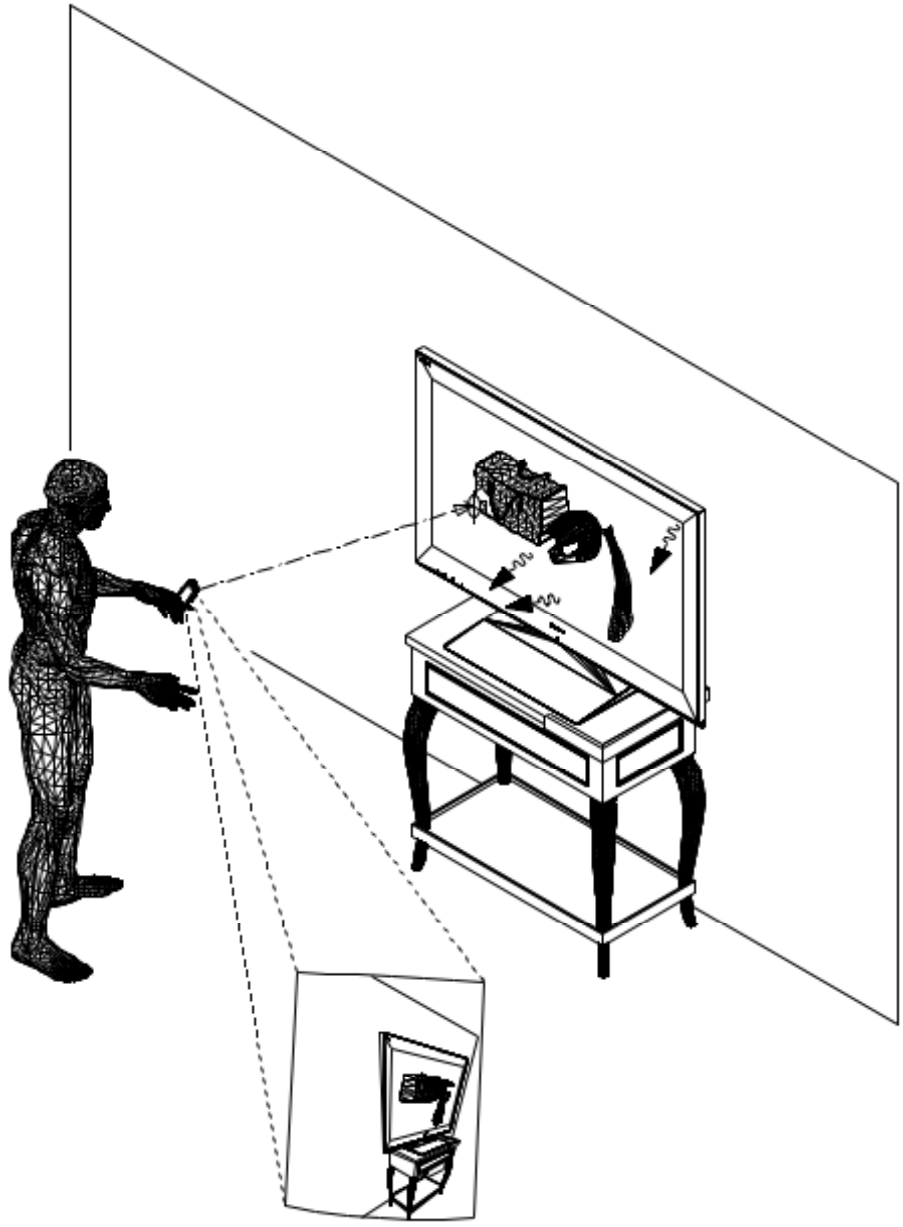
3D ABSOLUTE POSE RECOVERY WITH NAVISCRIBE: AN OVERVIEW OF GENERAL CONSIDERATIONS FOR SMARTPHONE APPLICATIONS

Six Degrees of Freedom Interface for Absolute Pointing and Mouse Modes



Supplied by:
Electronic Scripting Products, Inc.
555 Bryant Street, #142
Palo Alto, CA 94301
info@4espi.com

Note: The material and illustrations presented herein support the development of applications that employ true 3D pose (absolute position and orientation, also referred to as the six degrees of freedom or 6 D.O.F.) of a smart phone or analogous device. Certain aspects of this technology may be proprietary to Electronic Scripting Products, Inc (ESPi). Please contact us at the e-mail listed above for details if you wish to build and implement such application(s).

Revision 0.3 **17th December, 2011**

ABSTRACT

This document describes basic methods for determining the absolute pose (six degrees of freedom including position and orientation) of a smart phone in a stable reference frame. The frame is ascertained from on-board the phone by optical means such as its on-board camera. The camera is programmed to locate a sufficient number of non-collinear optical inputs derived from stationary objects in the environment to parameterize the stable frame. The stationary object can be the display screen of the host device. The 3D interface employs at least a portion of the phone's six absolute pose parameters recovered in the stable frame to generate user input.

TABLE OF CONTENTS

GENERAL BACKGROUND ON RIGID BODY MOTION IN 3D

When an item moves without constraints in a three-dimensional environment with respect to stationary objects, knowledge of the item's distance from and inclination to these objects can be used to derive a variety of the item's parameters of motion as well as its pose. Particularly useful stationary objects for pose recovery purposes include a ground plane, fixed points, lines, reference surfaces and other known features.

Over time, many useful coordinate systems and methods have been developed to parameterize stable reference frames defined by stationary objects. The pose of the item, as recovered and expressed in such stable frames with parameters obtained from the corresponding coordinate description of the frame, is frequently referred to as the item's absolute pose. Based on the most up-to-date science, we know that no absolute or stationary frame is available for defining truly absolute parameters. Stable frame is thus not to be construed to imply a stationary frame. More precisely stated, the stable frame in which the absolute pose is parameterized is typically not a stationary or even an inertial frame (for example, a reference frame defined on the Earth's surface is certainly stable, but not stationary and non-inertial due to gravity and Earth's rotation). Nevertheless, we shall refer to poses defined in stable frames as "absolute" in adherence to convention.

Many conventions have also been devised to track temporal changes in absolute pose of the item as it undergoes motion in the three-dimensional environment. Certain types of motion in three dimensions can be fully described by corresponding equations of motion (e.g., orbital motion, simple harmonic motion, parabolic motion, curvilinear motion, etc.). These equations of motion are typically expressed in the stable frame defined by the stationary objects.

The parameterization of stable frames is usually dictated by the symmetry of the situation and overall type of motion. For example,

motion exhibiting spherical symmetry is usually described in spherical coordinates, motion exhibiting cylindrical symmetry in cylindrical coordinates and generally linear motion in Cartesian coordinates. More advanced situations may even be expressed in coordinates using other types of parameterizations, e.g., sets of linearly independent axes.

Unconstrained motion of items in many three-dimensional environments, however, may not lend itself to a simple description in terms of equations of motion. Instead, the best approach is to recover a time sequence of the item's absolute poses and reconstruct the motion from them. For a theoretical background, the reader is referred to textbooks on classical mechanics and, more specifically, to chapters addressing various types of rigid body motion. An excellent overall review is found in H. Goldstein et al., *Classical Mechanics*, 3rd Edition, Addison Wesley Publishing, 2002.

Items associated with human users, e.g., items that are manipulated or worn by such users, generally do not move in ways that can be described by simple equations of motion. That is because human users exercise their own will in moving such items in whatever real three-dimensional environment they find themselves. It is, however, precisely the three-dimensional motion of such items that is very useful to capture and describe. That is because such motion may communicate the desires and intentions of the human user. These desires and intentions, as expressed by corresponding movements of the item (e.g., gestures performed with the item), can form the basis for user input and interactions with the digital domain (e.g., data input or control input).

Many common methods for pose recovery of items moving without constraints in 3D space employ optics. Such optical approaches to pose recovery are intuitive, since our own human vision system computes locations and motion trajectories of items in real three-dimensional environments in that manner. This includes recovery of

our own pose and movement in a three-dimensional environment based on images provided by our eyes. The pose recovery algorithms from those images are implemented by our senses, which develop as part of our natural proprioception in early childhood.

The high accuracy and precision of optical pose recovery and navigation is due in large part to the very short wavelength of electromagnetic radiation in comparison with typical dimensions of objects and items of interest. Furthermore, radiation incurs negligible latency in short distance measurements due to the extremely large speed of light. EM radiation in the optical range (e.g., visible and IR) is also relatively immune to interference. Thus, it is well known that the problem of determining an absolute pose or a sequence of such absolute poses that represent a recovered motion trajectory of an item in almost any real three-dimensional environment may be effectively addressed by the application of optical apparatus and methods.

A particularly acute need for efficient, accurate and low-cost determination of the absolute pose of an item in a real three-dimensional environment is found in the field of items associated with a human user. Such items may be held and manipulated by the user. Alternatively, such items may be worn by the user. In either case, the items are intended to help the user interact with the digital world. Besides smart phones that will be addressed in detail herein, such items encompass myriads of manipulated objects such as pointers, wands, remote controls, gaming objects, jotting implements, surgical implements, three-dimensional digitizers and various types of human utensils whose motion in real space is to be processed to derive a digital input for an application.

Unfortunately, motion mapping between space and cyberspace is not possible without the ability to digitize the absolute pose of the item in a well-defined and stable reference frame. All devices that do not solve the full motion problem, i.e., do not capture successive

absolute poses of the item with a method that accounts for all six degrees of freedom (namely, three translational and the three rotational degrees of freedom inherently available to rigid bodies in three-dimensional space) encounter limitations. Among many others, these limitations include information loss, appearance of an offset, position aliasing, gradual drift and accumulating position and orientation errors. The users of a Nintendo Wii will be familiar with many of these issues.

The challenges for 3D user interfaces with the digital world do not end with their ability to recover absolute pose in an efficient and accurate manner. Many additional issues need to be addressed and resolved, over and above those that we have discussed above. In fact, it may be in a large part due to the fact that some of the more basic challenges are still being investigated, that the questions about how to use the recovered poses are still unanswered.

OVERVIEW OF THE SMART PHONE APPLICATION

The present application targets an interface that derives or produces input to a software application based on an absolute pose of a smart phone operated in a three-dimensional environment. Absolute pose means both the position and the orientation of the smart phone as described in a stable reference frame defined in that three-dimensional environment.

The smart phone and its user are found in the three-dimensional environment. Such environment has a spatial extent that can be described with three dimensions or directions such as length, width and height, or the X, Y and Z axes. The absolute pose of the smart phone in the three-dimensional environment includes its position and its orientation. The position can change along any of the three directions. In other words, position has at least three translational degrees of freedom (i.e., translation along X, Y and Z, or, in general, any three linearly independent axes). In addition, the absolute pose of the smart phone also includes its orientation. The orientation exhibits at least three rotational degrees of freedom (i.e., rotation around axes defined by X, Y or Z or, in general, rotation around any three linearly independent axes). Therefore, the smart phone has available to it at least six (6) degrees of freedom in the three-dimensional environment.

The interface to be deployed with the aid of the smart phone requires knowledge about at least one stationary object that has at least one feature that is detectable via an electromagnetic radiation in the optical range (e.g., from ultra-violet to infra-red). This feature has to present a sufficient number of non-collinear optical inputs to enable one to establish a stable reference frame in the three-dimensional environment. In other words, the number and type of non-collinear optical inputs are sufficient to allow one to establish stable world coordinates. Such world coordinates (X_w,Y_w,Z_w) are used to describe or parameterize the stable frame in the three-dimensional environment.

In rare cases, world coordinates (X_w,Y_w,Z_w) may describe an actual inertial frame of reference in which the user resides (e.g., on a spacecraft in outer space). Normally, however, world coordinates (X_w,Y_w,Z_w) describe a non-inertial frame in which the user, the smart phone and the stationary object all reside. The simplest non-inertial frame is found on the surface of the Earth (due to gravity and our planet's motion, such as rotation around its axis). More complex non-inertial frames are encountered aboard planes, trains, cars or other aircraft or terrestrial vehicles that undergoes linear acceleration or some curvilinear motion.

The interface takes advantage of a camera on-board the smart phone for receiving the electromagnetic radiation. Based on the electromagnetic radiation received, the camera generates a signal that is related to at least one absolute pose parameter of the smart phone as defined in the stable frame. For example, in the simplest case, the signal amplitude, frequency or phase may be directly proportional to the at least one absolute pose parameter.

The interface communicates with an application (e.g., a software program) via any suitable communication link. The application employs the signal related to the one or more absolute pose parameters of the smart phone in the input. For example, the signal may constitute the complete input to the application. Alternatively, the signal may be used intermittently or present merely a subset of a continuous input stream.

As remarked above, in the three-dimensional environment, as dictated by the fundamental geometrical rules of 3D space, the absolute pose of a rigid body and hence of the smart phone exhibits at least three translational and at least three rotational degrees of freedom. Thus, in the most basic embodiment of the invention, the signal is related to at least one absolute pose parameter which, in turn, is

related to one or more from among the at least three translational and at least three rotational degrees of freedom by a mapping.

A mapping, as understood in general and herein, is any rule or set of rules for establishing correspondence between the one absolute pose parameter and the at least three translational and the at least three rotational degrees of freedom. For example, the mapping may associate the at least one absolute pose parameter to any portion of each of the at least six degrees of freedom (the at least 3 translational and the at least 3 rotational degrees of freedom). Alternatively, the mapping may associate the at least one absolute pose parameter to only some predetermined portions of the at least six degrees of freedom. In fact, any mapping supported by the geometrical limits imposed on translations and rotations of rigid bodies in three-dimensional environments is a permitted mapping. (Although it should be noted that in a cyberspace, a virtual space, an augmented reality space and a mixed space in which new effect are desired, the rules of real space geometry may sometimes be disregarded.)

In a one-to-one interface, the mapping is a one-to-one mapping. In other words, there is a one-to-one mapping between the six degrees of freedom and the at least one absolute pose parameter. Thus, when the smart phone executes two translations (e.g., along X and Y axes) and a rotation (e.g., around the Z axis), then these translations and the rotation are mapped one-to-one to the at least one absolute pose parameter. Specifically, in this case the mapping produces three absolute pose parameters corresponding to two translations (along X and along Y) and one rotation (around Z).

In other applications, the mapping includes a scaling in at least one of the six or more degrees of freedom. In particular, when the item executes two translations (e.g., in X and Y) and a rotation (e.g., around Z), the translations may be scaled 1:2 in the mapping. Such scaling will produce three absolute pose parameters corresponding to

½ the translation along X axis, ½ the translation along Y axis, and the full (unscaled) rotation around Z axis. Of course, one can instead scale the rotation and not the translations.

It is important for computational reasons to make a wise choice when defining the degrees of freedom given the application(s). For example, in many cases it is convenient to choose two translational degrees of freedom that define a plane in the three-dimensional environment; e.g., degrees of freedom in X and in Y can be used to define an X-Y plane. When the application involves the use of a display, it is convenient to set up the three-dimensional environment in such manner that the X-Y plane is plane-parallel with the display, or, more precisely, the screen of the display. In some such embodiments, the display is integrated into one of the stationary objects, such as the television or gaming system with which the smart phone cooperates.

In most cases, it is convenient to choose the at least three translational and at least three rotational degrees of freedom in such manner that they be not just linearly independent but mutually orthogonal. In other words, they should represent three mutually orthogonal translational degrees of freedom (e.g., X, Y and Z) and three mutually orthogonal rotational degrees of freedom. These can be described by (pitch, yaw and roll) or their mathematical equivalents. Other options include but are not limited to: (heading, elevation and bank) and their mathematical equivalents, Euler angles (φ, θ, ψ) or Tait-Bryan angles and their mathematical equivalents, Cayley-Klein parameters (related to Euler angles) and their mathematical equivalents.

Of course, it is also possible to choose other orthogonal and non-orthogonal descriptions to keep track of the rotational and translational degrees of freedom. Some of these involve convolutions of displacements and/or angles (e.g., the pan angle concept), direction cosines and/or descriptions involving homogeneous

coordinate system and quaternions and all corresponding mathematical equivalents.

It should be noted, that mathematically the many options for keeping track of all three rotations can always be reduced to Euler angles and their equivalents. The choice of the rotation convention should be made based on the nature of the application and the range of absolute poses that the smart phone is expected to assume as well as the method in which the on-board camera receives the electromagnetic radiation (e.g., camera rotation matrices and corresponding machine vision conventions may dictate the most useful choice). Also, even though from the mathematical standpoint, choosing orthogonal coordinate systems guarantees efficiency, the actual application may not require, or may be better served, by adopting a description involving merely linearly independent axes.

In most cases, however, the choice of mutually orthogonal translational degrees of freedom to correspond to the three orthogonal Cartesian axes will be most appropriate and useful. In these cases the orthogonal Cartesian axes are preferably used as world coordinates (X_w, Y_w, Z_w) to describe or parameterize the stable frame. Furthermore, a certain reference location or designated point on the smart phone is expressed in these world coordinates (X_w, Y_w, Z_w) to define a position of the smart phone in world coordinates and thereby in the stable frame. (Note that in some conventions, the point chosen on the smart phone to indicate its position is abstract, e.g., a point such as the center of mass (C.O.M.) or some other point associated with the phone but not physically a part of it — also note that depending on the phone's geometry, the C.O.M. is not always within the physical volume.)

Indeed, in many embodiments, the interface is conveniently parameterized in six degrees of freedom (6 D.O.F. interface). In other words, the at least one absolute pose parameter includes six absolute pose parameters that map to three of the at least three

translational degrees of freedom and to three of the at least three rotational degrees of freedom. This provides for a full parameterization of the absolute pose of the smart phone in the three-dimensional environment. The application may use such full parameterization of the smart phone's absolute pose in the input to the application. Moreover, the reader will realize that choosing orthogonal translational degrees of freedom (e.g., X, Y and Z axes) and orthogonal rotational degrees of freedom (e.g., Euler angles (φ,θ,ψ)) is particularly convenient for such full parameterization.

In some embodiments the application has a feedback unit for providing feedback to the user in response to at least one portion of the full parameterization. For example, the feedback unit uses the phone's display to show visual information, some or all of which may represent the feedback. For example, the visual information may be an image, a portion of an image, an icon, a series of images (e.g., a video) or other visual information rendered from a point of view of the smart phone in the three-dimensional environment.

The point of view of the item is derived from the at least one portion, and preferably from the full parameterization of the item's pose, i.e., from the six absolute pose parameters $(x,y,z,\varphi,\theta,\psi)$ or a subset of these. When employing an on-board camera it may also be convenient to work with alternative but mathematically equivalent parameterizations employing concepts such as surface normals (e.g., normal to the X-Y plane discussed above), pan angles (e.g., convolutions of two rotation angles), horizon lines, vanishing points and other optics and imaging concepts from projective geometry.

In some cases the feedback unit is a tactile feedback unit. It is associated with the application and provides tactile information also sometimes referred to as haptic feedback to the user. In particular, the tactile or haptic information may consist of vibration is derived from at least a portion of the full parameterization of the smart

phone. Audio feedback may also be used to indicate various states of the smart phone as it moves in the three-dimensional environment.

In many smart phone applications, the one or more stationary objects used for establishing the stable frame will include a display that is integrated into the object. In some of these cases it is advantageous to use the full parameterization of the smart phone's pose in the application to compute an intersection of a mechanical axis of the phone with the display, or more precisely, with the area spanned by the display or its screen. The optical axis of the camera's lens can be chosen as the mechanical axis. Thus, a user holding the phone will point or indicate along the direction of the optical axis. This choice is particularly useful when one of the intended uses in the context of the application is to point and click and/or to point and control/move or the like (absolute pointer of absolute 3D mouse).

In embodiments where a display is provided and pointing is available, it is also convenient to introduce a place-holder entity and place it at the intersection of the optical axis with the display or its screen. Thus, the user will get visual feedback via the place-holder entity of where the phone is pointing. The place-holder entity can contain additional information apparent from its character. For example, the place-holder entity may by an insertion cursor, a feedback cursor, a control icon, a display icon or any other visual feedback entity whose appearance communicates information to the user.

Depending on the three-dimensional environment and modes of operation, the interface may also use the smart phone's relative motion sensor. Relative motion sensors are to be understood as sensors that are not capable of recovering absolute pose in the stable frame established in the three-dimensional environment. Suitable relative motion sensors include accelerometers, gyros, magnetometers, optical flow meters, acoustic devices and the like.

Any such sensor (or combination of them) can be placed on-board the smart phone for producing data indicative of a change in at least one among the at least three translational and the at least three rotational degrees of freedom. This relative data can be used to supplement (e.g., interpolate) the signal that is related to the at least one absolute pose parameter. In fact, it is preferable, as will become apparent in the detailed description of the smart phone implementation, to employ the phone's on-board gyros and accelerometers in a sensor fusion that enables very efficient and robust 3D pose recovery.

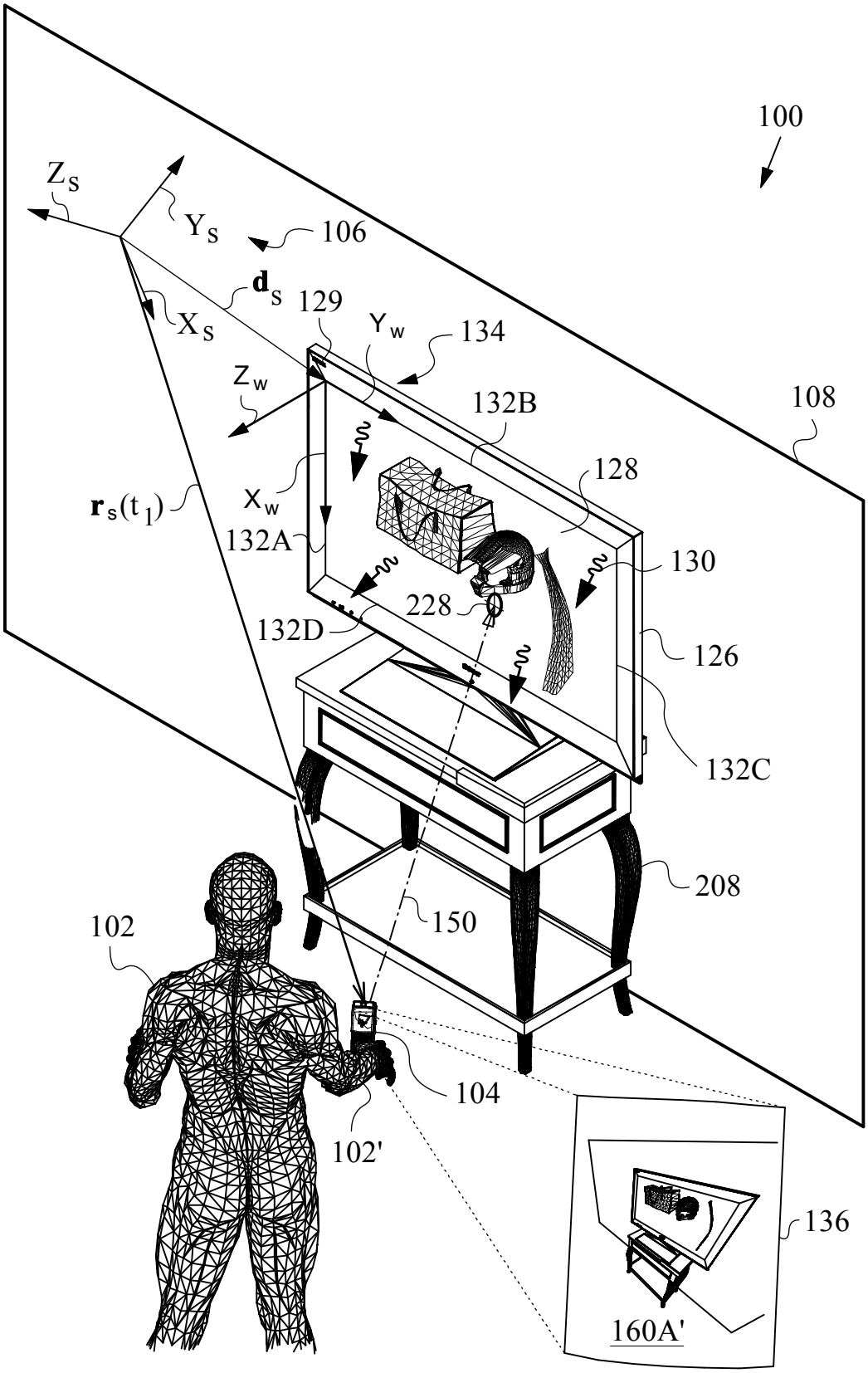

100
Z_S
Y_S
106
$\mathbf{d}_S$
X_S
129
Y_w
134
Z_w
132B
108
128
$\mathbf{r}_s(t_1)$
X_w
132A
130
228
132D
126
132C
208
102
150
104
102'
136
160A'

FIG. 1A

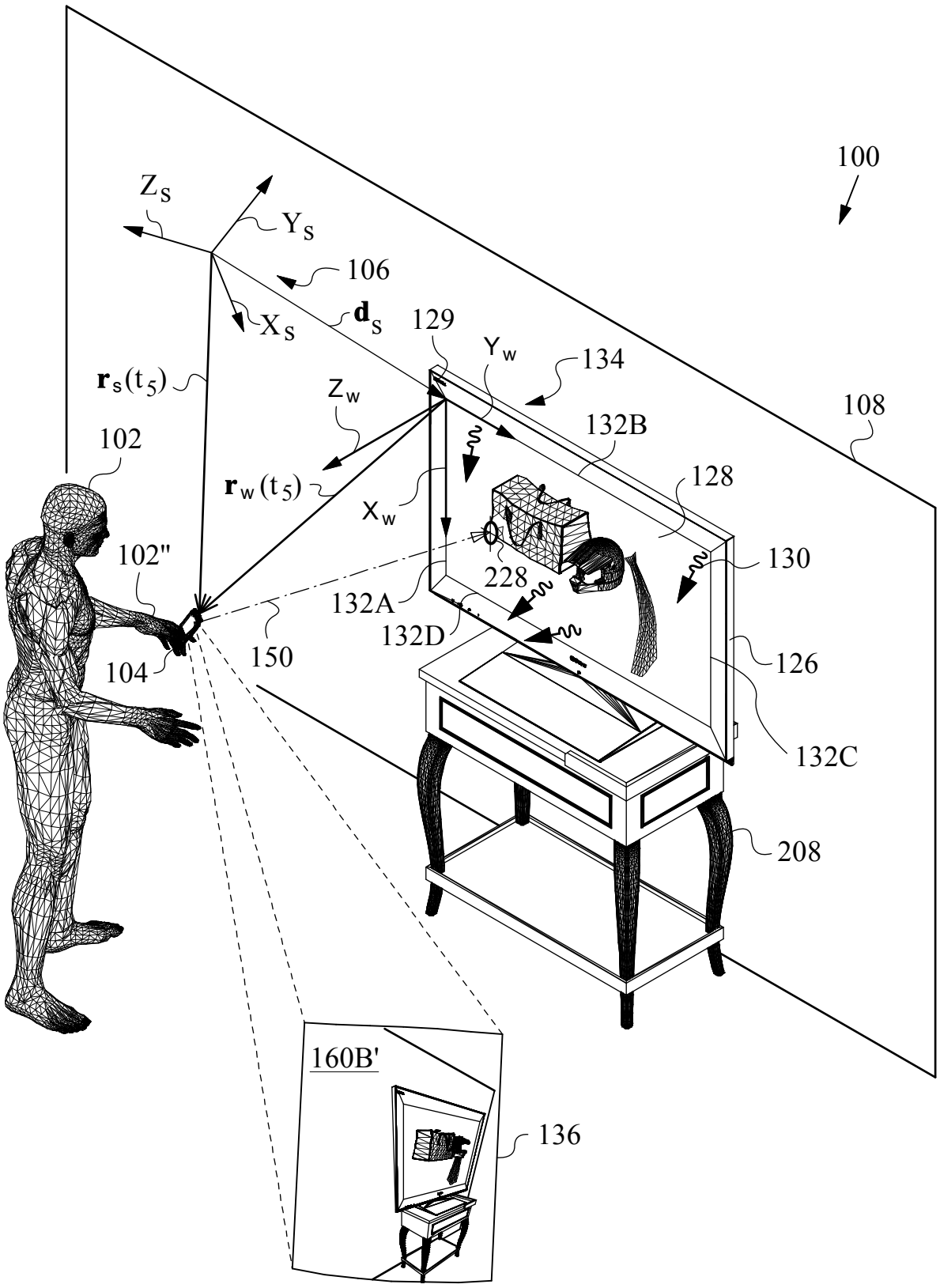

100
Z_S
Y_S
106
X_S
$\mathbf{d}_S$
129
Y_w
134
$\mathbf{r}_s(t_5)$
Z_w
108
102
132B
$\mathbf{r}_w(t_5)$
128
X_w
102"
130
228
132A
132D
150
126
104
132C
208
160B'
136

FIG. 1B

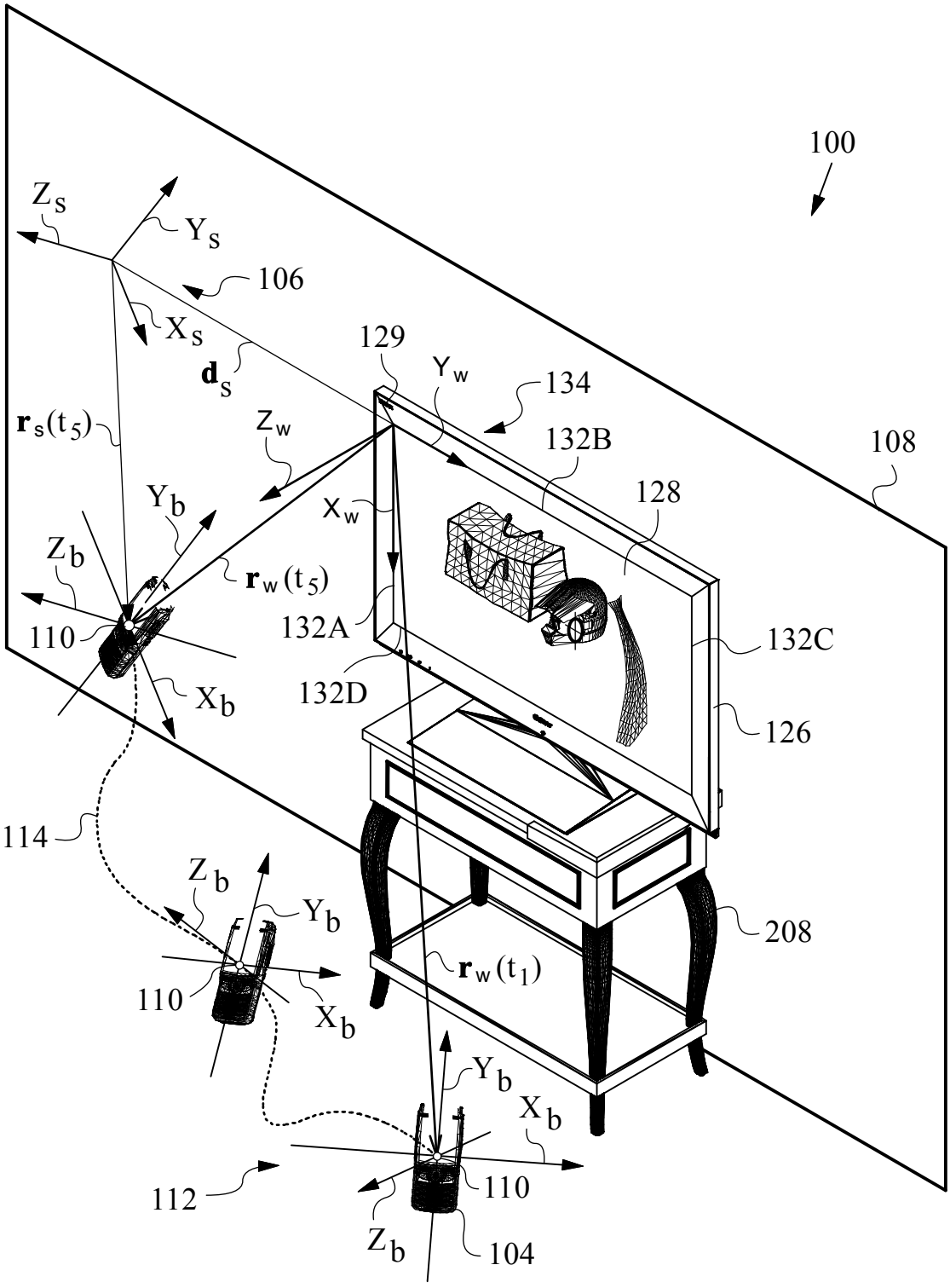

100
Z_s
Y_s
106
X_s
$\mathbf{d}_s$
129
Y_w
134
$\mathbf{r}_s(t_5)$
Z_w
132B
108
Y_b
X_w
128
Z_b
$\mathbf{r}_w(t_5)$
132A
110
132C
X_b
132D
126
114
Z_b
Y_b
208
$\mathbf{r}_w(t_1)$
110
X_b
Y_b
X_b
112
110
Z_b
104

FIG. 2

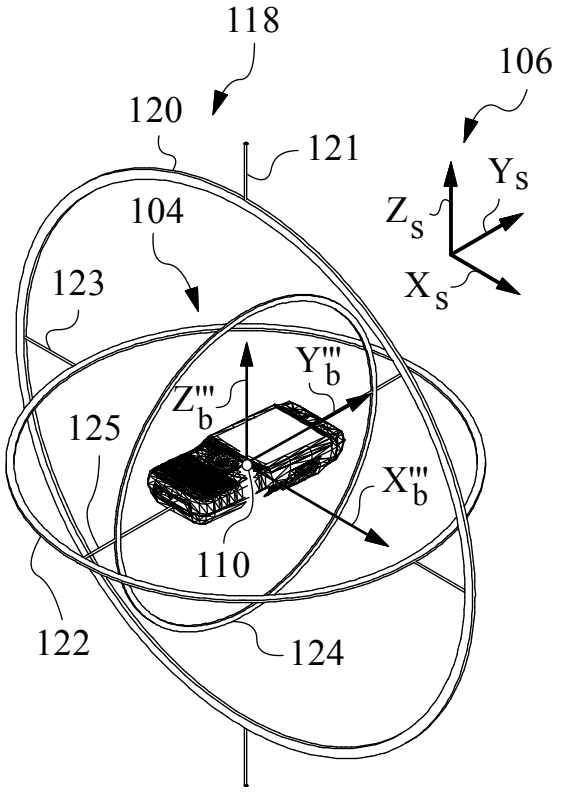


FIG. 3A

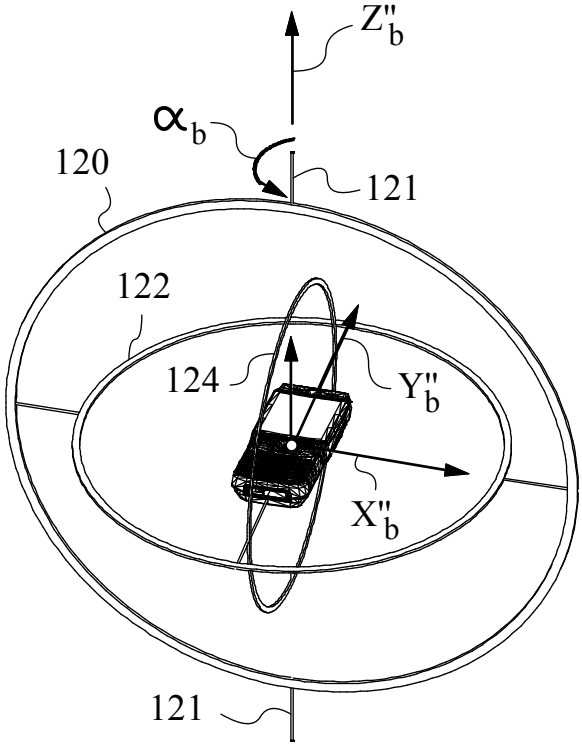


FIG. 3B

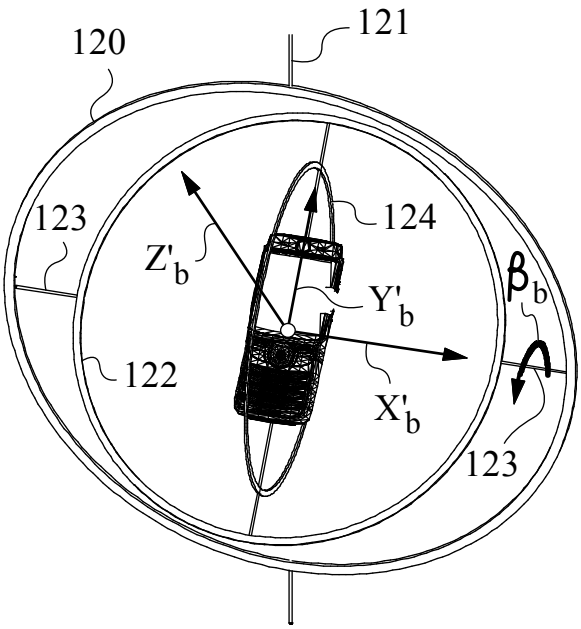


FIG. 3C

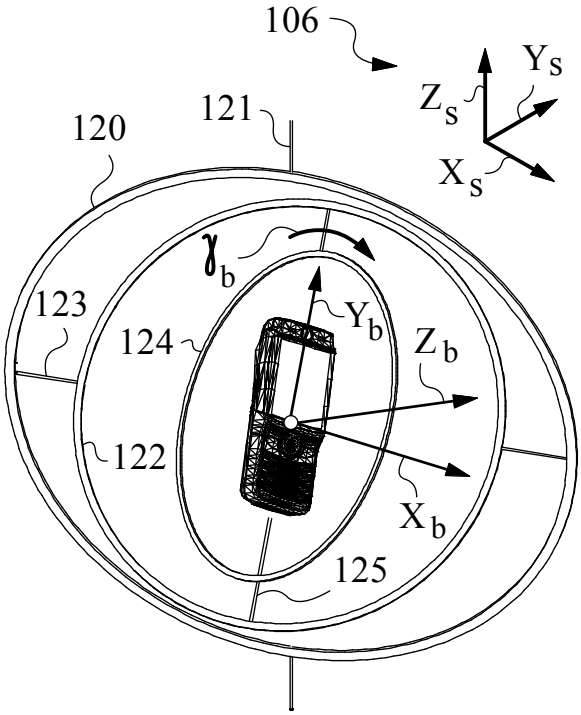


FIG. 3D

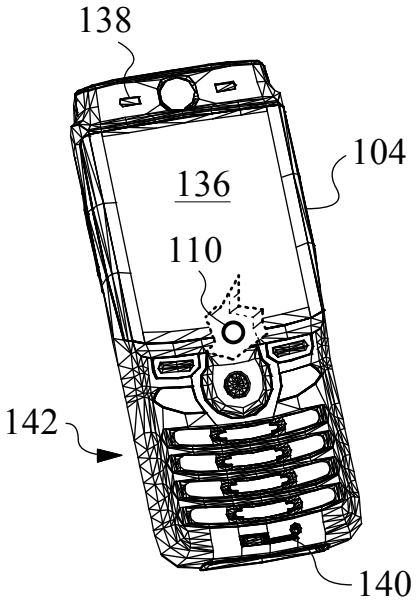


FIG. 4A

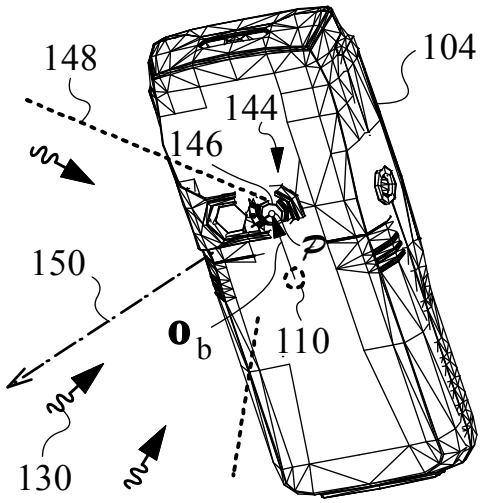


FIG. 4B

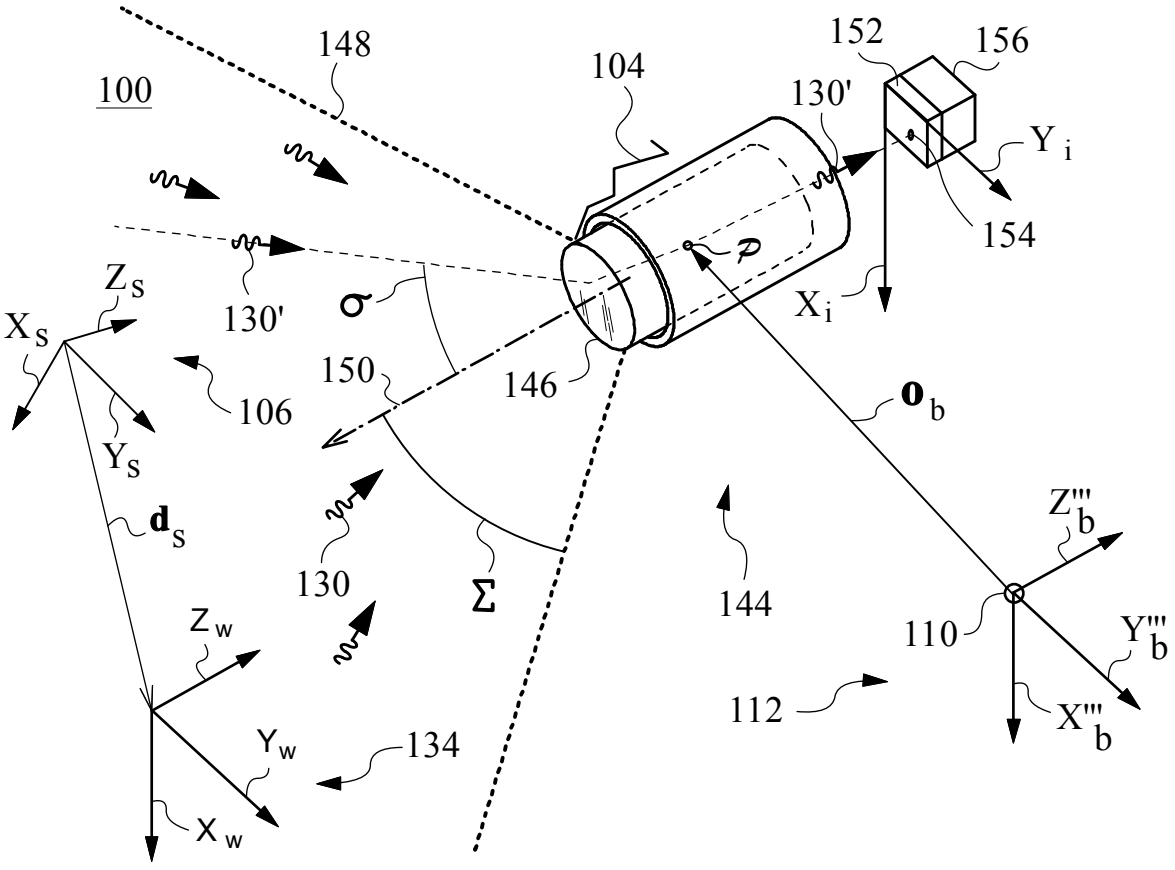


FIG. 5

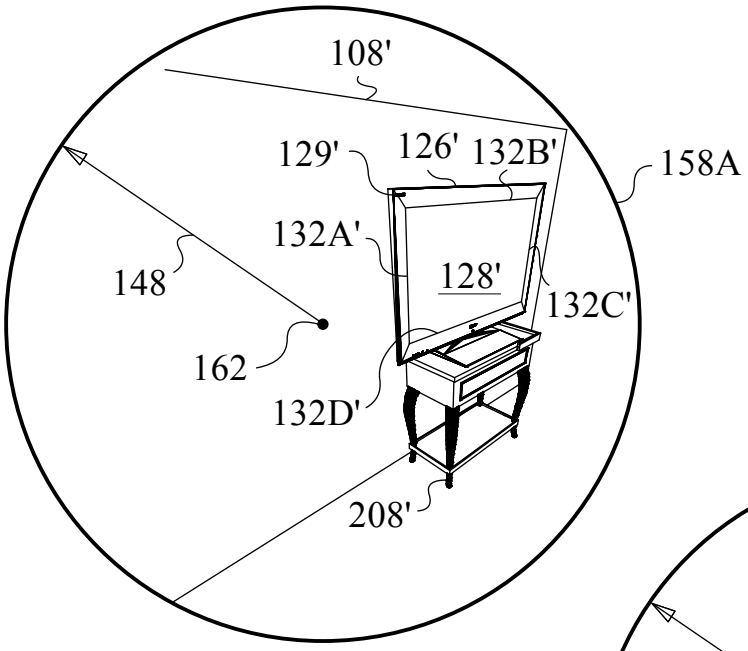


FIG. 6A

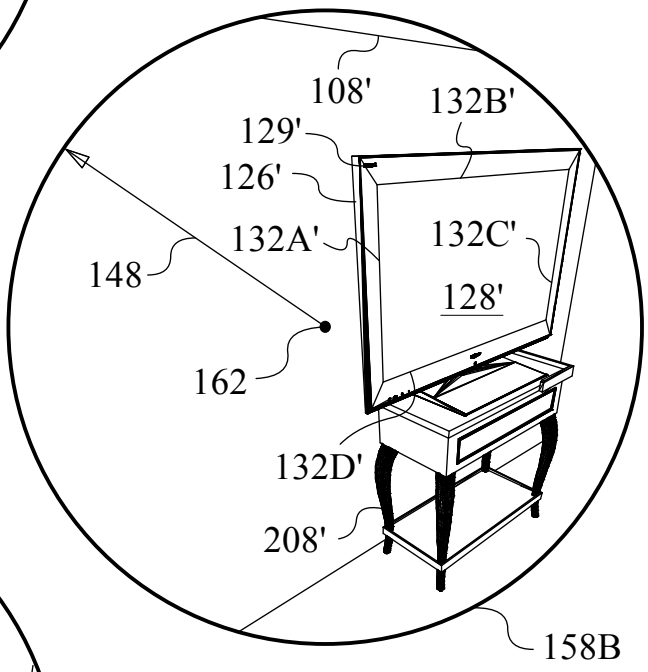


FIG. 6B

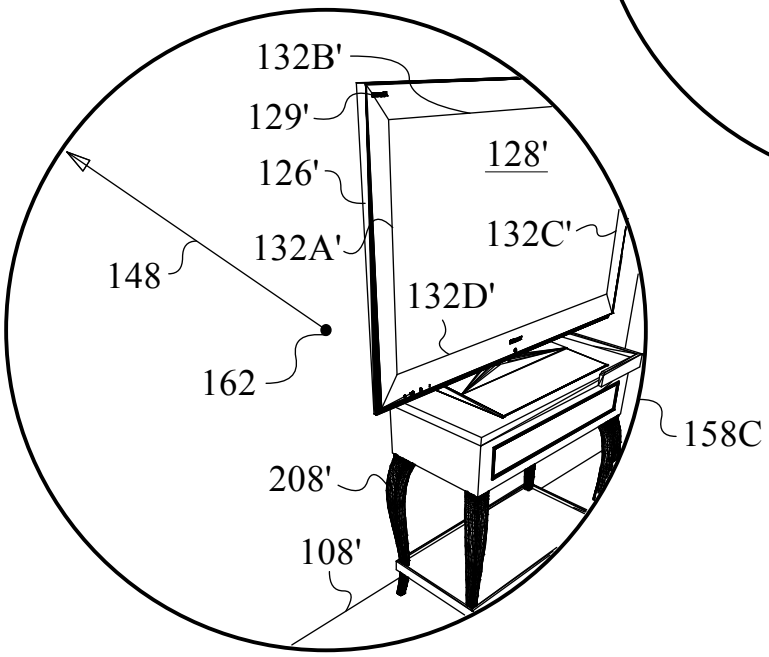


FIG. 6C

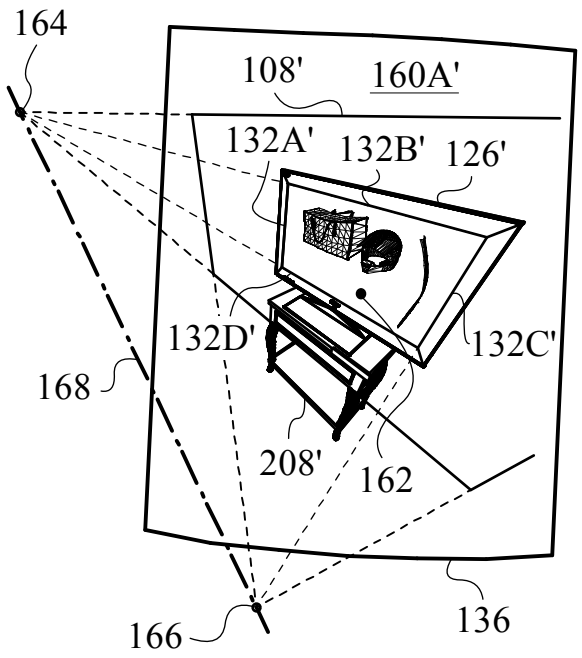


FIG. 7A

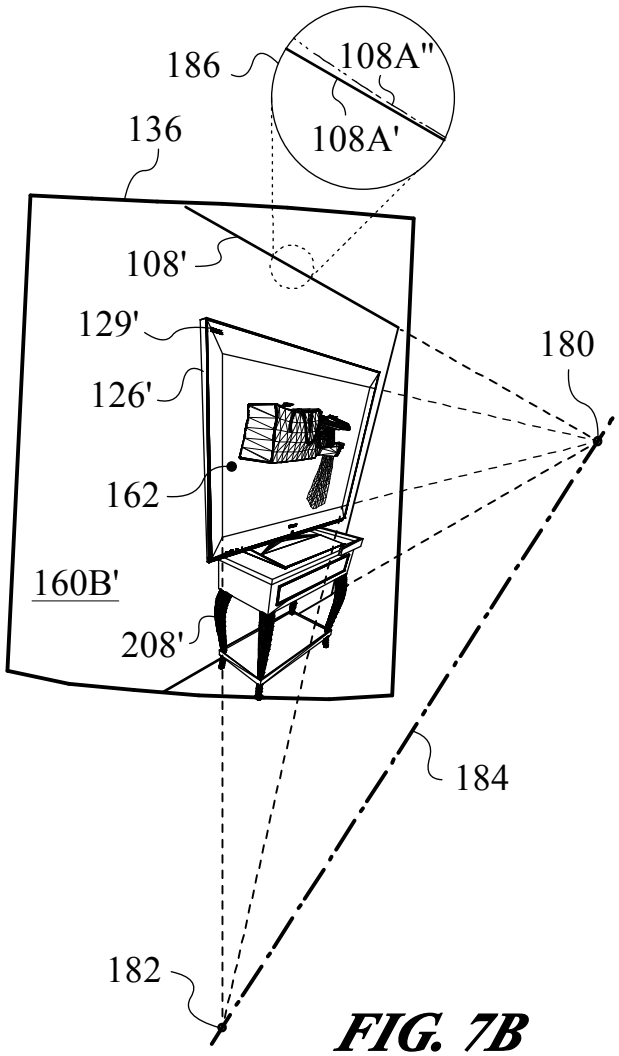


FIG. 7B

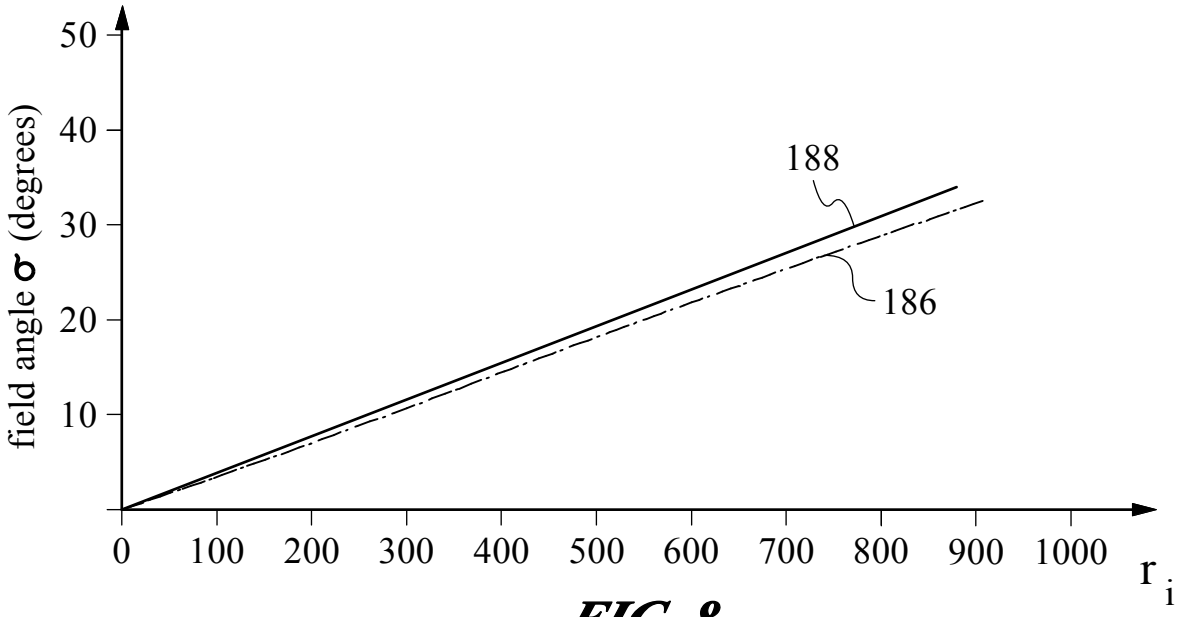


FIG. 8

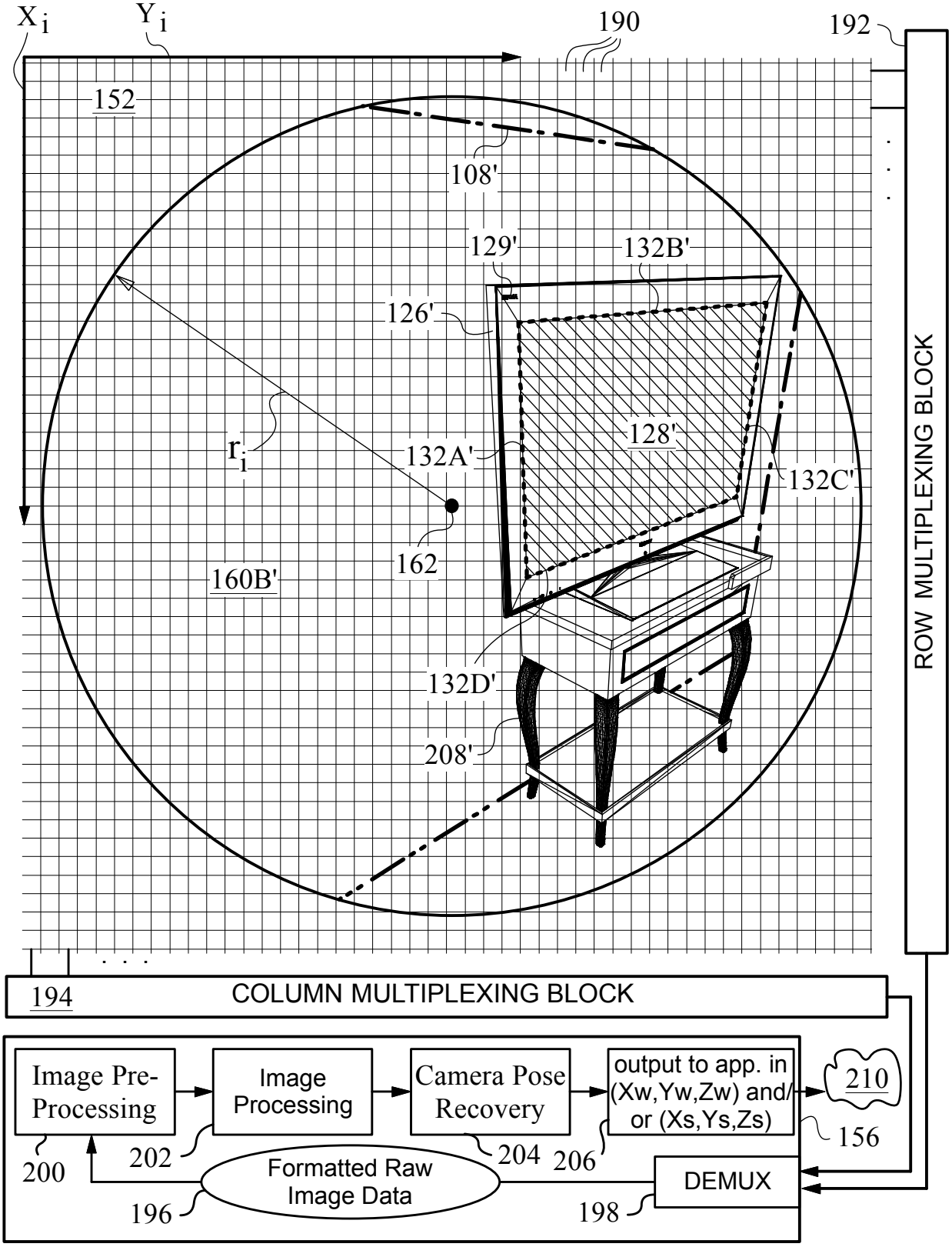

X i
Y i
190
192
152
108'
129'
132B'
126'
128'
132A'
132C'
r i
162
160B'
132D'
208'
ROW MULTIPLEXING BLOCK
194
COLUMN MULTIPLEXING BLOCK
Image Pre-Processing
Image Processing
Camera Pose Recovery
output to app. in (Xw,Yw,Zw) and/or (Xs,Ys,Zs)
210
156
200
202
204
206
Formatted Raw Image Data
DEMUX
196
198

FIG. 9

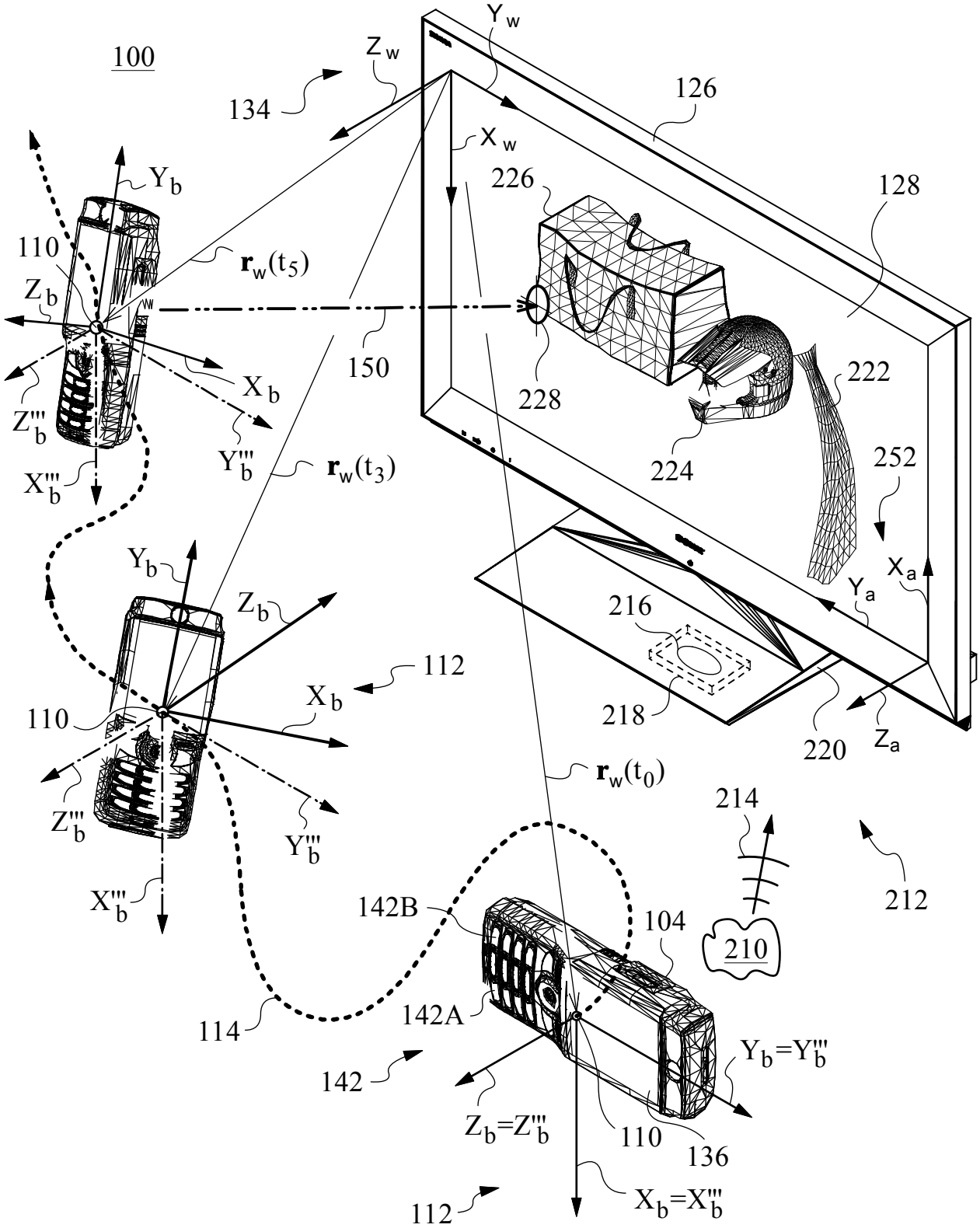

100
Y_w
Z_w
134
126
X_w
Y_b
226
128
110
$\mathbf{r}_w(t_5)$
Z_b
150
X_b
228
222
Z'''_b
Y'''_b
$\mathbf{r}_w(t_3)$
224
252
X'''_b
Y_b
Z_b
X_a
Y_a
216
112
X_b
110
218
220
Z_a
$\mathbf{r}_w(t_0)$
214
Z'''_b
Y'''_b
212
142B
104
X'''_b
210
114
142A
$Y_b=Y'''_b$
142
$Z_b=Z'''_b$
110
136
112
$X_b=X'''_b$

FIG. 10

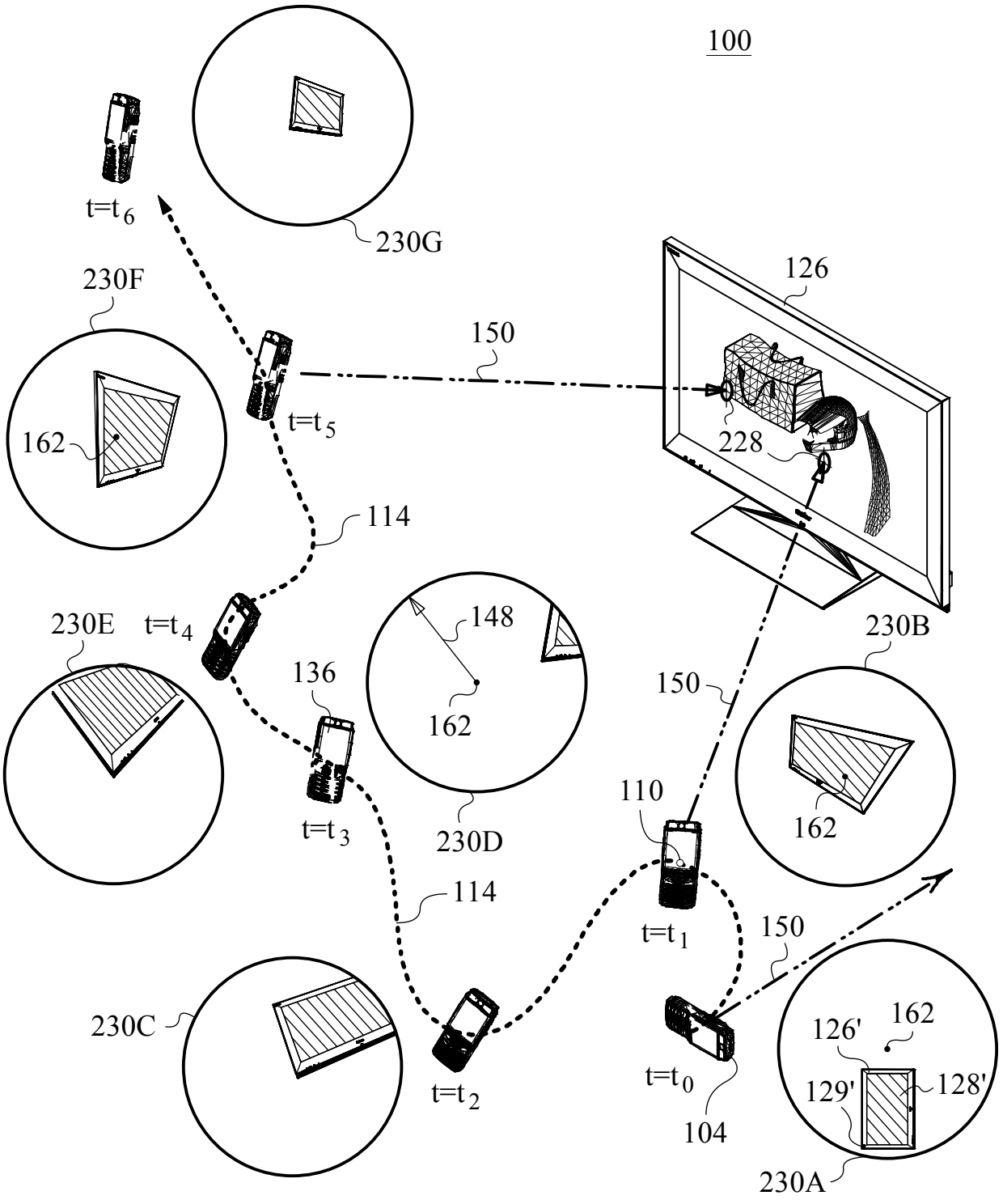

100
t=t6
230G
230F
126
150
162
t=t5
228
114
230E
t=t4
148
230B
136
150
162
162
110
t=t3
230D
114
t=t1
150
230C
126'
162
t=t0
129'
128'
t=t2
104
230A

FIG. 11

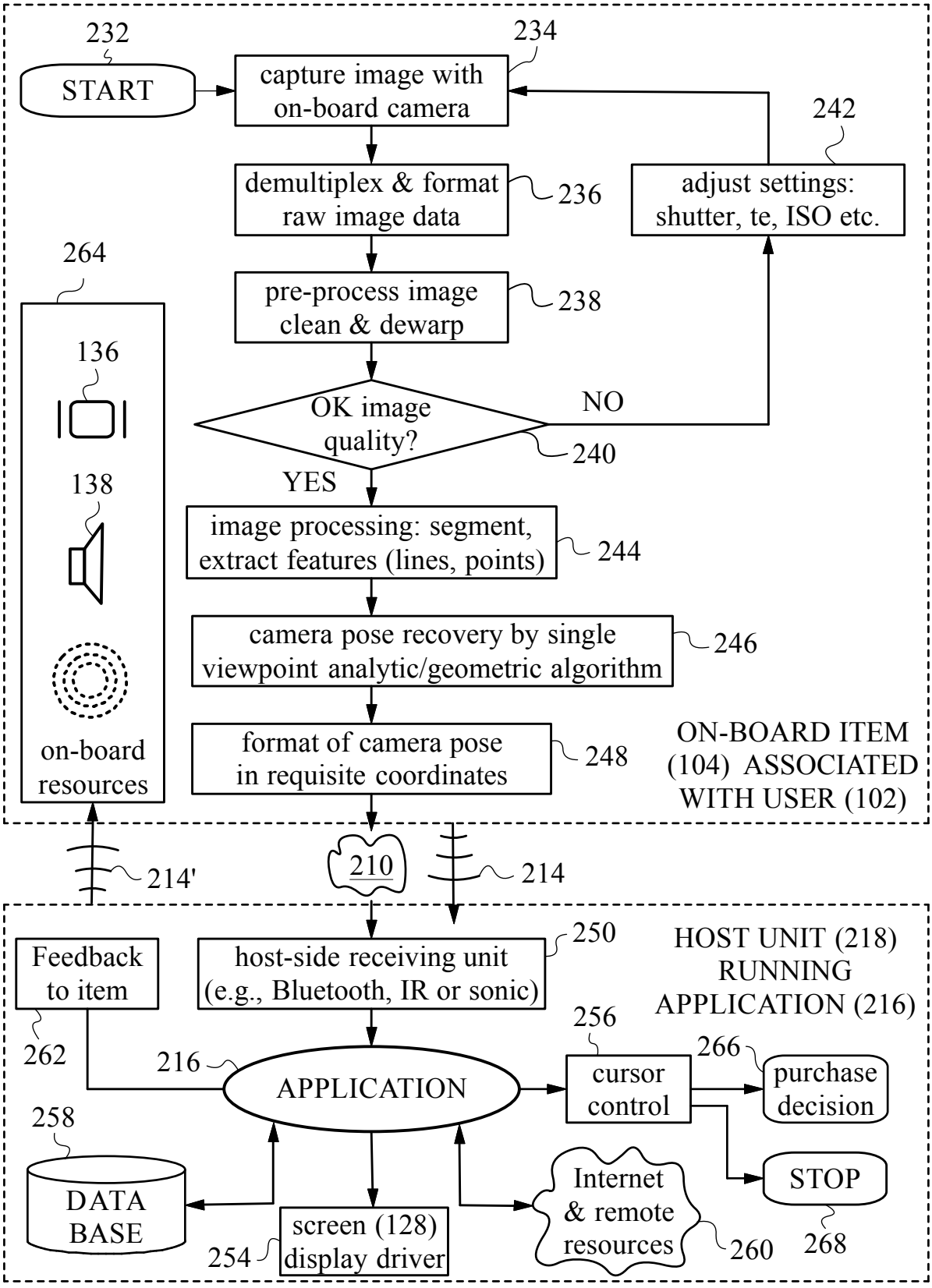

232
START
234
capture image with on-board camera
242
adjust settings: shutter, te, ISO etc.
demultiplex & format raw image data
236
pre-process image clean & dewarp
238
264
136
138
on-board resources
OK image quality?
NO
240
YES
image processing: segment, extract features (lines, points)
244
camera pose recovery by single viewpoint analytic/geometric algorithm
246
format of camera pose in requisite coordinates
248
ON-BOARD ITEM (104) ASSOCIATED WITH USER (102)
214'
210
214
Feedback to item
262
host-side receiving unit (e.g., Bluetooth, IR or sonic)
250
HOST UNIT (218) RUNNING APPLICATION (216)
216
APPLICATION
256
cursor control
266
purchase decision
258
DATA BASE
STOP
268
Internet & remote resources
260
254
screen (128) display driver

FIG. 12

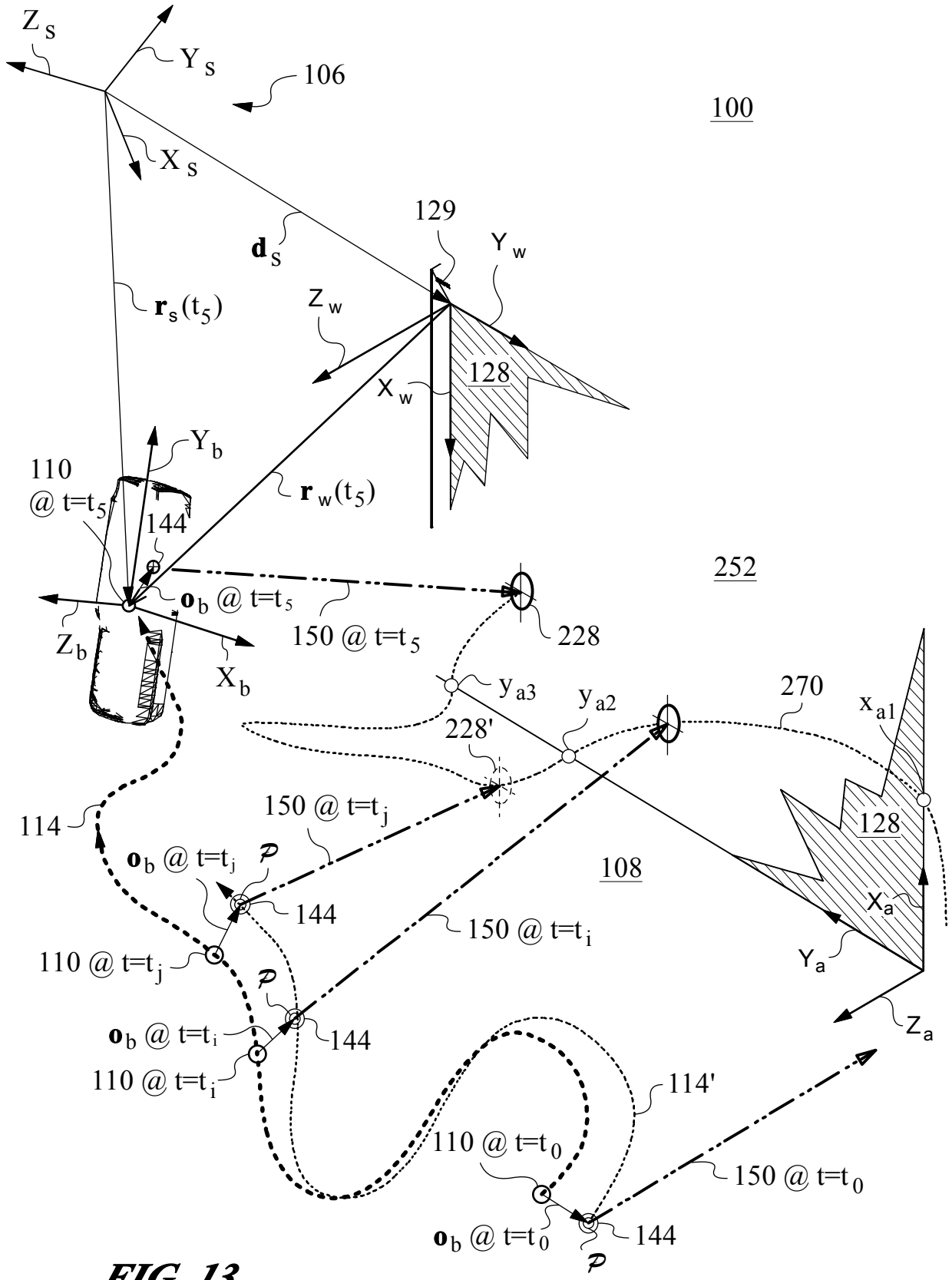

Z s
Y s
106
100
X s
129
d s
Y w
r s(t5)
Z w
128
X w
Y b
110
@ t=t5
r w(t5)
144
252
o b @ t=t5
150 @ t=t5
228
Z b
X b
y a3
y a2
270
x a1
228'
150 @ t=tj
114
o b @ t=tj
108
144
150 @ t=ti
110 @ t=tj
X a
Y a
o b @ t=ti
144
Z a
110 @ t=ti
114'
110 @ t=t0
150 @ t=t0
o b @ t=t0
144

FIG. 13

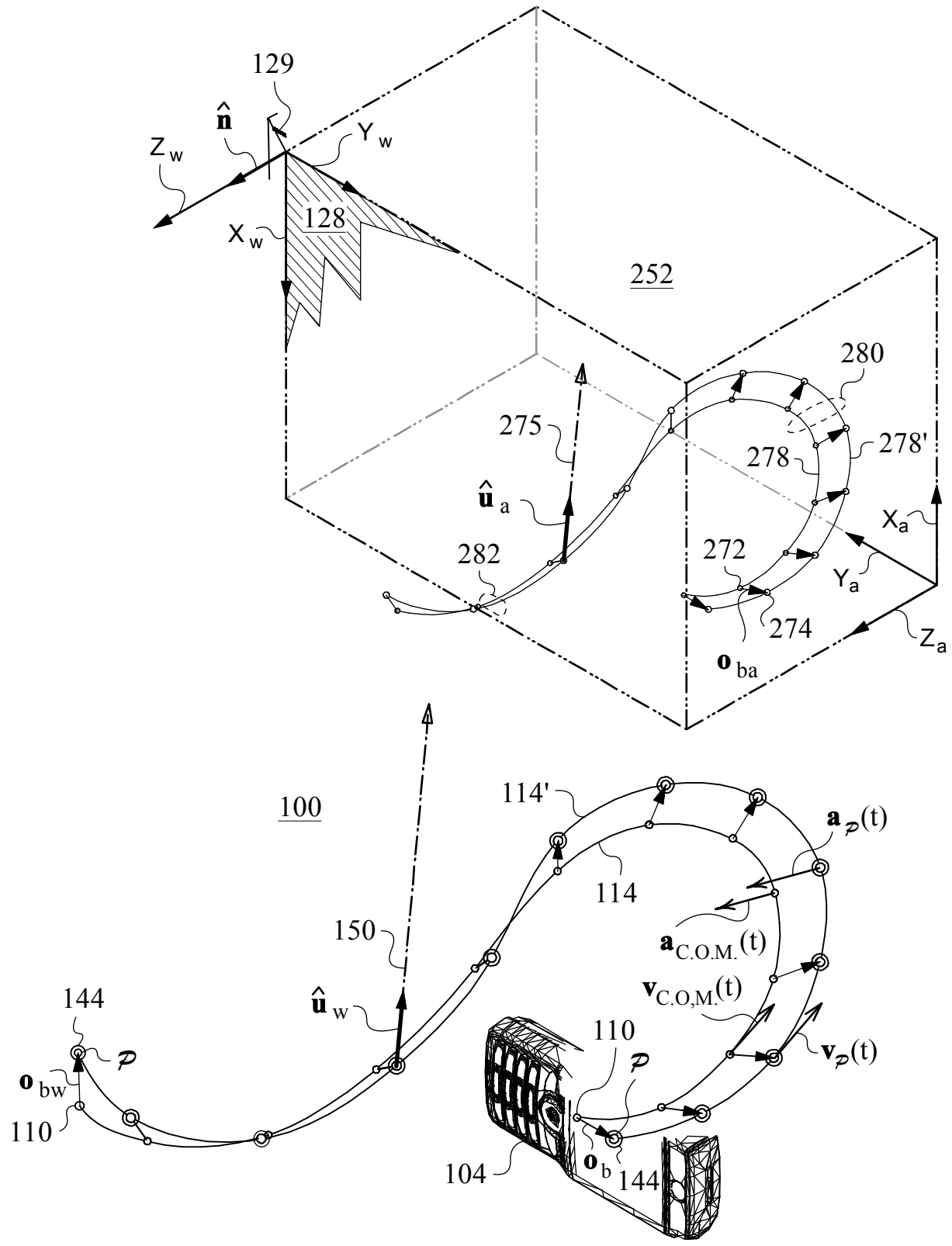

129
$\hat{\mathbf{n}}$
Z_w
Y_w
X_w
128
252
280
275
278
278'
$\hat{\mathbf{u}}_a$
X_a
282
272
Y_a
274
Z_a
$\mathbf{o}_{ba}$
100
114'
$\mathbf{a}_{\mathcal{P}}(t)$
114
150
$\mathbf{a}_{C.O.M.}(t)$
144
$\hat{\mathbf{u}}_w$
$\mathbf{v}_{C.O,M.}(t)$
110
$\mathcal{P}$
$\mathbf{v}_{\mathcal{P}}(t)$
$\mathbf{o}_{bw}$
110
$\mathcal{P}$
104
$\mathbf{o}_b$
144

FIG. 14

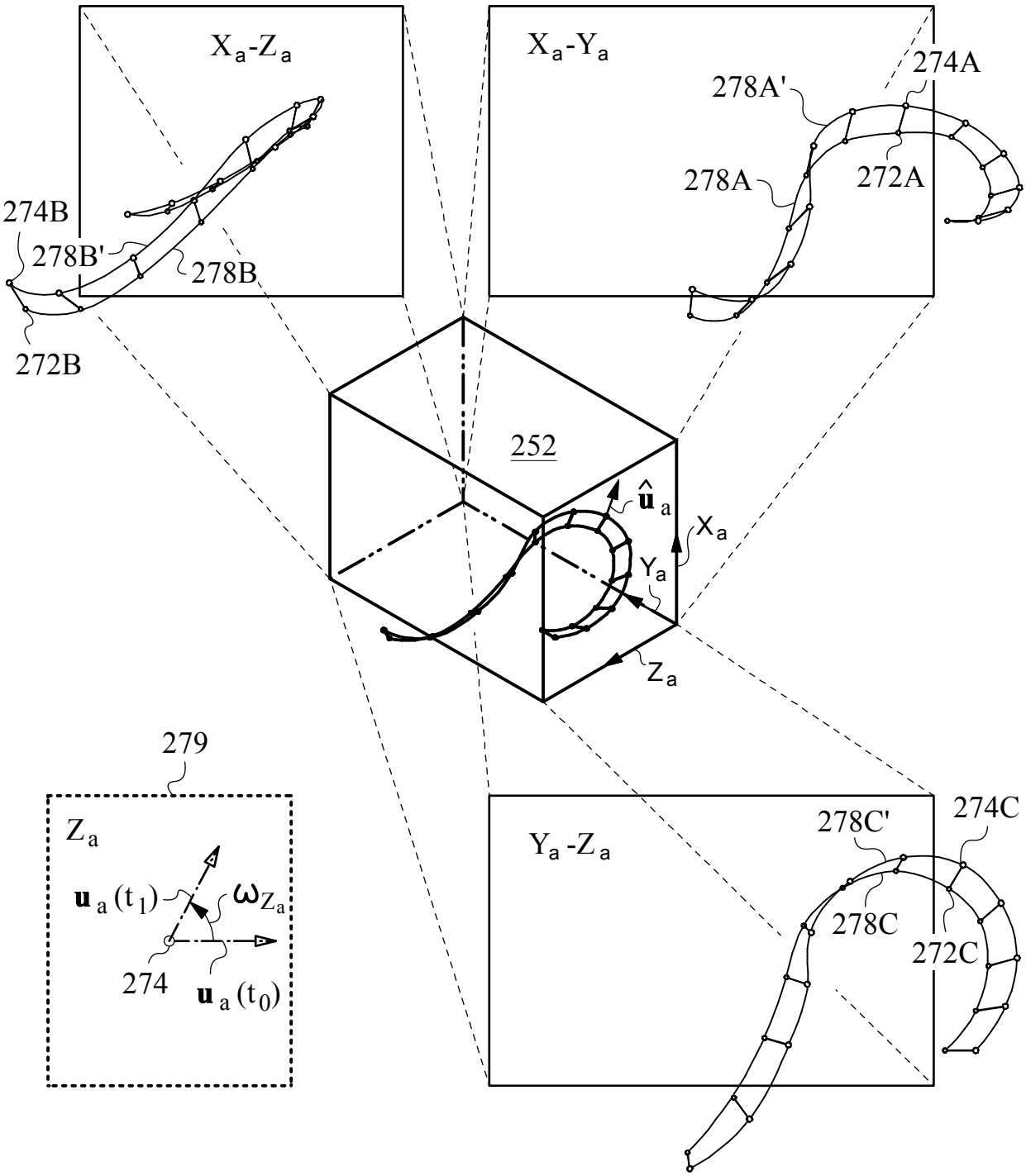

X_a-Z_a
X_a-Y_a
274A
278A'
278A
272A
274B
278B'
278B
272B
252
$\hat{\mathbf{u}}_a$
X_a
Y_a
Z_a
279
Z_a
$\mathbf{u}_a(t_1)$
ω_{Z_a}
274
$\mathbf{u}_a(t_0)$
Y_a-Z_a
278C'
274C
278C
272C

FIG. 15

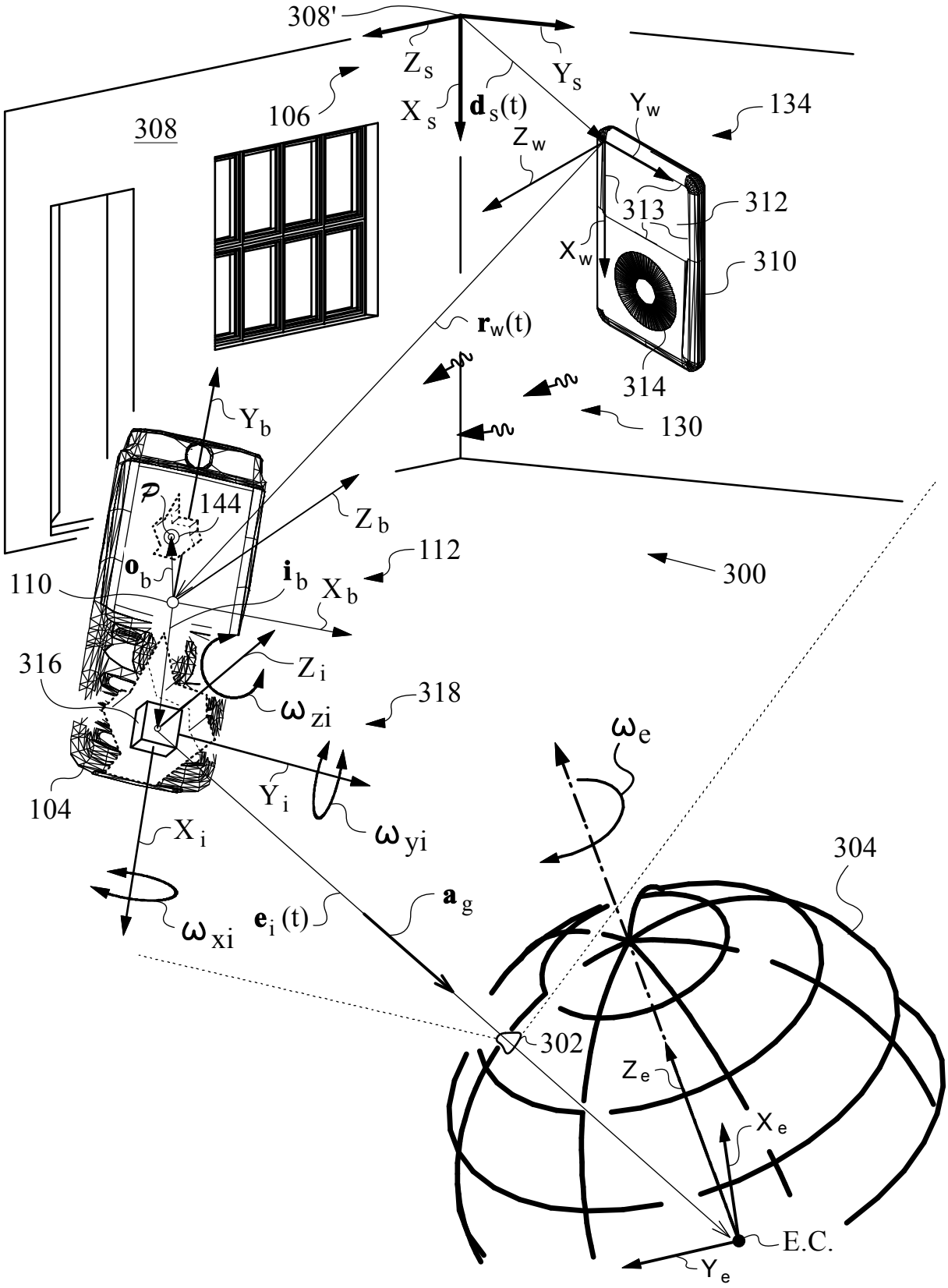

308'
Z s
Y s
X s
d s(t)
106
308
Y w
134
Z w
313
312
X w
310
r w(t)
314
130
Y b
Z b
144
112
110
o b
i b
X b
300
316
Z i
318
ω zi
ω e
Y i
ω yi
104
X i
304
ω xi
e i (t)
a g
302
Z e
X e
E.C.
Y e

FIG. 16

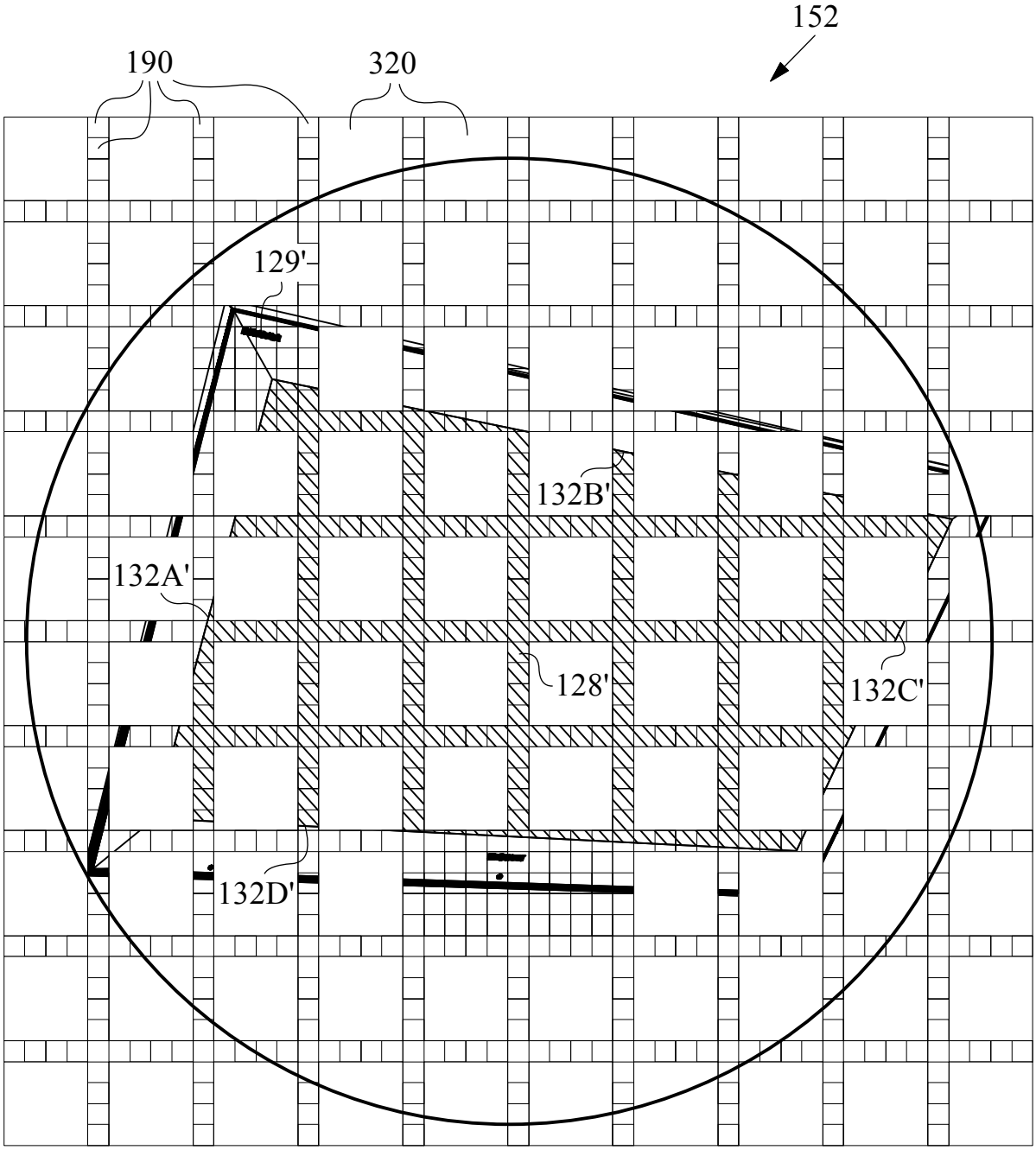


FIG. 17

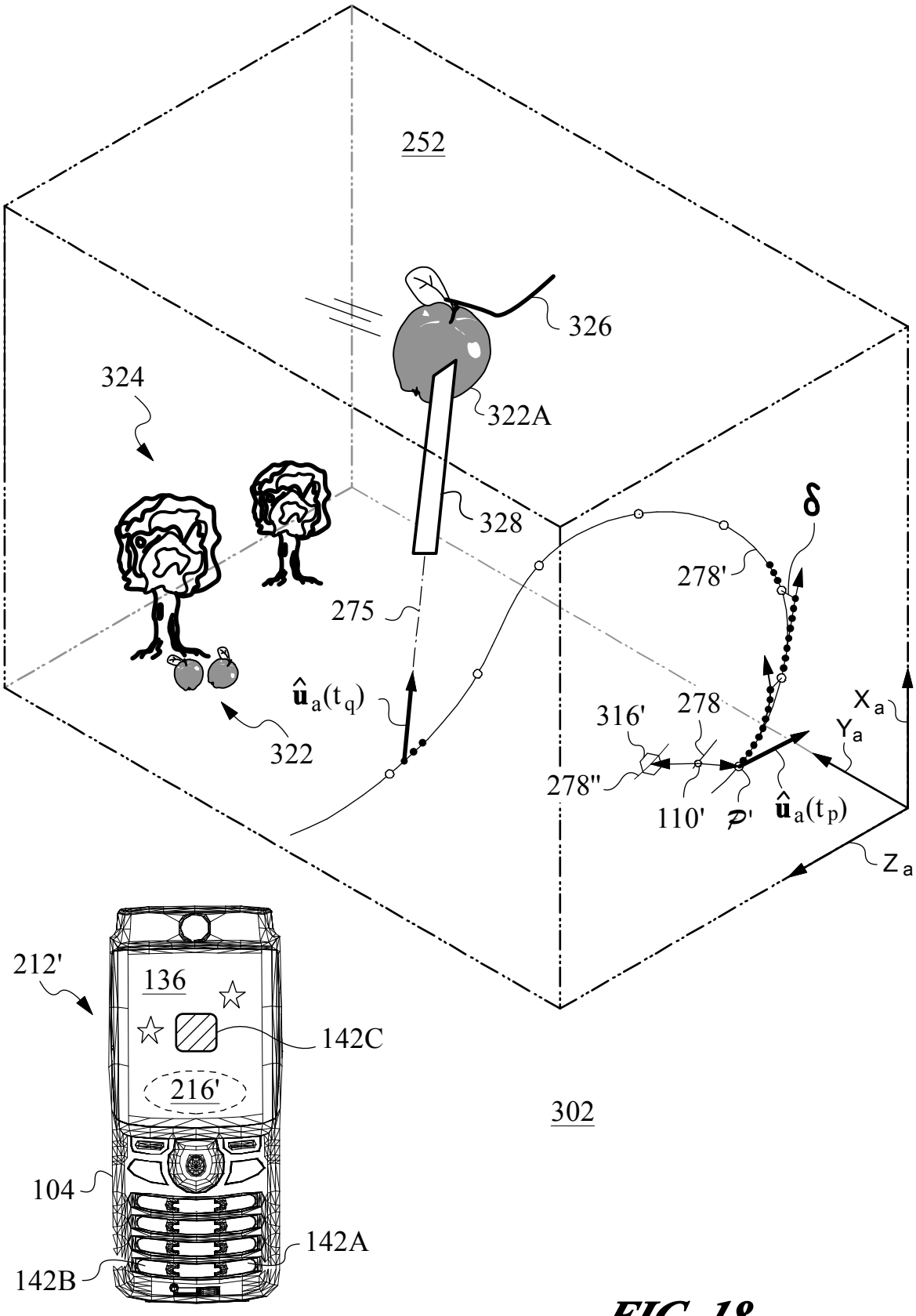

252
326
324
322A
328
δ
278'
275
$\hat{\mathbf{u}}_a(t_q)$
316'
278
X_a
Y_a
322
278"
110'
$\hat{\mathbf{u}}_a(t_p)$
Z_a
212'
136
142C
216'
302
104
142A
142B

FIG. 18

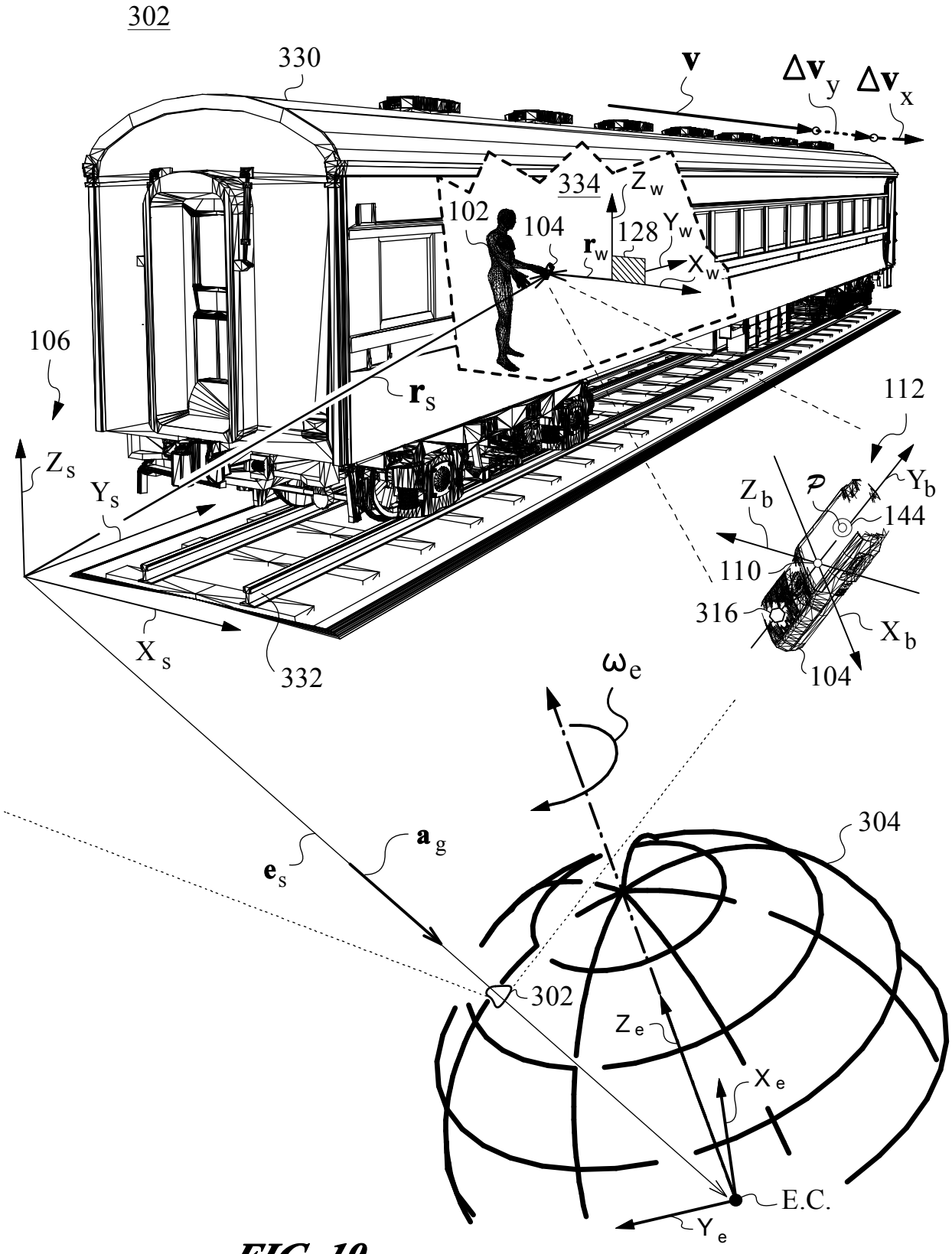

302
330
v
Δv_y
Δv_x
334
Z_w
102
104
128
Y_w
r_w
X_w
106
r_s
112
Z_s
Y_s
Z_b
P
Y_b
144
110
316
X_b
X_s
332
104
ω_e
304
a_g
e_s
302
Z_e
X_e
E.C.
Y_e

FIG. 19

DESCRIPTION OF THE DRAWING FIGURES

In order to explain in detail the smart phone application, the description will refer to the following drawings:

Figs. 1A-B are isometric views of a three-dimensional environment in which the absolute pose of a smart phone is employed for deriving user input.

Fig. 2 is an isometric view of the three-dimensional environment of Figs. 1A-B that illustrates in more detail the parameterization employed by a smart phone interface.

Figs. 3A-D are isometric views of a gimbal-type mechanism that aids in the visualization of the 3D rotation convention employed in describing the absolute orientation of the smart phone of Figs. 1A-B.

Figs. 4A-B are three-dimensional front and back views of the smart phone of Figs. 1A-B.

Fig. 5 is a three-dimensional schematic view of the on-board camera of the smart phone shown in Figs. 4A-B.

Figs. 6A-C are images of the three-dimensional environment of Figs. 1A-B acquired using three different types of camera lenses.

Figs. 7A-B are images of the environment as captured from the two vantage points corresponding to the first and second absolute poses shown in Figs. 1A and 1B respectively, as displayed on the screen of the smart phone.

Fig. 8 is a graph of a typical lens distortion curve.

Fig. 9 is a plan diagram of the CMOS photosensor and processing elements employed by the smart phone shown in Figs. 1A-B.

Fig. 10 is a three-dimensional view illustrating an interface deployed in the three-dimensional environment in which the absolute pose of the smart phone shown in Figs. 1A-B is employed for generating input.

Fig. 11 is a three-dimensional isometric view showing a detailed trajectory of the smart phone in the three-dimensional environment of Figs. 1A-B during operation of the interface.

Fig. 12 is a flow diagram illustrating the main steps executed by the interface.

Fig. 13 is a three-dimensional diagram showing in more detail how the input signal generated by the interface and being related to all six absolute pose parameters of the smart phone manipulated by the user is received and employed in the application.

Fig. 14 is a three-dimensional diagram illustrating how the signal related to all six absolute pose parameters of the smart phone is used to recover full trajectories of a point of interest (here point-of-view $\mathcal{P}$) and additional information about the smart phone in a three-dimensional digital environment of an application.

Fig. 15 is a three dimensional diagram illustrating the projections of full trajectories into 2D subspaces.

Fig. 16 is a three dimensional view of a preferred embodiment of the interface employing a relative motion sensor in addition to the CMOS camera.

Fig. 17 is a plan view of a preferred way of operating the CMOS camera employed in optical absolute pose recovery.

Fig. 18 is a three-dimensional diagram illustrating a gaming application employing the preferred embodiment of the interface operating the CMOS camera in the preferred way illustrated in Fig. 17 and using a relative motion sensor for interpolation.

Fig. 19 is a three dimensional perspective diagram which shows an additional advantage of the preferred embodiment employing optical absolute pose recovery supplemented by relative motion interpolation in a commonly encountered non-inertial reference frame.

DETAILED DESCRIPTION

The various aspects of a smart phone based interface will be best understood by initially referring to two isometric views of a real three-dimensional environment **100** as illustrated in **Figs. 1A-B.** A user **102** residing in environment **100,** which may be an indoor or an outdoor environment, is holding in his/her right hand **102'** an item **104** that also resides in environment **100.** It is important that item **104** be physically associated with user **102** such that the user's **102** physical disposition and changes therein are reflected by item **104.** In other words, the static positions, poses, actions, gestures and other movements of user **102** need to translate in some manner to corresponding absolute position(s) and orientation(s) of item **104** and/or changes to corresponding position(s) and orientations(s) of item **104.** For example, in the present case item **104** is a smart phone that is held in right hand **102'** by user **102** and manipulated.

Three-dimensional environment **100** has a spatial extent that may be described by orthogonal or non-orthogonal coordinate systems (e.g, linearly independent axes). Because of the efficiency of description, we will use orthogonal coordinates herein. Of course, it will be understood by those skilled in the art that linearly independent sets of basis vectors or other geometrical constructs can also be used. For example, surfaces and vectors bearing predetermined relationships to those surfaces (e.g., surface normal or tangent) can also be used to describe or parameterize three-dimensional environment **100.**

Three-dimensional environment **100** is associated with a stable reference frame **106.** For the sake of efficiency, stable frame **106** is parameterized by orthogonal coordinates. In particular, we chose a Cartesian coordinate system, referred to herein as stable coordinate system (X_s, Y_s, Z_s).

When parameterizing the various frames that we will encounter in the description, we will employ a certain convention. According to this convention, capital letters denote the axes of the coordinates that parameterize the frame and the subscripts on the axes refer to the frame (e.g., subscript "s" refers to stable frame **106**). The degrees of freedom as parameterized in the coordinates of the frame, e.g., displacements along axes X_s, Y_s and Z_s, will be denoted by lower case letters along with subscripts referring to that frame. Thus, in the stable coordinate system (X_s,Y_s,Z_s) parameterizing real three-dimensional environment **100** the actual numerical values of the three translational degrees of freedom (length, width and height or x, y and z) will be denoted by x_s, y_s and z_s. A similar convention will be employed for vectors, which will bear the subscript of the coordinate system in which they are expressed.

The orientation of the X_s-, Y_s-and Z_s-axes and the location of the origin (the (0,0,0) point) of stable coordinates (X_s,Y_s,Z_s) parameterizing stable frame **106** may be selected according to the interface designer's preferences. In the present example, the origin of stable coordinates (X_s,Y_s,Z_s) is set near the upper left corner of a wall **108** in environment **100**. The orthogonal or mutually perpendicular axes X_s, Y_s and Z_s make predetermined and known angles with respect to wall **108**.

The absolute pose of item **104**, in this case cell phone **104** in environment **100** includes its absolute position and its absolute orientation. The reason why the pose is referred to as absolute, is because it is expressed in stable frame **106** as parameterized by stable coordinates (X_s,Y_s,Z_s) that were selected by the designer. In contrast, many of today's input devices report relative position and/or relative orientation in 3D space. In some cases that is because the sensors on-board these devices can only make differential measurements. In other words, they provide measurements of changes in position and/or orientation without the ability to keep those measurements referenced to a stable external frame parameterized by a

stable coordinate system without some additional calibration mechanisms. Inertial units such as accelerometers and gyros are good examples of such relative motion sensors.

Now, to gain a deeper understanding, the absolute position of smart phone **104** and its absolute orientation will be introduced separately. These independent explanations will then be combined into one uniform description of absolute pose.

To define absolute position, a reference point whose (x_s,y_s,z_s) position will be tracked in stable coordinates (X_s,Y_s,Z_s) needs to be chosen on smart phone **104**. The choice of such reference point is arbitrary, but some conventions are more efficient than others. For example, in many cases it is convenient to choose the center of mass (C.O.M.) of smart phone **104** as the reference point. In other cases, a protruding point or some other prominent or important aspect of smart phone **104** may be selected. In still other cases, the point-of-view of an on-board optical sensing unit such as a directional photosensor, e.g., a digital camera or a lensed position-sensing device (PSD), may be selected as the reference point. The choice will depend on the type of smart phone **104**, the software application and the interface.

As shown in **Fig. 2**, in the present embodiment the center of mass (C.O.M.) of phone **104** is chosen as a reference point **110**. Further, to simplify the description, Cartesian body coordinates (X_b,Y_b,Z_b) whose origin coincides with C.O.M. **110** are associated with a moving frame **112** of phone **104** itself. To distinguish body coordinates (X_b,Y_b,Z_b) from stable coordinates (X_s,Y_s,Z_s) that describe stable frame **106** of three-dimensional environment **100**, we use the subscript letter "b" (b for body) throughout the present description and in the drawing figures.

A person skilled in the art will realize that body coordinates (X_b,Y_b,Z_b) are a useful tool for parameterizing moving frame **112**.

Indeed, body coordinates are a very well-known tool in classical mechanics for describing both the absolute position and the absolute orientation of bodies undergoing unconstrained motion in 3D space (or other spaces). Once again, such person will also realize that orthogonal and non-orthogonal conventions and systems may be employed in this description. The present description adheres to Cartesian coordinates merely for reasons of explanatory clarity and convenience without implying any limitations as to the types of descriptions and body coordinate choices that are available to the interface designer.

Body coordinates (X_b,Y_b,Z_b) centered on C.O.M. **110** of phone **104** allow us to define the absolute position of phone **104**, or more precisely the absolute position of its C.O.M. **110** in environment **100**. The absolute position of C.O.M. **110** can change along any of the three directions X_s, Y_s and Z_s defined by stable coordinates (X_s,Y_s,Z_s) that parameterize the three translational degrees of freedom in stable frame **106** established in environment **100**. In fact, successive absolute positions of C.O.M. **110** in time or, equivalently, the sequence of such positions of the origin of body coordinates (X_b,Y_b,Z_b), define an absolute trajectory **114** of the phone's **104** C.O.M. **110** through environment **100**.

To illustrate the above point, **Fig. 1A** shows user **102** holding cell phone **104** in his/her right hand **102'** in a first absolute position in environment **100** at a time t_1. **Fig. 1B** shows same user **102** holding cell phone **104** in his/her left hand **102"** in a second absolute position at a later point in time t_5. Trajectory **114** traversed by phone **104**, and specifically its C.O.M. **110** in traveling between these two positions, including the change over from right hand **102'** to left hand **102"** is shown in **Fig. 2**. Note that in the present embodiment in moving along trajectory **114** the absolute position of phone **104** or its C.O.M. **110** changes in all three degrees of translational freedom as parameterized by directions X_s, Y_s and Z_s. In other words, the absolute position of phone **104** exhibits three translational degrees

of freedom whose numerical values in stable coordinates (X_s,Y_s,Z_s) are expressed by x_s, y_s and z_s.

In order to simplify the description of trajectory **114** and express it directly in stable coordinates (X_s,Y_s,Z_s) we employ the concept of a vector $\mathbf{r}_s$. To distinguish vectors from scalars, we will designate them in boldfaced letters. To remain consistent, vectors will also carry the subscript of the coordinate system in which they are expressed (i.e., "s" in the present case). Vector $\mathbf{r}_s$ is represented by an ordered triple of numbers, namely the values x_s, y_s and z_s that represent the absolute position of C.O.M. **110**. Differently put, these three numbers are the numerical values of displacements along X_s-, Y_s- and Z_s-axes of stable coordinates (X_s,Y_s,Z_s) that need to be taken in order to arrive at C.O.M. **110** when starting out from the origin of stable coordinates (X_s,Y_s,Z_s). Thus, vector $\mathbf{r}_s$ corresponds in this representation to (x_s,y_s,z_s). It should be noted for completeness, that other vector representations are also available. For example, a vector may be represented by a magnitude and direction (e.g., in spherical coordinates) or a combination of the two (e.g., a magnitude and a direction in a 2D subspace together with a rectilinear coordinate in a third dimension).

Furthermore, in order to keep track of vector $\mathbf{r}_s$ in time, we express vector $\mathbf{r}_s$ as a function of time, i.e., $\mathbf{r}_s=\mathbf{r}_s(t)$. The two times indicated in **Figs. 1A-B** & **2** are: time t_1 when user **102** held phone **104** in right hand **102'** in the absolute pose shown in **Fig. 1A** and time t_5 when user **102** held phone **104** in left hand **102''** in the absolute pose shown in **Fig. 1B**. In accordance with our convention, we thus designate the corresponding vectors $\mathbf{r}_s(t_1)$ and $\mathbf{r}_s(t_5)$.

Now, in addition to absolute position, the absolute pose also includes the absolute orientation of phone **104**. As in the case of the absolute position, absolute orientation is expressed in stable coordinates (X_s,Y_s,Z_s) with the aid of body coordinates (X_b,Y_b,Z_b) centered on C.O.M. **110** of phone **104**. For a rigid body such as phone

104, absolute orientation exhibits three rotational degrees of freedom (i.e., rotation around axes X_b, Y_b, Z_b or other axes). Because rotations in 3D do not commute, in other words, the final orientation after several rotations in 3D depends on the order of the rotations, a careful and consistent description needs to be selected to describe absolute orientation of phone **104**. A person skilled in the art will realize that many such descriptions exist and indeed any of them can be used herein without limitation.

Figs. 3A-D illustrate a particular orthogonal rotation convention that takes the non-commutative nature of 3D rotations into account and is employed in the present embodiment. Specifically, this convention describes the absolute orientation of phone **104** in terms of three rotation angles α_b, β_b and γ_b. Here, the rotations are taken around the three body axes X_b, Y_b, Z_b, which are initially aligned with the axes of stable coordinates (X_s, Y_s, Z_s) that parameterize stable frame **106** in environment **100**. We keep the subscript "b" on rotation angles α_b, β_b and γ_b in order to remind ourselves that they are taken in body coordinates (X_b, Y_b, Z_b). However, since rotations do not require the definition of any new axes, they are expressed in lowercase letters. These letters will be also used to express the actual numerical values of the corresponding rotations to avoid the introduction of excessive notational rigor.

Our choice of rotation convention ensures that C.O.M. **110** of phone **104** does not move during any of the three rotations. It thus remains a reliable reference point for tracking trajectory **114** of C.O.M. **110** of phone **104** through environment **100**. A person skilled in the art will recognize the importance of this feature of the 3D rotation convention chosen herein and that similar considerations are employed in navigating terrestrial vehicles, marine vehicles, aircraft, spaceships and other navigable vehicles, objects and craft. Indeed, such convention may also be used to describe free or unconstrained motion of arbitrary objects in 3D space.

Fig. 3A shows phone **104** in an initial, pre-rotated condition centered in a gimbal mechanism **118** that will mechanically constrain the rotations defined by angles α_b, β_b and γ_b. Mechanism **118** has three progressively smaller concentric rings or hoops **120, 122, 124.** Rotating joints **121, 123** and **125** permit hoops **120, 122, 124** to be respectively rotated in an independent manner. For purposes of visualization of the present 3D rotation convention, phone **104** is rigidly fixed to the inside of third hoop **124** either by an extension of joint **125** or by any other suitable mechanical means (not shown).

In the pre-rotated state, the axes of body coordinates (X_b, Y_b, Z_b) parameterizing moving frame **112** of phone **104** are triple primed (X_b''', Y_b''', Z_b''') to better keep track of body coordinate axes after each of the three rotations. In addition, the pre-rotated axes (X_b''', Y_b''', Z_b''') of body coordinates (X_b, Y_b, Z_b) are aligned with axes X_s, Y_s and Z_s of stable coordinates (X_s, Y_s, Z_s) that parameterize stable frame **106** in environment **100**. However, pre-rotated axes (X_b''', Y_b''', Z_b''') are displaced from the origin of stable coordinates (X_s, Y_s, Z_s) by vector $\mathbf{r}_s$ introduced and explained above. C.O.M. **110** is at the origin of body coordinates (X_b, Y_b, Z_b) and at the center of gimbal mechanism **118**.

The first rotation by angle α_b is executed by rotating joint **121** and thus turning hoop **120,** as shown in **Fig. 3B**. Note that since body axis Z_b''' of phone **104** (see **Fig. 3A**) is co-axial with rotating joint **121** the physical turning of hoop **120** is equivalent to this first rotation in body coordinates (X_b, Y_b, Z_b) of phone **104** around body Z_b''' axis. In the present convention, all rotations are taken to be positive in the counter-clockwise direction as defined with the aid of the right hand rule (with the thumb pointed in the positive direction of the coordinate axis around which the rotation is being performed). Hence, angle α_b is positive and in this visualization it is equal to 30°.

After each of the three rotations is completed, body coordinates (X_b, Y_b, Z_b) are progressively unprimed to denote how many rotations have already been executed. Thus, after this first rotation by angle α_b, the axes of body coordinates (X_b, Y_b, Z_b) are unprimed once and designated ($X_b^{''}, Y_b^{''}, Z_b^{''}$) as indicated in **Fig. 3B**.

Fig. 3C depicts the second rotation by angle β_b. This rotation is performed by rotating joint **123** and thus turning hoop **122.** Since joint **123** is co-axial with once rotated body axis $X_b^{''}$ (see **Fig. 3B**) such rotation is equivalent to second rotation in body coordinates (X_b, Y_b, Z_b) of phone **104** by angle β_b around body axis $X_b^{''}$. In the counter-clockwise rotation convention we have adopted angle β_b is positive and equal to 45˚. After completion of this second rotation, body coordinates (X_b, Y_b, Z_b) are unprimed again to yield twice rotated body axes ($X_b^{'}, Y_b^{'}, Z_b^{'}$).

The result of the third and last rotation by angle γ_b is shown in **Fig. 3D.** This rotation is performed by rotating joint **125,** which turns innermost hoop **124** of gimbal mechanism **118.** The construction of mechanism **118** used for this visualization has ensured that throughout the prior rotations, twice rotated body axis $Y_b^{'}$ (see **Fig. 3C**) has remained co-axial with joint **125.** Therefore, rotation by angle γ_b is a rotation in body coordinates (X_b, Y_b, Z_b) parameterizing moving frame **112** of phone **104** by angle γ_b about body axis $Y_b^{'}$.

This final rotation yields the fully rotated and now unprimed body coordinates (X_b, Y_b, Z_b). In this example angle γ_b is chosen to be 40˚, representing a rotation by 40˚ in the counter-clockwise direction. Note that in order to return fully rotated body coordinates (X_b, Y_b, Z_b) into initial alignment with stable coordinates (X_s, Y_s, Z_s) the order of rotations by angles α_b, β_b and γ_b needs to be taken in exactly the reverse order (this is due to the order-dependence or non-commuting nature of rotations in 3D space mentioned above).

It should be understood that mechanism **118** was employed for illustrative purposes to show how any 3D orientation of phone **104** consists of three rotational degrees of freedom. These non-commuting rotations are described or parameterized by rotation angles α_b, β_b and γ_b around body axes Z_b''', X_b'' and finally Y_b'. What is important is that this 3D rotation convention employing angles α_b, β_b, γ_b is capable of describing any possible orientation that phone **104** may assume in environment **100**.

The description of trajectory **114** of C.O.M. **110** of phone **104** in environment **100** has been shown to have three translational degrees of freedom; here described in terms of displacements along X_s-, Y_s- and Z_s-axes of stable coordinates (X_s,Y_s,Z_s). A compact description of trajectory **114** in terms of vector $\mathbf{r}_s=(x_s,y_s,z_s)$ has also been introduced. We have additionally shown that the rotation of phone **104** can be described by three rotational degrees of freedom; parameterized by rotations around body axes Z_b''', X_b'' and Y_b' by angles α_b, β_b and γ_b in that order. The rotations are executed while C.O.M. **110** remains fixed in stable coordinates (X_s,Y_s,Z_s). Thus, the rotations do not change the definition of trajectory **114** as they do not affect the value of vector $\mathbf{r}_s$.

Since the descriptions of absolute position and absolute orientation of phone **104** using body coordinates (X_b,Y_b,Z_b) and stable coordinates (X_s,Y_s,Z_s) are mutually independent, they can be combined. Such combination of vector $\mathbf{r}_s$ and rotation angles $(\alpha_b,\beta_b,\gamma_b)$ provides a compact description of the six (6) degrees of freedom available to phone **104** in three-dimensional environment **100**. Specifically, the description of the six (6) degrees of freedom that will be employed herein is a direct combination of vector $\mathbf{r}_s$ with the rotation angles, namely: $(x_s,y_s,z_s,\alpha_b,\beta_b,\gamma_b)$. To avoid future confusion and indicate that body axes X_b, Y_b and Z_b were originally aligned with stable coordinate axes X_s, Y_s and Z_s, we will add the subscript "s" on the

three angles, thus referring to the degrees of freedom as: $(x_s, y_s, z_s, \alpha_{sb}, \beta_{sb}, \gamma_{sb})$.

The joint description of the absolute position and the absolute orientation of phone **104** in stable coordinates (X_s, Y_s, Z_s) is a parameterization of the absolute pose of phone **104**. Turning back to **Figs. 1A-B**, we can thus specify how phone **104** is held in stable coordinates (X_s, Y_s, Z_s) by user **102** in different absolute positions and in various absolute orientations at times t_1 and t_5 in terms of the phone's **104** absolute pose parameters $(x_s, y_s, z_s, \alpha_{sb}, \beta_{sb}, \gamma_{sb})$ that use C.O.M. **110** as the reference point. In the present description absolute pose is thus parameterized by A.P.$=(x_s, y_s, z_s, \alpha_{sb}, \beta_{sb}, \gamma_{sb})$ where A.P.=A.P.(t), meaning that the absolute pose is a function of time. Indeed, since phone **104** can be moved in arbitrary ways by user **102** (unconstrained 3D motion) all of the components of absolute pose A.P. are typically functions of time.

Of course, many descriptions including those utilizing other concepts and coordinates could have been employed to describe or parameterize the absolute pose of phone **104** in stable frame **106**. As a result, we need to clearly distinguish the six degrees of freedom available to phone **104** as a rigid body, from the description chosen to parameterize these six degrees of freedom. It is worth stressing that the model or description of the degrees of freedom is not the same as the degrees of freedom themselves. The model is merely a way to describe and talk about the degrees of freedom with the aid of the chosen parameters.

In the present embodiment, absolute pose will be expressed by the combination of vector $\mathbf{r}_s$ with the rotation angles as defined above, namely A.P.$=(x_s, y_s, z_s, \alpha_{sb}, \beta_{sb}, \gamma_{sb})$. Since in our model these are descriptors of the six degrees of freedom, we will refer to $(x_s, y_s, z_s, \alpha_{sb}, \beta_{sb}, \gamma_{sb})$ as absolute pose parameters dictated by our specific parameterization of the absolute pose of phone **104** in stable

frame **106**. In doing so, we also recognize the fact that other types of parameters can be deployed in other mathematical conventions and descriptions. However, a person skilled in the art, will recognize that at their core, all of these descriptions are mathematically equivalent, as they express the fundamental geometrical properties of rigid body motion in 3D space.

It should also be recognized that additional degrees of freedom are in general available to bodies in 3D space. In most conventional approaches, these are the roto-vibrational degrees of freedom. Although they may be important for some applications, e.g., when phone **104** consists of elements that move with respect to each other (such as in the case of a flip-phone), we will not explicitly keep track of these in the present embodiments. A person skilled in the art will understand how to parameterize these additional degrees of freedom and use them in a complete description of the absolute pose of phone **104** if and as necessary.

As seen in **Figs. 1A-B**, the interface further requires at least one stationary object **126** that has at least one feature **128** that is detectable via an electromagnetic radiation **130**. In this embodiment, stationary object is a television **126** sitting on a table **208** and the detectable feature is its display screen **128**. In the present embodiment, object **126** is thus stationary in stable frame **126**.

Electromagnetic radiation **130** by which screen **128** is detectable is predominantly emitted by display screen **128** during operation. In general, however, electromagnetic radiation **130** may include ambient radiation or any radiation purposely reflected from screen **128**.

It is important that feature **128**, in this case screen **128**, present a sufficient number and type of non-collinear optical inputs to establish a stable frame **134** in three-dimensional environment **100**. In general, stable frame **134** may not be the same as stable frame **106**. In fact, the positions and orientations of non-collinear optical

inputs of screen **128** may be stationary, moving or even unknown in stable frame **106**. We will discuss all situations below.

In the present embodiment, screen **128** defines a plane in 3D space of environment **100** and any number of points or regions on it, whether during active display operation or not, can be selected as the non-collinear optical inputs. Conveniently, it is edges **132** of screen **128** that are chosen as the non-collinear optical inputs. Edges **132** are line-like inputs and are mutually non-collinear. The reason for this choice is that edges **132** are most likely to provide high optical contrast and thus be more easily detectable via electromagnetic radiation **130** than any other portions of screen **128**. In addition, one other non-collinear optical input from television **126** is selected to break the intrinsic symmetry of the rectangle of screen **128**. In the present case, that additional non-collinear optical input is obtained from a feature or marking **129** on the upper left corner of television **126**. Alternatively, a feature displayed on screen **128** or any other feature associated with television **128** can be used for this purpose. Marking **129** is a point-like input, or, if its area is used, it is an area-like input.

Preferably, all four edges **132** of screen **128** and marking **129** are used for non-collinear optical inputs to establish stable frame **134**. Frame **134** is parameterized by frame coordinates which we will refer to as workspace or world coordinates (X_w, Y_w, Z_w) for the purposes of the application. The reasons for this choice will become apparent later.

In the present embodiment, the origin of world coordinates (X_w, Y_w, Z_w) is chosen to be coincident with the upper left corner of screen **128**. A person skilled in the art will recognize, however, that as few as four point-like, fixed non-collinear optical inputs, e.g., in the form of point sources or point-like inputs, are sufficient to establish stable frame **134** in terms of its parameterization by world coordinates (X_w, Y_w, Z_w). Even fewer points may be sufficient when more

information about these points is provided. It should be noted that non-collinear in the sense employed in the present invention, (since any two points will always be collinear according to Euclidean geometry) means that the points are not all mutually collinear and that they establish a convex hull, which will be defined below.

In all embodiments, world coordinates (X_w,Y_w,Z_w) are central to the interface because they define the position and orientation of the stationary object or television **126** in stable frame **134**. In other words, although absolute pose expressed with absolute pose parameters $(x_s,y_s,z_s,\alpha_{sb},\beta_{sb},\gamma_{sb})$ of phone **104** is completely defined in stable coordinates (X_s,Y_s,Z_s) using C.O.M. **110** as the reference point, for the purposes of many interfaces and applications these absolute pose parameters $(x_s,y_s,z_s,\alpha_{sb},\beta_{sb},\gamma_{sb})$ have to be related to world coordinates (X_w,Y_w,Z_w). In some cases, world coordinates (X_w,Y_w,Z_w) are more important that stable coordinates (X_s,Y_s,Z_s), as they may represent the coordinates of a workspace for human user **102**. In fact, world coordinates should be understood to subsume coordinates for workspaces, gaming spaces, operation spaces and the like.

The relationship between stable frames **106** and **134** and between their descriptions by stable coordinates (X_s,Y_s,Z_s) and world coordinates (X_w,Y_w,Z_w) can be captured in many ways. For example, one can fix the absolute pose of stationary object or television **126** in stable frame **106** and measure its position and orientation in it. For this purpose we introduce a vector $\mathbf{d}_s$ corresponding to the displacement of upper left corner of screen **128**. Vector $\mathbf{d}_s$ thus marks the displacement of the origin of world coordinates (X_w,Y_w,Z_w) parameterizing stable frame **134** from the origin of stable coordinates (X_s,Y_s,Z_s). It is helpful in this situation if stable coordinates (X_s,Y_s,Z_s) are coincident and aligned with a corner in environment **100**, e.g., the corner of a room. In such cases, vector $\mathbf{d}_s$ can be measured while arranging television **126**, and its absolute orientation in stable coordinates (X_s,Y_s,Z_s) could be ensured by aligning screen **128** plane-parallel to wall **108**.

Alternatively, when stationary object **126** is designed to stay in the same place in environment **100**, which is usually true of television **126** but may not be true of other objects (e.g., mobile objects) in other embodiments, one can simply choose world coordinates (X_w, Y_w, Z_w) of frame **134** to be the same as stable coordinates (X_s, Y_s, Z_s) that parameterize frame **106** of environment **100**. In this case, it is frame **134** and hence the position and orientation of television **126** in environment **100** that defines stable coordinates (X_s, Y_s, Z_s) and concurrently world coordinates (X_w, Y_w, Z_w).

In the embodiment depicted in **Figs. 1A-B** and in **Fig. 2** stable coordinates (X_s, Y_s, Z_s) are not coincident and not collinear with world coordinates (X_w, Y_w, Z_w). However, vector $\mathbf{d}_s$ is known (e.g., by direct measurement with a measuring tape) and the relative orientation of axes X_w, Y_w and Z_w with respect to axes X_s, Y_s and Z_s is also known. For example, direction cosines or even the same rotation convention as described in **Figs. 3A-D** can be used to describe the relative difference in orientation between stable coordinates (X_s, Y_s, Z_s) and world coordinates (X_w, Y_w, Z_w) with three rotation angles α_{sw}, β_{sw} and γ_{sw}.

In the present embodiment, a coordinate transformation between stable coordinates (X_s, Y_s, Z_s) and world coordinates (X_w, Y_w, Z_w) can be used to translate absolute pose parameters $(x_s, y_s, z_s, \alpha_{sb}, \beta_{sb}, \gamma_{sb})$ of phone **104** into its absolute pose parameters $(x_w, y_w, z_w, \alpha_{wb}, \beta_{wb}, \gamma_{wb})$ in world coordinates (X_w, Y_w, Z_w). In this transformation we introduce vector $\mathbf{r}_w$ from the origin of world coordinates (X_w, Y_w, Z_w) to C.O.M. **110** of phone **104**. In addition, the rotational angles α_{sb}, β_{sb} and γ_{sb} have to be converted into the orientation of fully rotated body coordinates (X_b, Y_b, Z_b) with respect to world coordinates (X_w, Y_w, Z_w) rather than stable coordinates (X_s, Y_s, Z_s). Such conversion is performed with the aid of a rotation matrix $\mathbf{R}_{sw}$ that keeps track of the rotations that are required to obtain alignment between the axes of stable

coordinates (X_s,Y_s,Z_s) and the axes of world coordinates (X_w,Y_w,Z_w). Note that in representing matrices we extend our convention adopted for vectors and designate such rotation matrices by boldfaced letters. However, because a matrix is higher-order than a vector (vectors are 1^{st} order tensors, matrices are 2^{nd} order tensors, while scalar quantities can be thought of as 0^{th} order tensors) we use capital letters for denoting matrices.

The resulting absolute pose in world coordinates (X_w,Y_w,Z_w) expressing the six degrees of freedom of phone **104** is then parameterized by $(x_w,y_w,z_w,\alpha_{wb},\beta_{wb},\gamma_{wb})$. More precisely, $\mathbf{r}_w=(x_w,y_w,z_w)$ is the new displacement vector of C.O.M. **110** and $(\alpha_{wb},\beta_{wb},\gamma_{wb})$ are the transformed angles expressing the orientation of phone **104**. As remarked above, the angles are obtained by applying rotation matrix $\mathbf{R}_{sw}$ containing rotation angles $(\alpha_{sw},\beta_{sw},\gamma_{sw})$ and thus adjusting for the misalignment of coordinate axes between stable coordinates (X_s,Y_s,Z_s) and world coordinates (X_w,Y_w,Z_w).

A person skilled in the art will recognize that coordinate transformations are routine operations. They are described by corresponding vector operations to account for displacements or offsets and rotation matrices to account for the rotations. It is important in doing such transformations to preserve the correct handedness of the coordinates chosen (right-handed or left-handed) in order to avoid improper solutions. The corresponding mathematics will not be discussed herein as it has been well known for several centuries. An excellent background on coordinate transformations in many different coordinate systems is found in G.B. Arfken and H.J. Weber, "Mathematical Methods for Physicists", Harcourt Academic Press, 5^{th} Edition.

Stable coordinates (X_s,Y_s,Z_s) typically parameterize a stable and stationary frame of reference **106** in which user **102** resides with phone **104** (e.g., on the surface of planet Earth). Also note that in some rare cases stable coordinates (X_s,Y_s,Z_s) may even parameterize an

actual inertial frame, e.g., on a spaceship in outer space. Meanwhile, object or television **126** and world coordinates (X_w,Y_w,Z_w) defined with the aid of its non-collinear features may be moving in environment **100**, i.e., its position in stable coordinates (X_s,Y_s,Z_s) that parameterize stable frame **106** in environment **100** may be changing. In this case, the coordinate transformation between stable coordinates (X_s,Y_s,Z_s) and world coordinates (X_w,Y_w,Z_w) is time-dependent may need to be updated on a frequent basis.

Embodiments in which stationary object **126** is actually at rest in stable coordinates (X_s,Y_s,Z_s) and is thus also stationary for the purposes of the interface and the application of the present invention are the simplest. In these embodiments, the coordinate transformation between stable coordinates (X_s,Y_s,Z_s) that parameterize frame **106** in environment **100** and world coordinates (X_w,Y_w,Z_w) that are used to parameterize frame **134** in environment **100** for the purposes of the interface and the application need only be computed once. Of course, both stable coordinates (X_s,Y_s,Z_s) and world coordinates (X_w,Y_w,Z_w) may also specify some other common non-inertial frame in which user **102**, item **104** and stationary object **126** all reside (e.g., when aboard a plane, train, car or other aircraft or terrestrial vehicle that undergoes accelerated or curvilinear motion). Those situations are more complex and will be discussed later (e.g., see **Figs. 16 & 17** and corresponding description).

Figs. 4A-B will now be referred to in order to develop a deeper understanding of phone **104** and its capabilities. **Fig. 4A** is a three-dimensional front view of phone **104** with a central cut-out section that exposes the reference point or C.O.M. **110**. Phone **104** has on-board resources that include a display screen **136**, speakers **138** adapted for a human ear, microphone **140** adapted for a human mouth and selection buttons **142**. Buttons **142** include dial buttons as well as other selection buttons that allow user **102** to activate a unit **144** on-board the phone **104** for receiving electromagnetic radiation **130**.

Fig. 4B shows the back of phone **104** where unit **144** resides. On-board unit **144** in this case is an on-board camera with an imaging lens **146**. Lens **146** has a field of view **148** and an optical axis **150**. Field of view **148** is sufficiently large to permit phone **104** to image a significant portion of environment **100**, and especially of television **126** from the various absolute poses in which user **102** is expected to hold phone **104**. Further, the letter $\mathcal{P}$ designates a point-of-view (P.O.V.) of camera **144**. In this embodiment, phone **104** is also configured to display a view of environment **100** as imaged by camera **144** from point of view $\mathcal{P}$ on its display screen **136** (although this is an optional feature from the point of view of the interface of present invention).

Note that point-of-view $\mathcal{P}$ of camera **144** does not coincide with C.O.M. **110** of phone **104** in this embodiment. Indeed, few if any phones are built in a way to ensure that C.O.M. **110** coincides with the point(s)-of-view of their on-board camera(s). Thus, there is usually an offset vector $\mathbf{o}_b$ (expressed in body coordinates (X_b,Y_b,Z_b) as defined below; also see **Fig. 5**) between C.O.M. **110** of phone **104** and point-of-view $\mathcal{P}$ of camera **144**.

Offset vector $\mathbf{o}_b$ is used to recover and properly report absolute pose of phone **104** in terms of absolute position parameters A.P. This is necessary because the determination of absolute pose of phone **104** is based on radiation **130** that is captured and imaged by camera **144** from point-of-view $\mathcal{P}$ rather than from the "point-of-view" of C.O.M. **110**. Transformation of spatial information from point-of-view $\mathcal{P}$ to C.O.M. **110** is once again accomplished by a coordinate transformation. In fact, since C.O.M. **110** and point-of-view $\mathcal{P}$ are fixed with respect to each other, the transformation simply involves adjustment of absolute pose by vector $\mathbf{o}_b$ without any rotations given proper choice of camera coordinates (e.g., alignment of camera image plane X_i-Y_i with the X_b-Y_b plane of body coordinates). In the event of lack of alignment, an additional rotation matrix will need to be applied. The details of

coordinate transformations required are well known and, as pointed out above, are discussed in detail in G.B. Arfken and H.J. Weber, "Mathematical Methods for Physicists", Harcourt Academic Press, 5th Edition.

Fig. 5 is a schematic view showing the relevant parts of on-board camera **144** required to support an interface of the present invention. Here, field of view **148** of lens **146** is parameterized in terms of a cone angle **Σ** measured from optical axis **150**. Radiation **130** arriving from field of view **148** within cone angle **Σ** is imaged by lens **146** onto a photosensor **152**. Photosensor **152** has an image plane parameterized by image coordinates (X_i,Y_i) that have an origin in the upper left corner of photosensor **152**. For purposes of simple coordinate transformation, image plane X_i-Y_i is preferably plane-parallel with the X_b-Y_b plane of body coordinates **112** (X_b,Y_b,Z_b) of phone **104**. Photosensor **152** is a photodetector such as a pixellated array of photodiodes or a CMOS camera capable of detecting radiation **130**.

The propagation of a particular photon bundle **130'** belonging to radiation **130** is shown explicitly in **Fig. 5**. Photon bundle **130'** undergoes refraction at the surface of lens **146**, passes through point-of-view **𝒫** and is imaged onto photosensor **152** at an image point **154**. Note that the location of image point **154** in image plane X_i-Y_i is largely determined by an angle **σ** of propagation of photon bundle **130'** with respect to optical axis **150** (also sometimes referred to as field angle), as well as the location in environment **100** from which photon bundle **130'** has arrived and its wavelength **λ**. In the approximation of ideal pinhole behavior of lens **146**, the imaging of radiation **130** emitted and/or reflected from different locations in environment **100** into cone angle **Σ** yields a perspective projection of the imaged portion of environment **100** on the surface of photosensor **152**. Photosensor **152**, in turn, is connected to image processing electronics **156** which include requisite firmware and software for processing radiation **130** imaged on photosensor **152**.

Due to the generally non-ideal nature of lens **146**, some common distortions and aberrations are inherited in the imaged portion of environment **100**. The removal of such distortions and aberrations (including barrel distortion, pincushion distortion, coma, astigmatism, dispersion, etc.) is well understood by persons skilled in the art. Preferably, image processing electronics **156** are capable of removing such distortions and aberrations prior to image processing for better interface performance.

It is also understood that although lens **146** is visualized as a single part, it may be a refractive lens, a reflective element, a compound lens, a catadioptric lens (refractive and reflective), a graded index lens (GRIN lens), a Fresnel element or any other optical element capable of gathering radiation **130** from field of view **148** and delivering it to photosensor **152** to produce a perspective projection of environment **100**.

The perspective projection of environment **100** needs to include at least one stationary object, here television **126** along with its non-collinear optical inputs, here edges **132A-D** of screen **128** and marking **129**. Images **132A'-D'**, **129'** of these non-collinear optical inputs are used to establish stable frame **134**, parameterized by world coordinates (X_w, Y_w, Z_w) with their origin in the upper left corner of screen **128**. This requirement will dictate in many cases the minimum cone angle **Σ** required for operating the interface over a range of absolute poses of phone **104** acceptable to user **102**.

Figs. 6A-C illustrate three images **158A-C** of environment **100** produced by refraction of radiation **130** as it passes through lens **146** and impinges on photosensor **156**. Images **158A-C** are acquired using three different types of lens **146** from the same absolute pose of phone **104**; namely about 6 feet (≈ 2 m) away from television **126**. All images **158A-C** are centered on the same point on wall **108** to the left of television **126**. As a result, image center **162** for all images **158A-C** is the same.

Image **158A** is obtained when lens **146** has a wide angular field of view **148**. In other words, cone angle **Σ** is large and may be on the order of 50˚ to 60˚ or more. Image **158B** is obtained with a type of lens **146** that has an intermediate angular field of view **148** with a cone angle **Σ** of between 30˚ and 50˚. Finally, image **158C** is produced when a type of lens **146** that has a narrow angular field of view **148** with a cone angle **Σ** of less than 30˚, such as about 20˚ or even less. All images contain imaged objects, including wall image **108'** of wall **108**, table image **208'** of table **208** and television image **126'** of television **126**. Note that all images **158A-C** are circular, since the angular field of view of lens **146** is usually circular.

In image **158A**, image **126'** of the stationary object for defining stable frame **134** — in our case television **126** parameterized by world coordinates (X_w, Y_w, Z_w) with an origin in the upper left corner of screen **128** — is rather small. In other words, image **126'** of television **126**, and in particular screen image **128'** does not subtend a significant angular extent of field of view **148**. This is indicated in the drawing figure by a radius from the field's center. On the other hand, image **158A** includes an image **108'** of a significant portion of wall **108**.

Lens **146** with such large angular field of view **148** is advantageous in situations where phone **104** will assume many different absolute poses within three-dimensional environment **100**. That is because image **126'** of television **126** and image **128'** of screen **128** along with images **132A'-D'** of its edges **132A-D** and image **129'** of marking **129** will remain in field of view **148** even when phone **104** is held at very oblique angles or close to screen **128**.

On the other hand, lens **146** with a large angular field of view **148** is not advantageous when phone **104** will be operated far from screen **128** of television **126**. That is because image **128'** of screen **128** and images **132A'-D'**, **129'** of its non-collinear optical inputs or edges

132A-D and marking **129** will subtend only a small field angle. In other words, screen image **128′** will represent a small portion of total image **158A**. Therefore, a convex hull or convex envelope of the set of all points along edges **132A-D** delimiting screen **128** and points on marking **129** i.e., the area defined by edge images **132A′-D′** and by image **129′** of marking **129** is small. Working with a small convex hull will limit the resolution of the interface. Differently put, it will negatively impact the accuracy in the recovery of absolute pose of phone **104** from the non-collinear optical inputs and thus restrict the performance of the interface of the invention.

Image **158C** shown in **Fig. 6C** is obtained with a lens **146** that has a small angular field of view **148**. Here, television image **126′** as seen by phone **104** from the same distance as image **158A**, subtends a large portion of field of view **148**. Indeed, a portion of television **126**, namely its upper right corner, is not even imaged because it falls outside field of view **148**.

Lens **146** with such a small field of view **148** is advantageous as it provides a large convex hull from images **132A′-D′**, **129′** of edges **132A-D** of screen **128** and its marking **129**. Thus, the accuracy in absolute pose recovery of phone **104** from these non-collinear optical inputs can be very good. Further, it is advantageous to use such type of lens **146** when phone **104** is operated far from screen **128** and is not expected to assume absolute poses at large angles to screen **128**. This ensures that a large portion of or preferably entire image **128′** of screen **128** and image **129′** of marking **129** will always be in field of view **148**.

Fig. 6B shows image **158B** obtained with lens **146** that has an intermediate field of view **148**. As seen by camera **144** of phone **104** from the same distance as images **158A** and **158C**, entire image **128′** of screen **128** and images **132A-D′**, **129′** of its edges **132A-D** and marking **129** are in field of view **148**. This type of lens **146** is preferred for most interfaces according to the invention because it strikes a good

balance between range of operation of phone **104** and accuracy of absolute pose recovery. In particular, it can capture images **128′**, **129′** of screen **128** and marking **129** from many absolute poses of phone **104**, both far and close to television **126**, e.g., between roughly about 15 feet (≈ 5 m) and roughly about 3 feet (≈ 1 m). This is true even when camera **144** is held at an oblique angle to screen **128**, e.g., 45° inclination with respect to the plane of screen **128** or the X_w-Y_w plane defined in world coordinates (X_w,Y_w,Z_w). At the same time, intermediate field of view **148** ensures that a sizeable convex hull or envelope of non-collinear optical inputs defined by images **132A-D′**, **129′** of edges **132A-D** and marking **129** will be present in image **158B**. This fact enables high accuracy absolute pose recovery and hence good interface performance.

For high performance, additional optical requirements on lens **146** and camera **144** should be enforced. These requirements derive from the specific design of the interface and the use cases. It should be noted that in general it is impossible to specify a set of optimal optical requirements to fit all embodiments. Therefore, the below guidelines are provided so that a person skilled in the art will be able to choose the best type of lens **146** and camera **144** based on a balance between operating conditions, performance and cost.

First and foremost, lens **146** needs to capture a sufficient level or intensity of electromagnetic radiation **130**. That is because image **158B** must provide a computable-quality image **128′** of screen **128** with its non-collinear optical inputs or edges **132A-D** and of image **129′** of marking **129**. Specifically, the quality of image **128′** must permit extraction of the imaged non-collinear optical inputs to enable absolute pose recovery of phone **104**. For this reason it is advantageous to choose a small F-number lens **146**, e.g., between about 1.2 and about 2.8, to ensure that even in low-light conditions lens **146** captures the requisite number of photons of radiation **130**.

Second, lens **146** should preferably have a large depth-of-field. In other words, lens **146** should preferably be a quasi-pinhole lens so that objects near and far within angular field of view **148** remain in focus. The main reason is that it is hard to extract features if the image is too defocused. In addition, pinhole behavior is desirable because algorithms for absolute pose recovery of phone **104** are based on image **158B** presenting a perspective projection of environment **100**. In other words, pose recovery algorithms assume that the images can be treated as if they had been taken with a pinhole camera.

Depending on the difference in wavelengths or spectrum of radiation **130** employed, chromatic dispersion could be a problem. A way to address chromatic dispersion, which alters the focal distance with wavelength λ, is to ensure that lens **146** is effectively corrected for chromatic aberration. Alternatively, radiation **130** of a single and well-known wavelength λ can be used to avoid chromatic dispersion issues.

Third, lens **146** and camera **144** should capture images **158B** of three-dimensional environment **100** at an appropriate frame rate and exposure time t_e. The frame rate will depend on the rate of change in absolute pose of phone **104**. The faster phone **104** is being translated and rotated by user **102** during operation, the higher the frame rate or corresponding shutter speed of camera **144** should be set for capturing image **158B**. Additionally, the exposure time t_e during the frame should be sufficiently long to capture enough radiation **130** to generate the best possible image **158B**, but not so long as to incur motion blur. Under operating conditions where rapid changes in absolute pose are expected a short exposure time is a must to avoid motion blur. In fact, there are certain parameters of absolute pose $(x_w, y_w, z_w, \alpha_{wb}, \beta_{wb}, \gamma_{wb})$, e.g., orientation parameters such as angles α_{wb} and γ_{wb}, that can produce massive motion blur even at relatively modest rates of change (e.g., on the order of several degrees per second). Meanwhile, pure translations of phone **104** (e.g., along the three translational degrees of freedom) tend to produce much lower

levels of motion blur. (The reader will realize that this is due to the large linear velocity associated with even a small angular velocity at a large distance away from the center of rotation.)

Fourth, lens **146** should produce minimal levels of optical aberrations and distortions in image **158B**. Optical distortions are deviations from perfect perspective projection of environment **100** to image **158B** on photosensor **152** of camera **144**. As already noted above, such distortions typically include barrel distortion or pincushion distortion. Parallax is a distortion inherent in most wide-angle (fisheye) lenses. It occurs when the chief rays of all object points do not all intersect optical axis **150** of lens **146** at a single point, i.e., at point-of-view ***P***. This can be avoided by using reflective imaging optics incorporating a conic section of revolution as a reflective surface. For details on such optics the reader is referred to U.S. Pats. 7,038,846 and 7,268,956 as well as the references cited therein.

Another kind of lens imperfection is aberration including spherical aberration, coma, astigmatism etc. These aberrations limit the ability of lens **146** to image rays of radiation **130** from a point object in environment **100** to a perfect point in image **158B**. Although some of these distortions and aberrations can be effectively removed by image processing electronics **156**, it is advantageous that lens **146** be relatively aberration- and distortion-free to reduce the amount of processing dedicated to remediation of these detrimental effects in the image.

Fifth, lens **146** should be small and easy to implement in phone **104**. It should preferably be moldable from typical optical materials, e.g., acrylic or other plastic, and it should be manufacturable in large quantities. That means that it should not involve difficult to mold surfaces, such as highly curved surfaces or surfaces having unusual lens prescriptions.

After selecting appropriate lens **146** based on the above guidelines and any further requirements specific to the application and interface, it is important to address any residual imaging problems. **Fig. 7A** illustrates image **160A'** (corresponding to full circular image **158B**) of three-dimensional environment **100** obtained with lens **146** having an intermediate angular field of view **148** as displayed on display screen **136** of phone **104**. Image **160A'** is captured at time t_1 when phone **104** is held in the first absolute pose by user **102** in his/her right hand **102'** as shown in **Fig. 1A**. Meanwhile, **Fig. 7B** illustrates image **160B'** of three-dimensional environment **100** also obtained with lens **146** having an intermediate angular field of view **148**, but taken at time t_5 when phone **104** is held in the second absolute pose by user **102** in his/her left hand **102''** as shown in **Fig. 1B**.

Images **160A'** and **160B'** as seen on screen **136** are rectangular rather than circular. This is unlike images **158A-C** shown in **Figs. 6A-C** that capture the entire angular field of view **148** of lens **146**. The reason is that in practice the entire image circle may not always be captured by camera **144**. Most photosensors such as pixellated photosensor **152** of camera **144** are rectangular or square. Thus, one option is for the image circle of image **160A'** to be inscribed within the rectangular pixel array and underfill photosensor **152** to ensure capture of the entire angular field of view **148** afforded by lens **146**. In this case, many peripheral pixels that lie in the corners of photosensor **152** are never used (no radiation **130** will be delivered to them through lens **146**). Alternatively, the image circle of images **160A'** can circumscribe or overfill by extending beyond the rectangular array of pixels of photosensor **152**. Thus a portion of images **160A', 160B'** near the periphery of the angular field of view **148** will "fall off" photosensor **152** and not be registered by camera **144**. In the present embodiment, images **160A', 160B'** underfill photosensor **152**. The entire image circle afforded by lens **146** is thus captured by photosensor **152** and digitized.

Referring now to **Fig. 7A**, we examine the perspective projection of three-dimensional environment **100** in two-dimensional image **160A'**. It is well known that perspective projections obey certain fundamental geometrical theorems on vanishing points, horizon lines, single and multiple-point perspectives, surface normals and the famous Desargues' theorem of projective geometry. In the present case, lines corresponding to extensions of the edges of wall image **108'** converge to two vanishing points **164**, **166**. More precisely, the perspective projection in image **160A'** exhibits two vanishing points **164**, **166** both lying on a horizon line **168** and yielding a two-point perspective view of environment **100**. Extensions of edge images **132A'-D'** (since television **126** was oriented to be plane-parallel with wall **108** and its edges **132A-D** are thus parallel to the edges of wall **108**) also converge to the same vanishing points **164**, **166** residing on horizon line **168** drawn in a dashed and dotted line.

Of course, vanishing points and horizon lines are mathematical constructs and not tangible parts of image **160A'**. Furthermore, for most absolute poses assumed by phone **104** in the hands of user **102** vanishing points **164**, **166** will fall well outside image **160A'** projected on photosensor **152** and displayed on screen **136**. Indeed, this is the case here as well, with only a small section of horizon line **168** actually intersecting image **160A'**.

In order for the interface to recover the absolute pose of phone **104** accurately it is crucial that the perspective projection of environment **100** in image **160A'** be very accurate. Indeed, it is well known in the fields of computer vision, robotic vision and navigation that very good imaging quality must be achieved if algorithms for camera pose recovery are to accurately report absolute pose parameters, such as $(x_w, y_w, z_w, \alpha_{wb}, \beta_{wb}, \gamma_{wb})$ or any other typical absolute pose parameters employed to parameterize the six degrees of freedom available to phone **104**. In view of the above, image deviations have to be cured to the extent possible.

Referring now to **Fig. 7B**, we see image **160B'** also yields a perspective projection of environment **100**. This time, image **160B'** corresponds to environment **100** as witnessed by camera **144** from point-of-view $\mathcal{P}$ (see **Fig. 5**) in the second absolute pose of phone **104**, when held by user **102** in left hand **102''** at time t_5 as shown in **Fig. 1B**. Here, extensions of the lines corresponding to edges of wall image **108'** and extensions of images of edges **132A'-D'** of television screen image **128'** converge to vanishing points **180**, **182** on a horizon line **184** on the other side of screen **136**.

Now, in an enlarged section **186** of image **160B'** we see that a portion of image **108'** of wall **108**, namely image **108A'** of the edge of wall **108**, shows a significant deviation **108A''** from a straight line. For purposes of better visualization, deviation **108A''** is greatly exaggerated in **Fig. 7B**. Deviation **108A''** increases as a function of distance from image center **162**. In other words, deviation **108A''** is a radial function and is just due to distortions caused by lens **146**. A person skilled in the art should realize that imperfections in lens **146**, overall misalignments between point of view $\mathcal{P}$ and center of photosensor **152**, imperfect plane alignment between image plane X_i-Y_i of lens **146** and the actual plane of photosensor's **152** surface, as well as various other mechanical tolerances may introduce significant distortions that can not be accounted for with a purely radial function associated with lens **146**. Those imperfections may introduce significant errors, since parallel lines will not necessarily intersect in a unique vanishing point due to lens imperfections. As a result, we deviate from the assumption of a perfect perspective projection and introduce errors in the calculated pose. These issues are well understood in the art and will not be reiterated herein. The reader is referred to resources such as the textbook by Warren J. Smith, "Modern Optical Engineering", SPIE Press, The McGraw-Hill Companies (ISBN 978-0-07-147687-4).

When the only significant deviation **108A''** is a radial function of lens **146**, undistortion of image **160B'** can be undertaken by a simple

undistortion correction or re-mapping of all points of image **160B'**. **Fig. 8** illustrates a radial distortion curve **186** of lens **146** that is used for such undistortion. A "perfect" curve **188**, depicted for comparison in **Fig. 8**, is a straight line according to which radiation **130** arriving at the refractive surface of lens **146** from environment **100** at different field angles σ is mapped at different image radii $\mathbf{r_i}$ measured from image center **162**. The actual value of radius $\mathbf{r_i}$ is indicated in pixels.

Our radial distortion curve **186** (barrel distortion) however, is not a straight line and its divergence from perfect curve **188** increases with radius $\mathbf{r_i}$. (The opposite distortion in which the divergence decreases with radius is called pincushion distortion.) In practice, distortion curve **186** may be approximated by a polynomial or a higher-order curve to directly assign field angle σ to image radius $\mathbf{r_i}$ or even directly to the corresponding pixel **190** in image plane X_i-Y_i. This may be done to save processing time in certain embodiments with a corresponding look-up table, rather than performing the undistortion calculation each time. In fact, fisheye lenses manufactured for video cameras regularly come with "warping" software for correction of barrel or pincushion distortion.

Fig. 9 is a diagram that shows the surface of photosensor **152** of camera **144** with image radius $\mathbf{r_i}$ indicated from image center **162** to the circular periphery of image **160B'**. Image radius $\mathbf{r_i}$ corresponds to angular field of view **148** and underfills photosensor **152**, as remarked above. Photosensor **152** is a pixellated CMOS sensor with pixels **190**. Note that radius $\mathbf{r_i}$ for curves **186**, **188** graphed in **Fig. 8** is quantified by number of pixels **190** from image center **162** rather than standard metric units. Meanwhile, the origin of the image coordinates (X_i,Y_i) is indicated in the upper left corner of CMOS **152** (see also image parameterization found in **Fig. 5**).

Image **160B'** is a perspective projection. It contains details such as images of non-collinear optical inputs or edges **132A'-D'** and of

marking **129'** as well as edges **108'** of wall **108** and table image **208'**. These are indicated directly on pixels **190** of CMOS **152**. Note that only a fraction of pixels **190** belonging to CMOS **152** is drawn in **Fig. 9** for reasons of clarity. A normal the array of pixels **190** in CMOS **152** will range from 1,000 x 1,000 to several thousands per side, and the pixel array need not be square. The radius $\mathbf{r_i}$ in practical and ideal radial distortion curves **186**, **188** of **Fig. 8** is measured from center **162**. Thus, for example, for a 2,000 by 2,000 array of pixels **190**, image center **162** from which $\mathbf{r}_i$ is measured will fall approximately on the $1{,}000^{th}$ pixel **190** along X_i-axis and on the $1{,}000^{th}$ pixel **190** along Y_i-axis. The reason that this relationship is approximate is due to the various mechanical misalignments, optical aberrations and distortions as well as other tolerances and errors. In fact, the exact location of image center **162** should preferably be ascertained and corrected for in well-known ways when high quality interface performance is desired.

The diagram of **Fig. 9** unveils the main parts of camera **144** and elements of image processing electronics **156**. Specifically, camera **144** has a row multiplexing block **192** for interacting with rows of pixels **190**. It also has a column multiplexing block **194** for interacting with columns of pixels **190**. Blocks **192**, **194** are connected to a demultiplexer **198** for receiving raw image data **196** from pixels **190** ordered in accordance with any multiplexing scheme. Depending on the level of sophistication of camera **144**, blocks **192**, **194** may be capable of collecting raw image data **196** only from designated rows or columns of pixels **190**. In an advanced camera **144**, blocks **192**, **194** may be able to designate regions of interest defined by groups of pixels **190** and only report raw image data **196** from such regions of interest. Note that in some cameras, blocks **192**, **194** are replaced by a single block or still other multiplexing and pixel control electronics.

In the present embodiment, blocks **192**, **194** simply report the image values of pixels **190** from an exposure taken during one frame (shutter

frame). Thus, for the purposes of the interface of invention, raw image data **196** are preferably simple gray scale values expressed in binary as 8-bit integers ranging from 0 to 255. Demultiplexer **198** is configured to receive such 8-bit raw image data **196** from all pixels **190** and to format it for image pre-processing. Such formatting may include, but is not limited to, the removal of latency and time effects due to shuttering conventions (e.g., use of rolling shutter vs. global shutter), enforcement of pixel reporting order and other functions well known in the art of formatting raw image data **196**.

Demultiplexer **198** is connected to image pre-processing unit **200**, which receives formatted raw image data **196**. Pre-processing unit **200** performs dewarping (a.k.a. un-warping), realignment, normalization and smoothing functions. Specifically, pre-processing unit **200** re-aligns image **160B'** based on relative position (distance, offset, tilt, etc.) of photosensor **152** with respect to lens **146**. Preferably, such relative position and its tolerances are determined prior to the use of camera **144** in the interface of the present invention.

Unit **200** also dewarps image **160B'** based on known distortions of lens **146** including re-mapping of the values of pixels **190** in accordance with radial distortion curve **186** from **Fig. 8**. In addition, unit **200** may normalize the values of pixels **190**, remove shot noise and dead pixels, apply smoothing functions and perform any other well known adjustments or image enhancements as necessary.

Unit **200** is connected to image processing unit **202**. Unit **202** receives the corrected image from unit **200** and applies the processing steps necessary to recognize images of the non-collinear features or edges **132A'-D'** belonging to screen **128** and of marking **129'** of television **126**. A person skilled in the art will recognize that numerous image segmentation, contrast enhancement, edge detection and noise reduction techniques are known for performing this task. Some of the best known include: the Sobel edge detector, the Canny edge detector and various versions of the Hough transform in combination

with Gaussian filters. In fact, any known technique can be employed herein based on the type of environment **100**, television **126**, amount of radiation **130** and other standard optics and signal processing considerations known to those skilled in the art.

Typically, unit **202** first applies a differential filter to image **160B'** to enhance edge contrast. In the present case, when television **126** is on and screen **128** is active, the edges of screen **128** present high contrast ratio non-collinear optical inputs (they are therefore relatively easy to find in image **160B'**). Unit **202** segments image **160B'** and applies the selected edge detection algorithm. Depending on the application and as discussed below, unit **202** may also be programmed to detect images of wall edges **108'** in order to ascertain stable coordinates (X_s, Y_s, Z_s) parameterizing stable frame **106** in environment **100**. Edges or other features of wall **108** may be used as the corresponding non-collinear optical inputs.

The output of unit **202** is a complete image description of the rectangle formed by screen image **128'** and either a point or area defined by marking image **129'**. Such complete image description of edges **132A'-D'** and marking **129'** may include line approximations or equations, including line fits, such as a least squares fit. In addition, if required, unit **202** also outputs an image description of the rectangle formed by images of wall edges **108'** and of table **208'**.

A camera pose recovery unit **204** is connected to unit **202** for receiving its output. Unit **204** employs the geometrical description of the non-collinear optical inputs, namely the lines and points of screen image **128'** and image of marking **129'** to recover the absolute pose of camera **144** in accordance with well-understood principles widely employed in computer vision and robotics. Pose recovery is mathematically possible because vanishing points **180** and **182** as well as horizon line **184** and the size of screen image **128'** fully determine the absolute pose of camera **144** based on its point-of-view $\mathcal{P}$. In

practice, robust methods are used to deal with noise and imperfect modeling.

It should be noted that pose is also sometimes referred to as exterior orientation and translation in the fields of computer vision and robotics. In fact, in pose recovery algorithms as may be applied by unit **204** it is common to work with parameters that are different from absolute pose parameters ($x_w, y_w, z_w, \alpha_{wb}, \beta_{wb}, \gamma_{wb}$) used in mechanics to describe the rigid body motion of phone **104**. Of course, any parameterization ultimately describes the six degrees of freedom available to phone **104** (or any rigid body bearing camera **144** whose pose is being recovered) and therefore a mathematical transformation can be used to move between the rigid body description predominantly used in mechanics (and physics) and the computer vision description.

In camera pose recovery unit **204** employing computer vision algorithms, absolute pose of phone **104** is described by means of a rotation and translation transformation that brings the object, in this case television **126** and more specifically its screen **128** and marking **129**, from a reference pose to the observed pose. This rotation transformation can be represented in different ways, e.g., as a rotation matrix or a quaternion. The specific task of determining the pose of screen **128** in image **160B'** (or stereo images or image sequence, as discussed further below) is referred to as pose estimation. The pose estimation problem can be solved in different ways depending on the image sensor configuration and choice of methodology.

A class of pose estimation methodologies involves analytic or geometric methods. Once photosensor **152** is calibrated the mapping from 3D points in the scene or environment **100** and 2D points in image **160B'** is known. Since the geometry and size of the object comprising the non-collinear optical inputs, i.e., screen **128**, is known, this means that the projected screen image **128'** is a well-known function of screen's **128** pose. Thus, it is possible to solve the pose

transformation from a set of equations which relate the 3D coordinates of the points along edges **132A-D** with their 2D image **132A'-D'** coordinates.

Another class of methodologies involves learning based methods. These methods use an artificial learning-based system, which learns the mapping from 2D image features to pose transformation. In short, this means that a sufficiently large set of images **128'**, **129'** of the non-collinear optical inputs produced by edges of screen **128** and marking **129** collected in different poses, i.e., viewed from different absolute poses of phone **104**, must be presented to unit **204** during a learning phase. Once the learning phase is completed, unit **204** will be able to present an estimate of the absolute pose of camera **144** given images **128'**, **129'** of screen **128** and marking **129**.

Yet another class of methodologies involves solving the pose estimation and image calibration simultaneously. In such an approach unit **200** does not dewarp (or un-warp) image **160B'** and instead an algorithm utilizes warped features. For example, an artificial learning-based system is presented with dewarped 2D image features for a large set of known poses. After the learning phase is completed the algorithm is then able to estimate pose from uncalibrated features.

In a vast majority of embodiments of the interface camera pose recovery unit **204** employs the first class of methods, i.e., analytic or geometric methods. That is because they are the most efficient, while keeping the computational burden within the limited computational range of image processing electronics **156**. Specifically, on-board units, in this case camera **144**, that are capable of receiving radiation **130** and processing images do not have sufficient computational resources and on-board power to implement processing-intensive algorithms for camera pose recovery. Thus, the algorithms being run by unit **204** should preferably consume just a small fraction of on-board processing resources.

To implement an efficient analytic or geometric method in unit **204**, it is important to first calibrate camera **144**. Calibration is performed prior to running the algorithm by presenting camera **144** with an image of screen **128'** in a set of canonical positions and providing its physical measurements. Of course, in the present embodiment, television **126** may communicate with phone **104** wirelessly and provide the necessary information about its screen **128** to phone **104** and more specifically to unit **204** upon inquiry. For requisite teachings on camera calibration the reader is referred to the textbook entitled "Multiple View Geometry in Computer Vision" (Second Edition) by R. Hartley and Andrew Zisserman. Another useful reference is provided by Robert Haralick, "Using Perspective Transformations in Scene Analysis", Computer Graphics and Image Processing 13, pp. 191-221 (1980). For still further information the reader is referred to Carlo Tomasi and John Zhang, "How to Rotate a Camera", Computer Science Department Publication, Stanford University and Berthold K.P. Horn, "Tsai's Camera Calibration Method Revisited".

Now, as already remarked, camera pose recovery unit **204** receives output of unit **202** in the form of a complete image description of the rectangle formed by the non-collinear optical inputs in the form of edges **132A'-D'** and marking **129'**. In addition, unit **202** also provides a complete image description of additional non-collinear optical inputs, such as the rectangle formed by images of wall edges **108'** and preferably of table **208'**. With this additional data, camera **144** can be calibrated with respect to both stable coordinates (X_s, Y_s, Z_s) parameterizing stable frame **106** of environment **100** as well as world coordinates (X_w, Y_w, Z_w) defined by television **126**.

In the most general case, unit **204** may use points from the complete description of images **132A'-D'**, **129'** as well as **108'**, **208'** for determining the absolute pose of camera **144** with an iterative closest point algorithm or any other suitable algorithm. Preferably, unit **204** estimates absolute pose of camera **144** in stable coordinates

(X_s,Y_s,Z_s) with respect to stable coordinate origin in the upper left area of environment **100**, and in world coordinates (X_w,Y_w,Z_w) with respect to world coordinate origin in the upper left corner of screen **128**.

Since in the present embodiment most non-collinear optical inputs are line-like, unit **204** preferably implements much faster algorithms than iterative closest point. For example, it employs a type of algorithm generally referred to in the art as pose estimation through comparison. In this approach a database of screen images **128'** obtained at different rotations and translations is compared to the complete image description provided by unit **202**. For efficiency reasons, such comparison preferably employs a homography. A homography is an invertible transformation from the real projective plane on the surface of photosensor **152** to a projective plane in a canonical position of camera **144** that maps straight lines to straight lines. Because straight lines are preserved under this type of operation, the transformation is also frequently called a collineation, a projective transformation or even projectivity by those skilled in the art. The reader is again referred to the textbook entitled "Multiple View Geometry in Computer Vision" (Second Edition) by R. Hartley and Andrew Zisserman.

When working with images of rectangles **128'** and **108'** there exists symmetry between certain absolute poses of camera **144**. Therefore, additional information from image **160B'** is necessary to break this symmetry. Differently put, additional non-collinear optical input is required to unambiguously define up, down, left and right. In the present embodiment, optical information from any point of table image **208'** can be used to break the symmetry for determining the absolute pose in stable coordinates (X_s,Y_s,Z_s). Similarly, optical information from any point of marking image **129'** can be used to break the symmetry for determining the absolute pose in world coordinates (X_w,Y_w,Z_w) that parameterize absolute reference frame **134** for the purposes of the interface.

Image processing electronics **156** have an output module **206** that is connected to camera pose recovery unit **204**. Module **206** receives information about the absolute pose of phone **104** computed by the pose recovery algorithm deployed by unit **204**. Specifically, it receives pose information in the format used by the camera pose recovery algorithm of computer vision. This description may contain descriptors such as angles with respect to surface normals — for example, the normal to the surface of screen **128** or the normal to the surface of wall **108**. Such descriptions are intrinsically expressed in world coordinates (X_w, Y_w, Z_w) that parameterize world frame **134** and in stable coordinates (X_s, Y_s, Z_s) that parameterize stable frame **106** in environment **100**. However, these descriptions may not be expressed in absolute pose parameters $(x_s, y_s, z_s, \alpha_{sb}, \beta_{sb}, \gamma_{sb})$ and $(x_w, y_w, z_w, \alpha_{wb}, \beta_{wb}, \gamma_{wb})$ that were previously used to parameterize the absolute pose of phone **104** in the language of mechanics of rigid body motion. Therefore, module **206** may need to translate the output of unit **204** to mechanical absolute pose parameters $(x_s, y_s, z_s, \alpha_{sb}, \beta_{sb}, \gamma_{sb})$ and $(x_w, y_w, z_w, \alpha_{wb}, \beta_{wb}, \gamma_{wb})$. Of course, some embodiments of the interface may be designed to work directly with the computer vision description from the point-of-view $\mathcal{P}$ of camera **144** and no such translation is necessary.

Any computer vision algorithm deployed by camera pose recovery unit **204** will inherently determine the absolute pose of phone **104** from the point-of-view $\mathcal{P}$ of camera **144** (see **Fig. 5**). Since in the present embodiment absolute pose is reported with respect to center of mass C.O.M. **110** that coincides with the origin of body coordinates (X_b, Y_b, Z_b) of phone **104**, module **206** needs to translate the absolute pose output of unit **204** from the point-of-view $\mathcal{P}$ of camera **144** into body coordinates (X_b, Y_b, Z_b) of phone **104**. This translation is accomplished by a coordinate transformation involving the addition of the fixed offset vector $\mathbf{o}_b$ (see **Fig. 5**) to the absolute pose output of unit **204**. Thus, output module **206** translates the output of unit **204** into mechanical absolute pose parameters $(x_w, y_w, z_w, \alpha_{wb}, \beta_{wb}, \gamma_{wb})$ and

$(x_s, y_s, z_s, \alpha_{sb}, \beta_{sb}, \gamma_{sb})$. Further, module **206** generates a signal **210** related to one or more of the recovered absolute pose parameters of phone **104**.

Signal **210** is related to one or more of the absolute pose parameters in any suitable manner. In the simplest case, signal **210** contains all absolute pose parameters expressed in both stable coordinates (X_s, Y_s, Z_s) and in world coordinates (X_w, Y_w, Z_w). In other embodiments, signal **210** contains a subset of the absolute pose parameters, e.g., just the absolute position or just the absolute orientation. Still other embodiments need signal **210** that only contains two absolute position parameters expressed in stable coordinates (X_s, Y_s, Z_s), such as (x_s, y_s), or in world coordinates (X_w, Y_w, Z_w), such as (x_w, y_w). Signal **210** may also be related to just one absolute pose parameter, e.g., the absolute distance from screen **128** along the z-direction expressed in stable coordinates (X_s, Y_s, Z_s) or in world coordinates (X_w, Y_w, Z_w). Still other applications may require signal **210** to provide one or more orientation angles, such as γ_{sb} or γ_{wb} in applications where the roll (twist) of phone **104** is important. Further, signal **210** may be related to the absolute pose parameter or parameters in linear and non-linear ways or in accordance with any function including scaling, transposition, subspace projection, reflection, rotation, quantization or other function applied to any one or to all of the absolute pose parameters contained in signal **210**. For example, signal **210** may contain derivatives, including first- and higher-order derivatives, integrals or re-scaled values of any of the absolute pose parameters or any linear combination thereof. Additionally, signal **210** may be related to the absolute pose parameter or parameters either in its amplitude, its frequency or its phase.

Fig. 10 illustrates in more detail the elements of an advantageous embodiment of an interface **212** according to the invention. Interface **212** is deployed in environment **100**, where human user **102** manipulates phone **104** to assume various absolute poses as introduced in **Figs. 1A-B**. We initially concentrate on a first absolute pose assumed by

phone **104** along trajectory **114** at a time t_0 before time t_1 illustrated in **Fig. 1A**. In this absolute pose at time t_0 all three angles $(\alpha_{wb},\beta_{wb},\gamma_{wb})$ describing the absolute orientation of phone **104** in world coordinates (X_w,Y_w,Z_w) happen to be equal to zero. This means that fully rotated body coordinates (X_b,Y_b,Z_b) are aligned with the triple primed body coordinates (X_b''',Y_b''',Z_b''') and with world coordinates (X_w,Y_w,Z_w) (see rotation convention in **Figs. 3A-D**). Meanwhile, at time t_0 the absolute position of phone **104** as described in world coordinates (X_w,Y_w,Z_w) by vector $\mathbf{r}_w(t_0)$ is not equal to zero. (Stable coordinates (X_s,Y_s,Z_s) and corresponding vector $\mathbf{r}_s(t_0)$ from their origin to C.O.M. **110** are not shown in **Fig. 10** for reasons of clarity — refer to **Fig. 2** where vector $\mathbf{r}_s$ is drawn explicitly.)

This absolute pose of phone **104** at time t_0 with no rotations as expressed in world coordinates (X_w,Y_w,Z_w) corresponds to absolute pose parameters as follows: $A.P.=(x_w,y_w,z_w,\alpha_{wb},\beta_{wb},\gamma_{wb})=(x_w,y_w,z_w,0,0,0)$. Although phone **104** does not ever need to assume such absolute pose for enabling the operation of interface **212**, it is nevertheless shown for pedagogical reasons. In this way, the reader can gain a more intuitive idea about when along trajectory **114** the phone's **104** pose does not involve any rotations.

At time t_0 camera **144** employs its image processing electronics **156** in the manner described above. As a result, phone **104** generates signal **210** related to at least one of its recovered absolute pose parameters $(x_w,y_w,z_w,\alpha_{wb},\beta_{wb},\gamma_{wb})$ expressed in world coordinates (X_w,Y_w,Z_w). As mentioned above, phone **104** can also determine and report recovered absolute pose parameters $(x_s,y_s,z_s,\alpha_{sb},\beta_{sb},\gamma_{sb})$ expressed in stable coordinates (X_s,Y_s,Z_s). In other words, absolute pose parameters $(x_w,y_w,z_w,\alpha_{wb},\beta_{wb},\gamma_{wb})$ and $(x_s,y_s,z_s,\alpha_{sb},\beta_{sb},\gamma_{sb})$ can be computed at time t_0 and used to construct related signal **210**. In the present embodiment of interface **212**, signal **210** is directly proportional to all six recovered absolute pose parameters $(x_w,y_w,z_w,\alpha_{wb},\beta_{wb},\gamma_{wb})$ in world coordinates (X_w,Y_w,Z_w). However, it is not related to, and more

strictly does not contain any of the absolute pose parameters $(x_s, y_s, z_s, \alpha_{sb}, \beta_{sb}, \gamma_{sb})$ as reported in stable coordinates (X_s, Y_s, Z_s). Signal **210** is thus proportional to the values of absolute pose parameters $(x_w, y_w, z_w, \alpha_{wb}, \beta_{wb}, \gamma_{wb}) = (x_w, y_w, z_w, 0, 0, 0)$ expressed in units corresponding one-to-one to real 3D space of environment **100** in the absolute frame of reference **134** parameterized by world coordinates (X_w, Y_w, Z_w).

Interface **212** takes advantage of communication link **214** of phone **104** to communicate signal **210** to an application **216** running on a host unit **218**. It is the objective of interface **212** to derive or produce input to application **216** based on the absolute pose of phone **104**. More precisely, application **216** is designed to employ signal **210** as an input of interface **212**. In the present embodiment, link **214** is the down-link of the phone's **104** Bluetooth wireless link. It will be appreciated by persons skilled in the art that any suitable link, wireless or wired, may be used to transmit signal **210** to application **216**.

Host unit **218** belongs to television **126** and is incorporated into its base **220**. Host unit **218** has a processor and other typical resources to implement application **216** and to drive screen **218**. In fact, it should be noted, that if on-board image processing electronics **156** cannot properly handle the camera pose recovery, this task could be assigned to host unit **218**, as it will typically have a stable power supply and considerable computing resources.

In the embodiment shown, application **216** is a home shopping application that displays to user **102** products **222**, **224**, **226** that can be purchased with the aid of interface **212**. Of course, products **222**, **224**, **226** may include any merchandise available from any commercial source or database, e.g., a web-based database or a home shopping network that application **216** can access via the Internet. Here, product **222** is a necktie, product **224** is a motorcycle helmet and product **226** is a bag. Application **216** displays necktie **222**, helmet

224 and bag **226** on screen **128** in a way that makes it easy for user **102** to make his or her selection.

In fact, in the present embodiment of interface **212**, signal **210** constitutes the complete input from user **102** to application **216**. The absolute pose of phone **104** supplied to application **216** is used to move a cursor **228** on screen **128** and to thus allow user **102** to select among products **222, 224, 226.** A person skilled in the art will recognize this functionality as absolute 3D pointing capability and/or as an absolute 3D mouse. In fact, cursor **228** can be employed in conjunction with depressing an agreed upon button (see below), to endow it with other capabilities such as scrolling or otherwise bringing up a larger selection of products.

We now examine the operation of interface **212** by referring to **Fig. 10** and to a more detailed view of trajectory **114** of phone **104** and corresponding images **230A-G** of screen **128** captured by camera **144** during operation, as shown in **Fig. 11.** We will also refer to the flow diagram of **Fig. 12** that illustrates the steps executed by application **216** and image processing electronics **156** on-board phone **104** during the operation of interface **212.**

Interface **212** is initialized when user **102** presses a predetermined button **142** or performs any suitable initialization action or sequence of actions. In the example shown in **Fig. 10,** interface **212** is initialized at time t_0 by depressing a button **142A** on phone **104.** Of course, it is understood that phone **104** does not need to be initialized in interface **212** while its orientation angles are zero.

Fig. 11 depicts image **230A** projected onto photosensor **152** of camera **144** at the time of initialization, t_0, of interface **212.** Additionally, time t_0 coincides with the start of trajectory **114** of phone **104.**

The steps performed by interface **212** at initialization are found in the flow diagram of **Fig. 12.** Initialization signal is used in step **232** to start interface **212** by activating camera **144** and image processing electronics **156.** In subsequent step **234,** camera **144** is instructed to capture image **230A** of environment **100.** Image **230A** may optionally be displayed to user **102** on screen **136** of phone **104.**

It is important in step **234** that image **230A** be captured at a sufficiently short exposure time, t_e, to ensure that it contains no appreciable motion blur. For example, exposure time t_e in situations where user **102** is expected to move phone **104** relatively slowly may be set on the order of 100 ms to 25 ms (1/10 to 1/40 sec). On the other hand, exposure time t_e should be significantly shorter, e.g., 10 ms or even less (1/100 sec and faster) in situations where person **102** is expected to move phone **104** relatively quickly. In the event a rolling shutter is employed, the exposure time t_e should be adjusted accordingly to ensure no significant time delay between the capture time of radiation **130** by first and last pixels **190.**

A person skilled in the art will recognize that the F/# of lens **146** must be sufficiently low and the ISO sensitivity of photosensor **152** must be set sufficiently high to enable camera **144** to capture image **230A** under the ambient illumination conditions and given the amount of radiation **130** emitted by screen **128.** Specifically, image **230A** has to be of sufficient quality to permit recognition of images **132A'-D'**, **129'** of edges **132A-D** and of marking **129** that are chosen as non-collinear optical inputs to image processing unit **204** and camera pose recovery module **206.**

In next step **236,** raw image data **196** is demultiplexed and formatted. It is then forwarded in step **238** for pre-processing of image **230A** by image pre-processing unit **200.** After step **238,** if possible, a determination should be made at step **240** whether image **230A** is of sufficient quality to warrant further processing, or if another image should be captured. For example, if image **230A** is of insufficient

quality to support image processing and feature extraction, i.e., line detection to find images of edges **132A'-D'** and image of marking **129'** that represent the non-collinear optical inputs, then another image should be captured at an adjusted exposure time t_e and ISO setting. The corresponding adjustments are made in step **242** in accordance with well-known principles of optics. In fact, if it is possible to make the image quality determination sooner, e.g., at step **236**, then an instruction to proceed to step **242** should be issued by interface **212** after that step.

A sufficiently high quality image **230A** is forwarded to step **244**, in which dewarped or corrected image **230A** is processed by image processing unit **202**. Step **244** involves filtering, image segmentation, contrast enhancement and extraction of images of non-collinear optical inputs in this embodiment the images of edges **132A'-D'** and image of marking **129'**. As indicated above, the requisite techniques are well-known in the art of robotic and computer vision. In general, feature extraction reduces the complexity of pose estimation by using a reduced representation of environment **100** instead of the raw image as input to a pose recovery algorithm. Edges, corners, blobs, ridges, intensity gradients, optical flow, etc. are all well-known image features familiar to a skilled artisan. Alternatively, more sophisticated features include Scale-Invariant Feature Transform (SIFT) by David Lowe or Speeded Up Robust Features (SURF) by Herbert Bay et al.

There is a significant amount of additional literature about the extraction of the invariant and non-collinear optical inputs from the images (a.k.a. feature extraction). Extraction of such features will require the application of suitable image segmentation modules, contrast thresholds, line detection algorithms (e.g., Hough transformations) and many others. For more information on edge detection in images and edge detection algorithms the reader is referred to U.S. Patents 6,023,291 and 6,408,109 and to Simon Baker and Shree K. Nayar, "Global Measures of Coherence for Edge Detector

Evaluation", Conference on Computer Vision and Pattern Recognition, June 1999, Vol. 2, pp. 373-379 and J. Canny, "A Computational Approach to Edge Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, Nov. 1986 for basic edge detection. Additional useful teachings can be found in U.S. Patent 7,203,384 to Carl and U.S. Patent 7,023,536 to Zhang et al. A person skilled in the art will find all the required modules in standard image processing libraries such as OpenCV (Open Source Computer Vision), a library of programming functions for real time computer vision. For more information on OpenCV the reader is referred to G. R. Bradski and A. Kaehler, "Learning OpenCV: Computer Vision with the OpenCV Library", O'Reilly, 2008.

Extracted non-collinear optical inputs **132A'-D'**, **129'** are supplied to camera pose recovery unit **204** in step **246**. Unit **204** applies the rules of perspective geometry in an analytic or geometric algorithm to solve the inverse problem of finding the collineation that maps the extracted non-collinear optical inputs, i.e., **132A'-D'** and **129'**, to what a reference or canonical position will produce (reference inputs **132A'-D'**, **129'** as seen from the reference or canonical pose). Due to the effects of noise, feature mismatch, imperfect calibration and/or incomplete modeling of environment **100**, the set of extracted features or inputs can never be mapped exactly into the reference set. In preferred embodiments, a robust method finds the collineation that minimizes the sum of algebraic errors between the set of extracted features and the reference set. The reader is invited to review K. Kanatani, "Geometric Computation for Machine Vision", pp. 153-155 for more details. For a simpler but less robust approach the reader is referred to Robert M. Haralick, "Determining Camera Parameters from the Perspective Projection of a Rectangle", Journal of Pattern Recognition, Vol. 22, Issue 3, 1989.

In step **248** the output of camera pose recovery unit **204** is provided to output module **206**. In one embodiment, the output of pose recovery unit **204** is the collineation computed in step **246**. This collineation

is converted to absolute pose parameters either by unit **204** or by output module **206**. Absolute pose parameters can be expressed in many different formats. In the present embodiment, they are expressed as a translation vector $\mathbf{r}_w$ and rotations $(\alpha_{wb},\beta_{wb},\gamma_{wb})$ in world coordinates (X_w,Y_w,Z_w) that parameterize frame **134**. In another embodiment, they are expressed as a reference vector $\mathbf{r}_{rw}$ and a surface normal **n** in body coordinates (X_b,Y_b,Z_b). In yet another embodiment, the orientation, regardless of reference, is expressed using a quaternion representation. Output module **206** uses the output to generate signal **210** that is related to at least one of the absolute pose parameters irrespective of how they are expressed or parameterized (i.e., absolute pose parameters of mechanics, computer vision or still some other convention).

Now, unit **204** expresses the absolute pose of phone **104** in reference to the point-of-view $\mathcal{P}$ of camera **144** (as defined by lens **146**) rather than center of mass C.O.M. **110** of phone **104**. That is because the geometric algorithm in step **246** operates on image **230A** as seen from point-of-view $\mathcal{P}$. Therefore, output module **206** must also convert the absolute pose of phone **104**. Such conversion to body coordinates **112** centered on C.O.M. **110** of phone **104** is accomplished once again by a coordinate transformation that adds offset vector $\mathbf{o}_b$. (In a more complicated case than that shown in **Fig. 5**, when image plane X_i-Y_i is not plane parallel with respect to plane X_b-Y_b of body coordinates **112**, a rotation matrix will also have to be applied as a part of the coordinate transformation. Coordinate transformation methods are known to those skilled in the art and the diligent reader is again referred to G.B. Arfken, *op. cit.*)

In addition to the coordinate transformation, interface **212** requires that the at least one absolute pose parameter of phone **104** be expressed or reported by unit **206** in a stable frame. In the present embodiment two choices of such stable frames for reporting the one or more pose parameters of phone **104** are available.

The first stable frame is defined by stable coordinates (X_s,Y_s,Z_s) that parameterize frame **106** in environment **100**. Stable coordinates (X_s,Y_s,Z_s) do not move as they are defined by wall **108** and other stationary objects that produce optically discoverable non-collinear optical inputs. Thus, frame **106** can be taken as the stable frame that defines environment **100** in the context of the surface of a very stable (and reliable) reference in the surroundings.

The second stable frame **134** is defined by world coordinates (X_w,Y_w,Z_w) (or workspace coordinates) in the Cartesian convention with respect to the upper left corner of screen **128**. In the present embodiment, frame **134** is usually stationary within first stable frame **106** because television **126** does not move. In other words, a coordinate transformation between stable coordinates (X_s,Y_s,Z_s) and world coordinates (X_w,Y_w,Z_w) is constant in time. This transformation is conveniently expressed by constant vector $\mathbf{d}_S$ (see **Fig. 2**) and a constant rotation matrix $\mathbf{R}_{SW}$ (not shown) if the axes of stable coordinates (X_s,Y_s,Z_s) parameterizing frame **106** and the axes of world coordinates (X_w,Y_w,Z_w) defining frame **134** are not aligned (they are not aligned in the present embodiment and hence a rotation matrix must be used).

Since in the present embodiment object **126** is a large television designed to stay in the same place in environment **100**, interface **212** employs frame **134** parameterized by world coordinates (X_w,Y_w,Z_w) as the stable frame. In other words, because television **126** is at rest in stable coordinates (X_s,Y_s,Z_s) defining stable frame **106** on the surface of the Earth, stable frame **134** parameterized by world coordinates (X_w,Y_w,Z_w) is also at rest as long as television **126** does not move. Therefore, unless application **216** needs to know and keep confirming where television **126** is located in stable frame **106**, interface **212** may dispense with recovering the absolute pose of phone **104** in stable coordinates (X_s,Y_s,Z_s) altogether. Thus, the non-collinear optical inputs of wall **108** (i.e., its edges and corners) and of table **208** do not need to be used for camera pose recovery with respect to these

objects to track phone **104** in stable coordinates (X_s,Y_s,Z_s). Note, however, that in embodiments where the object that user **102** is interacting with is not stationary in absolute reference frame **106**, it may be necessary to keep track of the item's **104** absolute pose in both stable coordinates (X_s,Y_s,Z_s) and in world coordinates (X_w,Y_w,Z_w) to achieve proper operation of interface **212**.

As remarked above, in the present embodiment signal **210** is related to all six absolute pose parameters $(x_w,y_w,z_w,\alpha_{wb},\beta_{wb},\gamma_{wb})$ by being directly proportional to them. In general, however, the relation between signal **210** and the one or more absolute pose parameters chosen to parameterize the six degrees of freedom of phone **104** is much more broadly defined. Signal **210** needs only be related to one absolute pose parameter of phone **104** as expressed in stable frame **134** (or stable frame **106**). Furthermore, signal **210** may be encoded in frequency, amplitude or phase.

The one or more absolute pose parameters to which signal **210** is related, e.g., by being directly proportional to them as in this case, need not directly correspond to one of the six absolute pose parameters defined by $(x_w,y_w,z_w,\alpha_{wb},\beta_{wb},\gamma_{wb})$. Instead, the one or more absolute pose parameters to which signal **210** is related needs only in turn be related by a mapping to at least one of the six degrees of freedom of phone **104** that may be parameterized in any manner (e.g., by mechanics conventions, computer vision conventions or still other conventions). The present case is the simplest, since the mapping is a one-to-one mapping of all six absolute pose parameters $(x_w,y_w,z_w,\alpha_{wb},\beta_{wb},\gamma_{wb})$, to which signal **210** is directly proportional, to the six degrees of freedom parameterized with these same absolute pose parameters, namely $(x_w,y_w,z_w,\alpha_{wb},\beta_{wb},\gamma_{wb})$. More complex mappings that are not one-to-one and involve scaling as well will be examined in subsequent embodiments.

The one-to-one mapping of all absolute pose parameters $(x_w,y_w,z_w,\alpha_{wb},\beta_{wb},\gamma_{wb})$ of phone **104** in the same convention as that used

to parameterize the six degrees of freedom of phone **104** in world coordinates (X_w,Y_w,Z_w) makes interface **212** a special type of interface. Interfaces where the one or more absolute pose parameters to which signal **210** is related map to all the translational and rotational degrees of freedom are referred to herein as fully parameterized interfaces. Under this definition, interface **212** is fully parameterized, since signal **210** contains a full parameterization of the absolute pose of phone **104**.

In the next step indicated in **Fig. 12**, on-board communication link **214** sends signal **210** to host unit **218** in the form of a Bluethooth formatted RF signal. In step **250**, signal **210** is received by a host-side receiving unit and forwarded as input to application **216**. It is noted that although Bluetooth is used in this embodiment, any other RF protocol, as well as IR or sonic (e.g., ultrasonic) link or other point-to-point connection (including a wired connection) may be used by interface **212** to transmit signal **210** in the corresponding format and code from phone **104** to host **218**.

Depending on the rate of motion of phone **104** and type of trajectory **114** that interface **212** is expected to support, it is important that the overall time duration between the capture of image **230A** and transmission of signal **210** to host unit **218** be kept relatively short. For example, the time required for completing steps **234** through **248** in flow diagram of **Fig. 12** should be kept at 10-20 msec. The time delay required for transmission to host **218** and reception as input to application **216** should also be kept as short as possible, and ideally at less than 10 msec. The reason for such rapid processing and transmission in interface **212** has to do with the human perception of delay. According to accepted standards and IEEE specifications human user interfaces should ideally produce a delay of less than 30 msec in order to be perceived as real-time by user **102**.

Application **216** receives signal **210** with full parameterization of phone **104** as an input of interface **212**. Specifically, the values

contained in signal **210** are employed as input of user **102** by application **216**.

To interpret the absolute pose of phone **104** application **216** uses a set of application coordinates (X_a, Y_a, Z_a) to parameterize its digital three-dimensional environment **252**. These application coordinates (X_a, Y_a, Z_a) with their origin in the lower right corner of screen **128** are shown in **Fig. 10**. Of course, the choice of origin and orientation of coordinates (X_a, Y_a, Z_a) is merely exemplary and can be selected by the interface designer as convenient or as dictated by application **216**. In the present embodiment, television **126** supports 3D viewing and thus having a three-dimensional coordinate system (X_a, Y_a, Z_a) makes sense. In 2D televisions the Z_a-axis may not be necessary.

Application **216** uses absolute pose parameters $(x_w, y_w, z_w, \alpha_{wb}, \beta_{wb}, \gamma_{wb})$ of phone **104** delivered by signal **210** as input of user **102**. In the present embodiment, it translates these pose parameters into application coordinates (X_a, Y_a, Z_a) to define the pose of phone **104** in digital three-dimensional environment **252**. Environment **252** is thus a cyberspace or a virtual space that is like real space.

At this stage, application **216** is capable of using its driver **254** of display screen **128** to display to user **102** a virtual phone **104'** in an absolute pose that mirrors the absolute pose of phone **104** in world coordinates (X_w, Y_w, Z_w). Such virtual phone **104'** can be displayed in digital three-dimensional environment **252** parameterized by application coordinates (X_a, Y_a, Z_a) and can be particularly advantageous when using a three-dimensional type of television **126**. Note that when environment **252** of application **216** is a cyberspace, a virtual space or a portion of a mixed space where the standard rules of 3D geometry apply, the ability to obtain full absolute pose of phone **104** is crucial to life-like interactions.

In the present simple home shopping application **216**, however, interface **212** is designed to only assists in computing the intersection of optical axis **150** of lens **146** with display **128**. Application **216** then instructs cursor control **256** to draw a place-holder entity, in this case a feedback cursor **228** at that intersection to provide visual feedback to user **102**.

At time t_0, we see from **Figs. 10** and **11** that optical axis **150** does not intersect with display **128**. This is further evidenced by the fact that in image **230A** taken at time t_0 image center **162**, which always lies along optical axis **150** of lens **146**, is not on image **128'** of screen **128**. Therefore, cursor control **256** does not draw feedback cursor **228** on screen **128**.

Instead, application **216** instructs screen driver **254** to keep products **222**, **224**, **226** displayed on screen **128**. Meanwhile, while user **102** is not pointing at screen **128**, application **216** may perform support, cross-check and other functions. For example, application **216** may cross-check with a database **258** of merchandise that products **222**, **224**, **226** are properly displayed. Application **216** may additionally verify with remote resources **260** that may include the Internet as well as proprietary resources and links that products **222**, **224**, **226** are still in stock and available for sale to user **102**. In performing these functions, application **216** may take advantage of data in signal **210**. For example, it may terminate them when cursor **228** is getting close to screen **128**.

Application **216** has a feedback module **262** that can send feedback to phone **104** for the benefit of user **102**. Application **216** can provide feedback to user **102** in any form supported by on-board resources **264** of phone **104**. Advantageously, the feedback is sent by an up-link **214'** of the Bluetooth wireless link employed to transmit signal **210** to host **218**.

For example, in the present embodiment feedback is in the form of audio information that is communicated to user **102** via on-board resources **264** that include speakers **138** (see **Fig. 4A**). Specifically, application **216** uses speakers **138** to send the following audio information to user **102** at time t_0 in response to the recovered absolute pose of phone **104**: "You are pointing off-screen. Please indicate the product you want to find out about by pointing at it".

At time t_1, interface **212** once again repeats steps **234** through **248** (see flow diagram of **Fig. 12**) to recover the absolute pose of phone **104** in world coordinates (X_w,Y_w,Z_w) and send it to application **216**. The time elapsed between time t_0 and t_1, also sometimes expressed in terms of frame rate by those skilled in the art, may either be dynamically controlled by application **216** or it may be pre-set.

When user **102** moves phone **104** rapidly and interface **212** requires accurate absolute pose information so that application **216** runs properly, the time between time t_0 and t_1 should be kept short. Put another way, a high frame rate is required to accurately capture absolute pose of phone **104** when user **102** is moving phone **104** quickly. In fact, images of screen **128'** may need to be captured and processed without significant latency at frame rates approaching 100 Hz or even 200 Hz in such situations. Note that a correspondingly short exposure time t_e needs to be chosen at such high frame rates to permit sufficient time between capturing radiation **130** for each frame.

On the other hand, much slower frame rates, e.g., on the order of 10 Hz, may be sufficient when user **102** is not moving phone **104** quickly. To optimize the on-board resources of phone **104** and to not overload its processors, it is thus preferable to dynamically adjust the frame rate according to the motion of phone **104**. When phone **104** moves slowly, a frame rate of near 10 Hz is selected, while at extremely fast speeds a frame rate in excess of 100 Hz is chosen.

In the present case, frame rate is initially set to 10 Hz at time t_0. Therefore, the time elapsed between t_0 and t_1 is $1/10^{th}$ of a second. At time t_1 the absolute pose of phone **104** is significantly different than it was at time t_0. Indeed, the absolute pose at time t_1 corresponds to user **102** holding phone **104** in his/her right hand **102'** as shown in **Fig. 1A**. It is clear from comparing image **230B** obtained by on-board camera **144** at time t_1 to image **230A** obtained at time t_0, that optical axis **150** now does intersect the surface of screen **128**, as center of image **162** is within edges **132A-D** of screen **128**. As a result, application **216** instructs cursor control **256** to draw feedback cursor **228** at the intersection of optical axis **150** and the surface of screen **128**, so that user **102** can see where he/she is pointing phone **104**.

In addition, application **216** instructs feedback module **262** at time t_1 to generate and send additional feedback to user **102**. This time the feedback is in the form of tactile or haptic information communicated to on-board resources **264** of phone **104** by communication link **214'**. The haptic information is a fast vibration of phone **104** achieved by deploying its on-board vibrator resource (not shown).

In the manner described above, camera **144** of interface **212** captures successive images **230C-G** at times t_2, t_3, t_4, t_5 and t_6. Note that at time t_5 user **102** is holding phone **104** in left hand **102''** as previously shown in **Fig. 1B**. At times t_3 and t_4 optical axis **150** once again does not intersect screen **128**. Hence, application **216** again uses speakers **138** to send the following audio information to user **102** at times t_3 and t_4 in response to the absolute pose of phone **104**: "You are pointing off-screen. Please indicate the product you want to find out about by pointing at it".

At time t_5 user **102** has manipulated phone **104** into an absolute pose in which optical axis **150** intersect screen **128** at the location of product **226**. In response, application **216** instructs cursor control **256** to draw feedback cursor **228** on top of displayed product **226**. In

addition, application **216** generates visual feedback via feedback module **262** and sends it to on-board resources **264** of phone **104**. The visual feedback is displayed on screen **136** of phone **104** and communicates the attributes of product **226** to user **102**. For example, the attributes include information such as: price, size, material, make, satisfaction, quality report, special features, number of facebook friends who have purchased same product, most recent tweet about product, etc. Of course, the feedback may be supplied in audio format and use speakers **138** to communicate the same information to user **102**.

User **102** may depress a corresponding selection button **142B** at time t_5, while pointing at product **226** as shown in **Fig. 10**, to communicate to application **216** that he/she wishes to purchase product **226**. Application **216** generates the corresponding signal indicating user's **102** purchase decision **266** and submits it for processing in any manner known to those skilled in the art of on-line sales. Preferably, purchase decision **266** is made by a single depression of selection button **142B**, thus making it a "one-click" transaction. In fact, any compatible "one-click" shopping technique can be applied in interface **212**. Feedback including visual and/or audio information congratulating user **102** on his/her purchase may be issued by application **216** via feedback module **262**.

At time t_6, user **102**, having purchased product **226**, moves far away from screen **128**. This is apparent from image **230G** in **Fig. 11**, where the size of screen image **128'** subtends only a small fraction of field-of-view **148**. At this time, user **102** presses still another selection button **142** (not specifically indicated herein) to instruct interface **212** to issue a stop command **268** to application **216**. Stop command **268** terminates application **216**, turns off camera **144** and places interface **212** on stand-by or turns it off completely.

At this point, we understand a basic embodiment of interface **212** and its method of operation. However, in the implementation described so

far, interface **212** did not take full advantage of the six degrees of freedom of phone **104**. Those are the absolute pose parameters $(x_w, y_w, z_w, \alpha_{wb}, \beta_{wb}, \gamma_{wb})$ in Cartesian world coordinates (X_w, Y_w, Z_w) that were set up to parameterize frame **134** of environment **100**. Moreover, all data pertaining to absolute pose of phone **104** in stable coordinates (X_s, Y_s, Z_s), i.e., absolute pose parameters $(x_s, y_s, z_s, \alpha_{sb}, \beta_{sb}, \gamma_{sb})$ were even discarded.

To take advantage of the full set of absolute pose parameters that interface **212** can recover at successive times, it is necessary to gain a still deeper appreciation of trajectory **114** of phone **104** and conventions used to describe it. **Fig. 13** illustrates in more detail phone **104** and trajectory **114** traversed between time t_0 and t_5 by its center of mass C.O.M. **110**. In the convention used herein, the motion of the rigid body of interest, namely of phone **104**, consists of translations and rotations.

In particular, the position and orientation of phone **104**, i.e., its absolute pose A.P. at any time $t>t_0$ is completely characterized by the position of its C.O.M. **110** and by the rotation matrix $\mathbf{R} \in SO(3)$ (special orthogonal matrix in 3D) that describes the rotational state of any point in its body coordinates (X_b, Y_b, Z_b) in the stable frame of our choice. As already noted above, we have two frame choices, namely frame **106** described by stable coordinates (X_s, Y_s, Z_s) and frame **134** described by world coordinates (X_w, Y_w, Z_w). The corresponding equations in frames **106** and **134** respectively are:

$$A.P._s(t) = \mathbf{R}_s(t)\mathbf{o}_b + \mathbf{r}_s(t) \qquad (1A)$$

$$A.P._w(t) = \mathbf{R}_w(t)\mathbf{o}_b + \mathbf{r}_w(t) \qquad (1B)$$

In these equations, we are using the notation conventions introduced above and in which A.P.(t) denotes time-dependent absolute pose of phone **104**. Uppercase bold letters denote matrices, lowercase boldface letters denote vectors and subscripts refer to the reference frames in which the quantities are expressed. We have chosen to

demonstrate the effect of the rotation matrix **R** on vector $\mathbf{o}_b$ expressing the offset from C.O.M. **110** to point-of-view ***P*** of camera **144** residing on-board phone **104**. The reason for this choice is because point-of-view ***P*** is a point of special interest on phone **104** as it is the vantage point from which the pose recovery algorithms recover camera pose.

The operation of a matrix on a vector produces another vector. We use two subscripts to denote the result. Therefore, in Eq. 1A the result of applying rotation matrix $\mathbf{R}_s(t)$ to vector $\mathbf{o}_b$ is vector $\mathbf{o}_{sb}$. We thus know that the resulting vector is expressed after rotation from the vantage point of stable coordinates (X_s, Y_s, Z_s). Similar logic applies to Eq. 1B that yields $\mathbf{o}_{wb}$.

In general, rotation matrix $\mathbf{R}_s$ incorporates all three rotations $(\alpha_{sb}, \beta_{sb}, \gamma_{sb})$ previously introduced in **Figs. 3A-D**. The individual rotations can be expressed by the components of rotation matrix $\mathbf{R}_s$ around the body axes Z_b, Y_b and X_b starting with the body axes being aligned with the axes of the frame being used, in this case axes X_s, Y_s and Z_s of frame **106**, as follows:

$$R_{zb}(\alpha_{sb}) = \begin{pmatrix} \cos\alpha_{sb} & -\sin\alpha_{sb} & 0 \\ \sin\alpha_{sb} & \cos\alpha_{sb} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{Eq. 2A}$$

$$R_{xb}(\beta_{sb}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\beta_{sb} & -\sin\beta_{sb} \\ 0 & \sin\beta_{sb} & \cos\beta_{sb} \end{pmatrix} \quad \text{Eq. 2B}$$

$$R_{yb}(\gamma_{sb}) = \begin{pmatrix} \cos\gamma_{sb} & 0 & \sin\gamma_{sb} \\ 0 & 1 & 0 \\ -\sin\gamma_{sb} & 0 & \cos\gamma_{sb} \end{pmatrix} \quad \text{Eq. 2C}$$

The complete rotation matrix $\mathbf{R}_s$ is obtained by simply multiplying the above individual rotation matrices in the order of the convention. In other words, $\mathbf{R}_s = \mathbf{R}_{yb}(\gamma_{sb})\mathbf{R}_{xb}(\beta_{sb})\mathbf{R}_{zb}(\alpha_{sb})$. It should be noted that

rotation matrices are always square and have real-valued elements. Algebraically, a rotation matrix in 3-dimensions is a 3x3 special orthogonal matrix (SO(3)) whose determinant is 1 and whose transpose is equal to its inverse:

$$\mathbf{R}^T=\mathbf{R}^{-1};\ \det(\mathbf{R})=1, \qquad \text{Eq. 3}$$

where superscript T indicates the transpose matrix and superscript -1 indicates the inverse matrix.

Having properly defined rotation matrix $\mathbf{R}_s$ and its behavior, we return to **Fig. 13**. Note first that the same rotation convention may be used to define the relative orientation of world coordinates (X_w,Y_w,Z_w) to stable coordinates (X_s,Y_s,Z_s) by a rotation matrix. Also, we can define the rotation matrix $\mathbf{R}_w$ that describes rotation angles $(\alpha_{wb},\beta_{wb},\gamma_{wb})$ from the point of view of world coordinates (X_w,Y_w,Z_w) of frame **134**. To recover trajectory **114** between any two absolute poses in stable coordinates (X_s,Y_s,Z_s) we thus need to use Eq. 1A with rotation matrix $\mathbf{R}_s$ as described above. To recover trajectory **114** between any two absolute poses in world coordinates (X_w,Y_w,Z_w) we use Eq. 1B with rotation matrix $\mathbf{R}_w$.

In the present embodiment, trajectory **114** between $A.P._s(t_i)$ and $A.P._s(t_j)$ is recovered by having camera pose recovery algorithm in step **246** executed by unit **204** first account for rotations and then for translations. In other words, the algorithm first recovers the absolute orientation of phone **104** as expressed by matrix $\mathbf{R}_s(t_i)$ in terms of the three rotation angles or the computer vision mathematical equivalent. The algorithm then computes the translation vector $\mathbf{r}_s(t_i)$. The same approach is taken in computing $A.P._w(t_i)$ and $A.P._w(t_j)$ to describe trajectory **114** in world coordinates. Alternatively, vector $\mathbf{d}_s$ and the rotation matrix between the two coordinate systems $\mathbf{R}_{sw}$ is used to calculate $A.P._w(t_i)$ and $A.P._w(t_j)$ from $A.P._s(t_i)$ and $A.P._s(t_j)$.

It is important to note that other conventions are possible. These will also recover trajectory **114** in stable and world coordinates. However, the exact description will differ. Therefore, once a trajectory convention is selected for interface **212** it is best to enforce it throughout.

Unit **204** of interface **212** provides complete absolute pose descriptions $A.P._w(t_0)$, … $A.P._w(t_i)$, $A.P._w(t_j)$, …, $A.P._w(t_5)$ at the corresponding times t_0, …, t_i, t_j, …, t_5 in signal **210**, which is proportional to all of the six degrees of freedom described by the absolute poses. Thus, application **216** has sufficient information to recover trajectory **114** of C.O.M. **110** of phone **104** along with the orientation of phone **104** at those times. In the present embodiment, application **216** uses the pose information just to draw feedback cursor **228** at the location where optical axis **150** of optic **146** happens to intersect the plane of screen **128** at the time of pose measurement.

With the aid of offset vector $\mathbf{o}_b$ between C.O.M. **110** and point-of-view $\mathcal{P}$ of camera **144**, **Fig. 13** illustrates trajectory **114** of C.O.M. **110** alongside trajectory **114'** of point-of-view $\mathcal{P}$. During the time between t_0 and t_5 vector $\mathbf{o}_b$ executes a complex motion about C.O.M. **110** due to changes in the absolute pose of phone **104**. At time t_0 point-of-view $\mathcal{P}$ is to the right of C.O.M. **110** and optical axis **150** extending from point-of-view $\mathcal{P}$ does not even intersect screen **128**. As phone **104** is moved by user **102**, optical axis **150** finally intersects screen **128** at point x_{a1} expressed in application coordinates (X_a, Y_a, Z_a).

Once optical axis **150** intersects screen **128**, application **126** draws feedback cursor **228** at that intersection point. In addition, application **126** draws a trajectory **270** on screen **128** to indicate the 2D trace traversed by cursor **228**. Trajectory **270** is 2D since it is a projection from 3D space of environment **100** into 2D space of screen

128. Since screen **128** is co-planar with plane X_a-Y_a of application coordinates (X_a,Y_a,Z_a) that parameterize digital three-dimensional environment **252**, trajectory **270** is expressed by coordinates (x_a,y_a) (or, more strictly $(x_a,y_a,0)$, since $z_a=0$ in plane X_a-Y_a).

Of course, trajectory **270** has to be interpolated between the times at which the sequence of absolute poses of phone **104** is recovered by unit **204**. The higher the frame rate of camera **144** the more absolute poses can be recovered between time t_0 and t_5. Correspondingly, more intersection points between screen **128** and optical axis **150** can be computed to thus improve the interpolation of trajectory **270**. As mentioned above, for rapid motion a frame rate in excess of 100 Hz is desirable.

At times when optical axis **150** is not intersecting screen **128**, application **216** does not generate optical user feedback on screen **128**. In other words, cursor **228** is absent at those times. This, however, does not mean that information derived from the absolute poses of phone **104** is not useful during those periods. For example, as seen in **Fig. 13**, between points y_{a2} and y_{a3} optical axis **150** is once again off screen **128**. If application **216** could draw cursor **228**, it would be a phantom cursor **228'** located along trajectory **270** extending onto wall **108**.

Of course, in the present configuration application **216** cannot draw outside its own screen **128** on wall **108**. However, since application **216** still knows where phone **104** is being pointed (unit **204** keeps providing it with the full absolute pose information in signal **210**), it may indicate to user **102** how to move phone **104** to bring cursor **228** back on screen **128**. In some embodiments, the location of phantom cursor **228'** could be displayed to user **102** with information that it is off-screen or its distance from screen **128** could be indicated by an audio feedback. Application **216** can in fact do much more with the absolute pose information of phone **104**. That is because, in accordance with the invention, signal **210** is proportional to all six

degrees of freedom parameterized by A.P.(t)=(x_w, y_w, z_w, α_{wb}, β_{wb}, γ_{wb}) in this embodiment.

To understand the capability of fully parameterized interface **212**, we refer now to **Fig. 14**. This isometric diagram illustrates how C.O.M. **110** trajectory **114** and point-of-view $\mathcal{P}$ trajectory **114'** are represented internally by application **216** in its application coordinates (X_a,Y_a,Z_a). Note that in this case, application **216** sets the origin of its application coordinates (X_a,Y_a,Z_a) in the lower left back corner of the volume corresponding to digital three-dimensional environment **252**. Of course, if screen **128** were a volumetric 3D display, application **216** could display trajectories **114**, **114'** to user **102** in a one-to-one or in a scaled format (e.g., 1:4). Indeed, even a non-3D display can be used to represent 3D information with appropriate calibration known to those skilled in video arts (e.g., illustrating trajectories **114**, **114'** in a perspective view).

Application **216** receives a succession of absolute poses of phone **104** from signal **210**. For the sake of simplicity, **Fig. 14** only shows the successive positions of C.O.M. **110** and of point-of-view $\mathcal{P}$ along with offset vector $\mathbf{o}_b$, rather than showing the entire phone **104** it its successive absolute poses. The absolute poses of phone **104** are measured at a constant frame rate. Therefore, the successive positions of C.O.M. **110** and point-of-view $\mathcal{P}$ are spaced equally in time. However, explicit reference to time has been dropped in this drawing figure for the sake of clarity. Furthermore, trajectories **114** and **114'** drawn in real three-dimensional environment **100** correspond to those actually executed by phone **104** due to manipulation by user **102** (actual quantities rather than measured and interpolated data).

The values of pose parameters in signal **210** in the present embodiment are mapped one-to-one to all six degrees of freedom of phone **104**.

Thus, application **216** receives signal **210** containing data about the six degrees of freedom at equal time intervals set by the frame rate.

Since it is difficult to show the orientation portion of absolute pose, we will use a different way to help visualize this information. To do this, we show how the absolute pose A.P. information contained in signal **210** is used. To do this, we pick two points namely the positions of C.O.M. **110** and point-of-view $\mathcal{P}$ at equal time intervals in world coordinates. These two points define between them a vector $\mathbf{o}_{bw}$, which is related to the original offset vector $\mathbf{o}_b$ that is fixed in body frame **112**. Vector $\mathbf{o}_{bw}$ is obtained by transforming offset vector $\mathbf{o}_b$ with the aid of vector $\mathbf{r}_w$ and rotation angles $\alpha_{wb},\beta_{wb},\gamma_{wb}$. In other words, the absolute pose A.P.(t)=$(x_w,y_w,z_w,\alpha_{wb},\beta_{wb},\gamma_{wb})$ recovered and supplied in signal **210** at the corresponding time is used to compute vector $\mathbf{o}_{bw}$ from vector $\mathbf{o}_b$.

In addition, to showing vector $\mathbf{o}_{bw}$, we indicate the direction of optical axis **150**. As shown, application **216** keeps track of it with a corresponding vector **275** extending from point **274** that represents point-of-view $\mathcal{P}$ in application coordinates (X_a,Y_a,Z_a). For visualization purposes it is only the direction of optical axis **150** that we are interested in, rather than the magnitude of the vector representing optical axis **150** to the point at which it intersects screen **128**. In this way we can simplify our example. To accomplish this, we introduce a unit vector $\hat{\mathbf{u}}_w$ along optical axis **150**.

A unit vector is defined to be a vector whose length is 1 (unit length) and is commonly denoted by a "hat". Differently put, a unit vector is a normalized vector that is particularly useful in defining a direction in space without carrying with it information about the magnitude along that direction. The method for extracting directional information from any vector $\mathbf{v}$ and converting it to a unit vector $\hat{v}$ is given by the following equation:

$$\hat{v} = \frac{v}{\|v\|}, \qquad \text{Eq. 4}$$

where $\|v\|$ is the norm or length of the vector (usually computed by employing the Pythagorean Theorem). In fact, when working in any basis, such as our Cartesian stable, world, body and application coordinates, introduced thus far, we may use the unit vector representation of that basis to more efficiently indicate directions. In particular, the convention for defining a Cartesian coordinate system by unit vectors is usually as follows:

$$\hat{i} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; \quad \hat{j} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}; \quad \hat{k} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \qquad \text{Eq. 5}$$

The use of unit vectors is extensive in coordinate transformations, such as those explained in supporting literature.

The absolute pose information carried in signal **210** is thus visualized by how vector $\mathbf{o}_b$ transforms into vector $\mathbf{o}_{bw}$. All six degrees of freedom are employed in this transformation, which is therefore indicative of the complete absolute pose information recovered by interface **212** and contained in signal **210**.

Application **216** shows vector $\mathbf{o}_{bw}$ in its own application coordinates (X_a,Y_a,Z_a), where this vector is mapped to vector $\mathbf{o}_{ba}$. In the same vein, unit vector $\hat{\mathbf{u}}_w$ is mapped to unit vector $\hat{\mathbf{u}}_a$ in application coordinates along optical axis **275**. The nature of the mapping employed is discussed below.

Note that in the present embodiment, application **216** keeps track of the absolute pose of phone **104** even outside of its three-dimensional digital environment **252**. This is reminiscent of the previous case, when application **216** could use only the portion of plane X_a-Y_a that corresponded to screen **128** as its two-dimensional digital environment

252. In that case, application **216** could only draw feedback cursor **218** at the intersection of optical axis **150** of phone **104** with its screen **128**. Nevertheless, application **216** knew where optical axis **150** intersected plane X_a-Y_a and could provide other kinds of feedback (e.g., audio, tactile/haptic, etc.) to user **102**.

Similarly, in the present case, application **216** keeps track of the absolute pose of phone **104** even when that absolute pose is not within the volume of its three-dimensional digital environment **252**. The limiting factor here is the ability of camera **144** to recover the absolute pose of phone **104**. If camera **144** can no longer see a sufficient number of non-collinear optical inputs (here edges **132A-D** and marking **129**), then the absolute pose of phone **104** cannot be recovered unambiguously. This is usually because camera **144** is too far away or turned at too steep an angle for edges **132A-D** and marking **129** to be within its field-of-view **148**. In other cases, camera **144** might not see a sufficient number of non-collinear optical inputs to recover absolute unambiguously pose due to occlusions and other causes interfering with line-of-sight.

The above bring us to an important aspect of the present invention pertaining to the subject of mapping. What the present application addresses is how to map between the one or more pose parameters contained in signal **210** and the six degrees of freedom. In simple cases, including the present embodiment, the parameterization used to define the six degrees of freedom is the same as the convention in which the one or more, and in this case all six, pose parameters are parameterized and reported in signal **210**. This shared parameterization makes it easier to explain the mapping and the associated issues.

It is important to stress, however, that the parameterizations of the six degrees of freedom and the description of the one or more pose parameters to which signal **210** is related do not need to be the same. For example, the rigid body motion of phone **104** could be

parameterized with Cartesian coordinates (or even cylindrical or spherical coordinates) and Euler angles that employ body coordinates (also sometimes referred to as object coordinates), while the camera pose recovery could use robotic vision parameterization such as surface normal **n** to screen **128** (which is collinear with the world coordinate axis Z_w in the present embodiment; see **Fig. 14**) and a quaternion to report the one or more pose parameters in signal **210**. A person skilled in the art will realize that since all descriptions share the same geometrical basis of rigid body motion in 3D space, they are mathematically equivalent. Of course, a skilled artisan will also realize that the best choice of parameterization is made based on environment **100**, application **216**, interface **212** and other factors.

We return now to **Fig. 14**, to discuss the issues of mapping of absolute pose parameters $(x_w, y_w, z_w, \alpha_{wb}, \beta_{wb}, \gamma_{wb})$ contained in signal **210** to the six degrees of freedom of phone **104** parameterized in the same manner. As already noted, the absolute pose parameters to which signal **210** is directly proportional are reported at regular time intervals (set by the frame rate of camera **144**). They are visualized with the aid of points **272**, **274**, or the transformation of offset vector $\mathbf{o}_b$ to vector $\mathbf{o}_{bw}$ and its mapping to vector $\mathbf{o}_{ba}$, and the mapping of unit vector $\hat{\mathbf{u}}_w$ to unit vector $\hat{\mathbf{u}}_a$. Application **216** thus has available to it all the data required to plot points **272**, **274** and unit vector $\hat{\mathbf{u}}_a$ in its virtual or digital 3D environment **252** in a one-to-one relationship to real 3D environment **100**.

In most cases, however, digital 3D environment **252** is either bigger or smaller than real 3D environment **100** in which phone **104** resides. In other words, the direct mapping of real 3D environment **100** to digital 3D environment **252** is rarely 1:1. Thus, re-plotting actual positions of points **272**, **274** and unit vector $\hat{\mathbf{u}}_a$ in a one-to-one mapping is usually not feasible. Therefore, it is convenient for the

mapping to comprise a scaling in at least one among the three translational and the three rotational degrees of freedom.

In the case of interface **212**, all three translational degrees of freedom are scaled 1:4 (note that **Fig. 14** is not showing the actual 1:4 scaling exactly for reasons of clarity). In other words, the values of (x_w, y_w, z_w) (or, equivalently, vector $\mathbf{r}_w$) are scaled 1:4 in the mapping so that the corresponding application values (x_a, y_a, z_a) expressed in application coordinates (X_a, Y_a, Z_a) are just one fourth of the values of (x_w, y_w, z_w). Also note that since the origins and orientations of world coordinates (X_w, Y_w, Z_w) and application coordinates (X_a, Y_a, Z_a) are not the same, a corresponding coordinate transformation has to be applied between them to correctly translate between (x_w, y_w, z_w) and (x_a, y_a, z_a).

Interface **212** does not use a mapping that scales or in some other way alters any of the three rotational degrees of freedom. That is because application **216** is designed to work with vector $\mathbf{o}_{ba}$ (between points **272**, **274**) that corresponds to transformed and scaled but otherwise undistorted offset vector $\mathbf{o}_{bw}$ between C.O.M. **110** and point-of-view $\mathcal{P}$. If offset vector $\mathbf{o}_{ba}$ were distorted due to scaling in any of the rotational degrees of freedom, than the rotations executed by user **102** in real 3D space of environment **100** would not correspond to those recovered in application **216**. For example, a full twist or rotation by 360° (2π) in angle α_{wb}, β_{wb} or γ_{wb} would not correspond to a complete twist or rotation by the corresponding angle in application coordinates (X_a, Y_a, Z_a) as interpreted in application **216**. Of course, in some cases scaling of one or more of the three rotational degrees of freedom may be useful.

Based on signal **210** periodically reporting the full absolute pose as visualized by vector $\mathbf{o}_{bw}$ and unit vector $\hat{\mathbf{u}}_w$, application **216** recovers corresponding vector $\mathbf{o}_{ba}$ between points **272**, **274** and unit vector $\hat{\mathbf{u}}_a$ in its digital 3D environment **252**. The result is a time series of

vectors $\mathbf{o}_{ba}$ that define points along recovered trajectories **278, 278′** and a series of unit vectors $\hat{\mathbf{u}}_a$. Trajectories **278, 278′** correspond to actual trajectories **114, 114′** to the extent that application **216** is able to interpolate between the successive values of vector $\mathbf{o}_{bw}$ in world coordinates (X_w, Y_w, Z_w). A person skilled in the art will recognize that simple interpolation between successive vectors $\mathbf{o}_{bw}$ can be performed naively, i.e., by simple curve fitting. This may be practicable when the frame rate is high, e.g., on the order of 100 Hz or higher. However, at lower frame rates interpolation quality can be improved by additional analysis of the data from signal **210**.

A temporal series or a time sequence of pose data containing the six degrees of freedom can be further processed to derive other quantities. These quantities may include, for example, first- and higher-order time derivatives of the translational and rotational degrees of freedom. Therefore, given a sufficient number of vectors $\mathbf{o}_{bw}$, application **216** can start computing reliable values of first and second order time derivatives of linear displacements (i.e., $\frac{dx_w}{dt}, \frac{dy_w}{dt}, \frac{dz_w}{dt}$ and $\frac{d^2x_w}{dt^2}, \frac{d^2y_w}{dt^2}, \frac{d^2z_w}{dt^2}$). These quantities can be used to construct vectors that describe the linear velocities and accelerations of C.O.M. **110**, denoted by $\mathbf{v}_{C.O.M.}(t)$ and point-of-view $\mathcal{P}$, denoted by $\mathbf{v}_{C.O.M.}(t)$, $\mathbf{a}_{C.O.M.}(t)$ and $\mathbf{v}_{\mathcal{P}}(t)$, $\mathbf{a}_{\mathcal{P}}(t)$, respectively.

The same procedure can be applied to the rotational degrees of freedom to find angular velocities (commonly denoted by ω_q with subscript "q" indicating the axis around which the rotation is taking place) and angular accelerations ($\frac{d\omega}{dt}$). A person skilled in the art will appreciate that when dealing with angular quantities, the axes around which the angular velocities and accelerations are computed need to be properly defined just as in the case of the rotation convention. For example, to keep matters simple the rotations can be defined along body coordinate axes (X_b, Y_b, Z_b) of phone **104**. With that

choice, the angular quantities can be: $\omega_{zb},\omega_{yb},\omega_{xb}$ and $\frac{d\omega_{zb}}{dt},\frac{d\omega_{yb}}{dt},\frac{d\omega_{xb}}{dt}$. It should, be understood that the rotations do not need to be defined in the same convention as the 3D rotation convention of phone **104** in body coordinates (X_b,Y_b,Z_b).

Once the linear and angular velocities and accelerations are computed, application **216** can employ them in any useful manner. For example, the values of these derived quantities may be used as additional input in application **216** including gesture input, control input or just plain data input. Also, in some embodiments, application **216** can suggest the most appropriate frame rate for camera **144** based on linear velocities and accelerations as well as angular velocities and accelerations to avoid motion blur and/or to improve accuracy and performance of interface **212**.

Even with scaling, portions of recovered trajectories **278, 278'** are still outside digital 3D environment **252**. Location **280** shows where trajectories **278, 278'** enter into digital environment **252**. Location **282** shows where trajectories **278, 278'** again leave environment **252**. Therefore, when environment **252** coincides with the volume in which visual display to user **102** can be generated, the portions of recovered trajectories **278, 278'** outside digital environment **252** cannot be visualized to user **102**. However, other feedback, including visual, audio, tactile/haptic, etc. may still be provided to user **102** as a function of trajectories **278, 278'** lying outside digital environment **252**. Another alternative in non-linear scaling (e.g., logarithmic) to effectively compress virtual trajectories to stay bounded with the physical dimensions of the display.

In the present embodiment, the three translational degrees of freedom available to phone **104** are conveniently parameterized by Cartesian coordinate axes X_w, Y_w and Z_w. Of these, two translational degrees of freedom, namely those parameterized by X_w and Y_w axes define a plane in environment **100**. This plane is plane-parallel, and indeed co-

planar with display screen **128**. The reason this is advantageous is that user motion in any plane that is plane-parallel with screen **128** is easily translated to motion in the plane of screen **128**. Hence this motion can be used directly to drive corresponding user feedback, such as generating motion of cursor **228**, producing a trace (e.g., digital ink) or drawing some other place-holder indicating the position of C.O.M. **110** of phone **104** in application coordinates (X_a, Y_a, Z_a).

Fig. 15 illustrates the above point with a further mapping by projection from digital 3D environment **252** into 2D subspaces. In the case shown, the 2D subspace is a plane X_a-Y_a defined in application coordinates (X_a, Y_a, Z_a). Plane X_a-Y_a is plane parallel to plane X_w-Y_w and thus to screen **128**. In this projection, information about Z_a recovered trajectories **278**, **278'** in the Z_w axis (corresponding to Z_a axis) is discarded. Incidentally, so it the Z_w axis component of unit vector $\hat{\mathbf{u}}_w$ (corresponding to Z_a-axis of unit vector $\hat{\mathbf{u}}_a$). The projected 2D trajectories **278A**, **278A'** and the 2D points **272A**, **274A** corresponding to projections of 3D points **272**, **274** are very useful in certain embodiments of application **216**. Specifically, for actions in which only information in the plane of screen **128** is required as input, 2D trajectories **278A**, **278A'** and 2D points **272A**, **274A** offer all the required information to generate user input.

A similar approach can be taken to obtain user input information from projections of 3D trajectories into 2D planes X_w-Z_w, Y_w-Z_w corresponding to planes X_a-Z_a, Y_a-Z_a in application coordinates (X_a, Y_a, Z_a). It is instructive to note that the 2D projections still contain a lot of information about the absolute pose of phone **104**. Indeed, even projections of certain degrees of freedom into 1D subspaces, i.e., their projections onto lines, may be sufficient to provide meaningful input data for application **216**.

Now, a mapping exists between the one or more absolute pose parameters to which signal **210** is related and at least one of the six degrees of freedom of phone **104**. Given the above examples of 3D to 2D projections we are ready to gain a better appreciation for the kinds of mappings that can be performed in principle, and those that may be particularly useful in a preferred implementation of the interface.

A mapping is a rule or set of rules of correspondence or relation between sets, that associate(s) each element in a set (also called the domain) with a unique element in the same or another set (also called the range). Any type of mapping including many-to-one (e.g., projections into lower-dimensional subspaces) and one-to-many (e.g., duplication of some elements into higher-dimensional subspaces) can be used in the present invention. For the purposes of the present description, we define the first set to contain between one and six degrees of freedom available to phone **104** in 3D environment **100**. We define the second set to contain the one or more absolute pose parameters to which signal **210** is related (e.g., by being directly proportional to them).

In the present embodiment, **Fig. 14** shows a one-to-one and proportional mapping between these two sets. The parameterization is full, and thus all six degrees of freedom in the first set are mapped to the second set. In addition, of the six degrees of freedom in the first set, the three translational degrees of freedom are mapped with a concurrent 1:4 scaling factor to the corresponding three absolute pose parameters (x_w, y_w, z_w) corresponding to these translational degrees of freedom in signal **210**. The rotational degrees of freedom in the first set are mapped one-to-one without scaling to the corresponding three absolute pose parameters $(\alpha_{bw}, \beta_{bw}, \gamma_{bw})$ corresponding to these rotational degrees of freedom in signal **210**.

It is important to realize that the mapping can be applied anywhere in interface **212**. In other words, although in the present embodiment

the mapping is performed on-board phone **104** by image processing electronics **156** during step **248** (see **Fig. 12**), it could also be performed elsewhere. For example, the mapping of the data in signal **210** could be carried out in host **218** either by application **216** or by other processing resources. In fact, the mappings of absolute pose parameters defined in world coordinates by signal **210** through projection into planes X_a-Y_a, X_a-Z_a, Y_a-Z_a in application **216** is also a permissible mapping. Clearly, the concept of mapping is very powerful and useful in generating user input in 3D interfaces.

Referring back to **Fig. 15**, we examine a useful mapping for representing the rotational degrees of freedom of phone **104** in application coordinates (X_a,Y_a,Z_a). In particular, **Fig. 15** shows unit vector $\hat{\mathbf{u}}_w$ fully recovered as vector $\hat{\mathbf{u}}_a$ in digital 3D environment **252** of application **216**. Now, unit vector $\hat{\mathbf{u}}_a$ is mapped by projection along the Z_a axis only, as shown in the lower left block **279**. Of course, such projection is generally not going to preserve the unit norm of a unit vector (unless the dot product of unit vector $\hat{\mathbf{u}}_a$ with the basis vector $\hat{k}$ for the Z_a axis as defined above is equal to one). Therefore, such projection of unit vector $\hat{\mathbf{u}}_a$ is designated by $\mathbf{u}_a$ without the "hat" to indicate that it may vary in magnitude. Furthermore, the projection of unit vector $\hat{\mathbf{u}}_a$ is time-dependent, so that $\mathbf{u}_a=\mathbf{u}_a(t)$. Block **279** illustrates the projection of unit vector $\hat{\mathbf{u}}_a$ at times t_0 and t_1, or $\mathbf{u}_a(t_0)$ and $\mathbf{u}_a(t_1)$. Clearly, the projection of the rotational state of phone **104** about world axis Z_w and thus around axis Z_a changes during that time period. We can ascertain this by looking back at **Fig. 13** or **Fig. 14**. To keep track of the change in time, application **216** computes an angular velocity ω_{Za} of vector $\mathbf{u}_a(t)$ about axis Z_a.

In fact, similar mappings can be applied to break down the rotational degrees of freedom around any one or more axes in world coordinates and application coordinates. In the art, such projections are given different names, including "pan angles", "attitude angles", "tilt

angles" and still other names. Clearly, the mapping of the three rotational degrees of freedom can recover any such angle or combinations thereof for use as input to application **216**. Furthermore, in order to adjust angular sensitivity, the mapping may include scaling of any of the three rotational degrees of freedom.

The above embodiments have been disclosed first, in order to present the foundations necessary for understanding the preferred embodiment shown in **Fig. 16**. Corresponding parts in this preferred implementation of an interface **300** are labeled with the same reference numerals as in prior embodiments for clarity. It will be appreciated by a person skilled in the art, however, that analogous parts or steps can be modified to suit the particular embodiment.

Fig. 16 shows item **104**, which is once again embodied by a phone, in a real three-dimensional environment **302** on the surface of planet Earth **304**. Environment **302** lies in the northern hemisphere and is shown along an expanded view indicated by dotted lines above Earth **304**. Earth **304** is parameterized by Earth coordinates (X_e,Y_e,Z_e) employing the Cartesian coordinate convention. The origin of Earth coordinates (X_e,Y_e,Z_e) is located at the center of mass of the planet and oriented such that rotation of Earth **304** described by angular velocity $\boldsymbol{\omega}_e$ (2π/day or 15˚/hour) is around axis Z_e.

Phone **104** has on-board camera **144** whose point-of-view $\boldsymbol{\mathcal{P}}$ is offset by vector $\mathbf{o}_b$ from its C.O.M. **110**, just as in the previous embodiments. The same body coordinates (X_b,Y_b,Z_b) are employed in describing moving frame **112** of phone **104**. In the present case, stable coordinates (X_s,Y_s,Z_s) of stable frame **106** within which the motion of phone **104** is measured are defined by a room **308**, and their origin is located in upper corner **308'**. It is important to note that as far as radiation **130** used by on-board camera **144** to recover the phone's **104** absolute pose is concerned, stable coordinates (X_s,Y_s,Z_s) parameterizing stable frame **106** in environment **302** and Earth coordinates (X_e,Y_e,Z_e)

parameterizing Earth frame **304** are fixed with respect to each other (barring earthquakes or other natural disasters affecting room **308**).

Interface **300** further includes a stationary object **310** having a screen **312** whose edges **313** embody a set of non-collinear optical inputs detectable via electromagnetic radiation **130**. World coordinates (X_w,Y_w,Z_w) parameterizing world frame **134**, or more precisely a gaming space in the present embodiment, are located in the upper left corner of screen **312**. Axes X_w and Y_w of gaming space **134** define plane X_w-Y_w that is co-planar with screen **312**.

In contrast to previous embodiments in which the stationary object, namely television **126**, did not move in stable frame **106**, in the present embodiment object **310** may move from time to time, or even frequently. That is because object **310** is a small game console. Thus, displacement vector $\mathbf{d}_s$ from stable coordinates (X_s,Y_s,Z_s) parameterizing frame **106** to world coordinates (X_w,Y_w,Z_w) defined by game console **310** is shown with an explicit dependence on time; $\mathbf{d}_s=\mathbf{d}_s(t)$. Additionally, note that rotation matrix $\mathbf{R}_{sw}$ for performing the 3D rotation that needs to be executed along with the addition of displacement vector $\mathbf{d}_s(t)$ to complete the coordinate transformation between stable frame **106** and gaming frame **134** is also time dependent in this embodiment; $\mathbf{R}_{sw}= \mathbf{R}_{sw}(t)$.

Game console **310** has a selection unit or touch control **314** that is used for operating it. Touch control **314** is also used for as a feature for breaking the symmetry of screen **312** for unambiguous pose recovery. Console **310** may have additional controls as well as mechanisms (not shown) for placing it in an appropriate location in room **308**.

In addition to camera **144**, phone **104** is equipped with a relative motion sensor **316** offset from C.O.M, **110** by an offset vector $\mathbf{i}_b$. Relative motion sensor **316** has the capability to produce data

indicative of a change in at least one among the six degrees of freedom of phone **104**. In fact, in the present case, sensor **316** is a compound inertial sensor including gyroscopes and accelerometers. These devices are well-known in the art. They can sense rotations about, and translations along, three orthogonal axes X_i, Y_i and Z_i that define inertial sensor coordinates (X_i,Y_i,Z_i) in an inertial sensor frame **318** that is attached to phone **104**. The rotations that are sensed by the gyroscopes of motion sensor **316** are explicitly indicated by angular velocities ω_{xi}, ω_{yi} and ω_{zi}.

Inertial devices such as MEMS accelerometers and solid state gyroscopes do not interact with real 3D environment **302** by detecting radiation **130**. Instead, solid state gyroscopes are sensitive to rotational speeds and accelerometers are sensitive to acceleration and gravity effects.

More precisely, the accelerometers sense Earth **304** due to its gravity along a vector $\mathbf{e}_i$ between the given accelerometer and the center of the Earth (E.C.). Since phone **104** moves, vector $\mathbf{e}_i$ exhibits an explicit dependence on time; $\mathbf{e}_i=\mathbf{e}_i(t)$. In most practical applications, what one needs to consider is that the accelerometer senses the gravitational acceleration $\mathbf{a}_g$ in stable frame **106** of environment **302**. In addition, the accelerometers are sensitive to the actual acceleration of phone **104** in frame **106**. Thus, since the accelerometers are sensitive to the actual acceleration and the influence of acceleration due to gravity, it is necessary to subtract the influence of gravity. To do this, the accelerometers need to obtain an estimate of the orientation of phone **104**. It is mainly due to the problems associated with pose estimation and gravitational acceleration that accelerometers drift in stable reference frame **106** or gaming frame **134** and only provide indication of relative motion by double integration.

Meanwhile, gyroscopes measure changes in the rotation of phone **104** about the axes of inertial coordinates (X_i, Y_i, Z_i) of frame **318** due to noise and imperfect initial conditions (angular bias). The output of the solid state gyroscope has to be integrated to estimate orientation. As a result, a constant bias error causes an angular error that grows linearly with time. In addition, the integrated noise introduces errors with standard deviation proportional to the square root of time.

During operation of interface **300** the recovery of absolute pose of phone **104** based on images obtained with the aid of radiation **130** is performed as already described in the previous embodiments. However, because console **310** is not always stationary in room **308**, signal **210** preferably includes absolute pose parameters $(x_w, y_w, z_w, \alpha_{wb}, \beta_{wb}, \gamma_{wb})$ and $(x_s, y_s, z_s, \alpha_{sb}, \beta_{sb}, \gamma_{sb})$. In other words, absolute pose parameters in gaming coordinates (X_w, Y_w, Z_w) of gaming frame **134** and in stable coordinates (X_s, Y_s, Z_s) of stable frame **106** of environment **302** are computed and reported in signal **210**. As a result, game application **216** can keep track not only of where phone **104** is with respect to console **130**, but also where they both are in environment **302**, i.e., in room **308**. This information may not be required for all game applications **216**. However, any application **216** that involves an augmented reality that overlaps with environment **302** will typically require this additional data.

In addition, interface **300** also receives signals related to changes in the pose of phone **104** from motion sensor **316**. Unfortunately, such relative pose data from motion sensor **316** is not calibrated with respect to either frame **106** or frame **134**. Thus, it cannot be used directly to corroborate, replace or augment absolute pose data obtained through camera pose recovery in frames **106** and **134**. Consequently, unless a simple "mouse-mode" or "relative pointing mode" is required for user input by gaming application **216**, the relative pose data from motion sensor **316** is not very helpful.

The main advantage of motion sensor **316** is its speed, which may be between 100 Hz and 200 Hz or even higher. Meanwhile, operating camera **144** at such frame rates is very resource intensive and may further be limited by the available level of radiation **130**. Simply put, at frame rates of 100 Hz and above the images recovered by camera **144** may be too dim to extract the non-collinear optical inputs **313** and **314** for algorithms of step **246** (see **Fig. 12**) to yield good camera pose recovery. In addition, processing image data at such rates is computationally intensive and requires a lot of on-board power.

Fortunately, the drift experienced by accelerometers and gyroscopes of motion sensor **316** is typically not significant over short time periods. Specifically, because of single integration and accumulation of errors the gyroscopes can provide good readings of rotations executed by phone **104** over periods of 10 sec or more. Double integration and errors due to imperfect cancellation of gravity due to errors in orientation estimates render accelerometers less robust. Their readings of changes in motion are reliable over periods of a few seconds. The constant bias error causes a position error that grows quadratically with time. Further, the integrated noise introduces errors with standard deviation proportional to time raised to the power of 3/2. This is in addition to imperfect gravity cancellation.

The preferred embodiment takes advantage of the strengths of optical pose recovery with camera **144** and relative pose information from motion sensor **316**. Specifically, absolute pose data from signal **210** is employed to periodically calibrate the gyroscopes and accelerometers of motion sensor **316**. In performing the calibrations, the accelerometers should be calibrated, for example, once every 1-5 sec and the gyroscopes should be calibrated, for example, once every 10-20 sec.

With this strategy, interface **300** can leverage the strength of motion sensor **316** to offset the weakness of camera **144**. By operating camera **144** at a frame rate of just a few Hz or even less than 1 Hz, interface **300** can employ high-quality absolute pose parameters recovered in frames **106** and **134** to keep the accelerometers and gyroscopes calibrated in these frames. For very high-performance, the accelerometers can be calibrated about once every second and the gyroscopes about once every two seconds. Then, while camera **144** is off and not taxing on-board resources of phone **104**, motion sensor **316** can provide its relative pose information to supplement or even interpolate between absolute pose parameters reported by signal **210**.

The relative pose data can be processed on-board phone **104** and submitted to host **310** along with signal **210**. Alternatively, it can be processed separately and sent to host **310** on a dedicated channel for processing off-board. Furthermore, the relative pose data can be related to just one absolute pose parameter or more. In a fully parameterized interface **300**, the relative pose data can be related to all six degrees of freedom. A person skilled in the art of sensor fusion will understand the various tradeoffs and optimizations involved in achieving the best performance with the least resource allocation and power consumption. Further information on this subject is provided by Oliver J. Woodman, "An Introduction to Inertial Navigation", Technical Report Number 696, University of Cambridge, August 2007.

In addition to the above, it is preferable to use data from motion sensor **316** to also stabilize camera **144**. This is important at times when camera **144** cannot support a sufficiently short exposure time t_e, either due to rolling shutter, insufficient level of radiation **130**, excessive angular movement by user **102** or other reasons. At such times, the data from sensor **316** should be sent to image processing electronics **156** to help remove motion blur from the image. Alternatively, or in addition, if lens **146** is adjustable, the data from motion sensor **316** can also be used to actively adjust lens **146**

to avoid motion blur. Active and passive motion blur removal is a subject known to those skilled in the art. The reader is referred to literature in the field of optical image stabilization for further information.

To further decrease the resources dedicated to camera **144** and its power consumption, it is preferable to implement sparse imaging. In fact, the preferred embodiment relates to changes in the typical operation of row and column multiplexing blocks **192**, **194** (see **Fig. 9**). The approach is referred to as sparse-imaging or selective imaging and it is illustrated in **Fig. 17**.

The plan view of photosensor **152** in **Fig. 17** shows a preferred method of allocating pixels **190** for sparse imaging. It is based on the previous embodiment where the stationary object is television **126** with screen **128**. Regions **320** of pixels **190** are not used in this embodiment. Instead, only selected rows and columns are activated by camera **144** to collect image data from radiation **130**.

For example, every 5^{th} or even every 10^{th} row, and every 5^{th} or even every 10^{th} column of pixels **190** belonging to photosensor **152** are active. In addition, regions of interest around image **129'** of marking **129** or around images of other features of interest (e.g., those that can further improve the quality of camera pose recovery) can include active pixels **190**, as shown. In the present embodiment every 10^{th} row and every 10^{th} column of pixels **190** are active, thus drastically reducing the number of pixels **190** that need to be processed by image processing electronics **156**. (Note that **Fig. 17** does not show all pixels **190** and is merely illustrative of the sparse sampling concept.)

Non-collinear optical inputs **132A-D**, **129** and therefore their images **132A'-D'**, **129'** are intrinsically high contrast. That is because edges **132A-D** are the light-to-dark transitions between illuminated screen **128** and the mechanical frame of television **126**. Marking **129**

is usually a highly visible feature by manufacturing design, although its contrast may be lower. In an alternative embodiment, if marking **129** does not provide sufficient optical contrast, the non-collinear optical input for breaking the rectangle symmetry of screen **128** can be the power light typically embedded in the mechanical frame of television **126** or still some other high optical contrast feature attached to or integrated with television **126.**

Sparse column and row imaging works well, because it is known that full images **132A'-D'** are lines. Thus, to reconstruct them, it is sufficient to detect a few of their line segments in the sparse image obtained only from active pixels **190.** The same goes for image **129'** with the additional simplification that image **129'** does not need to be as high-quality since it is may be used for symmetry braking only.

Smart camera technology methods can be applied concurrently or in addition to sparse imaging to further simplify the image capture process and reduce resource allocation on-board phone **104.** For example, when camera **144** is a modern smart camera, it may employ 12-bit grayscale values in pixels **190** to support operation in lower light conditions or, alternatively, to shorten exposure time t_e and/or support an increase in frame rate. Additionally, smart camera **144** may support frame averaging, multiple regions of interest (MROIs) as well as localized brightness adjustment and application of various filtering functions.

As a person skilled in the art will realize, it would be advantageous to apply such image processing functions in sensor **152** rather than having to apply them after demultiplexing in image processing electronics **156.** Furthermore, camera **144** can benefit from any number of the other improvements as well. For example, once images **132A'-D'** and **129'** of edges **132A-D** and marking **129** (representing the non-collinear features) are found in a first full frame image, camera **144** may set regions of interest around these images only. The margin around the images should be large enough to ensure that the

corresponding images do not move outside the region of interest from frame to frame. In this way, the number of pixels **190** needed to track images of the non-collinear optical inputs from frame to frame can be reduced still further than with simple column and row imaging.

The improved performance of interface **212** when phone **104** employs smart camera **144** and interpolates with motion sensor **316** can be leveraged for more involved applications. **Fig. 18** illustrates in a three-dimensional diagram in which an embodiment of application **216'** designed for gaming takes advantage of the preferred embodiment of interface **212** for a shooting game. Note that most of game application **216'** in this embodiment runs on-board phone **104**.

Specifically, game application **216'** employs screen **136** of phone **104** not only for providing visual feedback to user **102**, but also to enable additional interaction with user **102** via an interface **212'**. Game application **216'** of this variety is frequently referred to as a "mobile application" or simply an "app" by those skilled in the art. Such "apps" are typically written in JavaScript, C, C++ as well as many "app development" specific software languages. In the embodiment shown, interface **212'** employs the touch-sensitive screen **136** to display a touch button **142C**.

Game application **216'** takes advantage of the volume parameterized by application coordinates (X_a, Y_a, Z_a) to display digital 3D application environment or gaming environment **252** to user **102**. Note that it is possible to use screen **136** to display gaming environment **252** to user **102**. Normally, however, screen **136** is too small and screen **128** of television **126**, or, in this case screen **312** of game console **310** is better suited for visualizing for showing user **102** gaming environment **252**.

Nevertheless, certain important aspects of the game can be displayed to user **102** on screen **136**. These aspects can involve information that normally interferes with gaming environment **252**. For example,

information about the user's **102** status, score and gaming parameters may be more conveniently communicated to user **102** by visual feedback presented on screen **136**. In the present case, the stars on display **136** indicate to user **102** his/her score.

Fig. 18 shows recovered trajectory **278** of C.O.M. **110** in application coordinates (X_a,Y_a,Z_a). Also shown are short portions of recovered trajectory **278′** of point-of-view $\mathcal{P}$, as well as recovered trajectory **278′′** of motion sensor **316**. The recovered locations of C.O.M. **110**, point-of-view $\mathcal{P}$ and motion sensor **316** are indicated in application coordinates (X_a,Y_a,Z_a) by corresponding primed references **110′**, $\mathcal{P}'$, **316′** for more clarity.

Open points along recovered trajectory **278′** indicate the recovered positions of point-of-view $\mathcal{P}'$ from camera pose recovery performed in accordance with any suitable algorithm. As explained above, this data visualizes the full parameterization of the absolute pose A.P.(t) of phone **104** in application coordinates (X_a,Y_a,Z_a) at the corresponding point in time (when the image was captured by camera **144**). For example, A.P.(t_p) is associated with the first point along recovered trajectory **278′** at image capture time t_p when unit vector was $\hat{u}_a(t_p)$.

Black points along recovered trajectory **278′** indicate the recovered positions of point-of-view $\mathcal{P}'$ based on data from relative motion sensor **316**, and more precisely from its gyroscopes and accelerometers. Therefore, motion sensor **316** enables interpolation of all six degrees of freedom of phone **104** with relative poses collected between the times when camera **144** enables recovery of the absolute pose. In a practical application, motion sensor **316** may operate at up to 200 Hz and camera **144** at just 1 Hz. Therefore, the ratio of black points to open points would be about 200:1 (much larger than shown for illustrative purposes in **Fig. 18**).

It is important to note that motion sensor **316** initially recovers the relative pose with respect to itself. In other words, its relative pose data about phone **104** inherently pertains to trajectory **278′′** of motion sensor **316** and a unit vector drawn from its center (not shown). Therefore, to interpolate trajectory **278** of C.O.M. **110** or, as in this case, to interpolate trajectory **278′** of point-of-view $\mathcal{P}$ a coordinate transformation must be applied to the data provided by motion sensor **316**.

This is easily accomplished since offset vector $\mathbf{i}_{bw}$ in world coordinates of motion sensor **316** from C.O.M. **110** and its orientation can be determined from the optical pose recovery. In addition, offset vector $\mathbf{o}_{bw}$ of point-of-view $\mathcal{P}$ from C.O.M. **110** and its orientation is also known. Thus, the coordinate transformation to be applied to relative pose data of motion sensor **316** to interpolate the pose at point-of-view $\mathcal{P}'$ between optical absolute pose recovery points involves adding the total offset due to both offset vectors $\mathbf{i}_{bw}$ and $\mathbf{o}_{bw}$ and the application of the rotation matrix. Once again, the reader is referred to G.B. *Arfken* (op. cit.) for the various intricacies involved in coordinate transformations.

Focusing now on recovered trajectory **278′** we see the effect of drift in accelerometers and gyroscopes of motion sensor **316**. The drift manifests itself in an accumulating departure δ from trajectory **278′**. The orientation of unit vector $\hat{u}_a(t)$ also drifts with time within some solid angle (not shown). However, once camera **144** obtains the absolute pose from its algorithm, the departure δ from properly recovered trajectory **278′** and the orientation of unit vector $\hat{u}_a(t)$ can both be compensated. At the same time, the gyroscopes and accelerometers of motion sensor **316** should be re-calibrated with the newest absolute pose.

The designer of interface **300** may wish to smoothen the jumps in recovered trajectory **278′** and in camera orientation by applying any

suitable algorithm. Useful reference on this subject is provided by Kenneth Gade, "Introduction to Inertial Navigation and Kalman Filtering", INS Tutorial, Norwegian Centre, FFI (Norwegian Defense Research Establishment).

The considerably better quality of recovered trajectory **278'** and unit vector $\hat{u}_a(t)$ permit game application **216'** to engage user **102** in a more challenging implementation of interface **300** than shopping (which only required good absolute pointing capabilities). Thus, building on the preferred implementation of interface **212**, game application **216'** involves cutting down apples **322** from trees **324** displayed on gaming console **310** in gaming environment **252**. In a preferred embodiment, screen **312** of console **310** permits a very realistic display of this scene with proper depth perception for user **102** (along Z_a-axis of application coordinates (X_a,Y_a,Z_a)). In fact, a number of gaming consoles with high-definition 3D displays capable of realistic 3D scene rendering are now available.

At the present time, user **102** (in this case user **102** is a gamer) has already cut down two apples **322**. A particular apple **322A** is still hanging on branch **326** of a tree **324** that is being swayed by a gusty wind. The objective is to cut down apple **322A** by its stem, without damaging it, so that it remains edible. Gamer **102** moves phone **104** in real 3D environment **302** to get optical axis **152** in gaming coordinates (X_w,Y_w,Z_w) and recovered as axis **275** along unit vector $\hat{u}_a(t)$ in game or application coordinates (X_a,Y_a,Z_a), to cut the stem of virtual apple **322A**. Optical axis **150** of camera **144** thus extends along the correspondingly oriented virtual machete **328**.

Game application **216'** displays machete **328** in the form of a blade to facilitate the task. At time t_q, user **102** has machete **328** pointed directly at apple **322A** but in the wrong position and orientation for cutting. It is indeed clear from the location and orientation of the

blade that making a cut at this point by touching button **142C** on screen **136** and executing a "poke" or "swag" is not wise.

It should be noted that a number of choices are open to the designer of game application **216'** regarding the cutting action. First, for a very realistic gaming experience, it may be desirable to overlap world or gaming coordinates (X_w,Y_w,Z_w) with application coordinates (X_a,Y_a,Z_a) such that the motion of phone **104** in environment **302** is one-to-one with its motion in gaming environment **252.** This also means, that gaming environment **252** extends beyond what can be displayed on screen **312** into real 3D environment **302** in which user **102** resides.

Second, to make the game easier, the designer may choose to map the degrees of freedom of phone **104** with a down-scaling of the angular degrees of freedom. This will make it easier for user **102** to target the stem of apple **322A.**

Third, the distance along the Z_w axis of gaming coordinates (X_w,Y_w,Z_w) can also be scaled for further simplification. Of course, with such simplification gaming application **216'** is no longer as life-like, since a complete one-to-one mapping is lost. Thus, the various scaling functions or even removal of some degrees of freedom in the mapping (e.g., removal of rotation of phone **104** about optical axis **150**) should be weighed against the desired user experience. Indeed, if the application is to be completely life-like, the designer may dispense with internal application coordinates (X_a,Y_a,Z_a) altogether and work in gaming coordinates (X_w,Y_w,Z_w) only. This approach is viable for virtual reality games and life-like simulations.

Fig. 19 illustrates still another advantage of the preferred embodiment of interface **300** employing phone **104** with sensor fusion attained by contemporaneously employing camera **144** and motion sensor **316.** In this embodiment real three-dimensional environment **302** is once again located on the surface of planet Earth **304,** and it is

parameterized by Earth coordinates (X_e,Y_e,Z_e) as previously introduced in **Fig. 16**. Stable coordinates (X_s,Y_s,Z_s) that parameterize stable frame **106** have their origin on the ground (e.g., on a milepost) and are aligned with rails **332** of a train car **330** as shown in **Fig. 19**.

In contrast to previous embodiments, however, user **102** of phone **104** is not at rest in stable frame **106**. Instead, he/she is on train car **330** that is moving in stable frame **106**. User **102** perceives him/her to be in another stable frame **334** in environment **302** that is moving along with train car **330**. To complicate matters, stable frame **334** perceived by user **102** is not moving in a uniform manner. That is because a velocity of train car **330**, described by vector **v** in stable coordinates (X_s,Y_s,Z_s) attached to Earth **304**, is changing. Train car **330** is accelerating along its direction of motion and also slowly turning right along rails **332**. These changes in velocity vector **v** are described by quantities $\Delta\mathbf{v}_y$ and $\Delta\mathbf{v}_x$, respectively. (Strictly speaking, since quantities $\Delta\mathbf{v}_y$ and $\Delta\mathbf{v}_x$ indicate direction in stable coordinates (X_s,Y_s,Z_s) with the corresponding subscripts they are vector components, and therefore scalars. Thus, we do not need to consider them as vectors and it is technically not necessary to use boldface letters for them according to our convention.)

Accelerated frames, such as frame **334**, are referred to in the art as non-inertial. Here, stable frame **334** of user **102** is actually such a non-inertial frame. If user **102** could not see out the window of train car **330**, he/she would only be able to tell that his/her frame **334** is non-inertial by feeling the time rate of change in velocity **v**, or acceleration $\mathbf{a}=d\mathbf{v}/dt$. The ability of user **102** to feel acceleration in the same way as the force of gravity $\mathbf{F}_g$ is due to the principle of equivalence discovered by Albert Einstein. A similar situation is encountered on airplanes, in terrestrial vehicles such as buses or cars, on ships and on amusement rides, to give just a few examples.

Because motion sensor **316** contains gyroscopes and accelerometers, which are inertial sensors, they are subject to the same experiences as user **102** in accordance with the principle of equivalence. Thus, they will not be able to distinguish between motion of phone **104** within frame **334**, and specifically the changes in vector **v**, and the motion of frame **334** in stable frame **106** that is attached to the surface of planet Earth **304** and subject to acceleration $\mathbf{a}_g$ produced by gravity. Indeed, one of the major problems with inertial sensors is that their calibration in non-inertial frames becomes harder and their drift increases faster.

Of course, the reader will realize that Earth's frame **304** is non-inertial too. In fact, it is a rotating frame subject to effects including *pseudo-forces* such as the Coriolis effect and centripetal forces. However, effects due to angular velocity ω_e and acceleration $\mathbf{a}_g$ of Earth **304** are known and typically small (Earth's effects are negligible for large Rossby numbers). Thus, its effects can be compensated for in applications where user **102** is stationary in stable frame **106**. Unfortunately, train car **330** and its associated frame **334** parameterized by world coordinates (X_w,Y_w,Z_w) are in motion that is not known in advance and cannot be accounted for as easily. Hence, the limitations of motion sensor **316** are exacerbated in frame **334** producing much more rapid drift.

Fortunately, in the preferred embodiment of interface **300**, phone **104** is equipped with camera **144** which uses screen **128** to recover its absolute pose as defined world coordinates (X_w,Y_w,Z_w). The latter are attached to screen **128** at its bottom left corner (note that this is a different origin and orientation than in the first embodiment shown in **Figs. 1A-B**). This means that as long as screen **128** does not move inside train car **330** world coordinates (X_w,Y_w,Z_w) will undergo the exact same motion within stable coordinates (X_s,Y_s,Z_s) of stable frame **106** on Earth's surface as does train car **330**. Therefore, world

coordinates (X_w,Y_w,Z_w) are stationary from the vantage point of user **102** and interface **300**.

This means that absolute pose of phone **104** recovered optically in world coordinates (X_w,Y_w,Z_w) by camera **144** is automatically stationary in frame **334**. Therefore, the optically recovered absolute pose can be used to remove the errors due to *pseudo-forces* and drift that the motion sensor **316** experiences due to the changing velocity **v** of train car **330**. In practice, this also means that re-calibration of motion sensor **316** needs to be performed more frequently than in the embodiment described in **Fig. 16**.

The above embodiments are provided for the purpose of familiarizing the reader with the issues involved in properly designing and optimizing a 3D interface that employs the hardware resources available in current smart phones. The actual design and implementation of the interface may require additional information not provided herein. Furthermore, the example applications on which the explanations are based are not to be construed as being in any way recommended to the interface designer. A game developer should rely on his or her best judgment and gaming experience in determining where a 3D interface providing all six degrees of freedom would be most advantageous.

ADDITIONAL APPLICATION DEVELOPMENT RESOURCES

The researches at ESPi have built a number of alpha prototypes that recover absolute pose (all six degrees of freedom) from perspective views of various types of stationary objects and their non-collinear optical inputs. If you have a specific application in mind that employs optical inputs that we have already tested with any of our prototypes, then we may be able to provide you with the corresponding code. We have a YouTube video showing a few of these alpha units at the following link:

http://www.youtube.com/watch?v=Tcqu4NNRXQM

If you are unable to get to the video by clicking on the above link, simply search on YouTube with the key word "naviscribe".

You will notice that the alphas are implemented in several manipulated items such as a wand, a toy gun and a stylus/digital pen. In all of these embodiments, IR LEDs are used as the non-collinear optical inputs for pose recovery purposes. The IR LEDs emit at a wavelength of 950 nm and are time sequenced at about 3.3 kHz. Instead of a CMOS camera, the on-board photosensor is a position-sensing device (PSD) with an IR filter that passes radiation centered on 950 nm only. The PSD is a low-cost reverse-biased p-n photodiode that captures light from each of the IR LEDs (one at a time) and directly reports its centroid. Using the PSD and sequentially flashing IR LEDs eliminates the need for computationally intensive image processing that is required when using a pixellated photosensor. The delay you will notice in the above-referenced YouTube video is due to MatLab (approximately 200 ms delay). The ESPi optical navigation algorithms actually support pose recovery on the order of 10-20 ms.

ESPi researchers have also tested in a breadboard set-up the approach in which the on-board photosensor is a CMOS camera. In this case, the non-collinear optical inputs were the edges and corners of a

regular 8.5 x 11 sheet of white paper on a grey desk. The manipulated item was a long stylus with an inking nib. This breadboard unit was built to calibrate the computational load for absolute pose recovery at frame rates up to 100 Hz and a target in-plane accuracy of about +/- 1 mm. (For comparison, it is noteworthy that at a vastly lower processing load, the embodiments employing a PSD and flashing or strobing IR LEDs support in-plane accuracy of down to 0.2 mm or better.) The lighting conditions were low (typical office environment) and the contrast ratio on the corners and edges was hence fairly poor. Under those conditions, the computational load required an ARM7-type or faster processor. It was also observed that improvements in edge contrast had a large effect on the performance of pose recovery algorithms. Based on those findings, it is expected that designating edges and corners of an illuminated display screen as the non-collinear optical inputs will result in a massive performance improvement as compared to the breadboard set-up.

The code for recovering absolute poses in the PSD and strobing IR LEDs demos was written in MatLab. One specific navigation code was compiled into C++ to run on low-cost non-floating point processors such as the PXA270 by Intel (500 MHz clock). When using point-sources as the non-collinear optical inputs (e.g., strobing IR LEDs mounted around the bezel of the host display screen) the load on the PXA270 was less than 20% at full frame rate, and even under 15%. This is equivalent to about 120 MIPs (Million Instructions Per Second).

The number of instructions per second will increase as the optical inputs become more difficult to detect and identify. For example, when using lines representing the edges of a screen of the television unit, the computational load will likely increase by a factor of 5 or more. This is why sensor fusion with an inertial unit on-board the smart phone is an excellent solution. Since the gyros and accelerometers can provide readings at 200 Hz, the optical inputs only need to be processed at intervals over which the drift in the

inertial units starts being significant. Since these times are on the order of seconds, the optical computations can be performed at sub 1 Hz rates, thus vastly reducing the computational burden.

In addition to the optics, ESPi researchers have studied the sensor fusion problem. The drift and calibration problems of inertial devices are in some cases non-trivial. That is because of the different ways in which drift accumulates on gyros versus accelerometers. Knowing exactly when to re-calibrate and what situations to avoid is important. We can provide additional guidance to the app interface designer in making the right sensor fusion choices for designing an effective 6 D.O.F. interface.

For any additional information, please contact us at:
info@4espi.com

To arrange a meeting, call Marek Alboszta during regular business hours at: (650) 862-1085 (cell)