LOCATIVE VIEW:
Visualizing Geo-Referenced Objects in Space

Mark Carlotto, Mark Nebrich, and Paul Hylton
General Dynamics Advanced Information Systems

ABSTRACT

Locative Viewing is a method for visualizing geographically-referenced 3-D objects in the local coordinate system of a geographically-referenced observer. A computer-graphics rendering of nearby geo-objects is superimposed over the visual surroundings of the observer as seen by a camera. This rendering changes as the observer moves. Locative viewing can be accomplished with a mobile device that 1) is able to determine its geographic location, and orientation, 2) contains a camera and image display, and 3) can project and overlay objects within the field of view of the camera with the camera image. A preliminary implementation of a locative viewer using Apple's iPhone is described and results presented.

INTRODUCTION

Computer graphics techniques generate 2-D views of 3-D objects that exist in a global 3-D coordinate system. A *locative view* is a rendering of geo-referenced 3-D objects in a special kind of coordinate system – one that is based on the location of the observer. Like computer graphics rendering, locative views visually depict representations of points, lines, surfaces, volumes, etc. The difference is the rendering is in the view-space of the observer, using the visual surroundings of the observer as background and reference. This 1) eliminates the difficulty in visually correlating a map-view with one's surroundings and 2) significantly reduces mobile communications bandwidth (most of which is used in sending the map or image background over which geo-referenced objects are often displayed).

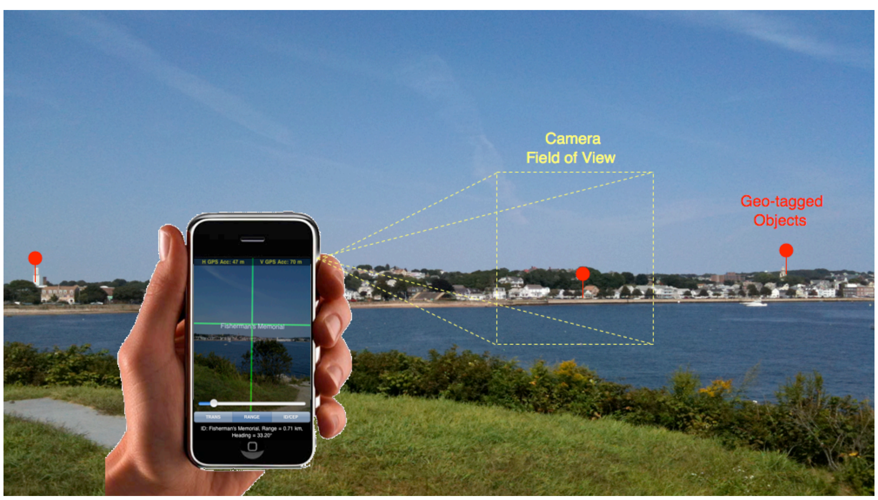


Fig. 1 Pointing a locative viewer toward nearby geo-tagged objects reveals their existence in space and provides information about them.

Key to locative viewing [1] is a mobile device that contains sufficient sensing resources to accurately determine its location and orientation in space. Six measurements are required: latitude, longitude, elevation, heading, tilt, and rotation. The first three are provided by a global positioning system (GPS) receiver. If the observer is moving and looking along the direction of motion, the heading can also be computed from successive GPS latitude and longitude readings. In general, however, a magnetometer is required to determine the direction of magnetic north, from which true north can be determined. The last two parameters, tilt and rotation, are determined from the local gravity gradient by means of a three-axis accelerometer.

Fig. 2 shows the functional architecture of a *locative viewing* device. GPS, magnetometer, and accelerometer sensors are used to determine the device's location and orientation. This establishes the location and orientation of the device's camera. Using this information, a projection algorithm determines which objects in a set of geo-referenced objects fall within the view frustum of the camera. These are then projected into the image plane of the camera and superimposed over the image obtained from the camera. This process occurs continuously at the frame rate of the camera system. The geo-objects may be either uploaded to the device from a laptop, or downloaded from a network server over a wireless network.

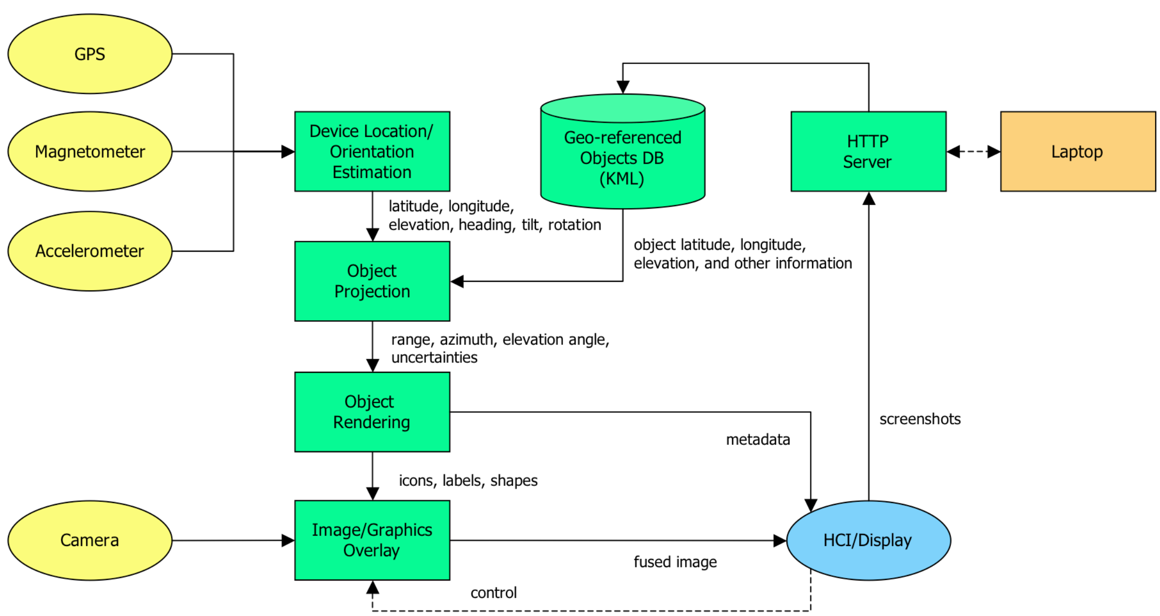


Fig. 2 Functional architecture of a locative viewer. Items in yellow are sensors, blue is the human-computer interface, green is the processing and storage on the device, and orange is an external networked computer.

Geo-referenced objects are defined by sets of latitude-longitude-elevation values, represented for example as a KML file [2]. Given a set of objects and a device location, the range, azimuth, and elevation angle between each object and the device can be computed, from which the screen coordinates are determined. Details are provided in Appendix A. The following sections describe our iPhone implementation and present experimental results. Due to the limitations of the GPS receiver, magnetometer, and accelerometer, the projected locations of objects may deviate (in some cases significantly) from their true location. A practical device must be able to estimate these errors and their effect on viewing. One such method is discussed later in the paper.

IMPLEMENTATION

An initial version of a locative viewer has been successfully implemented using Apple's iPhone 3GS. A screenshot from the application is shown in Fig. 3. The horizontal and vertical GPS location and magnetometer (compass) heading errors at the top of the screen are the errors reported by the iPhone's CoreLocation API [3]. The left button on the bottom of the app selects the slider controller mode: range eliminates objects beyond a cutoff distance that is controlled by the slider, transparency changes the graphical overlay from opaque (no camera image) to transparent (no graphics), and azimuth bias provides a means of correcting magnetometer errors (see discussion at end of the paper). Objects can be rendered in different ways: as "target" icons, as labels, or as ellipses. Information about objects (label, description, range, and heading) is displayed in the text window below the three control buttons. The description is of the object nearest the center of the display at any given moment. Touching the right control button toggles between showing reports, the iPhone's current geo-location, or the app's URL in the text window. Tapping the screen starts a screenshot utility that captures a frame once a second, saving screenshots in the documents directory, which can be downloaded wirelessly to a laptop from the app's URL. The URL is also used for uploading data files to the iPhone from a laptop.

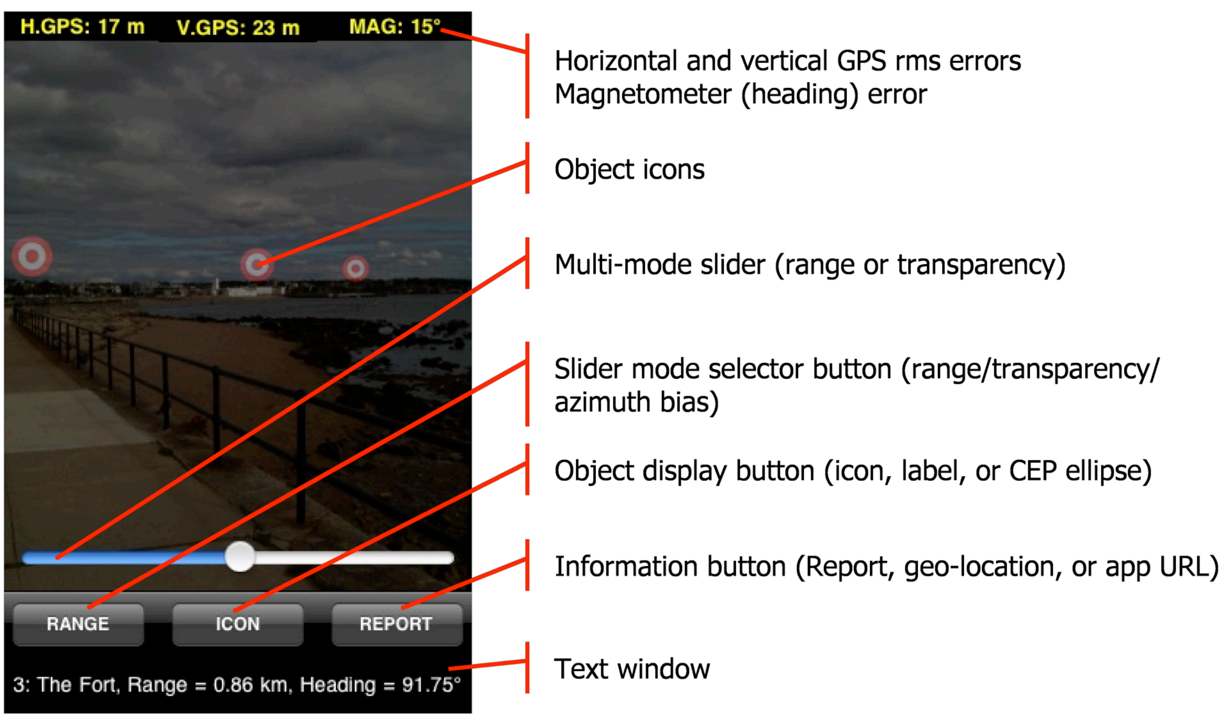
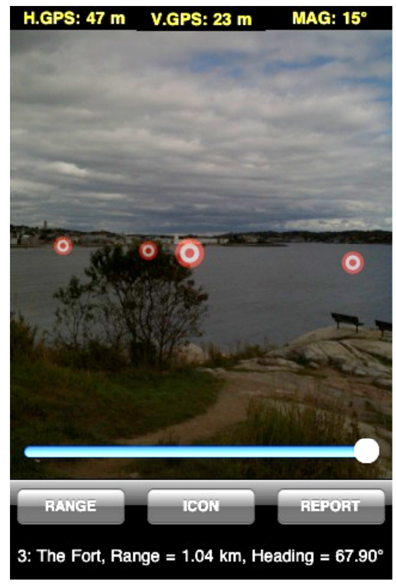


Fig. 3 Screenshot of Locative View implemented on Apple's iPhone.

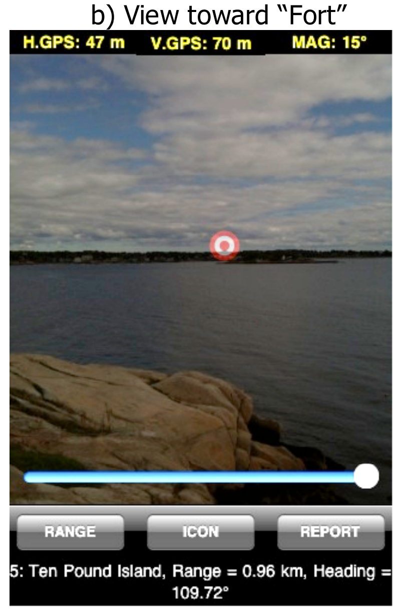
EXPERIMENTAL RESULTS

Locative View has been tested indoors and outdoors, in fixed and moving scenarios. We use Google Earth to create KML files of object locations that are uploaded to the device. In the first test, at a fixed location, we panned 180° to look at a number of points of interest along the waterfront in Gloucester MA (Fig. 4a). Representative screenshots are shown in Fig. 4b-c. The accuracy of the GPS and magnetometer are key to accurate localization. In this experiment the iPhone's GPS accuracy was 17-47 meters horizontal, 23-70 meters vertical, and 15° in heading (azimuth). Table 1 shows true and measured heading and range values for sample placemarks. Overall there is good agreement between true and measured values since the GPS errors are much less than the range.

a) *Locative View* and placemarks (Imagery courtesy *Google Earth)*



b) View toward "Fort"



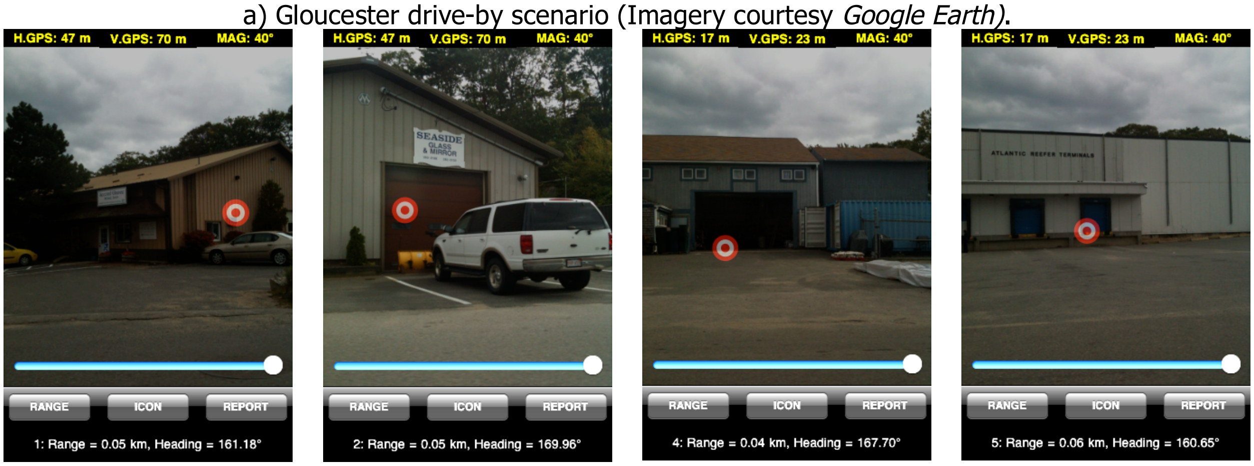
c) View toward Ten Pound Is.

Fig. 4 Fixed location testing scenario along Gloucester's waterfront

Table 1 Test results for selected points of interest along the waterfront

Point of Interest	*True Heading*	*Measured Heading*	*True Range*	*Measured Range*
Stage Fort Park	225	221	0.22	0.21
High School	0	355.4	0.88	0.89
Fisherman's Memorial	30	33.2	0.7	0.71
City Hall	45	40.8	1.44	1.45

a) Gloucester drive-by scenario (Imagery courtesy *Google Earth*).



b) Incorrect ID c) Incorrect ID d) Correct ID e) Correct ID

Fig. 5 Drive-by testing scenario in Gloucester

A "drive-by" test was then performed where the device was pointed left out the driver's side window as the car was driven past a number of businesses (Fig. 5a). Inside a car the iPhone's GPS and magnetometer tend not to perform as well as they do outdoors. Initially the accuracies were lower than before (47 meters horizontal, 70 meters vertical, and 40° in azimuth). As a result the first few locations are misidentified (Fig. 5b-c). Midway along the route the GPS accuracy improved and subsequent locations were correctly identified (Fig. 5d-e).

DISCUSSION

In general *Locative View* performs best outdoors in open natural environments. GPS accuracy is often reduced in urban/mountainous areas and indoors. Magnetometer accuracy is affected by electromagnetic interference and large metal objects (above and below ground). Let σ_H and σ_V be the horizontal and vertical rms accuracy (in meters) of the GPS, and σ_θ be the magnetometer accuracy (in degrees). Assuming the geo-objects themselves are accurately located, the range error is simply σ_H. The heading angle error is $\sigma_\theta + \tan^{-1}(\sigma_H / r_0)$, where r_0 is the range, and the elevation angle error is $\tan^{-1}(\sigma_V / r_0)$. Errors in range affect viewing to a lesser extent than errors in heading and elevation.

Several options are provided in the iPhone app to help mitigate GPS and magnetometer errors. When $\sigma_H \approx r_0$ the heading error is large and accurate viewing is not possible. This often happens when using the viewer indoors. One option when viewing from a fixed location is to determine the latitude-longitude-elevation of the device by some other means (e.g., using Google Earth) and set the coordinates manually (Fig. 6). In places where the number of GPS satellites in line of sight is reduced, the horizontal error may be acceptable, i.e., $\sigma_H << r_0$ but the vertical error may be large relative to range. In these situations we can estimate the elevation of the device as a weighted sum of the elevations of nearby points of interest, where the weighting factor is inversely proportional to distance.

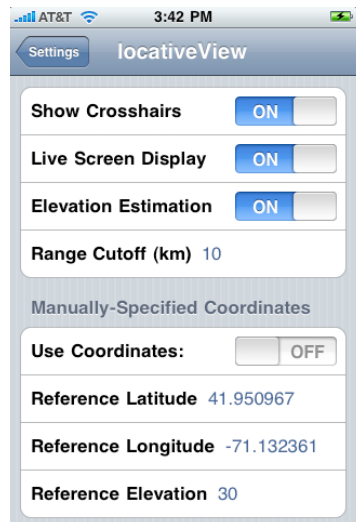


Fig. 6 Locative View settings

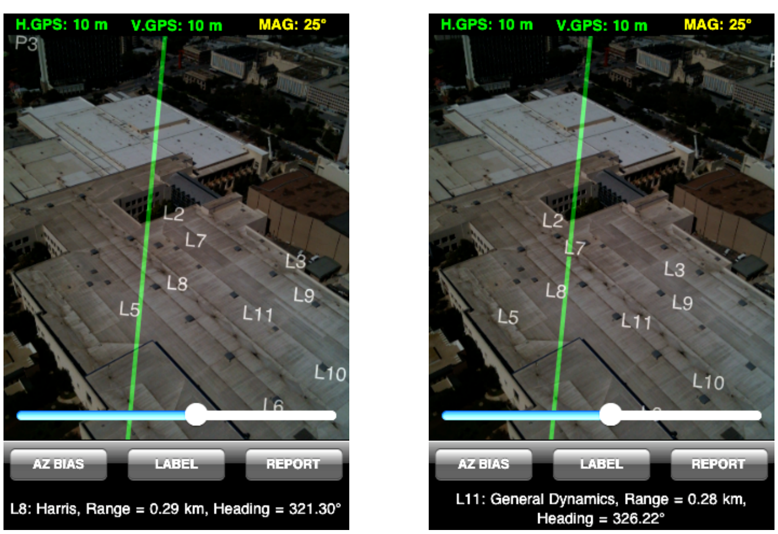


Fig. 7 Magnetometer correction by adjusting azimuth bias

Even if the GPS accuracy is good, the magnetometer error ultimately limits performance since it affects heading accuracy, regardless of range. In general given the limitations in sensing accuracy it may be necessary to use visual cues from the camera view to refine the location and orientation of the device. One method, which has been implemented on the iPhone, uses known landmarks to correct for magnetometer errors by aligning the landmarks with their corresponding icons. This results in a better localization of unknown objects within the field of view. The example in Fig. 7 is a view of the 2009 GEOINT conference exhibition hall looking down from the top of the Tower of the Americas in San Antonio TX. Knowing the location of the conference registration booth (L7) provides a more accurate location of vender booths on the floor.

SUMMARY

We have described the theory and implementation of a locative viewer on Apple's iPhone. The iPhone 3GS was the first commercially-available device that contained a camera, GPS, accelerometer, and magnetometer – the four requisite sensors needed for locative viewing. The device was tested outdoors in a number of places including Gloucester MA, Washington DC, and San Antonio TX. It has also been used indoors with limited success. While GPS accuracy is important we have found the magnetometer accuracy to be a critical factor in device performance. Future work will address this and on the development of visual (and inertial) navigation techniques that supplement the GPS and magnetometer.

References

[1] http://en.wikipedia.org/wiki/Locative_media

[2] http://code.google.com/apis/kml/documentation/kmlreference.html

[3] http://developer.apple.com/iPhone/library/documentation/CoreLocation/Reference/ CoreLocation_Framework/index.html

APPENDIX A

Here we provide details on the underlying mathematics. We start with the equations for computing the range, azimuth, and elevation angle of a geo-object relative to the viewer. From this we can compute the screen coordinates of geo-objects.

Range – To determine range, the locations of the device and the objects are first converted from latitude, longitude, and elevation to earth-centered earth fixed (ECEF) coordinates:

$$\begin{aligned} x &= (N+h)\cos\phi\cos\lambda \\ y &= (N+h)\cos\phi\sin\lambda \\ z &= [(N(1-e^2)+h]\sin\phi \end{aligned} \tag{1}$$

where ϕ, λ, and h are the geodetic latitude, longitude, and elevation above the ellipsoid, x, y, and z are the ECEF coordinates, $N = R_{eq} \big/ \sqrt{1-e^2\sin^2\phi}$, $e^2 = (R_{eq}^2 - R_{polar}^2)/R_{eq}^2$, and R_{eq} and R_{polar} are the Earth's equatorial and polar radii. The range between an object and the device is given by the Euclidean distance between points in 3-space:

$$d = \sqrt{(x_{object} - x_{device})^2 + (y_{object} - y_{device})^2 + (z_{object} - z_{device})^2} \tag{2}$$

Azimuth – The azimuth angle (relative to true north) of an object relative to the device is

$$\alpha = \frac{\pi}{2} - \tan^{-1}\left[\frac{R_{polar}\left(\phi_{object} - \phi_{device}\right)}{R_{eq}\cos\phi_{device}\left(\lambda_{object} - \lambda_{device}\right)}\right]. \tag{3}$$

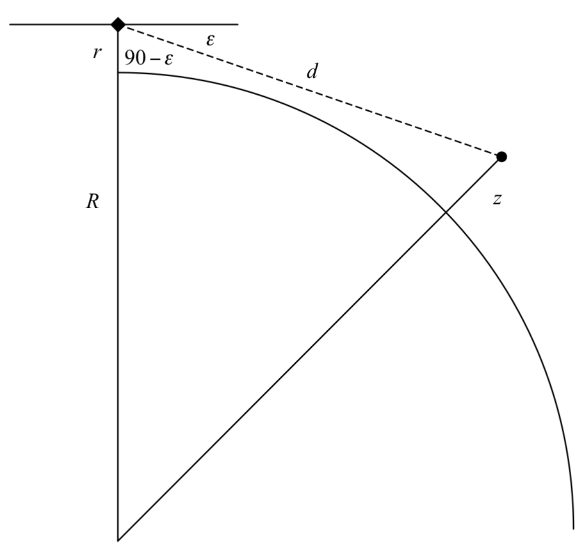


Fig. A-1 Determining the elevation angle of an object relative to the device taking into account its altitude, the altitude of the observer, and range.

Elevation Angle – With reference to Fig. A-1 the elevation of an object relative to the device is computed using the law of cosines: $c^2 = a^2 + b^2 - 2ab\cos\gamma$. Let R, r, ε, z, and d be average radius, device elevation, elevation angle, object elevation, and range, respectively. Making substitutions:

$$(R+z)^2 = (R+r)^2 + d^2 - 2(R+r)d\cos(90-\varepsilon) \quad (4)$$

and rearranging,

$$\begin{aligned}(R+z)^2 &= (R+r)^2 + d^2 - 2(R+r)d\cos(90-\varepsilon)\\ \varepsilon &= 90 - \cos^{-1}\left[\frac{(R+z)^2-(R+r)^2-d^2}{-2(R+r)d}\right]\end{aligned} \quad . \quad (5)$$

Screen Coordinates – Geo-objects can now be projected to screen coordinates by the transform:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\rho & \sin\rho \\ -\sin\rho & \cos\rho \end{bmatrix} \begin{bmatrix} \dfrac{X + X(\alpha - \theta + \rho)}{H} \\ \dfrac{Y - Y(\varepsilon - \tau)}{V} \end{bmatrix} \quad (6)$$

where *X* is the half-width and *Y* is the half-height of the screen (in pixels), τ, ρ, and θ are the tilt, rotation, and heading angles of the device, *2V* and *2H* are the vertical and horizontal field of view of the camera, and α and ε are the azimuth and elevation angles of the objects (all in degrees).

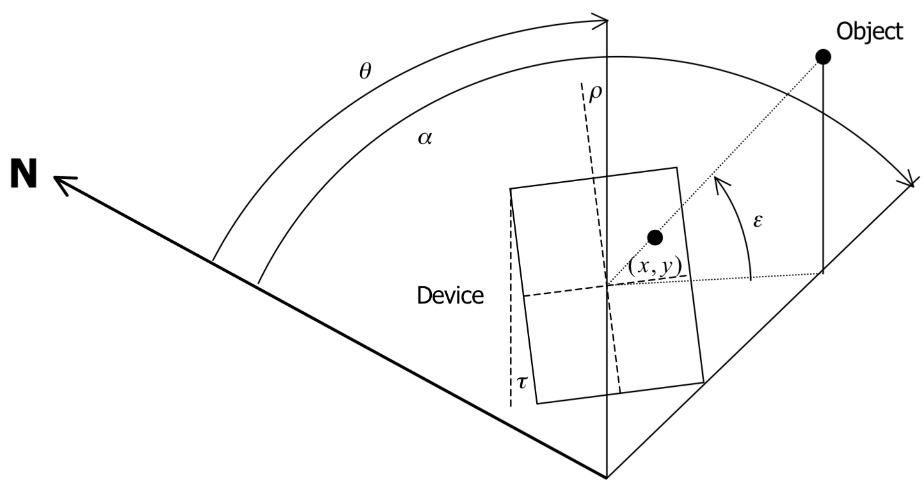


Fig. A-2 Geometry for projecting geo-referenced objects into the image plane of the device.

The iPhone app provides an option for overlaying cross-hairs on the display, which define the axes of the 2D screen coordinate system. The horizontal line defines the horizon line. The vertical line runs from the nadir point above to the sub-sensor point below.