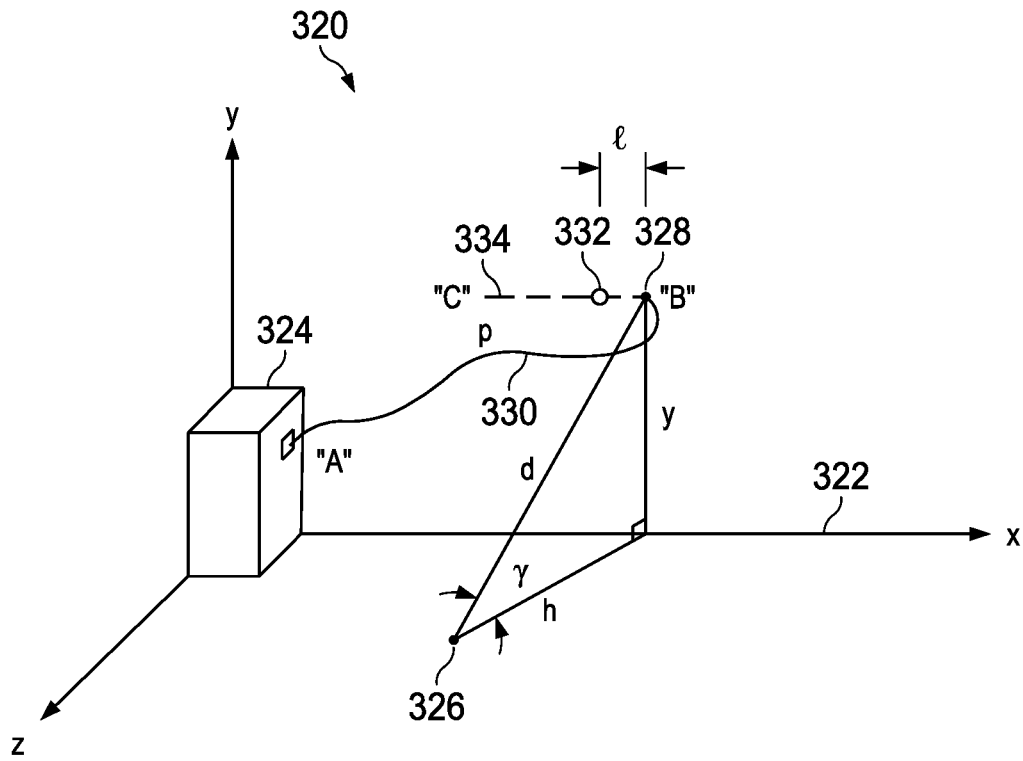


FIG. 32

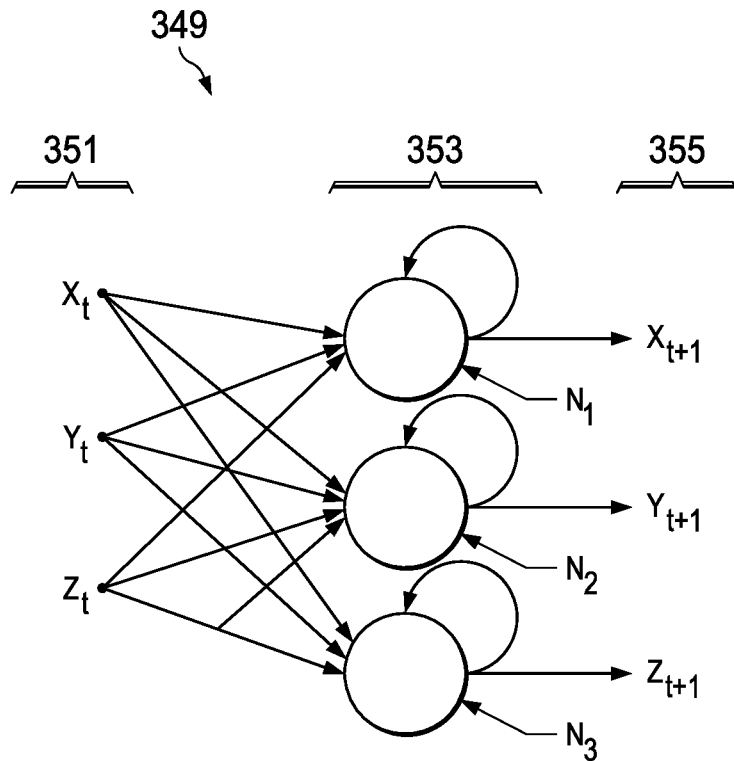


FIG. 35

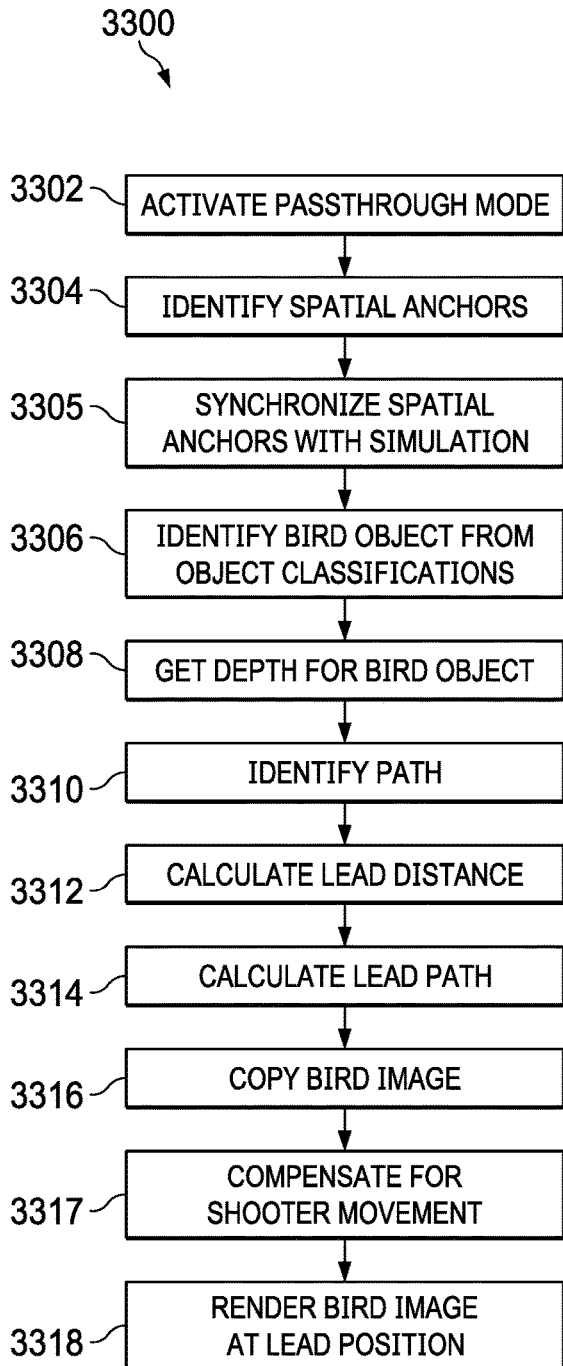


FIG. 33

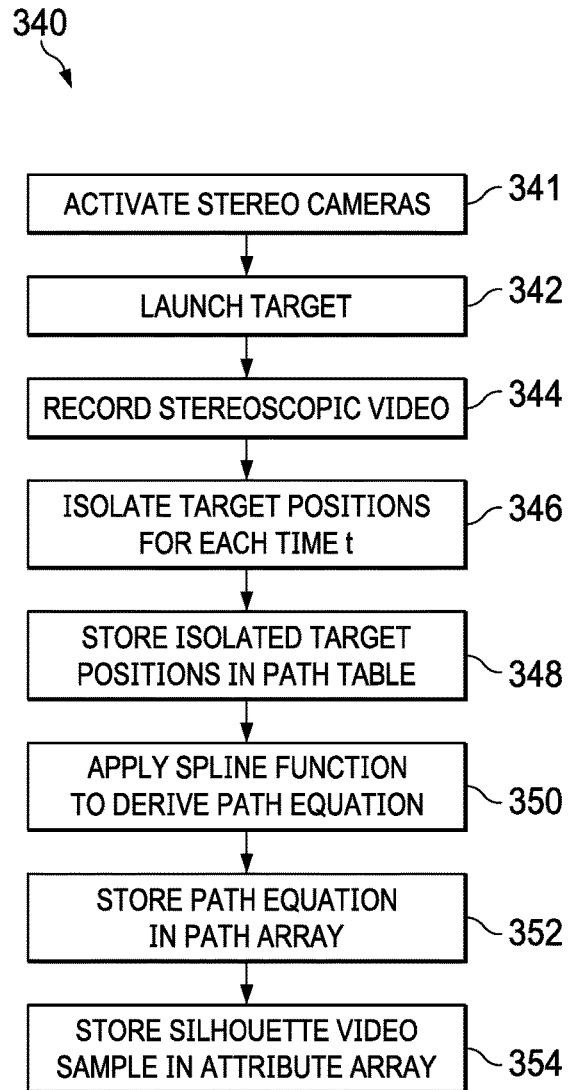


FIG. 34

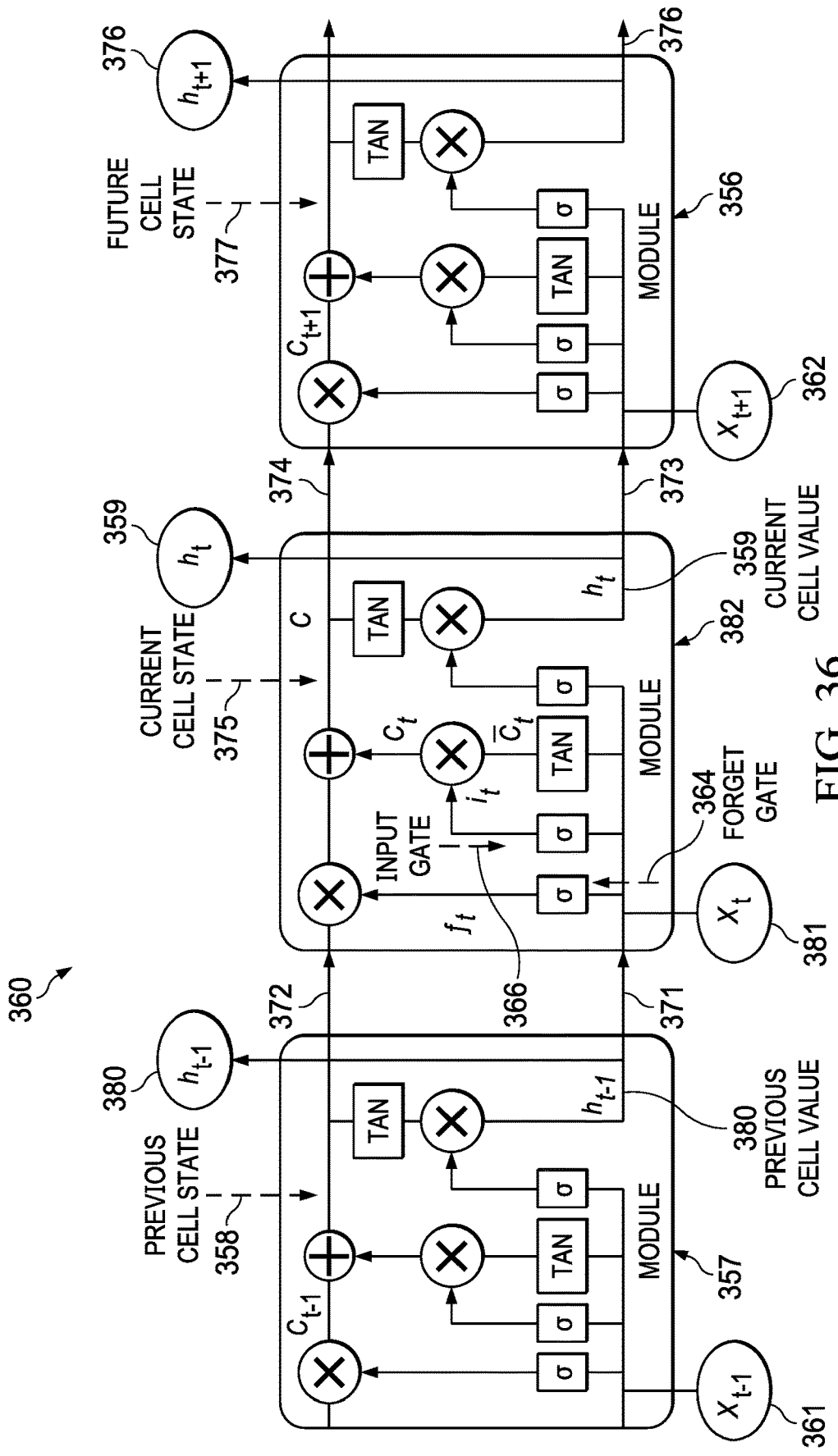


FIG. 36



3710 ↘

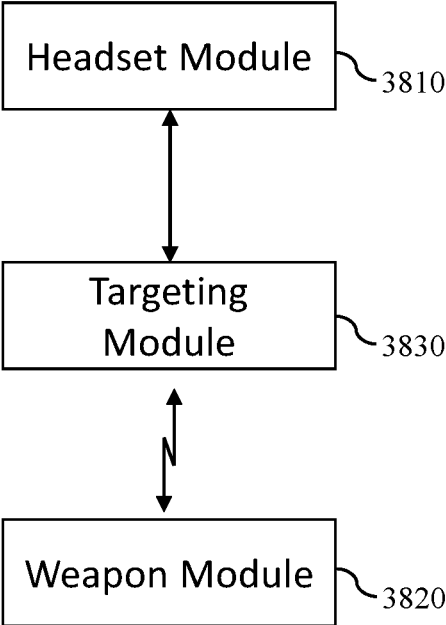


Figure 37A

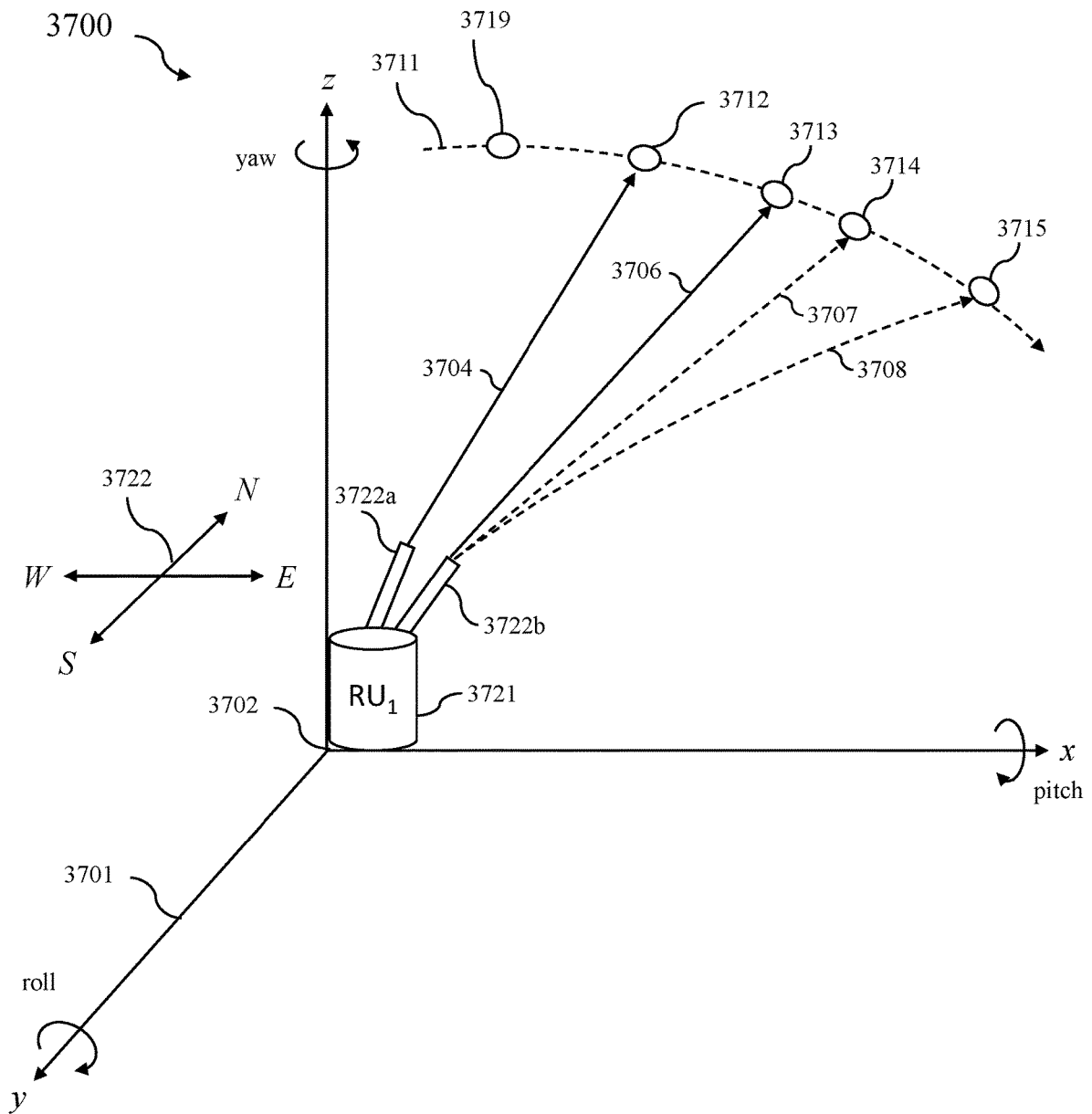


Figure 37B

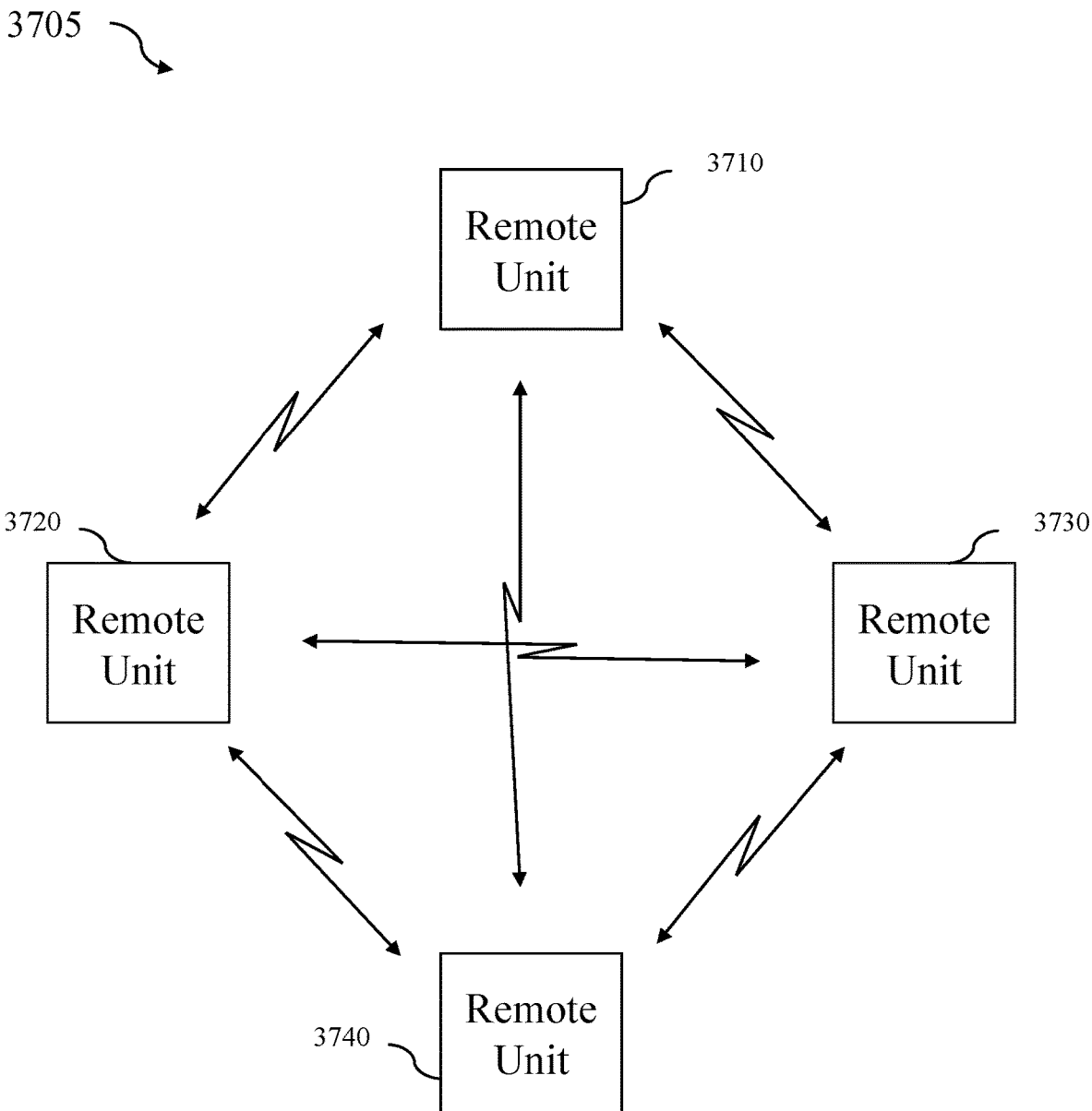


Figure 38A

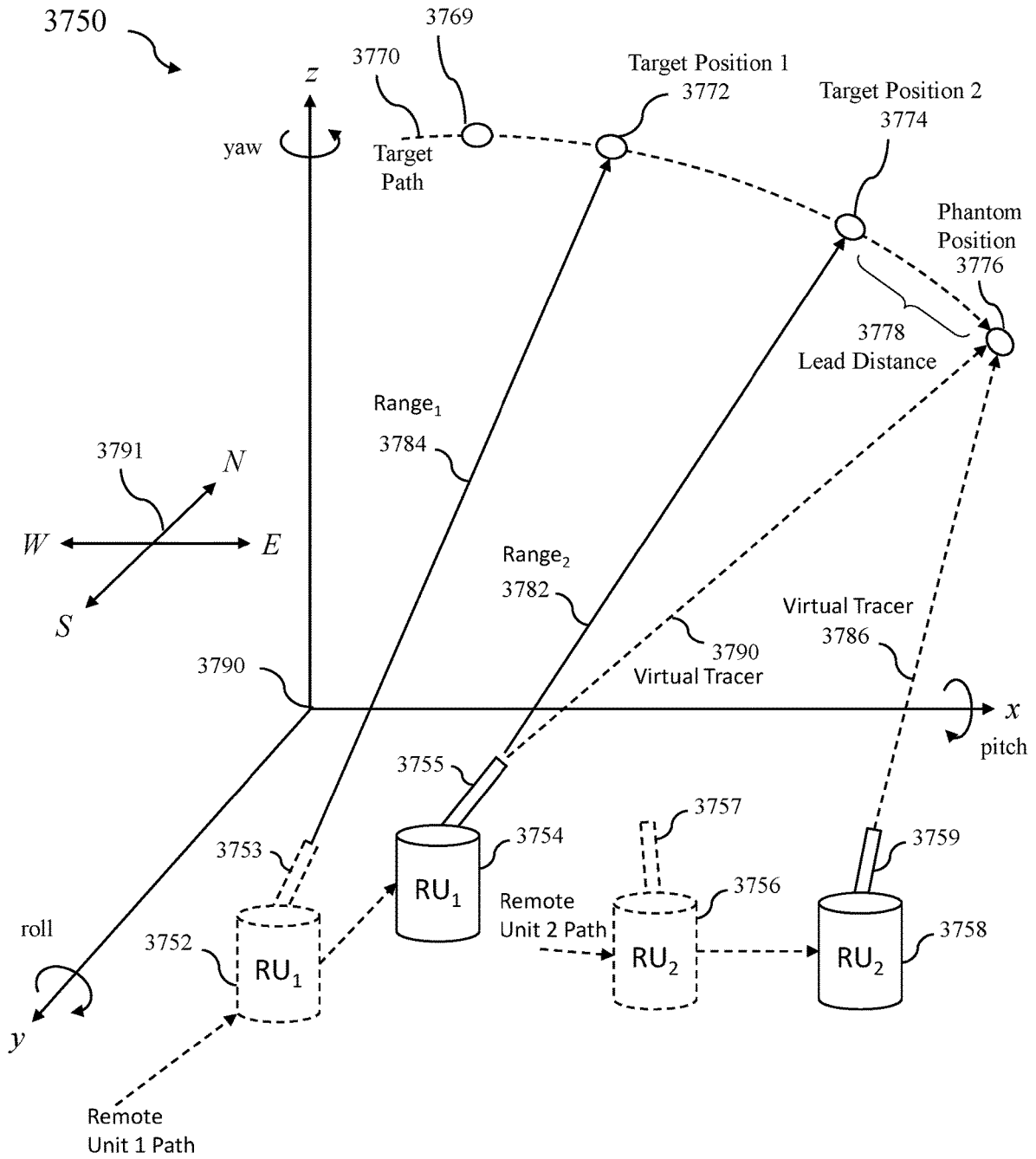


Figure 38B

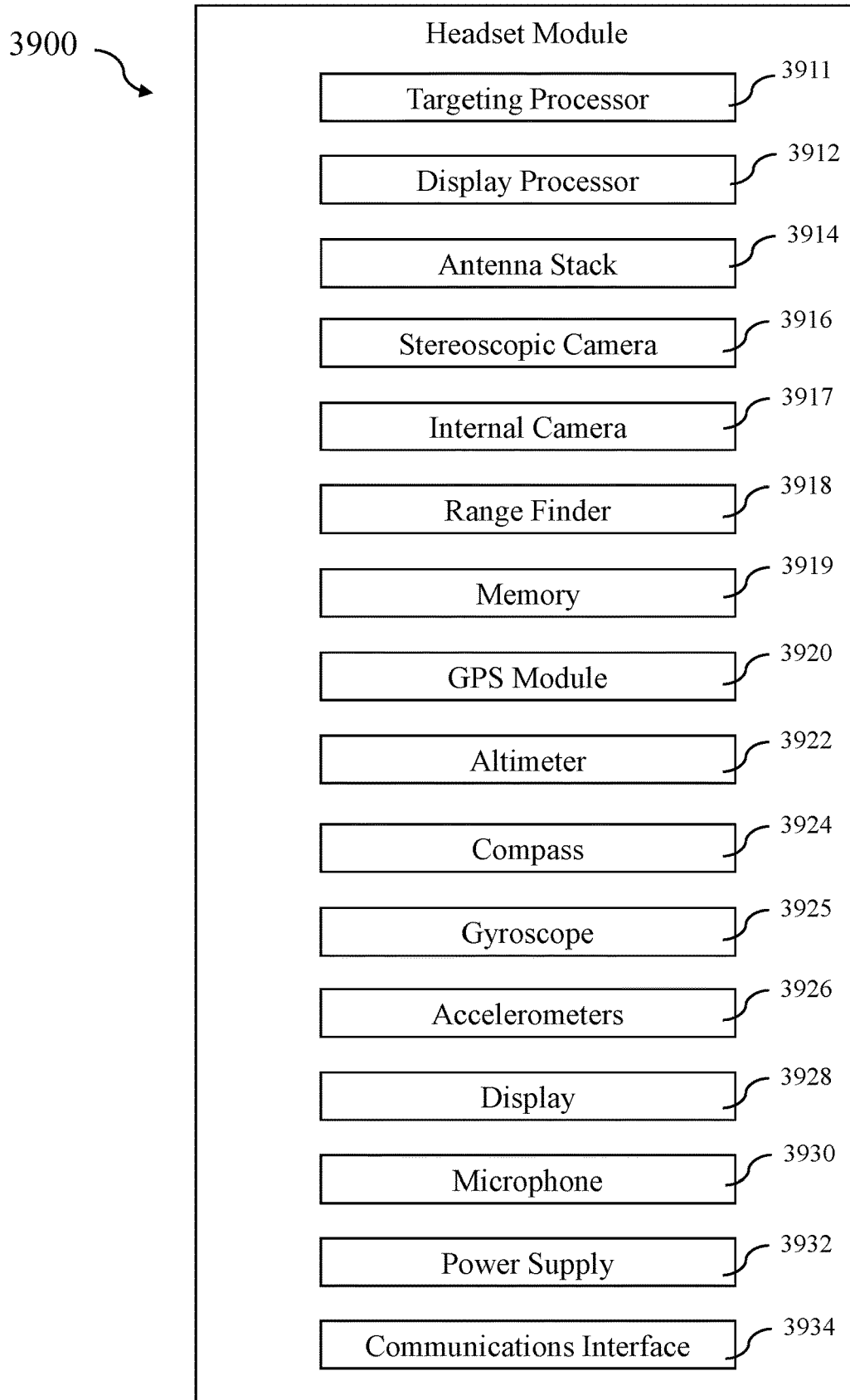


Figure 39

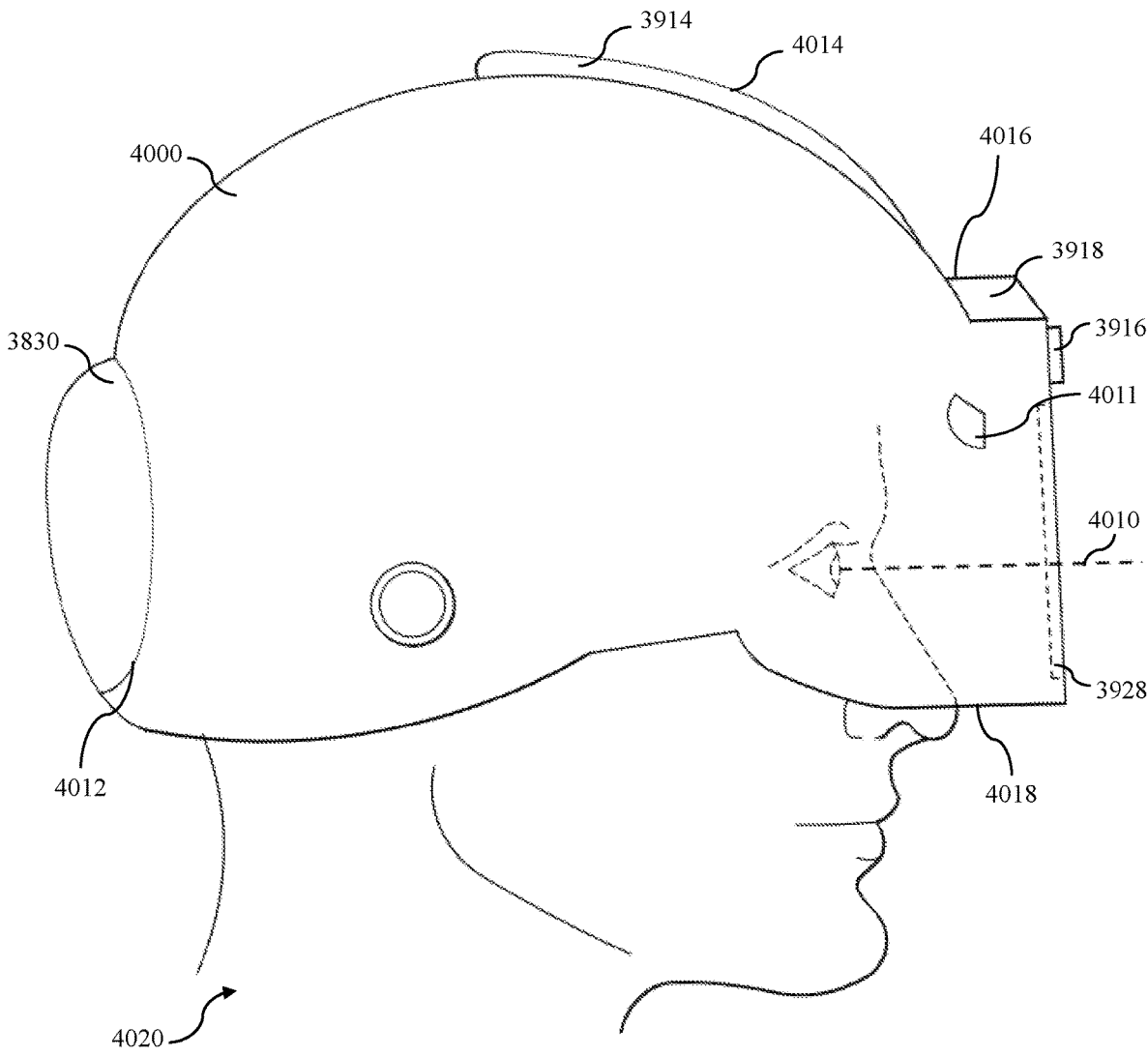


Figure 40A

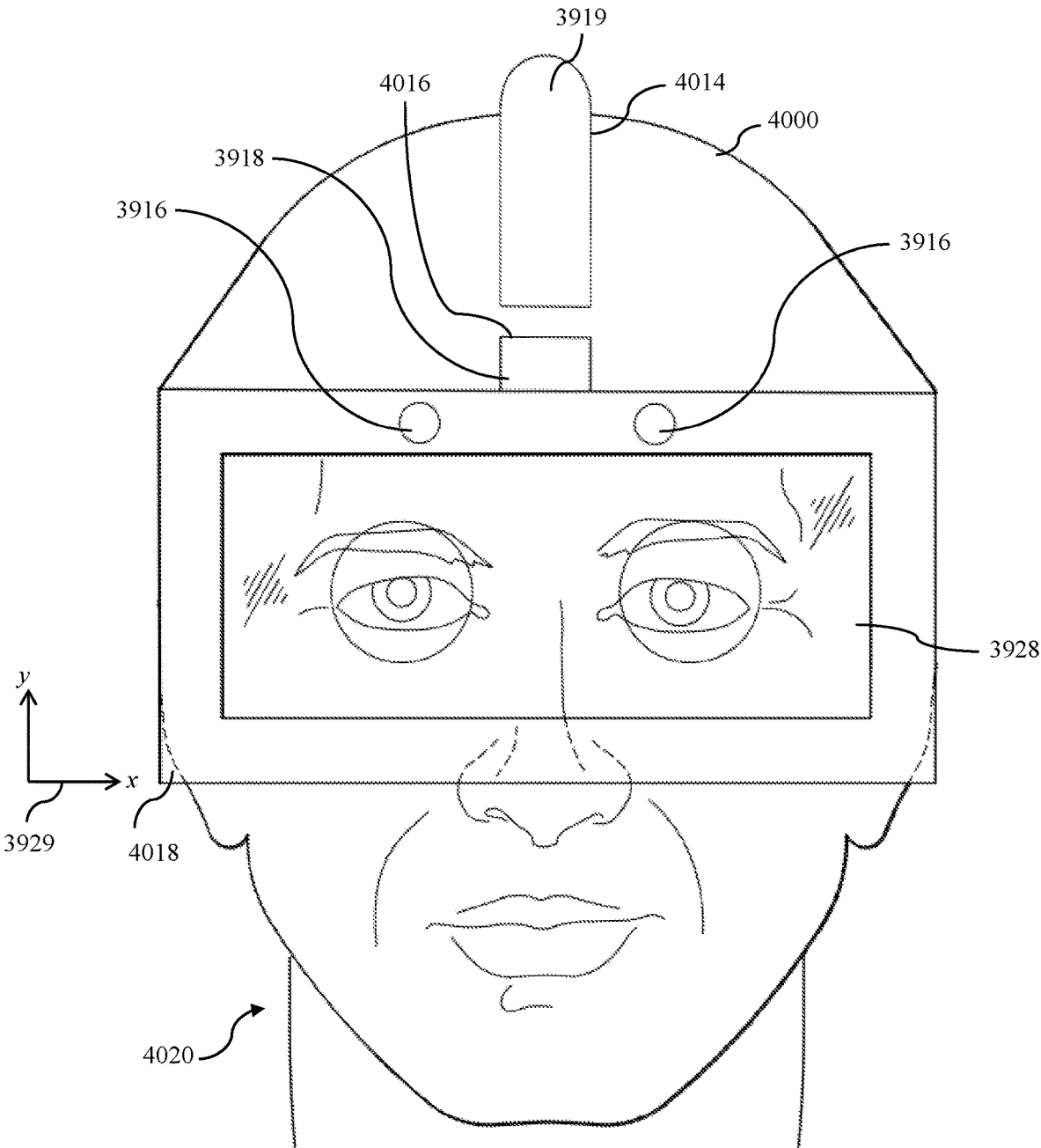



Figure 40B

3820 

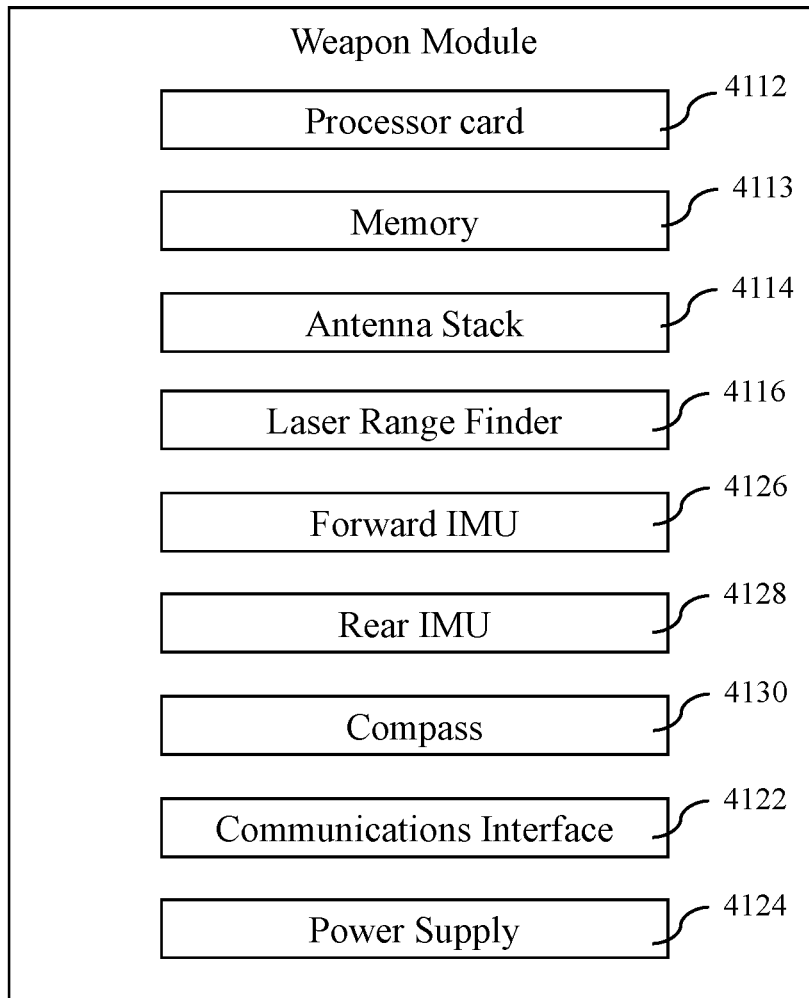


Figure 41A



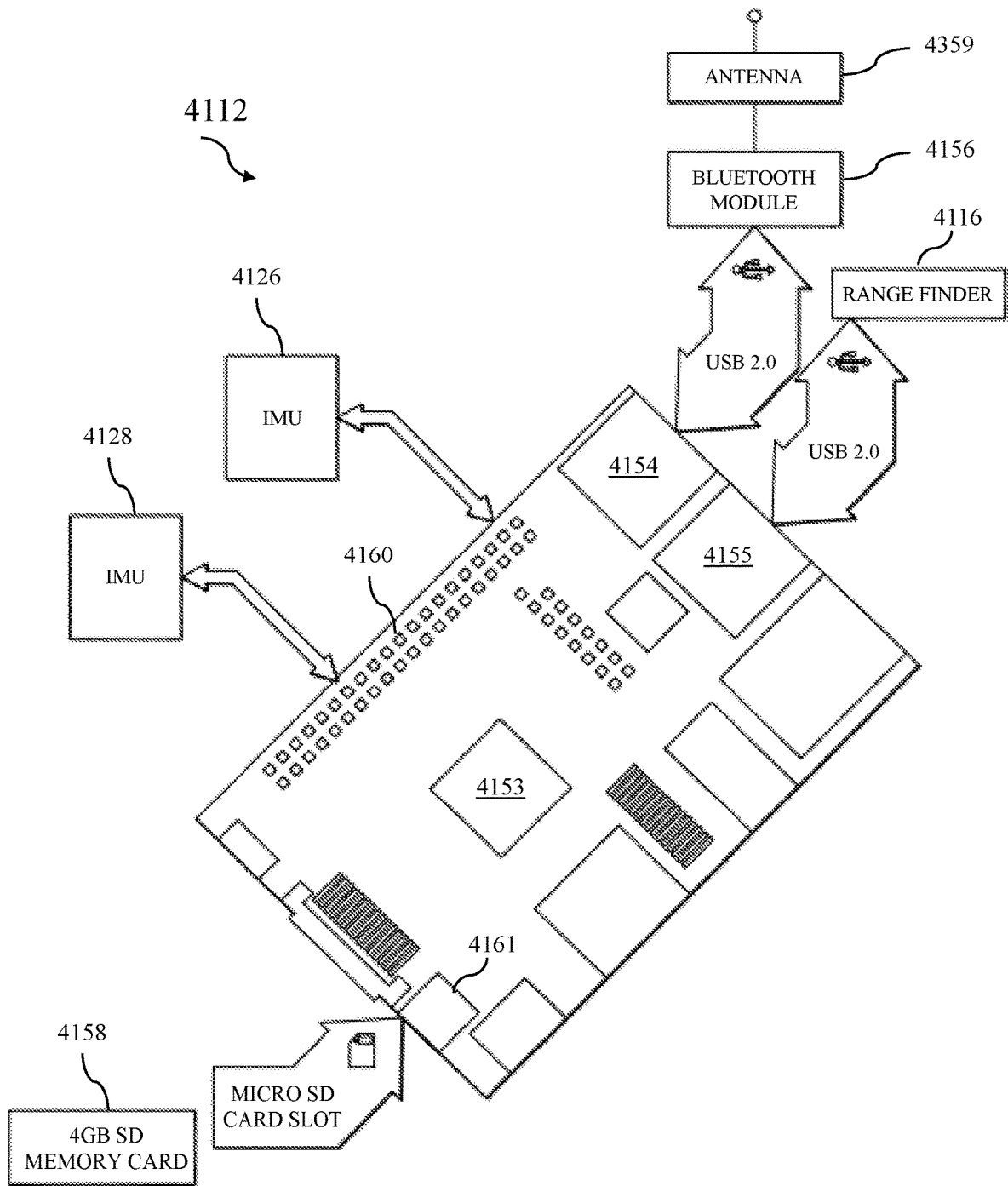


Figure 41B

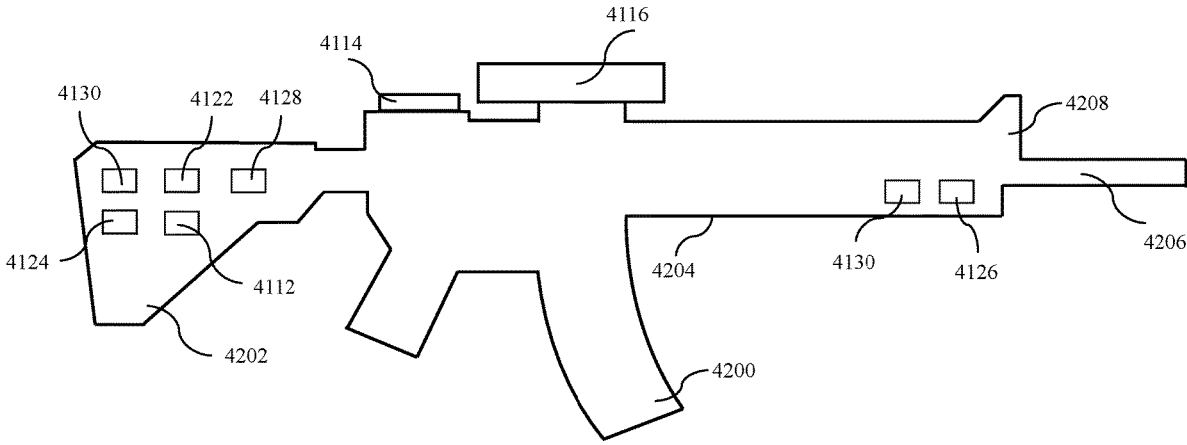


Figure 42A

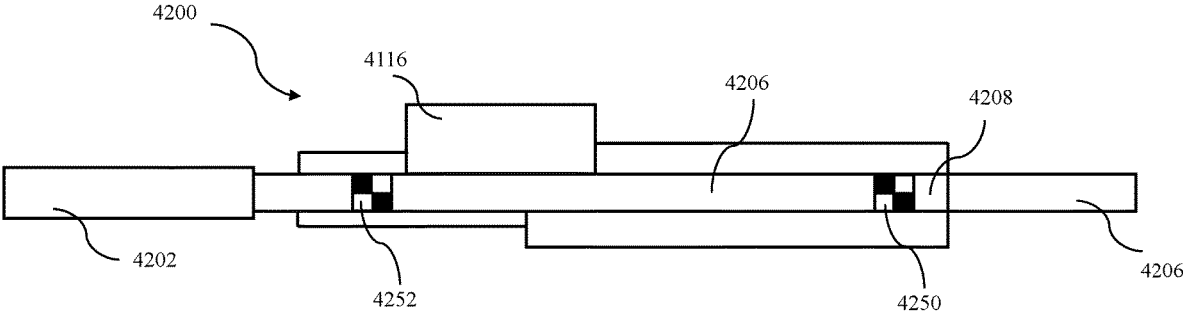


Figure 42B

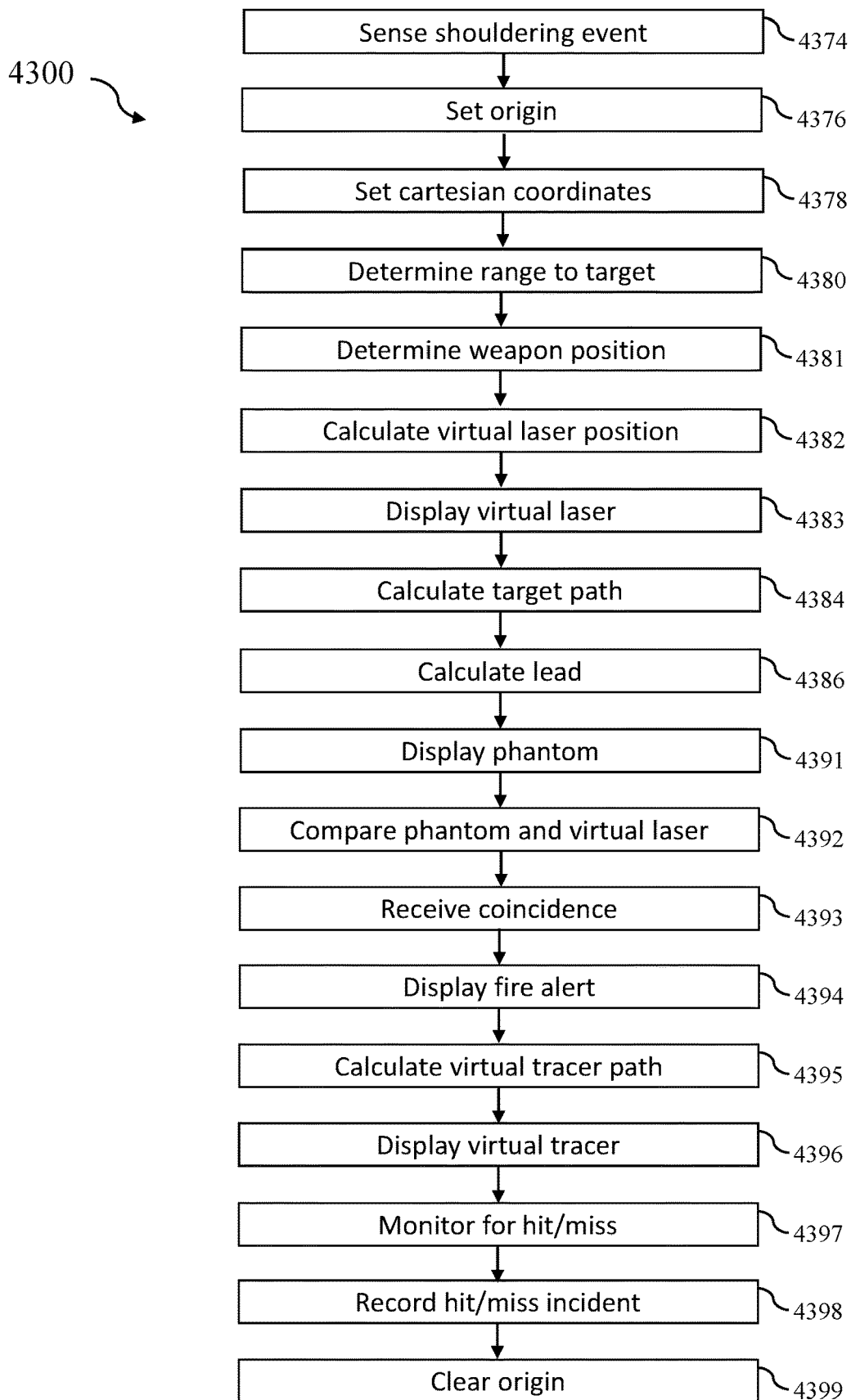


Figure 43A

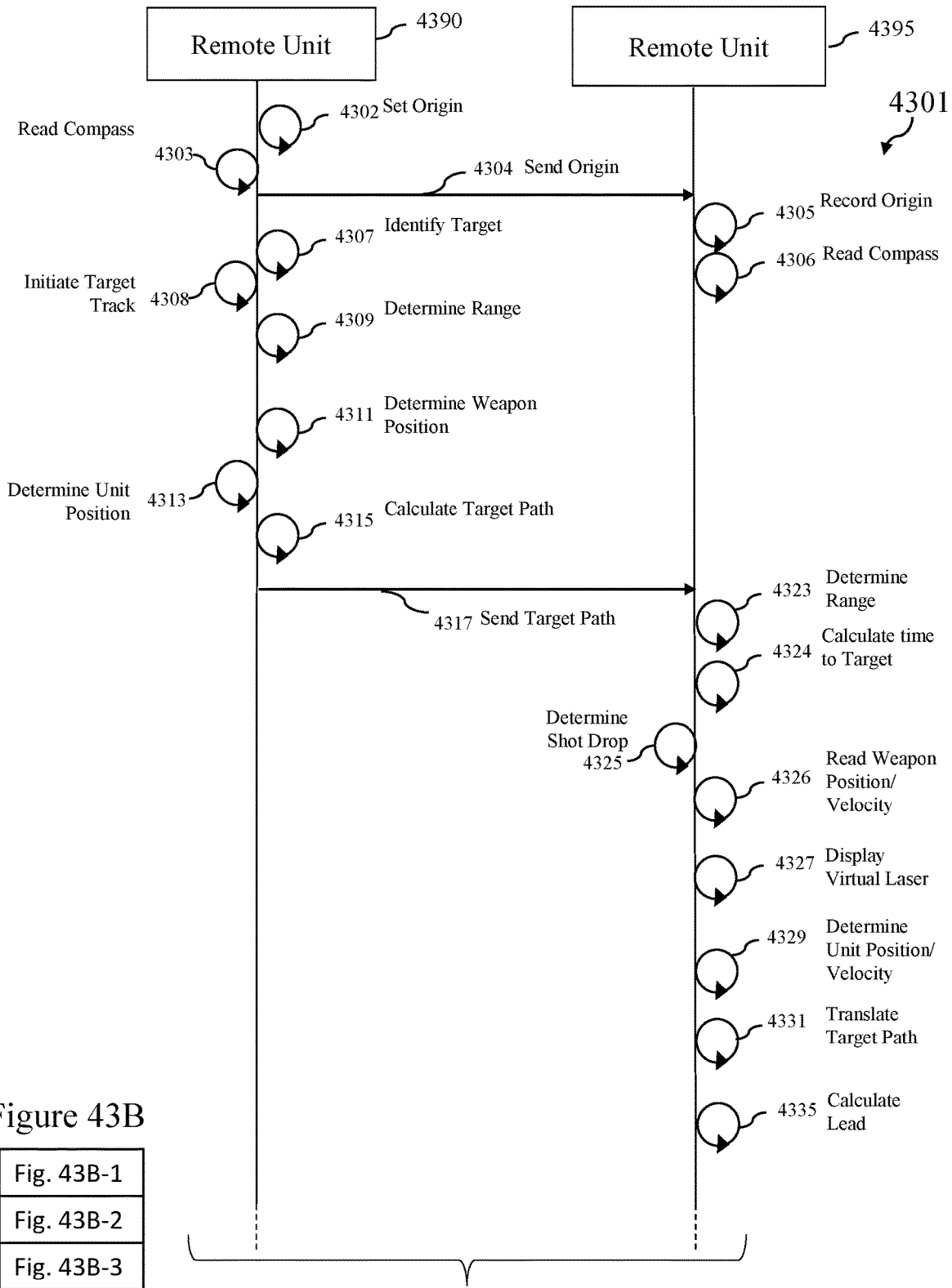


Figure 43B

Fig. 43B-1
Fig. 43B-2
Fig. 43B-3

To Figure 43B-2

Figure 43B-1

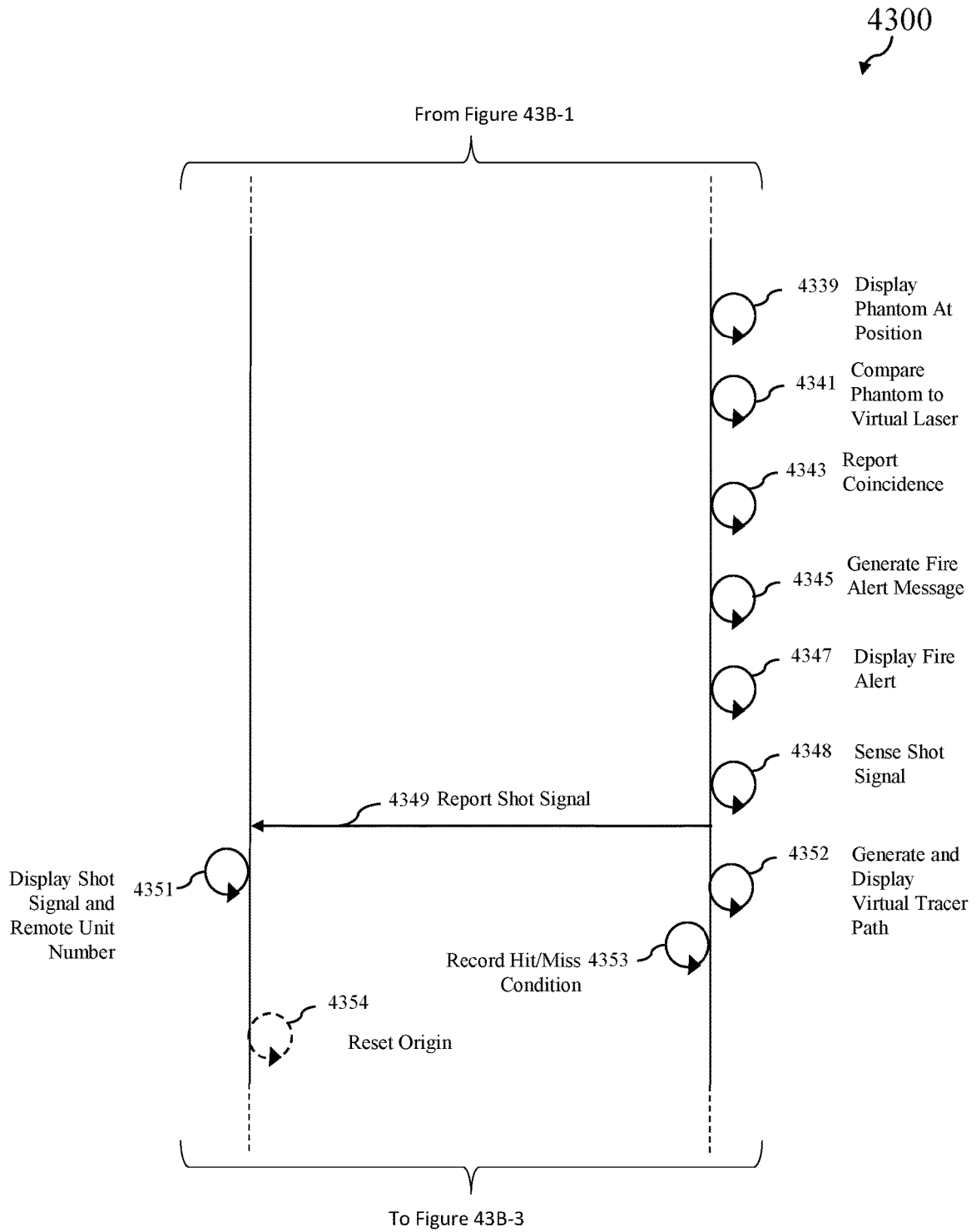


Figure 43B-2

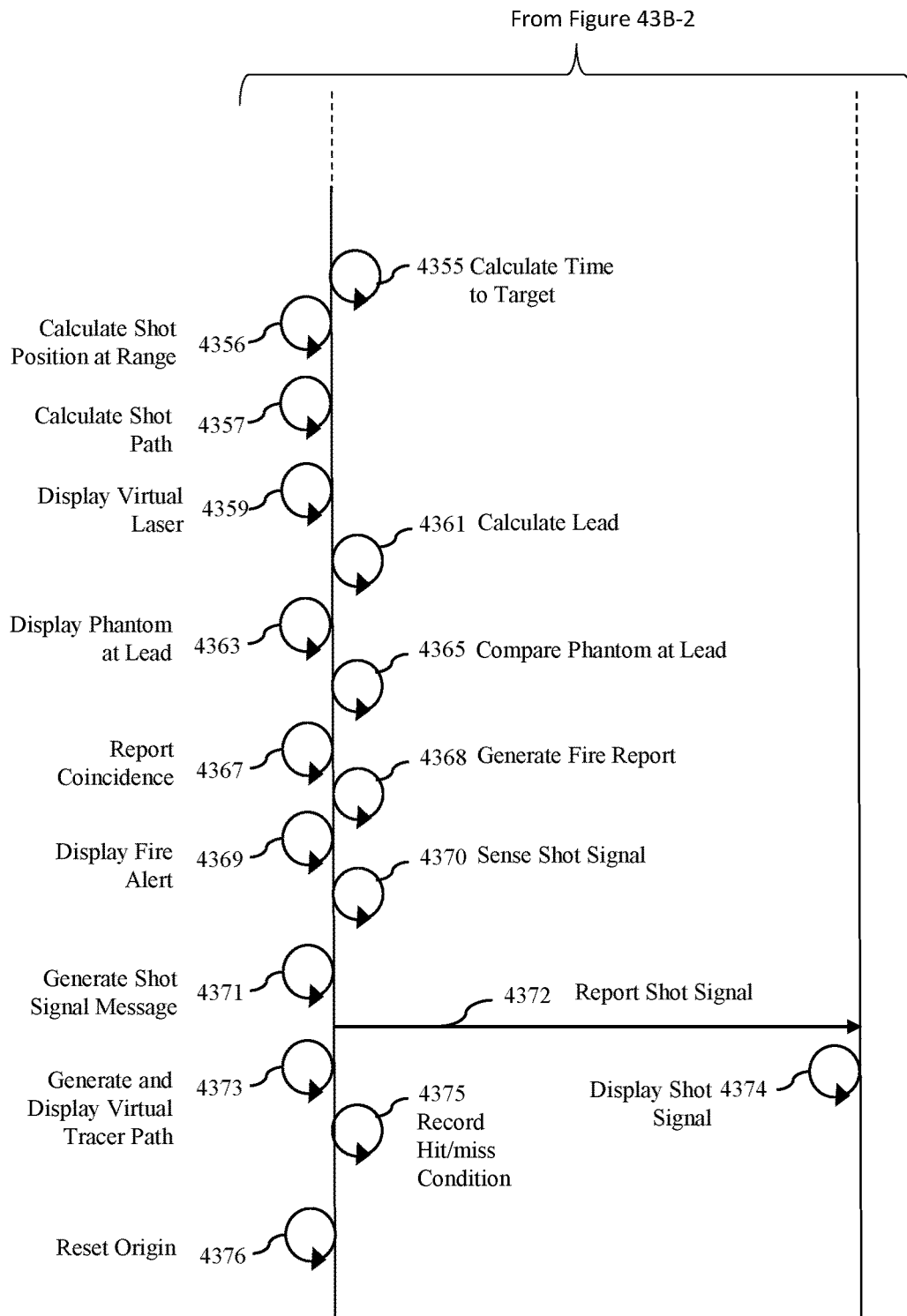


Figure 43B-3

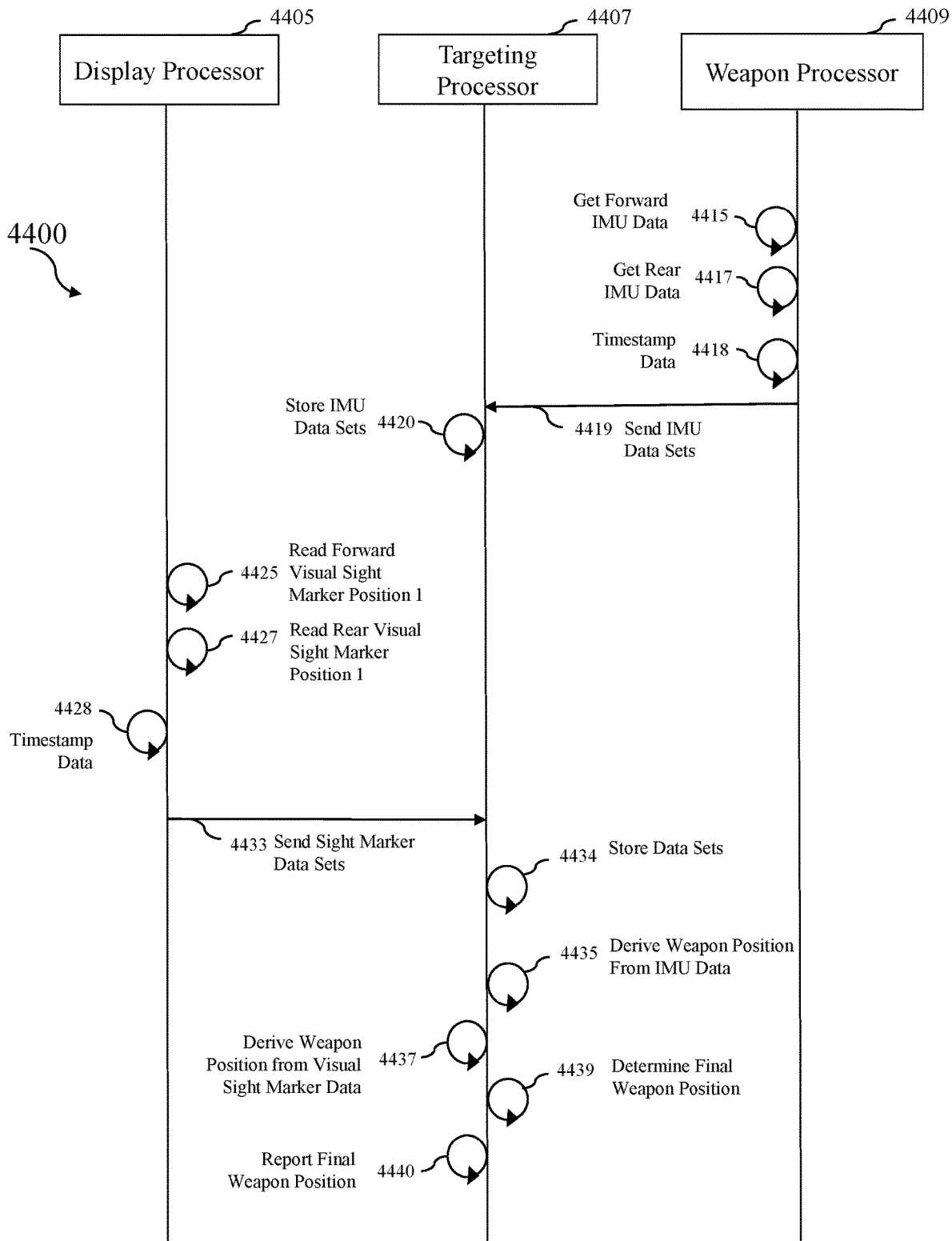


Figure 44  
Determine Weapon Position



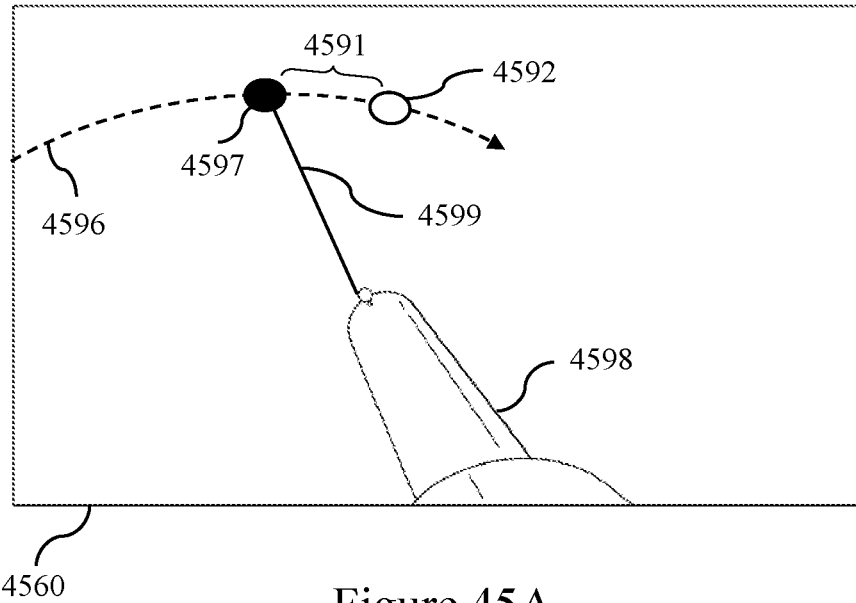


Figure 45A

Phantom Display

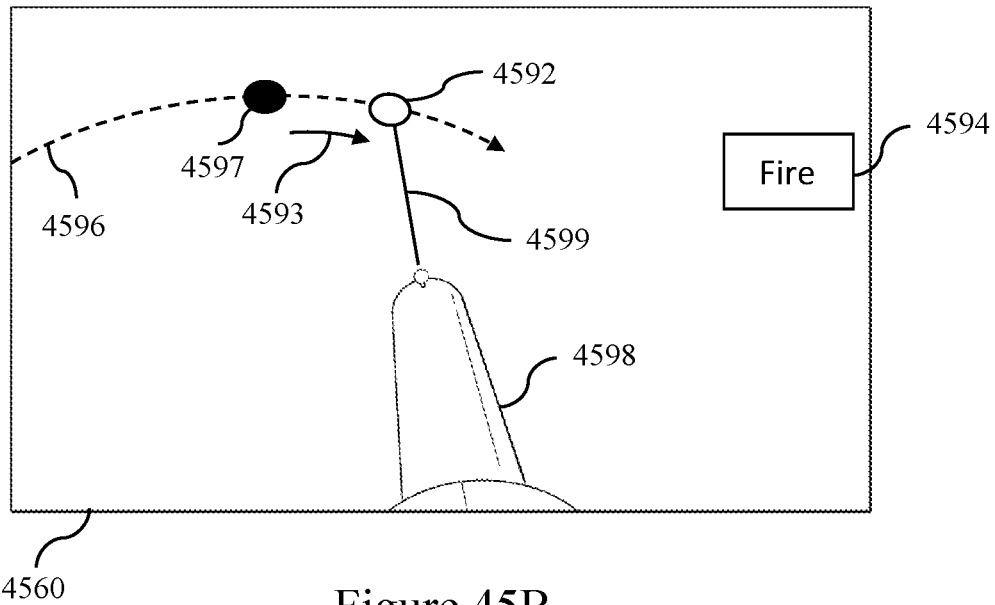


Figure 45B

Virtual Laser and Pull-Away Display

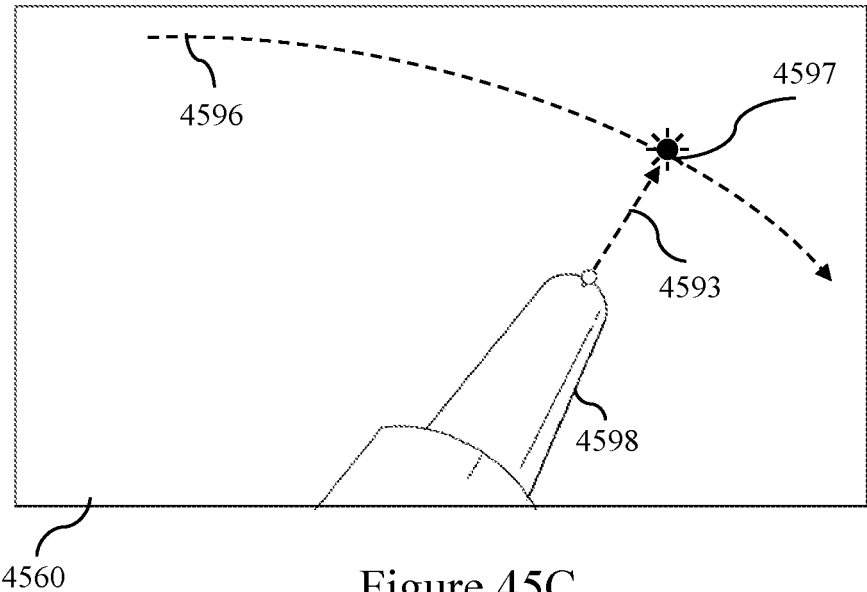
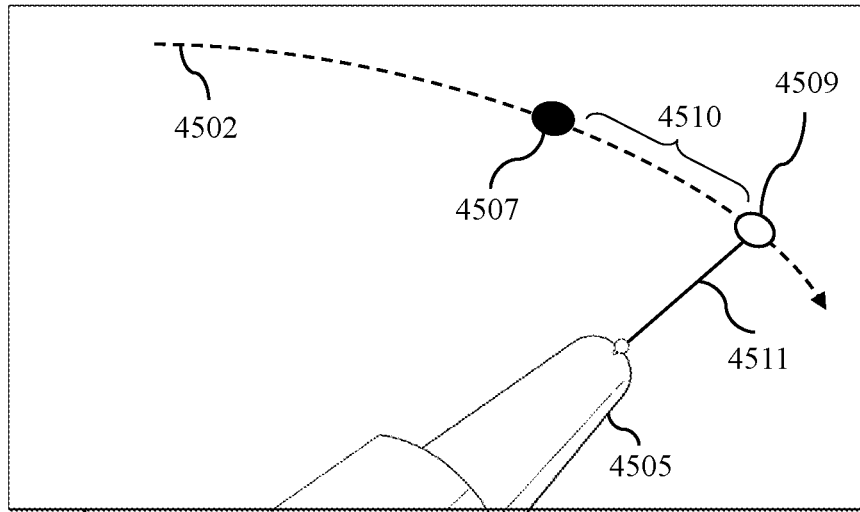


Figure 45C

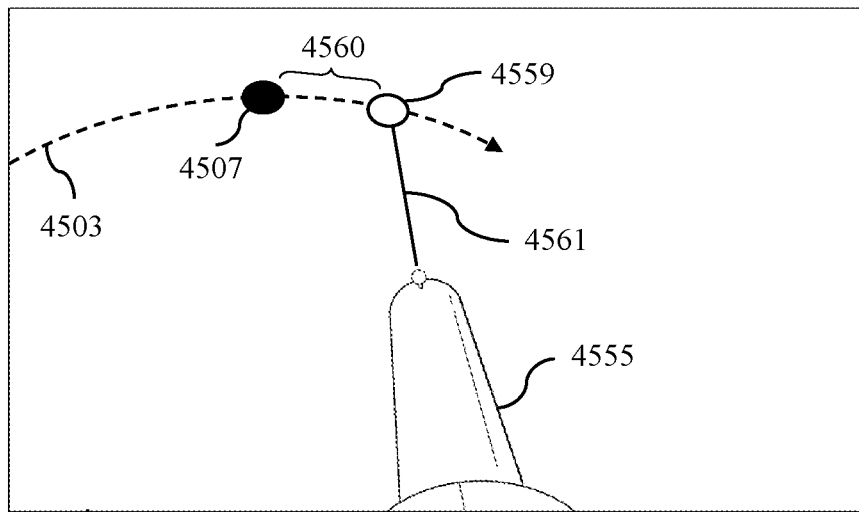
Virtual Tracer and Ballistic Intercept Display



4391

Figure 45D

Remote Unit 1 Virtual Laser Display



4396

Figure 45E

Remote Unit 2 Virtual Laser Display

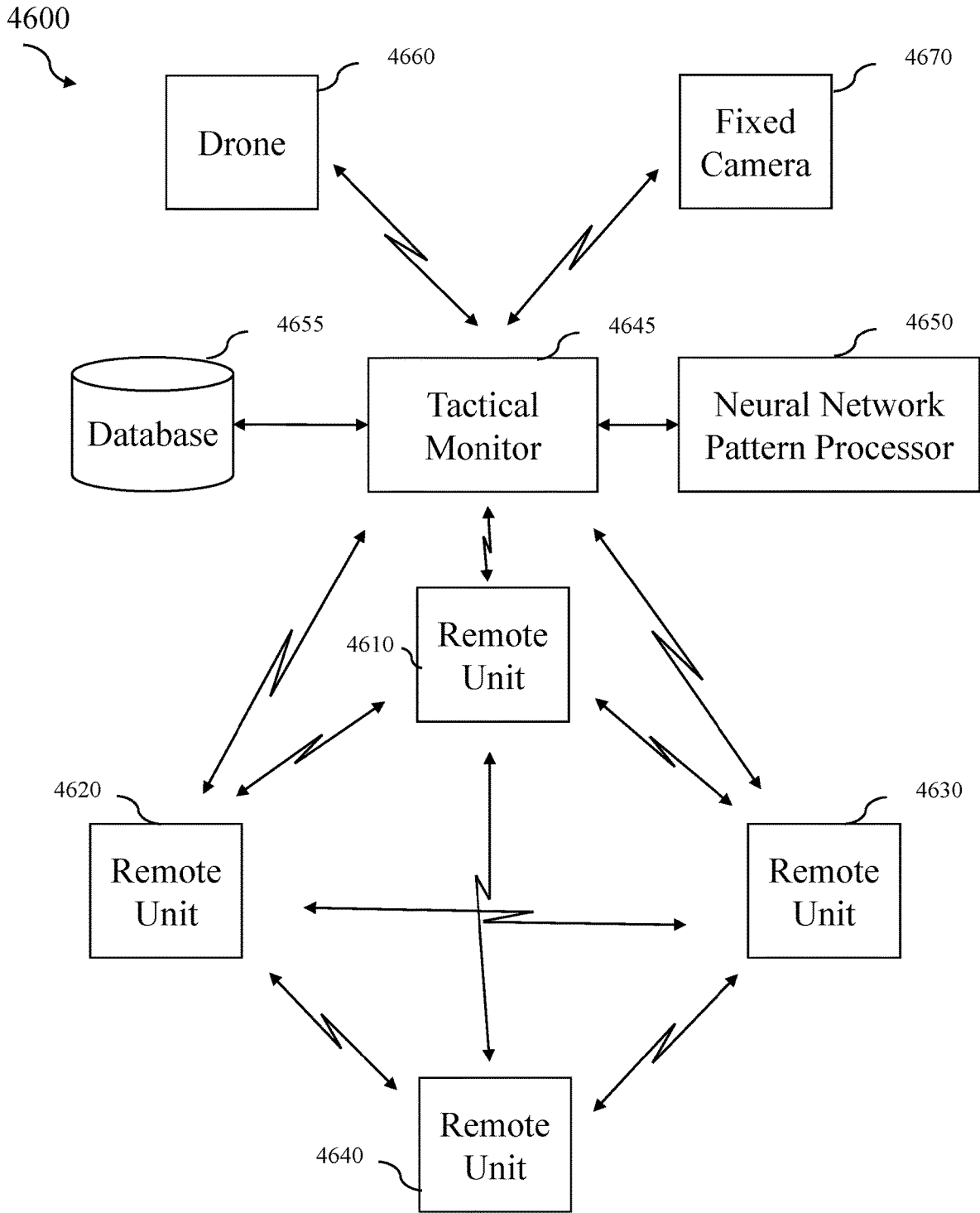


Figure 46A

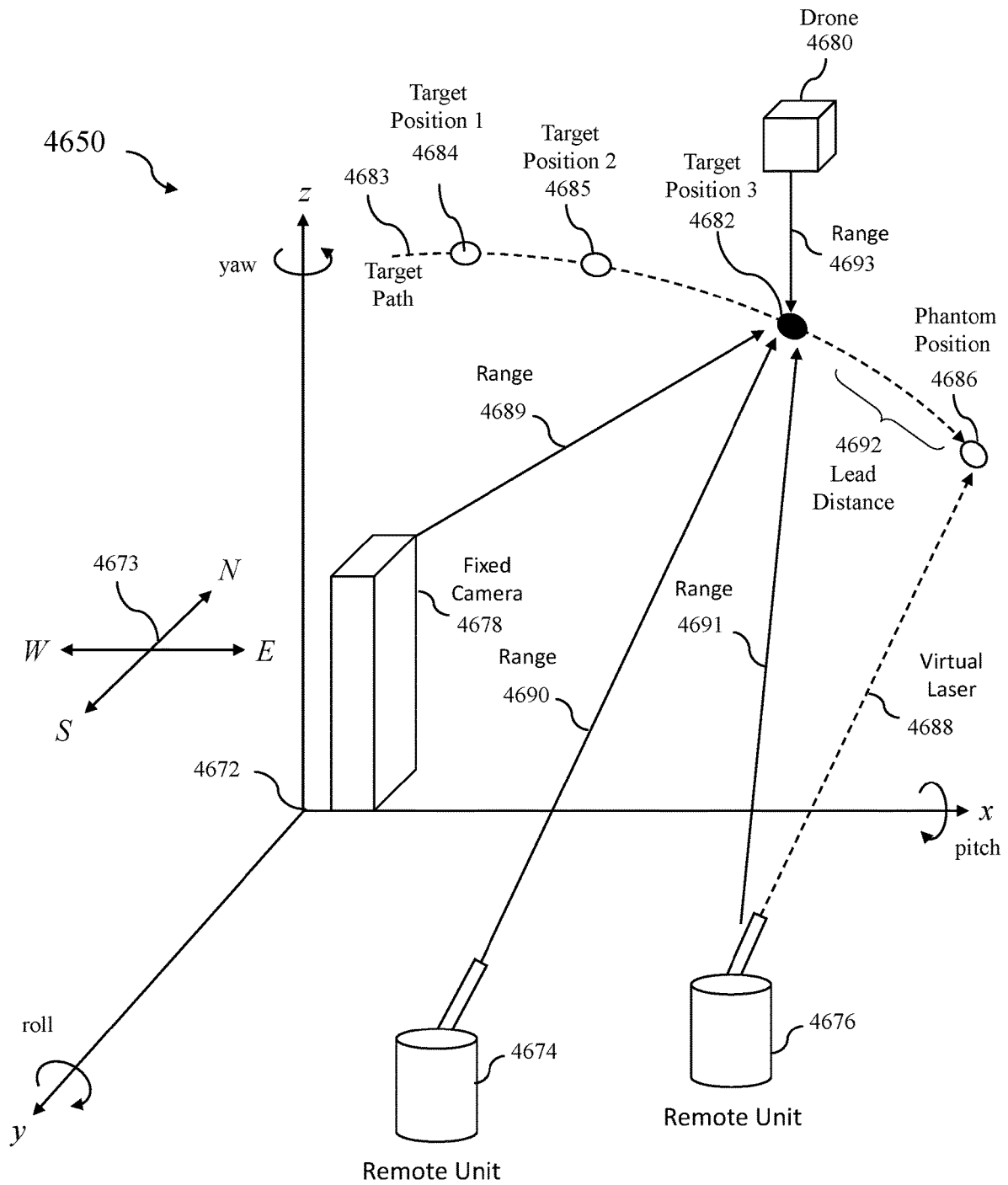


Figure 46B

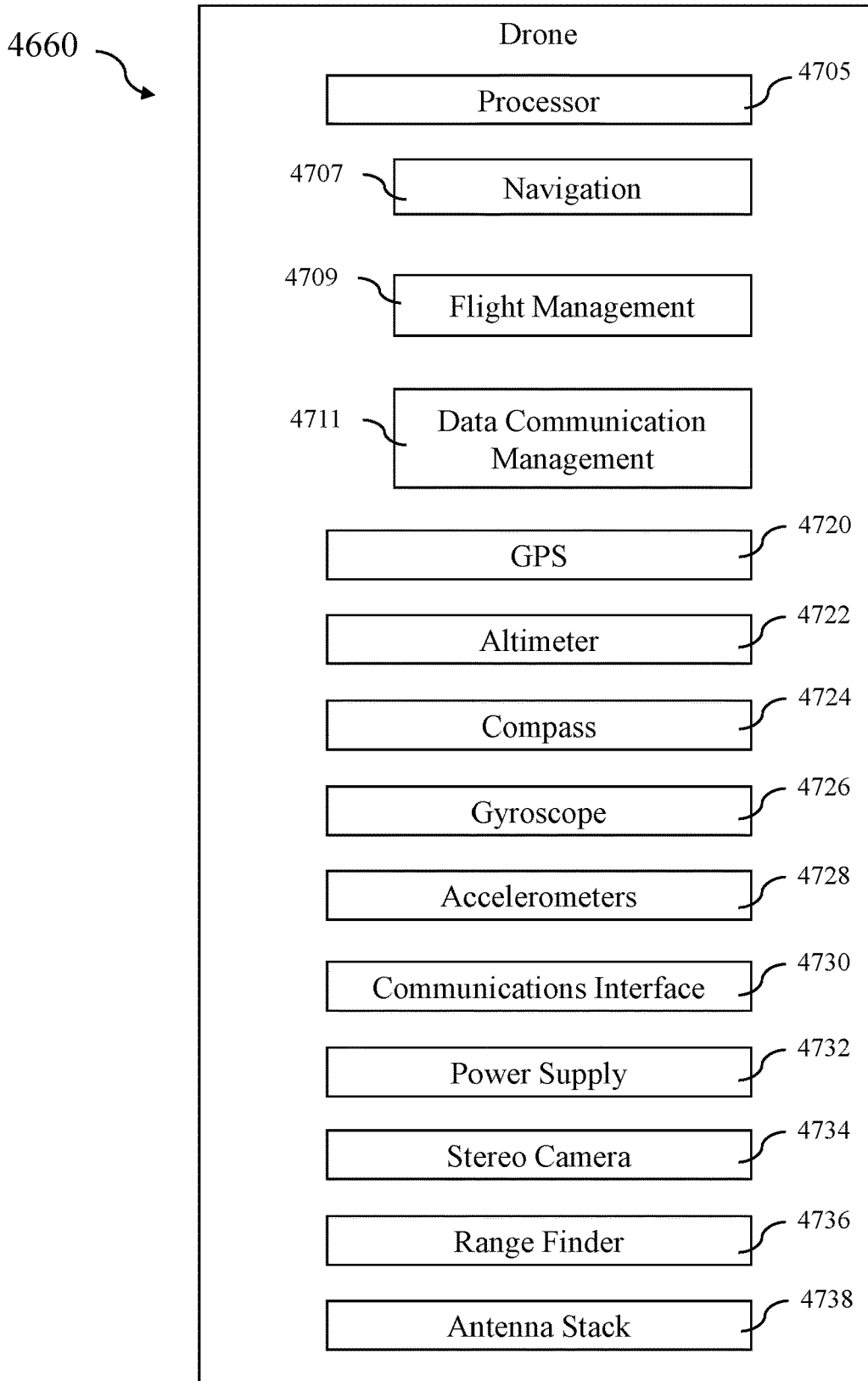


Figure 47

4670 ↘

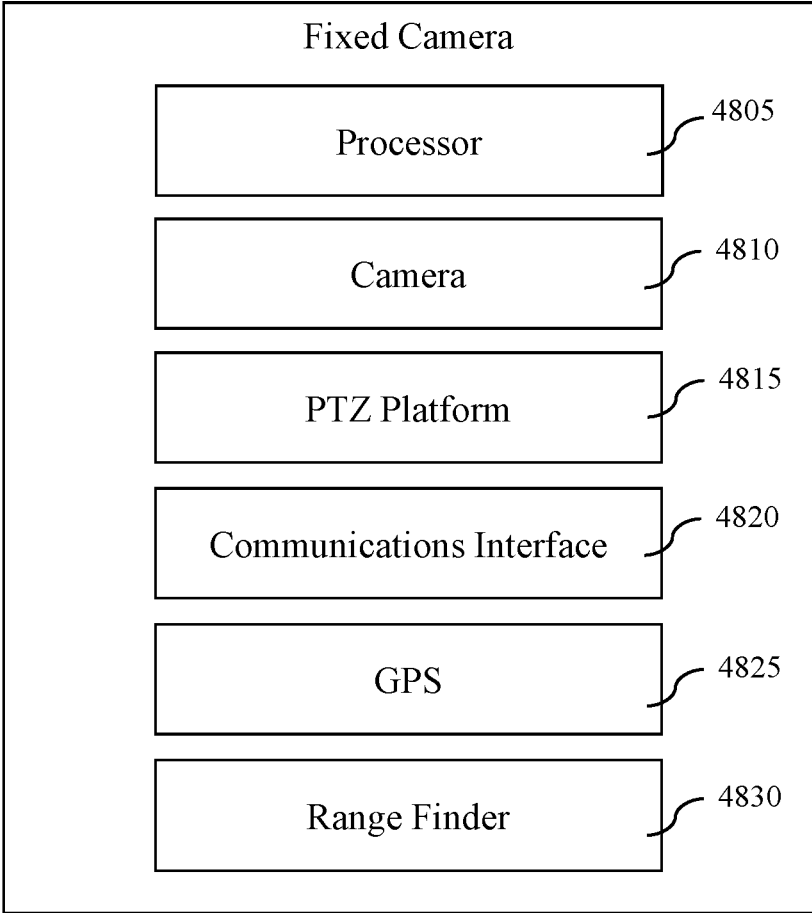
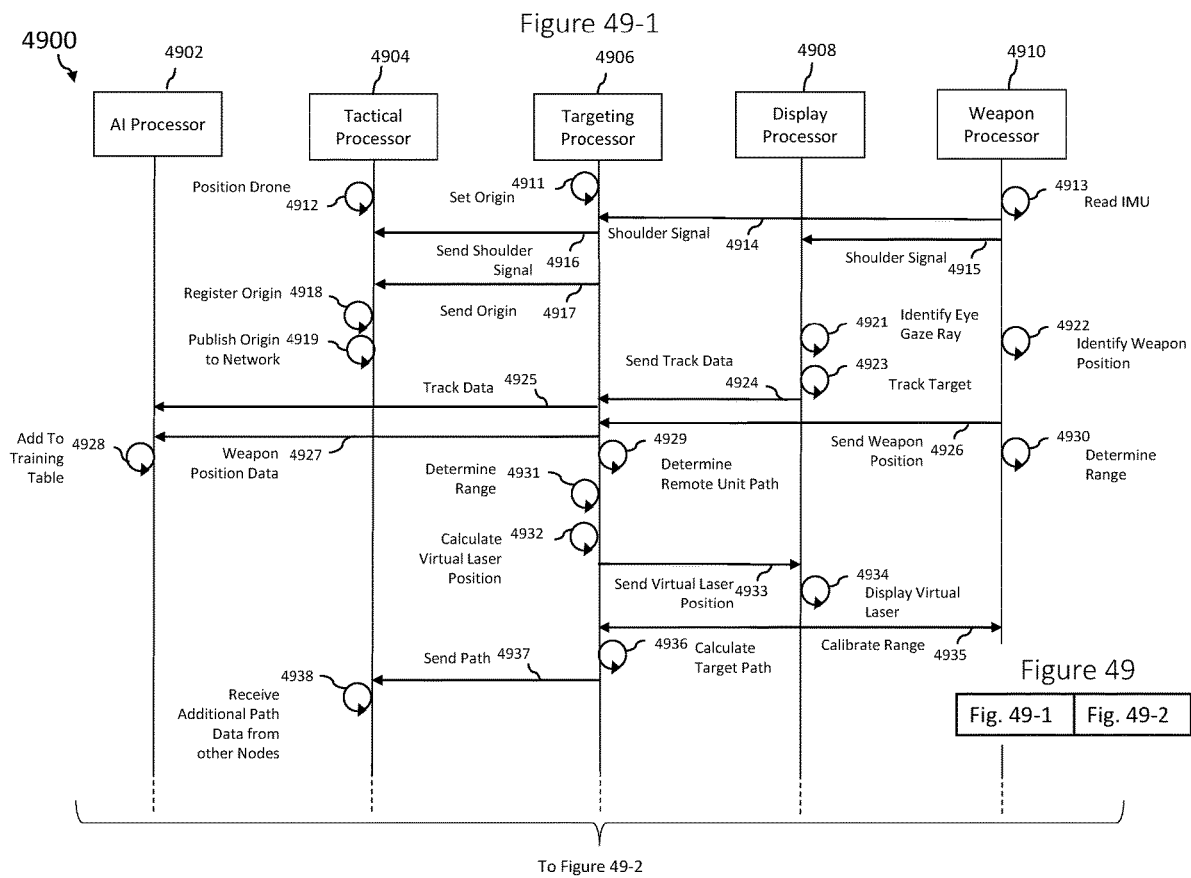


Figure 48





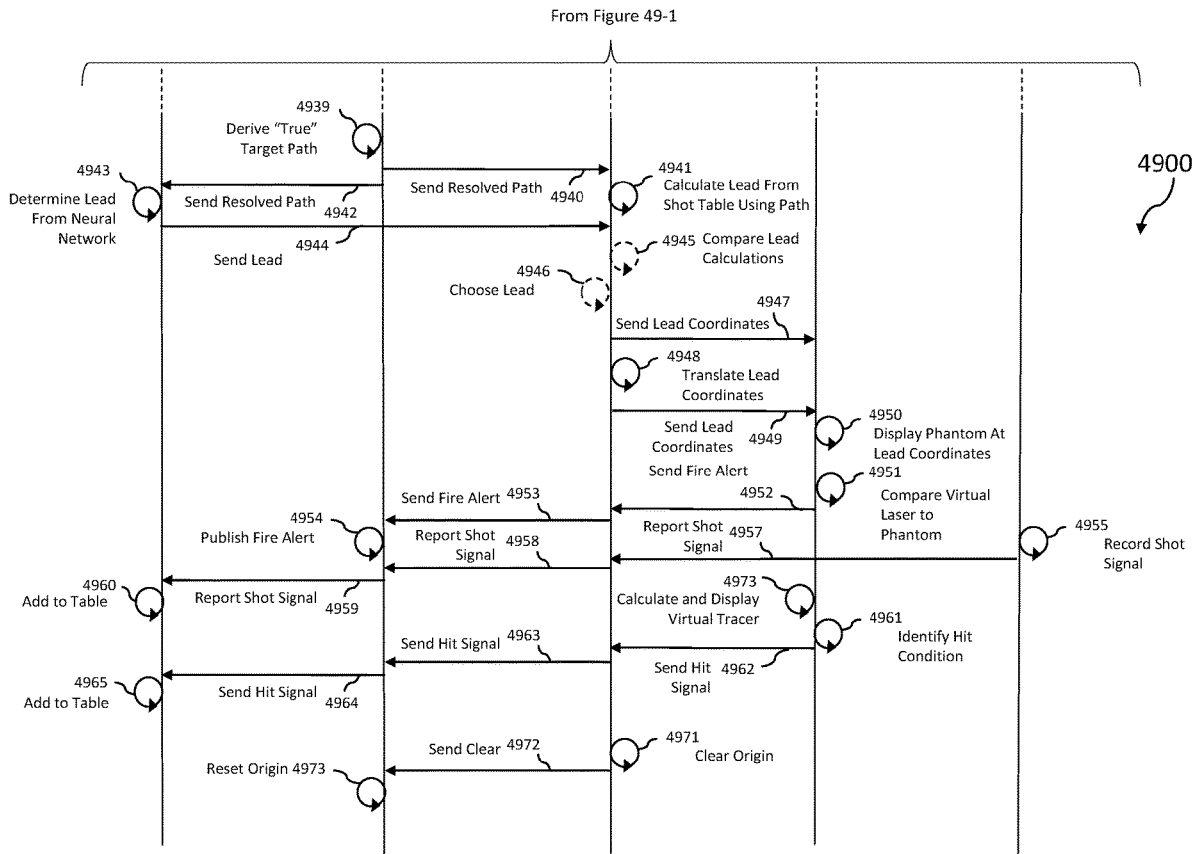


Figure 49-2

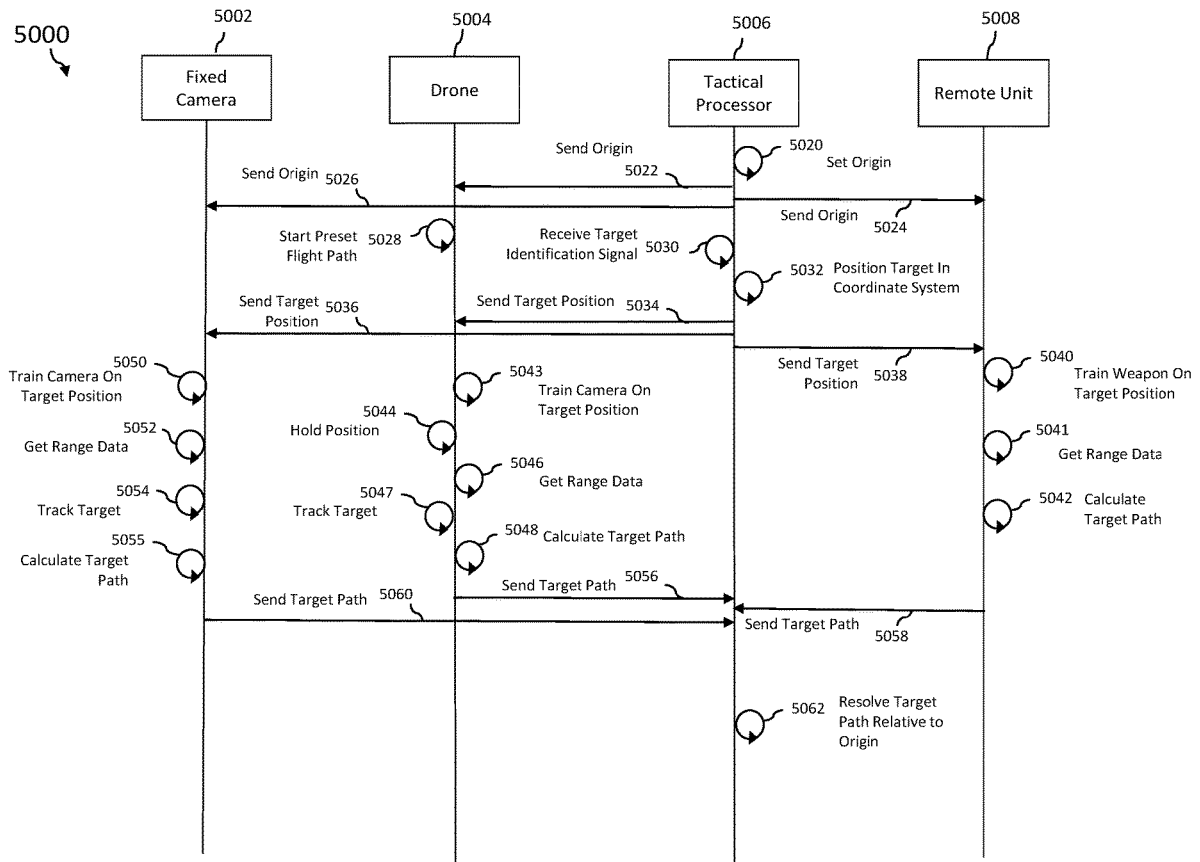


Figure 50

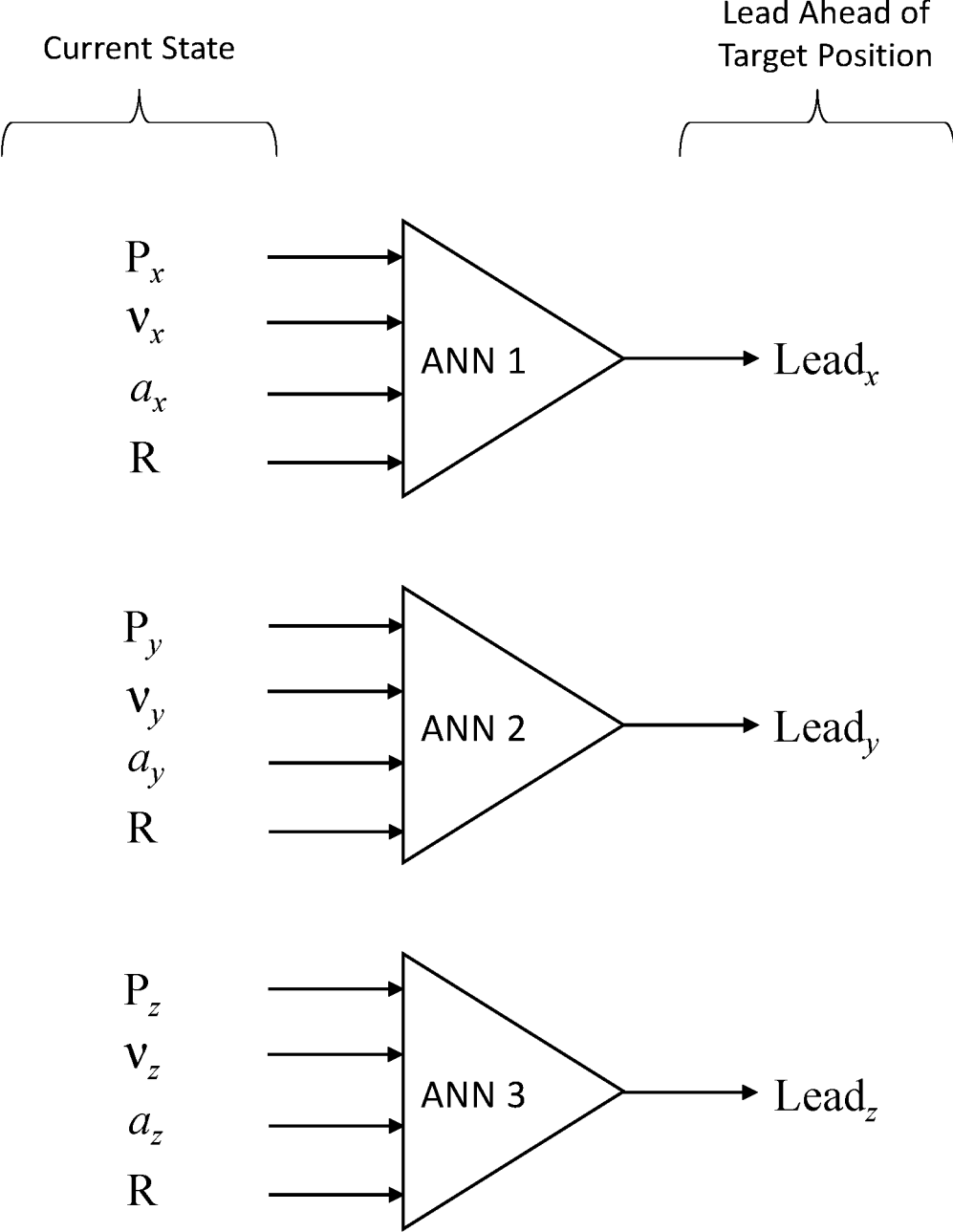


Figure 51

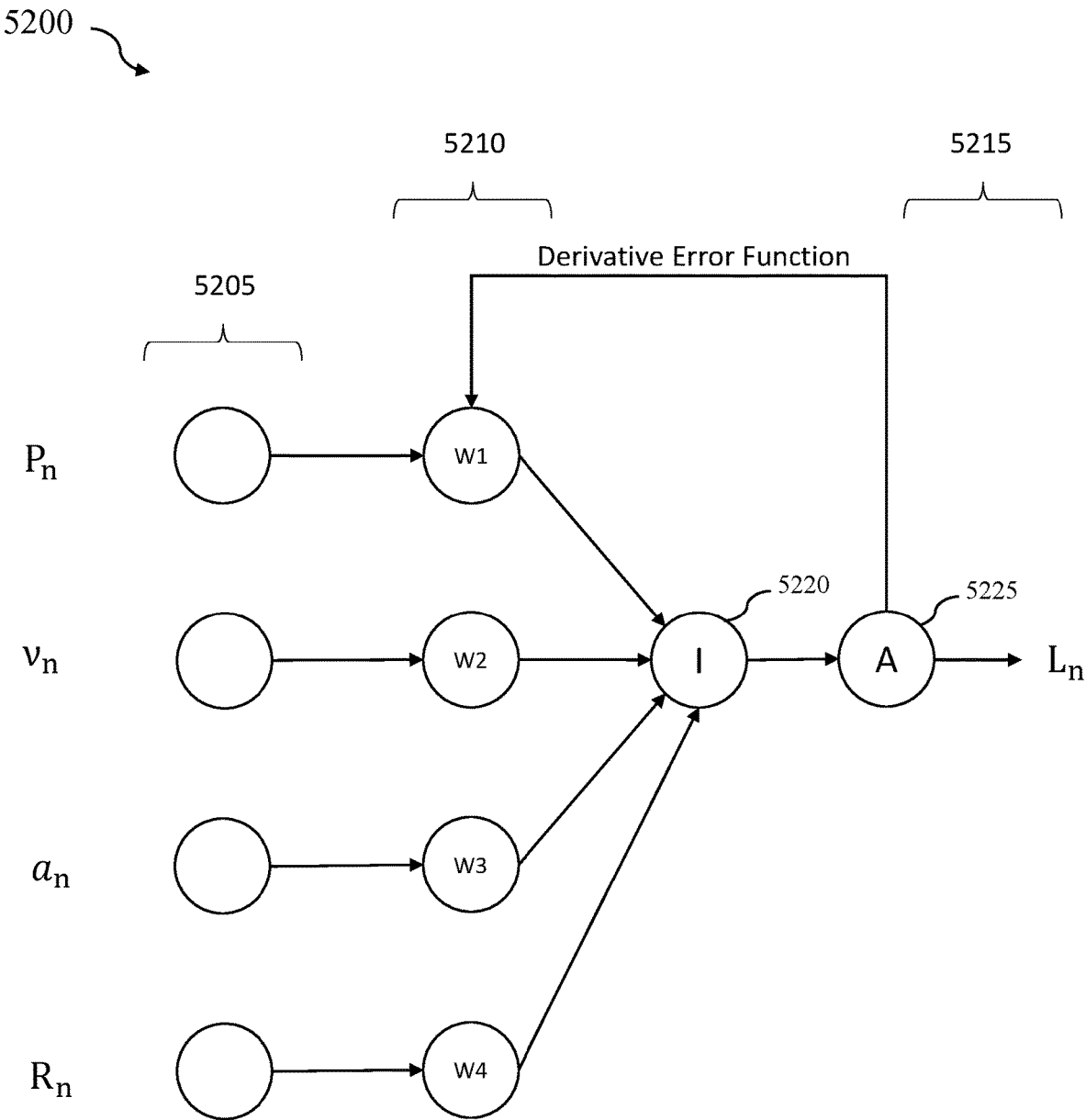


Figure 52

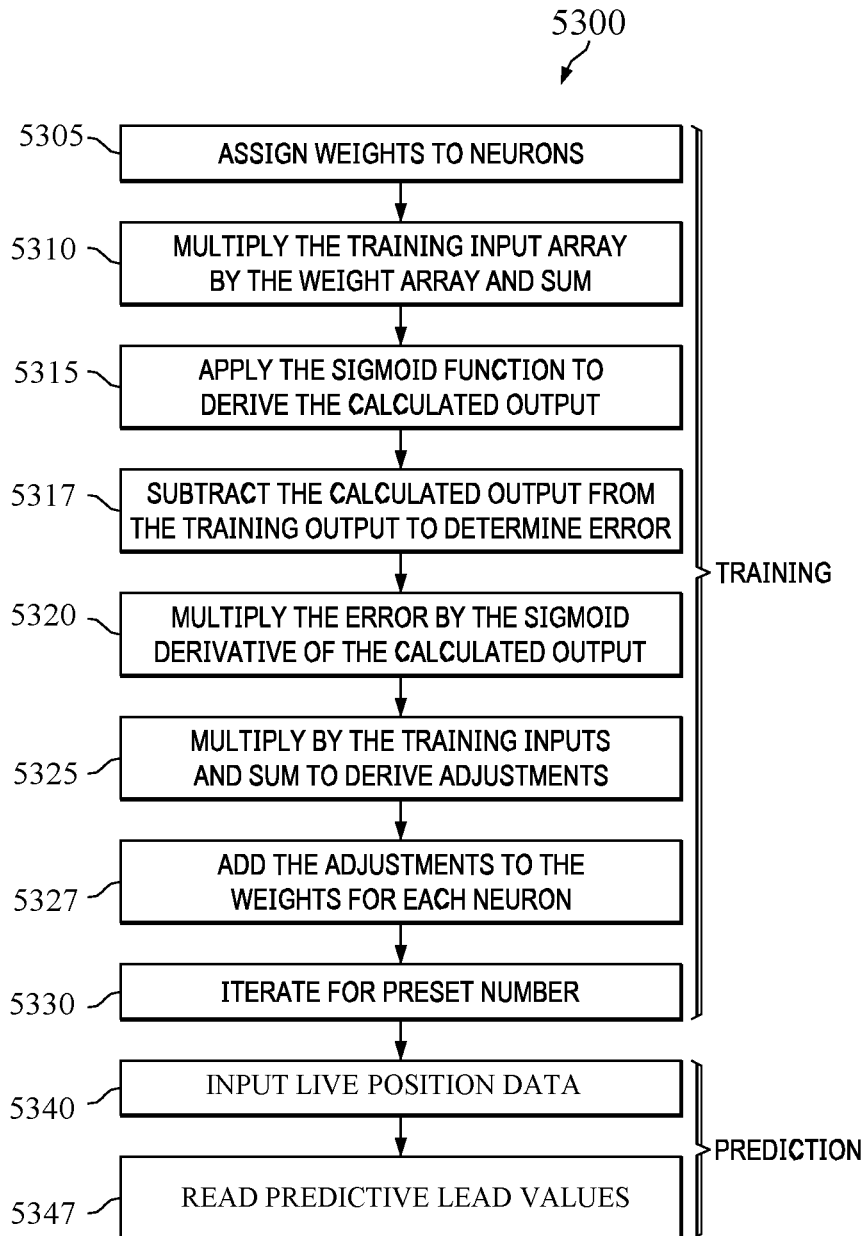


FIG. 53

```

class InterlockNeural():

    def __init__(self):
        # setting node weights – starting weight is any value between -1 and 1
        node.synaptic_weights = numpy.array([[.3,.4,0,-.3]])

    def sigmoid(self, x):
        # sigmoid function
        return 1 / (1 + numpy.exp(-x))

    def sigmoid_derivative(self, x):
        # derivative of sigmoid function
        return x * (1 - x)

    def train(self, training_inputs, training_outputs, training_iterations):

        #training the model
        for iteration in range(training_iterations):
            output = self.train(training_inputs)
            error = training_outputs - output
            adjustments = numpy.dot(training_inputs.T, error * self.sigmoid_derivative(output))
            node.synaptic_weights += adjustments

    def train(self, inputs):

        #passing inputs through the node
        inputs = inputs.astype(float)
        output = self.sigmoid(np.dot(inputs, self.synaptic_weights))
        return output

if __name__ == "__main__":

    #initializing
    neural_network = InterlockNeural()

    #training data consisting of 4 examples--4 input values and 1 output
    #in practice this array will hold a running accumulation of scaled path and lead data in a table
    of increasing size.
    training_inputs = numpy.array([[.25,.45,.75,.8],
                                   [.75,.35,.25,.9],
                                   [.4,.3,.60,.82],
                                   [.72,.41,.42,.76]])

    training_outputs = numpy.array([[.37,.42,.76,.99,1]])

    #training neural network
    neural_network.train(training_inputs, training_outputs, 100000)

```

Fig. 54

## SYSTEM AND METHOD FOR MARKSMANSHIP TRAINING

### CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation in part of U.S. application Ser. No. 16/397,983, filed on Apr. 29, 2019, now U.S. Pat. No. 10,584,940 granted on Mar. 10, 2020, which is a continuation in part of U.S. patent application Ser. No. 15/589,603 filed on May 8, 2017, now U.S. Pat. No. 10,274,287 granted on Apr. 30, 2019, which is a continuation in part of U.S. patent application Ser. No. 14/969,302 filed Dec. 15, 2015, now U.S. Pat. No. 10,234,240 granted on Mar. 19, 2019, which is a continuation in part of U.S. patent application Ser. No. 14/686,398 filed Apr. 14, 2015, now U.S. Pat. No. 10,030,937 granted on Jul. 24, 2018, which is a continuation in part of U.S. patent application Ser. No. 14/149,418 filed Jan. 7, 2014, now U.S. Pat. No. 9,261,332 granted on Feb. 16, 2016, which is a continuation in part of U.S. patent application Ser. No. 13/890,997 filed May 9, 2013, now U.S. Pat. No. 9,267,762 granted on Feb. 23, 2016. Each of the patent applications identified above is incorporated herein by reference in its entirety to provide continuity of disclosure.

### FIELD OF THE INVENTION

[0002] The present invention relates to devices for teaching marksmen how to properly lead a moving target with a weapon. More particularly, the invention relates to optical projection systems to monitor and simulate trap, skeet, and sporting clay shooting.

### BACKGROUND OF THE INVENTION

[0003] Marksmen typically train and hone their shooting skills by engaging in skeet, trap or sporting clay shooting at a shooting range. The objective for a marksman is to successfully hit a moving target by tracking at various distances and angles and anticipating the delay time between the shot and the impact. In order to hit the moving target, the marksman must aim the weapon ahead of and above the moving target by a distance sufficient to allow a projectile fired from the weapon sufficient time to reach the moving target. The process of aiming the weapon ahead of the moving target is known in the art as “leading the target.” “Lead” is defined as the distance between the moving target and the aiming point. The correct lead distance is critical to successfully hit the moving target. Further, the correct lead distance is increasingly important as the distance of the marksman to the moving target increases, the speed of the moving target increases, and the direction of movement becomes more oblique.

[0004] Trap shooting range 200 comprises firing lanes 201 and trap house 202. Stations 203, 204, 205, 206, and 207 are positioned along radius 214 from center 218 of trap house 202. Radius 214 is distance 216 from center 218. Distance 216 is 48 feet. Each of stations 203, 204, 205, 206, and 207 is positioned at radius 214 at equal arc lengths. Arc length 213 is 9 feet. Stations 208, 209, 210, 211, and 212 are positioned along radius 215 from center 218. Radius 215 is distance 217 from center 218. Distance 217 is 81 feet. Each of stations 208, 209, 210, 211, and 212 is positioned at radius 215 at equal arc lengths. Arc length 227 is 12 feet. Field 226 has length 221 from center 218 along center line 220 of trap

house 202 to point 219. Length 221 is 150 feet. Boundary line 222 extends 150 feet from center 218 at angle 224 from center line 220. Boundary line 223 extends 150 feet from center 218 at angle 225 from center line 220. Angles 224 and 225 are each 22° from center line 220. Trap house 202 launches clay targets at various trajectories within field 226. Marksman 228 positioned at any of stations 203, 204, 205, 206, 207, 208, 209, 210, 211, and 212 attempts to shoot and break the launched clay targets.

[0005] FIGS. 3A, 3B, 3C, and 3D depict examples of target paths and associated projectile paths illustrating the wide range of lead distances and distances required of the marksman. The term “projectile,” as used in this application, means any projectile fired from a weapon but more typically a shotgun round comprised of pellets of various sizes. For example, FIG. 3A shows a left to right trajectory 303 of target 301 and left to right intercept trajectory 304 for projectile 302. In this example, the intercept path is oblique, requiring the lead to be a greater distance along the positive X axis. FIG. 3B shows a left to right trajectory 307 of target 305 and intercept trajectory 308 for projectile 306. In this example, the intercept path is acute, requiring the lead to be a lesser distance in the positive X direction. FIG. 3C shows a right to left trajectory 311 of target 309 and intercepting trajectory 312 for projectile 310. In this example, the intercept path is oblique and requires a greater lead in the negative X direction. FIG. 3D shows a proximal to distal and right to left trajectory 315 of target 313 and intercept trajectory 316 for projectile 314. In this example, the intercept path is acute and requires a lesser lead in the negative X direction.

[0006] FIGS. 4A and 4B depict a range of paths of a clay target and an associated intercept projectile. The most typical projectile used in skeet and trap shooting is a shotgun round, such as a 12-gauge round or a 20 gauge round. When fired, the pellets of the round spread out into a “shot string” having a generally circular cross-section. The cross-section increases as the flight time of the pellets increases. Referring to FIG. 4A, clay target 401 moves along path 402. Shot string 403 intercepts clay target 401. Path 402 is an ideal path, in that no variables are considered that may alter path 402 of clay target 401 once clay target 401 is launched.

[0007] Referring to FIG. 4B, path range 404 depicts a range of potential flight paths for a clay target after being released on a shooting range. The flight path of the clay target is affected by several variables. Variables include mass, wind, drag, lift force, altitude, humidity, and temperature, resulting in a range of probable flight paths, path range 404. Path range 404 has upper limit 405 and lower limit 406. Path range 404 from launch angle  $\theta$  is extrapolated using:

$$x = x_o + v_{x_o}t + \frac{1}{2}a_x t^2 + C_x \quad \text{Eq. 1}$$

$$y = y_o + v_{y_o}t + \frac{1}{2}a_y t^2 + C_y \quad \text{Eq. 2}$$

where x is the clay position along the x-axis,  $x_o$  is the initial position of the clay target along the x-axis,  $v_{x_o}$  is the initial velocity along the x-axis,  $a_x$  is the acceleration along the x-axis, t is time, and  $C_x$  is the drag and lift variable along the x-axis, y is the clay position along the y-axis,  $y_o$  is the initial position of the clay target along the y-axis,  $v_{y_o}$  is the initial velocity along the y-axis,  $a_y$  is the acceleration along the y-axis, t is time, and  $C_y$  is the drag and lift variable along the x-axis. Upper limit 405 is a maximum distance along the x-axis with G at a maximum and a maximum along the y-axis with  $C_y$  at a maximum. Lower limit 406 is a minimum

distance along the x-axis with G at a minimum and a minimum along the y-axis with  $C_y$  at a minimum. Drag and lift are given by:

$$F_{drag} = \frac{1}{2} \rho v^2 C_D A \quad \text{Eq. 3}$$

where  $F_{drag}$  is the drag force,  $\rho$  is the density of the air,  $v$  is  $v_o$ ,  $A$  is the cross-sectional area, and  $C_D$  is the drag coefficient;

$$F_{lift} = \frac{1}{2} \rho v^2 C_L A \quad \text{Eq. 4}$$

where  $F_{lift}$  is the lift force,  $\rho$  is the density of the air,  $v$  is  $v_o$ ,  $A$  is the planform area, and  $C_L$  is the lift coefficient.

**[0008]** Referring to FIG. 5, an example of lead from the perspective of the marksman is described. Marksman **501** aims weapon **502** at clay target **503** moving along path **504** left to right. In order to hit clay target **503**, marksman **501** must anticipate the time delay for a projectile fired from weapon **502** to intercept clay target **503** by aiming weapon **502** ahead of clay target **503** at aim point **505**. Aim point **505** is lead distance **506** ahead of clay target **503** along path **504**. Marksman **501** must anticipate and adjust aim point **505** according to a best guess at the anticipated path of the target.

**[0009]** Clay target **503** has initial trajectory angles  $\gamma$  and  $\beta$ , positional coordinates  $y_1$  and a velocity  $v_1$ . Aim point **505** has coordinates  $x_2$ ,  $y_2$ . Lead distance **506** has x-component **507** and y-component **508**. X-component **507** and y-component **508** are calculated by:

$$\Delta x = x_2 - x_1 \quad \text{Eq. 5}$$

$$\Delta y = y_2 - y_1 \quad \text{Eq. 6}$$

where  $\Delta x$  is x component **507** and  $\Delta y$  is y component **508**. As  $\gamma$  increases,  $\Delta y$  must increase. As  $\beta$  increases,  $\Delta x$  must increase. As  $\beta$  increases,  $\Delta y$  must increase.

**[0010]** The prior art has attempted to address the problems of teaching proper lead distance with limited success. For example, U.S. Pat. No. 3,748,751 to Breglia, et al. discloses a laser, automatic fire weapon simulator. The simulator includes a display screen, a projector for projecting a motion picture on the display screen. A housing attaches to the barrel of the weapon. A camera with a narrow band-pass filter positioned to view the display screen detects and records the laser light and the target shown on the display screen. However, the simulator requires the marksman to aim at an invisible object, thereby making the learning process of leading a target difficult and time-consuming.

**[0011]** U.S. Pat. No. 3,940,204 to Yokoi discloses a clay shooting simulation system. The system includes a screen, a first projector providing a visible mark on the screen, a second projector providing an infrared mark on the screen, a mirror adapted to reflect the visible mark and the infrared mark to the screen, and a mechanical apparatus for moving the mirror in three dimensions to move the two marks on the screen such that the infrared mark leads the visible mark to simulate a lead-sighting point in actual clay shooting. A light receiver receives the reflected infrared light. However, the system in Yokoi requires a complex mechanical device to project and move the target on the screen, which leads to frequent failure and increased maintenance.

**[0012]** U.S. Pat. No. 3,945,133 to Mohon, et al. discloses a weapons training simulator utilizing polarized light. The simulator includes a screen and a projector projecting a two-layer film. The two-layer film is formed of a normal film and a polarized film. The normal film shows a background scene with a target with non-polarized light. The polarized

film shows a leading target with polarized light. The polarized film is layered on top of the normal non-polarized film. A polarized light sensor is mounted on the barrel of a gun. However, the weapons training simulator requires two cameras and two types of film to produce the two-layered film making the simulator expensive and time-consuming to build and operate.

**[0013]** U.S. Pat. No. 5,194,006 to Zaenglein, Jr. discloses a shooting simulator. The simulator includes a screen, a projector for displaying a moving target image on the screen, and a weapon connected to the projector. When a marksman pulls the trigger a beam of infrared light is emitted from the weapon. A delay is introduced between the time the trigger is pulled and the beam is emitted. An infrared light sensor detects the beam of infrared light. However, the training device in Zaenglein, Jr. requires the marksman to aim at an invisible object, thereby making the learning process of leading a target difficult and time-consuming.

**[0014]** U.S. Patent Publication No. 2010/0201620 to Sargent discloses a firearm training system for moving targets. The system includes a firearm, two cameras mounted on the firearm, a processor, and a display. The two cameras capture a set of stereo images of the moving target along the moving target's path when the trigger is pulled. However, the system requires the marksman to aim at an invisible object, thereby making the learning process of leading a target difficult and time-consuming. Further, the system requires two cameras mounted on the firearm making the firearm heavy and difficult to manipulate leading to inaccurate aiming and firing by the marksman when firing live ammunition without the mounted cameras.

**[0015]** The prior art fails to disclose or suggest a system and method for simulating a lead for a moving target using generated images of targets projected at the same scale as viewed in the field and a phantom target positioned ahead of the targets having a variable contrast. The prior art further fails to disclose or suggest a system and method for simulating lead in a virtual reality system. Therefore, there is a need in the art for a shooting simulator that recreates moving targets at the same visual scale as seen in the field with a phantom target to teach proper lead of a moving target in a virtual reality platform.

#### SUMMARY OF THE INVENTION

**[0016]** A system and method for simulating lead of a target includes a network, a simulation administrator connected to the network, a database connected to the simulation administrator, and a user device connected to the network. The user device includes a set of virtual reality unit, and a computer connected to the virtual reality unit and to the network. A set of position trackers are connected to the computer.

**[0017]** In a preferred embodiment, a target is simulated. In one embodiment, a simulated weapon is provided. In another embodiment, a set of sensors is attached to a real weapon. In another embodiment, a set of gloves having a set of sensors is worn by a user. The system generates a simulated target and displays the simulated target upon launch of the generated target. The computer tracks the position of the generated target and the position of the virtual reality unit and the weapon to generate a phantom target and a phantom halo. The generated phantom target and the generated phantom halo are displayed on the virtual reality unit at a lead distance and a drop distance from the live target as viewed through the virtual reality unit. The computer



determines a hit or a miss of the generated target using the weapon, the phantom target, and the phantom halo. In one embodiment, the disclosed system and method is implemented in a two-dimensional video game.

[0018] The present disclosure provides a system which embodies significantly more than an abstract idea including technical advancements in the field of data processing and a transformation of data which is directly related to real-world objects and situations. The disclosed embodiments create and transform imagery in hardware, for example, a weapon peripheral and a sensor attachment to a real weapon.

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0019] The disclosed embodiments will be described with reference to the accompanying drawings.

[0020] FIG. 1 is a plan view of a skeet shooting range.

[0021] FIG. 2 is a plan view of a trap shooting range.

[0022] FIG. 3A is a target path and an associated projectile path.

[0023] FIG. 3B is a target path and an associated projectile path.

[0024] FIG. 3C is a target path and an associated projectile path.

[0025] FIG. 3D is a target path and an associated projectile path.

[0026] FIG. 4A is an ideal path of a moving target.

[0027] FIG. 4B is a range of probable flight paths of a target.

[0028] FIG. 5 is a perspective view of a marksman aiming at a moving target.

[0029] FIG. 6 is a schematic of a simulator system of a preferred embodiment.

[0030] FIG. 7 is a schematic of a simulation administrator of a preferred embodiment.

[0031] FIG. 8 is a schematic of a user device of a simulator system of a preferred embodiment.

[0032] FIG. 9A is a side view of a user device of a virtual reality simulator system of a preferred embodiment.

[0033] FIG. 9B is a front view of a user device of a virtual reality simulator system of a preferred embodiment.

[0034] FIG. 9C is a side view of a user device of an augmented reality simulator system of a preferred embodiment.

[0035] FIG. 9D is a front view of a user device of an augmented reality simulator system of a preferred embodiment.

[0036] FIG. 10A is a side view of a simulated weapon for a virtual reality system of a preferred embodiment.

[0037] FIG. 10B is a side view of a real weapon with a set of sensors attached for a virtual reality system of a preferred embodiment.

[0038] FIG. 10C is a detail view of a trigger sensor of a preferred embodiment.

[0039] FIG. 10D is a detail view of a set of muzzle sensors of a preferred embodiment.

[0040] FIG. 10E is a detail view of a set of a transmitter base of a preferred embodiment.

[0041] FIG. 10F is a detail view of a set of muzzle sensors used with the transmitter base of FIG. 10E of a preferred embodiment.

[0042] FIG. 10G is a detail view of a removable plug with light emitting diodes for a weapon of a preferred embodiment.

[0043] FIG. 10H is a detail view of a removable plug with light emitting diodes attached to a weapon of a preferred embodiment.

[0044] FIG. 10I is a detail view of a removable collar with light emitting diodes attached to a weapon of a preferred embodiment.

[0045] FIG. 10J is a side view of a weapon with an adjustable stock for a virtual reality simulator system of a preferred embodiment.

[0046] FIG. 10K is a detail view of a trigger sensor of a preferred embodiment.

[0047] FIG. 11A is a simulation view of a weapon having an iron sight of a preferred embodiment.

[0048] FIG. 11B is a simulation view of a weapon having a reflex sight of a preferred embodiment.

[0049] FIG. 11C is a simulation view of a weapon having a holographic sight of a preferred embodiment.

[0050] FIG. 12 is a schematic view of a virtual reality simulation environment of a preferred embodiment.

[0051] FIG. 13 is a command input menu for a virtual reality simulator system of a preferred embodiment.

[0052] FIG. 14 is a flow chart of a method for runtime process of a virtual reality simulation system of a preferred embodiment.

[0053] FIG. 15A is top view of a user and a simulation environment of a preferred embodiment.

[0054] FIG. 15B is a flow chart of a method for determining a view for a user device with respect to a position and an orientation of the user device and the weapon.

[0055] FIG. 15C is a flow chart of a method for mapping the position and orientation of the user device and the weapon to the simulation environment for determining a display field of view a preferred embodiment.

[0056] FIG. 16A is a flowchart of a method for determining a phantom and halo of a preferred embodiment.

[0057] FIG. 16B is a plan view of a target and a phantom of a preferred embodiment.

[0058] FIG. 16C is an isometric view of a target and a phantom of a preferred embodiment.

[0059] FIG. 17 is a user point of view of a virtual reality simulation system of a preferred embodiment.

[0060] FIG. 18 is an isometric view of an input device configured to be mounted on a rail system of a weapon of a preferred embodiment.

[0061] FIG. 19 is a simulation view that shows beams being projected from a barrel of a weapon of a preferred embodiment.

[0062] FIG. 20A is a five stand field of a preferred embodiment.

[0063] FIG. 20B is a sporting clay field of a preferred embodiment.

[0064] FIG. 21A is diagram of a preferred embodiment of a simulation system.

[0065] FIG. 21B is a diagram of a virtual reality system of a preferred embodiment.

[0066] FIG. 21C is a diagram of an augmented reality system of a preferred embodiment.

[0067] FIG. 22A is a diagram of a system using a positioning detector at an end of a barrel in a preferred embodiment.

[0068] FIG. 22B is a diagram of a system using a positioning detector mounted under a barrel in a preferred embodiment.

[0069] FIG. 22C is a diagram of a system using sight markings in a preferred embodiment.

[0070] FIG. 22D is a diagram of a system using sight markings and a sensor thimble in a preferred embodiment.

[0071] FIG. 22E is a diagram of a positioning detector in a preferred embodiment.

[0072] FIGS. 23A and 23B are diagrams of a trigger unit in a preferred embodiment.

[0073] FIG. 23C is a diagram of a processor board of a trigger unit in a preferred embodiment.

[0074] FIGS. 24A and 24B are diagrams of a mounting arbor in a preferred embodiment.

[0075] FIGS. 24C and 24D are diagrams of a barrel clamp in a preferred embodiment.

[0076] FIGS. 25A through 25D are diagrams of electronic cartridges in preferred embodiments.

[0077] FIGS. 25E and 25F are diagrams of a sensor arbor in a preferred embodiment.

[0078] FIG. 25G is a diagram of a sensor thimble in a preferred embodiment.

[0079] FIG. 26 is a diagram of a computer implemented method for determining a launcher location of a preferred embodiment.

[0080] FIG. 27 is a diagram of graphs of a pellet spread of a preferred embodiment.

[0081] FIG. 28A is a diagram of a computer implemented method for simulating digital clay targets of a preferred embodiment.

[0082] FIG. 28B is a diagram of an original image captured by an augmented reality system in a preferred embodiment.

[0083] FIG. 28C is a diagram spatial map and anchors in an augmented reality system in a preferred embodiment.

[0084] FIG. 28D is a diagram of a virtual reality simulation in a preferred embodiment.

[0085] FIG. 29A is a diagram of initializing a computer implemented simulation of shooting a digital clay target.

[0086] FIG. 29B is a diagram for calculating a lead distance.

[0087] FIG. 29C is a flowchart of a preferred method of generating a simulation.

[0088] FIG. 29D is a diagram of a spatial map from the system.

[0089] FIG. 29E is a flowchart of a preferred method of generating a simulation.

[0090] FIG. 30 is a diagram control movements in a preferred embodiment.

[0091] FIG. 31 is a flowchart of a method for processing control signals in a preferred embodiment.

[0092] FIG. 32 is a diagram of a preferred embodiment of an augmented reality overlay of a simulation.

[0093] FIG. 33 is a preferred method of generating a phantom target ahead of a live bird target.

[0094] FIG. 34 is a flowchart of a preferred method of a deriving path equations.

[0095] FIG. 35 is a node architecture drawing of a preferred embodiment of a neural network for use with the system.

[0096] FIG. 36 is a node design drawing of a preferred embodiment.

[0097] FIG. 37A is an architecture of an exemplary embodiment of a tactical unit.

[0098] FIG. 37B is an overview of the operation of a preferred embodiment of a system employing a tactical unit.

[0099] FIG. 38A is a preferred embodiment of a system employing multiple remote units.

[0100] FIG. 38B is an overview of a preferred embodiment of the operation of a system employing multiple remote units.

[0101] FIG. 39 is a architecture diagram of a preferred embodiment of a headset module.

[0102] FIG. 40A is a side view of a preferred embodiment of a tactical helmet.

[0103] FIG. 40B is a front view of a preferred embodiment of a tactical helmet.

[0104] FIG. 41A is an architecture diagram of a preferred embodiment of a weapon module.

[0105] FIG. 41B is a drawing of a preferred embodiment of a processor card and memory.

[0106] FIG. 42A is a schematic side view of a weapon used in the system.

[0107] FIG. 42B is a schematic top view of a weapon used in the system.

[0108] FIG. 43A is a method flow chart of a single tactical unit operating in a tactical theatre.

[0109] FIG. 43B is a flow chart of a preferred embodiment of the functions of a plurality of remote units operating in a tactical theatre.

[0110] FIG. 44 is a flow chart of a preferred method for determining weapon position.

[0111] FIGS. 45A, 45B and 45C show examples of the display of a single tactical unit showing a phantom and a pull away lead.

[0112] FIGS. 45D and 45E show exemplary displays of two remote units operating in the same tactical theatre.

[0113] FIG. 46A shows a preferred embodiment of an architecture of an alternate system embodiment.

[0114] FIG. 46B shows an overview of a preferred embodiment of an alternate architecture of the system.

[0115] FIG. 47 shows a preferred embodiment of an architecture of a drone spotter unit.

[0116] FIG. 48 shows a preferred embodiment of an architecture of a fixed camera spotter unit.

[0117] FIG. 49 shows a preferred method of operation of an alternate embodiment of the system.

[0118] FIG. 50 shows a preferred embodiment of a method of target path resolution.

[0119] FIG. 51 is a preferred embodiment of the AI processor.

[0120] FIG. 52 shows a preferred embodiment of a single artificial neural network for predicting a vector component of a lead distance.

[0121] FIG. 53 shows a flow chart of a method for training and using an artificial neural network of a preferred embodiment.

[0122] FIG. 54 shows preferred implementation of a preferred embodiment of a neural network.

#### DETAILED DESCRIPTION OF THE INVENTION

[0123] It will be appreciated by those skilled in the art that aspects of the present disclosure may be illustrated and described herein in any of a number of patentable classes or context including any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof. Therefore, aspects of the present disclosure may be implemented entirely in hardware, entirely in software (including firmware, resident software,

micro-code, etc.) or combining software and hardware implementation that may all generally be referred to herein as a “circuit,” “module,” “component,” or “system.” Further, aspects of the present disclosure may take the form of a computer program product embodied in one or more computer readable media having computer readable program code embodied thereon.

**[0124]** Any combination of one or more computer readable media may be utilized. The computer readable media may be a computer readable signal medium or a computer readable storage medium. For example, a computer readable storage medium may be, but not limited to, an electronic, magnetic, optical, electromagnetic, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples of the computer readable storage medium would include, but are not limited to: a portable computer diskette, a hard disk, a random access memory (“RAM”), a read-only memory (“ROM”), an erasable programmable read-only memory (“EPROM” or Flash memory), an appropriate optical fiber with a repeater, a portable compact disc read-only memory (“CD-ROM”), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. Thus, a computer readable storage medium may be any tangible medium that can contain, or store a program for use by or in connection with an instruction execution system, apparatus, or device.

**[0125]** A computer readable signal medium may include a propagated data signal with computer readable program code embodied therein, for example, in baseband or as part of a carrier wave. The propagated data signal may take any of a variety of forms, including, but not limited to, electromagnetic, optical, or any suitable combination thereof. A computer readable signal medium may be any computer readable medium that is not a computer readable storage medium and that can communicate, propagate, or transport a program for use by or in connection with an instruction execution system, apparatus, or device. Program code embodied on a computer readable signal medium may be transmitted using any appropriate medium, including but not limited to wireless, wireline, optical fiber cable, RF, or any suitable combination thereof.

**[0126]** Computer program code for carrying out operations for aspects of the present disclosure may be written in any combination of one or more programming languages, including an object oriented programming language such as Java, Scala, Smalltalk, Eiffel, JADE, Emerald, C++, C#, VB.NET, Python or the like, conventional procedural programming languages, such as the “C” programming language, Visual Basic, Fortran 2003, Perl, COBOL 2002, PHP, ABAP, dynamic programming languages such as Python, Ruby and Groovy, or other programming languages.

**[0127]** Aspects of the present disclosure are described with reference to flowchart illustrations and/or block diagrams of methods, systems and computer program products according to embodiments of the disclosure. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable instruc-

tion execution apparatus, create a mechanism for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

**[0128]** These computer program instructions may also be stored in a computer readable medium that when executed can direct a computer, other programmable data processing apparatus, or other devices to function in a particular manner, such that the instructions when stored in the computer readable medium produce an article of manufacture including instructions which when executed, cause a computer to implement the function/act specified in the flowchart and/or block diagram block or blocks. The computer program instructions may also be loaded onto a computer, other programmable instruction execution apparatus, or other devices to cause a series of operational steps to be performed on the computer, other programmable apparatuses or other devices to produce a computer implemented process such that the instructions which execute on the computer or other programmable apparatus provide processes for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

**[0129]** Referring to FIG. 6, system 600 includes network 601, simulation administrator 602 connected to network 601, and user device 604 connected to network 601. Simulation administrator 602 is further connected to simulation database 603 for storage of relevant data. For example, data includes a set of target data, a set of weapon data, and a set of environment data.

**[0130]** In one embodiment, network 601 is a local area network. In another embodiment, network 601 is a wide area network, such as the internet. In other embodiments, network 601 includes a combination of wide area networks and local area networks, includes cellular networks.

**[0131]** In a preferred embodiment, user device 604 communicates with simulation administrator 602 to simulation database 603 to generate and project a simulation that includes a target, a phantom, and a phantom halo adjacent to the target as will be further described below.

**[0132]** In another embodiment, simulation administrator 602 generates a simulation that includes a target, a phantom, a phantom halo adjacent to the target, and a weapon image as will be further described below and sends the simulation to user device for projection.

**[0133]** FIG. 1 depicts the general dimensions of a skeet shooting range. Skeet shooting range 100 is a skeet field that includes eight shooter positions with 2 launcher locations. Cameras 150 and 151 are located in positions to view houses 101 and 102 and launchers 103 and 109. Skeet shooting range 100 has high house 101 and low house 102 separated by distance 111. Distance 111 is about 120 feet. Launcher 103 is adjacent high house 101. Launcher 109 is adjacent low house 102. Station 110 is equidistant from high house 101 and low house 102 at distance 112. Distance 112 is about 60 feet. Station 106 is equidistant from high house 101 and low house 102 and generally perpendicular to distance 111 at distance 113. Distance 113 is 45 feet. Station 106 is distance 114 from launcher 103. Distance 114 is about 75 feet. Stations 104 and 105 are positioned along arc 121 between launcher 103 and station 106 at equal arc lengths. Each of arc lengths 122, 123, and 124 is about 27 feet. Stations 107 and 108 are positioned along arc 121 between station 106 and launcher 109 at equal arc lengths. Each of arc lengths 125, 126, and 127 is 26 feet, 8<sup>3</sup>/<sub>8</sub> inches.

[0134] Target flight path 116 extends from high house 101 to marker 117. Marker 117 is positioned about 130 feet from high house 101 along target flight path 116. Target flight path 115 extends from low house 102 to marker 118. Marker 118 is about 130 feet from low house 102 along target flight path 115. Target flight paths 115 and 116 intersect at target crossing point 119. Target crossing point 119 is positioned distance 120 from station 110 and is 15 feet above the ground. Distance 120 is 18 feet. Clay targets are launched from high house 101 and low house 102 along target flight paths 116 and 115, respectively. Marksman 128 positioned at any of stations 104, 105, 106, 107, 108, and 110 and launchers 103 and 109 attempts to shoot and break the launched clay targets.

[0135] FIG. 2 depicts the general dimensions of a trap shooting range. Trap shooting range 200 is a trap field that includes five shooter locations with one launcher location. Cameras 250 and 251 are located in positions to view trap house 202. Once all of the coordinates are set and the field dimensions are known, one good video at a normal lens setting at 60 frames per second (fps) of one trajectory can be used to recreate a trajectory and phantom position from any point of view (POV).

[0136] In a preferred embodiment, cameras 150 and 151 (shown in FIG. 1) and 250 and 251 (shown in FIG. 2) can be used to record many target flights of clay targets from which flight paths may be derived for later use in simulations, as will be later described. In the same way, cameras 150 and 151 and 250 and 251 can be used to record the flight of live targets (such as birds) as they are released from the launch or other locations. Similarly, stereo cameras (such as that described in relation to FIG. 9C) can be used outside a controlled skeet range or trap range to record flight paths of either clay targets or live targets from which mathematical flight paths may be recorded for later use in simulation, as will be further described.

[0137] Referring to FIG. 34, a method storing launch target information in a path table and path array will be described.

[0138] At step 341, the stereo cameras are activated and directed toward the projected flight path of the target.

[0139] At step 342, the target is launched. At step 344, both cameras simultaneously record the flight path of the target.

[0140] At step 346, the synchronized video images from the stereoscopic cameras are analyzed to isolate the target position along the flight path for each time "t". In a preferred embodiment, each of the cameras records approximately 60 frames per second, or 360 frames per minute. In a preferred embodiment, the target positions are stored in cartesian coordinates. The origin of the cartesian coordinate system,  $x=0, y=0, z=0$ , is taken at the launch point of the target. The x-coordinate for each position is derived from the horizontal distance of the target from a launch point. The y-coordinate is derived from altitude of the target as the vertical distance from the ground. The depth, or z-coordinate is derived from the depth function of the stereoscopic cameras and is translated to agree with the origin.

[0141] At step 348, the isolated target positions are stored in a path table.

[0142] At step 350, a spline function available from the 3D unity engine is applied to interpolate path equations from the isolated target positions for each flight recorded. At step 352,

the path equation is stored in a path array indexed by the date and time of the target launch.

[0143] At step 354, a 3 second video sample of the target is recorded and stored in an attribute array, indexed according to date and time of the target launch. Other lengths of video samples can also be used.

[0144] Referring to FIG. 7, simulation administrator 701 includes processor 702, network interface 703 connected to processor 702, and memory 704 connected to processor 702. Simulation application 705 is stored in memory 704 and executed by processor 702. Simulation application 705 includes position application 706, statistics engine 707, and target and phantom generator 708.

[0145] In a preferred embodiment, simulation administrator 701 is a PowerEdge C6100 server and includes a PowerEdge C410x PCIe Expansion Chassis available from Dell Inc. Other suitable servers, server arrangements, and computing devices known in the art may be employed.

[0146] In one embodiment, position application 706 communicates with a position tracker connected to the user device to detect the position of the user device for simulation application 705. Statistics engine 707 communicates with a database to retrieve relevant data and generate renderings according desired simulation criteria, such as desired weapons, environments, and target types for simulation application 705. Target and phantom generator 708 calculates and generates a target along a target path, a phantom target, and a phantom halo for the desired target along a phantom path for simulation application, as will be further described below.

[0147] Referring to FIG. 8, user device 800 includes computer 801 connected to headset 802. Computer 801 is further connected to replaceable battery 803, microphone 804, speaker 805, and position tracker 806.

[0148] Computer 801 includes processor 807, memory 809 connected to processor 807, and network interface 808 connected to processor 807. Simulation application 810 is stored in memory 809 and executed by processor 807. Simulation application 810 includes position application 811, statistics engine 812, and target and phantom generator 813. In a preferred embodiment, position application 811 communicates with position tracker 806 to detect the position of headset 802 for simulation application 810. Statistics engine 812 communicates with a database to retrieve relevant data and generate renderings according desired simulation criteria, such as desired weapons, environments, and target types for simulation application 810. Target and phantom generator 813 calculates and generates a target along a target path, a phantom target, and a phantom halo for the desired target along a phantom path for simulation application 810, as will be further described below.

[0149] Input device 814 is connected to computer 801. Input device 814 includes processor 815, memory 816 connected to processor 815, communication interface 817 connected to processor 815, a set of sensors 818 connected to processor 815, and a set of controls 819 connected to processor 815.

[0150] In one embodiment, input device 814 is a simulated weapon, such as a shot gun, a rifle, or a handgun. In another embodiment, input device 814 is a set of sensors connected to a disabled real weapon, such as a shot gun, a rifle, or a handgun, to detect movement and actions of the real weapon. In another embodiment, input device 814 is a glove

having a set of sensors worn by a user to detect positions and movements of a hand of a user.

[0151] Headset **802** includes processor **820**, battery **821** connected to processor **820**, memory **822** connected to processor **820**, communication interface **823** connected to processor **820**, display unit **824** connected to processor **820**, and a set of sensors **825** connected to processor **820**.

[0152] Referring to FIGS. **9A** and **9B**, a preferred implementation of user device **800** is described as user device **900**. User **901** wears virtual reality unit **902** having straps **903** and **904**. Virtual reality unit **902** is connected to computer **906** via connection **905**. Computer **906** is preferably a portable computing device, such as a laptop or tablet computer, worn by user **901**. In other embodiments, computer **906** is a desktop computer or a server, not worn by the user. Any suitable computing device known in the art may be employed. Connection **905** provides a data and power connection from computer **906** to virtual reality unit **902**.

[0153] Virtual reality unit **902** includes skirt **907** attached to straps **903** and **904** and display portion **908** attached to skirt **907**. Skirt **907** covers eyes **921** and **916** of user **901**. Display portion **908** includes processor **911**, display unit **910** connected to processor **911**, a set of sensors **912** connected to processor **911**, communication interface **913** connected to processor **911**, and memory **914** connected to processor **911**. Lens **909** is positioned adjacent to display unit **910** and eye **921** of user **901**. Lens **915** is positioned adjacent to display unit **910** and eye **916** of user **901**. Virtual reality unit **902** provides a stereoscopic three-dimensional view of images to user **901**.

[0154] User **901** wears communication device **917**. Communication device **917** includes earpiece speaker **918** and microphone **919**. Communication device **917** is preferably connected to computer **906** via a wireless connection such as a Bluetooth connection. In other embodiments, other wireless or wired connections are employed. Communication device **917** enables voice activation and voice control of a simulation application stored in the computer **906** by user **901**.

[0155] In one embodiment, virtual reality unit **902** is the Oculus Rift headset available from Oculus VR, LLC. In another embodiment, virtual reality unit **902** is the HTC Vive headset available from HTC Corporation. In this embodiment, a set of laser position sensors **920** is attached to an external surface virtual reality unit **902** to provide position data of virtual reality unit **902**. In another preferred embodiment, virtual reality unit **902** can take the form of the Magic Leap One headset available from Magic Leap, Inc. of Plantation, Fla., the Oculus S, or Oculus Quest, available from Oculus VR, LLC or the HMD Odyssey from Samsung of San Jose, Calif. Any suitable virtual reality unit or mixed reality unit known in the art may be employed.

[0156] In certain embodiments, set of sensors **912** include sensors related to eye tracking. When the sensors related to eye tracking are based on infrared optical tracking, the set of sensors **912** includes one or more infrared light sources and one or more infrared cameras. Light from the infrared light sources is reflected from one or more surfaces of the user eye and is received by the infrared cameras. The reflected light is reduced to a digital signal which is representative of the positions of the user eye. These signals are transmitted to the computer. Computer **906** and processor **911** then determine the positioning and direction of the eyes of the user and record eye tracking data. With the eye tracking data, com-

puter **906** determines whether the user is focusing on the simulated target or on the phantom target; how quickly a user focusses on the simulated target or phantom target; how long it takes for the user to aim the weapon after focusing on the simulated target or phantom target; how long the user focusses on the simulated target or phantom target before pulling the trigger; how long it takes the user to see and focus on the next target; whether the user's eyes were shut or closed before, during, or after the pull of the trigger; and so on. Computer **906** also determines eye training statistics based on the eye training data and the eye tracking data collected over multiple shots and rounds of the simulation. Feedback is given to the user that includes and is based on the eye tracking data, the eye training data, and the eye training statistics.

[0157] Referring then to FIGS. **9C** and **9D** a preferred implementation of user device **800** is described as mixed reality unit **950**. User **901** wears mixed reality unit **950**. Mixed reality unit **950** is connected to computer **906** via connection **905**. Connection **905** provides data and power connection from computer **906** to processor **954** communication interface **952** and display **958**. Mixed reality unit **950** further comprises visor **956**. Visor **956** operatively supports display **958** in front of the user's eyes. When mixed reality unit **950** is in operation, it includes visual axis **960**. The visual axis is generally coaxial with the pupils of the user. Display **958** displays a stereoscopic view to the user.

[0158] Mixed reality unit **950** further supports stereo camera **925**. Stereo camera **925** incorporates two independent digital cameras, right camera **927** and left camera **929**. The central axis of each of the cameras is parallel with visual axis **960** and is positioned directly in line with one eye of the user. In a preferred embodiment, the digital input from each of right camera **927** and left camera **929** can be displayed on display **958** for viewing by user **901** in near real time.

[0159] In one embodiment, mixed reality unit **950** comprises the Oculus rift headset available from Oculus VR, LLC. In this embodiment, stereo camera **925** is the Ovrvision Pro PV high performance stereo camera USB 3.0 available from Ovrvision of Osaka, Japan. Camera **925** is attached to mixed reality unit **950** by screws or appropriate adhesive. It allows high resolution wide angle viewing with two eye synchronization with appropriately low delay times. In this embodiment, communication with the processor is carried out through the GPIO communications channel which supports game engines such as Unity 5 and the Unreal Engine. In a preferred embodiment, the wide angle lens is capable of supporting a 120° viewing angle, and a delay of 50 microseconds at 60 frames per second.

[0160] In another embodiment, mixed reality unit **950** is the HTC Vive mixed reality headset available from HTC of Taiwan. In this embodiment, stereo camera **925** are the onboard cameras available on the HTC Vive unit are employed in "pass through" mode.

[0161] In yet another embodiment, mixed reality unit **950** is the HMD Odyssey mixed reality headset from Samsung of Seoul, South Korea. In this embodiment, stereo camera **925** is likewise the onboard camera system of the HMD Odyssey system employed in "pass through" mode.

[0162] In certain embodiments, the laser position sensors **920** are light emitting diodes (LEDs) that act as markers that can be seen or sensed by one or more cameras or sensors. Data from the cameras or sensors is processed to derive the location and orientation of virtual reality unit **902** based on

the LEDs. Each LED emits light using particular transmission characteristics, such as phase, frequency, amplitude, and duty cycle. The differences in the phase, frequency, amplitude, and duty cycle of the light emitted by the LEDs allows for a sensor to identify each LED by the LED's transmission characteristics. In certain embodiments, the LEDs on virtual reality unit 902 are spaced with placement characteristics so that there is a unique distance between any two LEDs, which gives the appearance of a slightly randomized placement on virtual reality unit 902. The transmission characteristics along with placement characteristics of the LEDs on virtual reality unit 902 allows the simulation system to determine the location and orientation of virtual reality unit 902 by sensing as few as three LEDs with a camera or other sensor.

[0163] In a preferred embodiment, a simulation environment that includes a target is generated by computer 906. Computer 906 further generates a phantom target and a phantom halo in front of the generated target based on a generated target flight path. The simulation environment including the generated target, the phantom target, and the phantom halo are transmitted from computer 906 to virtual reality unit 902 for viewing adjacent eyes 916 and 921 of user 901, as will be further described below. The user aims a weapon at the phantom target to attempt to shoot the generated target.

[0164] Referring FIG. 10A in one embodiment, simulated weapon 1001 includes trigger 1002 connected to set of sensors 1003, which is connected to processor 1004. Communication interface 1005 is connected to processor 1004 and to computer 1009. Battery 1026 is connected to processor 1004. Simulated weapon 1001 further includes a set of controls 1006 attached to an external surface of simulated weapon 1001 and connected to processor 1004. Set of controls 1006 includes directional pad 1007 and selection button 1008. Battery 1026 is connected to processor 1004. Actuator 1024 is connected to processor 1004 to provide haptic feedback.

[0165] In a preferred embodiment, simulated weapon 1001 is a shotgun. It will be appreciated by those skilled in the art that other weapon types may be employed.

[0166] In one embodiment, simulated weapon 1001 is a Delta Six first person shooter controller available from Avenger Advantage, LLC. In another embodiment, simulated weapon 1001 is an airsoft weapon or air gun replica of a real weapon. In another embodiment, simulated weapon 1001 is a firearm simulator that is an inert detailed replica of an actual weapons, such as "blueguns" from Ring's Manufacturing. Other suitable simulated weapons known in the art may be employed.

[0167] In a preferred embodiment, set of sensors 1003 includes a position sensor for trigger 1002 and a set of motion sensors to detect an orientation of simulated weapon 1001.

[0168] In a preferred embodiment, the position sensor is a Hall Effect sensor. In this embodiment, a magnet is attached to trigger 1002. Other types of Hall Effect sensor or any other suitable sensor type known in the art may be employed.

[0169] In a preferred embodiment, the set of motion sensors is a 9-axis motion tracking system-in-package package sensor, model no. MP11-9150 available from InverSense®, Inc. In this embodiment, the 9-axis sensor combines a 3-axis gyroscope, a 3-axis accelerometer, an

on-board digital motion processor, and a 3-axis digital compass. In other embodiments, other suitable sensors and/or suitable combinations of sensors may be employed.

[0170] Referring to FIGS. 10B, 10C, and 10D in another embodiment, weapon 1010 includes simulation attachment 1011 removably attached to its stock. Simulation attachment 1011 includes on-off switch 1012 and pair button 1013 to communicate with computer 1009 via Bluetooth connection. Any suitable wireless connection may be employed. Trigger sensor 1014 is removably attached to trigger 1022 and in communication with simulation attachment 1011. A set of muzzle sensors 1015 is attached to a removable plug 1016 which is removably inserted into barrel 1023 of weapon 1010. Set of muzzle sensors 1015 include a processor 1017, battery 1018 connected to processor 1017, gyroscope 1019 connected to processor, accelerometer 1020 connected to processor 1017, and compass 1021 connected to processor 1017.

[0171] In one embodiment, set of muzzle sensors 1015 and removable plug 1016 are positioned partially protruding outside of barrel 1023 of weapon 1010.

[0172] In one embodiment, weapon 1010 includes rail 1025 attached to its stock in any position. In this embodiment, set of muzzle sensors 1015 is mounted to rail 1025.

[0173] In one embodiment, weapon 1010 fires blanks to provide live recoil to a user.

[0174] It will be appreciated by those skilled in the art that any weapon may be employed as weapon 1010, including any rifle or handgun. It will be further appreciated by those skilled in the art that rail 1025 is optionally mounted to any type of weapon. Set of muzzle sensors 1015 may be mounted in any position on weapon 1010. Any type of mounting means known in the art may be employed.

[0175] Referring to FIG. 10E, base 1028 comprises a sensor system that includes a magnetic field detector used to determine the location and orientation of a weapon, such as weapon 1010 with removable plug 1016 shown in FIG. 10F. Base 1028 includes processor 1032, which is connected to communication interface 1034, power source 1036, memory 1038, first coil 1040, second coil 1042, and third coil 1044. First coil 1040, second coil 1042, and third coil 1044 form the magnetic field detector of the sensor system of base 1028.

[0176] Processor 1032 of base 1028 receives positioning signals via first coil 1040, second coil 1042, and third coil 1044 that are used to determine the position and orientation of a weapon used in the simulation system. In a preferred embodiment, each of the positioning signals received via first coil 1040, second coil 1042, and third coil 1044 can be differentiated from one another by one or more of each positioning signal's phase, frequency, amplitude, and duty cycle so that each positioning signal transmitted by each coil is distinct. The differences in the positioning signals allow base 1028 to determine the position of a transmitting device, such as removable plug 1016 of FIG. 10F, based on the positioning signals that indicates the relative position between base 1028 and the transmitting device.

[0177] Referring to FIG. 10F, removable plug 1016 is inserted into an under barrel of weapon 1010 and transmits positioning signals used to determine the location and orientation of removable plug 1016 and the weapon removable plug 1016 is connected to. Removable plug 1016 includes processor 1017, which is connected to battery 1018, communication interface 1046, first coil 1048, second coil 1050,

and third coil **1052**. First coil **1048**, second coil **1050**, and third coil **1052** form magnetic field transmitters of a sensor system of removable plug **1016**. The magnetic fields generated and transmitted by first coil **1048**, second coil **1050**, and third coil **1052** are positioning signals used to determine the location and orientation of removable plug **1016**, for example, by base **1028** of FIG. **10E**.

[**0178**] Processor **1017** transmits positioning signals from first coil **1048**, second coil **1050**, and third coil **1052** that are received by processor **1032** of base **1028**. From the transmitted positioning signals, the relative location and orientation between removable plug **1016** and base **1028** is determined so that the precise location of removable plug **1016** with respect to base **1028** is derived. The determinations and derivations may be performed by one or more of processor **1032** of base **1028**, processor **1017** of removable plug **1016**, and a processor of another computer of the simulation system, such as computer **1009**. Once the position of removable plug **1016** is known, the position and orientation of weapon **1010** is determined based on the location and orientation of removable plug **1016**, the geometry of removable plug **1016**, the geometry of weapon **1010**, and the placement of removable plug **1016** on weapon **1010**. With the position and orientation of weapon **1010**, the simulation application can display a simulated version of weapon **1010**, calculate the proper position of a phantom target, and provide suggested adjustments to improve a user's marksmanship.

[**0179**] In an alternative embodiment, the sensor system of base **1028** includes the magnetic field transmitter and the sensor system of removable plug **1016** includes the magnetic field detector. In alternative embodiments, removable plug **1016** includes threading that corresponds to threading with the barrel of the weapon that is commonly used for a shotgun choke and removable plug **1016** is fitted and secured to the barrel of the weapon via the threading.

[**0180**] Referring to FIG. **10G**, removable collar **1054** fits onto barrel **1056** of a weapon, such as weapon **1010** of FIG. **10B**. Removable collar **1054** includes tip **1058** and three members **1060**, **1062**, and **1064**. Members **1060**, **1062**, and **1064** extend from a first side of tip **1058** that touches barrel **1056** when removable collar **1054** is fitted to barrel **1056**. Removable collar **1054** includes light emitting diodes (LEDs), such as LEDs **1066** on member **1060**, LEDs **1068** on member **1062**, and LEDs on member **1064**, and LEDs **1070** on tip **1058**. Removable collar **1054** includes additional LEDs that are occluded on FIG. **10G**, such as on member **1064** and on tip **1058**. The LEDs on removable collar **1054** may emit infrared light to be invisible to a user or may emit light in the visible spectrum. Removable collar **1054** acts as a marker from which the location and orientation of the weapon can be derived.

[**0181**] The LEDs on removable collar **1054** each emit light using particular transmission characteristics, such as phase, frequency, amplitude, and duty cycle. The differences in the phase, frequency, amplitude, and duty cycle of the light emitted by the LEDs allows for a sensor to identify each LED on removable collar **1054** by the LED's transmission characteristics. The LEDs on removable collar **1054** are spaced with placement characteristics so that there is a unique distance between any two LEDs, which gives the appearance of a slightly randomized placement on removable collar **1054**. The transmission characteristics along with placement characteristics of the LEDs on removable collar

**1054** allows the simulation system to determine the location and orientation of the removable plug by sensing as few as three LEDs with a camera or other sensor. Once the location and orientation of removable collar **1054** is determined, the location and orientation of the weapon to which removable collar **1054** is attached is derived based on the known geometries of removable collar **1054** and the weapon, which are stored in a database.

[**0182**] Referring to FIG. **10H**, removable collar **1054** is fitted onto barrel **1056** of a weapon. Inner portions of members **1060**, **1062**, and **1064** are rubberized and may contain an adhesive to prevent movement of removable collar **1054** with respect to the weapon it is attached to. After removable collar **1054** is installed for the first time to a weapon, the simulation system is calibrated to associate the location and orientation, including a roll angle, of removable collar **1054** to the location and orientation of the weapon.

[**0183**] In alternative embodiments, the portion of removable collar **1054** that fits against the barrel of the weapon is shaped to fit with only one orientation with respect to the weapon. The removable collar **1054** may include additional members that fit around the iron sight of the weapon so that there is only one possible fitment of removable collar **1054** to the weapon and the process of calibration can be reduced or eliminated.

[**0184**] Referring to FIG. **10I**, removable collar **1054** is fitted to weapon **1010**. Weapon **1010** is an over-under shotgun with barrel **1056**, under barrel **1057**, and top rail **1059**. Removable collar **1054** comprises a hollow portion **1055** that allows for the discharge of live or blank rounds of ammunition during the simulation. A front surface of removable collar **1054** is flush with the front surfaces of under barrel **1057** so that the position of removable collar **1054** with respect to each of barrels **1056** and **1057** is known and the trajectory of shots from weapon **1010** can be properly simulated. Removable collar **1054** includes hollow portion **1055**, member **1061**, mounting screws **1063**, battery **1018**, processor **1017**, and LEDs **1067**. Removable collar **1054** is customized to the particular shape of weapon **1010**, which may include additional iron sights. Removable collar **1054** does not interfere with the sights of weapon **1010** so that weapon **1010** can be aimed normally while removable collar **1054** is fitted to weapon **1010**.

[**0185**] Member **1061** is a flat elongated member that allows for removable collar **1054** to be precisely and tightly fitted to the end of under barrel **1057** of weapon **1010** after removable collar **1054** is slid onto the end of under barrel **1057**. Member **1061** with mounting screws **1063** operate similar to a C-clamp with mounting screws **1063** pressing into member **1061** and thereby securing removable collar **1054** to the end of under barrel **1057** with sufficient force so that the position and orientation of removable collar **1054** with respect to weapon **1010** is not altered by the firing of live rounds or blank rounds of ammunition with weapon **1010**.

[**0186**] Battery **1018** is connected to and powers the electrical components within removable collar **1054** including processor **1017** and LEDs **1067**. Processor **1017** controls LEDs **1067**. In additional embodiments removable collar **1054** includes one or more, accelerometers, gyroscopes, compasses, and communication interfaces connected to processor **1017**. The sensor data from the accelerometers, gyroscopes, and compasses is sent from removable collar **1054** to computer **1009** via the communication interface.

Removable collar **1054** includes button **1069** to turn on, turn off, and initiate the pairing of removable collar **1054**.

[**0187**] LEDs **1067** emit light that is sensed by one or more cameras or sensors, from which the locations and orientations of removable collar **1054** and weapon **1010** can be determined. The locations and orientations are determined from the transmission characteristics of the light emitted from LEDs **1067**, and the placement characteristics of LEDs **1067**.

[**0188**] Weapon **1010**, to which removable collar **1054** is fitted, is loaded with one or more live or blank rounds of ammunition that discharge through the hollow portion **1055** of removable collar **1054** when a trigger of weapon **1010** is pulled so that blank rounds or live rounds of ammunition can be used in conjunction with the simulation. Using blank rounds or live rounds with the simulation allows for a more accurate and realistic simulation of the shooting experience, including the experience of re-aiming weapon **1010** for a second shot after feeling the kickback from the discharge of a blank or live round from a first shot.

[**0189**] In alternative embodiments, the weapon is a multiple shot weapon, such as an automatic rifle, a semi-automatic shotgun, or a revolver. With a multiple shot weapon the simulation experience includes the feeling of the transition between shots, such as the cycling of the receiver of a semi-automatic shotgun. When the weapon comprises an automatic or semi-automatic receiver, the simulation displays the ejection of a spent shell casing that may not correspond to the actual path or trajectory of the actual spent shell casing. Additional embodiments track the location of the spent shell casing as it is ejected and match the location and trajectory of the simulated shell casing to the location and trajectory of the spent shell casing. Additional embodiments also include one or more additional sensors, electronics, and power supplies embedded within the housing of removable collar **1054**.

[**0190**] Referring to FIG. **10J**, weapon **1072** is adapted for use in a simulation by the fitment of removable collar **1054** to the barrel of weapon **1072**. Weapon **1072** is a try gun that includes a stock **1074** with adjustable components to fit users of different heights and statures. Each component may include electronic sensors that measure the length, angle, or position of the component so that weapon **1072** can be properly displayed in a simulation.

[**0191**] Stock **1074** of weapon **1072** includes comb **1076** with comb angle adjuster **1078** and comb height adjuster **1080**. Comb **1076** rests against a cheek of a user to improve stability of weapon **1072** during use. The height of comb **1076** is adjustable via manipulation of comb height adjuster **1080**. The angle of comb **1076** is adjustable via manipulation of comb angle adjuster **1078**.

[**0192**] Stock **1074** of weapon **1072** also includes butt plate **1082** with butt plate angle adjuster **1084** and trigger length adjuster **1086**. Trigger length **1088** is the length from trigger **1090** to butt plate **1082**. Butt plate **1082** rests against a shoulder of a user to improve stability of weapon **1072** during use. Trigger length **1088** from butt plate **1082** to trigger **1090** is adjustable via manipulation of trigger length adjuster **1086**. The angle of butt plate **1082** is adjustable via manipulation of butt plate angle adjuster **1084**.

[**0193**] When weapon **1072** used in a virtual reality simulation system with removable collar **1054**, suggested adjustments to comb **1076** and butt plate **1082** are optionally provided. If shots are consistently to the right or left of an

ideal shot placement for a right handed shooter, it may be suggested to increase or decrease trigger length **1088**, respectively. If shots are consistently above or below the ideal shot placement, it may be suggested to decrease or increase the height of comb **1076**, respectively.

[**0194**] Referring to FIG. **10K**, an alternative embodiment of trigger sensor **1014** is shown. Weapon **1010** includes trigger **1022** and trigger guard **1027**. Trigger sensor **1014** is specially shaped and contoured to fit securely to the front of trigger guard **1027**. Once trigger sensor **1014** is slid onto trigger guard **1027**, screws **1041** are tightened to further secure trigger sensor **1014** to trigger guard **1027** and weapon **1010**.

[**0195**] Pull ring **1029** is connected to string **1030**, which winds upon spindle **1031**. Spindle **1031** includes spring **1033**, which keeps tension on string **1030** and biases pull ring **1029** to be pulled away from trigger **1022** and towards trigger guard **1027** and trigger sensor **1014**. In the resting state, there is no slack in string **1030** and pull ring **1029** rests against trigger sensor **1014**.

[**0196**] Sensor **1035** provides data indicative of the rotation and/or position of spindle **1031**. In one preferred embodiment, sensor **1035** is a potentiometer that is connected to and turns with spindle **1031**, where a voltage of the potentiometer indicates the position of spindle **1031** and a change in voltage indicates a rotation of spindle **1031**. In another preferred embodiment, sensor **1035** includes one or more photo emitters and photo detectors that surround an optical encoder wheel that is attached to spindle **1031**, where light from the photo emitters passes through the encoder wheel to activate certain photo detectors to indicate the position of spindle **1031**.

[**0197**] Controller **1037** receives data from sensor **1035** to determine the state of trigger sensor **1014** and communicates the state of trigger sensor **1014** by controlling the output of LED **1039** to create a coded signal that corresponds to the state of trigger sensor **1014**. In a preferred embodiment, the states of trigger sensor **1014** include: pull ring not engaged, pull ring engaged but trigger not pulled, pull ring engaged and trigger is pulled. Controller **1037**, LED **1039**, and sensor **1035** are powered by battery **1043**.

[**0198**] The state of trigger sensor **1014** is communicated by controlling the output LED **1039** with controller **1037**. The output of LED **1039** forms a coded signal to indicate the state of trigger sensor **1014** and can also be used to aid in the determination of the position and orientation of weapon **1010** when the position of trigger sensor **1014** with respect to weapon **1010** and the geometry of weapon **1010** are known. The output of LED **1039** is cycled on and off to flash with a particular phase, frequency, amplitude, and duty cycle that form a set of output characteristics. Different output characteristics are used to indicate different states of trigger sensor **1014**. A first set of output characteristics or first code is used to indicate the pull ring not engaged state, a second set of output characteristics or second code is used to indicate the pull ring engaged but trigger not pulled state, and a third set of output characteristics or third code is used to indicate the pull ring engaged and trigger is pulled state. In one embodiment, the pull ring not engaged state is indicated by a set of output characteristics where the duty cycle is 0% and/or the amplitude is 0 so that LED **1039** does not turn on. An external sensor or camera, such as one of position trackers **1205**, **1206**, and **1215** can be used to determine the state of trigger sensor **1014** by detecting the



output from LED 1039 and decoding the output characteristics to determine which state trigger sensor 1014 is in.

[0199] In an alternative embodiment, pull ring 1029 and string 1030 each include conductive material, trigger sensor 1014 includes a pull-up resistor connected to an input of controller 1037, and controller 1037 is electrically grounded to trigger guard 1027. When trigger 1022 and trigger guard 1027 are electrically connected and conductive pull ring 1029 is touched to trigger 1022, the pull-up resistor is grounded to change the state of the input of controller 1037 so that controller 1037 can determine whether pull ring 1029 is touching trigger 1022. Assuming that the user only touches pull ring 1029 to trigger 1022 when attempting to pull trigger 1022, the determination of whether pull ring 1029 is touching trigger 1022 can be used to indicate that the trigger has been pulled, which is communicated by changing the output coding of LED 1039.

[0200] Referring to FIGS. 11A, 11B, and 11C, different types and styles of sights may be used on weapons used with the simulation. Additionally, the simulation may display a sight on a weapon that is different from the sight actually on the weapon to allow different types of sights to be tested. In alternative embodiments, the halo around the phantom target can be adjusted to match or include the sight profile of the sight being used on the weapon.

[0201] In FIG. 11A, weapon 1102 includes iron sight 1104. Iron sight 1104 comprises two components, one proximate to the tip of the barrel of weapon 1102 and one distal to the tip of weapon 1102, that when aligned indicate the orientation of weapon 1102 to a user of weapon 1102.

[0202] In FIG. 11B, weapon 1102 includes reflex sight 1106, also referred to as a red-dot sight, which may be in addition to an iron sight on weapon 1102. Reflex sight 1106 is mounted on the barrel of weapon 1102 and includes sight profile 1108 shown as a dot. Sight profile 1108 may take any size, shape, color, or geometry and may include additional dots, lines, curves, and shapes of one or more colors. A user can only see the sight profile 1108 when the head of the user is properly positioned with respect to reflex sight 1106.

[0203] In FIG. 11C, weapon 1102 includes holographic sight 1110, which may be in addition to an iron sight. Holographic sight 1110 is mounted to the receiver of weapon 1102 and includes sight profile 1112 shown as a combination circle with dashes. Sight profile 1112 may take any size, shape, color, or geometry and may include additional dots, lines, curves, and shapes of one or more colors. A user can only see the sight profile 1112 when the head of the user is properly positioned with respect to holographic sight 1110.

[0204] Referring to FIG. 12, in simulation environment 1200, user 1201 wears user device 1202 connected to computer 1204 and holds weapon 1203. Each of position trackers 1205, 1206, and 1215 is connected to computer 1204. Position tracker 1205 has field of view 1207. Position tracker 1206 has field of view 1208. Position tracker 1215 has field of view 1216. User 1201 is positioned in fields of view 1207, 1208, and 1216.

[0205] In one embodiment, weapon 1203 is a simulated weapon. In another embodiment, weapon 1203 is a real weapon with a simulation attachment. In another embodiment, weapon 1203 is a real weapon and user 1201 wears a set of tracking gloves 1210. In other embodiments, user 1201 wears the set of tracking gloves 1210 and uses the simulated weapon or the real weapon with the simulation attachment.

[0206] In a preferred embodiment, each of position trackers 1205, 1206, and 1215 is a near infrared CMOS sensor having a refresh rate of 60 Hz. Other suitable position trackers known in the art may be employed. For example, position trackers 1205, 1206, and 1215 can be embodiments of base 1028 of FIG. 10E.

[0207] In a preferred embodiment, position trackers 1205, 1206, and 1215 capture the vertical and horizontal positions of user device 1202, weapon 1203 and/or set of tracking gloves 1210. For example, position tracker 1205 captures the positions and movement of user device 1202 and weapon 1203, and/or set of tracking gloves 1210 in the y-z plane of coordinate system 1209 and position tracker 1206 captures the positions and movement of user device 1202 and weapon 1203 and/or set of tracking gloves 1210 in the x-z plane of coordinate system 1209. Further, a horizontal angle and an inclination angle of the weapon are tracked by analyzing image data from position trackers 1205, 1206, and 1215. Since the horizontal angle and the inclination angle are sufficient to describe the aim point of the weapon, the aim point of the weapon is tracked in time.

[0208] In a preferred embodiment, computer 1204 generates the set of target data includes a target launch position, a target launch angle, and a target launch velocity of the generated target. Computer 1204 retrieves a set of weapon data based on a desired weapon, including a weapon type e.g., a shotgun, a rifle, or a handgun, a set of weapon dimensions, a weapon caliber or gauge, a shot type including a load, a caliber, a pellet size, and shot mass, a barrel length, a choke type, and a muzzle velocity. Other weapon data may be employed. Computer 1204 further retrieves a set of environmental data that includes temperature, amount of daylight, amount of clouds, altitude, wind velocity, wind direction, precipitation type, precipitation amount, humidity, and barometric pressure for desired environmental conditions. Other types of environmental data may be employed.

[0209] Position trackers 1205, 1206, and 1215 capture a set of position image data of user device 1202, weapon 1203 and/or set of tracking gloves 1210 and the set of images is sent to computer 1204. In different preferred embodiments, the position trackers can include a light detection and ranging (LIDAR) system, a radio beacon system or a real time locating system such as an ultra-sonic ranging system (US-RTLS), ultra-wide band (UWB) or wide-over-narrow band wireless local area network, (WLAN, WiFi) Bluetooth system. Sensors in user device 1202, weapon 1203 and/or set of tracking gloves 1210 detect a set of orientation data and sends the set of orientation data to computer 1204. Computer 1204 then calculates a generated target flight path for the generated target based on the set of target data, the set of environment data, and the position and orientation of the user device 1202. The position and orientation of the user device 1202, the weapon 1203 and/or set of tracking gloves 1210 are determined from the set of position image data and the set of orientation data. Computer 1204 generates a phantom target and a phantom halo based on the generated target flight path and transmits the phantom target and the phantom halo to user device 1202 for viewing by user 1201. User 1201 aims weapon 1203 at the phantom target and the phantom halo to attempt to hit the generated target. Computer 1204 detects a trigger pull on weapon 1203 by a trigger sensor and/or a finger sensor and determines a hit or a miss of the generated target based on the timing of the trigger pull, the set of weapon data, the position and orientation of

user device **1202**, weapon **1203**, and/or set of tracking gloves **1210**, the phantom target, and the phantom halo.

[**0210**] In an alternative embodiment, the set of gloves is replaced by a thimble worn on the trigger finger of the shooter and a simulation attachment on the weapon. The simulation attachment on the weapon indicates the position and direction of the weapon and the trigger finger thimble is used to indicate when the trigger is pulled. The positions of the simulation attachment and the thimble are tracked by position trackers **1205**, **1206**, and **1215**. When the user provides a “pull” command, such as by vocalizing the word “pull” that is picked up via voice recognition, the system launches a target and arms the trigger finger thimble, so that when sufficient movement of the thimble relative to the weapon is detected, the system will identify the trigger as being pulled and fire the weapon in the simulation. When the thimble is not armed, movement of the thimble with respect to the weapon is not used to identify if the trigger has been pulled.

[**0211**] When weapon **1203** is loaded with live or blank rounds of ammunition, the discharge of the live or blank rounds of ammunition are detected by one or more sensors, such as a microphone, of user device **1202**. When the discharge of a live or blank round of ammunition is detected and weapon **1203** is a multi-shot weapon that includes a receiver that cycles between shots, the simulation displays the cycling of the receiver after the discharge of the live or blank round of ammunition is detected. When weapon **1203** is a revolver, the simulation displays the rotation of the cylinder. When the system detects the discharge of a number of rounds of live or blank ammunition that is equal to the maximum number of rounds that can be stored in weapon **1203**, the system provides an indication to the user, via user device **1202**, that it is time to reload weapon **1203**.

[**0212**] Referring to FIG. **13**, command menu **1300** includes simulation type **1301**, weapon type **1302**, weapon options **1312**, ammunition **1303**, target type **1304**, station select **1305**, phantom toggle **1306**, day/night mode **1307**, environmental conditions **1308**, freeze frame **1309**, instant replay **1310**, and start/end simulation **1311**. Simulation type **1301** enables a user to select different types of simulations. For example, the simulation type includes skeet shooting, trap shooting, sporting clays, and hunting. Weapon type **1302** enables the user to choose from different weapon types and sizes. Weapon types include shot guns, rifles, handguns, airsoft weapons, air guns, and so on. Weapon sizes include the different calibers or gauges for the weapon’s type. The user further enters a weapon sensor location, for example, in the muzzle or on a rail, and whether the user is right or left handed. Weapon options **1312** enables the user to select different weapon options relating the weapon selected via weapon type **1302**. Weapon options **1312** include optional accessories that can be mounted to the weapon, such as tactical lights, laser aiming modules, forward hand grips, telescopic sights, reflex sights, red-dot sights, iron sights, holographic sights, bipods, bayonets, and so on, including iron sight **1104**, reflex sight **1106**, and holographic sight **1110** of FIG. **11**. Weapon options **1312** also include one or more beams to be simulated with the weapon, such as beams **1906**, **1912**, **1916**, **1920**, **1924**, **1928**, **1932**, and **1936** of FIG. **19**, which show an approximated trajectory of a shot and are optionally adjusted for one or more of windage and gravity. Ammunition **1303** enables the user to select different types of ammunition for the selected weapon type. Target type

**1304** enables the user to select different types of targets for the simulation, including clay targets, birds, rabbits, drones, helicopters, airplanes, and so on. Each type of target includes a target size, a target color, and a target shape. Station select **1305** enables the user to choose different stations to shoot from, for example, in a trap shooting range, a skeet shooting range, a sporting clays course, or a field. The user further selects a number of shot sequences for the station select. In a preferred embodiment, the number of shot sequences in the set of shot sequences is determined by the type of shooting range used and the number of target flight path variations to be generated. For example, the representative number of shot sequences for a skeet shooting range is at least eight, one shot sequence per station. More than one shot per station may be utilized.

[**0213**] In a preferred embodiment, each simulation type **1301** is associated with one or more animated virtual reality shooting scenarios. As one example, when simulation type **1301** is hunting, the animated virtual reality shooting scenario includes a scenario for learning how to shoot over dogs. The shooting over dogs scenario displays an animated dog going on point as a part of the hunt in the simulation so that the user can learn to shoot the target and avoid shooting the dog.

[**0214**] Phantom toggle **1306** allows a user to select whether to display a phantom target and a phantom halo during the simulation. The user further selects a phantom color, a phantom brightness level, and a phantom transparency level.

[**0215**] In certain embodiments, phantom toggle **1306** includes additional help options that adjust the amount of “help” given to the user based on how well the user is doing, such as with aim sensitive help and with dynamic help. When aim sensitive help is selected, aim sensitive help is provided that adjusts one or more of the transparency, color, and size of one or more beams from weapon options **1312**, phantom targets, and halos based on how close the aim point of the weapon is to a phantom target. With aim sensitive help, the beams, phantom targets, and halos are displayed with less transparency, brighter colors, and larger sizes the further off-target the aim point of the weapon is. Conversely, the beams, phantom targets, and halos are displayed with more transparency, darker colors, and smaller sizes when the weapon is closer to being aimed on-target.

[**0216**] When dynamic help is selected, the amount of help provided to the user for each shot is adjusted dynamically based on how well the user is performing with respect to one or more of each shot, each round, and the simulation overall. When more help is provided, beams, phantom targets, and halos are given more conspicuous characteristics and, conversely, when less help is provided, the beams, phantom targets, and halos are shown more passively or not at all. The amount of help is dynamic in that when the previous one or more shots hit the target, a lesser amount of help is provided on the next one or more shots and, conversely, when the previous one or more shots did not hit the target, more help is provided for the subsequent one or more shots. As the user’s skill level advances, the brightness of the phantom target can diminish until it is transparent—the user has learned correct lead by rote repetition and no longer needs the phantom as a visual aide.

[**0217**] Day/night mode **1307** enables the user to switch the environment between daytime and nighttime. Environmental conditions **1308** enables the user to select different

simulation environmental conditions including temperature, amount of daylight, amount of clouds, altitude, wind velocity, wind direction, precipitation type, precipitation amount, humidity, and barometric pressure. Other types of environmental data may be employed. Freeze frame 1309 allows the user to “pause” the simulation. Instant replay 1310 enables the user replay the last shot sequence including the shot attempt by the user. Start/end simulation 1311 enables the user to start or end the simulation. In one embodiment, selection of 1301, 1302, 1312, 1303, 1304, 1305, 1306, 1307, 1308, 1309, 1310, and 1311 is accomplished via voice controls. In another embodiment, selection of 1301, 1302, 1312, 1303, 1304, 1305, 1306, 1307, 1308, 1309, 1310, and 1311 is accomplished via a set of controls on a simulated weapon as previously described.

[0218] Referring to FIG. 14, runtime method 1400 for a target simulation will be described. At step 1401, a baseline position and orientation of the user device and a baseline position and orientation of the weapon are set. In this step, the computer retrieves a set of position image data from a set of position trackers, a set of orientation data from a set of sensors in the user device, the weapon and/or a set of gloves and saves the current position and orientation of the user device and the weapon into memory. Based on the simulation choice, the virtual position of the launcher relative to the position and orientation of the user device is also set. If the user device is oriented toward the virtual location of the launcher, a virtual image of the launcher will be displayed. At step 1402, a set of target flight data, a set of environment data, and a set of weapon data are determined from a set of environment sensors and a database.

[0219] In a preferred embodiment, the set of weapon data is downloaded and saved into the database based on the type of weapon that is in use and the weapon options selected to be used with the weapon. In a preferred embodiment, the set of weapon data includes a weapon type e.g., a shotgun, a rifle, or a handgun, a weapon caliber or gauge, a shot type including a load, a caliber, a pellet size, and shot mass, a barrel length, a choke type, and a muzzle velocity. Other weapon data may be employed. In a preferred embodiment, the weapon options include one or more accessories and beams, including iron sight 1104, reflex sight 1106, and holographic sight 1110 of FIG. 11, and including beams 1906, 1912, 1916, 1920, 1924, 1928, 1932, and 1936 of FIG. 19.

[0220] In a preferred embodiment, the set of environment data is retrieved from the database and includes a wind velocity, an air temperature, an altitude, a relative air humidity, and an outdoor illuminance. Other types of environmental data may be employed.

[0221] In a preferred embodiment, the set of target flight data is retrieved from the database based on the type of target in use. In a preferred embodiment, the set of target flight data includes a launch angle of the target, an initial velocity of the target, a mass of the target, a target flight time, a drag force, a lift force, a shape of the target, a color of the target, and a target brightness level. In alternative embodiments, the target is a self-propelled flying object, such as a bird or drone, which traverses the simulated environment at a constant air speed.

[0222] At step 1403, the target and environment are generated from the set of target flight data and the set of environmental data. At step 1404, a virtual weapon image that includes the selected weapon options is generated and

saved in memory. In this step, images and the set of weapon data of the selected weapon and the selected weapon options for the simulation is retrieved from the database. At step 1405, the target is launched and the target and environment are displayed in the user device. In a preferred embodiment, a marksman will initiate the launch with a voice command such as “pull.”

[0223] At step 1406, a view of the user device with respect to a virtual target launched is determined, as will be further described below.

[0224] At step 1407, a phantom target and a phantom halo are generated based on a target path and the position and orientation of the user, as will be further described below. The target path is determined from the target position the target velocity using Eqs. 1-4. At step 1408, the generated phantom target and the generated phantom halo are sent to the user device and displayed, if the user device is oriented toward the target path. The generated weapon is displayed with the selected weapon options if the user device is oriented toward the position of the virtual weapon or the selected weapon options.

[0225] At step 1409, whether the trigger on the weapon has been pulled is determined from a set of weapon sensors and/or a set of glove sensors. In one preferred embodiment with the trigger sensor of FIG. 10K, the determination of whether the trigger is pulled is made responsive to detecting one of the codes that correspond to the state of trigger sensor 1014 from the output of LED 1039 by a sensor, such as one of position trackers 1205, 1206, and 1215 of FIG. 12.

[0226] If the trigger has not been pulled, then method 1400 returns to step 1405. If the trigger has been pulled, then method 1400 proceeds to step 1410.

[0227] At step 1410, a shot string is determined. In this step, a set of position trackers capture a set of weapon position images. In this step, a set of weapon position data is received from a set of weapon sensors. The shot string is calculated by:

$$A_{shot\ string} = \pi R_{string}^2 \quad \text{Eq. 7}$$

$$R_{string} = R_{initial} + v_{spread} t \quad \text{Eq. 8}$$

where  $A_{shot\ string}$  is the area of the shot string,  $R_{string}$  is the radius of the shot string,  $R_{initial}$  is the radius of the shot as it leaves the weapon,  $v_{spread}$  is the rate at which the shot spreads, and  $t$  is the time it takes for the shot to travel from the weapon to the target. An aim point of the weapon is determined from the set of weapon position images and the set of weapon position data. A shot string position is determined from the position of the weapon at the time of firing and the area of the shot string.

[0228] At step 1411, if the user device is oriented along the muzzle of the weapon, the shot string is displayed on the user device at the shot string position. Separately, a gunshot sound is played and weapon action is displayed. Weapon action is based on the type of the weapon and includes the display of mechanical movements of the weapon, such as the movement of a semi-automatic receiver and the strike of a hammer of the weapon.

[0229] At step 1412, whether the phantom target has been “hit” is determined. The simulation system determines the position of the shot string, as previously described. The simulation system compares the position of the shot string to the position of the phantom target. The shot string is optionally displayed as an elongated cloud of any color that moves from the tip of the user device towards the shot

location, which, ideally, is the target and provides visual feedback to the user of the path taken by the shot string. When the elongated cloud is close to the user device shortly after firing, the diameter of the elongated cloud is about one inch. When the elongated cloud is close to the target, about twenty five yards away from the user, the diameter of the cloud has expanded linearly to about twenty five inches.

[0230] If the position of the shot string overlaps the position of the phantom target, then the phantom target is “hit.” If the position of the shot string does not overlap the phantom target, then the phantom target is “missed.”

[0231] If the phantom target is hit and the user device is oriented toward the hit location, then method 1400 displays an animation of the target being destroyed on the user device at the appropriate coordinates and plays a sound of the target being destroyed at step 1413. At step 1414, the simulation system records a “hit” in the database.

[0232] If a “miss” is determined at step 1412, then method 1400 proceeds to step 1415. At step 1415, whether the phantom halo is hit is determined. In this step, whether the shot string overlaps an area of the phantom halo by a percentage greater than or equal to a predetermined percentage is determined. For example, the predetermined percentage is 50%. Whether the shot string overlaps at least 50% of the area of the phantom halo is determined. Any predetermined percentage may be employed.

[0233] If the position of the shot string overlaps the phantom halo by a percentage greater than or equal to the predetermined percentage, then a “hit” is determined and method 1400 proceeds to step 1413, where the target hit is displayed.

[0234] If at step 1415, the shot string does not overlap the area of the phantom halo by a percentage greater than or equal to the predetermined percentage, then a “miss” is determined and the simulation system records a “miss” in the database at step 1416.

[0235] The number of targets that are hit, the number of targets that are missed, the location of each shot with respect to the phantom target, and the location of the shot string with respect to the trajectory of the target are generated to form tracking data. The tracking data is analyzed to provide insights and suggested adjustments for how to improve the user’s performance with the simulation system.

[0236] At step 1417, whether an end command has been received to complete the simulation is determined. If not received, then method 1400 advances to the next target at step 1418.

[0237] If an end command has been received and the simulation is complete, then a trend of shot attempts is analyzed at step 1419 by retrieving a number of “hits” in the set of shot sequences and a number of “misses” in the set of shot sequences from the database. In this step, a shot improvement is determined by evaluating the number of hits in the set of shot sequences and the number of misses in the set of shot sequences. Method 1400 ends at step 1420.

[0238] Referring to FIG. 15A, user 1500 wears user device 1501 and holds weapon 1502 in simulation environment 1503. Simulation environment 1503 is a virtual sphere spanning 360° in all directions surrounding user 1500. User device 1501 has field of view 1504. Field of view 1504 is a cone that has angular range  $\alpha$  and spans an arcuate portion (in two dimensions) or a sectorial portion (in three dimensions) of simulation environment 1503. User device orientation vector 1505 bisects field of view 1504 and angular

range  $\alpha$  into equal angles  $\beta$ . Weapon 1502 has weapon orientation vector 1506. Each of user device orientation vector 1505 and weapon orientation vector 1506 is independent of each other. The positions of user device 1501, weapon 1502, user device orientation vector 1505, and weapon orientation vector 1506 have Cartesian x,y,z coordinates. Simulation environment 1503 has spherical coordinates. Simulation environment 1503 includes virtual target launcher 1507, virtual target 1508, phantom target 1509 and phantom halo 1510. As can be seen, weapon 1502, virtual target 1508, phantom target 1509, and phantom halo 1510 are in field of view 1504 of user device 1501. Virtual target launcher 1507 is not in field of view 1504 of user device 1501. Weapon 1502, virtual target 1508, phantom target 1509 and phantom halo 1510 will be displayed in user device 1501 and virtual target launcher 1507 will not be displayed in user device 1501.

[0239] In a preferred embodiment, angular range  $\alpha$  is approximately 110° and each of equal angles  $\beta$  is approximately 55°. Other angular ranges may be employed.

[0240] Referring to FIG. 15B, step 1406 will be further described as method 1511 for determining a view for a user device with respect to a position and an orientation of the user device and the weapon. Method 1511 begins at step 1512. At step 1513, a set of current position image data is retrieved from a set of position trackers and a set of current position and orientation data is retrieved from the user device and the weapon and/or set of gloves. At step 1514, a set of motion detection data is received from a set of sensors in the user device to determine movement of the user device and from the weapon and/or set of gloves to determine movement of the weapon. At step 1515, the set of motion detection data and the position of the user device and the weapon and/or set of gloves are combined to determine an x, y, z position of the user device and the weapon and a roll, pitch, and yaw or detection of the user device and the weapon. The current x, y, z orientation vectors for the user device and the weapon are calculated from the difference between the baseline position and orientation and the current position and orientation of the user device and the weapon. The set of motion detection data received is the roll, pitch, and yaw orientation movement of the head of the user and the weapon. At step 1516, the current positions and orientation vectors of the user device and the weapon are mapped to the simulation environment. In a preferred embodiment, the current positions and orientation vectors are a 1:1 ratio to the positions and orientation vectors in the simulation environment. For example, for every inch and/or degree that the user device and/or the weapon moves and/or rotates, the view of the user and/or the simulated weapon moves one inch and/or rotates one degree in the simulated environment. Other ratios may be employed. The mapping determines the display view, as will be further described below. At step 1517, the simulation environment that would be visible to the user based on the orientation of the user device and the weapon is displayed. Method 1500 ends at step 1518.

[0241] Referring to FIG. 15C, step 1516 will be further described as method 1519 for mapping the position and orientation of the user device and the weapon to the simulation environment for determining a display field of view. At step 1520, the x, y, z positions of the weapon and the weapon orientation vector are retrieved. At step 1521, the x, y, z positions of the weapon and the weapon orientation vector are converted to spherical coordinates ( $r, \theta, \phi$ ) using:

$$r = \sqrt{x^2 + y^2 + z^2} \tag{Eq. 9}$$

$$\theta = \arccos\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right) \tag{Eq. 10}$$

$$\varphi = \arctan\left(\frac{y}{x}\right) \tag{Eq. 11}$$

[0242] At step 1522, the weapon is rendered in the simulation environment at the spherical position and orientation vector. At step 1523, the x, y, z positions of the user device and the user device orientation vector are retrieved. At step 1524, the x, y, z positions of the user device and the user device orientation vector are converted to spherical coordinates (r, θ, φ) using Eqs. 9, 10, and 11. At step 1525, the display field of view is determined from the spherical orientation vector coordinates. In this step, equal angles β are measured from the user device orientation vector to define the display field of view as a sector of the simulation environment in spherical coordinates. At step 1526, the field of view sector is compared to the simulation environment to determine a portion of the simulation environment within the field of view sector. At step 1527, the portion of the simulation environment within the field of view sector is displayed on the user device as the display field of view. At step 1528, the spherical position and orientation vector of the weapon is compared to the field of view sector to determine whether the weapon is in the display field of view. If the weapon is not in the display field of view, then method 1519 returns to step 1520. If the weapon is in the display field of view, then at step 1529, the weapon is displayed on the user device at the spherical position and orientation. Method 1519 then returns to step 1520.

[0243] Referring to FIG. 16A, step 1407 will be further described as method 1600 for generating a phantom target and a phantom halo. At step 1601, a phantom path is extrapolated. Referring to FIGS. 16B and 16C, target 1606 is launched from launch point 1611 and moves along target path 1607 at position P<sub>1</sub>. Phantom target 1608 moves along phantom path 1609 ahead of target 1606 at position P<sub>2</sub>. Position P<sub>2</sub> is lead distance 1610 and drop distance 1616 from position P<sub>1</sub>. Phantom path 1609 varies as target 1606 and target path 1607 varies, thereby varying lead distance 1610. Marksman 1612 is positioned at distance 1613 from launch point 1611. Marksman 1612 aims at phantom target 1608 and shoots along shot path 1614 to intercept target 1606. Target path 1607 is extrapolated over time using the set of target flight data. Target path 1607 is calculated using Eqs. 1-4.

[0244] Referring to FIG. 16B, lead distance 1610 is calculated using target path 1607, the relative marksman location, and the set of weapon data.

$$D_{P_2} \approx \frac{D_{S_2} \tan \varphi_2}{\cos \theta \tan \varphi_2 - \sin \theta} \tag{Eq. 12}$$

$$D_{P_1} \approx \frac{D_{S_1} \tan \varphi_1}{\cos \theta \tan \varphi_1 - \sin \theta} \tag{Eq. 13}$$

where D<sub>P<sub>2</sub></sub> is the distance of phantom target 1608 at position P<sub>2</sub> from launch point 1611, D<sub>S<sub>2</sub></sub> is the distance from marksman 1612 to phantom target 1608 along shot path 1614, φ<sub>2</sub>

is the angle between shot path 1614 and distance 1613, θ is the launch angle between target path 1607 and distance 1613, D<sub>P<sub>1</sub></sub> is the distance of target 1606 at position P<sub>1</sub> from launch point 1611, D<sub>S<sub>1</sub></sub> is the distance from marksman 1612 to target 1606 along shot path 1615, φ<sub>1</sub> is the angle between shot path 1615 and distance 1613, θ is the launch angle between target path 1607 and distance 1613. Lead distance 1610 is:

$$D_{Lead} \approx D_{P_2} - D_{P_1} \tag{Eq. 14}$$

$$D_{Lead} \approx \frac{A \Delta D_S \tan C \Delta \varphi}{\cos B \theta \tan C \Delta \varphi - \sin B \theta} \tag{Eq. 15}$$

where D<sub>Lead</sub> is lead distance 1610, ΔD<sub>S</sub> is the difference between the distances of shot paths 1614 and 1615, Δφ is the difference between angles φ<sub>2</sub> and φ<sub>1</sub>, θ is the launch angle between target path 1607 and distance 1613, A is a variable multiplier for shot size, gauge, and shot mass, B is a variable multiplier for θ including vibration of a target thrower and a misaligned target in the target thrower, and C is a variable multiplier for drag, lift, and wind.

[0245] For example, the approximate times it takes for a 7½ shot size shell with an initial muzzle velocity of approximately 1,225 feet per second to travel various distances is shown in Table 1.

Time and Distances of a 7½ Shot

[0246] Distance from Barrel Time (Seconds)

30 feet	0.027
60 feet	0.060
90 feet	0.097
120 feet	0.139
150 feet	0.186
180 feet	0.238

Table 1

[0247] Various lead distances between target 1606 and phantom target 1608 for target 1606 having an initial velocity of approximately 30 mph is shown in Table 2.

Lead Distances with a 7½ Shot on a Full Crossing Shot

Distance from Barrel	Lead Distance
60 feet	2.64 feet
90 feet	4.62 feet
120 feet	5.56 feet

Table 2

[0248] Referring to FIG. 16C, phantom path 1609 is offset from target path 1607 by drop distance 1616 to simulate and compensate for the average exterior ballistics drop of a shot.

[0249] The “drop of a shot” is the effect of gravity on the shot during the distance traveled by the shot. The shot trajectory has a near parabolic shape. Due to the near parabolic shape of the shot trajectory, the line of sight or horizontal sighting plane will cross the shot trajectory at two points called the near zero and far zero in the case where the shot has a trajectory with an initial angle inclined upward with respect to the sighting device horizontal plane, thereby causing a portion of the shot trajectory to appear to “rise” above the horizontal sighting plane. The distance at which the weapon is zeroed, and the vertical distance between the sighting device axis and barrel bore axis, determine the

amount of the “rise” in both the X and Y axes, i.e., how far above the horizontal sighting plane the rise goes, and over what distance it lasts.

[0250] Drop distance **1616** is calculated by:

$$D_{Drop} \approx v_t \tau \ln \left[ \cosh \left( \frac{t_{impact}}{\tau} \right) \right] \quad \text{Eq. 16}$$

where  $D_{Drop}$  is drop distance **1616**,  $t_{impact}$  is the time required for a shot string fired by marksman **1612** to impact phantom target **1608**.  $T_{impact}$  is determined by a set of lookup tables having various impact times at predetermined distances for various shot strings.

$$v_t = \sqrt{\frac{2mg}{C\rho A}}, \quad \text{and} \quad \text{Eq. 17}$$

$$\tau = \frac{v_t}{g} \quad \text{Eq. 18}$$

where  $v_t$  is the terminal velocity of target **1606**,  $m$  is the mass of target **1606**,  $g$  is the vertical acceleration due to gravity,  $C$  is the drag coefficient for target **1606**,  $\rho$  is the density of the air,  $A$  is the planform area of target **1606**, and  $\tau$  is the characteristic time.

[0251] Referring to FIGS. **16A** and **16C**, at step **1602**, phantom halo **1617** is determined. Phantom halo **1617** is a simulation of a shot string at a distance of the phantom target from the position of the marksman. In a preferred embodiment, an area of phantom halo **1617** is determined from the set of weapon data and calculated by:

$$A_{shot\ string} = \pi R_{string}^2 \quad \text{Eq. 19}$$

$$R_{string} = \gamma R_{initial} + v_{spread} t \quad \text{Eq. 20}$$

$$A_{phantom\ halo} = A_{shot\ string} \quad \text{Eq. 21}$$

where  $A_{shot\ string}$  is the area of the shot string,  $R_{string}$  is the radius of the shot string,  $R_{initial}$  is the radius of the shot as it leaves the weapon,  $\gamma$  is a variable multiplier for any choke applied to the weapon as determined from the set of weapon data,  $v_{spread}$  is the rate at which the shot spreads, and  $t$  is the time it takes for the shot to travel from the weapon to the target.  $A_{phantom\ halo}$  is the area of phantom halo **1617**.

[0252] In one embodiment, the area of phantom halo **1617** varies as the amount of choke applied to the weapon varies.

[0253] Returning to FIG. **16A**, at step **1603**, a relative contrast value between the target and a background surrounding the target is analyzed by calculating the difference between a grayscale brightness of the target and an average brightness of the background surrounding the target and the difference between an average color of the target and a color of the background surrounding the target based on a desired day/night setting and a set of desired environmental conditions.

[0254] At step **1604**, a color and a contrast level of a phantom target is determined. In a preferred embodiment, the phantom target includes a set of pixels set at a predetermined contrast level. The predetermined contrast level is determined by the difference of the color between the phantom target and the target and the difference of the brightness between the phantom target and the target. In this

embodiment, the predetermined contrast level is a range from a fully opaque image to a fully transparent image with respect to the image of the target and the image of the background.

[0255] In a preferred embodiment, the set of pixels is set at a predetermined color. For example, blaze orange has a pixel equivalent setting of R 232, G 110, B 0.

[0256] At step **1605**, a color and contrast level of the phantom halo is determined. In a preferred embodiment, the phantom halo includes a set of pixels set at a predetermined contrast level. The predetermined contrast level is determined by the difference of the color between the phantom halo and the target and the difference of the brightness between the phantom halo and the target. In this embodiment, the predetermined contrast level is a range from a fully opaque image to a fully transparent image with respect to the image of the target and the image of the background.

[0257] In a preferred embodiment, the set of pixels is set at a predetermined color. For example, black has a pixel equivalent setting of R 0, G 0, B 0. Any color may be employed.

[0258] Referring to FIG. **17**, a view of a simulation from the perspective of a marksman wearing a user device, such as user device **900**, is shown. Through display **1700**, background environment **1701** and target **1702** are viewed. Phantom target **1703** is projected at a lead distance and at a drop distance from target **1702**. Phantom halo **1704** is projected surrounding phantom target **1703**. Marksman **1705** aims weapon **1706** at phantom target **1703**.

[0259] In a preferred embodiment, shot center **1707** appears on display **1700** when marksman **1705** pulls a trigger of weapon **1706**. Shot string **1708** surrounds shot center **1707**. In a preferred embodiment, shot string **1708** is a simulation of a shot pellet spread fired from weapon **1706**.

[0260] In an alternative embodiment, shot center **1707** is not displayed and shot string **1708** is displayed traveling from the barrel of weapon **1706** along a trajectory. The trajectory, size, positioning, and flight path of shot string **1708** are based on the location and orientation of weapon **1706** and are based on the type of ammunition selected for the simulation. When shot string **1708** intersects target **1702**, target **1702** is destroyed. An image of one or more of target **1702**, phantom target **1703**, and phantom halo **1704** can be paused and displayed at their respective locations when the trigger of weapon **1706** was pulled while the target **1702** continues to move along its trajectory and shot string **1708** continues to move along its trajectory.

[0261] Referring to FIG. **18**, an isometric view shows an input device configured to be mounted on a rail system of a weapon. Input device **1802** is to be mounted to rail interface system **1804** of weapon **1806**.

[0262] Weapon **1806** includes barrel **1808**, sight **1846**, frame **1842**, member **1844**, cylinder **1810**, hammer **1812**, handle **1814**, trigger **1816**, trigger guard **1818**, trigger sensor **1860**, and rail interface system **1804**. Weapon **1806** is a double-action revolver wherein operation of trigger **1816** cocks and releases hammer **1812**. Rotation of cylinder **1810** is linked to movement of hammer **1812** and trigger **1816**.

[0263] Barrel **1808** is connected to frame **1842** and member **1844**. Member **1844** supports barrel **1808** and is the portion of weapon **1806** to which rail interface system **1804** is mounted. In alternative embodiments, rail interface system **1804** is mounted to other parts or portions of weapon **1806**, such as being directly mounted to barrel **1808**.

[0264] Frame 1842 connects barrel 1808, member 1844, trigger guard 1818, trigger 1816, handle 1814, hammer 1812, and cylinder 1810. Frame 1842 and handle 1814 house the mechanisms that create action between trigger 1816, cylinder 1810, and hammer 1812.

[0265] Rail interface system 1804 is a rail system for interfacing additional accessories to weapon 1806, such as tactical lights, laser aiming modules, forward hand grips, telescopic sights, reflex sights, red-dot sights, iron sights, holographic sights, bipods, bayonets, and so on. Rail interface system 1804 may conform to one or more standard rail systems, such as the Weaver rail mount, the Picatinny rail (also known as MIL-STD-1913), and the NATO Accessory Rail. Rail interface system 1804 includes screws 1820, base 1822, member 1848, and rail 1826.

[0266] Screws 1820 fit and secure rail interface system 1804 to member 1844 of weapon 1806. Screws 1820 compress base 1822 and member 1848 of rail interface system 1804 against member 1844 of weapon 1806.

[0267] Rail 1826 includes ridges 1824, slots 1850, and angled surfaces 1856. The longitudinal axis of rail 1826 is substantially parallel to the longitudinal axis of barrel 1808. Slots 1850 are the lateral voids or slots between ridges 1824 that are perpendicular to both the longitudinal axis of rail 1826 and the longitudinal axis of barrel 1808. Rail 1826 also includes a longitudinal slot 1852 that runs along the length of rail 1826 and is substantially parallel to the longitudinal axis of barrel 1808. Angled surfaces 1856 of rail 1826 allow for the precise mounting of accessories to rail 1826.

[0268] Input device 1802 includes rail mount 1828, first portion 1830, second portion 1832, battery 1834, processor 1836, LEDs 1854, button 1838, and screws 1840. Input device 1802 slides longitudinally onto rail 1826 of rail interface system 1804 of weapon 1806 and its position is secured by screws 1840. The front surface of input device 1802 is flush with a ridge 1824 of rail 1826 so that the location and orientation of input device 1802 with respect to barrel 1808 is known and the firing of weapon 1806 can be accurately simulated.

[0269] Rail mount 1828 of input device 1802 includes first portion 1830, second portion 1832, and angled surfaces 1858. Angled surfaces 1858 of rail mount 1828 correspond to angled surfaces 1856 of rail 1826 to allow for a tight and precise fitment of input device 1802 to rail interface system 1804. Screws 1840 of input device 1802 compress first portion 1830 and second portion 1832 against rail 1826 of rail interface system 1804 with sufficient force to prevent changes in the positioning or orientation of input device 1802 with respect to weapon 1806 as weapon 1806 is being used.

[0270] Battery 1834 of input device 1802 is connected to and powers the electrical components within input device 1802 including processor 1836 and LEDs 1854. Processor 1836 controls LEDs 1854. In additional embodiments, input device 1802 includes one or more sensors, accelerometers, gyroscopes, compasses, and communication interfaces. The sensor data from the sensors, accelerometers, gyroscopes, and compasses is sent from input device 1802 to a computer, such as computer 801 of FIG. 8, via the communication interface. Input device 1802 includes button 1838 to turn on, turn off, and initiate the pairing of input device 1802.

[0271] LEDs 1854 emit light that is sensed by one or more cameras or sensors, from which the locations and orientations of input device 1802 and weapon 1806 can be deter-

mined. The locations and orientations are determined from the transmission characteristics of the light emitted from LEDs 1854, and the placement characteristics of LEDs 1854.

[0272] Trigger sensor 1860 detects the pull of trigger 1816 when trigger 1816 presses onto pressure switch 1862 with sufficient movement and force. When hammer 1812 is fully cocked, trigger 1816 rests just above pressure switch 1862 so that any additional movement will release hammer 1812 and will activate pressure switch 1862. One or more wires 1864 electrically connect trigger sensor 1860 to processor 1836 so that processor 1836 can determine when trigger 1816 is pulled when blanks or live rounds are not used. Trigger sensor 1860 is contoured to fit onto the back end of trigger guard 1818 behind trigger 1816 and trigger sensor 1860 is secured onto trigger guard 1818 by screws 1866.

[0273] In a two wire embodiment, current from processor 1836 through a first wire of wires 1864 to trigger sensor 1860 is returned through a second wire of wires 1864. In an alternative embodiment, wire 1864 is a single wire and a return path for the current from processor 1836 through wire 1864 to trigger sensor 1860 is created by electrically connecting trigger sensor 1860 to trigger guard 1818, which is electrically connected to frame 1842, rail interface system 1804, input device 1802, and processor 1836.

[0274] In alternative embodiments, weapon 1806 is loaded with one or more live or blank rounds of ammunition that discharge through barrel 1808 after hammer 1812 is cocked and trigger 1816 is then pulled. Weapon 1806 does not include sensors for measuring the precise location of cylinder 1810, hammer 1812, and trigger 1816. During simulation and after a round has been fired, the simulation shows the movement of cylinder 1810, hammer 1812, and trigger 1816 to prepare for a subsequent shot, which may or may not correspond to the actual state of weapon 1806.

[0275] In alternative embodiments, the computer that receives data from one or more sensors from input device 1802 derives the state of weapon 1806 from data received from one or more sensors and updates the display of weapon 1806 to show the state and/or firing of weapon 1806 in the simulation. For example, data from sensors, accelerometers, and gyroscopes within input device 1802 can indicate the click for when hammer 1812 is fully cocked, indicate the click for when cocked hammer 1812 is released and the chamber in cylinder 1810 is unloaded, and indicate the discharge of a live or blank round of ammunition. Data from a microphone, such as microphone 919 of FIG. 9, can be used to similarly detect one or more states of weapon 1806 and the discharge of live or blank rounds of ammunition. When cylinder 1810 is configured to hold six rounds of ammunition and six shots have been fired successively, the simulation may indicate to the user that it is time to reload weapon 1806. The simulation displays changes to the state of weapon 1806 as mechanical movements on weapon 1806 and displays the firing of weapon 1806 with associated mechanical movements of weapon 1806.

[0276] Referring to FIG. 19, a simulation view shows "beams" being projected from a barrel of a weapon. Weapon 1902 includes barrel 1904 with one or more simulated beams 1906, 1912, 1916, 1920, 1924, 1928, 1932, and 1936 that emanate from the tip of barrel 1904. Beams 1906, 1912, 1916, 1920, 1924, 1928, 1932, and 1936 follow and are adjusted with the movement of barrel 1904 of weapon 1902.

[0277] The beam of a laser in a real-world environment is generally not visible to an observer unless reflected from an object in the environment. In a virtual reality environment, however, a simulated laser beam can be calculated and displayed. Simulated beams can be displayed with any level of transparency and can demonstrate characteristics that are not possible in the real-world. For example, the simulated beam can be displayed as visible, and with a dispersion pattern or in a curved path.

[0278] As an example, beam 1906 is a beam of a simulated laser and is displayed as visible along its entire length. The beam is displayed as a line or as a tight cylinder. Beam 1906 emanates from point 1908 that is central to and aligned with barrel 1904. Beam 1906 indicates the precise direction that barrel 1904 is pointed. Beam 1906 extends to point 1910 that is on the central longitudinal axis of barrel 1904 and is a fixed distance away from barrel 1904.

[0279] In another embodiment, beam 1912 is displayed as a conical frustum starting from barrel 1904 and extending to circular cross section 1914. The increase of the radius of beam 1912 from the radius of barrel 1904 to circular cross section 1914 approximates the increasing spread of a shot as it travels away from barrel 1904. Circular cross section 1914 is displayed at the termination plane of beam 1912 and provides an indication of the maximum distance that a shot on target can reliably register as a hit.

[0280] Beams 1906 and 1912 maintain their respective shapes and orientation with respect to barrel 1904 as it is moved. Pulling the trigger of weapon 1902 while beam 1906 or beam 1912 is aligned with a phantom target or phantom target, such as phantom target 1703 or phantom halo 1704 of FIG. 17, registers as a hit to the simulated target.

[0281] Beam 1916 is displayed as a curved line that extends from point 1908 at barrel 1904. Beam 1916 is tangential to beam 1906 at point 1908 and ends at point 1918.

[0282] In another embodiment, beams 1916 and 1920 are curved to approximate the drop of a shot due to gravity. The curvature of beams 1916 and 1920 is calculated based on the amount of simulated force due to gravity 1940 and the angle of barrel 1904 when the trigger is pulled. Pulling the trigger of weapon 1902 while beam 1916 or beam 1920 is aligned with a phantom target or phantom target, such as phantom target 1703 or phantom halo 1704, registers as a hit to the simulated target.

[0283] In another embodiment, beam 1920 is displayed as a curved conical frustum beginning at barrel 1904 and ending at circular cross section 1922. Beam 1920 is curved to approximate the drop of a shot due to gravity and has a radius that increases along the length from barrel 1904 to circular cross section 1922 to simulate the spread of a shot.

[0284] In another embodiment, beams 1924 and 1928 are curved to approximate changes in shot trajectory due to windage 1942. The amount of curvature of beams 1924 and 1928 is based on the amount of simulated force due to windage 1942 and the angle of barrel 1904 with respect to windage 1942. The simulation of windage may approximate changes in wind velocity and direction, such as found in a gusty wind. In this embodiment, the simulation is calculated so that the beam moves with respect to the longitudinal axis of the barrel to indicate how the shot would be affected by windy conditions. When windage 1942, is simulated, pulling the trigger of weapon 1902 while beam 1924 or beam 1928

is aligned with a phantom target or phantom target, such as phantom target 1703 or phantom halo 1704, registers as a hit to the simulated target.

[0285] Beam 1924 is displayed as a curved line that extends from point 1908 at the tip of barrel 1904. Beam 1924 is tangential to beam 1906 at point 1908 and ends at point 1926.

[0286] Beam 1928 is displayed as a curved conical frustum starting at the circular tip of barrel 1904 and ending at circular cross section 1930. Beam 1928 is curved to approximate the drop of a shot due to gravity and has a radius that increases along the length from the tip of barrel 1904 to circular cross section 1930 to simulate the spread of a shot.

[0287] Beams 1932 and 1936 are curved to approximate changes in shot trajectory due to both gravity 1940 and windage 1942. The curvature of beams 1932 and 1936 is based on the amount of gravity 1940 and windage 1942 and based on the angle of barrel 1904 with respect to gravity 1940 and windage 1942. When both gravity 1940 and windage 1942 are simulated, pulling the trigger of weapon 1902 while beam 1932 or beam 1936 is aligned with a phantom target or phantom target, such as phantom target 1703 or phantom halo 1704, registers as a hit to the simulated target.

[0288] Beam 1932 is displayed as a curved line that extends from point 1908 at the tip of barrel 1904. Beam 1932 is tangential to beam 1906 at point 1908 and ends at point 1934.

[0289] Beam 1936 is formed as a curved conical frustum starting at barrel 1904 and ending at circular cross section 1938. Beam 1936 is curved to approximate the changes to the trajectory of a shot due to both gravity 1940 and windage 1942 and the radius of beam 1936 increases along the length from the tip of barrel 1904 to circular cross section 1938 to approximate the spread of a shot.

[0290] In one preferred embodiment, a video capture system, such as Microsoft HoloLens, in combination with prerecorded videos of the shooting field and multiple actual clay target launches are used to create a virtual model of the surroundings and trajectories of clay targets for display and use in the system.

[0291] The locations and orientations of the launchers are derived based on the known location of the camera with respect to the field, the known size and weight of the targets, and the known physical constraints of the environment (e.g., gravity). After deriving the launcher locations and orientations, virtual or holographic launchers can be placed at similar positions in virtual reality or augmented reality simulations of the fields, as will be further described.

[0292] Referring to FIG. 20A, five stand field 2000 includes five shooter locations with six launchers. Five stand field 2000 includes launchers 2002, 2004, 2006, 2008, 2010, and 2012 that launch targets onto paths 2014, 2016, 2018, 2020, 2022, and 2024, respectively. Cameras 2026 and 2028 are positioned to view all towers and launchers. A video of the high tower and the low tower shot with a normal lens at 60 fps from station 4 can be processed and used to show correct trajectory and correct lead from any point of view at any station. The trajectory of the target is the same, being viewed from different angles.

[0293] Referring to FIG. 20B, sporting clays field 2050 includes three shooter locations that each have four launcher locations. The shooter and launch locations in sporting clays are unique to the venue. Sporting clays field 2050 includes



four launchers labeled T1 through T4 for each of the three shooter positions S1, S2, and S3. Drones 2052 and 2054 include cameras that record the paths of the clay targets. Drones 2052 and 2054 are capable of sensing and recording their respective GPS locations while in flight. The same process can be used to record the flight trajectories of birds, drones, helicopters and airplanes for purposes of simulating correct spatial lead.

[0294] Referring to FIG. 21A, an alternate embodiment of the simulation system will be described. System 2100 includes system computer 2101. System computer 2101 includes programs 2102, 2103, and 2120. Program 2102 is software capable of operating the Microsoft HoloLens system, as will be further described. Program 2103 includes instructions to operate a unity 3D simulation of the system, as will be further described. Program 2120 is simulation software capable of communicating with programs 2102 and 2103. In a preferred embodiment, program 2120 is the Unity 3D simulation engine, as will be further described.

[0295] Head set 2104 is connected to system computer 2101. Head set 2104 includes an augmented reality display or a virtual reality display, as will be further described. System computer 2101 is further connected to camera 2105 and camera 2106. The cameras are used in registering fixed objects such as launchers and towers and in creating trajectory models of moving objects such as clay targets in the Microsoft HoloLens system, as will be further described.

[0296] System computer 2101 is attached to wireless interface 2108. In a preferred embodiment, wireless interface 2108 is a Bluetooth interface. System computer is also attached to dongle 2109. In a preferred embodiment, dongle 2109 is compatible with the Vive Tracker, available from HTC.

[0297] System 2100 further includes trigger unit 2114. Trigger unit 2114, in a preferred embodiment, is attached to the weapon and includes sensors to detect trigger pulls. The sensors communicate signals through an onboard wireless interface to wireless interface 2108.

[0298] System 2100 further includes electronic cartridge 2112 and barrel bore arbor mounted sensor 2110. In a preferred embodiment, both include onboard wireless interfaces which communicate with wireless interface 2108. Electronic cartridge 2112 communicates with barrel bore arbor mounted sensor 2110 via light signal 2111, as will be further described.

[0299] Electronic cartridge 2112 in a typical usage is chambered in the weapon. In a typical embodiment, barrel bore arbor mounted sensor 2110 is secured in the muzzle of the weapon.

[0300] System 2100 also includes positioning detector 2204, as will be further described.

[0301] Referring to FIG. 21B, in a preferred embodiment of a virtual reality system, a system computer 2101 is connected to head unit 2122 and positioning detector 2123.

[0302] System computer 2101 runs operating system 2124, which runs virtual reality simulation engine 2125. System computer 2101 receives input from head unit 2122 and positioning detector 2123 that includes measurement data, which is used to identify the positions of head unit 2122 and positioning detector 2123. System computer 2101 outputs images to head unit 2122 that are rendered using virtual reality simulation engine 2125.

[0303] Head unit 2122 includes sensors 2135 that provide measurement data that is used to identify the position of

head unit 2122. Head unit 2122 also includes display 2136 that shows three dimensional images. The measurement data is processed by system computer 2101 and used to generate the images displayed by the one or more display screens.

[0304] Positioning detector 2123 includes sensors 2137, is mounted to a weapon, and provides measurement data. System computer 2101 receives and processes the measurement data from positioning detector 2123 to update the position of the weapon inside of the simulation.

[0305] Operating system 2124 runs on system computer 2101 and provides standard interfaces for applications to run and access external hardware. Applications running under operating system 2124 on system computer 2101 access data provided by hardware devices, such as head unit 2122 and positioning detector 2123, through hardware drivers 2126.

[0306] Hardware drivers 2126 include device drivers for each of head unit 2122 and positioning detector 2123. Hardware drivers 2126 allows virtual reality simulation engine 2125 to access the measurement data provided by head unit 2122 and positioning detector 2123 and to send images to head unit 2122.

[0307] Virtual reality simulation engine 2125 runs under operating system 2124. In a preferred embodiment, the virtual reality simulation engine runs in program 2120. The simulation engine receives measurement data from head unit 2122 and positioning detector 2123, renders virtual reality images based on the measurement data and the state of the simulation, and sends the images back to head unit 2122 to be displayed to the user. In a preferred embodiment, virtual reality simulation engine 2125 uses one or more software objects to run the virtual reality simulation, including player object 2127, head unit object 2128, weapon object 2129, tracker object 2130, target object 2131, and launcher object 2132. Every time a new frame or image is generated, virtual reality simulation engine 2125 updates each of the objects based on the measurement data, the amount of time since the last update, and the previous state of the simulation.

[0308] Player object 2127 represents the user inside of virtual reality simulation engine 2125 and its location is based on the location of head unit 2122. Player object 2127 is linked to head unit object 2128, which stores the current location of head unit 2122. Head unit object 2128 identifies the current location of head unit 2122 by accessing the measurement data provided by head unit 2122 through hardware drivers 2126.

[0309] Weapon object 2129 represents, in virtual reality simulation engine 2125, the weapon to which positioning detector 2123 is attached. The position of weapon object 2129 is linked to the position of positioning detector 2123 so that movements of positioning detector 2123 result in movements of weapon object 2129 inside of virtual reality simulation engine 2125. Weapon object 2129 is linked to tracker object 2130 so that when tracker object 2130 updates its position, the position of weapon object 2129 is also updated.

[0310] Tracker object 2130 receives measurement data from positioning detector 2123 through hardware drivers 2126. Tracker object 2130 updates the position of positioning detector 2123, which is used by virtual reality simulation engine 2125 and weapon object 2129 to update the visible location of weapon object 2129 within virtual reality simulation engine 2125. Tracker object 2130 also receives button status data within the measurement data. The button status data is used to identify when a shot is fired and when a target should be launched.

[0311] Target object 2131 is a digital representation of a clay target. Target object 2131 is instantiated when a button is pressed on positioning detector 2123. The button press is identified by tracker object 2130 and target object 2131 is brought into the simulation at the location and direction specified by the launcher object. Target object 2131 is identified as a rigid body to a physics engine of virtual reality simulation engine 2125 and its position is updated based on the simulated weight, position, and velocity of target object 2131. Upon initial placement, target object 2131 a simulated force is applied to target object 2131 to make it move inside of virtual reality simulation engine 2125.

[0312] Launcher object 2132 represents the starting location of target object 2131 and can be placed at any position inside of virtual reality simulation engine 2125. For simulations that include a launcher in a high house, launcher object 2132 is located inside a digital representation of the high house.

[0313] Referring to FIG. 21C, an augmented reality system includes head unit 2122 and positioning detector 2123.

[0314] Head unit 2122 includes computer 2121, sensors 2135, and display 2136.

[0315] Positioning detector 2123 includes sensors 2137 and is mounted to the weapon. Positioning detector 2123 provides measurement data that allows is used to determine the location of positioning detector 2123 with respect to the environment and the location of head unit 2122.

[0316] Sensors 2135 of head unit 2122 are used to provide measurement data that identifies the position of head unit 2122 and generates and updates mesh object 2134. Camera 2138 of head unit 2122 are used to locate and track registration marks on the towers and the weapon, as will be further described.

[0317] Display 2136 is mounted within head unit 2122 and displays three dimensional images or holograms to the user.

[0318] Computer 2121 receives measurement data from sensors 2135 of head unit 2122 and from sensors 2137 of positioning detector 2123 and renders an overlay image or hologram for each time step that is shown in display 2136. Computer 2121 hosts operating system 2124.

[0319] Operating system 2124 runs on computer 2121 and contains several applications, including virtual reality simulation engine 2125 and hardware drivers 2126. Operating system 2124 provides standard interfaces for the applications to access data from hardware devices by using hardware drivers 2126. In a preferred embodiment, operating system 2124 is Windows 10 from Microsoft Corp.

[0320] Virtual reality simulation engine 2125 renders each image shown through display 2136 based upon the measurement data from sensors 2135 and 2137, the amount of time since the last image was rendered, and the state of the simulation. Virtual reality simulation engine 2125 includes several objects that are used to render an image, including player object 2127, head unit object 2128, weapon object 2129, tracker object 2130, target object 2131, launcher object 2132, spatial anchor 2133, and mesh object 2134. In a preferred embodiment virtual reality simulation engine 2125 is the Unity 3D engine from Unity Technologies.

[0321] Player object 2127 represents the user in virtual reality simulation engine 2125. In an augmented reality simulation, Player object 2127 is not shown, but the position of the player is constantly updated. The position of player object 2127 is associated with head unit object 2128 so that

when the position of head unit object is updated, the position of player object 2127 is also updated.

[0322] Head unit object 2128 maintains the current position of head unit 2122 within virtual reality simulation engine 2125. For each frame, the position of head unit object 2128 is updated based on measurement data from sensors 2135 that is received through hardware drivers 2126.

[0323] Weapon object 2129 is the representation of the weapon inside virtual reality simulation engine 2125. For an augmented reality simulation, weapon object 2129 is not graphically displayed. The position of weapon object 2129 is associated with the position of tracker object 2130 and is updated for each frame of the simulation based on the movement of positioning detector 2123. The location and orientation of weapon object 2129 is used to determine if a shot hits a target.

[0324] Tracker object 2130 represents positioning detector 2123 inside of virtual reality simulation engine 2125 and identifies the position of positioning detector 2123 and the status of one or more buttons connected to positioning detector 2123. Tracker object 2130 communicates with sensors 2137 of positioning detector 2123 through hardware drivers 2126. The measurement data provided by sensors 2137 of positioning detector 2123 include position data and button status data from which the current position of positioning detector 2123 is identified and stored into tracker object 2130.

[0325] Target object 2131 in virtual reality simulation engine 2125 represents the virtual clay target. In a preferred embodiment, target object 2131 is displayed as a hologram using display 2136. Target object 2131 is initially created and instantiated at the location of launcher object 2132 with the same direction as launcher object 2132. Target object 2131 is identified as an object to which physics apply (e.g., gravity) by making it a rigid body object. Once placed into virtual reality simulation engine 2125, target object 2131 is given an initial force that causes it to move through virtual reality simulation engine 2125. For each frame, the position of target object 2131 is updated by the physics engine of virtual reality simulation engine 2125 based on a simulated weight, velocity, and any other applied forces.

[0326] Launcher object 2132 represents the location of a launcher in virtual reality simulation engine 2125. Launcher object 2132 is locked to a specific point on mesh object 2134 that is represented by spatial anchor 2133. To position launcher object 2132, spatial anchor 2133 is placed on to mesh object 2134. In a preferred embodiment, launcher object 2132 is placed on or within a tower or high house. When spatial anchor 2133 is placed on or inside a real life tower, virtual reality simulation engine 2125 does not render a model of the tower. When spatial anchor 2133 is placed on the ground, virtual reality simulation engine 2125 renders and displays a model of tower, within which launcher object 2132 is located.

[0327] Mesh object 2134 represents the three dimensional environment in which the user is located. Mesh object 2134 is a three dimensional surface of the environment measured by sensors 2135 of head unit 2122 and includes representation of the buildings and trees or, if indoors, walls, ceilings, floors, and objects surrounding the user.

[0328] Referring to FIG. 22A, weapon 2200 is used with the simulation system. Trigger unit 2202 is secured to weapon 2200 with fasteners 2206 and 2208. Trigger unit 2202 includes paddle 2210. Upon deflection of the paddle,

the trigger unit sends electric signals utilized by the system. In one embodiment, trigger unit **2202** is in electronic communication with the simulation computer using a short range wireless communications protocol, such as Bluetooth, as will be further described. Positioning detector **2204** is fitted to a known position on weapon **2200** with respect to barrel **2212**, as will be further described. In one embodiment, positioning detector **2204** includes USB port **2224**. Cable **2226** connects the USB port to the trigger unit for communication of operational signals, as will be further described.

[0329] Referring to FIG. 22B, weapon **2200** is alternatively used with the simulation system. Weapon **2200** includes electronic cartridge **2213** chambered in the weapon (not shown). Weapon **2200** further includes sensor arbor **2215** secured in the muzzle of the weapon. The weapon further includes positioning detector **2204** positioned below and attached to barrel **2212**. Sensor arbor **2215** is connected to positioning detector **2204** by USB cable **2228**. Weapon **2200** includes sensor thimble or ring **2261**. Sensor arbor **2215** is connected to thimble **2261** by USB cable **2230**.

[0330] Referring to FIG. 22C, weapon **2200** is alternatively used in the simulation system. Trigger unit **2202** is secured to the weapon as previously described. Trigger unit **2202** is in electronic communication with the simulation computer as will be further described. Weapon **2200** includes visual sight markers **2250** and **2252** capable of being recognized by the Microsoft HoloLens system and are used to locate the position orientation of the weapon during a simulation, as will be further described.

[0331] Referring to FIG. 22D, weapon **2200** is alternatively used with the simulation system. Weapon **2200** includes electronic cartridge **2213** chamber in the weapon, as previously described. Weapon **2200** includes sensor arbor **2215** secured in the muzzle of the weapon, as previously described. Weapon **2200** includes sensor thimble **2261** connected to the sensor arbor, as will be further described. Weapon **2200** includes visual sight markers **2250** and **2252** capable of being recognized by the Microsoft HoloLens system and are used to locate the position orientation of the weapon during a simulation.

[0332] In a preferred embodiment, the augmented reality system is the Microsoft HoloLens running the Vuforia augmented reality platform and SDK with the Unity 3D engine. The visual sight markers **2250** and **2252** include an image (not limited to a barcode) that is printed on a flat two dimensional surface. The image is fixed to the weapon, either directly to the barrel of the weapon or to sensor arbor **2215**, so that movement of the weapon causes similar movements of the image. The images of visual sight markers **2250** and **2252** are in the field of view of a camera of the head unit when the weapon is being aimed by the user. The augmented reality system identifies the position and orientation of the head unit with respect to an origin of the current augmented reality scene. When the augmented reality system processes the data from its sensors, including the camera, the image is identified and compared with a reference image stored in a database. From this comparison, the augmented reality system determines the position and orientation of the image with respect to head unit. The augmented reality system identifies the position and orientation of the head unit with respect to an origin of the current augmented reality scene. The augmented reality system then also determines the position and orientation of the weapon

based on the positions and orientations of the image and the head unit with respect to the origin of the scene.

[0333] Referring to FIG. 22E, positioning detector **2204** includes USB port **2224**, battery **2271**, processor **2272**, memory **2273**, antenna **2274**, and sensors **2275**, all operatively connected together. Processor **2272** executes instructions stored in memory **2273** that cause positioning detector **2204** to continuously measure its position and orientation using sensors **2275** and to broadcast its position and orientation using antenna **2274**. In a preferred embodiment, positioning detector **2204** is a Vive Tracker manufactured by HTC Corporation. Positioning detector **2204** communicates over a short range wireless connection to the simulation computer through dongle **2109**, as will be further described. In other preferred embodiments, the positioning detector can transmit a launch signal or a shot signal to the system computer, as will be further described.

[0334] Referring to FIGS. 23A and 23B, trigger unit **2202** includes external case **2304** sealed by closure **2306**. Barrel clamps **2308**, and **2310** are rigidly attached to external case **2304**. Barrel clamps **2308** and **2310** are adapted to connect with a standard picatinny or weaver rail mount system. Paddle **2210** is pivotally attached to the enclosure at hinge **2312**. Switch **2314** is a spring loaded switch that is resident in external case **2304** and operatively connected to the paddle at pivot **2316**. In a preferred embodiment, all the mechanical components of the trigger unit are formed of high impact plastic.

[0335] Processor board **2318** is centrally mounted in external case **2304** through standoff **2320**. Processor board **2318** is operatively connected to battery **2322** which powers its operation. Processor board **2318** is connected to switch **2314**. Processor board **2318** also operatively connected to external USB port **2357**. In use, paddle **2210** is deflected in direction **2324** thereby activating switch **2314**. After deflection the spring loaded switch returns the paddle to its original position.

[0336] Referring then to FIG. 23C, a preferred embodiment of the electronics of trigger unit **2202** is shown. Processor board **2318** is a Razberi Pi 3 Model B board available from digikey.com. Processor board **2318** includes processor **2353**. In a preferred embodiment, processor **2353** is a Broadcom BCM 2837 1.2 GHz Quad-Core processor. Two USB ports **2354** and **2355** are included. USB port **2354** is connected to Bluetooth module **2356** which provides a short range wireless networking connection. The Bluetooth module in a preferred embodiment is Product ID 1327 Bluetooth 4.0 USB Module (v2.1 Back-Compatible) available from Ada Fruit at adafruit.com. The Bluetooth module includes antenna **2359**.

[0337] Processor **2353** is connected to general purpose input output pins **2360**, which are connected to switch **2314**. In one embodiment, switch **2314** is normally an open contact switch that when closed, completes a circuit to provide current through one of the pins to be detected by processor **2353**. Switch **2314** sends a signal to the processor which, in turn, sends a Bluetooth signal to the host computer, as will be further described.

[0338] Processor **2353** is connected to memory card **2358** via access slot **2361**. Code resident on the memory card is used to boot the processor and perform the operations necessary to control its operation, as will be further described.

[0339] FIGS. 24A, 24B, 24C, and 24D show alternate embodiments of mechanisms for attachment of the positioning detector to the barrel of the weapon.

[0340] Referring to FIGS. 24A and 24B, mounting arbor 2402 is positioned within muzzle 2401 of barrel 2412. Mounting arbor 2402 includes threads 2403 designed to fit choke threads 2405. Mounting arbor 2402 includes rigid extension 2404. Positioning detector 2204 is fitted to the rigid extension 2404 with receiver 2410. Mounting arbor 2402 also includes stabilizer 2406 connected to arbor body 2407 by standoff 2409. Arbor body 2407 includes rubberized grip cylinder 2411.

[0341] In a preferred embodiment, arbor body 2407 is formed of a durable plastic. Arbor body 2407 further includes removable closure 2444. In a preferred embodiment, the removable closure is connected to the arbor body with a suitable set of mating threads 2445. Arbor body 2407 includes window 2446. In a preferred embodiment, window 2446 is a ruby crystal. In a preferred embodiment, the window may be a transparent plexiglass capable of transmission of radiation in the 650 nanometer range.

[0342] Arbor body 2407 includes transmission tube 2450 adjacent window 2446. Transmission tube 2450 terminates in cavity 2448. Cavity 2448 includes standoffs (not shown) capable of supporting internal circuitry.

[0343] Cavity 2448 encloses photo cell 2437, circuit 2436, and battery 2435. Removable closure 2444 includes push pin connector 2438 and connector pins 2440. Photo cell 2437 is connected to circuit 2436 and generates a current based on incident laser beam 2442. Circuit 2436, in a preferred embodiment, forms a commonly known transistor amplifier, which uses current from the battery to amplify the signal from the photo cell and transmit it to push pin connector 2438. The signal generated by the circuit is received by positioning detector and used for operation of the simulation, as will be further described.

[0344] In use, the mounting arbor is threaded into the muzzle of the weapon using the rubberized grip cylinder. Laser beam 2442 from the electronic cartridge is incident on the photo cell during operation of the system. The photo cell sends a binary signal to push pin connector 2438 and connector pins 2440 which, in turn, activate the positioning detector.

[0345] Referring then to FIGS. 24C and 24D, an alternate embodiment of transmission tube 2450 will be described. Barrel clamp 2422 includes mating sections 2424A and 2424B. The sections have mating semi-cylindrical cavities 2426A and 2426B. Section 2424A includes hole 2428A. Section 2424B includes threaded hole 2428B.

[0346] When assembled, section 2424A and 2424B are fitted around barrel 2412 and into engagement with picatinny rail 2413. Bolt 2433 is positioned through hole 2428A and threaded into hole 2428B. Bolt 2435 is positioned in the hole formed by cavities 2431A and 2431B and threaded into receiver 2410. In this way, positioning detector is held securely adjacent the barrel of the weapon. The placement of the positioning detector below the barrel allows live rounds to be fired from the weapon for practice shooting in combination with the simulation system.

[0347] Referring to FIGS. 25A, 25B, 25C, and 25D, several embodiments of the electronic cartridge component will be described.

[0348] Referring to FIG. 25A, the generalized exterior of electronic cartridge 2500 of each embodiment includes rim

section 2501 and a shell case section 2502. The rim section and shell case form a hollow central chamber or cavity 2503 used for placement of electronic components. The two sections are joined by a threaded connection 2504 and may be disassembled to service interior components. In a preferred embodiment, the rim section and shell case are formed of a high impact plastic, such as polycarbonate or nylon. Generalized exterior of electronic cartridge 2500, in one preferred embodiment, includes a ruby window 2505 imbedded in shell case section 2502 at crimped end 2506. Other transparent plastics may be used. The window is graded to transmit radiation in the 650 nanometer range. In general, chambering electronic cartridge 2500 during operation of the simulation prevents the accidental discharge of a live round.

[0349] Referring to FIG. 25B, one embodiment of the electric cartridge is described.

[0350] Electronic cartridge 2510 includes cylindrical micro switch 2512. Cylindrical micro switch 2512 is centrally located in the rim section at the position of a primer. In a preferred embodiment, the micro switch is part no. EGT12, N12 available from Euchner. Cylindrical micro switch 2512 is connected to I/O pin 2513 of processor 2516. In a preferred embodiment, processor 2516 is a Razberi pi zero, machined to fit within cavity 2503. Processor 2516 is operatively connected to battery 2514. Processor 2516 is operatively connected to onboard memory 2518. Processor 2516 is operatively connected to Bluetooth module 2517. Bluetooth module 2517 is operatively connected to antenna 2520. In a preferred embodiment, Bluetooth module 2517 is the Arduino cc2541 Bluetooth 4.0 BOE data transmission module compatible with Razberi pi, available from newegg.com.

[0351] In operation, processor 2516 is booted by and receives instructions from onboard memory 2518. Once booted, the processor enters a wait state waiting for a closure signal from cylindrical micro switch 2512. Cylindrical micro switch 2512 generates a closure signal when impacted by the hammer of the weapon upon an actual trigger pull by the user. Once the signal is received, the processor activates Bluetooth module 2517 which sends a signal 2522 via antenna 2520, to wireless interface 2108.

[0352] Referring to FIG. 25C, an alternate embodiment of the electronic cartridge 2610 will be described. Electronic cartridge 2610 includes centrally positioned cylindrical micro switch 2612, as previously described. The micro switch is connected to I/O port 2613 of processor 2616, as previously described. Processor 2616 includes memory 2618 which provides boot-up and operating instructions on board. Processor 2616 is powered by battery 2614 as previously described. Processor 2616 is connected to Bluetooth module 2617 as previously described. Bluetooth module 2617 is connected to cylindrical Bluetooth antenna 2620. In this preferred embodiment, Bluetooth antenna 2620 is integrally constructed with the shell case section 2502 in a cylindrical pattern to direct radiation towards crimped end 2506. Bluetooth antenna 2620 produces Bluetooth signal 2624, upon receipt of a signal from processor 2616, as previously described.

[0353] Electronic cartridge 2621 includes micro slide switch 2611 connected to processor 2616. The micro slide switch activates the processor and the functions of the cartridge.

[0354] Processor 2616 is also connected to laser diode 2622 via I/O port 2623. In a preferred embodiment, the laser diode is a 5 milo watt 650 nanometer red laser product ID 1054 available from adafruit.com. In this and other preferred embodiments, the laser diode can take the form of an infrared LED and the various windows are designed to transmit the LED light signal.

[0355] In operation, micro slide switch 2611 is activated by the user, then the electronic cartridge is chambered. The micro switch sends a signal to processor 2616, which in turn activates laser diode 2622. Upon activation laser diode 2622 produces laser radiation or beam 2626 which is directed coaxially to the barrel of the weapon. In further operation, when the trigger of the weapon is pulled, the hammer (not shown) impacts the cylindrical micro switch 2612 which sends a signal to processor 2616, producing Bluetooth signal 2624, as previously described.

[0356] Referring to FIG. 25D, another embodiment of electronic cartridge 2710 is described. Electronic cartridge 2710 includes micro slide switch 2712 in the rim section of the cartridge. The micro slide switch is operatively connected to battery 2714. Battery 2714 is operatively connected to laser diode 2722. In another preferred embodiment, the laser diode may take the form of an infrared LED. Moving the slide switch to the “on” position activates the laser diode. When activated, the laser diode emits laser beam 2726 directed through ruby window 2723. After activation, the electronic cartridge is chambered in the weapon. In a preferred embodiment, laser beam 2726 is coaxial to the barrel of the weapon.

[0357] Referring to FIGS. 25E and 25F a preferred embodiment of sensor arbor 2570 will be described. Sensor arbor 2570 is comprised of a containment tube 2572. Containment tube 2572 is preferably construed of an aluminum alloy but also can be constructed of a rigid plastic such as polypropylene or delrin. Containment tube 2572 includes abutment flange 2574. In a preferred embodiment abutment flange 2574 is integrally formed with containment tube 2572. Containment tube 2572 is cylindrical and has the dimensions sufficient to allow placement within the muzzle of a standard 12-gauge shotgun. Other diameters may be used. Abutment flange 2574 includes interior threads 2576. Adjacent abutment flange 2574 on containment tube 2572 are retaining threads 2578. Retaining threads 2578 are arranged to mate with choke threads (not shown) in a standard 12-gauge shotgun. Window 2580 is affixed to containment tube 2572 with a suitable epoxy adhesive. Window 2580 in a preferred embodiment is plexiglass. In alternative embodiments, it may be ruby crystal. Containment tube 2572 is configured to receive indicator shield 2582. Indicator shield 2582, in a preferred embodiment, is a hemispherical frosted plexiglass material, which is translucent. Indicator shield 2582 includes threads 2584. Threads 2584 sized to mate with threads 2576 and hold indicator shield 2582 in place in containment tube 2572. Indicator shield 2582 includes rectangular USB ports 2573 and 2593. The USB ports are operatively connected to connectors 2571 and 2597, respectively.

[0358] Referring to FIG. 25F, sensor arbor 2570 includes processor 2590. Processor 2590 is functionally collected to memory 2592. In a preferred embodiment, processor 2590 is a Razberi zero, as previously described. Memory 2592 includes instructions to boot the processor and operate the functions of the sensor arbor when in use in the system.

Battery 2594 is connected to processor 2590 and supplies operational power for the functions of the device. Photo sensor 2596 is centrally located within the sensor arbor and positioned adjacent window 2580. Photo sensor 2596, in a preferred embodiment, is the four wire light sensor module available from Ugear and is compatible with the Razberi zero. Photo sensor 2596 is connected to processor 2590 through I/O connector 2597. Processor 2590 is also connected to Bluetooth module 2598. A preferred embodiment, Bluetooth module 2598 is the Arduino cc2541 Bluetooth 4.0 BOE data transmission module available from newegg.com. Bluetooth module 2598 is connected to antenna 2599. Processor 2590 is also connected to indicator LED 2595 at input output data port 2589.

[0359] In use, the sensor arbor is threaded into the muzzle of the weapon using retaining threads 2578. Abutment flange 2574 is held in place against the outside of the muzzle. USB port 2593 is connected to the positioning detector through a USB cable (not shown). USB port 2573 is connected to sensor thimble 2560 through a USB cable (not shown). Laser radiation 2591 from the electronic cartridge is incident on photo sensor 2596 during operation of the system. Photo sensor 2596 sends a first signal to the processor which, in turn, activates a status indicator signal 2588 created by indicator LED 2595. The status signal can be seen through the translucent indicator shield indicating the status of the system to the user or other observers. The processor also sends an activation signal to the positioning detector through USB port 2593.

[0360] In response to a second signal from USB port 2573, processor 2590 activates Bluetooth module 2598 and transmits a signal 2569 through antenna 2599. In a preferred embodiment, the Bluetooth signal is received by the system computer and translated into system instructions. In an alternate embodiment, in response to the second signal, processor 2590 transmits a signal to the positioning detector through USB port 2593. In this embodiment, the positioning detector then sends a third corresponding signal to the system computer.

[0361] In another preferred embodiment, upon receipt of the second signal from the USB port, processor 2590 also sends different signals to indicator LED 2595 causing it to illuminate red. In this way, in one embodiment, the indicator shield indicates a “ready” signal in green and a “shots fired” signal in red.

[0362] Referring to FIG. 25G, a preferred embodiment of sensor thimble 2560 is described. Sensor thimble 2560 includes ring cylinder 2561. In a preferred embodiment, ring cylinder 2561 is stainless steel. Attached to the exterior surface of ring cylinder 2561 is sensor 2562. Sensor 2562, in a preferred embodiment, is flexible pressure sensor part number SEN09375 available from Karlsson Robotics. The sensor can detect an impact of anywhere between 100 grams and 10 kilograms. In another preferred embodiment, sensor 2562 includes a photo emitter and a photo sensor combination, controlling circuits and a power supply, which enables the sensor to detect the proximity of the ring to a metallic object (such as a trigger).

[0363] Sensor 2562 is mechanically connected to the exterior surface of ring cylinder 2561 with an epoxy or other suitable adhesive. Sensor 2562 is electrically connected to USB port 2564. USB port 2564 is mechanically attached to the exterior surface of ring cylinder 2561 with epoxy or another suitable adhesive. USB port 2564 is connected to

USB tether **2566** through a removable connection. USB tether **2566** is also connected to USB port **2573** of sensor arbor **2570**.

[**0364**] In use, ring cylinder **2561** is placed on the trigger finger of the user and connected to USB tether **2566**. Sensor thimble **2560** is tapped on the trigger of the weapon one time to activate a target launch and a second time to simulate a trigger pull. In a preferred embodiment, the pressure exerted by the user on the thimble against the trigger of the weapon is sufficient to change the resistance in the sensor which is sensed by processor **2590**. In response, the processor sends a Bluetooth signal through antenna **2599** to the wireless interface **2108** indicating that a sensor event has occurred, as will be further described.

[**0365**] Referring to FIG. **26**, in use, the simulation system, generally, simulates launcher **26102** and digital clay target **26106**. Launcher **26102** is located at a fixed position in the simulation and provides the starting trajectory for digital clay target **26106**.

[**0366**] In the simulation, digital clay target **26106** is launched from the starting position and orientation of digital launcher **26102**. Digital clay target **26106** travels along path **26108**. In one embodiment, phantom target **26110** and hit sphere **26112** are collocated at the same point in the simulation. Phantom target **26110** and hit sphere **26112** lead digital clay target **26106** by the lead distance **26107**, along path **26108**.

[**0367**] When a trigger event occurs, the simulation program creates a “ray” object that starts at the muzzle of weapon **26104** and is coaxial to the central axis of the barrel. If ray **26114** intersects hit sphere **26112**, then a determination is made by the simulation program as to whether or not a hit has occurred. A “hit” is determined based on the statistical likelihood of a hit based on the Gaussian distribution of pellets in a typical spread pattern for the type of ammunition chosen in the simulation, as will be further described. The Gaussian distribution of pellets is also referred to as a shot distribution probability. The diameter of the hit sphere is also determined by the Gaussian distribution of pellets, as will be further described. In a preferred embodiment, the hit sphere is three standard deviations of the pellet spread.

[**0368**] Referring to FIG. **27**, the Gaussian distribution of pellets for a standard 12-gauge round at a target distance of 70 feet, as used in the simulation, is described. Spread pattern **27102** shows a particular spread pattern for a 12-gauge round. Spread patterns have different characteristics depending on pellet count, powder charge, weapon gauge, pellet size, and barrel length and distance to target.

[**0369**] Graph **27104** shows that the vertical distribution of pellets and obeys a standard Gaussian distribution. Similarly, graph **27104** shows that the horizontal distribution of pellets and obeys a standard Gaussian distribution. Each graph changes as a function of distance to target. As expected, the standard deviation distance increases with distance to target.

[**0370**] In this example, graph **27104** includes histogram **27107**, normal distribution **27108**, and standard deviation ( $\sigma$ ) **27110**. Histogram **27107** shows that highest concentration of pellets are in the center of the spread pattern **27102**. Standard deviation **27110** is located at 4.49 inches away from the center for the vertical axis.

[**0371**] In this example, graph **27112** analyzes the horizontal spread of pellets with histogram **27114** and normal

distribution **27116**. Standard deviation **27118** is 4.24 inches for graph **27112**, indicating that there is a tighter spread along the horizontal axis. There is a larger concentration of pellets in the central bucket of the histogram, as compared to graph **27104**, which correlates with standard deviation **27118** being smaller than standard deviation **27110**.

[**0372**] Ellipse **27105** identifies a boundary of spread pattern **27102** that is three standard deviations away from the center of the spread. The boundary of spread pattern **27102** that is two standard deviations away from the center of the spread is identified by ellipse **27106**.

[**0373**] Referring to FIG. **28A**, method **2800** is used to determine the location of a simulated launcher in a clay shooting field and a set of trajectories for the digital clay targets used for the simulation.

[**0374**] At step **2802**, several trajectories of actual targets are recorded by the video cameras as they are launched from actual towers at the clay shooting field. The cameras are placed at known GPS positions to record the flight path of a target for each tower and for each possible trajectory for a target from each tower.

[**0375**] In one embodiment, the actual targets are clay targets launched from actual clay target launchers. Because the clay targets are a regular shape they follow a generally arc path defined by physics, as would be expected. In another embodiment, the actual targets are live birds, for example, ducks, pigeons and chucker. Unlike actual clay targets, actual birds typically do not exhibit well defined flight paths or trajectories for a number of reasons, including first that the birds exhibit powered flight and second live animals exhibit unpredictable characteristics, upon occasion. Additionally, measurements of windage, humidity, temperature, and barometric pressure can be recorded for use by the simulation.

[**0376**] At step **2804**, the speed and trajectory of the target is determined from the video provided by the cameras. A mathematical model of each trajectory, of each target, from each tower is created by the simulation program, as will be further described. From these models the position of the target can be calculated and displayed relative to the tower as a function of time. However, slight variations from the mathematical model are necessary to provide the virtual target with a more realistic trajectory and appearance. For example, wind gusts randomly raise and lower the clay above the perfect trajectory. Likewise variations in velocity can occur due to wind and humidity. To correct for these variations the path of the mathematical model is compared frame to frame to the video viewed from a position in the simulation that matches the position of the camera that took the video. The mathematical model is changed to account for the variations and stored in a combined trajectory file. Additional embodiments incorporate trajectory variations from atmospheric conditions and other forces acting on the target, such as drag, turbulence, and powered flight into the mathematical models. The combined trajectory is stored as a file for use in the simulation engine.

[**0377**] In a preferred embodiment, the pure mathematical models are developed by a function of Unity 3D engine. For the digital clay target, a rigid body simulation object is created that includes the known quantities of the real life clay target, including, size, weight, launch angle, and launch velocity. Additional simulation parameters for the digital clay target are adjusted based on a comparison of the flight of the digital clay target compared with the real life video of

the clay target. For example, the angular dampening of the digital clay target may be adjusted so that the digital clay target will stay aloft for about the same amount of time as a real clay target would stay aloft. To launch the digital clay target, a simulated force is applied to the digital clay target as soon as the digital clay target is instantiated into the simulation. From the initial parameters for the digital clay target, which includes the simulated force, the physics engine of the simulation system handles moving the digital clay target along a trajectory that approximates that of a real life clay target. In the case of a digital bird target, the simulation object is created that includes both a rotation and a translation attribute. A series of points is garnered from each test video which then is fitted with a spline function to interpolate all points on the trajectory. An array of trajectory paths is created which includes each of the different animations for each of the training videos. To launch the digital bird target, one array of the series of animated arrays is accessed as soon as the digital bird target is instantiated into the simulation. The digital bird target object operates from initial parameters for the digital bird which include a thrust direction based on powered flight as well as interactions with windage and humidity to result in the rotation and translation attributes which define the trajectory.

[0378] In another embodiment, the camera used to record the real clay target is a 360 degree camera, such as the Omni from GoPro, Inc. From this video, the position of the clay target is recorded and can be used to adjust the mathematically generated model trajectory in the virtual or augmented reality simulation.

[0379] At step 2806, the location and orientation of each tower is determined and stored in the simulation program.

[0380] In a preferred embodiment, the tower locations are modeled and set by the unity 3D engine.

[0381] Referring to FIGS. 28B and 28C, the registration of the launcher locations in the unity 3D engine is described. "Registration" of a point in a virtual reality space to a fixed point in the real-world is typically accomplished by creating a virtual copy of the critical features of the real-world in the unity 3D system. In one embodiment, the high house, the low house, and shooter pad locations are defined at predetermined measurements from a predefined common origin. The house dimensions are created with the "box" function in Unity 3D. The boxes each are defined with a virtual launch point that corresponds to the muzzle of the launcher in the real-world. In a similar way, the locations of the shooter pads are measured in the real-world and registered in the unity 3D engine.

[0382] "Registration" of a point in an augmented reality space to a fixed point in the real-world is typically accomplished by an augmented reality camera such as that used in the Microsoft HoloLens. In this case, a "spatial anchor" is chosen. The spatial anchor is chosen from an array called a spatial map. The spatial anchor is chosen by calling a function known as "gaze ray". The gaze ray function returns a set of coordinates in the mesh that is then named and identified as the spatial anchor. For example, image 2951, from an augmented reality camera shows a high house 2952 and a low house 2953 in skeet field 2954. The Microsoft HoloLens system creates mesh 2955. Mesh 2955 is a three dimensional map of image 2951. The registration identifies spatial anchor 2996 at a location in the mesh that corresponds to the location of the high house. The registration

identifies spatial anchor 2997 is at a location in the mesh that corresponds to the location of a launcher.

[0383] Referring to FIG. 28D, a virtual reality simulation includes high house 28402 and low house 28404. Camera icon 28406 represents the current location of the user within defined space 28408. Defined space 28408 is the safe space inside of the simulation that corresponds to the safe space in real life where the user is experiencing the simulation.

[0384] Defined space 28408 has a specific origin and orientation. High house 28402 and low house 28404 are placed with respect to the origin and orientation of defined space 28408. Both high house 28402 and low house 28404 include launcher objects that are used for the launch of clay target objects in the simulation.

[0385] Referring to FIG. 29A, method 2900 is shown. Method 2900, performed by a simulation computer, to create a virtual reality or augmented reality shooting simulation is described.

[0386] At step 2902, the location, orientation, and settings of a launcher are set. The location of each launcher includes Cartesian coordinates that identify where each launcher is placed in the simulation. The orientation of each launcher indicates the initial direction for the digital targets when launched, and is defined by three Euler angles. The Euler angles are unique for each trajectory model.

[0387] At step 2904, ambient conditions for the simulated environment are set, which include simulated windage, humidity, temperature, and barometric pressure. In one embodiment, the simulated environmental factors are set to match the environmental factors that existed when the cameras recorded the images of the actual target trajectories.

[0388] At step 2906, the settings of the digital clay target are selected. The settings include size, color, and mass. The settings are incorporated into the trajectory models.

[0389] At step 2908, weapon ammunition settings are selected. The ammunition types include those that are appropriate for the selected weapon. The ammunition settings determine the Gaussian distributions used by the simulation to determine the probability of a "hit" and the diameter of the hit sphere.

[0390] At step 2910, the phantom target settings are selected. The phantom target settings identify the color, transparency, and size of the phantom target. In a preferred embodiment, the phantom target is the same size as the digital clay target, but includes a different color and transparency in order to distinguish it from the digital clay target.

[0391] At step 2912, the lead distance is selected.

[0392] The lead distance is the linear distance between the location of the center of the digital clay target and the location of the center of the phantom target. In a preferred embodiment, the lead distance is selected as a fixed distance, usually about three feet.

[0393] In an alternative embodiment, a lead time is selected and the lead distance is calculated based on velocity of the digital target. For example, the desired lead time is multiplied by the initial velocity of the digital target to calculate the lead distance.

[0394] Additionally, the lead time can be estimated using the known positions of the weapon and the digital target, the trajectory of the digital target, the velocity of the digital target, and the muzzle velocity for the selected weapon and ammunition type.

[0395] Referring to FIG. 29B,

$$A = X_{clay} - X_{weapon} \quad \text{Eq. 22}$$

$$B = V_{clay} \cdot t \quad \text{Eq. 23}$$

$$C = v_{muzzle} \cdot t \quad \text{Eq. 24}$$

$$D = B \cdot \cos \theta \quad \text{Eq. 25}$$

$$E = B \cdot \sin \theta, \quad \text{Eq. 26}$$

where:

[0396] A is the line segment of known length between the weapon location 2980 and the digital clay target location 2982;

[0397] B is the distance between the digital clay target location 2982 and the point of impact;

[0398] C is the distance between the weapon location and the point of impact;

[0399]  $\theta$  is the angle between D and B.

and:

$$(A + v_{clay} \cdot t \cdot \cos \theta)^2 + (v_{clay} \cdot t \cdot \sin \theta)^2 = (v_{muzzle} \cdot t)^2 \quad \text{(Eq. 27)}$$

Solving for t yields an estimate of the time it will take a shot to reach point of impact 2984 from weapon location 2980 of the weapon, as follows:

$$t = \pm \frac{\sqrt{\frac{v_{muzzle}^2 - 4 \cdot A \cdot v_{clay} \cdot v_{muzzle} \cdot \cos \theta - A \cdot v_{clay} \cdot \cos \theta}{4 \cdot A^2 \cdot v_{clay}^2 \sin^2 \theta}}}{2 \cdot v_{clay} \cdot (\sin^2 \theta + \cos^2 \theta)} \quad \text{Eq. 28}$$

[0400] Other lead calculation equations may be used in other embodiments.

[0401] Referring to FIG. 29C, method 29100 of generating a simulation of the system is described.

[0402] At step 29102, the system obtains the location orientation of the headset from the headset object. In a preferred embodiment, the location is a set of Cartesian coordinates and the orientation includes an angle view.

[0403] At step 29104, the system displays range graphics as previously described. In a preferred embodiment of the virtual reality system, the range graphics include a virtual image of a high house and a virtual image of a low house in appropriate background imagery. In a preferred embodiment of the augmented reality system, the images of the high house and the low house are set to “invisible” because the actual high house and the actual low house are visible to the user through the transparent headset.

[0404] At step 29106, the system obtains the location and orientation of the weapon from the weapon object.

[0405] At step 29107, the system processes control signals received from a peripheral connected to the weapon. As is described in FIG. 31 below, the control signals allow for the user to launch a digital target, display a laser from the weapon, and turn the point of view left or right.

[0406] At step 29108, the system displays a weapon image if in virtual reality mode.

[0407] At step 29110, the system updates the display in the headset object. In the augmented reality system, the towers and launchers are not displayed (or displayed as “invisible”) because they can be seen by the user through the transparent headset. In a virtual reality system or augmented reality

system, where the real towers and launchers are not present, images of the towers and launchers may be displayed in the overlay.

[0408] At step 29112, the method checks for a launch event signal from the trigger object in the weapon object. In one embodiment, the system computer generates the launch signal automatically at predetermined time intervals. In other embodiments, the user generates the launch signal through use of the trigger unit on thimble, as will be further described, which is then posted by the trigger object. If no launch event signal is received, the method returns to step 29110. If a launch signal is received, the method moves to step 29114.

[0409] At step 29114, the virtual target object and the phantom target object are launched. The target object path is drawn from the modified trajectory recorded after manual manipulation based on camera recordings of the actual flight paths. The phantom target path is drawn from the virtual target path modified by a lead distance function, as will be further described. The simulation engine displays the target and the phantom target according to the positions assigned to the objects by the Unity 3D engine. The hit sphere object is instantiated, but invisible to the user. The phantom target is rendered as leading the digital target by a fixed distance set or calculated as previously described.

[0410] At step 29116, the position and status of the digital target object is updated based on the time step and hit record. Updating the position of the target object updates the position of the phantom target object and the hit sphere object. For each update the new position and orientation of the digital target are calculated from the trajectory model of the target.

[0411] At step 29117, the weapon position is updated based on measurements from the positioning detector on the weapon or based on the position information retrieved from the registration mark in the hololens system. The phantom target position is updated based on a new lead time or distance calculated from the updated positions of the digital target.

[0412] The size of the hit sphere is updated based on the current distance between the weapon and the digital target. The hit sphere is a mathematical construct centered at the centroid of the phantom target object. The hit sphere is used to determine a theoretical “hit” of the target by shot. In one embodiment, the radius of the hit sphere is equal to the pellet spread at the distance to target, for the chosen ammunition. In another embodiment, the hit sphere is an ellipsoid with the vertical radius based on the vertical shot spread and the horizontal axis based on the horizontal shot spread at the distance between the weapon and the centroid of the phantom target.

[0413] At step 29118, the digital target and the phantom target are rendered. The rendering is based on the updated positions of the digital target and the phantom target. The appearance of the target and the phantom target are conditioned on the predetermined settings.

[0414] At step 29119, the system updates the display showing the new position of the weapon, in the virtual reality mode.

[0415] At step 29120, the system determines whether or not a shot signal event has occurred. When the shot signal event has not occurred, the simulation returns to step 29116. When the shot signal event has occurred, the method proceeds to step 29122.



[0416] At step 29122, the current location and orientation of the weapon are retrieved from the weapon object. In one embodiment, the data is retrieved from a memory that stores the positioning data that is continuously broadcast by the positioning detector on the weapon. In another embodiment, the data is retrieved from a server that stores the positioning data that is derived by the observation of the registration structure on the weapon by the Microsoft HoloLens system.

[0417] At step 29124, a ray is created. The ray is a mathematical vector whose starting point is the end of the barrel of the weapon. The orientation of the ray is set to be coaxial with the axis of the barrel of the weapon. As a result, the ray always points the same direction as the weapon.

[0418] At step 29126, it is determined if there is a “collision” between the ray and the hit sphere. When there is no collision then the method returns to step 29116. When there is a collision, then the method proceeds to step 29128.

[0419] At step 29128, the shortest distance between the ray and the center of the hit sphere is determined. This distance is tangential to the ray and includes a horizontal component and a vertical component.

[0420] At step 29130, the probability of hitting the digital target is determined from the Gaussian pellet distribution at the time of collision. In one embodiment, the Gaussian pellet distribution may be calculated. Values from a cumulative distribution function for the normal distribution of the shot spread pattern are calculated using the equation:

$$CDF(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}\sigma} e^{-\left(\frac{t-\mu}{2\sigma^2}\right)} dt \quad \text{Eq. 31}$$

where:

$\sigma$  is the standard deviation of the spread pattern; and,  
 $\mu$  is the mean of the spread pattern, which is set to zero.

[0421] Using the cumulative distribution function, the hit probability is calculated in both the horizontal and vertical dimensions that are orthogonal to the direction of the weapon:

$$P_{horizontal} = CDF(x+r_x) - CDF(x-r_x) \quad \text{Eq. 32}$$

$$P_{vertical} = CDF(y+r_y) - CDF(y-r_y) \quad \text{Eq. 33}$$

[0422] where

[0423]  $P_{horizontal}$  is the hit probability for the horizontal dimension;

[0424]  $P_{vertical}$  is the hit probability for the vertical dimension;

[0425]  $x$  is the horizontal distance between the ray and the center of the hit sphere;

[0426]  $r_x$  is the distribution hit radius for the horizontal dimension, which is calculated by multiplying the horizontal length of the digital target by the hit scaling factor;

[0427]  $y$  is the vertical distance between the ray and the center of the hit sphere; and,

[0428]  $r_y$  is the distribution hit radius for the vertical dimension, which is calculated by multiplying the vertical length of the digital target by the hit scaling factor.

[0429] The “hit scaling factor” is set to 1 so long as the size of the digital clay target is the same as the actual clay target.

[0430] At step 29132, a random number for each dimension is generated between 0 and 1.

[0431] At step 29134, a hit is recorded based on the Gaussian pellet distribution when the random number for the horizontal dimension is less than the horizontal hit probability and the random number for the vertical dimension is less than the vertical hit probability. In one embodiment, steps 29128 through 29132 are bypassed and a hit is recorded when the ray collides with the phantom target.

[0432] At step 29136, after recording a hit, the system identifies the point of impact, which is the point on the path of the digital clay target where the hit will occur in the future. The three dimensional position of the point of impact is the current three dimensional position of the phantom target. When the digital target reaches the point of impact, the hit will be displayed as a rapid disassembly or explosion of the digital target. After step 29136, the method returns to step 29116, to continue updating the simulation of the digital clay target until it is destroyed or until the trajectory model intersects the horizon line.

[0433] Referring to FIG. 29D, an augmented reality overlay of the simulation resulting from method 2900 is described.

[0434] Overlay 2957 is an augmented reality overlay that includes digital clay target 2958, phantom target 2959, digital clay target 2960, and phantom target 2961. Digital clay target 2958 and phantom target 2959 follow path 2962 from the high house. Digital clay target 2960 and phantom target 2961 are displayed as being launched from the low house and follow path 2963.

[0435] Preferred embodiments of the launch signal event of step 29112 and the shot signal event of step 29120 will be further described here.

[0436] In one preferred embodiment, trigger unit 2202 is attached to the weapon. Processor 2353 of the trigger unit is programmed to generate a first wireless signal indicative of a launch signal upon a first contact of the user with paddle 2210. Processor 2353 is programmed to send a second, different wireless signal, indicative of a shot signal upon a second contact with paddle 2210. When used in conjunction with the barrel clamp mechanism of FIGS. 24B and 24C, a live round may be loaded into the chamber of the weapon and discharged by pulling the trigger. In this way, the augmented reality system can be used in conjunction with actual clay targets and live ammunition on an actual shooting field in order to alternate practice scenarios in real time.

[0437] In another embodiment, the trigger unit is attached to the weapon and the electronic cartridge of FIG. 25B is loaded into the chamber. In this embodiment, processor 2353 is programmed to send a wireless signal indicative of a launch signal to wireless interface 2108 upon a first contact with paddle 2210. Upon physical release of the hammer by the trigger of the weapon, the trigger impacts cylindrical micro switch 2512 whereby processor 2516 sends a wireless signal indicative of a shot signal to wireless interface 2108.

[0438] In another embodiment, the sensor arbor of FIG. 25F is secured in the muzzle of the weapon. Micro slide switch 2611 of the electrical cartridge of FIG. 25C is activated thereby instructing processor 2616 to activate laser diode 2622. The electronic cartridge is then chambered in the weapon. Laser diode 2622 sends beam 2626 down the barrel of the weapon which is received by photo sensor 2596 of the sensor arbor. Upon receipt of the signal the processor

activates indicator LED 2595 to a “green” state thereby illuminating the indicator shield to indicate system ready.

[0439] Upon a trigger pull of the weapon, the hammer (not shown) impacts cylindrical micro switch 2612 of the electronic cartridge. A signal generated by the micro switch is sensed by processor 2616. Upon sensing the signal, the processor is programmed to send a signal from the wireless interface of the electronic cartridge, indicative of a shot signal to wireless interface 2108. In an alternate embodiment, upon sensing the signal, the processor is programmed to change the signal sent by laser diode 2622 using a digital coding. When the digitally coded signal is received by photo sensor 2596 of the sensor arbor, processor 2616 activates Bluetooth module 2598 to send a shot signal 2569 from antenna 2599, to wireless interface 2108. At the same time, processor 2590 sends a second signal to indicator LED 2595 to eliminate “red” indicating a live fire condition. In this embodiment, the launch signal is generated automatically without warning to the shooter.

[0440] In another embodiment, the electronic cartridge of FIG. 25D is activated and chambered in the weapon. The sensor arbor of FIG. 25F is secured in the muzzle of the weapon. Activation of the electronic cartridge is accomplished by moving micro slide switch 2712 to an “on” position. The micro switch thereby activates laser diode 2722. Laser diode 2722 generates laser beam 2726 which is incident upon photo sensor 2596. Photo sensor 2596 sends a signal to processor 2590 which activates indicator LED 2595 to illuminate “green”.

[0441] In another embodiment, the electronic cartridge of FIG. 25D is activated and chambered in the weapon. The mounting arbor of FIGS. 24A and 24B are secured in the muzzle of the weapon. Activation of the electronic cartridge is accomplished by moving micro slide switch 2712 to an “on” position. Micro switch thereby activates laser diode 2722. Laser diode 2722 generates laser beam 2726 which is incident upon photo cell 2437. Photo cell 2437 sends signal to push pin connector 2438 and then to positioning detector 2204. Positioning detector 2204 then activates itself and sends a “ready” signal to dongle 2109. Dongle 2109 communicates the “ready” signal to system computer 2101.

[0442] Sensor thimble 2560 and USB tether 2566 are connected to USB port 2224 of positioning detector 2204. A first impact of the thimble on the trigger of the weapons sends a first signal to positioning detector which forwards it to the dongle and then on to the system computer. This first signal is interpreted as a “launch” signal. In the same way, a second impact of the thimble on the trigger of the weapon sends a signal to positioning detector which forwards it again to the dongle and the system computer. The second signal is interpreted as “shot” signal.

[0443] In this embodiment, sensor thimble 2560 is attached by USB tether 2566 to USB port 2573 of the sensor arbor. Upon impact of the ring cylinder against the trigger of the weapon, impact sensor sends a signal through USB tether 2566 to the sensor arbor. The signal is sensed first by processor 2590 which activates Bluetooth module 2598. Bluetooth module 2598 sends a wireless signal to wireless interface 2108, indicative of a launch signal. Upon a second impact of the ring cylinder on the trigger of the weapon, impact sensor 2562 sends a second signal through USB tether 2566 to USB port 2573. The signal is received by processor 2590 which sends a second signal to indicator LED 2595 to eliminate “red” indicating a live condition.

Processor 2590 also activates Bluetooth module 2598 to send a second different wireless signal to wireless interface 2108. The second wireless signal is indicative of a shot signal.

[0444] Referring to FIG. 29E, method 29200 of generating a simulation of the system is described. This method is preferably used with live or powered targets that exhibit attributes of powered flight trajectories. In a preferred embodiment, method 29200 is applied in association with a mixed reality headset set in “pass through” mode which allows “inside out” tracking from the display screen in the headset.

[0445] At step 29201, the system obtains the location and orientation of the headset from the headset object. In a preferred embodiment, the location is a cartesian set of coordinates and the orientation includes an angle of view.

[0446] At step 29202, the spatial anchors are located and the images of the high house and the low house are set to “invisible” because the actual high house and the actual low house are visible to the user through the pass through mode of the mixed reality headset.

[0447] At step 29204, the system implements the spatial anchors and the digital overlay onto the signal from the cameras to be displayed for the user. In this way, the digital objects in the simulation such as the high house and the low house, are synchronized with the real-world background objects visible to the user.

[0448] At step 29206, the system obtains the location and orientation of the weapon from the weapon object, as previously described.

[0449] At step 29208, the system processes control signals received from the weapon peripheral, as previously described.

[0450] At step 29210, the method checks for a launch event signal from the trigger attribute of the weapon object, as previously described. If no launch event signal is received, the method returns to step 29206. If a launch event signal is received, the method moves to step 29214.

[0451] At step 29214, the digital bird target object and the phantom target object are launched. The launch point is derived from the object identified as a high tower in the spatial anchors in the point cloud. The digital bird target is displayed as “flying” along an object path. The digital bird target object path is drawn from a path equation in the path array. Preferably, the path array stores different paths derived from videos recorded by cameras 150, 151, 250 and 251, as previously described. Since the path array is capable of storing many thousands of flight paths and path equations, one path equation may be chosen at random. In other embodiments, an ordered set of paths may be chosen to originate from different launch points with different targets to simulate competition skeet, trap or other ordered shooting events. At this step an animation array is also accessed. The animation array includes video samples of target attributes such as bird wing and head movement and different bird call audio files. The phantom target path is drawn from the digital bird target path. The simulation engine displays the digital bird target and the phantom target on the same path, but with the phantom target leading the digital bird target by a proper lead distance, calculated from the ballistic table and the distance to target as previously described. A hit sphere object is instantiated, and located at the position of the phantom, but yet is invisible to the user.

[0452] At step 29216, the position and orientation of the digital bird object is updated and displayed based on the time step and hit record. The new position of the digital bird object is calculated from the path model. The orientation of the digital bird object is preferably drawn from the attribute array.

[0453] At step 29218, the weapon position is updated based on measurements from the positioning detector on the weapon or based on the position information retrieved from the registration mark in the HoloLens system. The phantom target position is updated based on a lead distance calculated from the updated position of the digital bird target.

[0454] At step 29220, the digital bird target and the phantom target are rendered. The phantom target is rendered as a semi-transparent bird target leading the digital bird target by a fixed distance set or calculated as previously described. The rendering is based on the rotation and translation attributes of the digital bird target and the animation array attributes based on predetermined settings. At step 29222, the video image from the stereo camera is accessed.

[0455] At step 29224, the spatial anchors are located in the camera image. At step 29226, the change in view coordinates, Ax, Ay and Az are calculated from the last position of the spatial anchors in order to determine head movement of the user. In this way, the movement of the user is synchronized with the simulation and the background image.

[0456] At step 29228, the digital bird trajectory is adjusted to compensate for the head movement. In this embodiment, the trajectory of the digital bird is “tied” to either the high house or the low house object, which appears to the user to be stationary. As the user moves his head, the display is changed so that the path the bird appears to be consistent with actual flight.

[0457] At step 29230, the system determines whether or not a shot signal event has occurred. When the shot signal event has not occurred, the simulation returns to step 29216. When the shot signal even has occurred, the method proceeds to step 29232.

[0458] At step 29232, the current location or orientation of the weapon are retrieved from the weapon object, as previously described.

[0459] At step 29234, a ray is created, as previously described.

[0460] At step 29236, it is determined whether or not a “collision” between the ray and the hit sphere has occurred. When there is no collision, then the method returns to step 29216. When there is a collision, then the method proceeds to step 29238.

[0461] At step 29238, the shortest distance between the ray and the center of the hit sphere is determined, as previously described.

[0462] At step 29240, the probability of hitting the digital bird object is determined according to the Gaussian Pellet Distribution, as previously described.

[0463] At step 29242, a random number is generated for each dimension.

[0464] At step 29244, a hit is recorded based on the Gaussian Pellet Distribution, as previously described.

[0465] At step 29246, an animation graphic is triggered when the digital bird reaches the current position of the phantom target. In a preferred embodiment, the animation is provided by the animation array specific to the bird object chosen in the predefined set of attributes. In one preferred embodiment, the animation array shows bird activity termi-

nating and the bird falling along a physically correct arc path for an inanimate object starting with the altitude, speed and trajectory of the digital bird when the hit activity occurred.

[0466] Referring to FIG. 30, weapon movements can be used in the place of controller movements. In a preferred embodiment, the hardware used in a virtual reality simulation includes weapon 2200, positioning detector 2204, and a sensor thimble (not shown) worn by user 2201. After a long press of the sensor thimble, directional movements of weapon 2200 are interpreted as controller commands or specific actions in the simulation, an example of which is shown in the table below.

Direction	Action
Up 3002	Launch target
Down 3004	Laser toggle
Left 3006	Turn point of view within simulation to the left
Right 3008	Turn point of view within simulation to the right

[0467] In preferred embodiment, a long or slow press by the sensor thimble uses a threshold duration of 0.5 seconds and the movement has a minimum threshold of 0.5 inches. After holding the sensor thimble down for 0.5 seconds and moving the end of the barrel of the weapon up 3002 by at least 0.5 inches, the system registers a launch target command, e.g., launch signal 29112, and will launch a target after a random delay of up to two seconds. A long press of the sensor thimble followed by a downward movement 3004 of the end of the barrel of weapon 2200 will toggle on or off the display of a laser that emanates from the end of weapon 2200 during the simulation and identifies the orientation of weapon 2200 in the simulation, such as one or more of beams 1906, 1912, 1916, 1920, 1924, 1928, 1932, and 1936 of FIG. 19. Moving the barrel left 3006 or right 3008 after holding the sensor thimble for a long press rotates the point of view of the user within the simulation left or right until the sensor thimble is released. Different movements, different actions, and different mappings between movements and actions can be used.

[0468] In an alternative embodiment, voice commands are used to perform the actions listed in the table above. For example, when the user says “pull!”, the system recognizes the word, identifies that the word is mapped to the launch target action, and initiates launching the target based on the recognized voice command by activating the launch signal, such as in step 29112 of FIG. 29C. Additional voice commands can be mapped to the actions performed by the system and multiple voice commands can be mapped to the same action. The table below enumerates several voice commands that are mapped with system actions.

Voice Command	Action
“Pull” or “Launch”	Launch target
“Toggle”	Laser toggle
“Turn left” or “Look left”	Turn point of view within simulation a fixed number of degrees to the left
“Turn right” or “Look right”	Turn point of view within simulation a fixed number of degrees to the right

[0469] Referring to FIG. 31, computer implemented method 3100 is a further description of step 29107 from FIG. 29C for processing a control signal.

[0470] At step 3102, the system receives a control signal from a peripheral attached to the weapon. In a preferred embodiment, the control signal is the press of a sensor thimble connected to a positioning detector.

[0471] At step 3104, after receiving the control signal, the method determines the initial position of the weapon. In a preferred embodiment, the system stores the current position (location and orientation) of the weapon with the current time.

[0472] At step 3106, it is determined whether or not the control signal has been active for longer than a threshold amount of time. In a preferred embodiment, the threshold amount of time is 0.5 seconds and is referred to as a “long press” or “long touch” of the sensor thimble. The current time is compared to the time stored at step 3104. If the control signal has been active for longer than the threshold amount of time, then the method proceeds to step 3110, otherwise the method proceeds to step 3134, and ends.

[0473] At step 3110, it is determined if the weapon has moved a threshold distance. In a preferred embodiment, the current position of the weapon is compared to the initial position stored at step 3104 and a difference is calculated. If the distance is greater than the threshold, then the method proceeds to step 3114. If the difference is not greater than the threshold, then the method proceeds to step 3134, and ends.

[0474] At step 3114, it is determined if the movement of the weapon is in the up direction. If so, the method proceeds to step 3116. If not, then the method proceeds to step 3118.

[0475] At step 3116, the method triggers the launch of a clay target in the simulation in response to the movement of the weapon by the user. Afterwards, the method for handling control signals ends at step 3134.

[0476] At step 3118, the method determines if the movement is in a “downward” direction. If so, then the method proceeds to step 3120. If not, then the method proceeds to step 3122.

[0477] At step 3120, the method toggles on or off a “laser” image that emanates from the barrel of the weapon during the simulation, such as one or more of beam images 1906, 1912, 1916, 1920, 1924, 1928, 1932, and 1936 of FIG. 19. After toggling the laser image, the method moves to step 3134, and ends.

[0478] At step 3122, if the weapon was moved to the left, then the method proceeds to step 3124. If not, then the method proceeds to step 3128.

[0479] At step 3124, the method rotates the point of view of the user within the simulation to the left.

[0480] At step 3126, the method then checks to see whether or not the control signal is active. If so, then the method returns to step 3124. If not, then the method moves to step 3134, and ends.

[0481] At step 3128, the method determines whether or not the movement of the weapon is to the right. If so, then the method moves to step 3130.

[0482] At step 3130, the method turns the point of view of the user to the right. The method then moves to step 3132.

[0483] At step 3132, a determination is made as to whether or not the control signal is active. If so, then the method returns to step 3130. If not, then the method moves to step 3134, and ends.

[0484] Referring then to FIG. 32, an alternate embodiment of an augmented reality overlay of the simulation resulting from method 2900 is described.

[0485] Overlay 320, is a mixed reality overlay that includes both images from stereo camera 925 and a rendering of simulation objects. In this example, a mixed reality overlay includes actual high house 324, actual shooter 326 and an actual live target (such as a bird) 328, all present in the background view of the augmented reality display. The actual high house, actual shooter and live target exist in cartesian coordinate system 322 including x-axis, y-axis and z-axis with an origin at the high house. The mixed reality overlay further includes phantom target 332 rendered by the simulation and displayed on display 958. Actual live target 328 travels along actual path 330. The simulation generates projected path 334, as will be further described. Phantom target 332 leads actual live target 328 by a lead distance “i” as shown. Distance “d” is the distance between the shooter and the live target at a specific instance in time, “t”. Live target 328 is flying altitude “y”. The angle between the horizontal plane “h” and actual live target 328 at the position of actual shooter 326 is denoted by angle  $\gamma$ .

[0486] Referring to FIGS. 32 and 33, method 3300 of generating a phantom target ahead of a live bird target will be described.

[0487] At step 3302, pass through mode of mixed reality unit 950 is activated. In pass through mode, the images from stereo camera 925 are projected on display 958 with a delay of approximately 50 microseconds.

[0488] At step 3304, the system identifies a spatial anchor. In one preferred embodiment, the spatial anchor is comprised of mapped environment stored in the Microsoft Point Cloud. Once identified, the spatial anchors are uploaded to the point cloud. In this example, spatial anchors comprise actual high house 324 and actual live target 328 in the background. At step 3305, the spatial anchors are synchronized with the background so that movement of the images can be translated into movement of the position of the stereo camera. At step 3306, the actual live target is recognized as a bird object is from the object classifications available from the point cloud. In a preferred embodiment, the bird object is recognized through an API call available from the Microsoft HoloLens system.

[0489] At step 3308, the depth “d” for the bird object (the distance from the headset to the target bird object) is measured. In a preferred embodiment, the depth is obtained from API function call to processor 954. In other embodiments, the distance to target information can be obtained from a LIDAR system, a US\_RTLS system, a UWB system or a WLAN, WiFi system as previously described.

[0490] At step 3310, actual path 330 is identified. In a preferred embodiment live target 328 travels from actual high house 324 to the position shown along actual path 330. Actual path 330 is observed by stereo camera 925. Velocity is recorded at each point along the path. A set of uniformly spaced points along the path is recognized as cartesian coordinates and stored in an array. A mathematical model of the path is then calculated using the points in the array by a spline function available in the Unity 3D engine. The points in the array are passed to the spline function. The spline function generates a continuous path by interpolating between known points on the path from the array. In this example, the continuous path is shown from point “A” to point “B”. The spline function also allows the path to be

extrapolated to point “C” as shown in FIG. 32, within a certain predefined confidence interval.

[0491] At step 3312, the depth is assumed to be the distance to target and is used to calculate the lead distance based on a ballistic table for the weapon, as previously described.

[0492] At step 3314, the correct lead path is calculated. In one embodiment, the Unity 3D engine spline function is used to extrapolate the future path of the live bird up to the appropriate lead position. A set of uniformly spaced discrete points for the target in Cartesian coordinates is identified from the moving image of the bird from the stereo camera and recorded in a table. The discrete points are used by the spline function to interpolate the remaining points along the path and to project movement of the target a short distance into the future. The speed of the live bird is assumed to remain constant. In most cases, the lead distance will be about 2 feet. Because this distance is relatively short, the lead position extrapolated from the known position data using the spline function is usually sufficient accurate to be useful. The spline function takes the form of the spline interface ispline.cs in Unity 3D. The hermite spline interpolation function is employed to derive intermediate points between known points. In order to extrapolate the path, the ray function of Unity 3D is called and passed the last known point (in this case “B”) as the origin. The direction for the ray function is taken as the direction defined by the last two known points along the path.

[0493] In another preferred embodiment, image processing from the stereo camera can be used to determine the position, direction, and speed of the live bird in order to determine the correct lead path. In this case, the speed of the live bird is determined by reviewing a constantly updating moving window of the sixty most recent video frames in time. The average speed of the live bird determined from the moving window and is assumed to be constant for the entirety of the lead path. The position of the bird is determined by the most recent video frame analyzed. The direction of the live bird is determined by processing the video image to determine where the bird is “looking.” In a preferred embodiment, image processing can determine the relative positions of the head of the bird relative to the body of the bird over any number of video frames. The direction of the bird is taken as the vector direction from the centroid of the body of the bird and the centroid of the head of the bird. The vector is recorded and averaged for each frame of the sixty-frame moving window. The vector average is taken as the direction.

[0494] A ray function is used through the centroid of the body of the bird and the centroid of the head of the bird to determine the direction for the lead path.

[0495] In an alternate embodiment, the lead path is calculated using a neural network as will be further described.

[0496] At step 3316, the image of the live target available from the stereo camera is copied into memory. At step 3317, the system compensates for movement of the position of the shooter and the orientation of the cameras by adjusting the simulation to change the position of the data, the lead path and the lead distance to account for the movement, as previously described. In this way, the images of the bird at the lead position will appear normal to the shooter.

[0497] At step 3318 the copy of the bird image is rendered on the lead path at the lead distance on the lead path ahead of the live target position.

[0498] Referring then to FIG. 35, a preferred neural network 349 for use at step 3314 will be further described.

[0499] In a preferred embodiment, the neural network is a recurrent neural network (RNN) applying long short term memory (LSTM) modules. In general, recurrent networks accept as input current data and data from previous node states. In this way the RNN projection of future target paths is more accurate than other more simple feed forward neural networks. Each LSTM module implements a set of gates. The gates propagate data by using the Sigmoid function,  $\sigma$ , as will be further described.

[0500] Neural network 349 includes input layer 351, node layer 353 and output layer 355. Node layer 353 further comprises nodes N1, N2 and N3. In this example, the input layer includes the positional data  $x_t$ ,  $y_t$  and  $z_t$  at time “t”. The output layer comprises a predicted position,  $x_{t+1}$ ,  $y_{t+1}$  and  $z_{t+1}$ , at time t+1.

[0501] In use, the network is trained by the positional data available in the path array. The positional data is the actual string of points in cartesian coordinates at each time “t” observed by stereo camera 925 for many thousands of separate target flights, each originating from the same point, in this case the high house. The lead path is calculated by submitting the actual path 330 of the actual live target 328 at point “B,” at time “t”, into input layer 351 and extracting from the output layer the appropriate future position of the target at point C, at time t+1. The time step t+1 is the lead time required for the target to reach the lead distance based on a ballistic table for the weapon, as previously described.

[0502] Referring then to FIG. 36, a preferred embodiment of the LSTM network node structure 360 will be described. Each of nodes N1, N2 and N3 comprises a separate instance of the LSTM network node structure 360.

[0503] LSTM network node structure 360 comprises module 357, module 382, and module 356, operatively connected by signal flow 371, 372, 373 and 374. Signal flow 372 comprises previous cell state 358, denoted in the Figure as  $C_{t+1}$ . Signal flow 371 comprises previous cell value 380, denoted in the drawing as  $h_{t-1}$ . Signal flow 373 comprises current cell value 359, denoted as  $h_t$ . Signal flow 374 comprises current cell state 375, denoted in the drawing as C.

[0504] Module 357 further comprises input data 361,  $x_{t-1}$ , previous cell state 358,  $C_{t-1}$ , and an output of previous cell value 380, or output value  $h_{t-1}$ .

[0505] Module 382 likewise comprises input data 381,  $x_t$ , current cell state 375, C, and an output of current cell value 359, or output value  $h_t$ .

[0506] Likewise, module 356 comprises input data 362,  $x_{t+1}$ , future cell state 377,  $C_{t+1}$ , and an output of future cell value 376, or output value  $h_{t+1}$ .

[0507] Each of the modules functions in a similar way. Therefore, as an example, module 382 will be described.

[0508] Generally, the LSTM network node will output a new value “ $h_t$ ” based on an previous cell value “ $h_{t-1}$ ” and a new signal “ $x_t$ ”. To prevent information overflow, gates which are comprised of Sigmoid functions and hyperbolic tangent functions are employed. The Sigmoid functions assign weights that vary between 0 and 1. A value of 1 will allow the data to flow through the gate unimpeded while a value of 0 will stop the data from exiting the gate. The hyperbolic tangent functions filter the data between -1 and 1.

[0509] For example, module 382 includes forget gate 364 and input gate 366.

[0510] The value  $f_t$ , the output of forget gate 364, can be described as

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad \text{Eq. 34}$$

where  $W_f$  is a gate weight and  $b_f$  is a bias.

[0511] Input gate 366 determines the next values that will be stored in the new state value  $C_t$ .  $C_t$  is multiplied a scaling factor,  $i_t$ . The factor  $i_t$  can be described as

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad \text{Eq. 35}$$

[0512] where  $W_i$  is a weight and  $b_i$  is a bias.

[0513] A new vector of cell "candidates" is described as  $C_t$  and can be described as.

$$\tilde{C}_t = \tan h(W_c[h_{t-1}, x_t] + b_c) \quad \text{Eq. 36}$$

where  $W_c$  is a weight and  $b_c$  is a bias.

[0514] The new cell value  $C_t$  is described as

$$C_t = f_t * [C_{t-1} + i_t] * \tilde{C}_t \quad \text{Eq. 37}$$

[0515] The current cell value  $h_t$ , 359 is given as

$$h_t = \sigma_t * \tan h(C_t) \quad \text{Eq. 38}$$

Where:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad \text{Eq. 39}$$

$W_f$ ,  $W_i$ ,  $W_c$  and  $W_o$  form a coefficient matrix and,  $b_f$ ,  $b_i$ ,  $b_c$  and  $b_o$  form a bias matrix.  $\sigma$  is the Sigmoid function. Likewise,  $\tan h$  denotes the hyperbolic tangent function.

[0516] Normalization of the path data is necessary before using it to train the LSTM network. Min-Max normalization is a linear strategy. It transforms the features of the data to values between 0 and 1.

[0517] In a preferred embodiment the RNN is written with Keras, an open source neural network library written in Python. Preferably, Keras is run on top of the Microsoft Cognitive Toolkit and CNTK, available from Microsoft of Redmond, Wash.

[0518] Referring to FIG. 37A, the architecture of an exemplary embodiment of tactical unit 3710 will be described. Exemplary tactical unit 3710 is comprised of headset module 3810, weapon module 3820 and targeting module 3830.

[0519] The three modules each have their own processor and communicate via a combination of local area networks. The headset module communicates with the targeting module through a hardwired bus. The weapon module communicates with the targeting module through a wireless connection. All wireless communication in the system is encrypted, in a preferred embodiment. In a preferred embodiment, a symmetric cypher is used to promote rapid data transfer rates.

[0520] In general, headset module 3810 is responsible for gathering visual information from internal facing and external facing cameras and information from the targeting module, and then processing and displaying that information to the user, on a dedicated augmented reality display, as will be further described.

[0521] In general, weapon module 3820 is responsible for gathering positional, ranging and firing data from the weapon and communicating it to the targeting module, as will be further described.

[0522] In general, targeting module 3830 is responsible for gathering data from a GPS transceiver, an IMU, a laser

range finder and a compass and then calculating and communicating target paths and the relative positions of the weapon, headset and other remote units, as will be further described.

[0523] Referring to FIG. 37B, an overview of the operation of system 3700 will be described. System 3700 includes a single tactical unit 3721, including weapon 3722, operating in a tactical theatre. Tactical unit 3721 operates in a cartesian coordinate system 3701 having origin 3702 and three axes, x, y and z. The x axis is aligned with the west east cardinal directions of windrose 3722. The y axis is aligned with the north to south cardinal directions of windrose 3722. The z axis is vertical.

[0524] Target 3719 moves in the coordinate system along path 3711. While traveling on the path, target 3719 traverses positions 3712, 3713, 3714 and 3715.

[0525] Weapon 3722 is shown in two positions, 3722a and 3722b. In both positions the weapon is trained on the target. In position 3722a, the range to target between tactical unit 3721 and position 3712 of target 3719 is shown as 3704. In position 3722b, the range to target between target 3719 at position 3713 is shown as 3706.

[0526] Virtual laser 3707 is shown projected from weapon 3722 to target 3719 at position 3714. The virtual laser is displayed and pointed at the target.

[0527] The movement of the weapon is used to determine the trajectory of the target. The trajectory of the target is then reduced to a target path equation. The target path equation is used to predict the position of phantom 3715 which is then translated and rotated to account for changes in position of the unit which take place after the path equation is calculated but before the shot is fired.

[0528] In a preferred embodiment, the shooter aims virtual laser 3707 at the phantom target as it moves from position 3714 to position 3715 in front of the target before triggering the shot.

[0529] Virtual tracer 3708 is shown between weapon 3722 and target 3719 at position 3715. The virtual tracer is displayed after the shot is fired and before the target is hit or missed.

[0530] Referring to FIG. 38A, system 3701 architecture will be described. System 3701 includes a plurality of remote units, 3710, 3720, 3730 and 3740. Each of the remote units supports a fast, bidirectional wireless data connection with each of the other remote units via a local area network or a wide area network, as will be further described. The number of remote units can be different than shown. In a preferred embodiment, the number of remote units is best suited to a squad or section of between 4 and 10 participants. However, other embodiments, the number of remote units can accommodate a platoon of between 16 and 40 participants. Each remote unit in the system of remote units is of identical architecture in these embodiments. However, in other embodiments the architecture of the remote units can differ to accommodate specialization for different tactical assignments.

[0531] A remote unit can also serve as a sentinel and spotter for other remote units, tracking the target from a concealed location, but not firing on it.

[0532] Multiple remote units in a tactical theatre is important because when any one remote unit triggers a shot, the user at that remote unit loses track of the target when he shoots. However, the other remote units do not loose track

of the target. Hence, the other remote units can be essential in reacquiring a target track after it has been obscured from the shooters vantage point.

[0533] Referring to FIG. 38B an overview of the operation of system 3750 will be described.

[0534] As can be seen from the drawing, remote unit 1 and remote unit 2 are positioned in a cartesian coordinate system having an origin 3790 an x and pitch axis, a y and roll axis, and a z and yaw axis, as shown. As shown in windrose 3791, the x axis lies west to east and the y axis lies north to south. The z axis is vertical.

[0535] Remote unit 1 moves from position 3752 to position 3754 along path 3751. At position 3752, remote unit 1 has weapon position 3753. At position 3754, remote unit 1 has weapon position 3755.

[0536] Remote unit 2 moves from position 3756 to position 3758 along path 3760. At position 3756, remote unit 2 has weapon position 3757. At position 3758, remote unit 2 has weapon position 3759.

[0537] Target 3769 moves along target path 3770 from position 3772 to position 3774. At position 3772, the target has a range 3784 from remote unit 1. At position 3774, the target has a range 3782 from remote unit 1.

[0538] Remote unit 1 derives a target path from its relative positions 3752 and 3754, weapon positions 3753 and 3755 and ranges 3784 and 3782, as will be further described. Remote unit 1 projects phantom target 3776 ahead of the target at lead distance 3778 along target path 3770, on its display as seen from its perspective.

[0539] Remote unit 1 displays virtual tracer 3790 on its display based on its position and the position of the weapon. The virtual tracer follows the path that a round of known caliber would assume given the launch angle of the weapon in position 3755.

[0540] Target path 3769 is transmitted to remote unit 2 by remote unit 1. The target path is translated by remote unit 2 into a proper display format, for remote unit 2. Remote unit 2 generates and displays virtual tracer 3786 and phantom position 3776 on its display, as seen from its perspective.

[0541] Either or both of remote unit 1 and remote unit 2 can trigger a directed shot along the virtual tracer once the virtual tracer and the phantom position are coincident on their respective displays.

[0542] Referring to FIG. 39, an exemplary embodiment of headset module 3900 will be described. The architecture of headset module 3900 comprises display processor 3912, strategically located in a tactical helmet shell, as will be further described. In a preferred embodiment, display processor 3912 is a Qualcomm Snapdragon 850 processor in combination with a companion holographic processing unit as found in the Microsoft HoloLens 2 headset. Other processors known in the art of similar capability will suffice. The display processor includes local internal memory sufficient to store and execute required display information, boot code and operational instructions.

[0543] Headset module 3900 further comprises targeting processor 3911. Targeting processor 3911 is preferably implemented by the AMD Radeon RX 5500M mobile graphics chip. The targeting processor is capable of accelerated geometric calculations such rotation and translation of vertices into different quadrant system including oversampling and interpolation techniques to produce high precision positional calculations. Targeting processor is further capable of high speed matrix and vector operations which

accommodate the neural network aspect of the invention, as will be later described. Targeting processor 3900 is operably connected to memory 3919. Memory 3919 is of sufficient size to store boot code, positioning information and operational code required to implement the functions of the targeting processor. In a preferred embodiment, a 120 gigabyte memory card has been found to be sufficient. Targeting processor 3900 is connected to the display processor 3912 through a hardwired internal high speed bus embedded in the tactical helmet (not shown).

[0544] Communications interface 3934 is operatively connected to targeting processor 3911. In a preferred embodiment, the communications interface can comprise a Bluetooth module, available from Intel, Part No. Intel 9260NGW IEEE 802.11ac Bluetooth 5.0—Wi-Fi/Bluetooth Combo Adapter. Communications interface 3934 can also include a wide area communication interface such as a cellular transceiver or a satellite radio transceiver. In a preferred embodiment, the cellular transceiver module is a TP-Link AC1300 PCIe wireless 2.4G/5G dual band wireless PCI express adapter. In a preferred embodiment, the satellite radio transceiver is an Iridium 9603 Two Way Satellite Data Transceiver. In a preferred embodiment, the communications interface is capable of encrypting and decrypting data sent and received. Encryption keys are stored in and deployed by the targeting processor for each new tactical theatre after an origin reset. In this way, data integrity and security is maintained between tactical operations.

[0545] Communications interface 3934 is operatively connected to antenna stack 3914. Antenna stack 3914 is strategically placed atop the tactical helmet, as will be further described. In a preferred embodiment, antenna stack 3914 includes, WiFi, Bluetooth, satellite and cellular antennas in a single removable module. Antenna stack 3914 also preferably comprises GPS transponder antenna such as the Symbol GPS Antenna 8508851K59 including a low noise amplifier.

[0546] Headset module 3900 further comprises stereoscopic camera 3916 operatively connected to targeting processor 3914. In a preferred embodiment, stereoscopic camera 3916 is the Mynt Eye S stereoscopic camera flat board module available from Slightech, Inc. of Beijing, China. The stereoscopic camera is capable of 60 frames per second depth map resolution of 752×480 pixels. Accurate depth sensing is provided between about 0.5 and about 20 meters. In a preferred embodiment, stereoscopic camera 3916 also provides infrared capability with a field of view of 120° horizontal by 75° vertical. The unit provides frame synchronization accuracy of less than 1 millisecond.

[0547] Display processor 3912 is operatively connected to internal camera 3917. The internal camera is focused on and constantly records movements of the eyes of the user during tactical operations after shouldering of the weapon.

[0548] Targeting processor 3914 is further connected to range finder 3918. In a preferred embodiment, range finder 3918 is forward mounted on the tactical helmet, as will be further described. In a preferred embodiment, the laser range finder is the LRF 3013 available from Safran Vectronix AG of Heerbrugg, Switzerland. The laser range finder includes a range capability of up to 3 kilometers with a typical accuracy of about 0.75 meters.

[0549] Tactical processor 3911 is further operatively connected to GPS transceiver 3920. In a preferred embodiment, GPS module 3920 is Part No. 511-TESEO-LIV3F available from STMicroelectronics.

[0550] Tactical processor 3911 is further operatively connected to altimeter 3922, compass 3924, accelerometer 3926 and gyroscope 3925. In a preferred embodiment, the altimeter, compass, accelerometer and gyroscope are all provided in an onboard IMU module available from Vectornav Part No. VN-100IMU/AHRS which comprises an attitude and heading reference system including a 3-axis accelerometer, a 3-axis gyro, a 3-axis magnetic sensor, and a barometric pressure sensor. In use, real time 3-D orientation positions of the remote unit are continually transmitted to tactical processor 3911, when the unit is activated, at approximately 800 Hz. A 0.5° static pitch/roll capability is provided, as well as a 1° dynamic pitch/roll capability. The internal gyro provides a 5° per hour in run bias. Data is transmitted to the processor at approximately 800 Hz. The accelerometer provides a range of ±16 grams. The gyroscope provides a tolerance of ±2000° per second.

[0551] Display processor 3912 is operatively connected to display 3928. In a preferred embodiment, the display is a pair of see through holographic lenses, positioned in front of the users eyes, providing 2.3 megapixel widescreen capability. The preferred display is provided in the Microsoft HoloLens 2 system and Integrated Visual Augmentation System (“IVAS”), both available from Microsoft.

[0552] Microphone 3930 is operatively connected to display processor 3912 for input of voice commands.

[0553] The headset module is powered by onboard power supply 3932. In a preferred embodiment, the onboard power supply is portable lithium ion battery pack attached to the headset module or carried by the user.

[0554] Referring to FIGS. 40A and 40B, a preferred embodiment of the tactical helmet unit will be further described. Headset module 3810 and targeting module 3830 are resident on carbon fiber tactical helmet shell 4000 worn by user 4020. In a preferred embodiment, the carbon fiber helmet shell is available as the EXFIL Carbon Helmet, Zorbium Liner, Part No. 71-Z41S-B31, available from Opticsplanet.com. In another preferred embodiment, the carbon fiber helmet shell is Team Wendy EXFIL Ballistic Helmet, Rail, Part No. 73-R3-41S-E31 available from Opticsplanet.com.

[0555] Targeting module 3830 is mounted at the rear of the tactical helmet and is removable. The removable nature of the targeting module is important to support quick correction of malfunctions in the field. Further, the targeting module includes watertight seal 4012, which allows the unit to be completely submersible without effecting operation.

[0556] In a preferred embodiment, the targeting processor, display processor, GPS transponder, altimeter, compass, accelerometer, gyroscope and communications interface are all hermetically sealed in targeting module 3830. In a preferred embodiment, the targeting module is encapsulated in epoxy resin and is removably attached to the tactical helmet by a single mechanical toggle (not shown).

[0557] Antenna stack 3914 is optimally positioned on top of the tactical helmet. In a preferred embodiment the antenna stack is removable and is encased by a rearward sloping attachment shroud 4014, which provides for deflection of debris and brush during tactical operations. The removable

nature of the antenna stack is important to allow quick reconfiguration of the helmet and correction of antenna malfunctions in the field.

[0558] Range finder 3918 is forward mounted on the tactical helmet and protect by rearward sloping shroud 4016.

[0559] Stereoscopic camera 3916 is forward mounted on the tactical helmet above display 3928 and is positioned to view an outward facing direction generally parallel with line of sight 4010. Forward shroud 4018 is permanently affixed to the tactical helmet positioned in front of the user's eyes along line of sight 4010. Line of sight 4010 is centrally positioned to enable a view of external environment and a target through display 3928 by the user. Display 3928 observes a local coordinate system 3929 of “x” in the horizontal direction relative to the display and “y” in the vertical direction relative to the display. Internal facing camera 4011 is affixed inside the forward shroud positioned to view each of the users eyes to readily identify the gaze ray of the line of sight.

[0560] Referring to then FIG. 41A, an example of weapon module 3820 will be described.

[0561] Weapon module 3820 includes processor card 4112. In a preferred embodiment, processor card 4112 is a Raspberry Pi 3, Model B available from Digikey. Processor card 4112 is operatively connected to memory 4113.

[0562] Antenna stack 4114 is operatively connected to processor card 4112. Processor card 4112 is further connected to laser range finder 4116. In a preferred embodiment, the laser range finder is the LRF 3013 available from Safran Vectronix AG of Heerbrugg, Switzerland. The laser range finder includes a range capability of up to 3 kilometers with a typical accuracy of about 0.75 meters.

[0563] Weapon module 3820 further comprises communications interface 4122. The communications interface allows encrypted communication between the targeting module and the weapon module during tactical operations. In a preferred embodiment, the communications interface can comprise a Bluetooth module, available from Intel, Part No. Intel 9260NGW IEEE 802.11ac Bluetooth 5.0—Wi-Fi/Bluetooth Combo Adapter. Communications interface 3934 can also include a wide area communication interface such as a cellular transceiver or a satellite radio transceiver. In a preferred embodiment, the cellular transceiver module is a TP-Link AC1300 PCIe wireless 2.4G/5G dual band wireless PCI express adapter. In a preferred embodiment, the satellite communications interface is an Iridium 9603 Two Way Satellite Data Transceiver.

[0564] Weapon module 3820 further comprises power supply 4124. In a preferred embodiment, power supply 4124 is a lithium ion battery located in the stock of the weapon, as will be further described.

[0565] Weapon module 3820 further comprises forward IMU 4126 operatively connected to processor card 4112. Processor card 4112 is further connected to rear IMU 4128 and compass 4130. In a preferred embodiment, forward IMU 4126 and rearward IMU 4128 are each an IMU module available from Vectornav Part No. VN-100IMU/AHRS which comprises an attitude and heading reference system including a 3-axis accelerometer, a 3-axis gyro, a 3-axis magnetic sensor, and a barometric pressure sensor. The compass function of rear IMU can be used by the system to provide the direction orientation of the weapon in the x, z plane of the cartesian coordinate system.



[0566] Referring then to FIG. 41B, a preferred embodiment of processor card 4112 and memory 4113 are described. Processor card 4112 includes processor 4153. In a preferred embodiment, processor 4153 is a Broadcom BCM2837 1.2 GHz Quad-Core processor.

[0567] Two USB ports 4154 and 4155 are operatively connected to processor 4153. USB port 4154 is connected to Bluetooth module 4156 which provides a short-range wireless networking connection for the communications interface. The Bluetooth module in a preferred embodiment is Bluetooth 4.0 USB Module (v2.1 Back-Compatible), Product ID 1327, available from Ada Fruit at adafruit.com. The Bluetooth module includes antenna 4159, Positioned in antenna stack 4114. USB port 4155 is operatively connected to laser range finder 4116.

[0568] Processor 4153 is connected to general purpose input output pins 4160. In a preferred embodiment, data from IMU 4126 and IMU 4128 is received through these pins.

[0569] Processor 4153 is connected to memory card 4158 via access slot 4161. Code resident on the memory card is used to boot the processor and perform the operations necessary to control its operation, as will be further described.

[0570] Referring then to FIG. 42A, a preferred implementation of weapon 4200 will be described. Antenna stack 4114 can be seen to be positioned atop the picatinny rail of the weapon adjacent stock 4202. The position atop the rail allows for reliable wireless communication with the targeting processor of the headset module through the communications interface.

[0571] Communications interface 4122 is located remotely from the antenna stack in stock 4202 and is connected to the antenna stack by a coaxial bundle (not shown). Stock 4202 also houses processor 4112, power supply 4124 and rear IMU 4128. In a preferred embodiment, the stock is sealed in epoxy resin to protect the components from shock and moisture.

[0572] Forward IMU 4126 is positioned in forearm 4204 adjacent barrel 4206 and iron sights 4208. Laser range finder 4116 is positioned atop the weapon on the picatinny rail as shown. In a preferred embodiment, laser range finder 4116 is side mounted thereby avoiding interference with a line of sight through iron sights 4208. These components are connected to processor card 4112 by a hard-wired ribbon cable bus (not shown).

[0573] Referring then to FIG. 42B, a top view of weapon 4200 will be described. Weapon 4200 includes forward visual sight marker 4250 positioned adjacent iron sights 4208 of barrel 4206. Weapon 4200 further includes rearward visual sight marker 4252 positioned adjacent antenna stack 4114. Both visual sight markers are in the visual line of sight of stereoscopic camera 3916 and the view of the display of the headset module, when the weapon is shouldered. In an alternate embodiment, the orientation of the barrel can be determined and tracked from the IR sensors and forward facing cameras of the HoloLens 2 headset.

[0574] Referring to FIG. 43A, the functions of a single tactical unit operating in a preferred embodiment of system 4300, will be described. In this preferred embodiment, a single tactical unit is identified as an example and operates alone in the tactical theatre. A spotter (not shown) may also be provided.

[0575] At step 4374, the weapon module senses a shouldering event. In this embodiment, the forward IMU sends a signal to the targeting module indicating a rapid succession of position changes of the weapon in the vertical direction, which is interpreted to be a shouldering event.

[0576] At step 4376, the unit sets an origin of the tactical theatre in cartesian coordinates. In this embodiment, the origin is set at the current location of the rear IMU of the unit. Other origin points may be used, so long as the origin remains fixed during the tactical maneuver. All positional changes are sensed from this origin by the IMU of the tactical processor and/or the forward IMU and rear IMU of the weapon module.

[0577] At step 4378, the compass is read to determine the cardinal directions in order to set the cartesian axes. The x axis is assigned east west. The y axis is assigned north south. The z axis is assumed vertical.

[0578] At step 4380, the unit determines range to target distance. In this embodiment, the target is assumed to be a physical entity on which the weapon is trained. The range to target distance is read from the laser range finder of the weapon while the weapon is trained on the target. In an alternate embodiment, the range to target distance is read from the stereoscopic camera. In another preferred embodiment, the range to target distance is read from the laser range finder of the tactical helmet.

[0579] At step 4381, the weapon position is determined, as will be further described.

[0580] At step 4382, the virtual laser position is calculated. The virtual laser moves with the weapon as the weapon moves, in order to mimic the appearance of a real world targeting laser. The position of the virtual laser must account for shot drop at range in order to accurately predict the position of the round after any period of time after launch. The virtual laser is projected as a straight line from the barrel of the weapon extending to the shot drop position, for a distance equal to or surpassing the range to target distance. The position of the weapon is used to determine the correct orientation of the virtual laser with respect to the weapon, as will be further described. In a preferred embodiment, the virtual laser position is calculated and animated by the display processor by making a function call to the Microsoft HoloLens 2, Integrated Visual Augmentation System ("IVAS"), available from Microsoft, or Unity 3D gaming engine.

[0581] At step 4383, the display processor of the remote unit displays the virtual laser entity on the augmented reality display.

[0582] At step 4384, target path is calculated. The flight path of a moving object can be segmented into discrete pieces by using the range finder, the tracked movement of the gun barrel and the displays frame rate acting as a clock. Each frame defines a discrete time period and a distance along the path. With a change in distance over time, speed can be derived as can rate of change, acceleration and deceleration. Since images are typically generated at 60 or 90 frames per second, the trajectory can be segmented at the same rate, and used to predict and display the aim point in terms of "frames" ahead.

[0583] In general, the target path is derived from the position, velocity and acceleration of the weapon, evaluated as a rigid body, relative to the origin. The position, velocity and acceleration of the weapon as a rigid body are calculated by recording the positions of the forward IMU and the rear

IMU at discrete time intervals, called “frames.” A virtual ray object is then calculated as a “projection” coaxial with the axis of the barrel. The axis of the barrel is approximated from the positions of the rear IMU and the forward IMU. The virtual ray object is then extended mathematically from the position of the forward IMU away from the rear IMU, by a distance equivalent to the range to target to a termination point. The termination point is assumed to be the target position. The termination point moves as the weapon moves and, at the distance of range to target, is assumed to be the target path. An equation of the target path may be derived for use in predicting future movements of the target.

**[0584]** In a preferred embodiment, the target path equation is derived path in spherical coordinates from changes of the weapon position over time, for at least two time periods, and the range to target data. In a simplified example, where the range to target is constant, the following equations of motion are employed to derive r, v and a in spherical coordinates:

$$r = r(t) = r\hat{e}_r \tag{Eq. 40}$$

$$v = v\hat{e}_r + r\frac{d\theta}{dt}\hat{e}_\theta + r\frac{d\varphi}{dt}\sin\theta\hat{e}_\varphi \tag{Eq. 41}$$

$$a = \left( a - r\left(\frac{d\theta}{dt}\right)^2 - r\left(\frac{d\varphi}{dt}\right)^2\sin^2\theta \right)\hat{e}_r + \left( r\frac{d^2\theta}{dt^2} + 2v\frac{d\theta}{dt} - r\left(\frac{d\varphi}{dt}\right)^2\sin\theta\cos\theta \right)\hat{e}_\theta + \left( r\frac{d^2\varphi}{dt^2}\sin\theta + 2v\frac{d\varphi}{dt}\sin\theta + 2r\frac{d\theta}{dt}\frac{d\varphi}{dt}\cos\theta \right)\hat{e}_\varphi \tag{Eq. 42}$$

Where:

**[0585]** r is the range to target taken along the axis of the weapon from the forward IMU position;

**[0586]** v=target velocity;

**[0587]** a=target acceleration;

**[0588]** θ is the angle of the weapon from the z axis;

**[0589]** φ is the angle of the weapon from the x axis;

**[0590]** ê=vector notation.

$$\frac{d\theta}{dt} = \frac{\theta_1 - \theta_2}{t_1 - t_2} \tag{Eq. 43}$$

$$\frac{d\varphi}{dt} = \frac{\varphi_1 - \varphi_2}{t_1 - t_2} \tag{Eq. 44}$$

Where:

**[0591]** θ<sub>1</sub> and φ<sub>1</sub> are taken at t<sub>1</sub>;

**[0592]** θ<sub>2</sub> and φ<sub>2</sub> are taken at t<sub>2</sub>.

**[0593]** The target path between time t and t<sub>2</sub> may be considered a “segment.” The time between t<sub>1</sub> and t<sub>2</sub> may be considered a “frame.”

**[0594]** The target position, velocity and acceleration of the weapon may then be derived for successive segments in successive frames.

**[0595]** In a preferred embodiment, the mathematical projection of the virtual ray object takes the form of a function

call from the targeting processor to the Microsoft HoloLens 2 or the Unity 3D gaming engine, available from Unity Labs or similar.

**[0596]** At step 4386, lead distance is calculated. The lead position must account for the distance that the target will move during a period of time between the trigger event and the arrival of the round at the target position. The lead must also account for the shot drop in the round between the triggering event and arrival of the round at the position of the target.

**[0597]** In a preferred embodiment, to accurately represent correct lead on real targets in an AR headset, the aim point is animated at a frame rate, the gun barrel is motion-tracked and the range is updated on the target at approximately the same rate as the frame rate of the animation.

**[0598]** Once the correct lead is established, the system ceases updating the target path from the weapon position and animation of the phantom proceeds using only the target path equation, which is translated and rotated to account for changes in the position of the unit. In one embodiment, the position of the unit is drawn from the IMU of the targeting module.

**[0599]** In another preferred embodiment, lead is calculated by consulting a ballistics table stored in memory of the targeting processor of the unit. The ballistics table provides the shot drop for each range and for each potential type of weapon and round used by the unit. The velocity of each potential round used by the unit is also provided in the ballistics table. Given the velocity of the round and the range to target, the elapsed time between a trigger event and arrival of the round at the target can be calculated as follows:

$$t = \frac{r}{v} \tag{Eq. 45}$$

Where:

**[0600]** t=time to target

**[0601]** r=range

**[0602]** v=velocity

**[0603]** The time to target is then substituted into the target path equation, which is solved for the position of the target at the ballistic intercept point. The lead distance is the difference between the position of the target at the time the shot is fired to the position of the target at the time of ballistic intercept. Once the target path and the lead distance are derived, the use of current weapon position data to update the target path equation ceases, and the display relies on the then calculated target path equation for animation of the phantom target image. The display of the virtual laser and the virtual tracer remain dependent on weapon tracking data, to determine their positions in the display.

**[0604]** At step 4391, a phantom target is displayed at the lead position.

**[0605]** At step 4392, the display processor compares the phantom target position to the virtual laser position to determine a coincidence event. A “coincidence event” is an overlap between the graphic displays of the virtual laser and the phantom target.

**[0606]** At step 4393, the display processor receives a coincidence event signal and sends it to the targeting processor.

[0607] At step 4394, the display processor generates and displays a fire alert. A shot is assumed to be triggered at or near the time of the display of the fire alert.

[0608] At step 4395, a virtual tracer path is calculated. In a preferred embodiment, the virtual tracer path is determined by calculating a predicted shot trajectory given the current position and launch angle of the weapon, translated and rotated to account for the position, velocity and acceleration of the unit. The virtual laser path is stored as an equation which is sent to the display processor of the unit. The virtual tracer path remains dependent on constantly updated weapon position data.

[0609] Upon indication of a shot being fired, at step 4396, the virtual tracer is displayed by the display processor. The virtual tracer is displayed along its path as a generally hyperbolic broken bright line.

[0610] At step 4397, the display processor monitors the display for a hit condition. A "hit condition" is defined for the display processor as a presumed hit of the target being tracked. A "hit condition" event in one embodiment occurs when the laser range finder of the weapon reports an infinity value for the range to target distance or the IVAS or similar camera records the reflection of a spotter round or incendiary round hitting the target or a spotter verifies a hit. A hit condition is logged upon occurrence. If a hit condition event is not reported within a shot predetermined time frame, the shot is logged as a "miss."

[0611] At step 4398, the display processor records a hit (or miss) incident and transmits it to the targeting processor.

[0612] At step 4399, the origin point is cleared.

[0613] Referring to FIG. 43B, the functions of a plurality of remote units operating in a preferred embodiment of system 4301, will be described. In this preferred embodiment, two remote units are identified as examples and operate in the tactical theatre. However, in other embodiments communications and functions such as those described can occur between a larger number of remote units, all in communication with each other through the local area network or a wide area network. In this example, remote unit 4390 initially acts as a "spotter" for remote unit 4395.

[0614] At step 4302, remote unit 4390 sets an origin of the tactical theatre in the cartesian coordinates. In this embodiment, the origin is taken at the GPS location and elevation of the tactical processor of remote unit 4390. At step 4303, the compass is read to determine the cardinal directions. The x axis is assigned east west. The y axis is assigned north south. The z axis is assumed vertical.

[0615] At step 4304, the origin position is sent to remote unit 4395 through the local area network. At step 4305, remote unit 4395 stores the origin and adopts it as the origin of the tactical theatre. In a preferred embodiment, the GPS location of remote unit 4395 is read and the origin is translated and rotated to account for the difference in position between remote unit 4390 and remote unit 4395. At step 4306, remote unit 4395 reads its internal compass for cardinal directions and assigns them to the x and y axes, as described. The vertical direction is assigned the z axis. In one preferred embodiment, the cardinal directions are synchronized and matched as between the remote units.

[0616] At step 4307, remote unit 4390 identifies a target to track. The target is "identified" by the system when the user shoulders the weapon and assumes a stable pattern of weapon movement. In a preferred embodiment, a stable

pattern of weapon movement occurs when the weapon is directed along a continuous path for greater than about 750 milliseconds.

[0617] At step 4308, remote unit 4390 initiates a target tracking routine. The target tracking routine implements a sequential sampling of range to target distance, weapon position and unit position. Each sampling of data points is taken simultaneously from the range finder of the weapon and the forward IMU and rear IMU, repeatedly at discrete time intervals or frames. The data is stored in a table for later use, indexed by time or frame rate. In a preferred embodiment, the discrete steps are about 25 milliseconds apart. In other embodiments, a 60 to 90 second frame rate is used. The frame rate may be synchronized to that of the display processor.

[0618] At step 4309, range to target is read from the laser range finder of remote unit 4390. At step 4311, the weapon position is read, as will be further described. At step 4313, remote unit 4390 determines its own current position in the operating theatre, relative to the origin and the cartesian coordinate system, using repeated calls to the onboard IMU.

[0619] At step 4315, the target path equation relative to the origin is calculated, as previously described. In this embodiment, the apparent target path as seen from the position of remote unit 4390 must be appropriately translated and rotated by remote unit 4390 to derive the target path relative to the origin.

[0620] At step 4317, the target path equation derived by remote unit 4390 is transmitted to remote unit 4395 via the wireless network.

[0621] At step 4323, the targeting processor of remote unit 4395 polls the laser range finder attached to its weapon and calculates the distance to target as if it were a stand-alone unit, as previously described.

[0622] At step 4324, the targeting processor of remote unit 4395 calculates the time to target given the distance to target. At step 4325, shot drop at range is determined. The shot drop is derived for the particular weapon and round being used, from the ballistic table.

[0623] At step 4326, remote unit 4395 determines its weapon position, as will be further described.

[0624] At step 4327, the display processor of remote unit 4395 displays a virtual laser image on the augmented reality display. The virtual laser image is displayed as a straight bright line.

[0625] At step 4329, remote unit 4395 determines its position, velocity and acceleration. In a preferred embodiment, the targeting processor of remote unit 4395 polls the onboard IMU to determine instantaneous position, velocity and acceleration with respect to the origin. In an alternate embodiment, the targeting processor polls the onboard GPS transceiver to determine position at least several points in order to obtain position and vector values for velocity and acceleration.

[0626] At step 4331, the targeting processor of remote unit 4395 translates and rotates the target path equation received from remote unit 4390 for proper display from the perspective of remote unit 4395.

[0627] At step 4335, the targeting processor of remote unit 4395 calculates the lead distance for the particular round in the weapon, based on the round velocity, shot drop, range to target, translated target path, and target position, velocity and acceleration and sends it to the display processor. In a preferred embodiment, the lead position is calculated by

determining the time that the round will take to reach the target, and then extrapolating the path of the target from the target path equation ahead of the target for this period of time. The phantom target position is raised in the z direction by the shot drop distance to more precisely indicate the ballistic interrupt point.

[0628] In another embodiment, the phantom is advanced along the path, in predicted segments, at the same frame rate as the display is sampled, for the number of frames as would be required for the round to travel from the weapon to the target.

[0629] At step 4339, the display processor of remote unit 4395 displays a phantom image ahead of the target along the target path at the lead distance and accounting for shot drop.

[0630] At step 4341, the display processor compares the phantom position to the position of the virtual laser. In a preferred embodiment, a “coincident” function call is made to the HoloLens 2 system to compare for overlap between the two visual elements of the virtual laser and the phantom image.

[0631] At step 4345, when the images of the phantom target and the virtual laser are coincident, the display processor reports a coincidence event at step 4343 to the targeting processor.

[0632] At step 4345, the targeting processor generates a fire alert message. At step 4347, remote unit 4395 displays a fire alert message on the display of the headset module. At step 4348, the forward IMU senses a shot signal and transmits it to the targeting module. A shot is assumed to be triggered at or near the time of the display of the fire alert message.

[0633] At step 4349, remote unit 4395 sends the shot signal message to remote unit 4390 via the local area network. At step 4351, the display processor of remote unit 4390 displays the shot signal message and an indicator of which remote unit sent the message for the user on the augmented reality display of remote unit 4390.

[0634] At step 4352, remote unit 4395 generates and displays a virtual tracer image along the shot path, as previously described.

[0635] At step 4353, remote unit 4395 monitors for and records a hit or miss condition.

[0636] Optionally, at step 4354, remote unit 4390 resets the origin.

[0637] At step 4355, the targeting processor of remote unit 4390 calculates time to target, as previously described. At step 4356, remote unit 4390 calculates the shot position at range by consulting a ballistics table to determine shot drop at range for the particular round being fired.

[0638] At step 4357, remote unit 4390 calculates the shot path based on the weapon position, velocity and acceleration, as previously described.

[0639] At step 4359, the display processor of remote unit 4390 displays a virtual laser the display of remote unit 4390, as previously described.

[0640] At step 4361, targeting processor of remote unit 4390 calculates a lead distance along the target path, relative to the display of remote unit 4390, as previously described.

[0641] At step 4363, the display processor of remote unit 4390 displays a phantom target, at the lead distance, along the target path, and accounting for shot drop as previously described.

[0642] At step 4365, the display processor of remote unit 4390 compares the displayed phantom position to the virtual

laser position, as previously described. At step 4367, the display processor of remote unit 4390 records a coincidence between the displayed phantom and the virtual laser.

[0643] At step 4368, upon receiving a coincidence condition, the targeting processor of remote unit 4390 generates a fire alert message. At step 4369, the display processor of remote unit 4390 displays the fire alert message on the display to the user. At step 4370, remote unit senses a shot signal. At step 4371, remote unit 4390 generates a shot signal message. At step 4372, remote unit 4390 sends the shot signal message to remote unit 4395, through the local area network. At step 4373, remote unit 4395 displays the shot signal message, along with an indicator that remote unit 4390 has fired.

[0644] At step 4373, remote unit 4390 generates and displays a virtual tracer, as previously described.

[0645] At step 4374, remote unit 4395 displays the shot signal.

[0646] At step 4375, remote unit 4390 records a hit or miss condition.

[0647] At step 4376, remote unit 4390 resets the origin.

[0648] Referring then to FIG. 44, method of determining weapon position for a remote unit 4400 will be described.

[0649] At step 4415, weapon processor 4409 polls the forward IMU sensor to derive forward IMU data. In a preferred embodiment, the forward IMU data comprises a position, velocity and acceleration of the forward end of the weapon relative to the origin.

[0650] At step 4417, weapon processor 4409 polls the rear IMU sensor to derive rear IMU data. In a preferred embodiment, the rear IMU data comprises a position, velocity and acceleration of the rear end of the weapon relative to the origin.

[0651] The forward IMU and rear IMU data pairs are taken repeatedly so that a weapon position may be accurately derived. The forward IMU and the rear IMU data pairs are individually time stamped so that they can be associated together for later use.

[0652] At step 4418, the data pairs are time stamped by appending a clock field to each data set including the current time. At step 4419, weapon processor 4409 sends the time stamped forward IMU and rear IMU data pairs to the targeting processor over the local area network.

[0653] At step 4420, targeting processor 4407 stores the forward IMU and the rear IMU data pairs.

[0654] At step 4425, display processor 4405 reads the position of the forward visual sight marker position, velocity and acceleration and stores then in memory. At step 4427, the display processor reads the position of the rear visual sight marker position, velocity and acceleration and stores then in memory. The process is repeated creating data pairs. The forward and rear visual sight marker data pairs are time stamped for later use.

[0655] At step 4428, the data sets are time stamped.

[0656] At step 4433, display processor 4405 sends the forward visual sight marker position data, and the rear visual sight marker position data pairs to targeting processor 4407. At step 4434, the data sets are stored.

[0657] At step 4435, the targeting processor determines the weapon position, velocity and acceleration from the IMU data. In a preferred embodiment, the processor determines the weapon position from the IMU data by establishing a vector between the rearward IMU position and the forward IMU position for each pairing of the standard data.

The two positions are assumed to be separate positions on the same rigid body. A weapon path equation is derived using kinematic equations, as will be further described.

[0658] At step 4437, targeting processor 4407 derives weapon position, velocity and acceleration from the visual sight marker data. In a preferred embodiment, a path is derived for the first position of the forward visual sight marker and the second position of the forward visual sight marker. Similarly, a rear visual sight marker path is derived between the first position of the rear visual sight marker and a second position of the rear visual sight marker. A vector is established between the rear visual sight marker position to the forward visual sight marker position for each pairing of time stamped data. The two positions are assumed to be separate positions on the same rigid body. A weapon path equation is derived.

[0659] In general, the weapon path equation is derived from the following kinematics equations:

[0660] The angular velocity of a rigid body B in a reference frame N is equal to the sum of the angular velocity of a rigid body D in N and the angular velocity of B with respect to D:

$${}^N\omega^B = {}^N\omega^D + \omega^B \tag{Eq. 46}$$

[0661] For any set of three points P, Q, and R, the position vector from P to R is the sum of the position vector from P to Q and the position vector from Q to R:

$$r^{PR} = r^{PQ} + r^{QR} \tag{Eq. 47}$$

[0662] The velocity of point P in reference frame N is defined as the time derivative in N of the position vector from O to P:

$$N_V P = \frac{N_d}{dt} (r^{OP}) \tag{Eq. 48}$$

Where:

[0663] O=the origin;

[0664] N=indicates that the derivative is taken in reference frame N.

[0665] The acceleration of point P in reference frame N is defined as the time derivative in N of its velocity:

$$N_a P = \frac{N_d}{dt} (N_V P) \tag{Eq. 49}$$

[0666] For two points P and Q that are fixed on a rigid body B, where B has an angular velocity  ${}^N\omega^B$  in the reference frame N, the velocity of Q in N can be expressed as a function of the velocity of P in N:

$$N_V Q = N_V P + {}^N\omega^B \times r^{PQ} \tag{Eq. 50}$$

[0667] By differentiating the equation for the Velocity of two points fixed on a rigid body in N with respect to time, the acceleration in reference frame N of a point Q fixed on a rigid body B can be expressed as:

$$N_a Q = N_a P + {}^N\omega^B \times ({}^N\omega^B \times r^{PQ}) + N\alpha_B \times r^{PQ} \tag{Eq. 51}$$

Where:

[0668]  ${}^N\alpha^B$  is the angular acceleration of B in the reference frame N.

[0669] In a specific case of these equations, the 3 reference points on the rigid body are assumed to be collinear, along the weapon barrel between the rear IMU position and the forward IMU position.

[0670] In another embodiment, recognition of barrel orientation and motion can be done by the HoloLens 2, Integrated Visual Augmentation System (“IVAS”) or Leap Motion tool available from the developer archive of Leap Motion of San Francisco, Calif.

[0671] In optional step 4439, the final weapon position, velocity and acceleration is determined by averaging the vector coordinates of the weapon position derived from IMU data and the vector coordinates of weapon position derived from the visual sight marker data.

[0672] At step 4440, the final weapon position, velocity, acceleration and the weapon path equation are reported for later use.

[0673] In an alternate preferred embodiment, steps 4415, 4417, 4419, and 4437 are optional. In an alternate preferred embodiment, steps 4425, 4427, 4428, 4433 and 4437 are optional.

[0674] Referring to FIGS. 45A, 45B and 45C, an example of the display of a single unit showing a phantom and a pull-away lead will be described.

[0675] Referring to FIG. 45A, display 4560 can view weapon 4598 trained on target 4597 moving along path 4596. Virtual laser 4599 can be seen directed to target 4597.

[0676] Phantom image 4592 is shown traversing path 4596 to an anticipated position ahead of the target by lead distance 4591.

[0677] Referring to FIG. 45B, display 4560 can now visualize weapon 4598 having “pulled-away” from target 4597 along direction 4593. Virtual laser 4599 can now be seen directed toward a coincident position with phantom display 4592. At this point, a fire signal is sent and shown in the display at 4594. When the shot is fired, virtual laser 4599 disappears and a virtual tracer is displayed, as will be further described.

[0678] Referring to FIG. 45C, a display of a virtual tracer and ballistic intercept display will be described.

[0679] Display 4560 shows weapon 4598 trained on target 4597. Target 4597 travels along path 4596. Virtual tracer 4593 is shown displayed from weapon 4598 along a generally hyperbolic path to a ballistic intercept of target 4597. The virtual tracer is activated upon a shot being triggered and disappears upon recording of a hit or miss incident.

[0680] Referring to FIGS. 45D and 45E, exemplary displays of remote unit 4390 and remote unit 4395, respectively, operating in the same tactical theatre, will be described.

[0681] Referring to FIG. 45D, display 4391 of remote unit 4390, as seen from the perspective of the user is shown. Weapon 4505 is shown directed toward phantom target 4509. Target 4507 proceeds along path 4502. Phantom target 4509 is shown at lead distance 4510 ahead actual target 4507. Virtual laser 4511 is shown coincident with phantom target 4509.

[0682] Referring to FIG. 45E, display 4396 of remote unit 4395, as seen from the perspective of the user, is shown. Weapon 4555 is shown directed at phantom target 4559. Target 4507 is shown proceeding along path 4503. Phantom

target **4559** is shown leading actual target **4507** at distance **4560**. Virtual laser **4561** is shown coincident with phantom target **4559**.

[**0683**] As can be seen, although the actual real-world trajectory of target **4507** in the cartesian coordinates is the same, that the display of the trajectory to remote unit **4390** differs from the display of the trajectory to remote unit **4395**. Similarly, there is a difference between lead distance **4510** and lead distance **4560** as seen from the perspective of remote unit **4390** and remote unit **4395**, respectively.

[**0684**] Referring to FIG. **46A**, the architecture of an alternate system embodiment **4600** will be described.

[**0685**] System **4600** comprises a plurality of remote units. In this example, the architecture shows remote units **4610**, **4620**, **4630** and **4640**. In other embodiments, there may be a greater or smaller number of remote units. Each of the remote units is configured as previously described. Each of the remote units communicates wirelessly with each of the other remote units through a local wireless network, as will be further described. All communications conducted wirelessly are encrypted in a preferred embodiment. A symmetric cypher is preferred to maximize encryption and decryption speed.

[**0686**] The addition of drones, spotters, remote and fixed cameras is important because when any one shooter or any plurality of shooters triggers a shot, they lose track of the target when the weapon is fired. However, the spotters, drones and fixed cameras will not lose track of the target that is otherwise obscured from one or more shooters vantage points.

[**0687**] Each of the remote units also communicates with tactical monitor **4645** through the wireless network. Tactical monitor **4645** further communicates with drone **4660** and fixed camera **4670** through a wireless network. In a preferred embodiment, drone **4660** and fixed camera **4670** operate in the same tactical theatre as the remote units. In other embodiments, a plurality of fixed cameras and a plurality of drones, all in communication with the tactical monitor are provided and are operational in the tactical theatre.

[**0688**] Tactical monitor **4645** is operatively connected to local database **4655** and neural network pattern processor **4650**.

[**0689**] Referring to FIG. **46B**, an overview of a preferred embodiment will be described.

[**0690**] The tactical theatre is defined by a cartesian axis with origin **4672** and axes defined by windrose **4673**. In the operational theatre, remote unit **4674** and remote unit **4676** are free to move in the x, y plane. Fixed camera **4678** maintains a fixed position in the cartesian system. Drone **4680** is free to move in three dimensions in the cartesian system.

[**0691**] Target **4682** moves from target position **4684** to target position **4685** along target path **4683**.

[**0692**] Remote unit **4674** has a range to target of **4690**. Remote unit **4676** has a range to target of **4691**. Fixed camera **4678** has a range to target of **4689**. Drone **4680** has a range to target of **4693**.

[**0693**] Remote unit **4676** displays virtual laser **4688** and a phantom **4686**. Phantom **4686** is displayed at lead distance **4692** ahead of target **4682** along target path **4683**.

[**0694**] Referring to FIG. **47**, drone **4660** will be further described.

[**0695**] Drone **4660** includes processor **4705**. Processor **4705** includes three concurrently running modules including

navigation module **4707**, flight management module **4709** and data communication management module **4711**. Processor **4705** also includes memory sufficient to store and process flight and positional instructions necessary to carry out its functions.

[**0696**] Navigation module **4707** is responsible for executing a predetermined flight path, according to a flight schedule in the tactical theatre. In an alternate embodiment, the navigation module receives flight path corrections from the processor responsive to a remote set of commands from the tactical monitor. Flight management module **4709** is responsible for activation and maintenance of motor speed, collision avoidance and in-flight stability. Data communication module is responsible for gathering, formatting and transmitting data from the IMU, GPS transponder, camera and range finder to the tactical monitor through the communication interface. The data communication module is also responsible for receiving and distributing course correction instructions and camera positioning instructions from the tactical monitor.

[**0697**] Processor **4705** is operatively connected to GPS transceiver **4720**.

[**0698**] Drone **4660** further comprises altimeter **4722**, compass **4724**, gyroscope **4726** and accelerometer **4728**. In a preferred embodiment, the altimeter, compass, gyroscope and accelerometer are contained in an internal IMU unit which communicates directly with the processor.

[**0699**] Drone **4660** further comprises communication interface **4730**. In a preferred embodiment, communications interface **4730** accommodates a wireless local area network and a wireless wide area network such as the Internet.

[**0700**] Drone **4660** is powered by power supply **4732**. In a preferred embodiment, power supply **4732** is a lithium ion battery power source capable of supplying approximately about 30 minutes flying time.

[**0701**] Processor **4705** is further connected to stereo camera **4734**. In a preferred embodiment, the stereo camera is mounted on a pan, tilt and zoom platform which communicates directly to the processor and can be remotely positioned by the tactical monitor, as will be further described.

[**0702**] Processor **4705** is further connected to laser range finder **4736**. In a preferred embodiment, the laser range finder is physically attached to the pan, tilt, zoom platform of the camera and is moved with the camera. The processor and communications interface are operatively connected to antenna stack **4738**, which is externally mounted on the drone airframe (not shown).

[**0703**] In a preferred embodiment, drone **4660** is the Yuneec H520-E90 Configurable Bundle available from Vertigo Drones of Webster, N.Y. In this embodiment, the drone is a six rotor configuration airframe, including direct communication with tactical monitor **4645** through mission control software, as will be further described. Drone **4660** is capable of carrying out predetermined flight plans, including both positional, rotation and altitude maneuvers. Live feed video transmissions, including position data, range data and GPS data are communicated directly and constantly to the tactical monitor via the communications interface, as will be further described.

[**0704**] Referring to FIG. **48**, fixed camera **4670** will be further described.

[**0705**] In a preferred embodiment, fixed camera **4670** includes processor **4805**. Processor **4805** is operatively connected to camera **4810**. Camera **4670** further is posi-

tioned on a motorized platform **4815** capable of pan, tilt and zoom functions, positioned locally by the processor according to commands from the tactical monitor.

**[0706]** Fixed camera **4670** further comprises communications interface **4820** operatively connected to processor **4805**. GPS transponder **4825** is optionally included in the fixed camera and is operatively connected to the processor and required antennas. If included, the GPS transponder communicates with the processor and the tactical monitor through the communications interface. The GPS transponder is included to allow periodic repositioning of the camera unit. In a preferred embodiment, fixed camera **4670** further comprises range finder **4830**, operatively connected to processor **4805**, and movable with the camera by the pan, tilt, zoom platform.

**[0707]** The communications interface is capable of communication with the tactical monitor through either or both a wireless local area network or wireless wide area network, such as the Internet.

**[0708]** In a preferred embodiment, the fixed camera is the military grade MX6 FLIR PTZ thermal imaging long range multi sensor pant tilt MWIR camera system, available from Sierra Pacific Innovations Corp. of Las Vegas, Nev. The camera is capable of LFIR thermal imaging at up to 55 kilometers distance. The fixed camera is further capable of laser range finding to approximately 50 millimeter tolerance.

**[0709]** Referring to FIG. **49**, a preferred method of operation of preferred system **4900**, will be described. In a preferred embodiment, tactical processor **4906**, display processor **4908** and weapon processor **4910** are all connected through a local area network. In a preferred embodiment, targeting processor **4906** is connected to tactical processor **4904** and AI processor **4902** through a wide area network. In this embodiment, the targeting processor, display processor and weapon processor are resident on a single remote unit.

**[0710]** At step **4911**, targeting processor **4906** sets the cartesian origin of the tactical theatre. In a preferred embodiment, the origin is set via GPS coordinates. At step **4916**, the origin coordinates are sent to the tactical processor.

**[0711]** At step **4912**, the tactical processor launches and positions the drone along a predetermined flight path. The drone processor coordinates local flight operations of the drone along the predetermined flight path.

**[0712]** At step **4913**, weapon processor **4910** reads the forward weapon IMU to determine movement. The movement is interpreted as a shoulder signal. At step **4914**, weapon processor **4910** sends the shoulder signal to targeting processor **4906**. At step **4915**, weapon processor **4910** forwards the shoulder signal to the display processor **4908**. At step **4917**, targeting processor **4906** forwards the shoulder signal to tactical processor **4904**.

**[0713]** At step **4918**, tactical processor **4904** registers the cartesian origin in GPS coordinates and determines cardinal axes. At step **4919**, the tactical processor publishes the origin position, in GPS coordinates, to all the nodes of the network. In a preferred embodiment, the nodes of the network include a plurality of remote units, a drone and a fixed camera. In other embodiments, a greater or fewer number of remote units, drones and fixed cameras may be employed.

**[0714]** At step **4922**, weapon processor **4910** identifies the weapon position, as previously described.

**[0715]** At step **4921**, display processor **4908** identifies a gaze ray position. In a preferred embodiment, the gaze ray position is determined by a function call to the Microsoft

HoloLens 2 system. The gaze ray function call returns the line of site of the users eyes relative to the display of the remote unit. The gaze ray is assumed to identify a target of preference moving along a target path. At step **4923**, a series of gaze ray positions is tracked by sequential gaze ray function calls to track the target. The gaze ray positions are translated to account for the relative changes in the headset module as recorded by the headset IMU, relative to the origin to derive a set of gaze ray track data. The gaze ray track data is interpreted as the target track. At step **4924**, the gaze ray track data is sent to targeting processor **4906**. At step **4925**, targeting processor **4906** forwards the gaze ray track data to AI processor to **4902**.

**[0716]** At step **4926**, weapon processor **4910** sends the weapon position data to targeting processor **4906**. At step **4927**, targeting processor **4906** forwards weapon position data to AI processor **4902**. At step **4928**, AI processor adds the weapon position data and the gaze ray track data to a training table, as will be further described.

**[0717]** At step **4929**, targeting processor **4906** determines the path of the remote unit. In a preferred embodiment, the path of the remote unit is tracked based on the series of polls of the IMU by the targeting processor. The IMU presents a position of the targeting processor and the remote unit relative to the origin at any point in time. Instantaneous velocity, position, velocity and acceleration are taken from the IMU at this step.

**[0718]** At step **4930**, weapon processor **4910** determines a range to target by polling the laser range finder attached to the weapon. At step **4931**, targeting processor **4906** determines a range to target by the polling laser range finder attached to the tactical helmet. In another preferred embodiment, targeting processor **4906** determines range to target by polling the stereoscopic camera attached to the tactical helmet.

**[0719]** At step **4932**, targeting processor **4906** calculates the virtual laser position from the weapon position based on its position, velocity and acceleration.

**[0720]** At step **4933**, targeting processor **4906** sends the virtual laser position to display processor **4908**. At step **4934**, display processor **4908** displays the virtual laser position.

**[0721]** At step **4935**, targeting processor **4906** and weapon processor **4910** calibrate to determine a “true” range to target. In a preferred embodiment, the true range to target is determined by the targeting processor by averaging the ranges reported by the weapon processor, and by the targeting processor by both the stereo cameras and the laser range finder.

**[0722]** At step **4936**, the targeting processor **4906** calculates a target path. In a preferred embodiment, the target path is calculated using true range from the remote unit to the target, the path of the remote unit (if moving) and the weapon position data, as previously described.

**[0723]** Alternatively, the step of calculating the target path uses the path of the remote unit, the true range, and the gaze ray track data according to the following equation:

$$\frac{d\theta}{dt} = \frac{\theta_{\text{gaze ray 1}} - \theta_{\text{gaze ray 2}}}{t_1 - t_0}$$

Eq. 52

-continued

$$\frac{d\phi}{dt} = \frac{\varphi \text{ gaze ray } 1 - \varphi \text{ gaze ray } 2}{t_1 - t_0} = r_0 + vt - \frac{1}{2}at^2 \quad \text{Eq. 53}$$

[0724] At step 4937, targeting processor 4906 sends the target path data to tactical processor 4904. In a preferred embodiment, the target path data is sent in spherical coordinates. At step 4938, tactical processor 4904 receives external target path data relative to the origin from other nodes of the network.

[0725] At step 4939, tactical processor 4904 derives an equation for the true target path. In a preferred embodiment, the “true” target path is resolved by averaging the path data received from each of the nodes on the network. However, the true target path may be based on just one of the range reported by the weapon processor, the range reported by the laser range finder of the tactical helmet, or the range reported by the stereoscopic camera. Once an equation is derived for the true target path, the target path is decoupled from the current weapon position data and is not updated with new weapon position data until after a hit or miss condition is received.

[0726] At step 4940, the true path data is sent from the tactical processor to targeting processor 4906. At step 4941, targeting processor 4906 calculates a lead position. Time to target is derived from the ballistic table given the true range. The shot drop is then determined by consulting the ballistic table at the true range. The true path is then projected forward in time by the time to target for the ballistic round at the current range. In a preferred embodiment, the path is projected forward by the display processor in a segmented fashion for an appropriate number of frames to match the time of flight of the round to the target at range.

[0727] At step 4942, tactical processor 4904 sends the resolved path data to AI processor 4902. At step 4943, AI processor 4902 determines lead position from the neural network using the true target path data, as will be further described.

[0728] At step 4944, AI processor 4902 sends the lead predicted by the neural network to targeting processor 4906.

[0729] At step 4945, optionally, targeting processor 4906 compares the lead calculated from the ballistic table to the lead predicted by the neural network. In a preferred embodiment, the total number of recorded hits based on shots on the lead calculated from the ballistic table and the total number of recorded hits based on shots taken according to the neural network recommendation are compared. The lead associated with the highest number of recorded hits is taken as true. “Hits” can also be determined by the use of spotter rounds, which mark where they hit. Such markings can be recognized by the fixed camera or the drone. Hits can also be determined by the use of conventional tracer rounds.

[0730] At step 4946, optionally, targeting processor 4906 chooses the lead associated the highest number of recorded hits. At step 4947, targeting processor 4906 sends the chosen lead distance to display processor 4908. At step 4948, the targeting processor translates the lead coordinates to match the position, velocity and acceleration of the remote unit. At step 4949, the translated lead coordinates are transmitted from the targeting processor to the display processor.

[0731] At step 4950, display processor 4908 displays a phantom target at the lead coordinates.

[0732] At step 4951, the display processor compares the virtual laser position to the phantom position to determine coincidence.

[0733] Upon occurrence of coincidence, at step 4952, display processor 4908 displays a fire alert on the display and sends a “fire alert” signal to targeting processor 4906. At step 4953, targeting processor 4906 forwards the fire alert signal to tactical processor 4904. At step 4954, tactical processor 4904 publishes the fire alert signal, and the identity of the remote unit that triggered the shot, to all nodes on the network.

[0734] At step 4955, weapon processor 4910 registers a shot signal. In a preferred embodiment, the forward IMU of the weapon registers a “shot signal” upon firing of the weapon. The shot signal is sent from the forward IMU to the targeting processor.

[0735] At step 4956, the display processor calculates and displays a virtual tracer, as previously described. At step 4957, weapon processor 4910 forwards the shot signal to targeting processor 4906. At step 4958, targeting processor 4906 forwards the shot signal to tactical processor 4904. At step 4959, tactical processor 4904 forwards the shot signal to AI processor 4902. At step 4960, AI processor adds the shot signal to a training table, as will be further described.

[0736] At step 4961, display processor 4908 identifies a hit or miss condition. In a preferred embodiment, the display processor monitors the image of the identified target with the stereoscopic camera for disappearance. Upon disappearance, a “hit” signal is generated. If no hit signal is generated within a predetermined period of time, a “miss” signal is generated. Spotter rounds and tracers can also be used to validate a hit, as previously described.

[0737] At step 4962, display processor 4908 sends the hit or miss signal to targeting processor 4906. At step 4963, targeting processor 4906 forwards the hit or miss signal to tactical processor 4904. At step 4964, tactical processor 4904 forwards the hit or miss signal to AI processor to 4902. At step 4965, AI processor 4902 adds the hit or miss signal a training table, as will be further described.

[0738] At step 4971, targeting processor 4906 clears the cartesian origin. At step 4972, targeting processor 4906 sends a clear origin signal to tactical processor 4904. At step 4973, tactical processor 4904 resets the cartesian origin of all nodes in the tactical theatre.

[0739] Referring then to FIG. 50, method 5000 of target path resolution will be described.

[0740] In a preferred embodiment, fixed camera 5002, drone 5004, tactical processor 5006 and remote unit 5008 are all connected through a wireless local area network. In another preferred embodiment, fixed camera, drone, tactical processor unit are connected through a wireless wide area network.

[0741] At step S020, tactical processor 5006 sets the cartesian origin. In a preferred embodiment, the cartesian origin is set at the location of a remote unit upon instance of a shouldering signal. In one embodiment, the “x” direction is north to south, as registered by the onboard compass. The “y” direction is east to west. The “z” direction is straight up.

[0742] At step S022, tactical processor 5006 sends the cartesian origin to drone 5004. Similarly, at step S024 tactical processor 5006 sends the cartesian origin to remote unit 5008. At step S026, tactical processor 5006 sends the cartesian origin to fixed camera 5002.



[0743] At step S028, drone 5004 initiates a preset flight path recorded in memory. In another preferred embodiment, the flight path may be controlled manually by the tactical processor through a separate set of flight controls.

[0744] At step S030, tactical processor 5006 receives a target identification signal, as previously described. At step S032, tactical processor 5006 positions the target in the coordinate system relative to the origin. At step S034, tactical processor 5006 sends the position of the target to drone 5004. At step S036, tactical processor 5006 sends the target position to fixed camera 5002. At step S038, tactical processor 5006 sends the target position to remote unit 5008.

[0745] At step S040, remote unit 5008 trains the weapon on the target position. At step S041, remote unit 5008 gets range data from the weapon as previously described. At step S042, remote unit 5008 calculates a path for the target, including position, velocity, acceleration and a target path equation, as previously described. At step S058, remote unit 5008 sends the target path to tactical processor 5006.

[0746] At step S043, drone 5004 trains its camera on the target position. The camera may be trained alternatively by automatically maintaining a constant range to target or manually by instructions from the tactical monitor. At step S044, drone 5004 suspends the flight path and holds position. At step S046, drone 5004 gets range data from the onboard laser range finder. At step S047, drone 5004 tracks the target by monitoring its change in location, velocity and acceleration. At step S048, drone 5004 calculates the target path relative to the origin. In this embodiment, the camera position tracks the target, the range data and the PTZ control instructions are used to derive the target path equation. At step S056, drone 5004 sends the calculated target path to tactical processor 5006.

[0747] At step S050, fixed camera 5002 trains its camera on the target position, as previously described. At step S052, fixed camera 5002 obtains range data from its onboard laser range finder. At step S054, fixed camera 5002 tracks the target to obtain relative positions over time. At step S055, fixed camera 5002 calculates the target path from the target track and from the range data and the PTZ movement instructions required to track the target. At step S060, camera 5002 sends the target path to tactical processor 5006.

[0748] At step S062, tactical processor 5006 resolves the target path relative to the origin. In general, the target path is resolved by averaging the perceived target positions, velocities and accelerations reported from each node reporting on the network. In this embodiment, the target path is resolved relying on a data from the drone, the remote unit and the fixed camera are relative to the same set of cartesian coordinates. In a simplified example, assuming linear motion, the following equations are used.

$$r_1 = r_{01} + v_1 t - \frac{1}{2} a_1 t^2 \tag{Eq. 54}$$

$$r_2 = r_{02} + v_2 t - \frac{1}{2} a_2 t^2 \tag{Eq. 55}$$

$$r_3 = r_{03} + v_3 t - \frac{1}{2} a_3 t^2 \tag{Eq. 56}$$

Where:

- [0749]  $r_1$ =position from drone perspective;
- [0750]  $r_{01}$ =drone initial position;
- [0751]  $v_1$ =velocity from drone perspective;
- [0752]  $a_1$ =acceleration from drone perspective;
- [0753]  $r_2$ =position from fixed camera perspective;

- [0754]  $r_{02}$ =fixed camera initial position;
- [0755]  $v_2$ =velocity from fixed camera perspective;
- [0756]  $a_2$ =acceleration from fixed camera perspective;
- [0757]  $r_3$ =position from fixed camera perspective;
- [0758]  $r_{03}$ =fixed camera initial position;
- [0759]  $v_3$ =velocity from fixed camera perspective;
- [0760]  $a_3$ =acceleration from fixed camera perspective;

Where:

[0761]

$$r_{resolved} = \frac{\begin{pmatrix} r_{01} + r_{02} + \\ r_{03} \end{pmatrix}}{3} + \frac{\begin{pmatrix} v_1 + v_2 + \\ v_3 \end{pmatrix} t}{3} - \frac{\begin{pmatrix} a_1 + a_2 + \\ a_3 \end{pmatrix} t^2}{6} \tag{Eq. 57}$$

[0762] The target path appears differently to each of fixed camera 4678, remote unit 4674, remote unit 4676 and drone 4680. The target path is translated for proper display of remote unit 2 whereupon phantom position 4686 is calculated and displayed. At lead distance 4692, virtual tracer 4688 is displayed on remote unit 4676 and a shot signal is generated when virtual tracer 4688 is coincident with phantom position 4686.

[0763] As shown in FIG. 51, in a preferred embodiment, the AI processor maintains a separate running artificial neural network for each cardinal direction x, y and z in the tactical theatre for each participant. In a preferred embodiment, each of these neural networks is the same, as will be further described. The input for each artificial neural network is the position, velocity, acceleration and true range of the target for each of the cardinal x, y and z directions in the tactical theatre for its particular participant. Each artificial neural network then predicts the vector component of the distance of the lead ahead of the target position, provided the input of the position, velocity, acceleration and range in each of the cardinal directions.

[0764] A preferred embodiment of a single, artificial neural network for predicting a vector component, x, y or z, of the lead distance for one participant is shown in FIG. 52. Neural network 5200 includes input layer 5205, weighting layer 5210, and output layer 5215. The inputs are weighted and processed through input function 5220 and activation function 5225 for reaching an output value. Back propagation is provided by the activation function applied to the weighted neurons. In a preferred embodiment, input function 5120 is a weighted sum of the inputs. In a preferred embodiment, activation function 5125 is the Sigmoid function, as will be further described. One of skill in the art will recognize that other arrangements, numbers and layers of neurons are possible that may provide the desired predictive features of the invention.

[0765] The Sigmoid function is preferred for the activation function because its output can be conveniently used to generate its derivative. For example, if the output variable is "x" then its derivative will be  $x(x-1)$ . The Sigmoid function is shown below:

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} \tag{Eq. 58}$$

[0766] In a preferred embodiment, output layer 5215 assumes a value between 0 and 1 and is appropriately scaled to match the coordinates of the tactical theatre. The output value “h” is the predicted vector component of the lead ahead of the target in one cardinal direction “h” at the position, vector, acceleration and range data is input. With each of three neural networks providing a single component of the lead position, processing is extremely fast. The lead ahead of the target at any given position, velocity, acceleration and range can be predicted to assist in targeting the weapon.

[0767] Training for each artificial neural network requires a training input and a training output. The training input for each neural network is provided by a path table for each direction maintained in database 4655 by tactical monitor 4645 from position, velocity and acceleration of the target. Acceleration and range data is received from the remote units, the drone and the fixed camera. The data can be appropriately scaled. In a preferred embodiment, the distance component of the data is scaled by simply dividing each distance by the maximum distance in the tactical theatre. In another embodiment, the data is scaled by dividing each entry by the largest whole number in the data set. Other scaling methods may be used. In a preferred embodiment, the lead distance entered in the table is characterized by a lead “hit” distance and a lead “miss” distance. In this way, the lead is characterized as a “successful” lead or a “failure” lead. An example table is shown below:

TABLE 3

Time	P <sub>x</sub>	P <sub>y</sub>	P <sub>z</sub>	v <sub>x</sub>	v <sub>y</sub>	v <sub>z</sub>	a <sub>x</sub>	a <sub>y</sub>	a <sub>z</sub>	L <sub>h</sub>	L <sub>m</sub>
t <sub>0</sub>	P <sub>x0</sub>	P <sub>y0</sub>	P <sub>z0</sub>	v <sub>x0</sub>	v <sub>y0</sub>	v <sub>z0</sub>	a <sub>x0</sub>	a <sub>y0</sub>	a <sub>z0</sub>	L <sub>h</sub>	
t <sub>1</sub>	P <sub>x1</sub>	P <sub>y1</sub>	P <sub>z1</sub>	v <sub>x1</sub>	v <sub>y1</sub>	v <sub>z1</sub>	a <sub>x1</sub>	a <sub>y1</sub>	a <sub>z1</sub>		L <sub>m</sub>
t <sub>2</sub>	P <sub>x2</sub>	P <sub>y2</sub>	P <sub>z2</sub>	v <sub>x2</sub>	v <sub>y2</sub>	v <sub>z2</sub>	a <sub>x2</sub>	a <sub>y2</sub>	a <sub>z2</sub>	L <sub>h</sub>	
t <sub>3</sub>	P <sub>x3</sub>	P <sub>y3</sub>	P <sub>z3</sub>	v <sub>x3</sub>	v <sub>y3</sub>	v <sub>z3</sub>	a <sub>x3</sub>	a <sub>y3</sub>	a <sub>z3</sub>		L <sub>m</sub>
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
T <sub>n</sub>	P <sub>xn</sub>	P <sub>yn</sub>	P <sub>zn</sub>	v <sub>xn</sub>	v <sub>yn</sub>	v <sub>zn</sub>	a <sub>xn</sub>	a <sub>yn</sub>	a <sub>zn</sub>	L <sub>hn</sub>	L <sub>mn</sub>

[0768] As can be seen from the table, in this example, each line contains only a single entry for either the lead hit designation or the lead miss designation. In a preferred embodiment, the only lines from the table used for training include only those which include “a lead hit” designation and distance. In this way, the neural network is trained to more accurately predict the lead distance based only on successful hit training data.

[0769] Referring to FIG. 53, a flow chart of method 5300 for training and using each of the artificial neural networks for each of the lead vectors will be described.

[0770] At step S305, each neuron of the weighted layer is assigned a random number between -1 and 1, having a mean value of zero, as initial weight (w).

[0771] At step S310, for each ANN, the training input array is multiplied by the weight array and is summed in a matrix operation. The input data must be appropriately scaled. In a preferred embodiment, the inputs are supplied to the algorithm as a “4xn” matrix, where “n” is the number of time periods where path data is available.

[0772] At step S315, for each iteration, the sigmoid function is applied to derive a calculated output. At step S317, for each iteration, the calculated output is subtracted from the training output to determine an error.

[0773] At step S320, for each iteration, the error is multiplied by the derivative of the sigmoid function of the calculated output. At step S325 the result is multiplied by the training inputs in a matrix operation, to derive an adjustment which complies with the error weighted derivative formula. In a preferred embodiment, the error weighted derivative formula is an algorithm based on gradient descent. In this case, the derivative of the sigmoid function guarantees that the adjustment to each weight changes in a way that always decreases the error for the weight of each neuron.

[0774] At step S327, the adjustment for each neuron is added to the current weight for that neuron.

[0775] At step S330, the process is repeated for a preset number of iterations. In a preferred embodiment, the preset number of iterations is anywhere from 20,000 to 100,000. Other iterations counts can be used. A higher iteration count increases the accuracy of the node weights.

[0776] Once step S330 is complete, the neural network is “trained.”

[0777] At step S340, live target path and range data from the tactical theatre from the remote units, the drone and the fixed camera is scaled and input into the trained neural network by the remote AI data acquisition pattern processor 4650. At step S347, the output for the predictive lead values is read. In a preferred embodiment, the predictive lead values are then transmitted from the pattern processor to the tactical monitor for the distribution to nodes active in the tactical theatre.

[0778] An example of computer code written Python to perform one example of the method is shown in FIG. 54. Of course, other code may be used to implement this and other embodiments of the neural network described.

[0779] It will be appreciated by those skilled in the art that the described embodiments disclose significantly more than an abstract idea including technical advancements in the field of data processing and a transformation of data which is directly related to real-world objects and situations in that the disclosed embodiments enable a computer to operate more efficiently. For example, the disclosed embodiments transform positions, orientations, and movements of a user device and a weapon into a graphical representations of the user and the weapon in a simulation environment.

[0780] It will be appreciated by those skilled in the art that modifications can be made to the embodiments disclosed and remain within the inventive concept, such as by omitting various described features, rearranging features, and using features from one embodiment in another embodiment. Therefore, this invention is not limited to the specific embodiments disclosed, but is intended to cover changes within the scope and spirit of the claims.

1. An augmented reality ranging and lead determination system comprising:

- a set of processors, operatively connected to a set of memories;
- an augmented reality display, connected to the set of processors;
- the set of memories further comprising instructions that when executed by the set of processors cause the system to:
  - derive a weapon path, from a movement of a weapon trained on a target;
  - derive a range distance from the weapon to the target;
  - extend a ray object, from the weapon to the range distance, based on the weapon path;

- derive a target trajectory from the ray object;  
 derive a lead position from the range and the target trajectory; and,  
 render a phantom, at lead position, on the augmented reality display.
2. The system of claim 1 further comprising instructions that when executed by the set of processors cause the system to:
- calculate a virtual laser position from the weapon path;  
 and,
  - render a virtual laser, at the virtual laser position, on the augmented reality display.
3. The system of claim 2 wherein the virtual laser position is generally coaxial with a barrel of the weapon.
4. The system of claim 2 further comprising instructions that when executed by the set of processors cause the system to:
- compare the phantom and the virtual laser for a coincident condition; and,
  - send an alert signal upon the coincident condition.
5. The system of claim 1 further comprising instructions that when executed by the set of processors cause the system to:
- calculate a virtual tracer path from the weapon position;  
 and,
  - render a virtual tracer, at the virtual tracer path, on the augmented reality display.
6. The system of claim 1 further comprising instructions that when executed by the set of processors cause the system to:
- monitor the target for a hit event; and,
  - record the hit event in the memory.
7. The system of claim 1 wherein the weapon further comprises:
- a forward motion sensor, positioned adjacent a barrel of the weapon, operatively connected to the set of processors;
  - a rear motion sensor, positioned adjacent a stock of the weapon, operatively connected to the set of processors; and,
- wherein the movement of the weapon is derived from the forward motion sensor and the rear motion sensor.
8. The system of claim 1 further comprising a laser range finder, attached to the weapon, operatively connected to the set of processors; and,
- wherein the range distance is derived from the laser range finder.
9. The system of claim 1 further comprising a stereoscopic camera, operatively connected to the set of processors; and,
- wherein the range distance is derived from the stereoscopic camera.
10. An augmented reality ranging and lead determination system for a target comprising:
- a first remote unit, having a first processor, and a first memory and a first augmented reality display, operatively connected to the first processor;
  - a second remote unit, having a second processor, and a second memory, and a second augmented reality display, operatively connected to the second processor;
- the first memory and the second memory including a set of instructions that, when executed, cause the system to perform the steps of:
- initiating, by the first remote unit, a track of the target;
  - determining, by the first remote unit, a first range to the target;
  - determining, by the first remote unit, a first weapon position;
  - determining, by the first remote unit, a first remote unit position;
  - calculating, by the first remote unit, a target path based on the first weapon position, the first remote unit position and the first range to target;
  - sending, from the first remote unit to the second remote unit, the target path;
  - determining, by the second remote unit, a second range to target;
  - calculating, by the second remote unit, a first time to target;
  - determining, by the second remote unit, a second weapon position;
  - displaying, on the second augmented reality display, a first virtual laser based on the second weapon position;
  - calculating, by the second remote unit, a first lead distance based on the first time to target;
  - displaying, on the second augmented reality display, a first phantom target at the first lead distance;
  - comparing, by the second remote unit, the first phantom target to the first virtual laser to determine a first coincidence condition; and,
  - generating a fire alert message, by the second remote unit, upon receipt of the first coincidence condition.
11. The system of claim 10 further comprising instructions that, when executed, cause the system to perform the steps of:
- displaying, at the second remote unit, the fire alert message, on the second augmented reality display.
12. The system of claim 10 further comprising instructions that, when executed, cause the system to perform the steps of:
- sensing, at the second remote unit, a shot signal;
  - generating, by the second remote unit, a virtual tracer path upon receipt of the shot signal; and,
  - displaying, on the second augmented reality display, a virtual tracer on the virtual tracer path.
13. The system of claim 12 further comprising instructions that, when executed, cause the system to perform the steps of:
- sending, from the second remote unit to the first remote unit, the shot signal; and,
  - displaying, on the first augmented reality display, the shot signal.
14. The system of claim 10 further comprising instructions that, when executed, cause the system to perform the steps of:
- recording, at the second remote unit, one of a target hit condition and a target miss condition.
15. The system of claim 10 further comprising instructions that, when executed, cause the system to perform the steps of:
- determining, by the second remote unit, a shot drop based on the first time to target; and,
  - translating the target path based on the shot drop.
16. The system of claim 10 further comprising instructions that, when executed, cause the system to perform the steps of:

determining, by the second remote unit, a change in position of the second remote unit; and, translating and rotating the target path based on the change in position.

**17.** The system of claim **10** further comprising instructions that, when executed, cause the system to perform the steps of:

setting, by the first remote unit, an origin position; and, sending, by the first remote unit to the second remote unit, the origin position.

**18.** The system of claim **10** further comprising instructions that, when executed, cause the system to perform the steps of:

receiving, by the first remote unit, a set of cardinal directions; and,

setting, by the first remote unit, a set of cartesian coordinates based on the origin and the set of cardinal directions.

**19.** An augmented reality ranging and lead determination system for a target comprising:

a tactical computer, having a first processor and a first memory;

a spotter unit, having a second processor, a second memory, and a camera, operatively connected to the tactical computer;

a remote unit, having a third processor, a third memory, and an augmented reality display, operatively connected to the tactical computer;

the first memory, the second memory and the third memory including a set of instructions, that when executed, cause the system to perform the steps of:

deriving a weapon path from a movement of a weapon trained on the target;

deriving a first range to target from the remote unit;

deriving a first target trajectory from the weapon path and the first range to target;

deriving a camera path from a movement of the camera trained on the target;

deriving a second range to target from the spotter unit;

deriving a second target trajectory from the camera path and the second range to target;

deriving a third target trajectory from the first target trajectory and the second target trajectory;

calculating a lead from the third target trajectory; and, displaying a phantom target, on the third target trajectory, at the first lead, on the augmented reality display.

**20.** The system of claim **19** wherein the spotter unit further comprises an airborne platform and the set of instructions include further instructions that, when executed, cause the system to perform the steps of:

moving the airborne platform on a flight path; and,

the step of deriving the second trajectory further comprises the step of correcting the second trajectory for the flight path.

**21.** The system of claim **20** wherein and the set of instructions include further instructions that, when executed, cause the system to perform the step of:

controlling the flight path by the tactical computer.

**22.** The system of claim **19** wherein the spotter unit further comprises a fixed platform, supporting the camera.

**23.** The system of claim **22** wherein and the set of instructions include further instructions that, when executed, cause the system to perform the step of:

controlling the camera path by the tactical computer.

**24.** An augmented reality ranging and lead determination system for a target comprising:

a tactical computer, having a first processor and a first memory;

a neural network, having an input layer and an output layer, operatively connected to the tactical computer and the first memory;

a remote unit, having a second processor and a second memory, and an augmented reality display, operatively connected to the tactical computer;

the first memory and the second memory including a set of instructions, that when executed, cause the system to perform the steps of:

deriving a weapon path from a movement of a weapon trained on the target;

deriving a range to target from the remote unit;

deriving a target trajectory from the weapon path and the range to target;

training the neural network with the target trajectory and the range to target;

submitting a current set of target path data to the input layer; and,

reading a predictive set of lead data from the output layer.

**25.** The system of claim **24** wherein the step of training further comprises training the neural network with a set of successful lead data.

\* \* \* \* \*