How to get lost, while knowing where you are

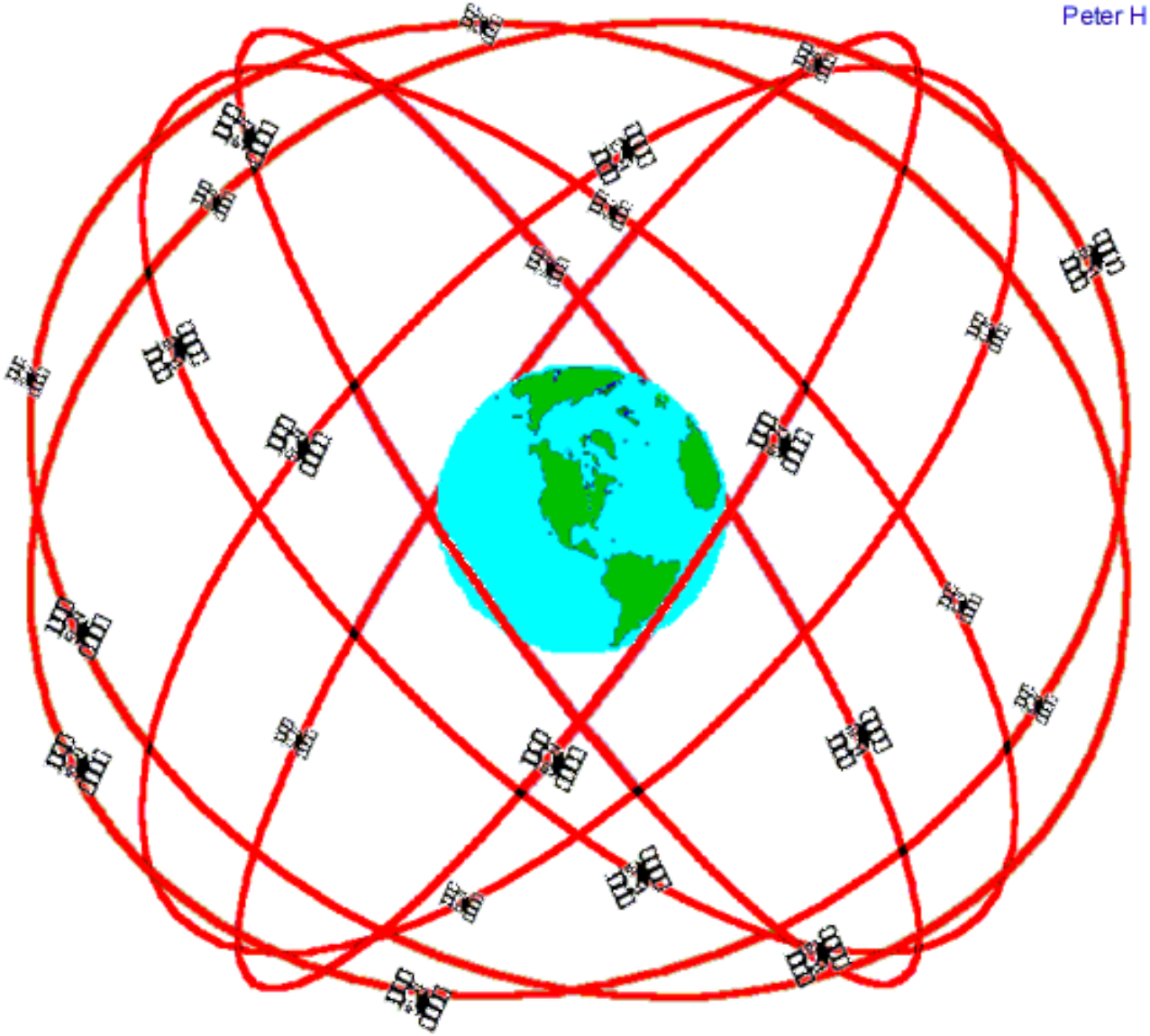
Ted Logan

Boulder Linux User's Group

10 June 2010

http://en.wikipedia.org/wiki/File:GPS_Satellite_NASA_art-iif.jpg

Peter H. Dana 9/22/98



GPS Nominal Constellation
24 Satellites in 6 Orbital Planes
4 Satellites in each Plane
20,200 km Altitudes, 55 Degree Inclination

http://www.colorado.edu/geography/gcraft/notes/gps/gif/orbits.gif

Math

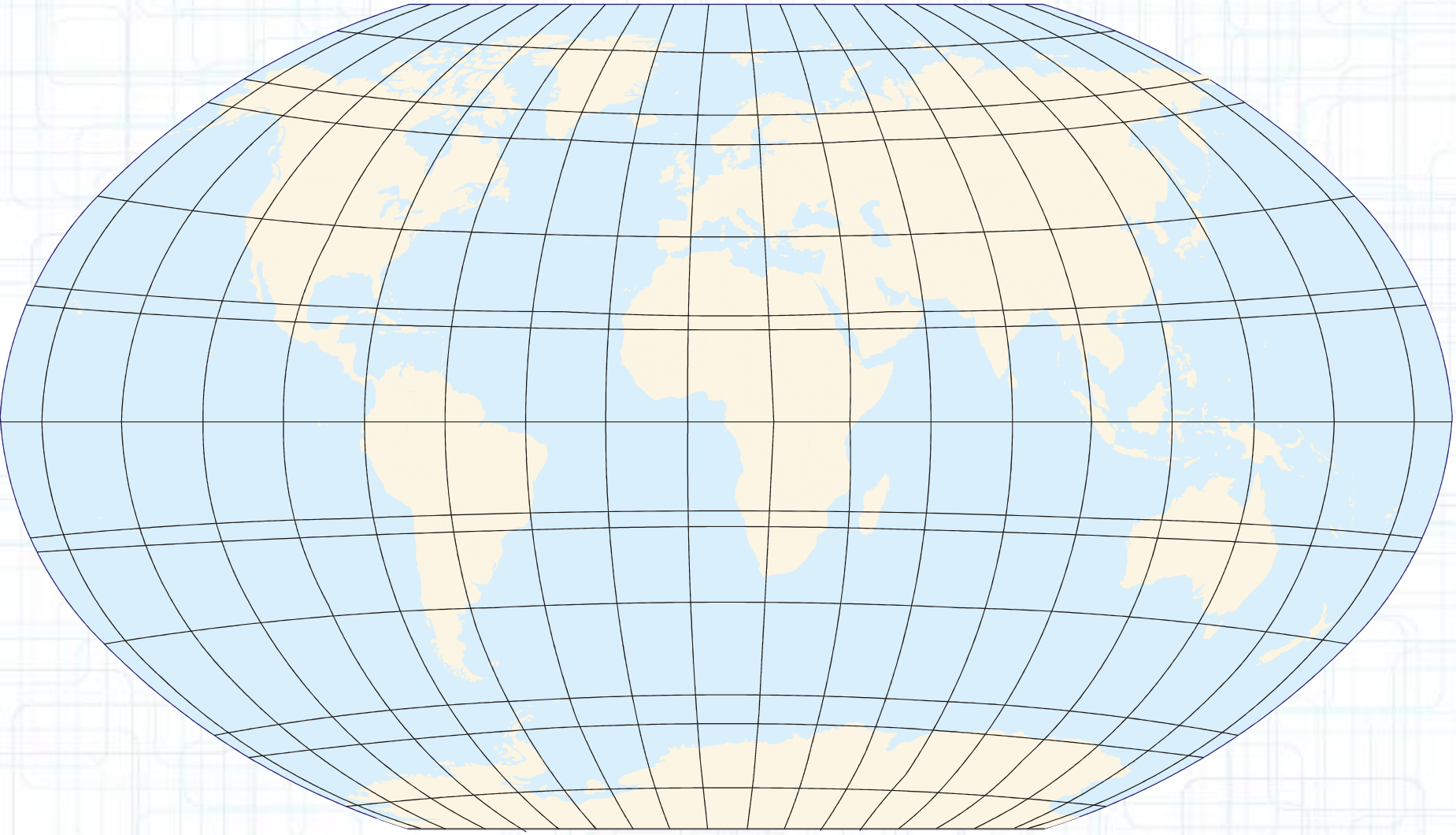
- Given:
  - Four satellites, known positions
  - Four time measurements
- Solve:
  - Your position
  - Your time

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = \left([tr_i + b - t_i]c\right)^2,$$

$$p_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} - bc, \; i = 1, 2, 3, 4.$$

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

Longitude and Latitude



http://upload.wikimedia.org/wikipedia/commons/5/56/World_map_longlat.svg

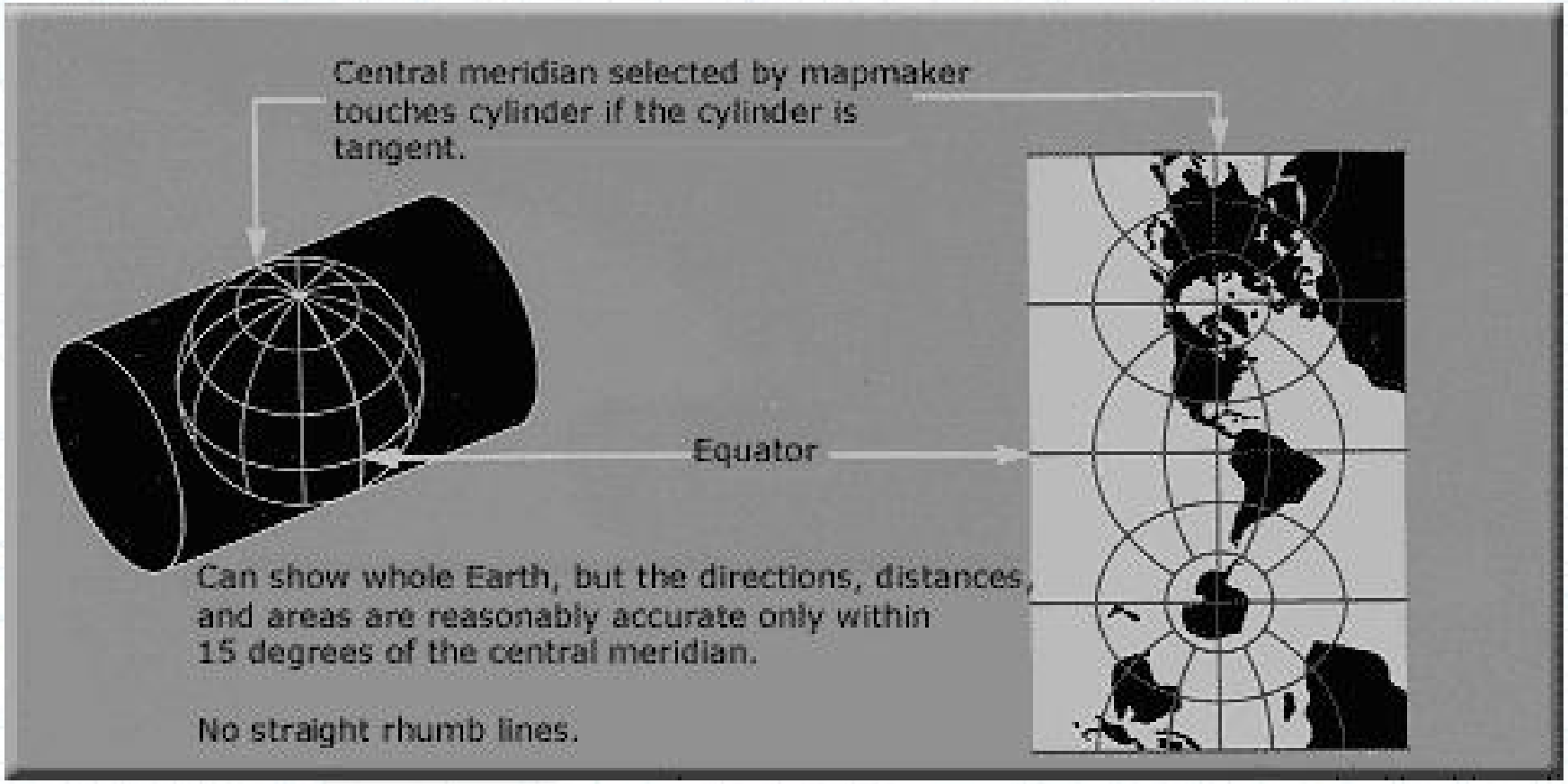
Where are we?

- 40°01.409 N 105°13.600 W

Latitude

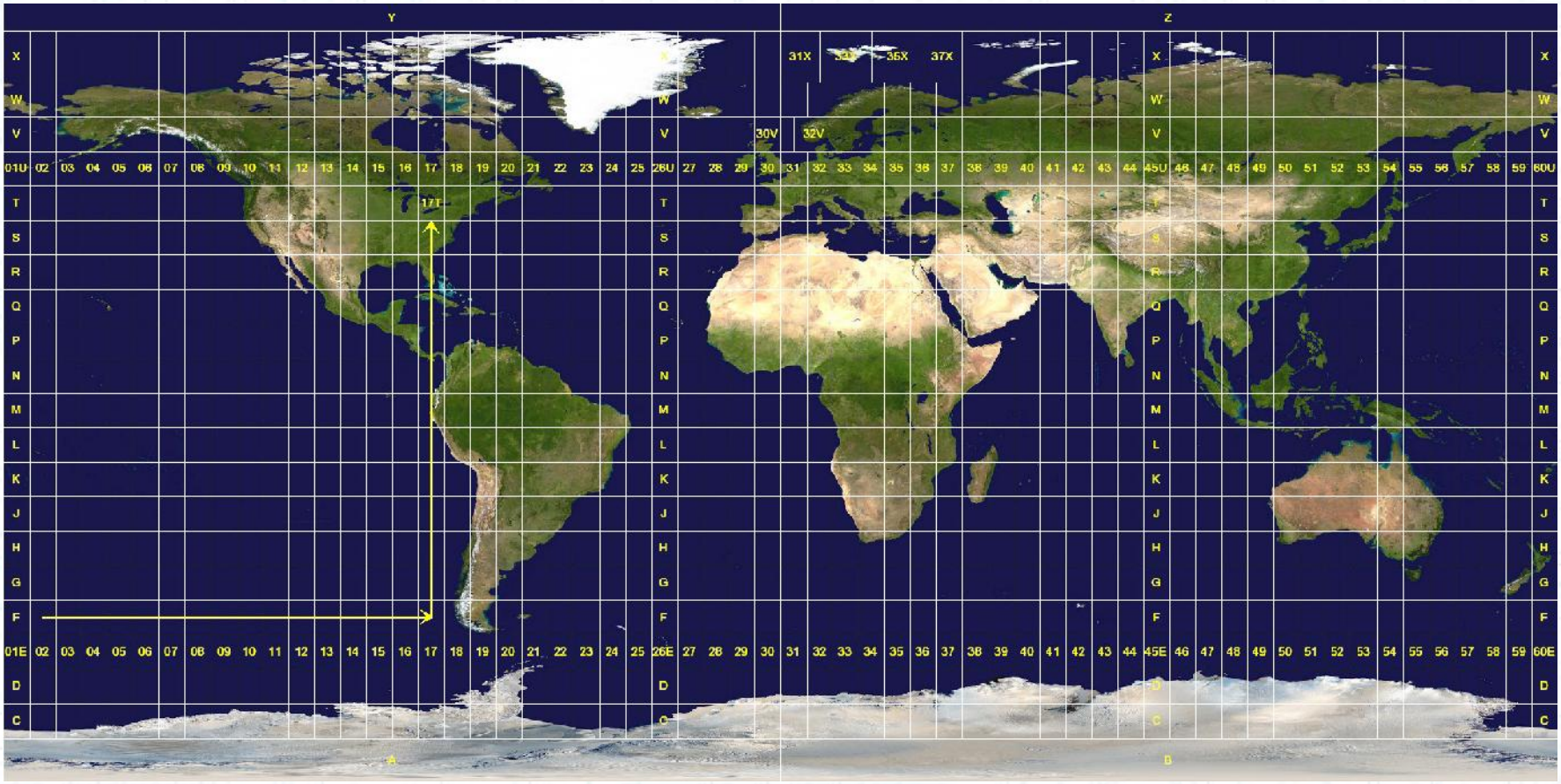
Longitude

Transverse Mercator



http://en.wikipedia.org/wiki/File:Usgs_map_traverse_mercator.PNG

UTM Zones



http://upload.wikimedia.org/wikipedia/commons/e/ed/Utm-zones.jpg

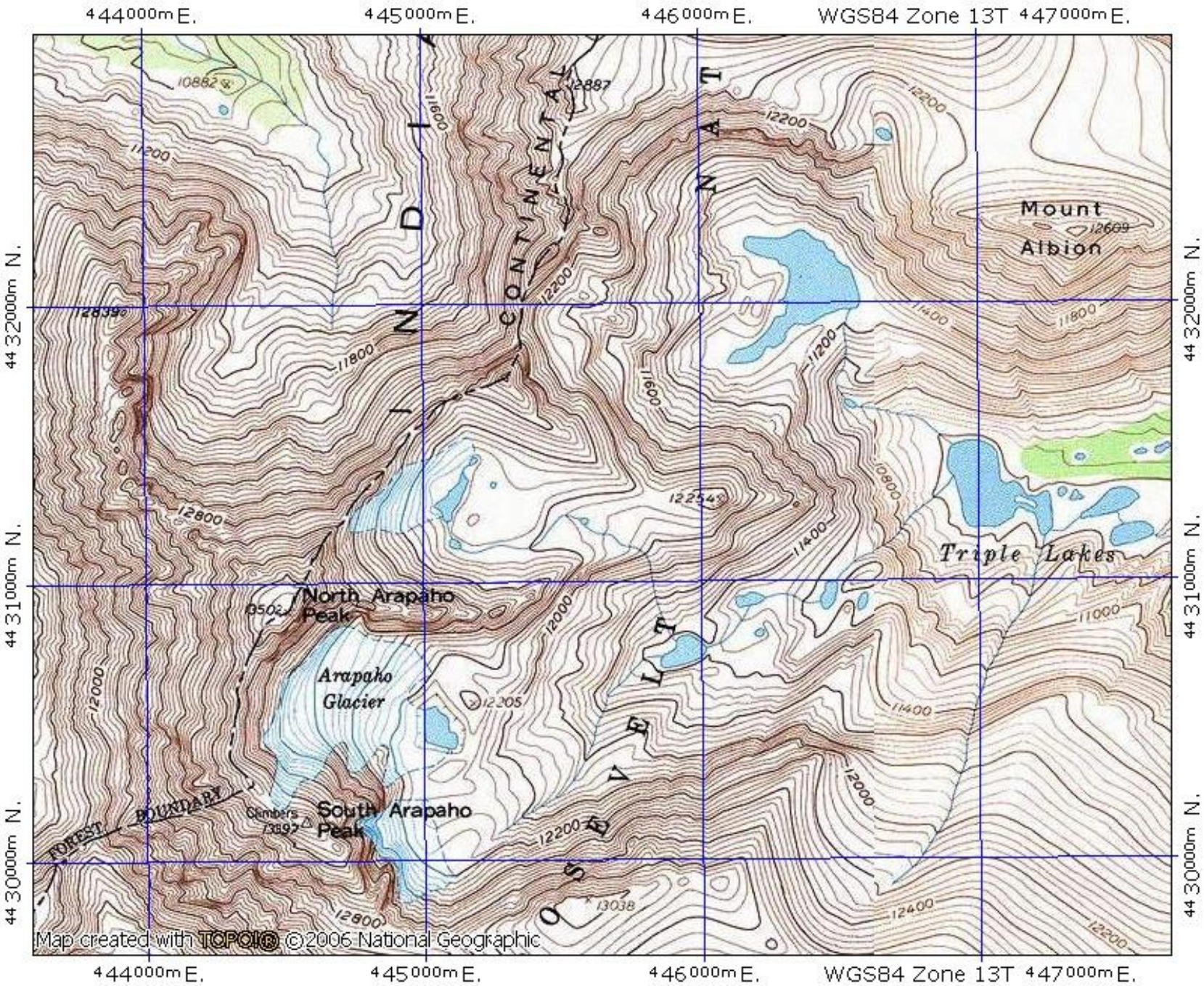
Where are we?

- 13T 480658 4430388

Northing

Easting

Zone



NATIONAL GEOGRAPHIC

0.0 0.5 1.0 miles

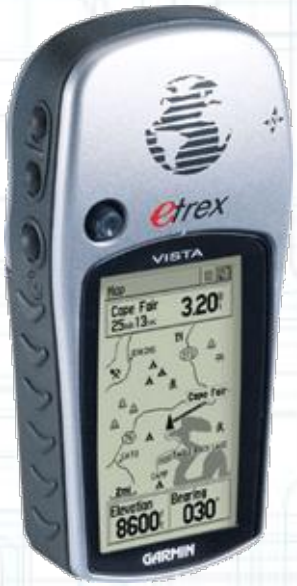
0.0 0.5 1.0 1.5 km

TN MN

10°

06/09/10

Demo: Where am I now?



$GPBOD,,T,,M,,*47
$GPVTG,5.4,T,355.3,M,0.0,N,0.0,K*4F
$PGRME,5.5,M,8.4,M,10.0,M*13
$PGRMZ,5035,f*37
$PGRMM,WGS 84*06
$HCHDG,83.2,,,10.1,E*20
$GPRTE,1,1,c,*37
$GPRMC,024028,A,4010.7105,N,10504.2886,W,0.0,5.4,300610,10.1,E,A*32
$GPRMB,A,,,,,,,,,,,A,A*0B
$GPGGA,024028,4010.7105,N,10504.2886,W,1,07,1.5,1536.8,M,-20.9,M,,*42
$GPGSA,A,3,,04,09,12,17,20,,27,28,,,,2.4,1.5,1.7*33
$GPGSV,3,2,12,17,55,046,46,20,14,061,42,26,00,235,00,27,48,276,45*70
$GPGLL,4010.7105,N,10504.2886,W,024028,A,A*5B

Demo: gpsd

```
$ gpsd /dev/ttyS0
$ telnet localhost 2947
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
d
GPSD,D=2010-06-30T02:52:28.0Z
q
GPSD,Q=7 0.00 2.50 1.30 2.88 5.24
p
GPSD,P=40.178491667 -105.071470000
a
GPSD,A=1531.900
```

Time and date

Number of satellites

Position, in longitude and latitude

Altitude, in meters

Demo: Where have I been?

Use track log

Input format

Input file

- gpsbabel -t -i garmin -f /dev/ttyS0 -o gpx -F track-2010-06-05.gpx

Output file

Output format

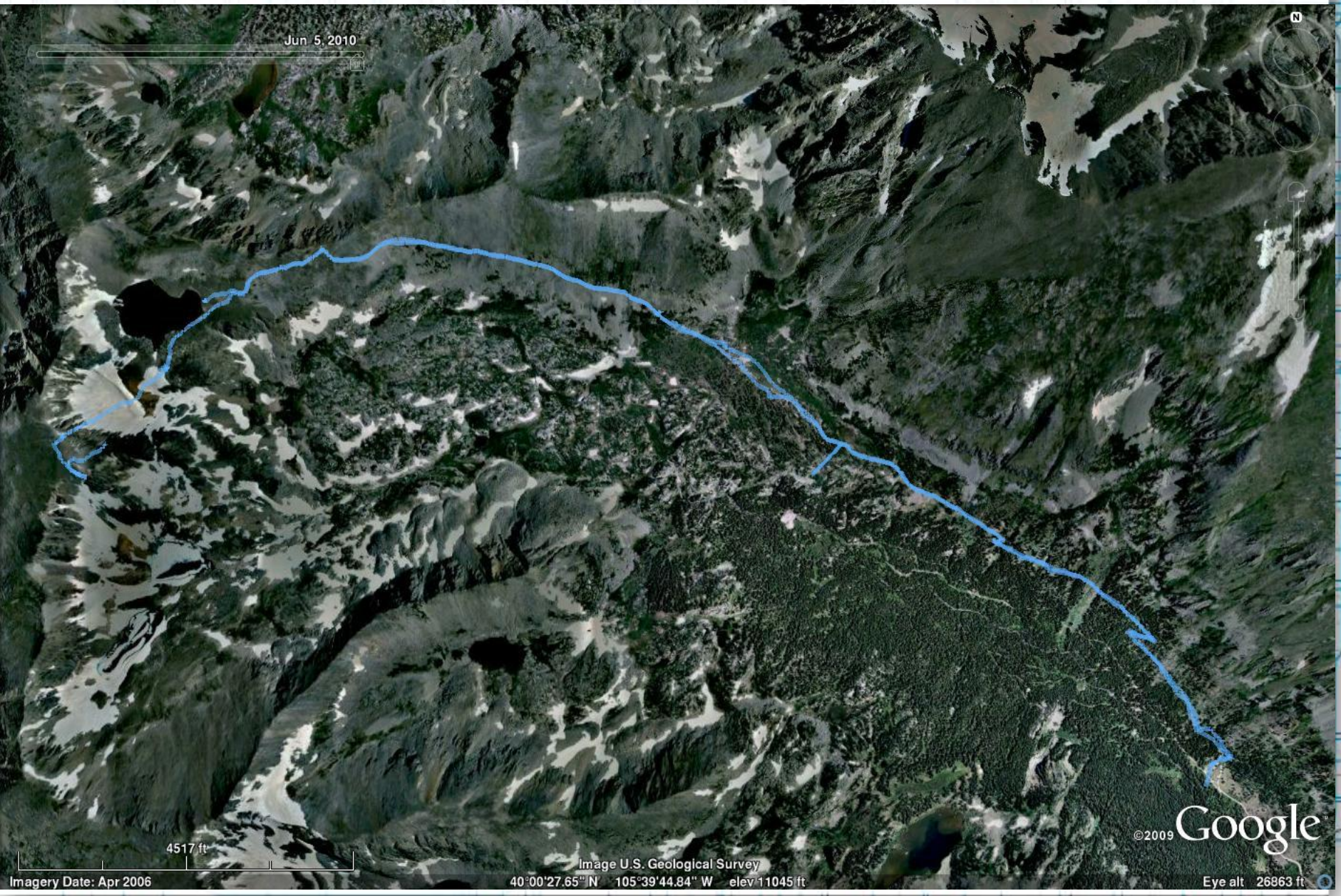
GPX file: track-2010-06-05.gpx


```
  <name>ACTIVE LOG #260</name>
<number>259</number>
<trkseg>
<trkpt lat="40.012764931" lon="-105.683648586">
  <ele>3685.832031</ele>
  <time>2010-06-05T17:53:25Z</time>
</trkpt>
<trkpt lat="40.012722015" lon="-105.683627129">
  <ele>3684.870605</ele>
  <time>2010-06-05T17:53:27Z</time>
</trkpt>
```

Demo: Google Earth

- gpsbabel -t -i gpx -f track-2010-06-05.gpx -o kml,points=0 -F track-2010-06-05.kml

Fourth of July to Mount Neva



Mount Neva: Juliet

Demo: Bolder Boulder Splits

```
forerunner-305$ garmin_save_runs
Extracting data from Garmin Forerunner305 Software Version 2.90
Files will be saved in '/home/jaeger/gps/forerunner-305'
...
Skipped: /home/jaeger/gps/forerunner-305/2010/05/20100525T125916.gmn
Skipped: /home/jaeger/gps/forerunner-305/2010/05/20100529T112922.gmn
Wrote:   /home/jaeger/gps/forerunner-305/2010/05/20100531T072448.gmn
```

Raw Forerunner data

```
forerunner-305$ xxd 2010/05/20100531T072448.gmn | head
0000000: 3c40 6741 724d 694e 403e 0000 6400 0000  <@gArMiN@>..d...
0000010: 3b63 0000 0100 0000 3363 0000 5601 0000  ;c......3c..V...
0000020: 0300 0000 5601 0000 f103 0000 a902 0000  ....V...........
0000030: 9e00 4903 4f03 0000 0000 0000 0000 0000  ..I.O...........
0000040: 2200 0000 0000 0000 0000 0000 0500 0000  "...............
0000050: 0010 0010 b37c 1a00 ffff ffff ffff ffff  .....|..........
0000060: 0000 0005 02ff 0000 0000 0000 0000 0000  ................
0000070: 0400 0000 0c36 2000 c501 ffff ffff ffff  .....6 .........
0000080: 4906 0001 02ff 0000 009f 0a04 cc35 2000  I............5 .
0000090: 286d 1a04 f05d 2000 ffff ffff ffff ffff  (m...] .........
forerunner-305$ garmin_dump 2010/05/20100531T072448.gmn | head
<run type="1009" track="158" sport="running" multisport="no">
 <laps first="841" last="847"/>
</run>
<lap type="1015" index="841" start="2010-05-31T07:24:48-06:00"
duration="0:07:15.51" distance="1609.34399" trigger="distance">
 <begin_pos lat="40.03350821" lon="-105.25376676"/>
 <end_pos lat="40.02222240" lon="-105.25935531"/>
 <max_speed>4.42204285</max_speed>
 <calories>137</calories>
 <avg_hr>158</avg_hr>
 <max_hr>169</max_hr>
```

Parsed Forerunner data

```
forerunner-305$ ~/src/perl/show_runs.pl 2010/05/20100531T072448.gmn
2010-05-31 (Monday)

Lap      Distance  Pace      Time     Avg Hrt  Max Hrt
         (miles)   (min/mi)  (min)    (bpm)    (bpm)
------   --------  --------  -------  -------  -------
Lap  1       1.00      7:15     7:15      158      169
Lap  2       1.00      7:37     7:37      170      182
Lap  3       1.00      8:01     8:01      169      181
Lap  4       1.00      7:59     7:59      167      174
Lap  5       1.00      8:00     8:00      167      173
Lap  6       1.00      7:42     7:42      171      183
Lap  7       0.30      7:25     2:13      176      179
         --------  --------  -------
Total        6.30      7:44    48:49
```

Demo: Where am I going?

- U.S. Board on Geographic Names: http://geonames.usgs.gov/pls/gnispublic/

USGS GNIS search

USGS

Geographic Names Information System (GNIS)

Query Form For The United States And Its Territories

Feature Name: longs peak

Exact Match Exclude Variants

State: Colorado

County:

Feature ID:

Feature Class: Press Ctrl to select more than one.

Airport
Arch
Area
Arroyo

Definitions

Elevation*:

Feet Meters

Send Query Erase Query

USGS GNIS results

USGS
Geographic Names Information System (GNIS)

Feature Query Results

Click the feature name for details and to access map services
Click any column name to sort the list ascending ▲ or descending ▼

Feature Name ▲	ID	Class	County	State	Latitude	Longitude	Ele(ft)*	Map**	BGN	Entry Date
East Longs Peak Trail	177938	Trail	Boulder	CO	401620N	1053602W	11729	Longs Peak	-	13-OCT-78
Longs Peak	204979	Summit	Boulder	CO	401517N	1053657W	14239	Longs Peak	01-JAN-91	13-OCT-78
Longs Peak Campground	197715	Locale	Larimer	CO	401629N	1053331W	9383	Longs Peak	-	01-SEP-92
Longs Peak Filtration Plant	178338	Locale	Boulder	CO	401235N	1050941W	5210	Hygiene	-	13-OCT-78
Longs Peak Mine	1683952	Mine	Boulder	CO	400500N	1050342W	4964	Erie	-	29-APR-96
Longs Peak Number 1 Dam	198508	Dam	Boulder	CO	401236N	1050938W	5226	Hygiene	-	01-SEP-94
Longs Peak Number 2 Dam	198510	Dam	Boulder	CO	401236N	1050926W	5197	Hygiene	-	01-SEP-94
Longs Peak Post Office (historical)	2026900	Post Office	Lake	CO	Unknown	Unknown	-	Unknown	-	05-FEB-04
Longs Peak Ranger Station	177950	Locale	Larimer	CO	401621N	1053326W	9396	Longs Peak	-	13-OCT-78
Longs Peak Reservoir Number 1	198509	Reservoir	Boulder	CO	401236N	1050938W	5226	Hygiene	-	01-SEP-94
Longs Peak Reservoir Number 2	198511	Reservoir	Boulder	CO	401236N	1050926W	5197	Hygiene	-	01-SEP-94
Tahosa Valley	204978	Valley	Larimer	CO	401450N	1053055W	8327	Allenspark	01-JAN-16	13-OCT-78

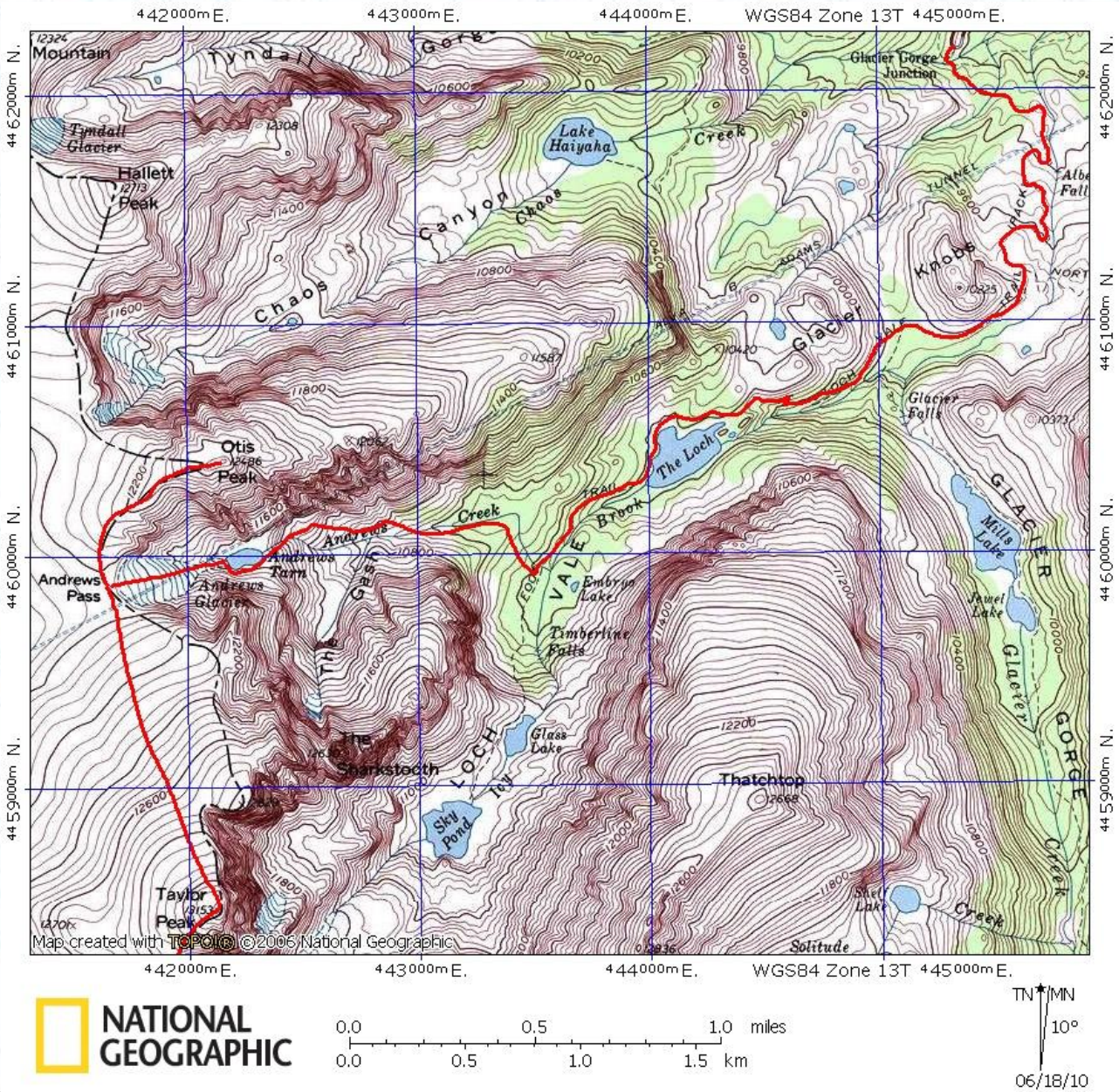
1 - 12

USGS GNIS Coordinates

Coordinates (One point per USGS topographic map containing the feature, NAD83)

Sequence	Latitude(DEC)	Longitude(DEC)	Latitude(DMS)	Longitude(DMS)	Map Name
1	40.2547069	-105.6158365	401517N	1053657W	Longs Peak

Andrews Glacier



NATIONAL GEOGRAPHIC

0.0 0.5 1.0 miles
0.0 0.5 1.0 1.5 km

TN MN
10°
06/18/10

Demo: Create a route (1)

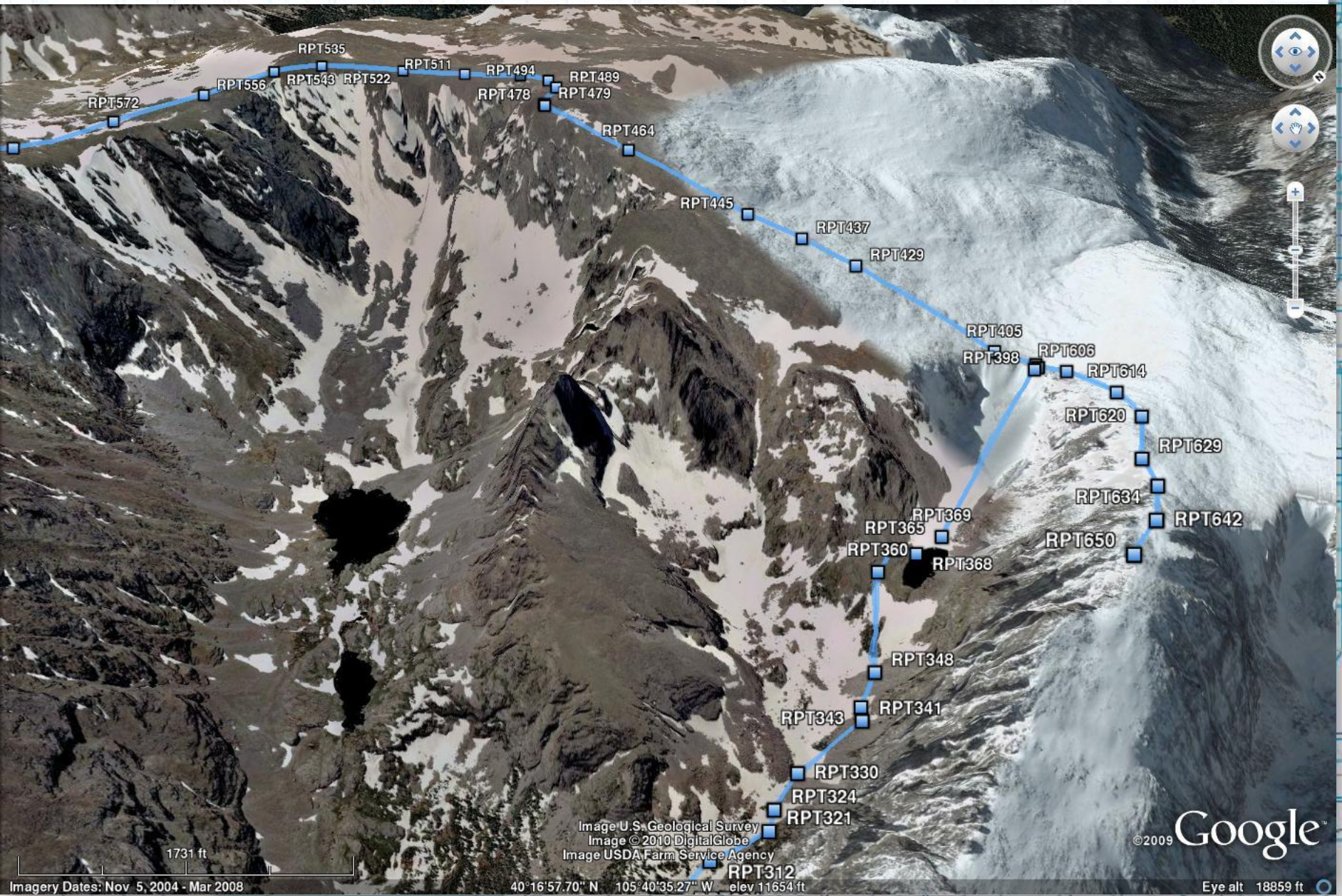
- gpsbabel -t -i tpo3 -f andrews_glacier.tpo -o gpx -F andrews_glacier.gpx
- gpsbabel -t -i gpx -f andrews_glacier.gpx -o kml,points=0 -F andrews_glacier.kml

Andrews Glacier in Google Earth

Demo: Create a route (2)

- gpsbabel -i gpx -f andrews_glacier.gpx -x transform,rte=trk,del -x simplify,error=0.01k -o kml -F andrews_glacier_route.kml

Route in Google Earth

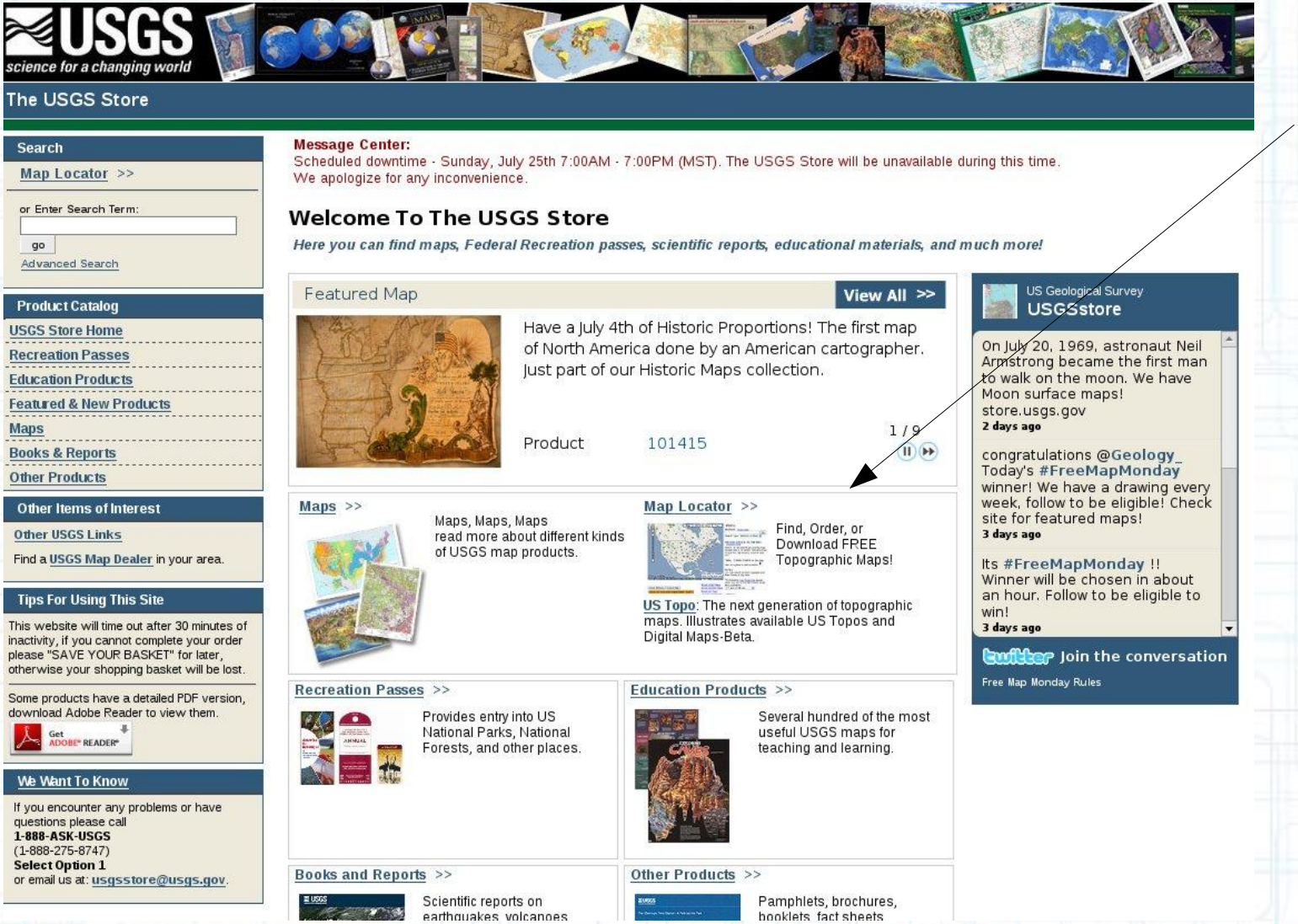


Demo: Create a route (3)

- gpsbabel -i gpx -f andrews_glacier.gpx -x transform,rte=trk,del -x simplify,error=0.01k -o garmin -F /dev/ttyS0

Demo: USGS Maps

- http://store.usgs.gov

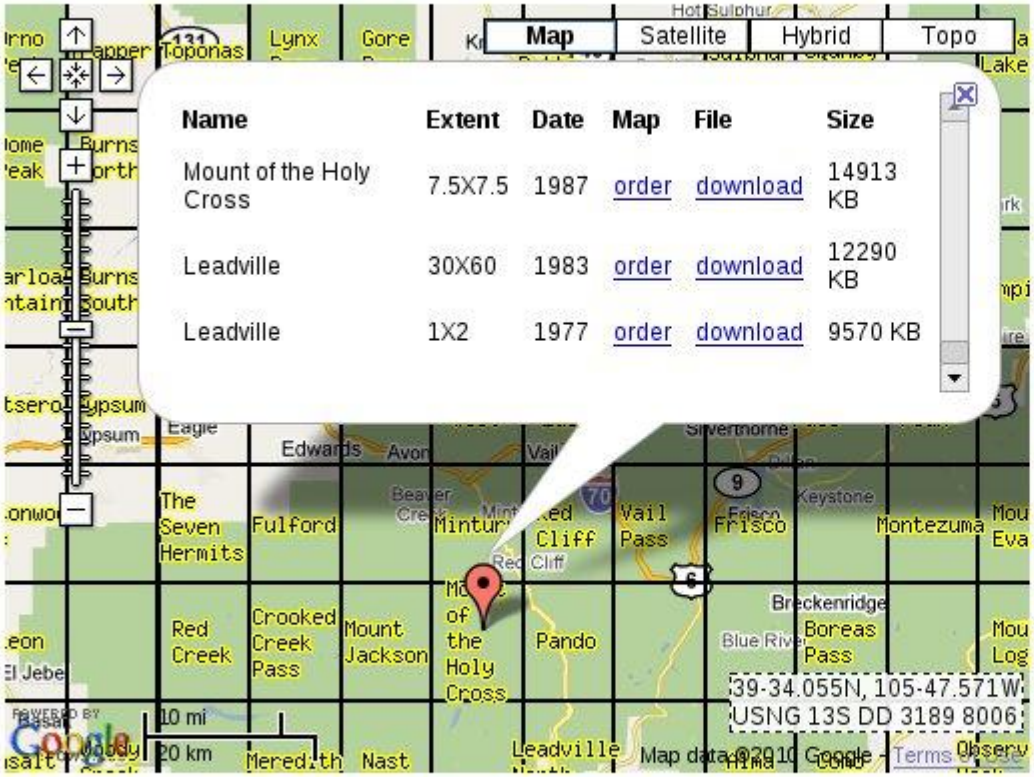


USGS Map Download

Don't see the Map Locator? Help

Map Locator

Having trouble? Call: 1-888-ASK-USGS (1-888-275-8747) or Write: usgsstore@usgs.gov for help.



Clear Markers | Reset Map

Show US Topo and "Digital Maps - Beta"

About USGS Maps
About GeoPDF Maps
About US Topo
US Topo Quickstart

STEP 1.

SEARCH: [Search Help]

Go

Search Type: Address or Place

OR FIND A PLACE ON THE MAP [Navigation Help]

FIRST, ○ NAVIGATE around the map: double click to re-center, click and drag to pull the map around, zoom in and out.

THEN, ◉ MARK POINTS on the map:

click on a place to add a marker .

NOTES:
You can switch between Navigate and Mark Points at any time.

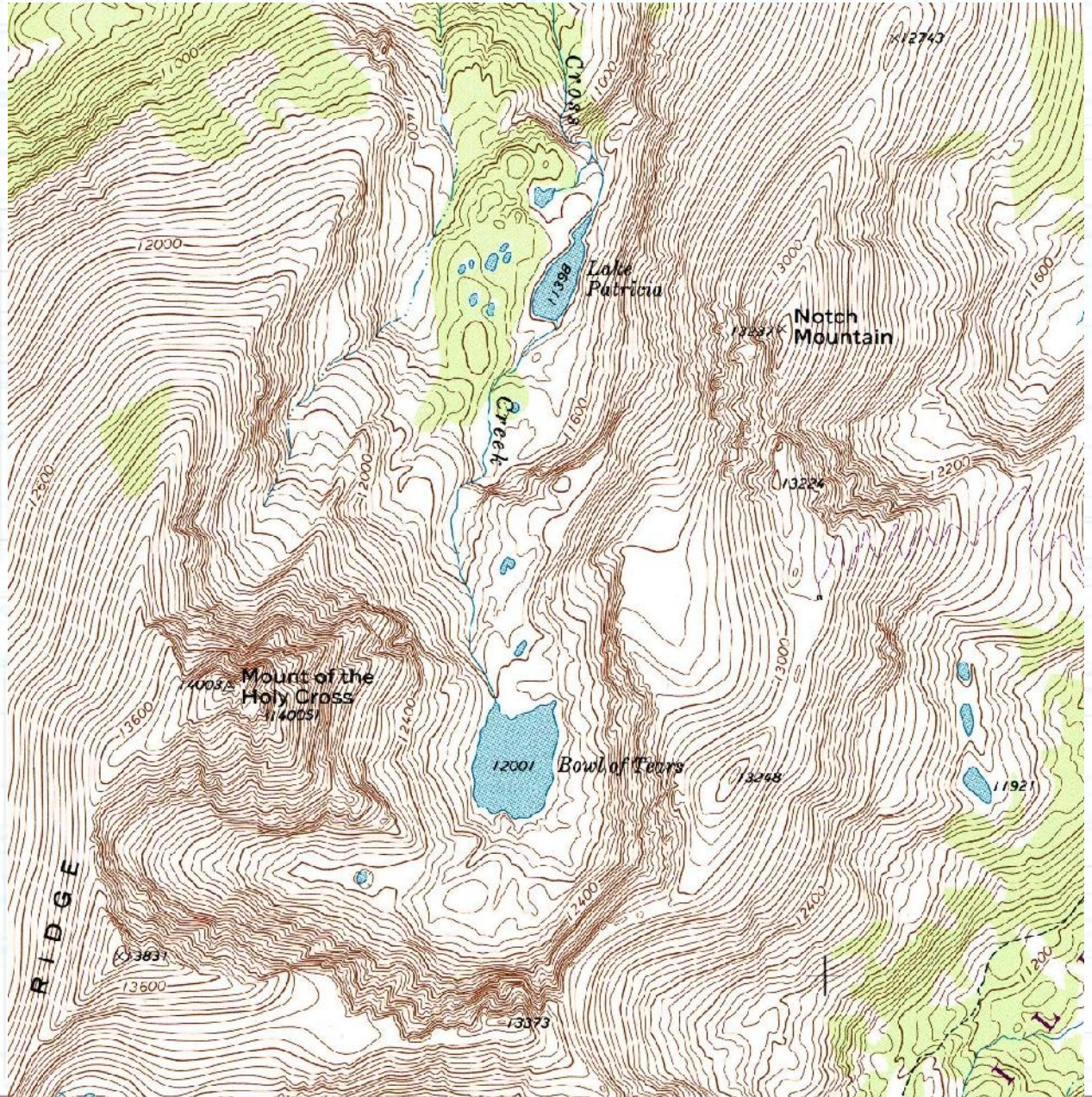
The following map footprints appear when you are in the Mark Points mode and zoomed in:

7.5 and 15 Minute

STEP 2.

SELECT AND GET YOUR MAPS

USGS topo


Cross
Lake Patricia
Notch Mountain
Creek
Mount of the Holy Cross
Bowl of Tears
RIDGE

Topo demo

```
$ topo_tile.pl CO_Mount_of_the_Holy_Cross_1987_geo.pdf
Rasterizing CO_Mount_of_the_Holy_Cross_1987_geo.pdf at 200 dpi

PDF rasterized as:
        CO_Mount_of_the_Holy_Cross_1987_geoKxkW.png

Enter state: [CO]
Enter quad name: [Mount of the Holy Cross]
Enter pixel coordinates of top-left corner: 417,226
Enter pixel coordinates of top-right corner: 3942,226
Enter pixel coordinates of bottom-left corner: 415,4808
Enter pixel coordinates of bottom-right corner: 3948,4803
Enter latitude of the northern edge of the map: 39 30
Enter latitude of the southern edge of the map: 39 22 30
Enter longitude of the eastern edge of the map: -106 22 30
Enter longitude of the western edge of the map: -106 30


Image should be rotated by 0.03280 degrees
Crop geometry: 3529x4579+414+224
Rotating and cropping CO_Mount_of_the_Holy_Cross_1987_geoKxkW.png (to
/tmp/FLCkb804Ws/CO_Mount_of_the_Holy_Cross_1987_geoKxkW.jpg)...
done.
Generating CO_Mount_of_the_Holy_Cross_1987_geoKxkW.kmz ...
  adding: CO_Mount_of_the_Holy_Cross_1987_geoKxkW.jpg (deflated 0%)
  adding: doc.kml (deflated 45%)
Done.
```

Topo demo: Google Earth

References

- http://www.gpsbabel.org/
- http://code.google.com/p/garmintools/
- http://earth.google.com/
- http://www.natgeomaps.com/topo.html

Thanks!

ted.logan@gmail.com

How to get lost, while knowing where you are

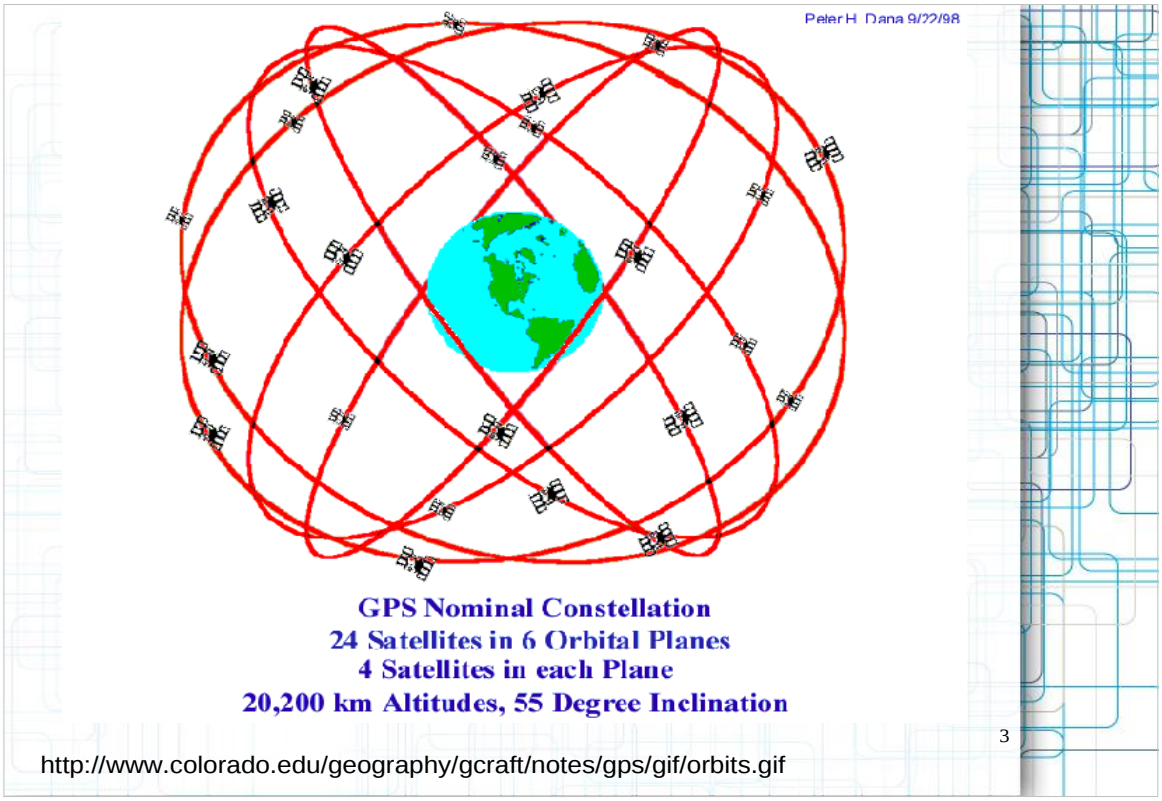
Ted Logan

Boulder Linux User's Group

10 June 2010

Just because you can find your position relative to some ideal model of your planet's surface doesn't mean you have any idea where you really are. The first part of this presentation discusses the operation of the Global Positioning System and some of the background mathematics behind coordinate systems. The second part of this presentation takes position data to answer three questions: “Where am I?”, “Where have I been?”, and “Where am I going?”

This is a GPS satellite. Wikipedia tells me there are 31 of them orbiting the planet (as of 2008).



The satellites are arranged such that, from any point on the Earth's surface, one should be able to see more than enough satellites to get a three-dimensional fix.

Math

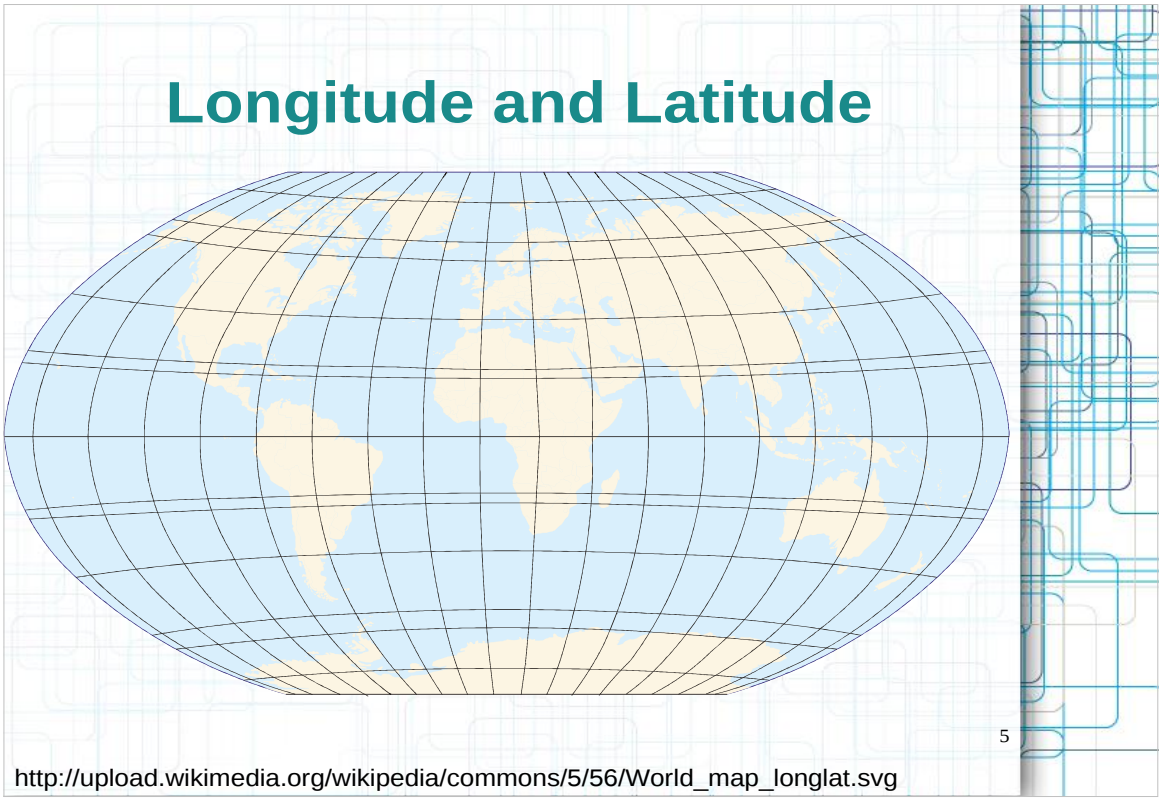
- Given:
  - Four satellites, known positions
  - Four time measurements
- Solve:
  - Your position
  - Your time

$$(x - x_i)^2 + (y - y_i)^2 - (z - z_i)^2 - ([tr_i + b - t_i]c)^2,$$

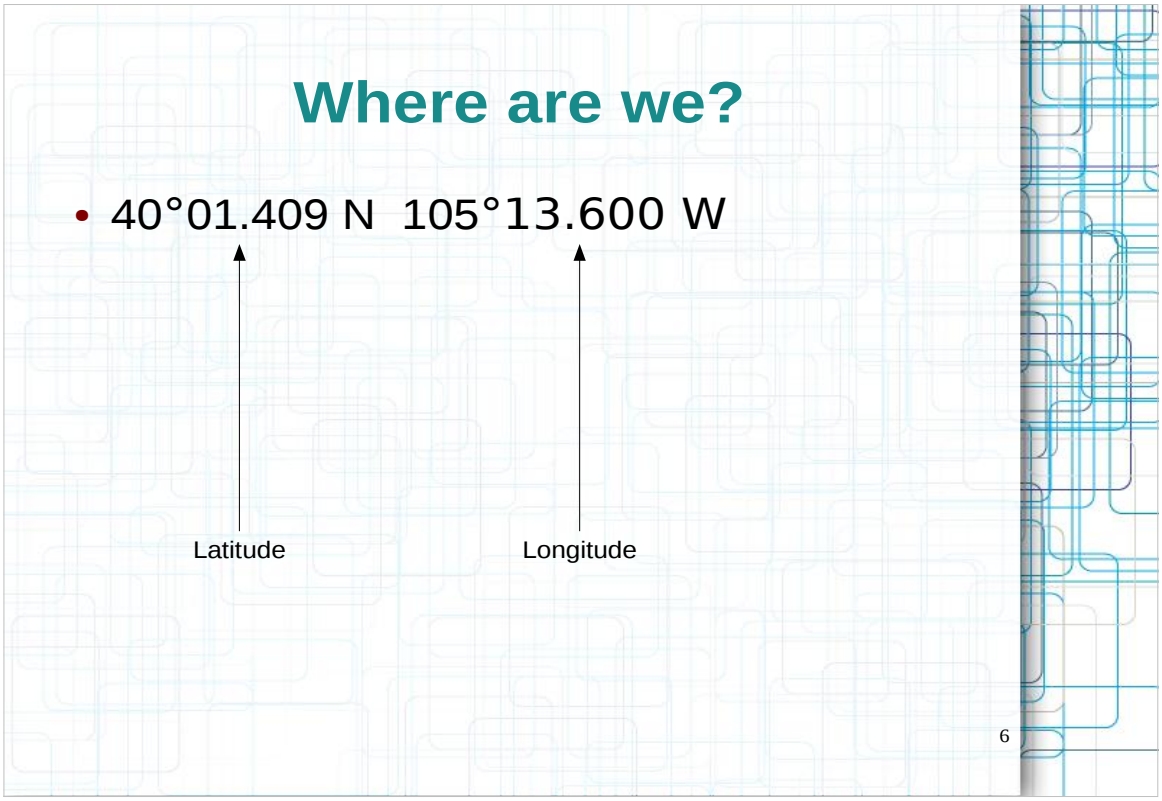
$$p_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} - bc, \quad i = 1, 2, 3, 4.$$

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

Each satellite broadcasts its orbital position and the time. If the receiver has a line-of-sight view of four satellites, it has enough information to solve for four unknowns: its position in space and time. (In reality, physics intervenes and nasty real-word phenomena like special relativity, atmospheric density, and multipath interference wreak havoc on GPS signals, so receivers try to get as many data points as possible for the best position. GPS satellites broadcast low-power microwaves, so they're blocked by water in tree leaves (and people), as well as mountains and buildings.)



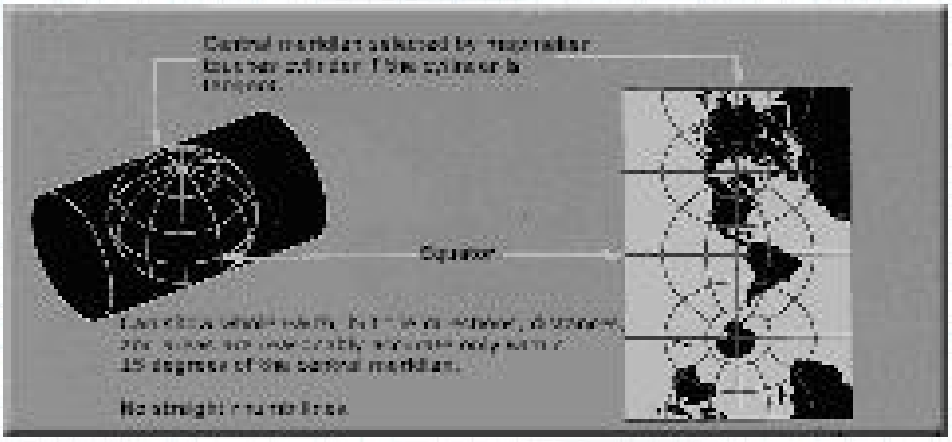
Once a GPS receiver knows where it is, it needs to represent that position in some meaningful coordinate system. Longitude and latitude are one popular coordinate system: latitude is the distance north or south of the equator, and longitude is the distance east or west of Greenwich.



This is the latitude and longitude of Aztek Networks, site of tonight's BLUG meeting.

There's one problem with polar coordinates – they're fine for crossing oceans, but don't make it easy to navigate on the ground. A degree of longitude is 111 km at the equator but only 79 km at 45 degrees north.

Transverse Mercator



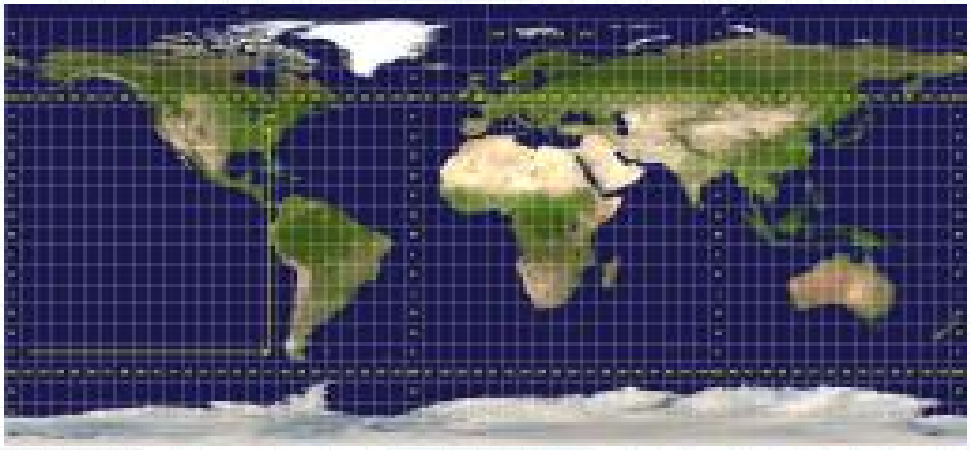
http://en.wikipedia.org/wiki/File:Usgs_map_traverse_mercator.PNG

For any arbitrarily-small chunk of ground, it's reasonable to pretend the planet's surface is flat, rather than the surface of an oblate spheroid. The Mercator projection does a good job of accurately representing the ground around the equator, but stretches out at the higher latitudes.

Take the Mercator cylinder and rotate it ninety degrees, so it stretches out on either side of the planet. The cylinder touches the planet at a narrow strip running north and south. Use this cylinder to map areas close to this north-south strip.

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

UTM Zones

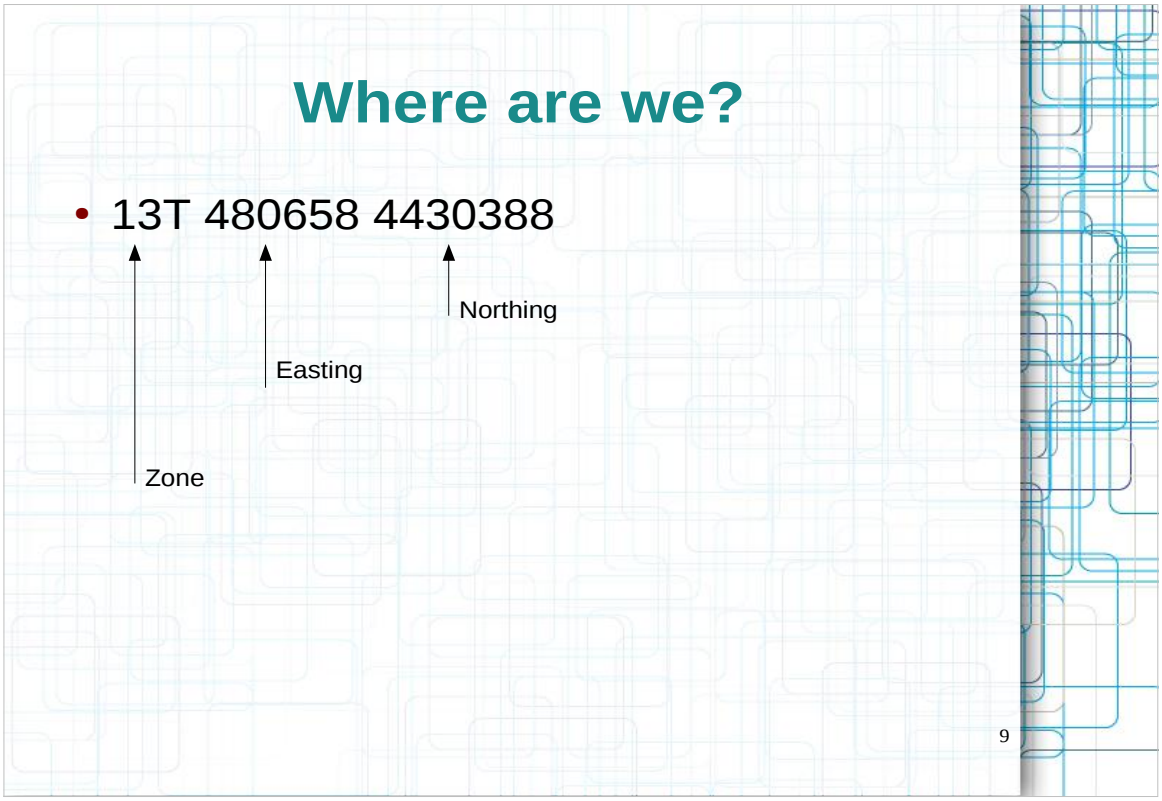


http://upload.wikimedia.org/wikipedia/commons/e/ed/Utm-zones.jpg

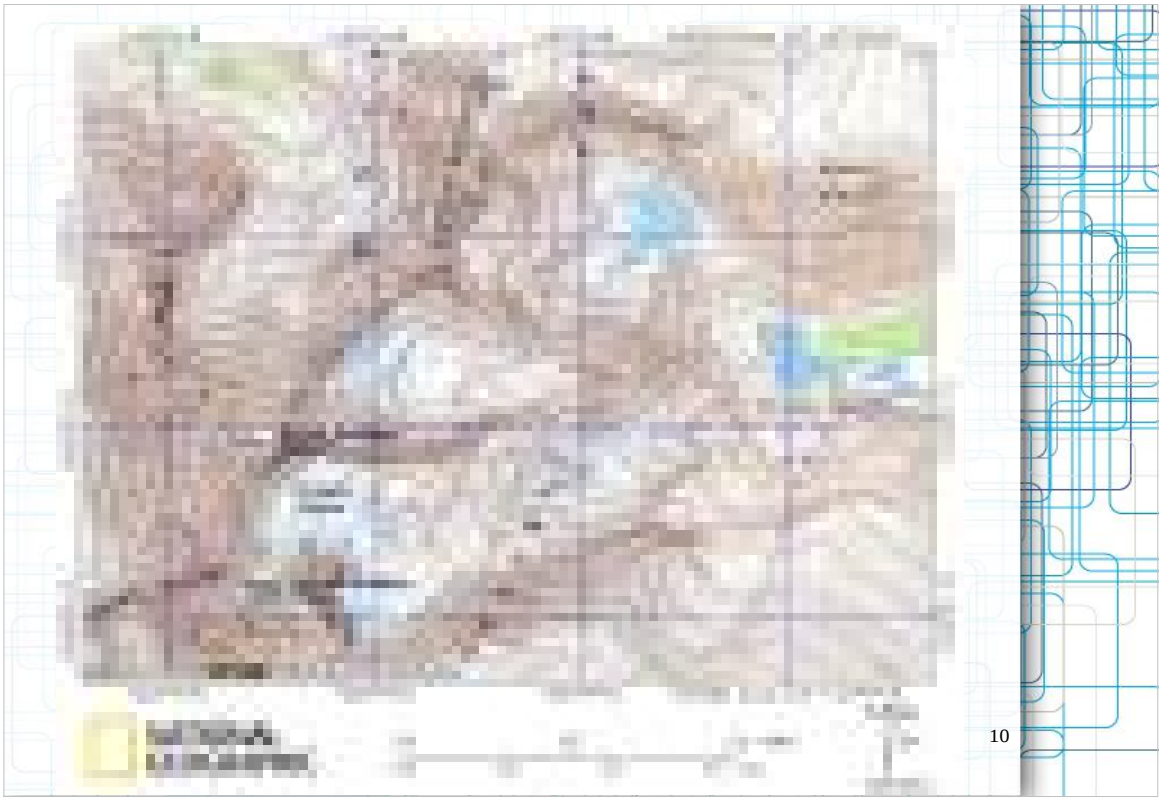
To map the entire planet, rotate the cylinder six degrees to get another strip. Continue rotating until you've covered the entire planet.

This is the full UTM grid of the entire planet. Each north-south strip is its own transverse Mercator cylinder. Grids are numbered east to west, and lettered north to south.

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



This is our meeting's coordinates in UTM. The “13T” is the zone; “13” is the 13th cylinder (counting from the left of the map); “T” represents how far north we are of the equator. Easting measures how many meters we are east of an arbitrary meridian, and northing measures how many meters we are north of the equator. Within the same zone, we can pretend the planet is perfectly flat and treat it just like an ideal Cartesian coordinate system. This makes mapping and navigating much easier.



Here's a concrete example of how UTM makes mapping easier: This is a topographic map of Boulder's very own Arapaho Glacier. The blue grid marks a one-kilometer grid; I can measure the distance between the grid lines to quickly find the coordinates for a given point, or to map given coordinates onto the map. A transparent square of plastic called a UTM grid reader makes this easy for printed maps with common scales.

Demo: Where am I now?

```
$GPBOD,,T,,M,,*47
$GPVTG,5.4,T,355.3,M,0.0,N,0.0,K*4F
$PGRME,5.5,M,8.4,M,10.0,M*13
$PGRMZ,5035,f*37
$PGRMM,WGS 84*06
$HCHDG,83.2,,,10.1,E*20
$GPRTE,1,1,c,*37
$GPRMC,024028,A,4010.7105,N,10504.2886,W,0.0,5.4,300610,10.
1,E,A*32
$GPRMB,A,,,,,,,,,,,,A,A*0B
$GPGGA,024028,4010.7105,N,10504.2886,W,1,07,1.5,1536.8,M,-
20.9,M,,*42
$GPGSA,A,3,,04,09,12,17,20,,27,28,,,,2.4,1.5,1.7*33
$GPGSV,3,2,12,17,55,046,46,20,14,061,42,26,00,235,00,27,48,276
,45*70
$GPGLL,4010.7105,N,10504.2886,W,024028,A,A*5B
```

11

Having covered the basics of GPS and mapping (and ignoring nasty details like the math necessary to do all this stuff, and reference ellipsoids), let's ask the first question I can answer with a GPS receiver, “Where am I?”

Many GPS receivers speak NMEA over serial. I can connect to my eTrex Vista via RS232, 4800 baud, and see a bunch of unintelligible text scroll by. I think I can see my longitude and latitude in there somewhere.

Demo: gpsd

```
$ gpsd /dev/ttyS0
$ telnet localhost 2947
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
d
GPSD,D=2010-06-30T02:52:28.0Z
q
GPSD,Q=7 0.00 2.50 1.30 2.88 5.24
p
GPSD,P=40.178491667 -105.071470000
a
GPSD,A=1531.900
```

Time and date

Number of satellites

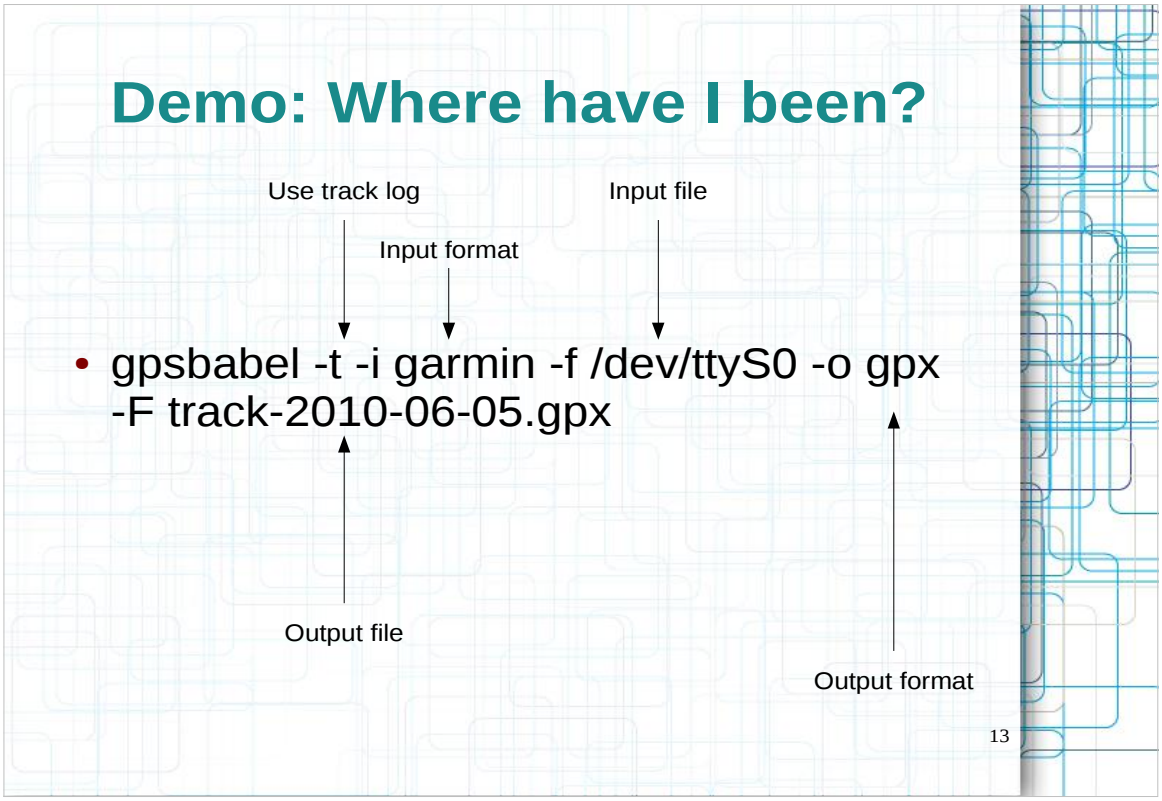
Position, in longitude and latitude

Altitude, in meters

Fortunately, I'm not stuck with writing my own NMEA parser. gpsd is a daemon that lets me query my current position via (scriptable) telnet. Most location-aware Linux applications use gpsd.

Here I type a one-letter command and get a machine-readable response from gpsd, while sitting in my back yard. You can copy and paste the longitude and latitude into your favorite mapping application to figure out where I live.

(My demo uses the older gpsd protocol, since that's what I happen to have installed on my notebook; please consult the relevant documentation for yourself.)



My second standard question is, “Where have I been?” I took my GPS receiver hiking the weekend before BLUG and asked it to record my track to its non-volatile memory. When I get back, I can connect my receiver to my computer via serial and download the track log. A fantastic program called gpsbabel converts many formats. This is my command line to download my track and save it as the human-readable, machine-parsable gpx format.

GPX file: track-2010-06-05.gpx

```
  <name>ACTIVE LOG #260</name>
<number>259</number>
<trkseg>
<trkpt lat="40.012764931" lon="-105.683648586">
  <ele>3685.832031</ele>
  <time>2010-06-05T17:53:25Z</time>
</trkpt>
<trkpt lat="40.012722015" lon="-105.683627129">
  <ele>3684.870605</ele>
  <time>2010-06-05T17:53:27Z</time>
</trkpt>
```


Here's a snippet of the GPX file. It's all well-formed XML, easily parsed by your favorite XML parser. This file is available in my presentation notes.

Demo: Google Earth

- gpsbabel -t -i gpx -f track-2010-06-05.gpx -o kml,points=0 -F track-2010-06-05.kml

Google Earth is not free-as-in-speech, but it's free-as-in-beer, and works well on Linux. It's also really cool. Google Earth has its own xml-based file format, kml (Keyhole Markup Language), which gpsbabel happens to speak. (These days, Google Earth actually uses gpsbabel to download GPS data, doing most of this demo magically behind the scenes.)

Fourth of July to Mount Neva

This is my hike from Fourth of July Trailhead to Arapaho Pass, and my snow-climb from Lake Dorothy to Mount Neva.

Mount Neva: Juliet

I can zoom in arbitrarily close, and compare the amount of snow I saw in early June to the amount of snow in the aerial photo from July in a previous year. (Lake Dorothy was frozen, and snow covered the entire north slope of Mount Neva.) We can also see where my batteries died on my descent, halfway down the north slope.

Demo: Bolder Boulder Splits

```
forerunner-305$ garmin_save_runs
Extracting data from Garmin Forerunner305 Software Version 2.90
Files will be saved in '/home/jaeger/gps/forerunner-305'
...
Skipped: /home/jaeger/gps/forerunner-305/2010/05/20100525T125916.gmn
Skipped: /home/jaeger/gps/forerunner-305/2010/05/20100529T112922.gmn
Wrote:   /home/jaeger/gps/forerunner-305/2010/05/20100531T072448.gmn
```


I also have a Garmin Forerunner 305, which I take running with me. It records my track, but also useful (for runners, anyway) things like mile splits and heartrate.

I can use gpsbabel to download the tracklog, but the garmintools package implements more of the Garmin protocol and lets me see my mile splits and heartrate at each point it records.

Raw Forerunner data

```
forerunner-305$ xxd 2010/05/20100531T072448.gmn | head
0000000: 3c40 6741 724d 694e 403e 0000 6400 0000  <@gArMiN@>..d...
0000010: 3b63 0000 0100 0000 3363 0000 5601 0000  ;c......3c..V...
0000020: 0300 0000 5601 0000 f103 0000 a902 0000  ....V...........
0000030: 9e00 4903 4f03 0000 0000 0000 0000 0000  ..I.O...........
0000040: 2200 0000 0000 0000 0000 0000 0500 0000  "...............
0000050: 0010 0010 b37c 1a00 ffff ffff ffff ffff  .....|..........
0000060: 0000 0005 02ff 0000 0000 0000 0000 0000  ................
0000070: 0400 0000 0c36 2000 c501 ffff ffff ffff  .....6 .........
0000080: 4906 0001 02ff 0000 009f 0a04 cc35 2000  I............5 .
0000090: 286d 1a04 f05d 2000 ffff ffff ffff ffff  (m...] .........
forerunner-305$ garmin_dump 2010/05/20100531T072448.gmn | head
<run type="1009" track="158" sport="running" multisport="no">
 <laps first="841" last="847"/>
</run>
<lap type="1015" index="841" start="2010-05-31T07:24:48-06:00"
duration="0:07:15.51" distance="1609.34399" trigger="distance">
 <begin_pos lat="40.03350821" lon="-105.25376676"/>
 <end_pos lat="40.02222240" lon="-105.25935531"/>
 <max_speed>4.42204285</max_speed>
 <calories>137</calories>
 <avg_hr>158</avg_hr>
 <max_hr>169</max_hr>
```


garmin_save_runs outputs a binary format, but garmin_dump outputs a format very much like xml that (with a bit of tweaking) I can parse it with my favorite xml parser.

Parsed Forerunner data

```
forerunner-305$ ~/src/perl/show_runs.pl 2010/05/20100531T072448.gmn
2010-05-31 (Monday)

Lap      Distance  Pace      Time      Avg Hrt  Max Hrt
         (miles)   (min/mi)  (min)     (bpm)    (bpm)
------   --------  --------  -------   -------  -------
Lap  1      1.00      7:15      7:15       158      169
Lap  2      1.00      7:37      7:37       170      182
Lap  3      1.00      8:01      8:01       169      181
Lap  4      1.00      7:59      7:59       167      174
Lap  5      1.00      8:00      8:00       167      173
Lap  6      1.00      7:42      7:42       171      183
Lap  7      0.30      7:25      2:13       176      179
         --------  --------  -------
Total       6.30      7:44     48:49
```


I wrote a Perl script to parse the output into something human-readable. Here are my splits from the last Bolder Boulder, where we see my average and maximum heartrate for each mile of the race. For most of the race, I ran just below my lactate-threshold intensity at about 170 beats per minute. (This sort of thing is useful when training, when I try to run at a particular intensity for a particular distance.)

Demo: Where am I going?

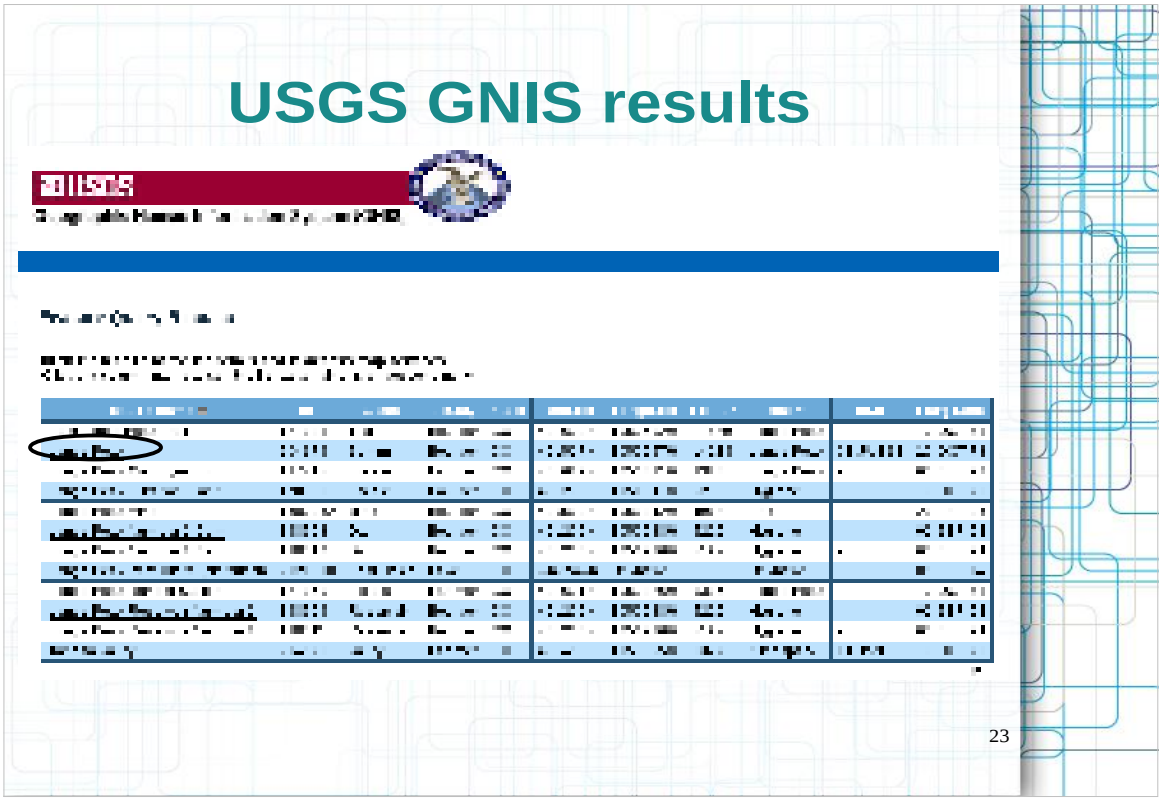
- U.S. Board on Geographic Names: http://geonames.usgs.gov/pls/gnispublic/

The third question I can answer is, “Where am I going?”

The US Geological Survey keeps a public, web accessible database of geographic points. I can query the website to find the official names and locations of various points of interest.

USGS GNIS search

Let's search for my favorite fourteener, Longs Peak.



The search came back with twelve results for various things that relate, in some way, to Longs Peak. The summit is circled.

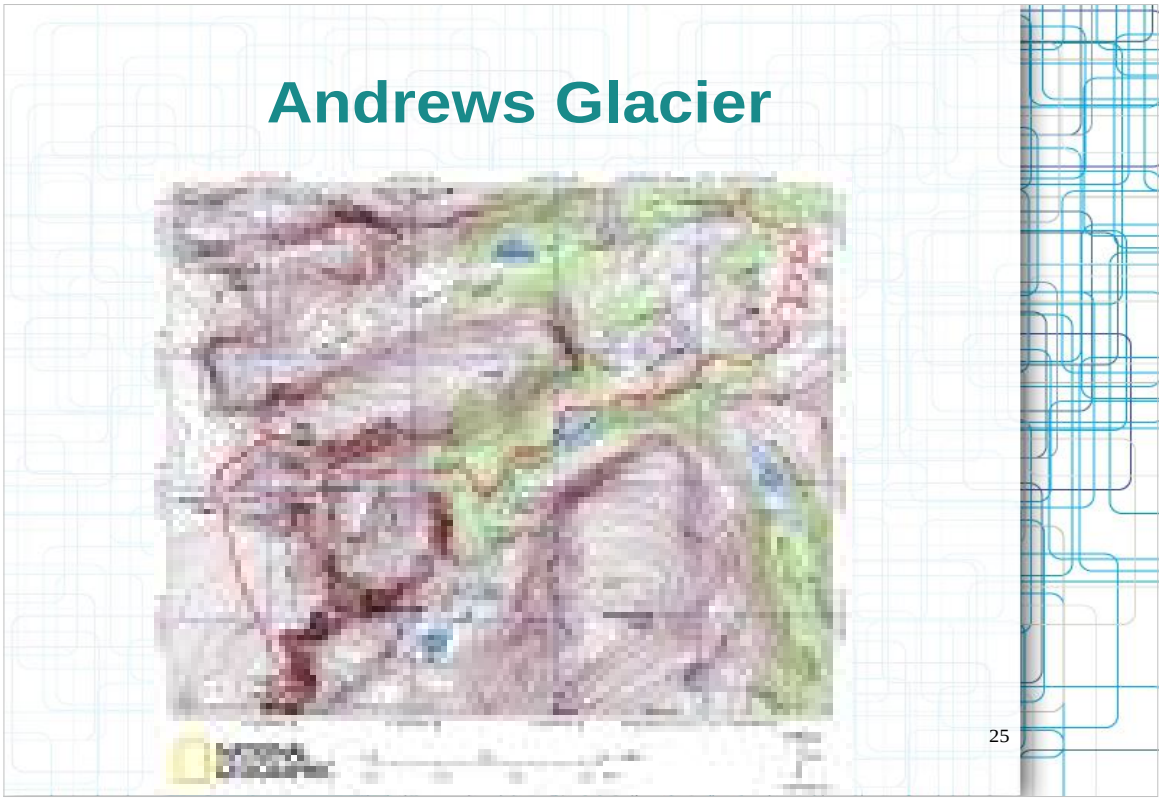
USGS GNIS Coordinates

Coordinates (One point per USGS topographic map containing the feature, NAD83)

Sequence	Latitude(DEC)	Longitude(DEC)	Latitude(DMS)	Longitude(DMS)	Map Name
1	40.2547069	-105.6158365	401517N	1053657W	Longs Peak

Clicking on the result gives me more information, including the longitude and latitude in decimal degrees, which I can enter in my favorite mapping program or enter directly into my GPS receiver as a waypoint. I can use the waypoint to navigate on the ground to guide me to the destination.

Of course, on the ground, terrain often intervenes to make blindly following waypoints inadvisable. If one were to start at the Longs Peak Trailhead and follow the waypoint straight to the summit, one would end up climbing over Mount Lady Washington and down into Chasm Lake before staring up the East Face. Proper route-finding involves following a detailed route with many waypoints. gpsbabel can help.



Here's another weekend snow-climb: Andrews Glacier in Rocky Mountain National Park. I traced the trail in National Geographic's Topo software, which costs actual money (and runs under Windows) but includes scans of USGS quads (and runs fine under VMWare, as long as I don't try to run too many memory-hungry applications at once). Topo has its own proprietary binary format, but gpsbabel can read it just fine.

(I ran into trouble during my presentation while trying to run the demo in real time, when my memory manager overcommited memory between Google Earth and VMWare and spent the next ten minutes thrashing, which was a bit embarrassing)

Demo: Create a route (1)

- gpsbabel -t -i tpo3 -f andrews_glacier.tpo -o gpx -F andrews_glacier.gpx
- gpsbabel -t -i gpx -f andrews_glacier.gpx -o kml,points=0 -F andrews_glacier.kml

Having created the track in Topo, I use gpsbabel to convert the track from Topo's native, binary format to Google Earth's native format, with an interim stop at gpx. (If I were to convert straight from Topo to kml, gpsbabel drops the track points for reasons I haven't fully uncovered.)

Andrews Glacier in Google Earth

Here's the track shown in Google Earth. Andrews Glacier is the large white patch toward the right of the image.

Demo: Create a route (2)

- gpsbabel -i gpx -f andrews_glacier.gpx -x transform,rte=trk,del -x simplify,error=0.01k -o kml -F andrews_glacier_route.kml

The track still has too many points to make a worthwhile GPS route. I can use gpsbabel to convert the track to a route, then apply the “simplify” filter to remove route points that introduce less than 10 meters of error. This cuts the number of intermediate route points down to a number that I can actually use for navigation in the field.

Route in Google Earth

Here is the simplified route shown in Google Earth. Each box is an intermediate route point, and at no point will the route deviate from the map I traced by more than ten meters.

Demo: Create a route (3)

- gpsbabel -i gpx -f andrews_glacier.gpx -x transform,rte=trk,del -x simplify,error=0.01k -o garmin -F /dev/ttyS0

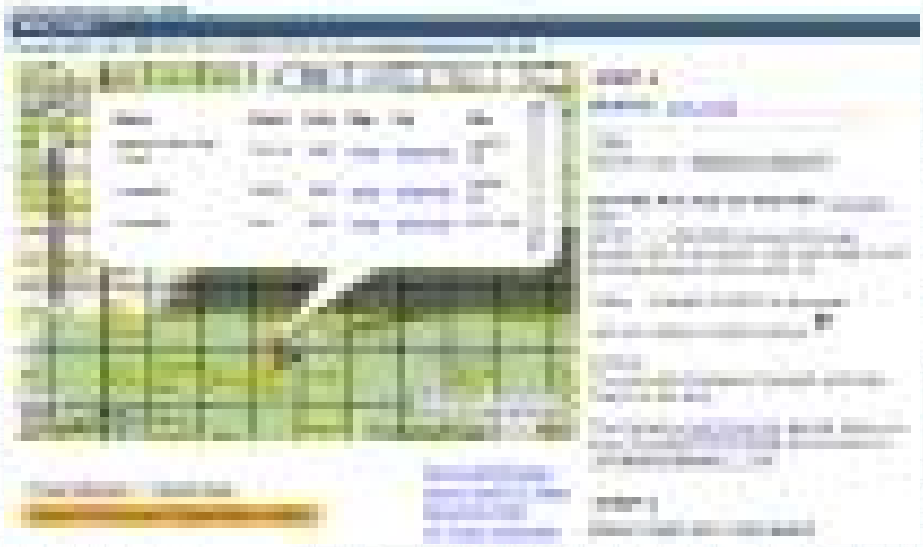
Finally, I can upload the track onto my GPS receiver using the Garmin protocol over my serial port.

Demo: USGS Maps

- http://store.usgs.gov

The good people at the US Geological Survey post full PDFs of all of their maps. Which, being works of the United States federal government, are public domain. Rip, Mix, Enjoy. Click “Map Locator”.

USGS Map Download



I've zoomed in and found an interesting map somewhere in the middle of Colorado, and the website offers me the PDF.

USGS topo

Here's a zoomed-in shot from the PDF I downloaded from the USGS website. Exactly what I'd find on the physical copy, in an attractive digital format. And did I mention it's free, and public domain?

Topo demo

```
$ topo_tile.pl CO_Mount_of_the_Holy_Cross_1987_geo.pdf
Rasterizing CO_Mount_of_the_Holy_Cross_1987_geo.pdf at 200 dpi

PDF rasterized as:
        CO_Mount_of_the_Holy_Cross_1987_geoKxkW.png

Enter state: [CO]
Enter quad name: [Mount of the Holy Cross]
Enter pixel coordinates of top-left corner: 417,226
Enter pixel coordinates of top-right corner: 3942,226
Enter pixel coordinates of bottom-left corner: 415,4808
Enter pixel coordinates of bottom-right corner: 3948,4803
Enter latitude of the northern edge of the map: 39 30
Enter latitude of the southern edge of the map: 39 22 30
Enter longitude of the eastern edge of the map: -106 22 30
Enter longitude of the western edge of the map: -106 30


Image should be rotated by 0.03280 degrees
Crop geometry: 3529x4579+414+224
Rotating and cropping CO_Mount_of_the_Holy_Cross_1987_geoKxkW.png (to
/tmp/FLCkb804Ws/CO_Mount_of_the_Holy_Cross_1987_geoKxkW.jpg)...
done.
Generating CO_Mount_of_the_Holy_Cross_1987_geoKxkW.kmz ...
  adding: CO_Mount_of_the_Holy_Cross_1987_geoKxkW.jpg (deflated 0%)
  adding: doc.kml (deflated 45%)
Done.
```


This didn't make it into my presentation, but I want to show it off here anyway. I wrote a Perl script to take the PDFs of USGS topographic maps and import them into Google Earth, so I can superimpose the maps onto the terrain and other things I have on the map.

The script uses ImageMagick to rasterize the PDF, asks the user to locate the corners of the map in pixels and in geographic coordinates, crops and rotates the image, and writes a kml file that overlays the map at the right place on the planet.

At 200 dpi, a full 7.5-minute quad has a lot of pixels. Manipulating it using my script takes about a gig of free memory on my machine, and would bring it to its knees if I didn't have ample ram.

Topo demo: Google Earth

Here's a screenshot of Google Earth viewing the same topographic map. I can zoom in and out, and trace trails and routes at will, layering the map with everything else I want to show.

References

- http://www.gpsbabel.org/
- http://code.google.com/p/garmintools/
- http://earth.google.com/
- http://www.natgeomaps.com/topo.html

All the files used in my demos are available at
http://tedlogan.com/blug-2010-06/

Thanks!

ted.logan@gmail.com