Analysis Methods in Atmospheric and Oceanic Science

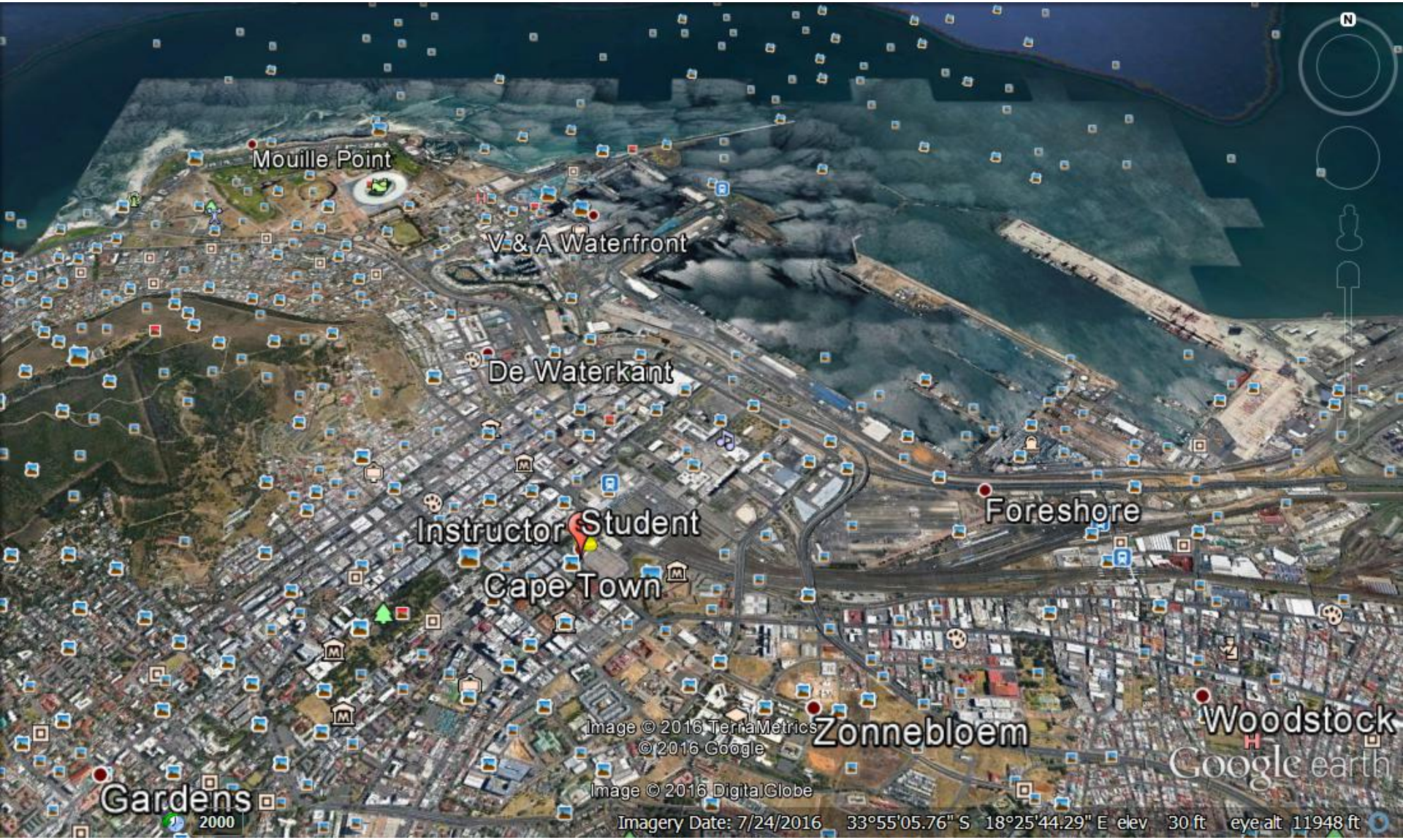
AOSC 652

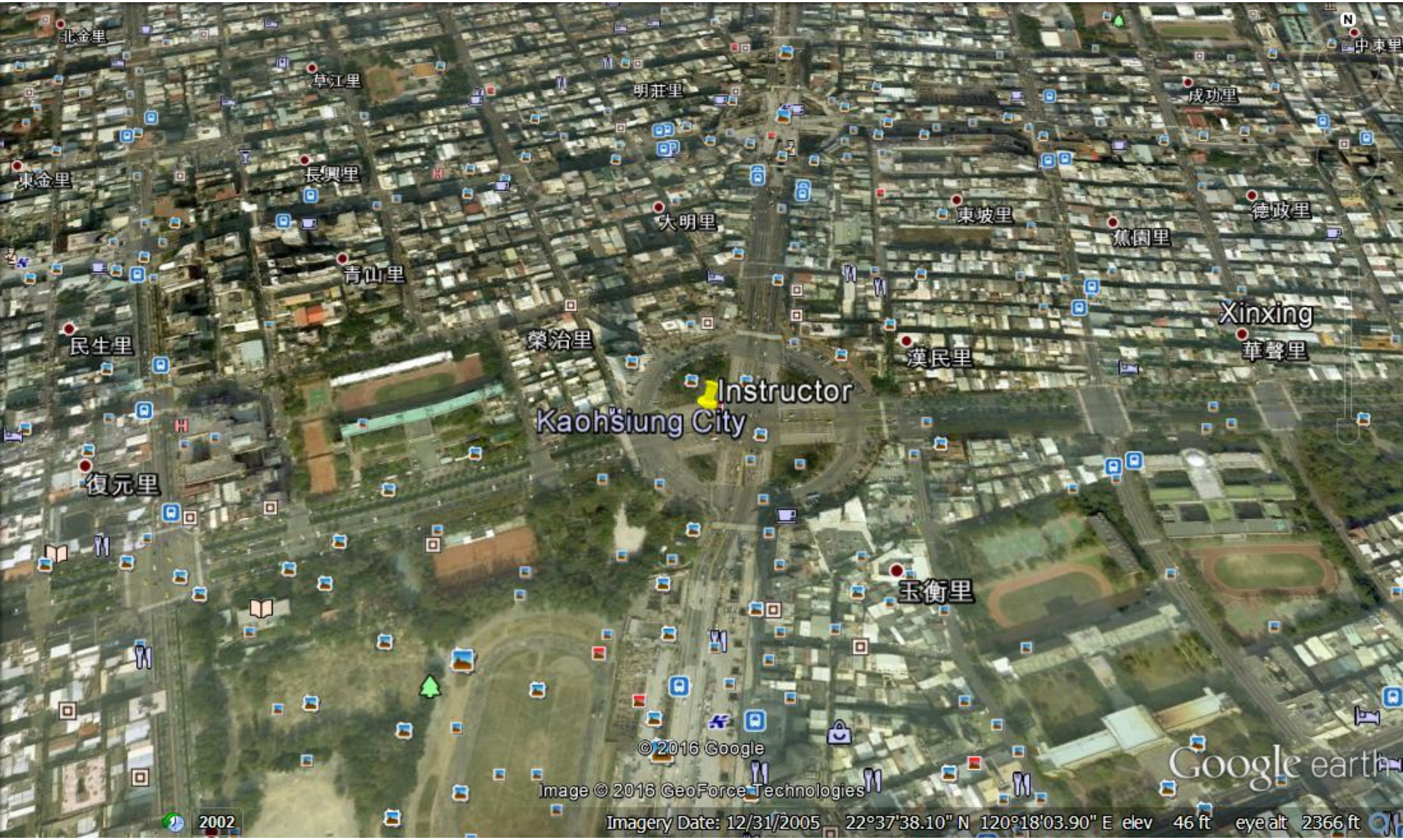
Introduction to Graphics and Analysis of Satellite Measurements of Atmospheric Composition: Day 3

- **Review prior assignment**
- **Guidance with plotting and use of Open Office Impress**
- **General help with HW #3**

16 Sep 2016


10/2014
1943
2015
Instructor
N
Google earth
1943
Imagery Date: 8/29/2014 43°46'23.42" N 11°15'21.55" E elev 159 ft eye alt 2322 ft


Mouille Point
V & A Waterfront
De Waterkant
Instructor Student
Cape Town
Foreshore
Zonnebloem
Woodstock
Gardens
Image © 2016 TerraMetrics
© 2016 Google
Image © 2016 DigitalGlobe
Google earth
2000
Imagery Date: 7/24/2016 33°55'05.76" S 18°25'44.29" E elev 30 ft eye alt 11948 ft


北金里
草江里
明莊里
中東里
成功里
東金里
長興里
大明里
東波里
德政里
蕉園里
青山里
民生里
榮治里
漢民里
Xinxing
華聲里
Instructor
Kaohsiung City
復元里
玉衡里
© 2016 Google
Image © 2016 GeoForce Technologies
Google earth
2002
Imagery Date: 12/31/2005 22°37'38.10" N 120°18'03.90" E elev 46 ft eye alt 2366 ft


4/29/2013
N
Instructor
Image © 2016 DigitalGlobe
Google earth
2004
Imagery Date: 4/29/2013 22°32'40.90" N 88°20'35.02" E elev 35 ft eye alt 1083 ft


6/2013
Instructor
Student
N
Google earth
1994
Imagery Date: 6/8/2013 44°27'35.07" N 110°49'37.42" W elev 7378 ft eye alt 1408 ft

AOSC 652: HW 02

How many sig figs needed to specify target for 0.2 km accuracy?

At equator:

Radius of Earth $\approx$ 6370 km and Earth is represented by 360 deg of longitude

360 deg of longitude $\Rightarrow$ 2π x Radius of Earth = 2 π 6370 km = 40,024 km

Therefore, 1 deg = 111 km

AOSC 652: HW 02

How many sig figs needed to specify target for 0.2 km accuracy?

At equator:

Radius of Earth $\approx$ 6370 km and Earth is represented by 360 deg of longitude

360 deg of longitude $\Rightarrow$ 2π x Radius of Earth = 2 π 6370 km = 40,024 km

Therefore, 1 deg = 111 km

Also, 1 deg = π /180 deg = 0.017 radians

AOSC 652: HW 02

How many sig figs needed to specify target for 0.2 km accuracy?

At equator:

Radius of Earth $\approx$ 6370 km and Earth is represented by 360 deg of longitude

360 deg of longitude $\Rightarrow$ 2π x Radius of Earth = 2 π 6370 km = 40,024 km

Therefore, 1 deg = 111 km

Also, 1 deg = π /180 deg = 0.017 radians

So, 0.017 radians = 111 km & one can show 0.2 km = 3.1×10^{-5} radians

AOSC 652: HW 02

How many sig figs needed to specify target for 0.2 km accuracy?

At equator:

Radius of Earth $\approx$ 6370 km and Earth is represented by 360 deg of longitude

360 deg of longitude $\Rightarrow$ 2π x Radius of Earth = 2 π 6370 km = 40,024 km

Therefore, 1 deg = 111 km

Also, 1 deg = π /180 deg = 0.017 radians

So, 0.017 radians = 111 km & one can show 0.2 km = 3.1×10^{-5} radians

At least 5 sig figs after decimal point are needed to represent 0.000031 radians

6 sig figs are recommended because we want our numerical value to be better than 0.2 km accuracy or, as a student wrote, “better safe than sorry”

More than 6 sig figs after decimal point is TMI

AOSC 652: HW 02

How many sig figs needed to specify target for 0.2 km accuracy?

At equator:

Radius of Earth $\approx$ 6370 km and Earth is represented by 360 deg of longitude

360 deg of longitude $\Rightarrow$ 2π x Radius of Earth = 2 π 6370 km = 40,024 km

Therefore, 1 deg = 111 km

Also, 1 deg = π /180 deg = 0.017 radians

So, 0.017 radians = 111 km & one can show 0.2 km = 3.1×10^{-5} radians

At least 5 sig figs after decimal point are needed to represent 0.000031 radians

6 sig figs are recommended because we want our numerical value to be better than 0.2 km accuracy or, as a student wrote, "better safe than sorry"

More than 6 sig figs after decimal point is TMI

At 60°N:

Distance covered by longitude = 2π x Radius x cos (60°)

0.2 km = {2 π radians/(2 π 6370 km x 0.5)} x 0.2 km = 6.2×10^{-5} radians

Latitude: <u>same as at equator</u>

AOSC 652: HW 02

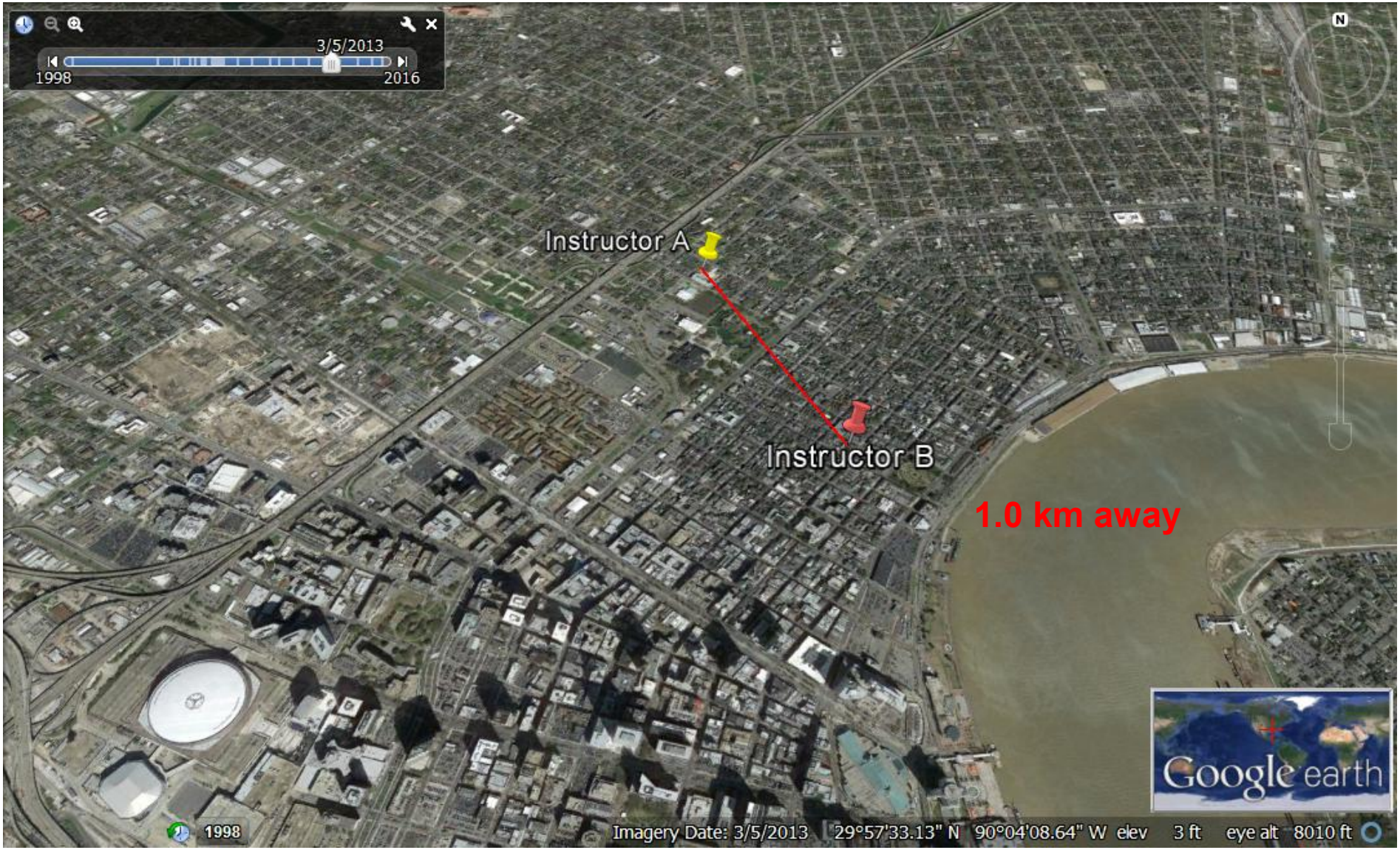
For geographic calculations using Cartesian coordinates, longitude in the Western Hemisphere is preceded by a negative sign

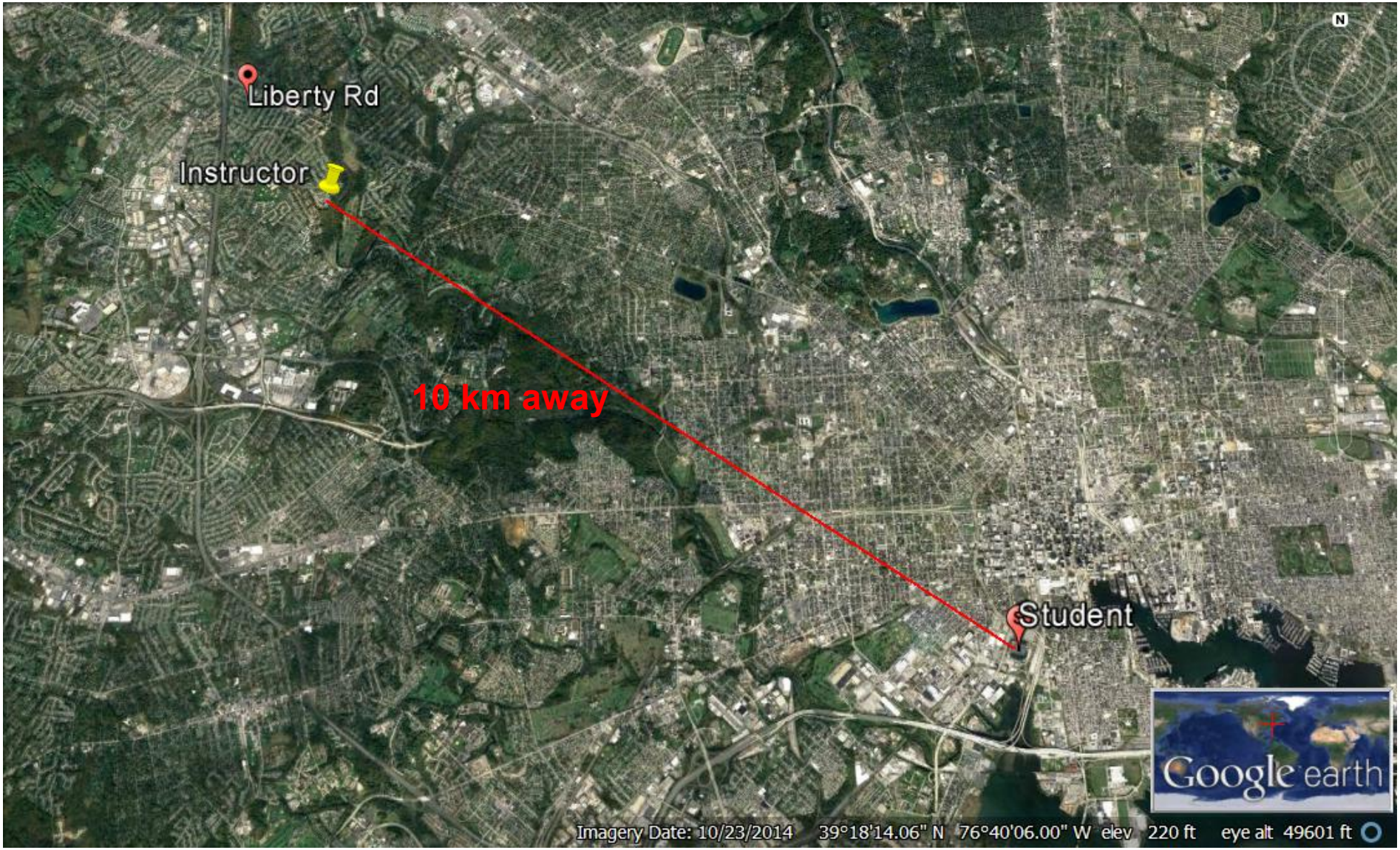
If you specify location, should use:

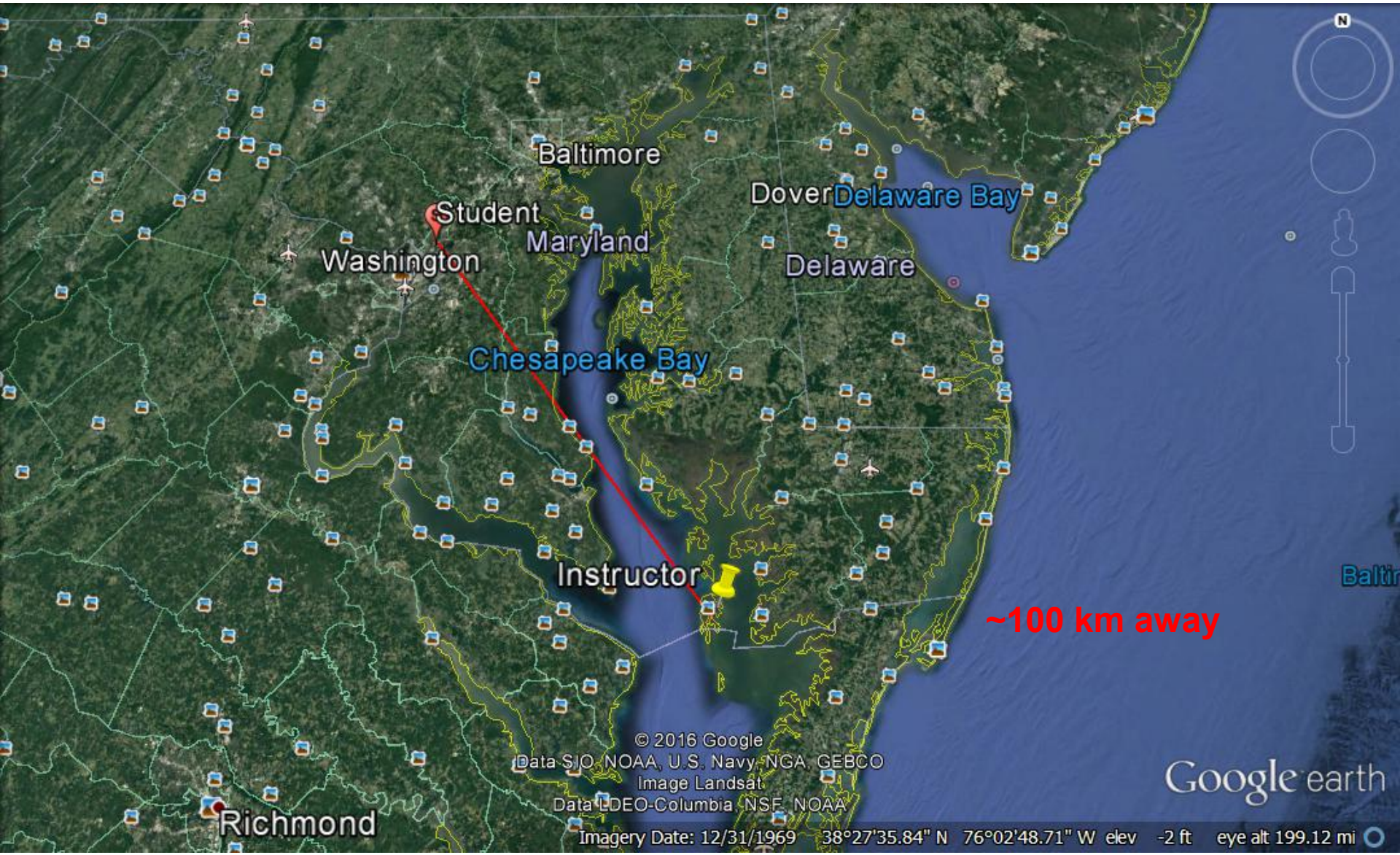
N/S or +/– for latitude
E/W or +/– for longitude

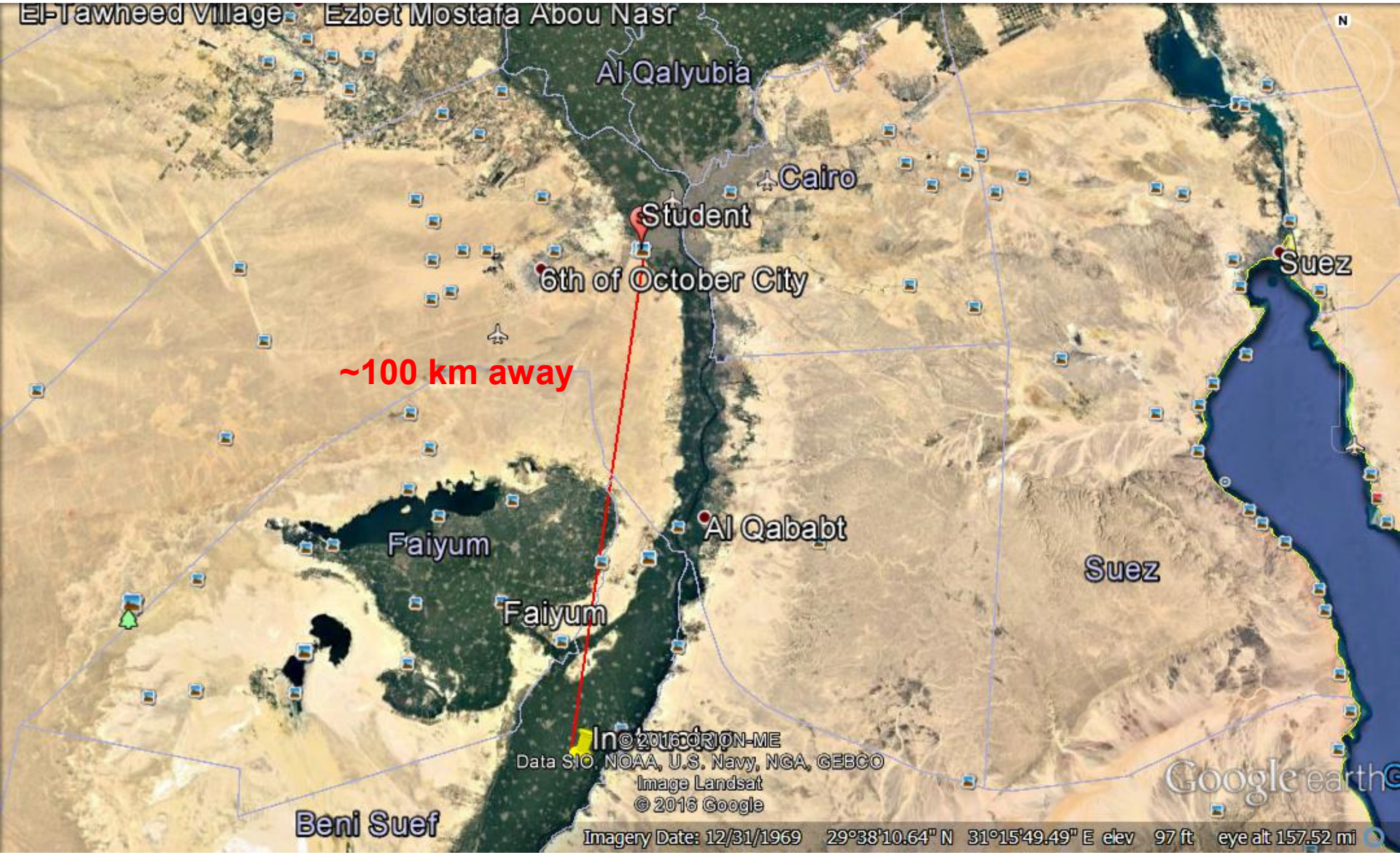
http://en.wikipedia.org/wiki/Longitude

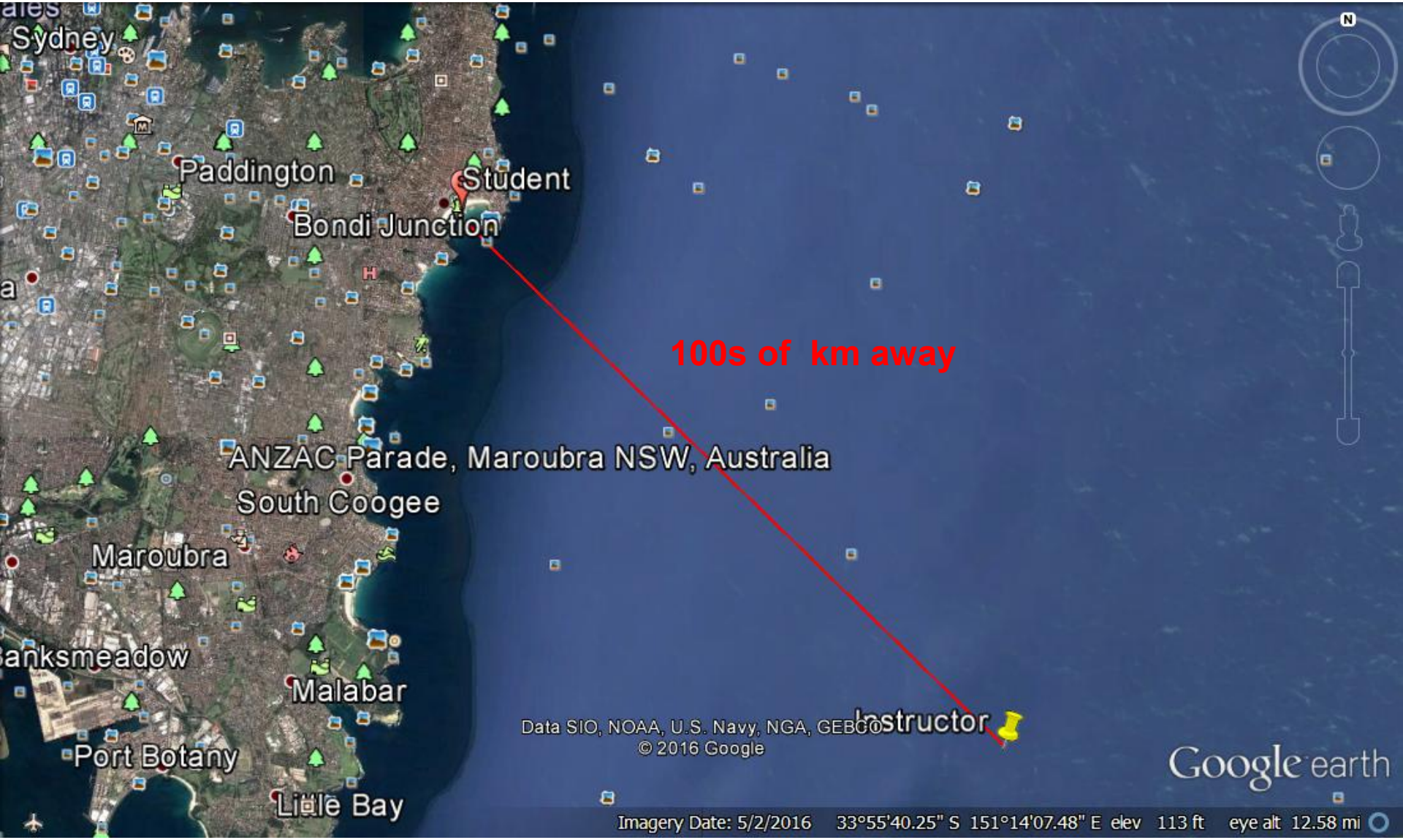

Student
Instructor
0.22 km away
© 2016 Google
Google earth
1994
Imagery Date: 10/23/2014 39°00'09.81" N 76°27'24.90" W elev 0 ft eye alt 1307 ft

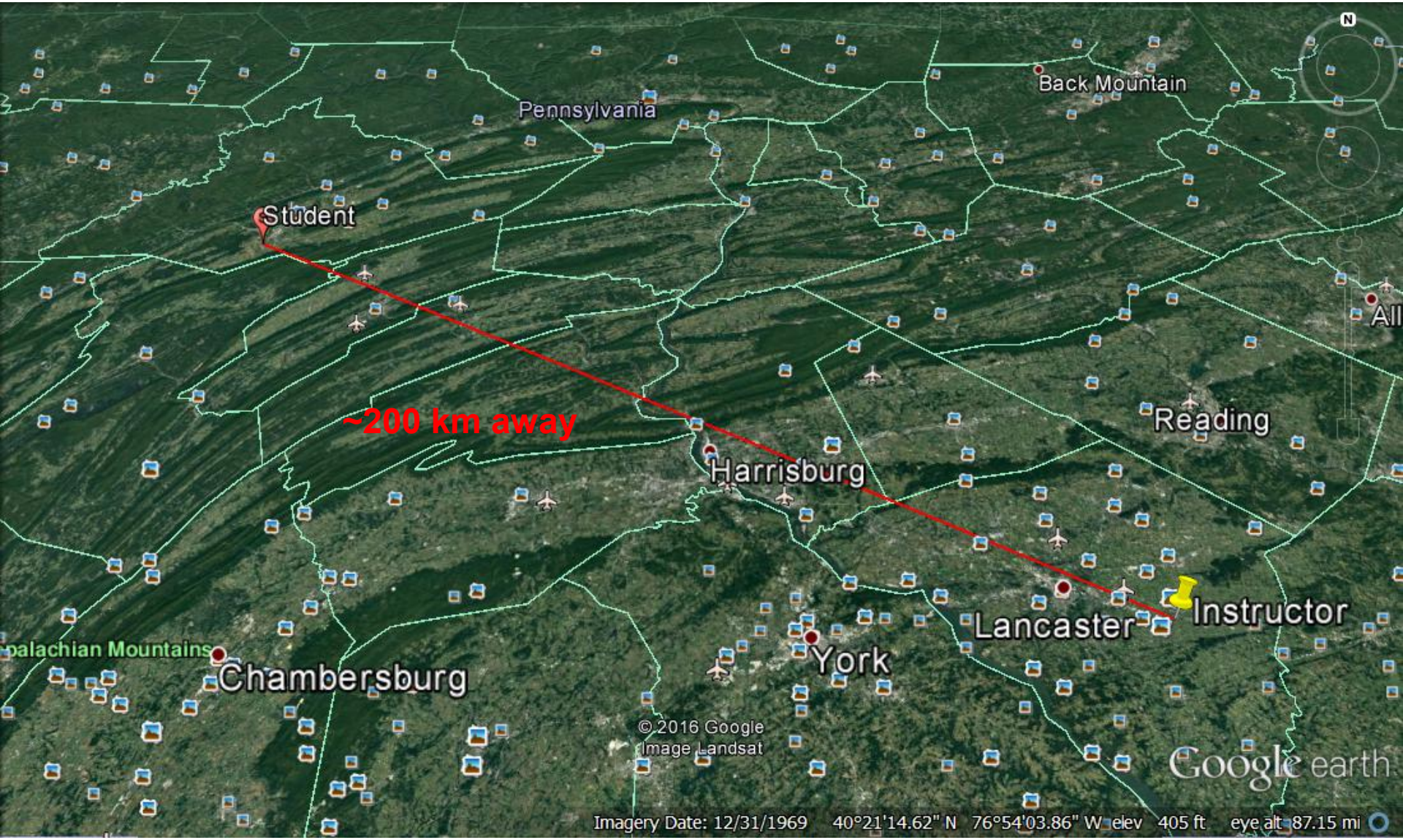

3/5/2013
1998
2016
Instructor A
Instructor B
1.0 km away
N
Google earth
1998
Imagery Date: 3/5/2013 29°57'33.13" N 90°04'08.64" W elev 3 ft eye alt 8010 ft

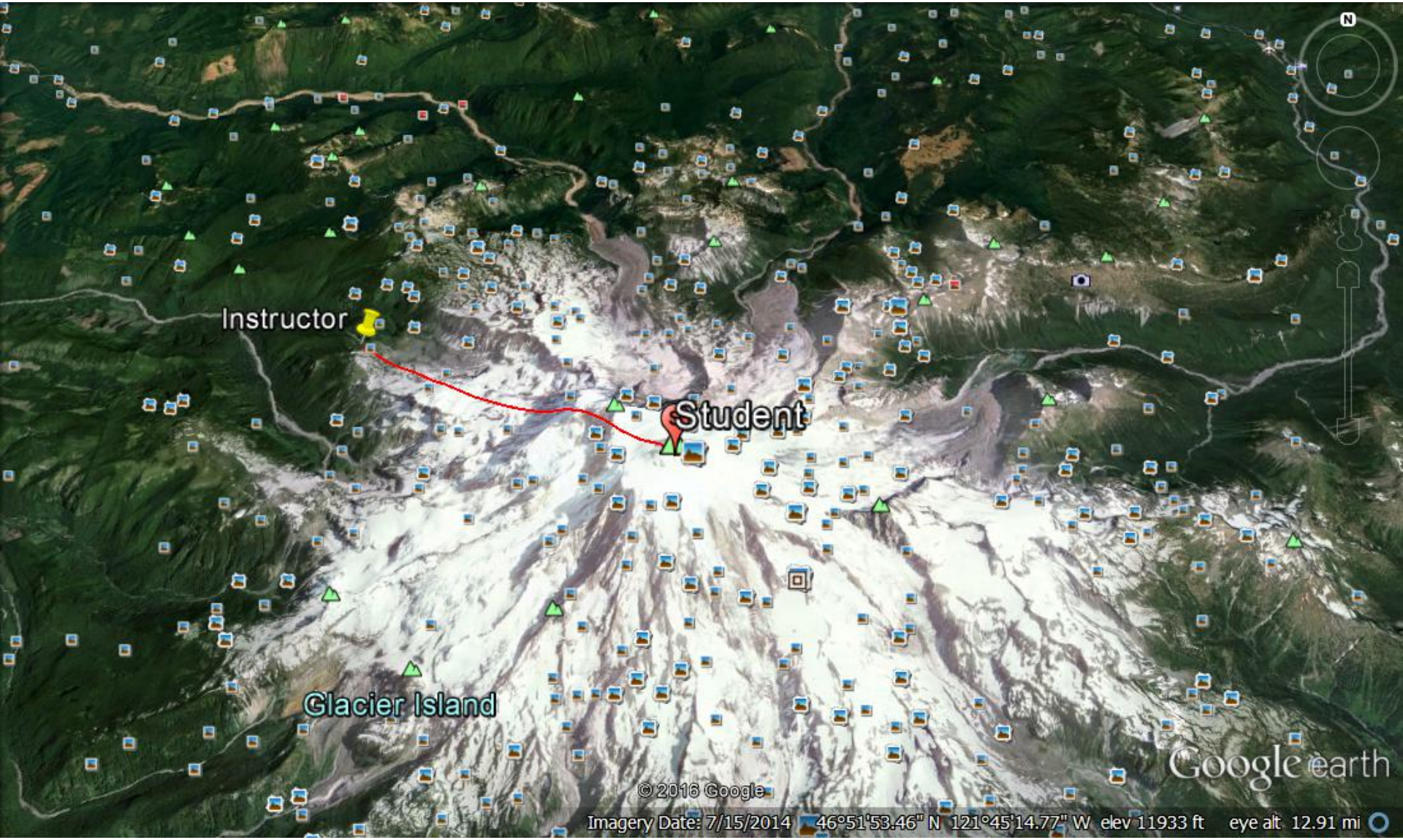

Liberty Rd
Instructor
10 km away
Student
Google earth
Imagery Date: 10/23/2014 39°18'14.06" N 76°40'06.00" W elev 220 ft eye alt 49601 ft

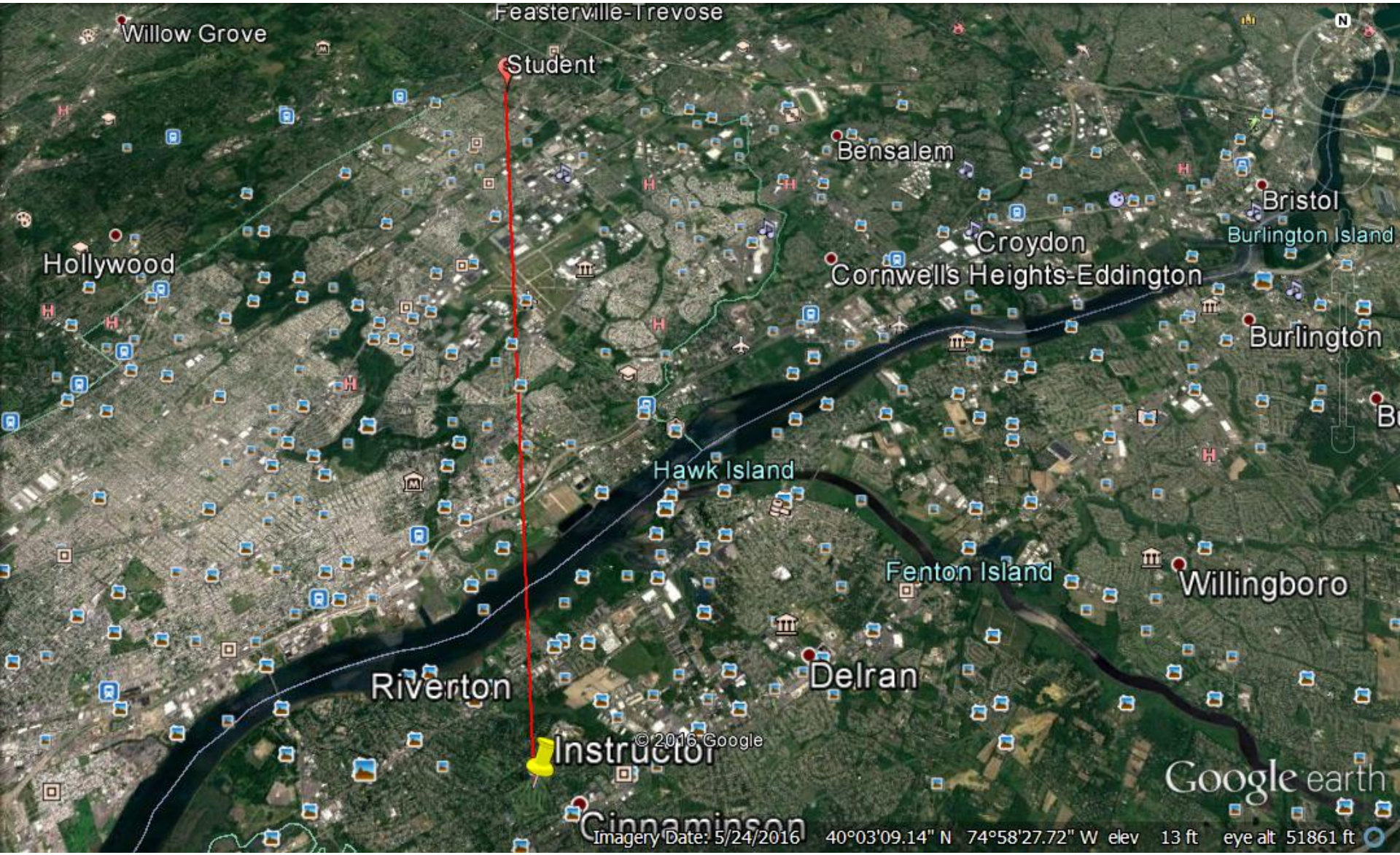

Baltimore
Dover
Delaware Bay
Student
Maryland
Washington
Delaware
Chesapeake Bay
Instructor
~100 km away
Richmond
© 2016 Google
Data SIO, NOAA, U.S. Navy, NGA, GEBCO
Image Landsat
Data LDEO-Columbia, NSF, NOAA
Google earth
Imagery Date: 12/31/1969 38°27'35.84" N 76°02'48.71" W elev -2 ft eye alt 199.12 mi


El-Tawheed Village
Ezbet Mostafa Abou Nasr
Al Qalyubia
Cairo
Student
6th of October City
Suez
~100 km away
Faiyum
Al Qababt
Suez
Faiyum
Beni Suef
Data SIO, NOAA, U.S. Navy, NGA, GEBCO
Image Landsat
© 2016 Google
Google earth
Imagery Date: 12/31/1969 29°38'10.64" N 31°15'49.49" E elev 97 ft eye alt 157.52 mi

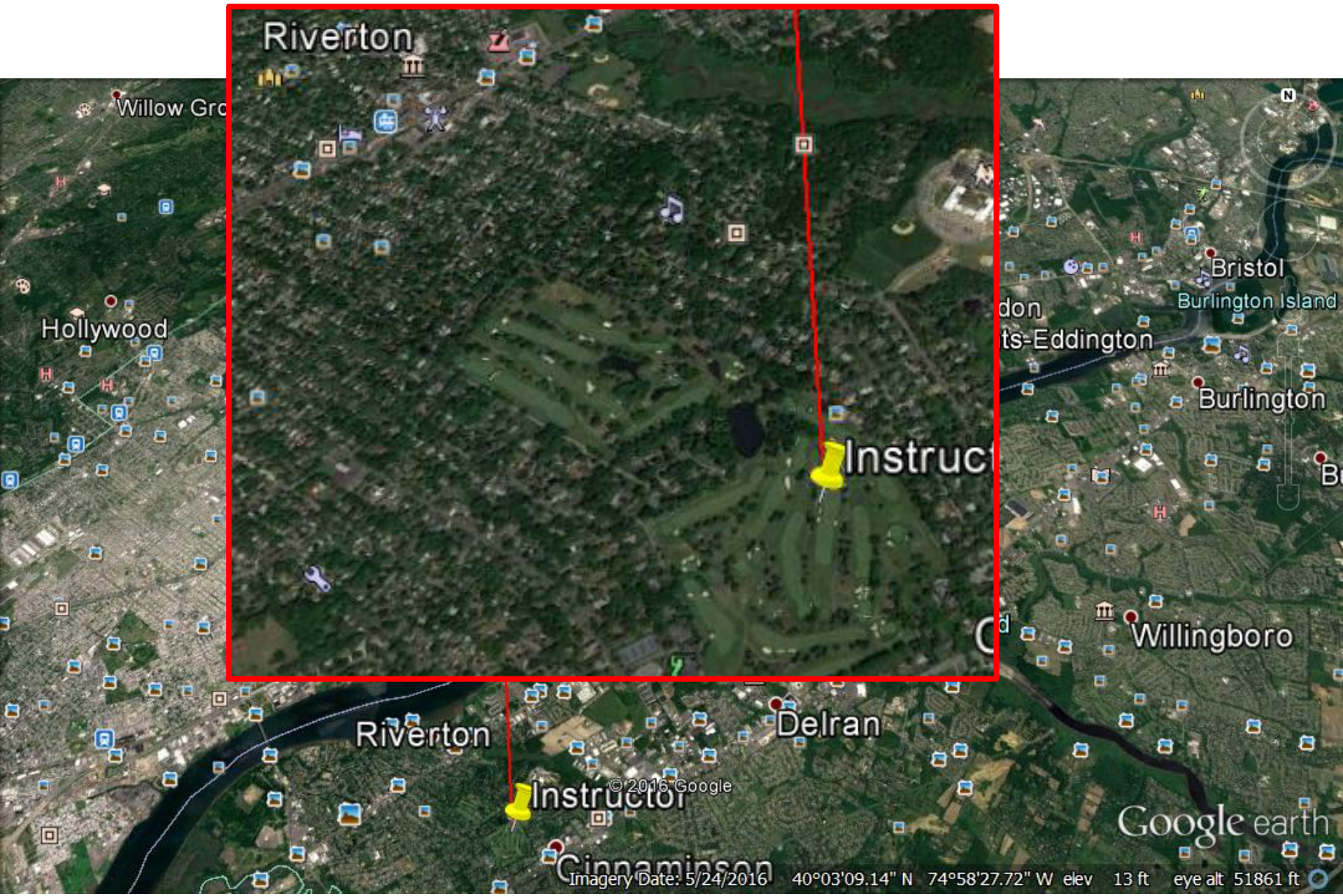

Sydney
Paddington
Student
Bondi Junction
100s of km away
ANZAC Parade, Maroubra NSW, Australia
South Coogee
Maroubra
Malabar
Port Botany
Instructor
Data SIO, NOAA, U.S. Navy, NGA, GEBCO
© 2016 Google
Google earth
Imagery Date: 5/2/2016 33°55'40.25" S 151°14'07.48" E elev 113 ft eye alt 12.58 mi


Back Mountain
Pennsylvania
Student
~200 km away
Harrisburg
Reading
Instructor
Lancaster
York
palachian Mountains
Chambersburg
© 2016 Google
Image Landsat
Google earth
Imagery Date: 12/31/1969 40°21'14.62" N 76°54'03.86" W elev 405 ft eye alt 87.15 mi


Instructor
Student
Glacier Island
© 2016 Google
Google earth
Imagery Date: 7/15/2014 46°51'53.46" N 121°45'14.77" W elev 11933 ft eye alt 12.91 mi


Feasterville-Trevose
Willow Grove
Student
Bensalem
Bristol
Burlington Island
Croydon
Hollywood
Cornwells Heights-Eddington
Burlington
Hawk Island
Fenton Island
Willingboro
Delran
Riverton
Instructor
Cinnaminson
© 2016 Google
Google earth
Imagery Date: 5/24/2016 40°03'09.14" N 74°58'27.72" W elev 13 ft eye alt 51861 ft

~100 km away


Riverton
Willow Gro
Hollywood
Instruct
Bristol
Burlington Island
ts-Eddington
Burlington
Willingboro
Riverton
Delran
Instructor
Cinnaminson
© 2016 Google
Google earth
Imagery Date: 5/24/2016 40°03'09.14" N 74°58'27.72" W elev 13 ft eye alt 51861 ft

~100 km away


3/2016
Student
~440 km away
Instructor
Image © 2016 DigitalGlobe
Image NASA
Google earth
41°09'47.21" N 110°04'25.27" W elev 6870 ft eye alt 2491.45 mi

```
      PROGRAM DEG_TO_RAD
C
C A program to convert coordinates of latitude and longitude from
C degrees to radians.
C
C Define variables
      REAL LAT_D, LAT_M, LAT_S, LAT_R
      REAL LONG_D, LONG_M, LONG_S, LONG_R
      CHARACTER LAT_HEM, LONG_HEM
C
C Set constants
      PARAMETER (PI=3.14159, DEC_M=60, DEC_S=360)
C Read latitude and longitude
      PRINT *, 'Enter latitude degs,mins,secs separated by commas: '
      READ *, LAT_D, LAT_M, LAT_S
      PRINT *, 'N or S?: '
      READ *, LAT_HEM
C
      PRINT *, 'Enter longitude degs,mins,secs separated by commas: '
      READ *, LONG_D, LONG_M, LONG_S
      PRINT *, 'E or W?: '
      READ *, LONG_HEM
C
C Convert latitude and longitude to decimal degrees
      LAT_M = LAT_M/DEC_M
      LAT_S = LAT_S/DEC_S
      LAT_D = LAT_D + LAT_M + LAT_S
      IF (LAT_HEM.EQ.'S') THEN
            LAT_D = LAT_D*(-1)
      END IF
...
```

```
      PROGRAM DEG_TO_RAD
C
C A program to convert coordinates of latitude and longitude from
C degrees to radians.
C
C Define variables
      REAL LAT_D, LAT_M, LAT_S, LAT_R
      REAL LONG_D, LONG_M, LONG_S, LONG_R
      CHARACTER LAT_HEM, LONG_HEM
C
C Set constants
      PARAMETER (PI=3.14159, DEC_M=60., DEC_S=3600.)
C Read latitude and longitude
      PRINT *, 'Enter latitude degs,mins,secs separated by commas: '
      READ *, LAT_D, LAT_M, LAT_S
      PRINT *, 'N or S?: '
      READ *, LAT_HEM
C
      PRINT *, 'Enter longitude degs,mins,secs separated by commas: '
      READ *, LONG_D, LONG_M, LONG_S
      PRINT *, 'E or W?: '
      READ *, LONG_HEM
C
C Convert latitude and longitude to decimal degrees
      LAT_M = LAT_M/DEC_M
      LAT_S = LAT_S/DEC_S
      LAT_D = LAT_D + LAT_M + LAT_S
      IF (LAT_HEM.EQ.'S') THEN
            LAT_D = LAT_D*(-1.0)
      END IF
...
```

```
      PROGRAM  conversion_radian
C This program converts the user inputed lattitude and longitude to
C radians
      real latdeg,latmin,latsec,Reclon,Radlat
      real londeg,lonmin,lonsec,Declon,Radlon

      pi=3.1459

      print *, 'Please enter the latitude (deg, min, sec): '
      read *,latdeg,latmin,latsec
      Declat= latdeg+latmin/60.+latsec/3600.
      Radlat= Declat*(2.*pi/360.)

      print *,'Please enter the longitude (deg, min, sec): '
      read *,londeg,lonmin,lonsec
      Declon= londeg+lonmin/60.+lonsec/3600.
      Radlon= Declon*(2.*pi/360.)

      Write (*,100) Radlat
100 format ('The latitude in radians is', F8.6)
      write (*,110) Radlon
110 format ('The longitude in radians is', F8.6)
       write (*,110) Radlon
110   format ('The longitude in radians is', F8.6)

C The above format statement puts the radian calculation into a field
C width of 8 with 6 decimal places. This was determined to be the
C appropriate number of significant digits for 0.2km accuracy.

      stop
      end
```

```
      PROGRAM  conversion_radian
C This program converts the user inputed lattitude and longitude to
C radians
      real latdeg,latmin,latsec,Reclon,Radlat
      real londeg,lonmin,lonsec,Declon,Radlon

      pi=3.14159

      print *, 'Please enter the latitude (deg, min, sec): '
      read *,latdeg,latmin,latsec
      Declat= latdeg+latmin/60.+latsec/3600.
      Radlat= Declat*(2.*pi/360.)

      print *,'Please enter the longitude (deg, min, sec): '
      read *,londeg,lonmin,lonsec
      Declon= londeg+lonmin/60.+lonsec/3600.
      Radlon= Declon*(2.*pi/360.)

      Write (*,100) Radlat
100 format ('The latitude in radians is', F8.6)
      write (*,110) Radlon
110 format ('The longitude in radians is', F8.6)
       write (*,110) Radlon
110   format ('The longitude in radians is', F8.6)

C The above format statement puts the radian calculation into a field
C width of 8 with 6 decimal places. This was determined to be the
C appropriate number of significant digits for 0.2km accuracy.

      stop
      end
```

```
      program degrad
      implicit double precision(a-h,o-z)
      character*5 rep
      pi = 3.1415
100   continue
      write(6,200)
200   format('Enter latitude (deg,min,sec): ')
      read *,xlatd,xlatm,xlats
      write(6,201)
201   format('Enter longitude (deg,min,sec): ')
      read *,xlond,xlonm,xlons
      xladeg = (xlatd)+(xlatm/60.)+(xlats/3600.)
      xlodeg = (xlond)+(xlonm/60.)+(xlons/3600.)
      write(6,401)xladeg,xlodeg
      write(6,401)xladeg,xlodeg
401   format(/,'degree lat =',F9.5,/,'degree lon =',F9.5,//)
      xlarad = xladeg*pi/180
      xlorad = xlodeg*pi/180
      write(6,202)xlarad,xlorad
202   format(/,'radian lat =',F8.5,/,'radian lon =',F8.5,//)
      write(6,500)
500   format('do you want to proceed?'/)
      read *,rep
      if(rep.eq.'n') then
            goto 300
      end if
      goto 100
300   continue
      stop
```

```
      program degrad
      implicit double precision(a-h,o-z)
      character*5 rep
      pi = 3.1416
100   continue
      write(6,200)
200   format('Enter latitude (deg,min,sec): ')
      read *,xlatd,xlatm,xlats
      write(6,201)
201   format('Enter longitude (deg,min,sec): ')
      read *,xlond,xlonm,xlons
      xladeg = (xlatd)+(xlatm/60.)+(xlats/3600.)
      xlodeg = (xlond)+(xlonm/60.)+(xlons/3600.)
      write(6,401)xladeg,xlodeg
      write(6,401)xladeg,xlodeg
401   format(/,'degree lat =',F9.5,/,'degree lon =',F9.5,//)
      xlarad = xladeg*pi/180.
      xlorad = xlodeg*pi/180.
      write(6,202)xlarad,xlorad
202   format(/,'radian lat =',F8.5,/,'radian lon =',F8.5,//)
      write(6,500)
500   format('do you want to proceed?'/)
      read *,rep
      if(rep.eq.'n') then
            goto 300
      end if
      goto 100
300   continue
      stop
```

```
      program degrad
      implicit double precision(a-h,o-z)
      character*5 rep
      pi = 3.1416d0
100   continue
      write(6,200)
200   format('Enter latitude (deg,min,sec): ')
      read *,xlatd,xlatm,xlats
      write(6,201)
201   format('Enter longitude (deg,min,sec): ')
      read *,xlond,xlonm,xlons
      xladeg = (xlatd)+(xlatm/60.)+(xlats/3600.)
      xlodeg = (xlond)+(xlonm/60.)+(xlons/3600.)
      write(6,401)xladeg,xlodeg
      write(6,401)xladeg,xlodeg
401   format(/,'degree lat =',F9.5,/,'degree lon =',F9.5,//)
      xlarad = xladeg*pi/180.d0
      xlorad = xlodeg*pi/180.d0
      write(6,202)xlarad,xlorad
202   format(/,'radian lat =',F8.5,/,'radian lon =',F8.5,//)
      write(6,500)
500   format('do you want to proceed?'/)
      read *,rep
      if(rep.eq.'n') then
            goto 300
      end if
      goto 100
300   continue
      stop
```

```
      proGram LatLon convert

      pi=DACOS(-1.D0)
      a = pi/180

      write(6,700)
700   format ('Input latitude (degrees, minutes, seconds): ',$)
      read(*,*) wlat_deg,xlat_min,xlat_sec
      write(6,704)
704   format ('Input longitude (degrees, minutes, seconds): ',$)
      read(*,*) wlon_deg,wlon_min,wlon_sec

      wlat= wlat_deg+wlat_min/60.+xlat_sec/3600.
      wlon= wlon_deg+wlon_min/60.+xlon_sec/3600.
C  print *,'Lat (degrees) =',wlat, 'Lon (degrees) =',wlon

      rlat = wlat*a
      rlon = wlon*a
      print *,'Lat(radians) = ',rlat, 'Lon (radians) = ',rlon
      stop
      end
```

```
      proGram LatLon convert

      pi=DACOS(-1.D0)
      a = pi/180

      write(6,700)
700   format ('Input latitude (degrees, minutes, seconds): ',$)
      read(*,*) wlat_deg,xlat_min,xlat_sec
      write(6,704)
704   format ('Input longitude (degrees, minutes, seconds): ',$)
      read(*,*) wlon_deg,wlon_min,wlon_sec

      wlat= wlat_deg+wlat_min/60.+xlat_sec/3600.
      wlon= wlon_deg+wlon_min/60.+xlon_sec/3600.
C  print *,'Lat (degrees) =',wlat, 'Lon (degrees) =',wlon

      rlat = wlat*a
      rlon = wlon*a
      print *,'Lat(radians) = ',rlat, 'Lon (radians) = ',rlon
      stop
      end
```

```
      proGram LatLon convert

      pi=DACOS(-1.D0)
      dtr = pi/180.D0

      write(6,700)
700   format ('Input latitude (degrees, minutes, seconds): ',$)
      read(*,*) wlat_deg,wlat_min,wlat_sec
      write(6,704)
704   format ('Input longitude (degrees, minutes, seconds): ',$)
      read(*,*) wlon_deg,wlon_min,wlon_sec

      wlat= wlat_deg+wlat_min/60.+wlat_sec/3600.
      wlon= wlon_deg+wlon_min/60.+wlon_sec/3600.
C  print *,'Lat (degrees) =',wlat, 'Lon (degrees) =',wlon

      rlat = wlat*dtr
      rlon = wlon*dtr
      print *,'Lat(radians) = ',rlat, 'Lon (radians) = ',rlon
      stop
      end
```

```
      proGram LatLon convert
      implicit double precision (a-h, o-z)
      pi=DACOS(-1.D0)
      dtr = pi/180.D0

      write(6,700)
700   format ('Input latitude (degrees, minutes, seconds): ',$)
      read(*,*) wlat_deg,wlat_min,wlat_sec
      write(6,704)
704   format ('Input longitude (degrees, minutes, seconds): ',$)
      read(*,*) wlon_deg,wlon_min,wlon_sec

      wlat= wlat_deg+wlat_min/60.+wlat_sec/3600.
      wlon= wlon_deg+wlon_min/60.+wlon_sec/3600.
C  print *,'Lat (degrees) =',wlat, 'Lon (degrees) =',wlon

      rlat = wlat*dtr
      rlon = wlon*dtr
      print *,'Lat(radians) = ',rlat, 'Lon (radians) = ',rlon
      stop
      end
```

```
      program latlon_to_rad
      double precision total_lat_deg,total_lon_deg,lat_rad,lon_rad

      pi = 3.14159265358979323846D0

100   continue

! user inputs lat and lon coordinates as deg,min,sec (enter 999 to end)
! script reads and stores
      print *, 'Remember to put negative values for South and West'
      write(6,704)
704   format('Enter latitude in (deg,min,sec)..(999,0,0) to end: ',$)
      read(5,*)lat_deg,lat_min,lat_sec
      if(lat_deg.eq.999) goto 999
      write(6,706)
706   format('Enter longitude in (deg,min,sec): ',$)
      read(5,*)lon_deg,lon_min_lon_sec

! conversion from deg to to decimal deg then to radians
      total_lat_deg = lat_deg + lat_min/60 + lat_sec/3600
! conversion from deg to to decimal deg then to radians
      total_lat_deg = lat_deg + lat_min/60 + lat_sec/3600
      total_lon_deg = lon_deg + lon_min/60 + lon_sec/3600
      lat_rad = total_lat_deg * pi/180
      lon_rad = total_lon_deg * pi/180

! writes lat and lon in radians with 5 sig figs (5 after decimal?)
      write(6,702)lat_rad,lon_rad
702   format(/,'Lat (radians): ',F9.5,'    Long (radians): ',F9.5,//)

! loops back to beginning for another coordinate
      goto 100

999   continue

      stop
      end
```

```
Enter latitude in (deg,min,sec)..(999,0,0) to end: 38.,58.,53.50
 lat_deg =          38
 lat_min =          58
 lat_sec =          53
Enter longitude in (deg,min,sec): 76.,56.,40.23
 lon_deg =          76
 lon_min =           0
 lon_sec =           0

Lat (radians):   0.66323    Long (radians):   1.32645
```

```
      program latlon_to_rad
      double precision total_lat_deg,total_lon_deg,lat_rad,lon_rad

      pi = 3.14159265358979323846D0

100   continue

! user inputs lat and lon coordinates as deg,min,sec (enter 999 to end)
! script reads and stores
      print *, 'Remember to put negative values for South and West'
      write(6,704)
704   format('Enter latitude in (deg,min,sec)..(999,0,0) to end: ',$)
      read(5,*)lat_deg,lat_min,lat_sec
      if(lat_deg.eq.999) goto 999
      write(6,706)
706   format('Enter longitude in (deg,min,sec): ',$)
      read(5,*)lon_deg,lon_min_lon_sec

! conversion from deg to to decimal deg then to radians
      total_lat_deg = lat_deg + lat_min/60 + lat_sec/3600
! conversion from deg to to decimal deg then to radians
      total_lat_deg = lat_deg + lat_min/60 + lat_sec/3600
      total_lon_deg = lon_deg + lon_min/60 + lon_sec/3600
      lat_rad = total_lat_deg * pi/180
      lon_rad = total_lon_deg * pi/180

! writes lat and lon in radians with 5 sig figs (5 after decimal?)
      write(6,702)lat_rad,lon_rad
702   format(/,'Lat (radians): ',F9.5,'    Long (radians): ',F9.5,//)

! loops back to beginning for another coordinate
      goto 100

999   continue

      stop
      end
```

```
      program latlon_to_rad
      double precision total_lat_deg,total_lon_deg,lat_rad,lon_rad
      pi = 3.14159265358979323846D0
100   continue
! user inputs lat and lon coordinates as deg,min,sec (enter 999 to end)
! script reads and stores
      print *, 'Remember to put negative values for South and West'
      write(6,704)
704   format('Enter latitude in (deg,min,sec)..(999,0,0) to end: ',$)
      read(5,*)lat_deg,lat_min,lat_sec
      if(lat_deg.eq.999) goto 999
      write(6,706)
706   format('Enter longitude in (deg,min,sec): ',$)
      read(5,*)lon_deg,lon_min_lon_sec
! conversion from deg to to decimal deg then to radians
      total_lat_deg = lat_deg + lat_min/60 + lat_sec/3600
! conversion from deg to to decimal deg then to radians
      total_lat_deg = lat_deg + lat_min/60 + lat_sec/3600
      total_lon_deg = lon_deg + lon_min/60 + lon_sec/3600
      lat_rad = total_lat_deg * pi/180
      lon_rad = total_lon_deg * pi/180
! writes lat and lon in radians with 5 sig figs (5 after decimal?)
      write(6,702)lat_rad,lon_rad
702   format(/,'Lat (radians): ',F9.5,'    Long (radians): ',F9.5,//)
! loops back to beginning for another coordinate
      goto 100
999   continue
      stop
      end
```

```
      program latlon_to_rad
      double precision total_lat_deg,total_lon_deg,lat_rad,lon_rad
      pi = 3.14159265358979323846D0
100   continue
! user inputs lat and lon coordinates as deg,min,sec (enter 999 to end)
! script reads and stores
      print *, 'Remember to put negative values for South and West'
      write(6,704)
704   format('Enter latitude in (deg,min,sec)..(999,0,0) to end: ',$)
      read(5,*)lat_deg,lat_min,lat_sec
      if(lat_deg.eq.999) goto 999
      write(6,706)
706   format('Enter longitude in (deg,min,sec): ',$)
      read(5,*)lon_deg,lon_min_lon_sec
! conversion from deg to to decimal deg then to radians
      total_lat_deg = lat_deg + lat_min/60 + lat_sec/3600
! conversion from deg to to decimal deg then to radians
      total_lat_deg = lat_deg + lat_min/60 + lat_sec/3600
      total_lon_deg = lon_deg + lon_min/60 + lon_sec/3600
      lat_rad = total_lat_deg * pi/180
      lon_rad = total_lon_deg * pi/180
! writes lat and lon in radians with 5 sig figs (5 after decimal?)
      write(6,702)lat_rad,lon_rad
702   format(/,'Lat (radians): ',F9.5,'    Long (radians): ',F9.5,//)
! loops back to beginning for another coordinate
      goto 100
999   continue
      stop
      end
```

```
      program latlon_to_rad
      double precision total_lat_deg,total_lon_deg,lat_rad,lon_rad
      pi = 3.14159265358979323846D0
100   continue
! user inputs lat and lon coordinates as deg,min,sec (enter 999 to end)
! script reads and stores
      print *, 'Remember to put negative values for South and West'
      write(6,704)
704   format('Enter latitude in (deg,min,sec)..(999,0,0) to end: ',$)
      read(5,*)lat_deg, lat_min, lat_sec
      if(lat_deg.eq.999) goto 999
      write(6,706)
706   format('Enter longitude in (deg,min,sec): ',$)
      read(5,*)lon_deg, lon_min, lon_sec
! conversion from deg to to decimal deg then to radians
      total_lat_deg = lat_deg + lat_min/60 + lat_sec/3600
! conversion from deg to to decimal deg then to radians
      total_lat_deg = lat_deg + lat_min/60 + lat_sec/3600
      total_lon_deg = lon_deg + lon_min/60 + lon_sec/3600
      lat_rad = total_lat_deg * pi/180
      lon_rad = total_lon_deg * pi/180
! writes lat and lon in radians with 5 sig figs (5 after decimal?)
      write(6,702)lat_rad,lon_rad
702   format(/,'Lat (radians): ',F9.5,'    Long (radians): ',F9.5,//)
! loops back to beginning for another coordinate
      goto 100
999   continue
      stop
      end
```

```
      program degrees_to_radians
      character*6 canswr
      double precision xlat_deg,xlat_min,xlat_sec,xlon_deg,xlon_min,xlon_sec
      pi=3.14159265358979323846D0
      dtr=2.D0*pi/360.D0
100   write(6,700)
700   format('Enter latitude of location (deg, min, sec) : ',$)
      read(5,*)xlat_deg,xlat_min,xlat_sec
      write(6,704)
704   format('Enter longitude of location (deg, min, sec) : ',$)
      read(5,*)xlon_deg,xlon_min,xlon_sec
C     Conversion to decimal degrees
      xlat=xlat_deg+xlat_min/60.D0+xlat_sec/3600.D0
      xlon=xlon_deg+xlon_min/60.D0+xlon_sec/3600.D0
C     Conversion of decimal degrees to radians
C     Conversion of decimal degrees to radians
      xlat_rad=xlat*dtr
      xlon_rad=xlon*dtr
      write(6,702)xlat_rad,xlon_rad
702   format('Lat (in radians) =',F9.5, ' Lon (in radians) =',F9.5)
      write(6,706)
706   format('Do you want to do another conversion? (Y/N) : ',$)
      read(5,*)canswr
      if(canswr.eq.'Y') goto 100
      stop
      end
```

```
      program degrees_to_radians
      character*6 canswr
      double precision xlat_deg,xlat_min,xlat_sec,xlon_deg,xlon_min,xlon_sec
      pi=3.14159265358979323846D0
      dtr=2.D0*pi/360.D0
100   write(6,700)
700   format('Enter latitude of location (deg, min, sec) : ',$)
      read(5,*)xlat_deg,xlat_min,xlat_sec
      write(6,704)
704   format('Enter longitude of location (deg, min, sec) : ',$)
      read(5,*)xlon_deg,xlon_min,xlon_sec
C     Conversion to decimal degrees
      xlat=xlat_deg+xlat_min/60.D0+xlat_sec/3600.D0
      xlon=xlon_deg+xlon_min/60.D0+xlon_sec/3600.D0
C     Conversion of decimal degrees to radians
C     Conversion of decimal degrees to radians
      xlat_rad=xlat*dtr
      xlon_rad=xlon*dtr
      write(6,702)xlat_rad,xlon_rad
702   format('Lat (in radians) =',F9.5, ' Lon (in radians) =',F9.5)
      write(6,706)
706   format('Do you want to do another conversion? (Y/N) : ',$)
      read(5,*)canswr
      if(canswr(1:1).eq.'Y') goto 100
      stop
      end
```

```
      program degrees_to_radians
      character*1 canswr
      double precision xlat_deg,xlat_min,xlat_sec,xlon_deg,xlon_min,xlon_sec
      pi=3.14159265358979323846D0
      dtr=2.D0*pi/360.D0
100   write(6,700)
700   format('Enter latitude of location (deg, min, sec) : ',$)
      read(5,*)xlat_deg,xlat_min,xlat_sec
      write(6,704)
704   format('Enter longitude of location (deg, min, sec) : ',$)
      read(5,*)xlon_deg,xlon_min,xlon_sec
C     Conversion to decimal degrees
      xlat=xlat_deg+xlat_min/60.D0+xlat_sec/3600.D0
      xlon=xlon_deg+xlon_min/60.D0+xlon_sec/3600.D0
C     Conversion of decimal degrees to radians
C     Conversion of decimal degrees to radians
      xlat_rad=xlat*dtr
      xlon_rad=xlon*dtr
      write(6,702)xlat_rad,xlon_rad
702   format('Lat (in radians) =',F9.5, ' Lon (in radians) =',F9.5)
      write(6,706)
706   format('Do you want to do another conversion? (Y/N) : ',$)
      read(5,*)canswr
      if(canswr.eq.'Y') goto 100
      stop
      end
```

Other Coding Issues

One student completed the assignment in

/homes/metogra/username/www/aosc652/sub_dir

Another student turned in code that worked, but forgot to update the numbers on the front page, which had been computed with an older, non-working code I was able to locate in the student's directory

This is the second time a student has not turned in the code (or poem) that was actually used (different students)

Please be sure your submitted code is the same that was used to find your numerical answers

hppltd guidance

- Each time hp ### is executed, where ### is extension number of stncl.* file, a file called hp.log is generated. The hp.log file can be helpful for understanding what has happened ☺

- A modest amount of documentation for column selection syntax is available in ~rjs/aosc652/week_03/hp.macro

- Info about special symbols is provided in file ~rjs/aosc652/week_03/hp_symbols.map

- Use linux command:

  covert –density 200 tmp331.eps tmp331.png

  to convert EPS to PNG for animation

- Have a look at ~rjs/aosc652/week_03/.convert for script to convert lots of tmp*eps files to tmp*png files

hppltd guidance

file hp.macro:

To select data in column 1 vs data in column 2, for those instances where data in column 3 lies between values of 25. and 35., enter:

1,2;$3#(25.,35.)

file hp_symbols.map

To write 30°N, must enter

30@SM16@SR

To write z < 30 km, must enter

z @MA40@SR 30 km

SR is Simplex Roman, SM is Simplex Math, and MA is Math

There are 21 character fonts and 11 symbol fonts available

Open Office Impress

- To open: % ooimpress

- First time: select empty presentation (first query) &
  slide (second query)

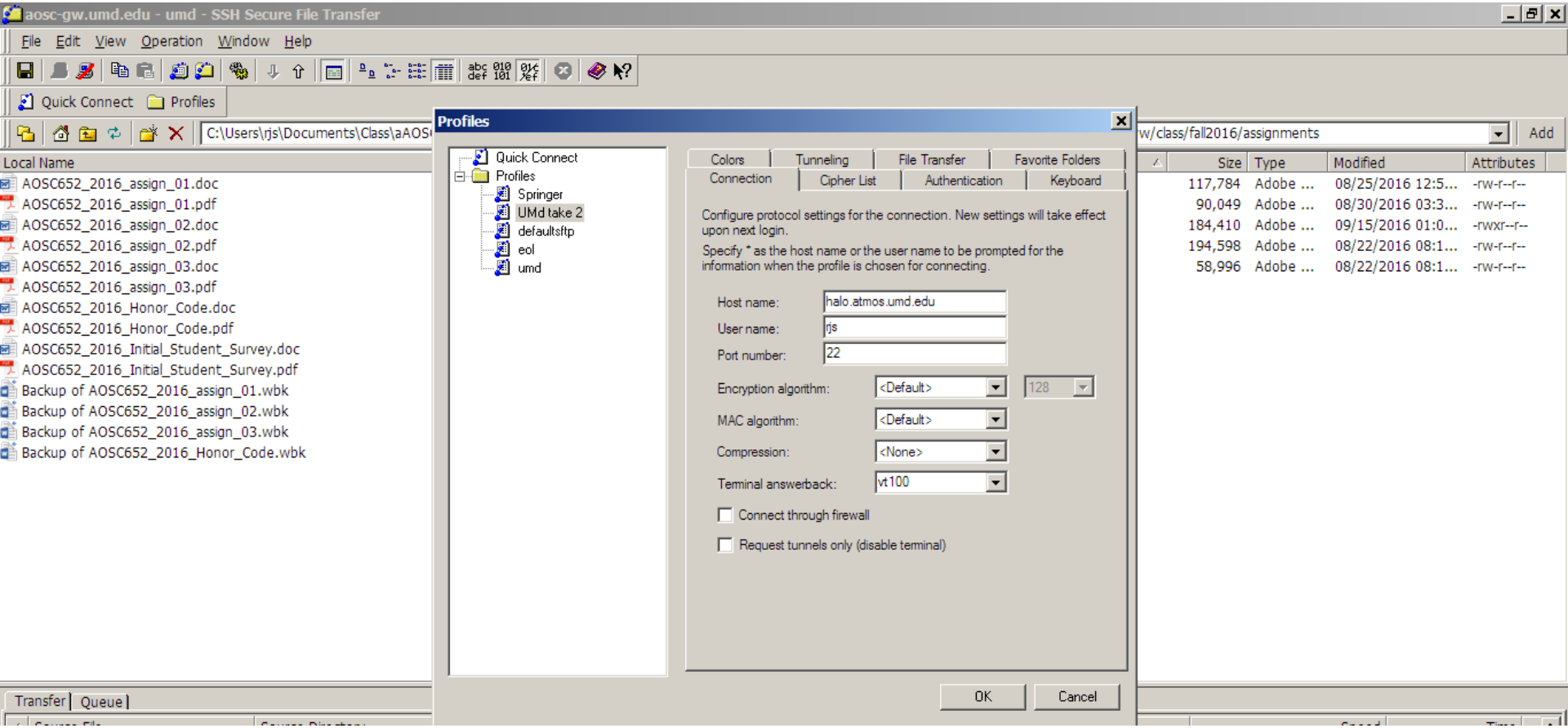
- Once you set up initial placement of figures on a slide:
  Top: global map, png file from web
  Bottom: time series, png file, your creation

  then if you:
  duplicate the slide
  make figure hot
  insert new figure w/ old figure hot

  the new figure will be placed at exact same spot as old figure
  (very nice feature of Open Office Impress)

SSH Secure Shell File Transfer to get files to Windows

- Process must be initiated from Windows machine
- See slides shown 9 Sept 2016, at start of class



SSH Secure Shell File Transfer to get files to Windows