NASA/TM–2017–219675

Map Projection Induced Variations in Locations of Polygon Geofence Edges

Paula Neeley and Anthony Narkawicz
Langley Research Center, Hampton, Virginia,

October 2017

NASA/TM–2017–219675

Map Projection Induced Variations in Locations of Polygon Geofence Edges

Paula Neeley and Anthony Narkawicz
Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

October 2017

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an offical endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:

NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320
443-757-5802

Abstract

This Paper under-estimates answers to the following question under various constraints: If a geofencing algorithm uses a map projection to determine whether a position is inside/outside a polygon region, how far outside/inside the polygon can the point be and the algorithm determine that it is inside/outside (the opposite and therefore incorrect answer)? Geofencing systems for unmanned aircraft systems (UAS) often model stay-in and stay-out regions using 2D polygons with minimum and maximum altitudes. The vertices of the polygons are typically input as latitude-longitude pairs, and the edges as paths between adjacent vertices. There are numerous ways to generate these paths, resulting in numerous potential locations for the edges of stay-in and stay-out regions. These paths may be geodesics on a spherical model of the earth or geodesics on the WGS84 reference ellipsoid. In geofencing applications that use map projections, these paths are inverse images of straight lines in the projected plane. This projected plane may be a projection of a spherical earth model onto a tangent plane, called an orthographic projection. Alternatively, it may be a projection where the straight lines in the projected plane correspond to straight lines in the latitude-longitude coordinate system, also called a Plate Carrée projection. This paper estimates distances between different edge paths and an oracle path, which is a geodesic on either the spherical earth or the WGS84 ellipsoidal earth. This paper therefore estimates how far apart different edge paths can be rather than comparing their path lengths, which are *not* considered. Rather, the comparision is between the actual locations of the edges between vertices. For edges drawn using orthographic projections, this maximum distance increases as the distance from the polygon vertices to the projection point increases. For edges drawn using Plate Carrée projections, this maximum distance increases as the vertices become further from the equator. Distances between geodesics on a spherical earth and a WGS84 ellipsoidal earth are also analyzed, using the WGS84 ellipsoid as the oracle. Bounds on the 2D distance between a straight line and a great circle path, in an orthographically projected plane rather than on the surface of the earth, have been formally verified in the PVS theorem prover, meaning that they are mathematically correct in the absence of floating point errors.

Contents

1 Introduction

The Federal Aviation Administration recently released Part 107 of the Federal Aviation Regulations, which presents policy for commercial use of small unmanned aircraft systems (UAS). These regulations cover UAS weighing less than 55 pounds, operating within visual line-of-sight, and operating under 400 feet above ground level or within 400 feet of a structure. The guidelines are designed to enable commercial use of UAS in non-segregated airspace, the motivation being due, in part, to a dramatic increase in UAS traffic in recent years. Ensuring that aircraft stay in approved operational areas can help mitigate the safety concerns presented by this increase. One way this can be accomplished is by using a *geofencing* system, which can ensure that an aircraft stays inside stay-in regions (authorized safe areas) and outside stay-out regions (registered or dangerous areas). NASA has developed several systems to provide geofencing capability, including Safeguard [2] and ICAROUS [1]. Safeguard enforces stay-in and stay-out regions in a framework that is designed for (but does not necessarily guarantee) reliability and dependability, through its ability to terminate flight by discontinuing power to a UAS's motor. ICAROUS is open-source and relies on the NASA PolyCARP package [7] of algorithms for computations on polygons.

PolyCARP is a suite of algorithms that includes software implementations as well as formal models for calculating containment and collision information for polygons. For example, PolyCARP can be used to determine whether a given point is inside or outside of a particular polygonal region. In the Safeguard and ICAROUS systems, stay-in and stay-out regions are modeled using 2D polygons with minimum and maximum altitudes. The vertices of the polygons are input as latitude-longitude pairs, and the edges as paths between adjacent vertices. One problem with modeling regions this way is that it is not immediately clear what is meant by *the path* between two adjacent vertices, since there is more than one way to generate such a path over the nonlinear surface of the earth. It can be generated as a geodesic on the surface of the earth, where the earth is modeled as either a perfect sphere or an oblate ellipsoid. In many geofencing applications, the path between two polygon vertices is the inverse image of a straight line in a map projection of the earth's surface, such as an orthographic or Plate Carrée projection. ICAROUS and Safeguard both use orthographic projections, which map latitude-longitude points canonically onto the sphere and then project them onto a tangent plane. Map projections reduce the containment problem to a 2D problem where standard polygon containment algorithms can be used, but due to the nonzero curvature of the earth, they necessarily introduce distortion, and this paper attempts to quantify the effect of this distortion on geofencing.

This paper presents upper bounds for the maximum distance between a polygon edge path and the WGS84 ellipsoidal geodesic, which is used as an oracle, for several path generation methods, including great circles and straight lines from several different map projections. For edges drawn using orthographic projections, this maximum distance increases as the distance between the polygon vertices and the projection point increases. For edges drawn using Plate Carrée projections, this maximum distance increases distance between the vertices and the equator increases. Distances between geodesics on a spherical earth and the WGS84 ellipsoidal earth are calculated. Bounds on the 2D distance between a straight line and a great circle path, in an orthographically projected plane rather than on the surface of the earth, have been formally verified in the PVS theorem prover, meaning that they are mathematically correct if executed with exact real arithmetic. The analysis in this paper is intended to aid the design process for geofencing systems. In particular, it provides data useful in determining the maximum polygon size that can be used for geofencing applications.

Sections 2 and 3 provide background on map projections, with a focus on orthographic, Mercator, and Plate Carrée projections. Section 4 provides formally verified upper bounds for the maximum distance between a great circle and the curve derived from a straight line after an orthographic projection to a tangent plane is taken. These bounds have been proved using SRI's Prototype Verification System (PVS). Section 5 estimates the variations in locations of polygon edges due to different map projections and spherical earth models. Sections 6 and 7 discuss related work and conclusions.

2 Background

The earth can be approximated by a variety of spherical models, including a sphere of equal surface area or volume to that of the earth. A more common choice is known as the *navigation sphere*, defined as a sphere in which one minute of arc on a great circle is equal to one nautical mile, or 1852 meters [3].

The shape of the earth, however, more closely resembles an oblate ellipsoid, i.e. an ellipsoid of revolution attained by the rotation of an ellipse around its shorter axis. The reference ellipsoid most commonly used in engineering applications is the *WGS84 ellipsoid*, an ellipsoid for which one minute of longitude at the equator is equal to 1.0018 nautical miles (approximately 1855 meters) [3]. WGS84 is also the reference ellipsoid used in GPS navigation, and in most cases, is the oracle ellipsoid used to measure the accuracy of methods for generating polygon edges. For these reasons, the navigation sphere and WGS84 ellipsoid are the spheroids used in this paper. Polygon edges on the WGS84 ellipsoid are modeled as geodesics using the formulas of Karney [5] via the Python components of the GeographicLib package [6].

3 Map Projections

A *map projection* is a transformation of latitudinal and longitudinal points from the surface of the earth onto points on a plane. Many UAS geofencing systems use map projections to reduce the containment problem to a 2D problem where standard polygon containment algorithms can be used. This includes the NASA systems Safeguard and ICAROUS.

Given the nonzero curvature of the earth's surface, map projections necessarily introduce distortion. Map projections can produce distortion of multiple properties, including angle, area, bearing, distance, scale, and shape. However, map projections can often be constructed to preserve at least one metric property, and maps are typically categorized by the extent to which they preserve a given property. For this reason, there exist various types of map projections, each with advantages depending on its intended use.

The Mercator projection, historically used for navigation, is a *cylindrical, conformal* map projection that preserves rhumb lines, i.e. lines of constant bearing. However, it overemphasizes areas far from the equator. See Figure 1. Following Snyder, a *cylindrical* projection is one in which lines of longitude are represented as equidistant parallel lines, with straight latitudinal lines intersecting the longitude lines at right angles. A map is *conformal* if all angles are preserved locally [8].

The NASA geofencing systems Safeguard and ICAROUS use orthographic projections, which map latitude-longitude points canonically onto the sphere and then project them onto a tangent plane at some point $\boldsymbol{p}$ on the surface. An orthographic projection characterizes exactly one hemisphere of the globe, centered at $\boldsymbol{p}$, with the outline of the globe forming a great circle on the plane. See Figure 2. In the classification of map projections, it is an *azimuthal, perspective* projection. An *azimuthal* projection is one in which the direction from a given central point to any other point is represented correctly. A *perspective* projection is one in which straight lines are emitted either from a given point or from infinity through the surface of the sphere onto a tangent plane [8]. Orthographic projections are chosen for Safeguard and ICAROUS because they produce very small changes in relative locations

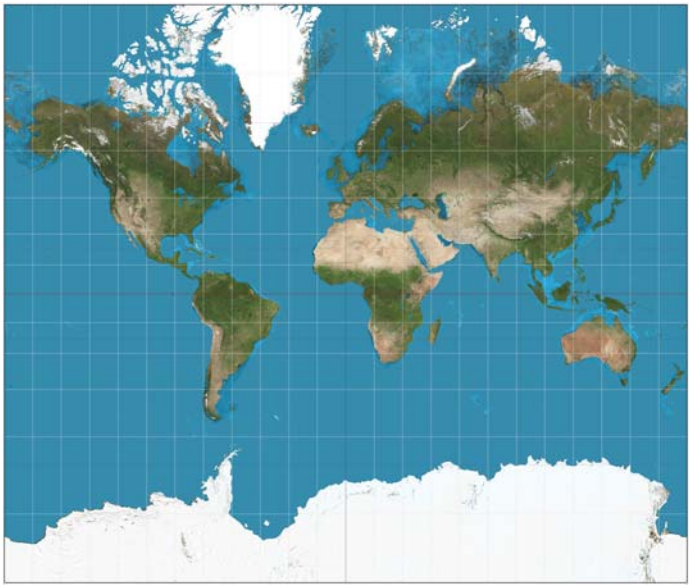

Figure 1: A Mercator Projection

CC BY-SA 3.0 - image by Daniel R. Strebe - August 15, 2011 - https://commons.wikimedia.org/wiki/File:Mercator_projection_SW.jpg

for points close to the projection point. This statement is not made precise in this section because it is, in part, the point of this paper, and is therefore discussed in later sections.

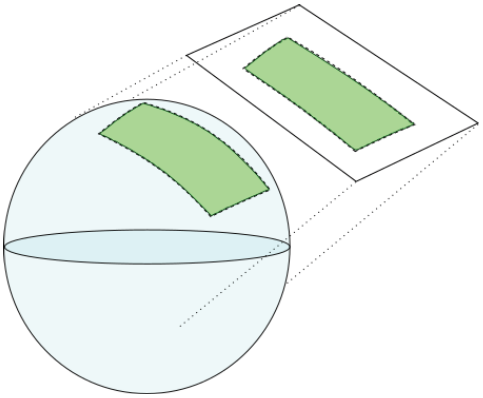

Figure 2: Image of a Polygon Under an Orthographic Projection

The trigonometric formulas for orthographic projections are as follows [9]. Let R be the radius of the earth under a spherical model, and (ϕ_0, λ_0) be the latitude and longitude of the projection point $\boldsymbol{p}$ on the sphere. Then for latitudes and longitudes (ϕ, λ), an orthographic projection onto the (x, y) tangent plane is given by:

$$\begin{aligned} x &= R\cos\phi\sin(\lambda - \lambda_0), \\ y &= R(\cos\phi_0\sin\phi - \sin\phi_0\cos\phi\cos(\lambda - \lambda_0)). \end{aligned}$$

To ensure that points on the opposite hemisphere are not plotted, latitudes outside the range of the projection are eliminated by calculating the angular distance c from the point of projection $\boldsymbol{p}$. A point represented by a latitude-longitude pair (ϕ, λ) is not plotted if $\cos(c)$ is negative:

$$\cos(c) = \sin\phi_0\sin\phi + \cos\phi_0\cos\phi\cos(\lambda - \lambda_0).$$

The inverse formulas are given by:

$$\lambda = \lambda_0 + \arctan\left(\frac{x \sin c}{\rho \cos c \cos \phi_0 - y \sin c \sin \phi_0}\right),$$
$$\phi = \arcsin\left(\cos c \sin \phi_0 + \frac{y \sin c \cos \phi_0}{\rho}\right),$$

where $\rho = \sqrt{x^2 + y^2}$ and $c = \arcsin \frac{\rho}{R}$.

In addition to Mercator and orthographic projections, this paper also considers Plate Carrée projections, where points on the earth's surface are simply mapped to their corresponding points on a scalar dilation of the latitude-longitude plane. This projection is considered because it is the simplest map projection to implement and *should* have minimal impact on geofencing for *small* polygons, a claim that is more precisely quantified in later sections. The general formulas for a Plate Carrée projection are given by:

$$x = (\lambda - \lambda_0),$$
$$y = (\phi - \phi_0)$$

where (ϕ_0, λ_0) is the latitude-longitude of a point around which the projection has *minimal distortion*. Often, the x coordinate is multiplied by $\cos(\phi_0)$, making the projection a more general *Equirectangular projection* [8].

4 Formally Verified Upper Bounds

An Orthographic projection is sometimes used in geofencing applications because it minimizes distortion near the projection point. This section presents formally verified bounds on the distance between a polygon edge, calculated using using an orthographic projection, and the great circle between the same vertices. The distance between these two paths is computed in the orthographically projected plane. That is, the presented bounds are bounds on the distance between two paths in the projected plane: (1) a straight line between two projected vertices, and (2) the projected image of the great circle between those vertices. The calculated bound, for an example edge, is shown in red in Figure 3. In that figure, containment in the polygon region is computed by first orthographically projecting the four vertices to the plane and then drawing the polygon that they form with straight lines as sides. The polygon is shown in green in the orthographic plane, and its preimage is shown in green on the surface of the sphere. The two regions are both shown in green to indicate that they correspond to each other under the projection.

These bounds are therefore bounds on the distance between two curves in the 2D tangent plane rather than on the surface of the sphere. This may be seen as a drawback of the proved bounds, but it should be noted that the bounds presented in this section are expected to be a very close approximation to the great circle distance between the corresponding paths on the surface of the sphere. It is intended that they can be used, along with the numerically computed bounds on edge location variations for orthographic projections, presented in Section 5, to inform users of geofencing systems regarding the size of acceptable geofencing polygons.

These bounds are formally specified and verified for a finite set of predefined distances using the PVS theorem prover. The specification of the function `sphere_to_2D_plane` in PVS is considered to be the formal definition of the orthographic projection algorithm.

Throughout this paper, **bold-face** letters denote 3D vectors unless otherwise noted. Vector operations such as addition, subtraction, scalar multiplication, dot product, i.e.

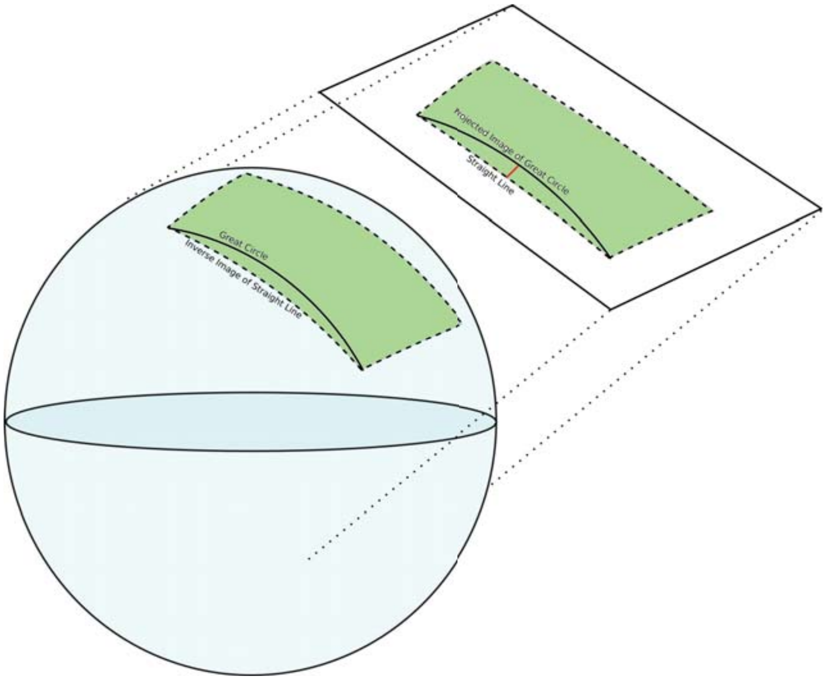
Projected Image of Great Circle
Straight Line
Great Circle
Inverse Image of Straight Line

Figure 3: Distance, shown in red, between the orthographically projected image of the great circle (on the sphere) and the straight line

$\boldsymbol{s} \cdot \boldsymbol{v} \equiv s_x v_x + s_y v_y + s_z v_z$, and the norm of a vector, i.e. $\|s\| \equiv \sqrt{\boldsymbol{s} \cdot \boldsymbol{s}}$, are defined as in 3D Euclidean geometry.

Given two points $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$ on the surface of the sphere of radius R, the predicate `on_chord?` determines if a point $\boldsymbol{c}$ is on the chord that cuts through the earth between $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$:

$$\texttt{on\_chord?}(\boldsymbol{q_1}, \boldsymbol{q_2})(\boldsymbol{c}) : \texttt{bool} \equiv \exists\, t \in [0,1] : \boldsymbol{c} = \boldsymbol{q_1} + t(\boldsymbol{q_2} - \boldsymbol{q_1}).$$

Every point $\hat{\boldsymbol{c}}$ on the great circle between $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$ is a constant multiple of some point $\boldsymbol{c}$ on the chord between $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$. In fact, it sits on the surface of the sphere directly above $\boldsymbol{c}$. The function `lift_to_surface` is defined that lifts the point $\boldsymbol{c}$ to the point $\hat{\boldsymbol{c}}$ on the surface.

$$\texttt{lift\_to\_surface}(R \in \mathbb{R}^+, \boldsymbol{c} \neq \boldsymbol{0}) : \{\hat{\boldsymbol{c}} : \|\hat{\boldsymbol{c}}\| = R\} \equiv \frac{R}{\|\boldsymbol{c}\|} \cdot \boldsymbol{c}$$

An orthographic projection onto a tangent plane can be defined in more than one way. One method to define an orthographic projection is by using the formulas from Section 3, which assumes that the input points are in spherical coordinates. The orthographic projection is defined in PVS as a function called `sphere_to_2D_plane`, which assumes that the input points are in Euclidean coordinates, i.e. 3D vectors with norm R. The inputs to this function are a projection point $\boldsymbol{p}$ at which the tangent plane is located, and a point near $\boldsymbol{p}$ to be projected onto the tangent plane. This function first rotates the sphere so that the projection point $\boldsymbol{p}$ is sent to the coordinates $(R, 0, 0)$, and then projects the rotated points onto the tangent plane at the point $(R, 0, 0)$ by simply removing the x-coordinate. The rotation matrix that accomplishes this rotation via matrix multiplication is given by

$$\begin{pmatrix} \texttt{orthonormal\_to\_x}(\boldsymbol{p}) \\ \texttt{orthonormal\_to\_y}(\boldsymbol{p}) \\ \texttt{orthonormal\_to\_z}(\boldsymbol{p}) \end{pmatrix},$$

where the three rows are orthonormal and the first row is a unit vector in the direction of the projection point. After this rotation, the function projects onto its last two coordinates and negates the value in the last coordinate. The complete function sphere_to_2D_plane is defined below.

$$
\begin{aligned}
&\texttt{sphere\_to\_2D\_plane}(\boldsymbol{p})(\hat{\boldsymbol{c}}) : \boldsymbol{v} \in \mathbb{R}^2 \equiv \\
&\qquad\qquad \texttt{let}\ \boldsymbol{ymult} = \texttt{orthonormal\_to\_y}(\boldsymbol{p}), \\
&\qquad\qquad\qquad \boldsymbol{zmult} = \texttt{orthonormal\_to\_z}(\boldsymbol{p}) \\
&\qquad\qquad \texttt{in}\ (\boldsymbol{ymult} \cdot \hat{\boldsymbol{c}},\ \ -\boldsymbol{zmult} \cdot \hat{\boldsymbol{c}})
\end{aligned}
$$

The predicate dist_valid?, defined below, depends on the following parameters:

- R - radius of earth
- const - distance endpoints of an edge are allowed to be from a common point (often a projection point)

The predicate dist_valid? is true when, for every pair $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$ of input vectors in $\mathbb{R}^3$ and every point $\boldsymbol{c}$ on the chord between them, the following conditions are all met:

- The point $\boldsymbol{p}$ is on the surface of the sphere of radius R.
- The points $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$ are on the surface of the sphere of radius R and lie in the hemisphere centered at $\boldsymbol{p}$.
- The points $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$ are each at most a distance const from $\boldsymbol{p}$ in the 3D Euclidean norm distance (not the great circle distance).

If these conditions hold, the orthographically projected points sphere_to_2D_plane($\boldsymbol{p}$)($\boldsymbol{c}$) and sphere_to_2D_plane($\boldsymbol{p}$)(lift_to_surface($R, \boldsymbol{c}$)) are within distance DIST of each other. This statement can be interpreted as saying that, for all pairs of points within radius const of the point $\boldsymbol{p}$, the great circle path between these two points is no more than DIST away from the chord between these points after being orthographically projected at $\boldsymbol{p}$.

$$
\begin{aligned}
&\texttt{dist\_valid?}((R, \mathsf{const}, \mathsf{DIST}) \in \mathbb{R}^+,\ \boldsymbol{p}) : \texttt{bool} \equiv \\
&\|\boldsymbol{p}\| = R\ \texttt{and}\ \forall(\boldsymbol{q_1}, \boldsymbol{q_2}, \boldsymbol{c} \neq \boldsymbol{0}) : \\
&\qquad\qquad \boldsymbol{q_1} \cdot \boldsymbol{p} > 0\ \texttt{and}\ \|\boldsymbol{q_1}\| = R\ \texttt{and} \\
&\qquad\qquad \boldsymbol{q_2} \cdot \boldsymbol{p} > 0\ \texttt{and}\ \|\boldsymbol{q_2}\| = R\ \texttt{and} \\
&\qquad\qquad \|(\boldsymbol{q_1} - \boldsymbol{p})\| \leq \mathsf{const}\ \texttt{and}\ \|(\boldsymbol{q_2} - \boldsymbol{p})\| \leq \mathsf{const}\ \texttt{and} \\
&\qquad\qquad \boldsymbol{c} \neq \boldsymbol{0}\ \texttt{and}\ \texttt{on\_chord?}(\boldsymbol{q_1}, \boldsymbol{q_2})(\boldsymbol{c}) \\
&\qquad \Longrightarrow \|(\texttt{sphere\_to\_2D\_plane}(\boldsymbol{p})(\texttt{lift\_to\_surface}(R, \boldsymbol{c})) - \\
&\qquad\qquad\qquad\qquad \texttt{sphere\_to\_2D\_plane}(\boldsymbol{p})(\boldsymbol{c}))\| \leq \mathsf{DIST}
\end{aligned}
$$

The predicate dist_valid? has been proved in PVS to hold for a variety of values for const and DIST. As expected, the length DIST for this which this predicate holds increases significantly as const gets large. That is, the difference between the orthographically projected chord path and the orthographically projected great circle path between two points increases as the distance between these points and the projection point increases. Figure 4 and Table 1 (see appendix) show the relationship between const and DIST.

Each bound in Figure 4 and Table 1 is proved as a lemma in PVS. For example, the following lemma states that the orthographically projected image of the great circle path is no more than 0.0001 meters from the straight line path when const is 1852 meters (1

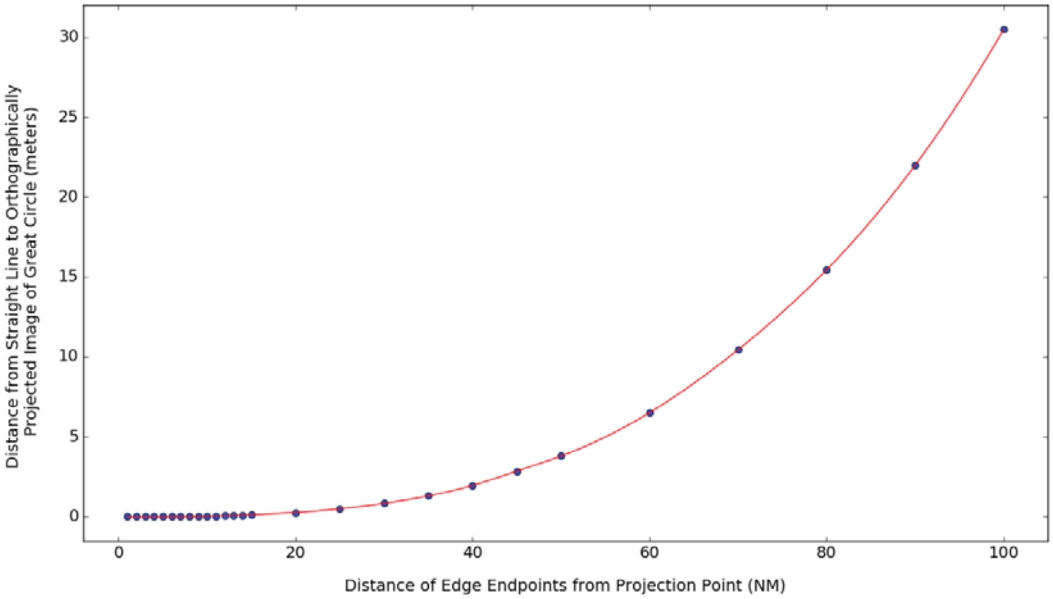

Figure 4: Formally proved maximum variation (meters in the projected plane) in path location between orthographic projection path and great circle vs. distance of endpoints from projection point (nautical miles)

nautical mile), meaning that the endpoints of the path are within 1 nautical mile of the projection point.

$$\begin{array}{l}\texttt{sphere\_to\_2D\_plane\_dist\_1852:}\ \ \texttt{Lemma}\\ \quad \texttt{let}\ \mathsf{const} = 1852,\ \mathsf{DIST} = 0.0001\\ \quad \texttt{in}\ \forall \boldsymbol{p} : \|\boldsymbol{p}\| = R \implies \texttt{dist\_valid?}(\boldsymbol{p},\ R,\ \mathsf{const},\ \mathsf{DIST})\end{array}$$

The results above help to quantify the amount that polygon edge location variations can be reduced by limiting the distance from vertices to the common projection point. Another approach to control the edge location variation introduced by an orthographic projection is to bound the length of each edge by a constant pairdist, in addition to also limiting the distance from vertices to the projection point. The predicate dist_valid_edgebound?, defined below, depends on the same two parameters as dist_valid?, namely R and const, as well as on the following parameter:

- pairdist - maximum distance that two endpoints of an edge are allowed to be from each other.

The predicate dist_valid_edgebound? has the same conditions as dist_valid?, as well as the following additional condition:

- The points $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$ are no more than a fixed distance pairdist from each other in the 3D Euclidean norm distance (not the great circle distance).

If these conditions are met, then dist_valid_edgebound? is true if the orthographically projected point sphere_to_2D_plane($\boldsymbol{p}$)($\boldsymbol{c}$) and the orthographically projected great circle point sphere_to_2D_plane($\boldsymbol{p}$)(lift_to_surface($R, \boldsymbol{c}$)) are within distance DIST of each other,

$$
\begin{aligned}
&\texttt{dist\_valid\_edgebound?}((R, \mathsf{const}, \mathsf{DIST}, \mathsf{pairdist}) \in \mathbb{R}^+,\ \boldsymbol{p}) : \texttt{bool} \equiv\\
&\|\boldsymbol{p}\| = R \ \texttt{ and }\ \forall(\boldsymbol{q_1}, \boldsymbol{q_2}, \boldsymbol{c} \neq \boldsymbol{0}) :\\
&\qquad\qquad \boldsymbol{q_1} \cdot \boldsymbol{p} > 0 \ \texttt{ and }\ \|\boldsymbol{q_1}\| = R,\\
&\qquad\qquad \boldsymbol{q_2} \cdot \boldsymbol{p} > 0 \ \texttt{ and }\ \|\boldsymbol{q_2}\| = R,\\
&\qquad\qquad \|(\boldsymbol{q_1} - \boldsymbol{p})\| \leq \mathsf{const} \ \texttt{ and }\ \|(\boldsymbol{q_2} - \boldsymbol{p})\| \leq \mathsf{const},\\
&\qquad\qquad \|(\boldsymbol{q_1} - \boldsymbol{q_2})\| \leq \mathsf{pairdist},\\
&\qquad\qquad \boldsymbol{c} \neq \boldsymbol{0} \ \texttt{ and }\ \texttt{on\_chord?}(\boldsymbol{q_1}, \boldsymbol{q_2})(\boldsymbol{c})\\
&\qquad \Longrightarrow \|(\texttt{sphere\_to\_2D\_plane}(\boldsymbol{p})(\texttt{lift\_to\_surface}(R, \boldsymbol{c})) -\\
&\qquad\qquad\qquad\qquad \texttt{sphere\_to\_2D\_plane}(\boldsymbol{p})(\boldsymbol{c}))\| \leq \mathsf{DIST}.
\end{aligned}
$$

This predicate has been proved in PVS to hold for the values $\mathsf{const} = 100$ NM, $\mathsf{pairdist} = 15$ NM, and $\mathsf{DIST} = 0.44$ m. This means that the images under orthographic projection of the great circle path and the straight line between two endpoints are less than 0.44 meters apart when the endpoints of the path are within 15 NM of each other and 100 NM of the projection point. This assumes a spherical model of the earth with radius 6370997 meters.

$$
\begin{aligned}
&\texttt{sphere\_to\_2D\_plane\_dist\_185200\_27780:}\quad \texttt{Lemma}\\
&\quad \texttt{let}\ \ R = 6370997, \mathsf{const} = 185200, \mathsf{DIST} = 0.44, \mathsf{pairdist} = 27780\\
&\quad \texttt{in}\ \forall \boldsymbol{p} \in \mathbb{R}^3 : \|\boldsymbol{p}\| = R \implies\\
&\qquad \texttt{dist\_valid\_edgebound?}(\boldsymbol{p}, R, \mathsf{const}, \mathsf{DIST}, \mathsf{pairdist})
\end{aligned}
$$

For many geofencing applications, one meter of inaccuracy introduced by an orthographic map projection is acceptable. The above lemma implies that two potential requirements on such a system that may help mitigate the effects of the orthographic projection are that no polygon vertex is more than 100 NM from the common projection point and that no edge is more than 15 NM long.

5 Estimated Differences in Polygon Edge Locations for Various Projections

This section under-estimates the answer to the following question, under various constraints: If a geofencing algorithm uses a map projection to determine whether a position on the WGS84 ellipsoid is inside a polygon region whose vertices are also points on the WGS84 ellipsoid, how far outside/inside the polygon can the point be and the algorithm determine that it is inside/out (the opposite and therefore incorrect answer)? That is, this section provides under-estimates for the variations in locations of polygon edges due to different map projections and spherical earth models. The analysis in this section uses Python implementations of orthographic, Mercator and Plate Carrée (Equirectangular) projections, as well as geodesics on the sphere and the WGS84 ellipsoid. Given two endpoints of a polygon edge as inputs, each of these implementations generates a path between the endpoints. This section estimates the distance between each such path and the ellipsoidal geodesic path. The WGS84 ellipsoid and the geodesic between edge endpoints (vertices) are therefore used as an oracle of the "correct" location of each edge.

There are five path generating functions implemented in Python called `Ortho`, `Merc`, `Equi_Rec`, `Great_Circle`, and `WGS84_geo`. Each of these five functions takes as inputs:

1. A latitude-longitude pair ($\texttt{lat}_p, \texttt{lon}_p$) representing a projection point,

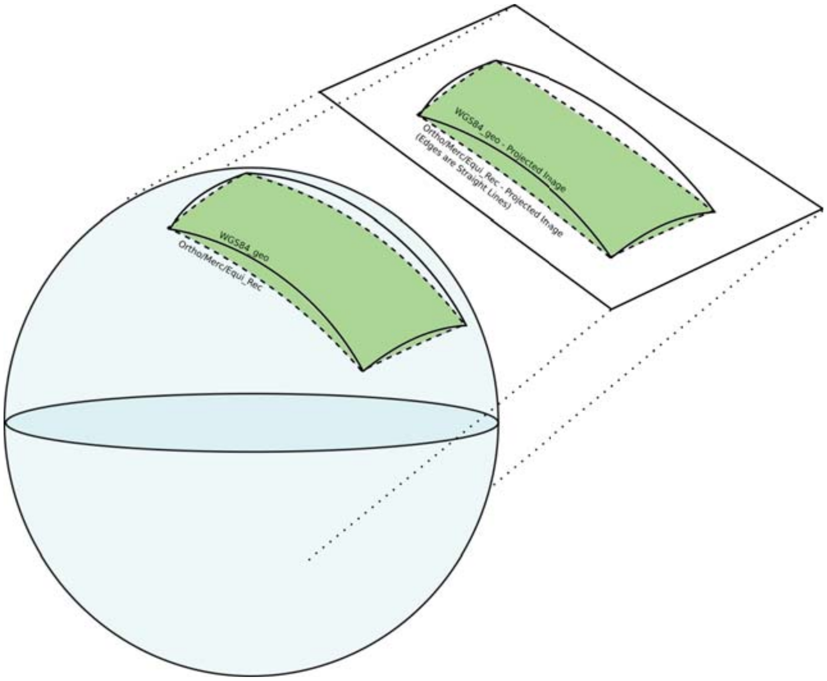
WGS84_geo
Ortho/Merc/Equi_Rec
WGS84_geo - Projected Image
Ortho/Merc/Equi_Rec - Projected Image
(Edges are Straight Lines)

Figure 5: Polygon region with sides given by a projection function (`Ortho`, `Merc`, or `Equi_Rec`)

2. Two latitude-longitude pairs ($\texttt{lat}_1, \texttt{lon}_1$) and ($\texttt{lat}_2, \texttt{lon}_2$) representing endpoints of a polygon edge, and

3. A scalar t in the closed interval $[0, 1]$.

Each of these five functions returns a latitude-longitude pair. As the parameter t moves from 0 to 1, each function can be seen as producing a sequence of latitude-longitude pairs that move from the pair ($\texttt{lat}_1, \texttt{lon}_1$) to the pair ($\texttt{lat}_2, \texttt{lon}_2$). At $t = 0$, it returns ($\texttt{lat}_1, \texttt{lon}_1$), and at $t = 1$, it returns ($\texttt{lat}_2, \texttt{lon}_2$). The `Great_Circle` and `WGS84_geo` functions return the latitude-longitude pairs of the geodesic paths on the sphere and ellipsoid, respectively.

Figure 5 shows that the functions `Equi_Rec`, `Merc`, and `Ortho` compute the boundary paths of the polygon in their respective projective planes. In that figure, the *true* polygon is shown with a solid boundary whose curves are WGS84 geodesics, and the region whose boundary is computed by these three projection functions is shown with a dotted boundary. Figure 6 shows the *true* polygon (in green) on the surface of the WGS84 ellipsoid, with a solid boundary. In that figure, as in Figure 5, the region with curves given by these three projection functions is also shown with a dotted boundary. Given one of the corresponding projections, the endpoints of a polygon edge are projected to a plane. The straight line between the projected images of the endpoints determines whether a point near the edge is either inside or outside. The functions `Equi_Rec`, `Merc`, and `Ortho`, respectively, compute the pre-image (before projection) of this straight line. They therefore compute the boundary paths (as latitude-longitude pairs) of the collection of points that the containment algorithm says are inside. Because they return latitude-longitude pairs, they can be viewed as paths on the surface of the WGS84 ellipsoid.

This Section computes various distances between the paths `Ortho`, `Merc`, `Equi_Rec`, `Great_Circle`, and `WGS84_geo`. A closeup of the paths on the surface of the WGS84 ellipsoid in Figure 5 are shown in Figure 7. The great circle path given by `Great_Circle`, which returns latitude-longitude pairs and can therefore also be plotted on the surface of the

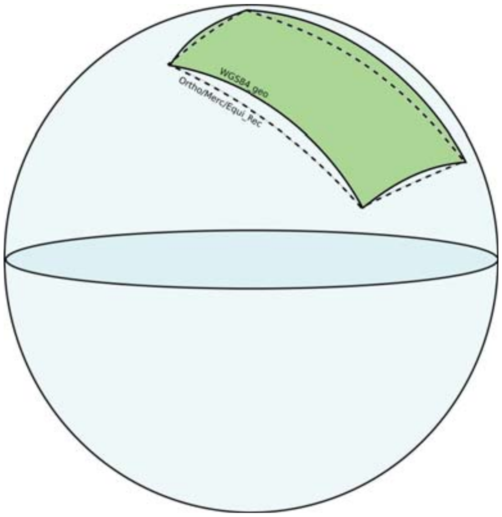

Figure 6: Polygon region with WGS84 geodesic sides

WGS84 ellipsoid, is also shown in Figure 8. The distance between the paths given by the WGS84 geodesic and the respective projection function is shown in red. The distance between the great circle and the projection function is shown in blue. The distance between the WGS84 geodesic and the great circle path is shown in green.

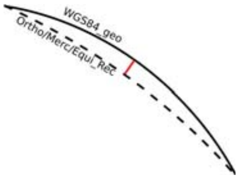

Figure 7: Closeup of two paths

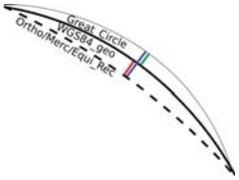

Figure 8: Closeup of three paths

Each of the four functions `Great_Circle`, `WGS84_geo`, `Equi_Rec`, and `Merc` uses standard, well-known formulas to compute paths. The functions `Great_Circle` and `WGS84_geo` both use Python implementations of Karney's direct and inverse formulas [5] which are called directly from the GeographicLib package [6]. The Python implementation of `Equi_Rec` uses the formula defined in Section 3 for the Plate Carrée projection, and the implementation of `Merc` uses standard formulas.

The other path-generating function, `Ortho`, is defined in terms of its associated PVS specifications. The calls to `orthog_project` produce orthographic projections of $(\texttt{lat}_1, \texttt{lon}_1)$ and $(\texttt{lat}_2, \texttt{lon}_2)$ from the navigation sphere, where $(\texttt{lat}_p, \texttt{lon}_p)$ is the point of projection. The call to `orthog_inverse` produces the inverse image of the point $(\texttt{x}_u, \texttt{y}_u)$ =

$(1-t)\cdot(\mathtt{lat}_1, \mathtt{lon}_1) + t\cdot(\mathtt{lat}_2, \mathtt{lon}_2)$ from the projected plane back to the sphere.

```
$\mathtt{Ortho}(\mathtt{lat}_p, \mathtt{lon}_p, \mathtt{lat}_1, \mathtt{lon}_1, \mathtt{lat}_2, \mathtt{lon}_2, t) \equiv$
  $\mathtt{q}_1\mathtt{\_plane} = \mathtt{orthog\_project}(R, \mathtt{lat}_p, \mathtt{lon}_p, \mathtt{lat}_1, \mathtt{lon}_1)$
  $\mathtt{q}_2\mathtt{\_plane} = \mathtt{orthog\_project}(R, \mathtt{lat}_p, \mathtt{lon}_p, \mathtt{lat}_2, \mathtt{lon}_2)$
  $\mathtt{x}_u = (1-t)\cdot \mathtt{q}_1\mathtt{\_plane.x} + t\cdot \mathtt{q}_2\mathtt{\_plane.x}$
  $\mathtt{y}_u = (1-t)\cdot \mathtt{q}_1\mathtt{\_plane.y} + t\cdot \mathtt{q}_2\mathtt{\_plane.y}$
  $\mathtt{u\_plane\_projback} = \mathtt{orthog\_inverse}(R, \mathtt{lat}_p, \mathtt{lon}_p, \mathtt{x}_u, \mathtt{y}_u)$
  return u_plane_projback
```

Note that the correctness of the data presented in Section 5 depends on two things, namely the author's Python implementations of the functions `Ortho`, `Merc`, and `Equi_Rec`, and the Python implementations of Karney's formulas in the Geographiclib package [6]. In contrast, the correctness of the data presented in Section 4 depends only on one thing, namely the PVS implementation of orthographic projections, and the numerical bounds are formally proved in PVS.

The analyses in this section depends on the parameters R, const, and pardist, as defined in Section 4. Each of the functions described above for generating an edge between polygon vertices is compared with the WGS84 ellipsoidal geodesic using the following approach:

1. A grid of projection points $(\mathtt{lat}_p, \mathtt{lon}_p)$, where each point is a latitude-longitude pair, is generated covering the surface of the earth. In this implementation, there are 555 projection points generated, consisting of 15 points for each of 37 different evenly spaced latitudes.

2. A distance of const is designated from each projection point $(\mathtt{lat}_p, \mathtt{lon}_p)$. Only edges with endpoints that are at distance const from $(\mathtt{lat}_p, \mathtt{lon}_p)$ are considered in this approach.

3. For each projection point $(\mathtt{lat}_p, \mathtt{lon}_p)$, 60 pairs $(\mathtt{lat}_1, \mathtt{lon}_1), (\mathtt{lat}_2, \mathtt{lon}_2)$ on the surface of the WGS84 ellipsoid are randomly generated, with each point at an exact distance const from $(\mathtt{lat}_p, \mathtt{lon}_p)$ as stated in Step 2. For Figures 14 and 15 and Tables 3 and 4, each pair is generated with a distance approximately pairdist from each other. For the other tables and graphs, no such constraint is placed on the distance between pairwise points.

4. The four points

$$\mathtt{Great\_Circle}(\mathtt{lat}_p, \mathtt{lon}_p, \mathtt{lat}_1, \mathtt{lon}_1, \mathtt{lat}_2, \mathtt{lon}_2, t),$$
$$\mathtt{Ortho}(\mathtt{lat}_p, \mathtt{lon}_p, \mathtt{lat}_1, \mathtt{lon}_1, \mathtt{lat}_2, \mathtt{lon}_2, t),$$
$$\mathtt{Merc}(\mathtt{lat}_p, \mathtt{lon}_p, \mathtt{lat}_1, \mathtt{lon}_1, \mathtt{lat}_2, \mathtt{lon}_2, t), \text{ and}$$
$$\mathtt{Equi\_Rec}(\mathtt{lat}_p, \mathtt{lon}_p, \mathtt{lat}_1, \mathtt{lon}_1, \mathtt{lat}_2, \mathtt{lon}_2, t)$$

   are computed on the WGS84 ellipsoid, and their WGS84 geodesic distance to the point $\mathtt{WGS84\_geo}(\mathtt{lat}_p, \mathtt{lon}_p, \mathtt{lat}_1, \mathtt{lon}_1, \mathtt{lat}_2, \mathtt{lon}_2, t)$ is calculated. These four points correspond to points on a great circle, as well as orthographic, Mercator, and Plate Carrée projections. This calculation is done for 11 different evenly spaced values of t.

5. For each path function described above (`Great_Circle`, `Ortho`, `Merc`, and `Equi_Rec`), the maximum of the distance calculated in step 4, over all pairs generated in step 3, is calculated and is used as an estimate of the maximum distance from the geodesic to the given path.

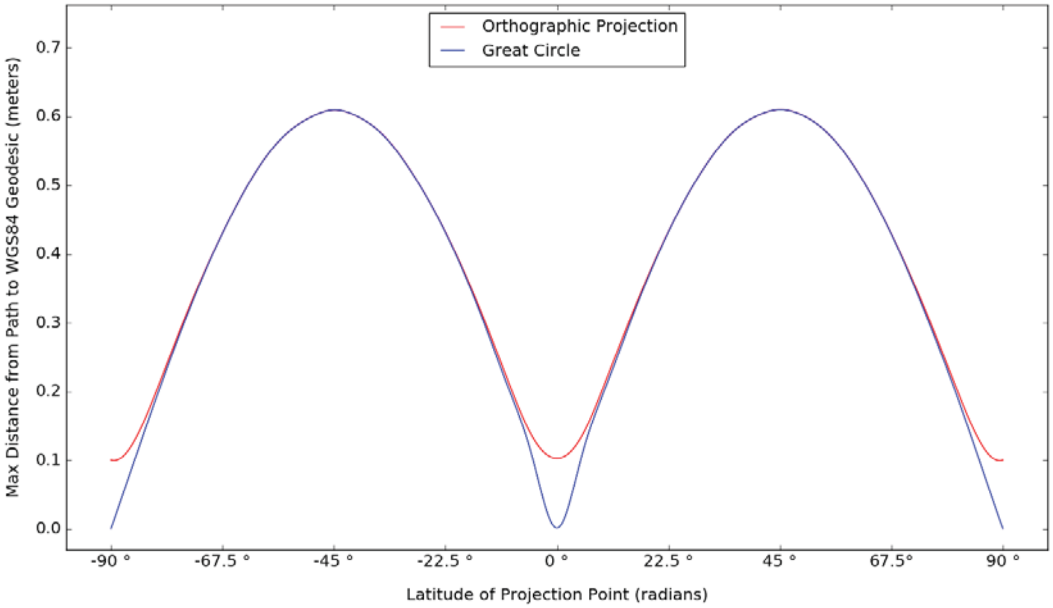

Figure 9: Estimated distance (meters) of WGS84 geodesic path to great circle and orthographic projection paths by latitude (radians) of projection point when each endpoint of the path is no more than 15NM from the projection point

Outcomes of specific analyses are described below.

Figure 9 displays dependence on the latitude (of the projection point) of the distance between the WGS84 ellipsoid geodesic and the path functions `Great_Circle` (blue) and `Ortho` (red), for `const` equals 15 nautical miles, meaning that the endpoints of the paths are each a distance of 15 NM from the projection point. The figure represents path distances for projection points at the prime meridian and a range of latitudes between south and north poles. A fixed longitude was chosen because the difference in output is mathematically independent of longitude. The results show that great circle paths and edges generated through orthographic projections differ most from WGS84 geodesics at latitude $\pi/4$. Further, this graph illustrates that for edges with endpoints no more than 15 NM from the projection point, distances from great circle and orthographic projection paths to WGS84 geodesics are of comparable magnitudes, except at the poles and equator.

Figure 10 displays a similar analysis for Mercator (green) and Plate Carrée (orange) projections. It depicts dependence on the latitude of the projection point for the distance between the WGS84 ellipsoid geodesic and the path functions `Merc` and `Equi_Rec`, where `const` is set to 15 NM, again meaning that the endpoints of the paths are each 15 NM from the projection point. The figure shows that for latitudes far from the equator, the Mercator and Plate Carrée projections produce edge paths at enormous distances from WGS84 ellipsoid geodesics. This result suggests that making use of these projections for geofencing far from the equator produces unreliable outcomes if polygons have large edges (e.g. if both endpoints are more than 15 nautical miles from some common point).

The graphs discussed above emphasize errors for multiple points over a range of latitudes. Figure 11 shows the distance to the WGS84 geodesic for two edges generated by the path functions `Great_Circle` (blue) and `Ortho` (red), respectively, where the parameter t for these functions moves from 0 to 1. As expected, the distance to the geodesic is largest around $t = 1/2$, the middle of the path between the endpoints. This graph shows this distance for a single projection point $\boldsymbol{p}$ corresponding to the latitude-longitude pair $(\frac{\pi}{4}, \frac{\pi}{4})$ (a latitude-

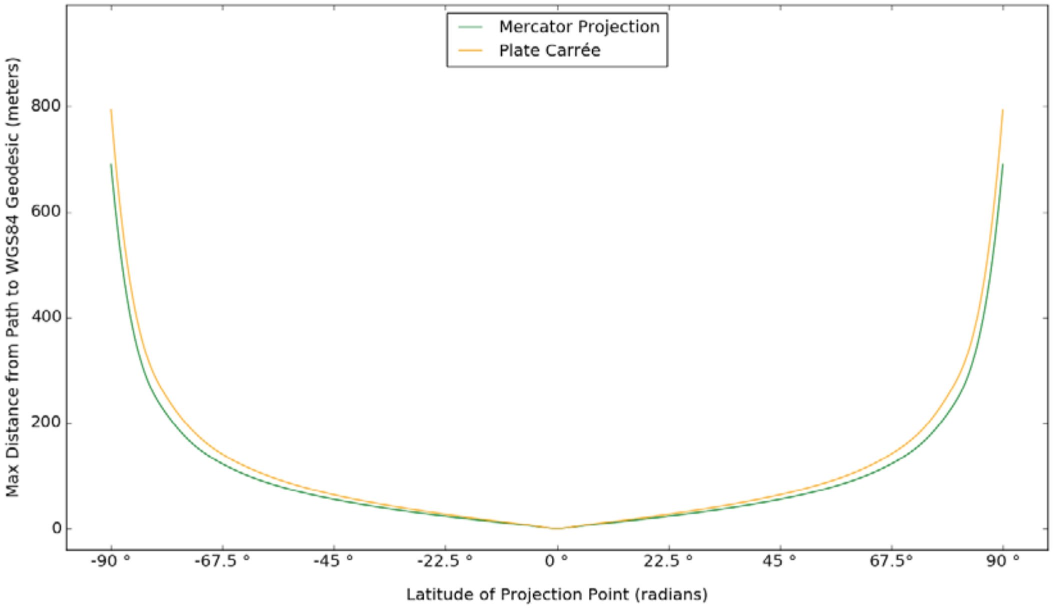

Figure 10: Estimated distance (meters) of WGS84 geodesic path to Mercator and Plate Carrée projection paths by latitude (radians) of a point that is no more than 15NM from each endpoint of the path

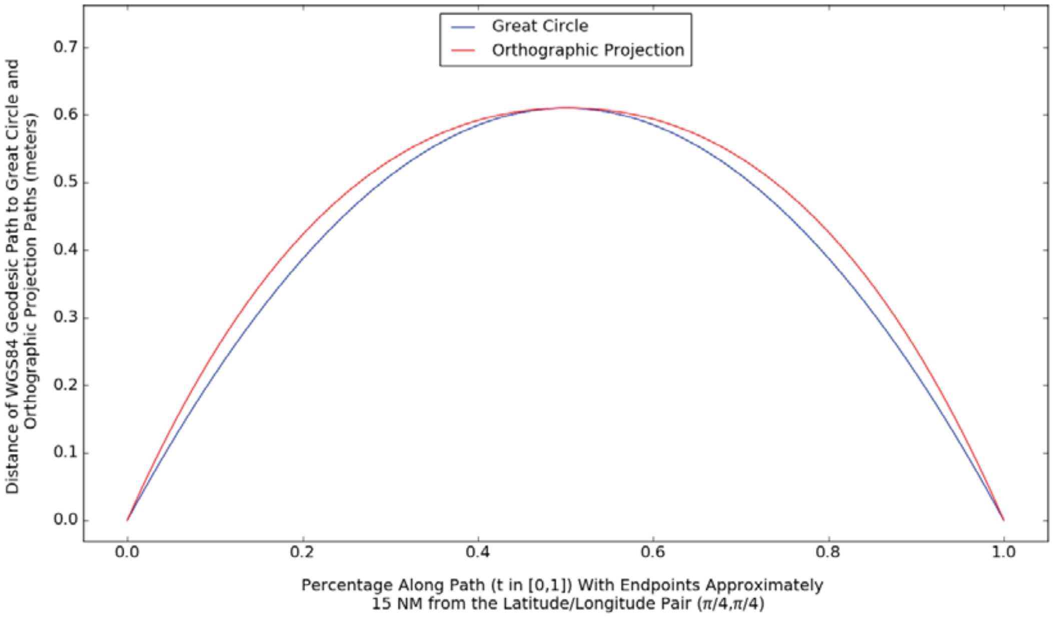

Figure 11: Estimated distance (meters) of WGS84 geodesic path to great circle and orthographic projection paths versus percentage along path ($t \in [0, 1]$) when each endpoint of the path is no more than 15NM from the projection point

longitude pair) and a pair of endpoints 15 nautical miles away from the projection point.

Figure 12 is analogous to Figure 11, except that it displays the output for the functions `Merc` and `Equi_Rec`, corresponding to Mercator (green) and Plate Carrée (orange) projec-

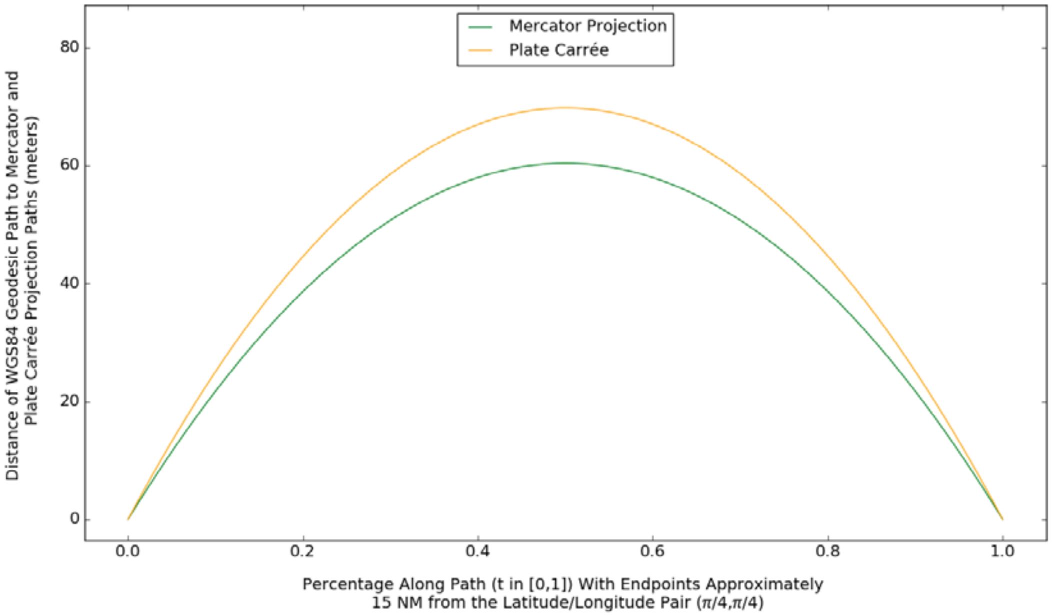

Figure 12: Estimated distance (meters) of WGS84 geodesic path to Mercator and Plate Carrée projection paths versus percentage along path ($t \in [0, 1]$) when each endpoint of the path is no more than 15NM from the projection point

tions. As in Figure 11, for Figure 12 the endpoints of the edge are located 15 nautical miles away from the projection point $\boldsymbol{p}$ corresponding to the latitude-longitude pair $(\frac{\pi}{4}, \frac{\pi}{4})$.

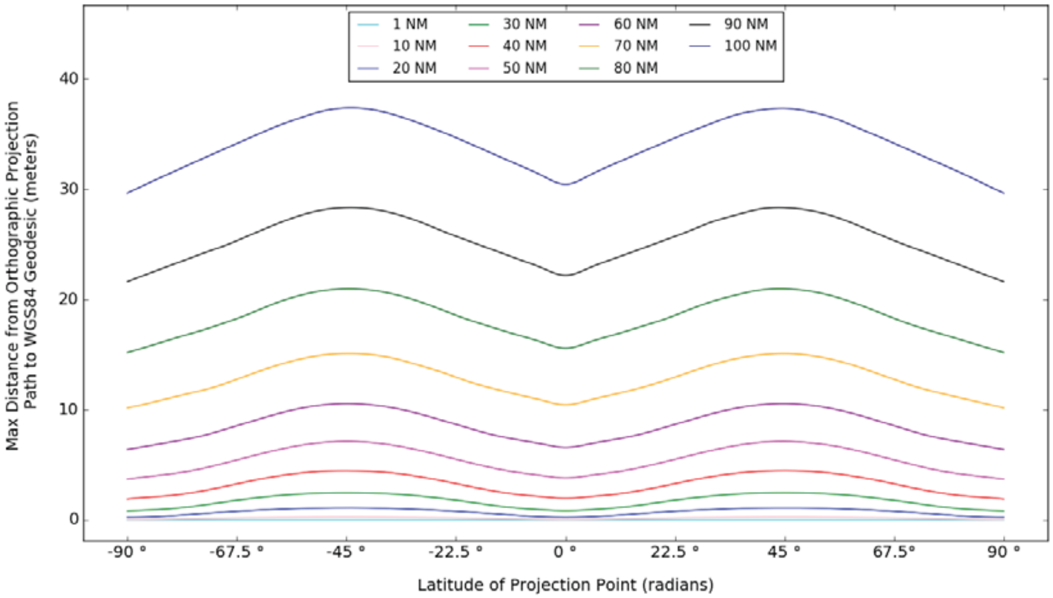

Figure 13: Estimated distance (meters) of orthographic projection path to the great circle path versus latitude (radians) of the projection point for increasing distances (NM) (shown in different colors) between the endpoints and the projection point

One important aspect of Figures 9 through 12 is that they compute distances between paths with the WGS84 geodesics as the oracle. However, the PVS development described in Section 4 bounds distances between the images of the path-generating functions `Ortho` and `Great_Circle` under an orthographic projection. As a complement to this work carried out in PVS, Figure 13 illustrates the results of evaluating the distance between an orthographic projection and a great circle on the surface of a sphere, where the great circle is the oracle, using the functions `Ortho` and `Great_Circle`. The figure shows the distance between these function outputs for edges of increasing distance (const) from the projection point. The smallest value of const, which is one nautical mile, results in almost zero difference between the functions `Ortho` and `Great_Circle`. The figure continues in multiples of 10 up to 100 nautical miles, in which the largest distance between paths of approximately 36 meters occurs. As mentioned above, these values are taken over a range of latitudes and a fixed longitude at the prime meridian.

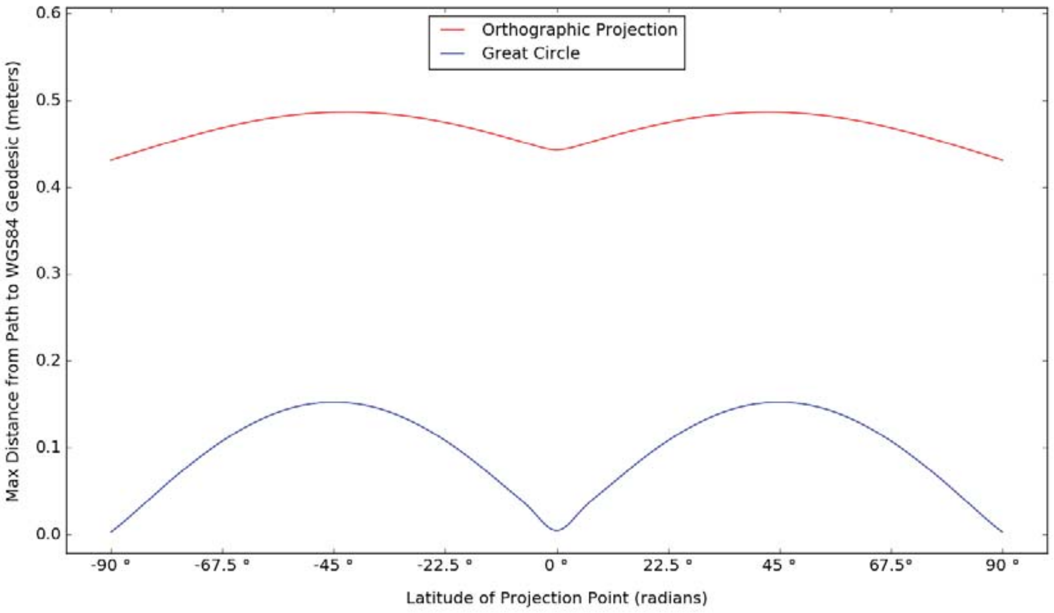

Figure 14: Estimated distance of WGS84 geodesic path to great circle and orthographic projection paths versus latitude (radians) of projection point with endpoints no more than 100NM from the projection point and edges no more than 15NM long

Figures 9 through 13 place constraints on the maximum distance (named const) that each endpoint of an edge can be from the projection point, but they place no other restrictions on the length of the edge. The data in Figure 14 depends also on the parameter pairdist, which is the maximum length of an edge. This figure displays the distance between paths generated by the functions `Ortho` and `Great_Circle` when the endpoint vertices $(\mathtt{lat}_1, \mathtt{lon}_1)$ and $(\mathtt{lat}_2, \mathtt{lon}_2)$ are at a fixed distance pairdist $= 15$ NM from each other and a distance of const $= 100$ NM from the projection point. The longitude of the projection point $\boldsymbol{p}$ in Figure 14 is taken to be the prime meridian. Figure 14 suggests that an orthographic projection still produces an edge that is within half a meter of the geodesic on the WGS84 ellipsoid, even if the vertices are 100 NM from the projection point, as long as the distance between adjacent vertices is less than 15 NM. Tables 3 and 4 show estimated edge location variation distances for orthographic and Plate Carrée projections, respectively, over a variety of values of pairdist and const.

Figure 15 shows the distance to the WGS84 geodesic for a single example pair of endpoints near a common projection point. In this example, the edge endpoints are pairdist $= 15$

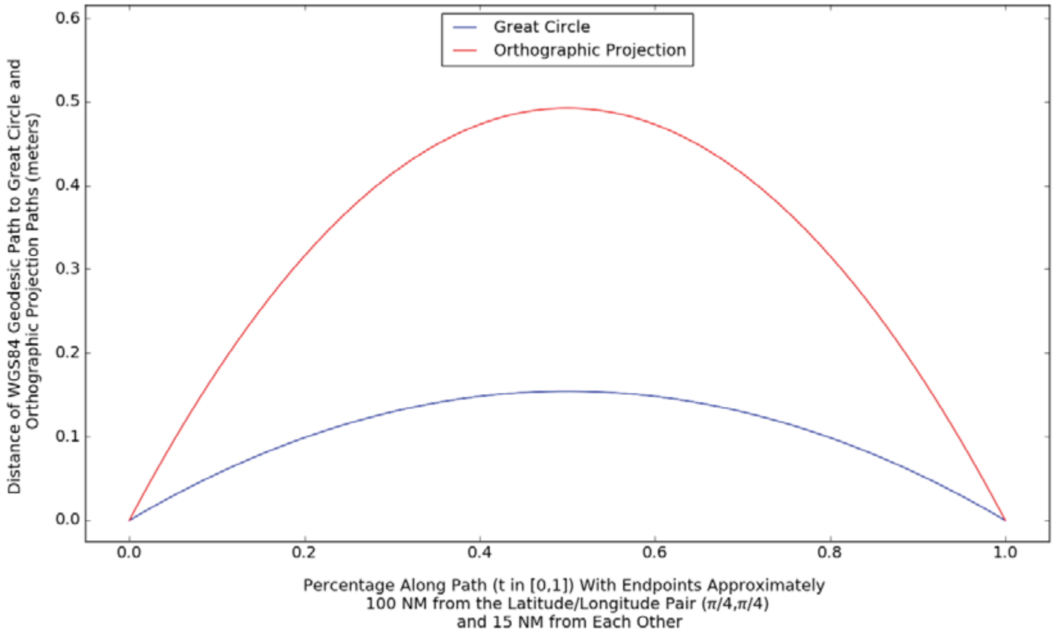

Figure 15: Estimated distance (meters) of WGS84 geodesic path to great circle and orthographic projection paths versus percentage along path ($t \in [0, 1]$) when each endpoint of the path is no more than 100NM from the projection point and the path is approximately 15NM long

NM apart. Paths are generated by the functions `Great_Circle` and `Ortho`. This graph shows this distance to the geodesic as the percentage along each path increases. The projection point used has latitude-longitude coordinates given by the pair $(\frac{\pi}{4}, \frac{\pi}{4})$, and the endpoints are approximately 100 NM apart.

Table 2 in the Appendix shows the distance between the orthographic projection path function `Ortho` and the functions `Great_Circle` and `WGS84_geo` at the midpoints of the paths. The distance is measured on the surface of the WGS84 ellipsoid using a geodesic.

6 Related Work

The study of the effects of map projections on UAS geofencing is not new. In their paper entitled *Towards a generic and modular geofencing strategy for civilian UAVs*, authors Gurriet and Ciarletta discuss the distance between an ellipsoidal geodesic and straight lines on Mercator and Equirectangular projections. For their geofencing algorithm, the Mercator projection was taken to be the projection of choice due to its continued prominence [4]. One difference between their work and this paper is that they do not provide explicit estimates of bounds intended to be valid for all inputs on the earth's surface. In contrast, one purpose of the current paper is to provide esimates of such bounds so that readers can choose appropriate constraints on geofence sizes, depending on their position uncertainty tolerances. Finally, Gurriet and Ciarletta depend on Mercator projections, while the analysis carried out in the previous section is focused more on orthographic projections.

The professional paper entitled *Map Projections - A Working Manual*, by John P. Snyder of the United States Geological Survey [9], is an excellent resource on map projections. In addition to descriptions of most commonly used projections, it includes data on various metrics related to distortions introduced by map projections.

7 Conclusion

This paper estimates location variations in the sides of geofencing polygons induced by map projections. There are multiple paths between two polygon vertices on the surface of the earth, including great circles, WGS84 ellipsoid geodesics, and paths generated by straight lines after a map projection has taken place. This paper estimates how far apart these paths can be. Their path lengths are *not* compared. Rather, the comparision is between the actual locations of the edges between vertices.

This paper presents formally verified upper bounds for the distance between a great circle path between vertices and the curve derived from a straight line path between the images of the vertices after an orthographic projection to a tangent plane is taken. These bounds have been verified with PVS for a few specific scenarios, such as when the vertices are each within 15 nautical miles of the projection point. The proved bounds are distances in the projected plane, rather than on the surface of the earth. For vertices with distances less than 30 nautical miles from the projection point, the distances between great circle paths and straight lines in the projected plane are less than one meter, and for vertices less than 5 nautical miles from the projection point, the variation is negligible (< 0.004 meters). This paper also presents Python-generated data on the location variations of sides of polygons. Multiple edge-generation methods are examined, including great circles, orthographic projections, Mercator projections, and Plate Carrée Equirectangular projections. This data is calculated using available Python packages for the projections and great circle functions, and its correctness therefore depends on the correctness of those available packages, in particular the GeographicLib package [6]. The bounds generated in Python are estimates, and in particular, they have not been formally proved in PVS in the sense of the bounds mentioned above. As mentioned in Section 4, one meter of inaccuracy introduced by an orthographic map projection is acceptable for many geofencing applications. Given this, two potential requirements on such a system that may help mitigate the distortion effects of the orthographic projection are that no polygon vertex is more than 100 NM from the common projection point and that no edge is more than 15 NM long.

References

1. María Consiglio, César Muñoz, George Hagen, Anthony Narkawicz, and Swee Balachandran. ICAROUS: Integrated Configurable Algorithms for Reliable Operations of Unmanned Systems. In *Proceedings of the 35th Digital Avionics Systems Conference (DASC 2016)*, Sacramento, California, US, September 2016.

2. Evan T. Dill, Steven D. Young, and Kelly J. Hayhurst. SAFEGUARD: An assured safety net technology for UAS. *Proceedings of the 35th Digital Avionics Systems Conference (DASC 2016)*, September 2016.

3. Michael A. Earle. Sphere to spheroid comparisons. *Journal of Navigation*, 59(3):491496, 2006. doi:10.1017/S0373463306003845.

4. T. Gurriet and L. Ciarletta. Towards a generic and modular geofencing strategy for civilian UAVs. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 540–549, June 2016. doi:10.1109/ICUAS.2016.7502603.

5. Charles F. F. Karney. Algorithms for geodesics. *Journal of Geodesy*, 87(1):43–55, Jan 2013. URL: https://doi.org/10.1007/s00190-012-0578-z, doi:10.1007/s00190-012-0578-z.

6. Charles F. F. Karney. The geodesic routines from geographiclib. Apr 2017. URL: https://geographiclib.sourceforge.io/1.48/python/.

7. Anthony Narkawicz, César Muñoz, and Aaron Dutle. The MINERVA software development process. In *Proceedings of the Workshop on Automated Formal Methods 2017 (AFM 2017)*, Meno Park, California, USA, 2017.

8. John P. Snyder and Philip M. Voxland. An album of map projections U.S. Geological Survey professional paper 1453. *US Government Printing Office*, 1989.

9. John Parr Snyder. *Map projections–A working manual*, volume 1395. US Government Printing Office, 1987.

8 Appendix: Tables

PVS Upper Bounds for Difference Between Projected Image of Great Circle and Chord Paths	
const (nautical miles) - i.e. max distance of endpoints to projection point	DIST (meters) - i.e. edge location variation
1	0.0001
2	0.0003
3	0.0009
4	0.002
5	0.004
6	0.007
7	0.011
8	0.016
9	0.022
10	0.031
11	0.041
12	0.053
13	0.067
14	0.083
15	0.11
20	0.25
25	0.49
30	0.83
35	1.3
40	1.95
45	2.85
50	3.79
60	6.51
70	10.45
80	15.45
90	22
100	30.5

Table 1: PVS Proved Maximum Euclidean 2D Distance Between Straight Line in Orthographic Projection Plane and Projected Image of the Great Circle

Python Estimates for Difference Between Orthographic Projection and Great Circle Paths on the Sphere		
`const` - i.e. max distance (nautical miles) of endpoints to projection point	Distance (meters) of Path from Orthog. Projection to Great Circle	Distance (meters) of Path from Orthog. Projection to WGS84 Ellipsoid Geodesic
10	0.0303	0.2584
15	0.1021	0.5814
20	0.2420	1.0336
25	0.4727	1.6148
30	0.8168	2.3259
35	1.2971	3.1659
40	1.9362	4.1350
45	2.7569	5.2351
50	3.7818	6.5289
60	6.5350	9.8685
70	10.3775	14.3656
80	15.4910	20.2109
90	22.0570	27.5935
100	30.2574	36.7027

Table 2: Estimated Distance of Path from Othographic Projection to the Great Circle and WGS84 Ellipsoid Geodesic Paths

Python Estimates for Difference Between Orthographic Projection and WGS84 Geodesic Paths for Fixed Polygon Edge Length pairdist and Max Distance of Endpoints to Projection Point const											
		pairdist									
		0.25NM	0.5NM	1NM	2NM	5NM	10NM	15NM	25NM	50NM	100NM
const	1NM	4.04 e-5m	0.00016m	0.0006m							
	5NM		0.00016m	0.00066m	0.0026m	0.0166m					
	10NM			0.00069m	0.0027m	0.017m	0.068m	0.149m			
	25NM				0.0033m	0.0207m	0.0823m	0.184m	0.495m		
	50NM					0.0306m	0.122m	0.273m	0.75m	2.77m	
	75NM						0.168m	0.377m	1.042m	4.013m	13.32m
	100NM						0.22m	0.487m	1.348m	5.27m	19.115m

Table 3: Estimated Distance of Path from Orthographic Projection and WGS84 Ellipsoid Geodesic, by Distance const of the Endpoints to the Projection Point

Python Estimates for Difference Between Plate Carrée Projection and WGS84 Geodesic Paths for Fixed Polygon Edge Length pairdist and Max Distance of Endpoints to Projection Point const									
		pairdist							
		0.25NM	0.5NM	1NM	2NM	5NM	10NM	15NM	25NM
const	1NM	0.0084m	0.034m	0.137m					
	5NM		0.035m	0.137m	0.547m	3.38m			
	10NM			0.138m	0.543m	3.382m	13.55m	30.41m	
	25NM				0.55m	3.4m	13.69m	30.851m	85m
	50NM					3.46m	13.8m	31.05m	86.37m
	75NM						14m	31.54m	87.98m
	100NM						14.21m	31.96m	88.9m

Table 4: Estimated Distance of Path from Plate-Carrée Projection to WGS84 Ellipsoid Geodesic, by Distance const of Endpoints to a Common Point

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD-MM-YYYY)* 01-10-2017	**2. REPORT TYPE** Technical Memorandum	**3. DATES COVERED** *(From - To)*

4. TITLE AND SUBTITLE
Map Projection Induced Variations in Locations of Polygon Geofence Edges

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Paula Neeley and Anthony Narkawicz

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
411931-02-51-07-01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, Virginia 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER
L–20883

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)
NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
NASA/TM–2017–219675

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category 03
Availability: NASA STI Program (757)864-9658

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This Paper under-estimates answers to the following question under various constraints: If a geofencing algorithm uses a map projection to determine whether a position is inside/outside a polygon region, how far outside/inside the polygon can the point be and the algorithm determine that it is inside/outside (the opposite and therefore incorrect answer)? Geofencing systems for unmanned aircraft systems (UAS) often model stay-in and stay-out regions using 2D polygons with minimum and maximum altitudes. The vertices of the polygons are typically input as latitude-longitude pairs, and the edges as paths between adjacent vertices. There are numerous ways to generate these paths, resulting in numerous potential locations for the edges of stay-in and stay-out regions. These paths may be geodesics on a spherical model of the earth or geodesics on the WGS84 reference ellipsoid. In geofencing applications that use map projections, these paths are inverse images of straight lines in the projected plane. This projected plane may be a projection of a spherical earth model onto a tangent plane, called an orthographic projection. Alternatively, it may be a projection where the straight lines in the projected plane correspond to straight lines in the latitude-longitude coordinate system, also called a Plate Carr´ee projection.

15.

Geofencing; Safety

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			STI Help Desk (email: help@sti.nasa.gov
U	U	U	UU	28	**19b. TELEPHONE NUMBER** ***(Include area code)*** 757) 864-9658