goGPS

a navigation software to enhance the accuracy of low-cost GPS receivers

Eugenio Realini
Osaka City University, Japan

Mirko Reguzzoni
OGS c/o Politecnico di Milano, Italy

Oct. 21st
FOSS4G2009
Why goGPS?

- Diffusion of GPS-enabled devices (smartphones, PDAs, cameras)
- Miniaturization of low-cost and low-power GPS modules and antennas
- !!! BAD ACCURACY !!! (~ 5 meters)
Basic ideas behind goGPS

- develop a tool which allows to modify Kalman filter parameters to study new approaches to GPS navigation (not possible with black-box commercial algorithms)
- exploit the availability of networks of permanent GPS stations and wireless connectivity to enhance the navigation accuracy of low-cost GPS devices

Kinematic surveying
Precise off-road navigation
Location Based Services
Low-cost mapping
Italy ← goGPS → Japan

Geomatics Laboratory
Politecnico di Milano
Como Campus

Media Center
Osaka City University
Double freq. vs single freq.

GPS satellites broadcast signal on two carriers: **L1** and **L2**

- **High-end professional receivers** use both **L1** and **L2** (double frequency receivers)
  - Accuracy: 2-3 cm (real-time)

- **Low-cost commercial receivers** use **just L1** (single frequency receivers)
  - Accuracy: 3-5 m (real-time)
RTK vs stand-alone

Real-Time Kinematic (RTK) positioning (via Internet connection)

2-3 cm (real-time)

Stand-alone (but nowadays it is easy to add Internet access)

3-5 m (real-time)

nothing in between?

US $ 15000 - 25000

US $ 70 - 150
DGPS vs stand-alone

Single frequency Differential GPS (DGPS) positioning (via Internet connection)

15-30 cm (real-time)

Stand-alone (but still they cost much less)

3-5 m (real-time)

nothing in between!

US $ 1000 - 1500

US $ 70 - 150
goGPS niche

- RTK: 2-3 cm
- DGPS: 15-30 cm
- Stand-alone: 3-5 m
- US $ 15000 - 25000
- US $ 1000 - 1500
- US $ 70 - 150

L1 RTK positioning: 40 cm - 1 m
Raw GPS data!

To apply RTK, raw GPS observations are needed!

u-blox Evaluation Kits (AEK-4T / EVK-5T)

goGPS: L1 RTK positioning (via Internet connection)
goGPS system design

Data acquisition

Data processing (Kalman filter)

Display & log result
It is the core of the software.

It updates the position of the receiver in real-time on the basis of:

- new measurements
- the state of the system at the previous epoch

To implement it, it is needed to define:

- state variables
- dynamic model
- observations
In order to improve the height positioning quality, a new observation from a DTM is introduced:

\[ h_{\text{DTM}} = h(x_r, y_r, z_r) + v_{\text{DTM}} \]

\[ \sigma_v \approx 30 \text{ cm} \]

A DTM obtained from a LiDAR DSM 2m x 2m produced by Lombardy Region (Italy) was used during tests.

DTM loading time was optimized by subdividing the DTM in buffered tiles.

approx. position → Tile search → detection of the 4 nearest vertices → Interpolation → KF
Constrained motion

If the rover is moving along a path that is known a priori (e.g. road, railway, ...) a linear constraint can be introduced, making the motion mono-dimensional.

The constraint is modeled as 3D interconnected segments and the motion is described by a curvilinear coordinate \(c\):

\[
\begin{bmatrix}
    c_0 \\
    \vdots \\
    c_{r-1} \\
    N_{rm}^{p_1} \\
    \vdots \\
    N_{rm}^{p_{32}}
\end{bmatrix}
\]

New state variable: \(X_t = (X_0, Y_0, Z_0)\)
- developed in MATLAB environment

- 1 Hz data acquisition rate by means of “Instrument Control” toolbox (standard TCP-IP and USB)

- real-time update of receiver position (computation time about 15 ms on Intel Centrino CPU 1.66 GHz)

- real-time update of the position with respect to a known reference or on Google Earth

- Post-processing (post-mission) analysis by means of RINEX files or goGPS data saved during a real-time test
TIMING
epoch 1: GPStime=1517:322088

ROVER DATA
u-blox: 1.3622 sec (232 bytes --> 232 bytes)
decoding: 1.4563 sec (#1 messages)
GPStime=322088 (9 satellites)
P1 SAT: 02 04 07 13 16 20 23 25 32
L1 SAT: 04 07 13 16 20 23

MASTER DATA
irealp: 1.5365 sec (182 bytes --> 182 bytes)
decoding: 1.6896 sec (1019 1002 1006)
GPStime=322088 (8 satellites)
P1 SAT: 02 04 07 08 13 20 23 25
L1 SAT: 02 04 07 08 13 20 23 25

BUFFER (ROVER): ooooooooooooooooooox
BUFFER (MASTER): oooooooooooooooooox

POSITIONING
no position/velocity are computed
EPH SAT: 13
OBS SAT: 13
Hardware/1

Internet

Base station(s)

NTRIP (RTCM 3.1)

USB
goGPS

3G

Internet
device with:
- goGPS
- GPS receiver
- wireless internet
eBonTek egps597
chipset: ANTARIS 4
signal: C/A, L1

Provides NMEA in output over a Bluetooth connection. Stand-alone positioning.

u-blox AEK–4T
chipset: ANTARIS 4
signal: C/A, L1

It has an external patch antenna and it provides raw data and/or processed data (NMEA format) by USB connection. Its parameters are fully customizable.
Leica GPS1200
signal: C/A, P, L1, L2

Double frequency receiver with RTK capabilities.

Leica GS20
signal: C/A, L1

Mid-level receiver (single freq.), designed for cartographic update and quick decimeter-level surveys. It supports DGPS positioning.
Como permanent station, used as base station (through GPSLombardia positioning service)
Devices:
- Leica GPS1200
- Leica GS20
- eBonTek eGPS 597
- ev. kit u-blox + goGPS

Fixed on the rooftop of a car driven on a road with **good** sky visibility.
Devices:
- Leica GPS1200
- Leica GS20
- eBonTek eGPS 597
- ev. kit u-blox + goGPS

---

goGPS
(cutoff = 10°)

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>std</th>
</tr>
</thead>
<tbody>
<tr>
<td>goGPS</td>
<td>1.13 m</td>
<td>0.67 m</td>
</tr>
</tbody>
</table>

---

goGPS
(cutoff = 30°)

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>std</th>
</tr>
</thead>
<tbody>
<tr>
<td>goGPS</td>
<td>0.78 m</td>
<td>0.47 m</td>
</tr>
</tbody>
</table>

---

Leica GS20
(mod. “Max Accuracy”)

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>std</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>0.30 m</td>
<td>0.15 m</td>
</tr>
<tr>
<td>std</td>
<td></td>
<td>1.70 m</td>
</tr>
</tbody>
</table>

---

eBonTek

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>std</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>4.03 m</td>
<td></td>
</tr>
<tr>
<td>std</td>
<td></td>
<td>1.70 m</td>
</tr>
</tbody>
</table>
goGPS conversion from MATLAB to C/C++

MATLAB → C

new (C++)

- goGPS GUI
- goGPS core

already existing (C)

- RTKLIB by T.Takasu
  (GPLv3)

- positioning func.
- read/write
- plotting
  (...)

http://gpspp.sakura.ne.jp/rtklib/rtklib.htm
goGPS development could also include WPS functionality, to shift the computational / storage burden from the rovers to a central server.

Examples:

Server providing DTM data interpolation as WPS

Server running goGPS with WPS functionality

DTM height

rough position

accurate positioning

raw code and phase observations

Rovers running goGPS

Rovers just acquiring raw data
goGPS MATLAB code:

http://sourceforge.net/projects/gogps

Thank you!