GPS Post Processing System

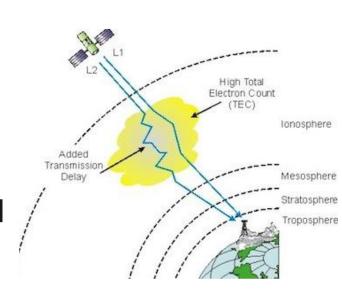
Dominic Fuller-Rowell, Michael Brandt, Waverly Hinton

General Principle

We set out to create a program that would apply trilateration corrections to a set of GPS data to provide a more accurate report of a person's location.

Differential GPS

- Handheld GPS receivers suffer from a number of sources of error.
- Two GPS receivers in the same geographical area suffer from the same sources of error.
- The difference between the calculated and true positions of two receivers is the same!
- Δ position1(X,Y,Z) == Δ position2(X,Y,Z)

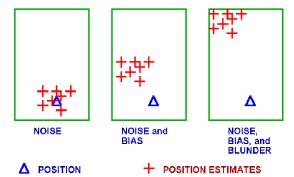


Array of Base Stations



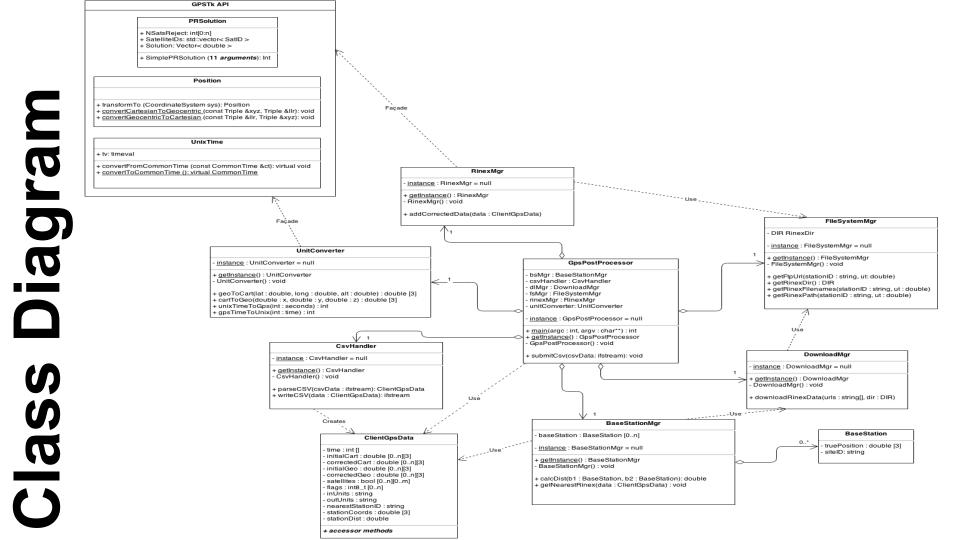
- CORS/NGS provide RINEX GPS data for a vast array of dual frequency commercial receivers.
- GPS Toolkit (C++) provides the means to utilize the RINEX data.
- Our code interfaces GPS Toolkit with CORS to provide more accurate postprocessed geographical data.





Design Patterns used

- Singleton GpsPostProcessor.cpp
- Façade UnitConverter.cpp and RinexMgr. cpp both act as Façades to the GPSTk API



<u>a</u>

UnitConverter A: double WGS84Ellipsoid.a() - eccSq : double = WGS84Ellipsoid.eccSquared() + geoToCart(lat : double, long : double, alt : double) : vector<double> + cartToGeo(double : x, double : y, double : z) :vector<double> Facade **GPSTk API** Position + convertCartesianToGeodetic (const Triple &xyz, Triple &llr): void + convertGeodeticToCartesian (const Triple &IIr, Triple &xyz): void Triple + Triple(): Triple + Triple(double a, double b, double c): Triple WGS84Ellipsoid + a(): double + eccSquared(): double

Demo

OMG it's a demo

Lessons Learned

- git commit often
- C++ experience
- time spent planning helps guide coding
 - 1 minute spent planning saves 2 minutes coding

Going Forward

- integrate Rails app with C++ code
- make real time corrector
- flesh out the Android app