O S A K A   C I T Y   U N I V E R S I T Y

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?

OSAKA CITY UNIVERSITY

miniaturization of low-cost and low-power GPS modules and antennas

diffusion of GPS-enabled devices (smartphones, PDAs, cameras)

!!! 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

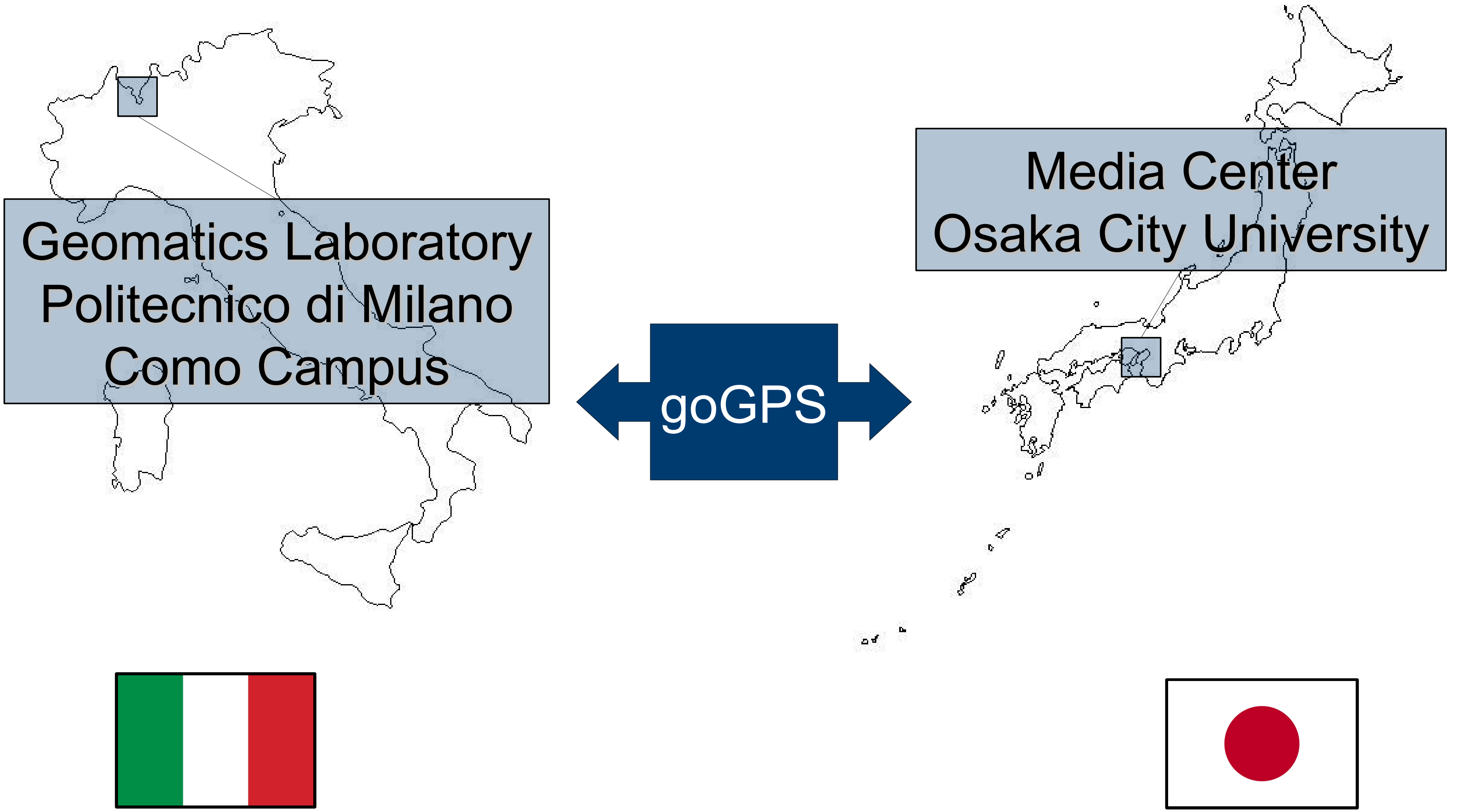
Kinematic surveying

Precise off-road navigation

Location Based Services

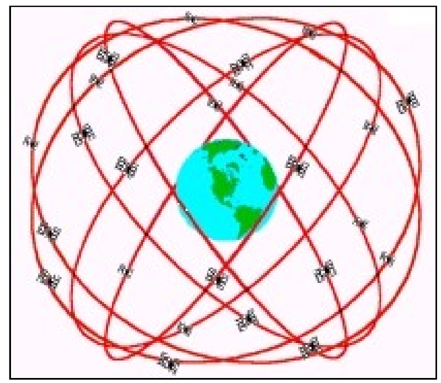
Low-cost mapping

Italy ← goGPS → Japan



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)

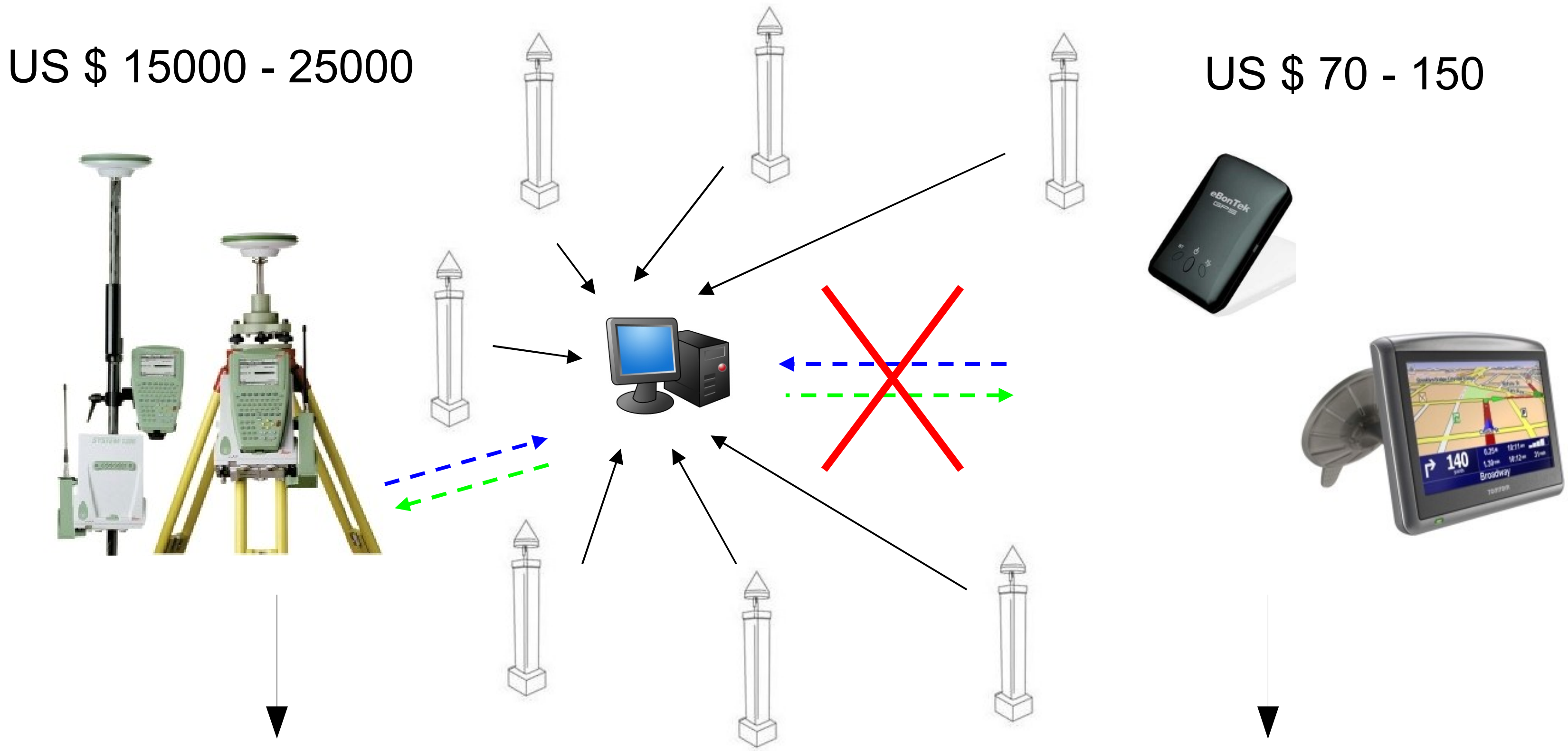
Accuracy: 2-3 cm (real-time)

Low-cost commercial receivers use just L1 (single frequency receivers)

Accuracy: 3-5 m (real-time)

RTK vs stand-alone

OSAKA CITY UNIVERSITY



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

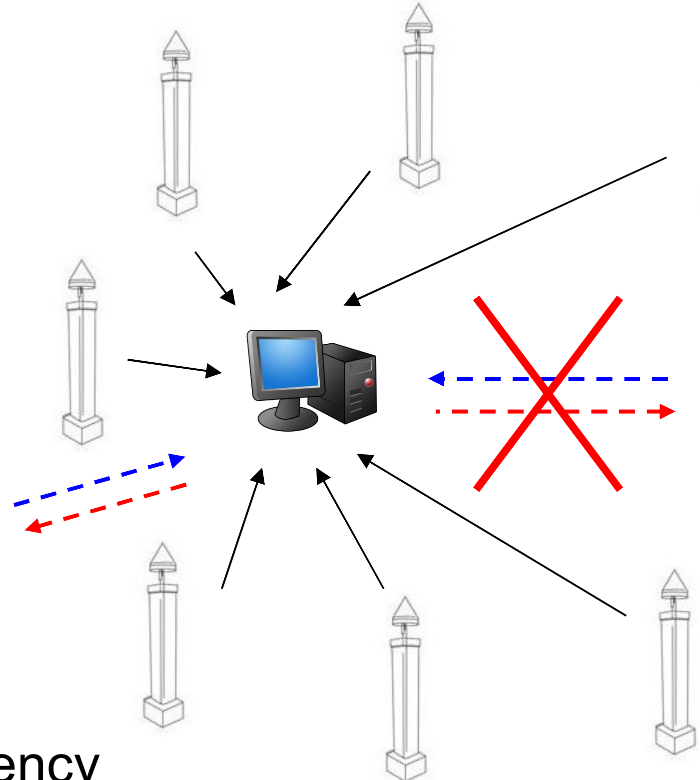
Stand-alone (but nowadays it is easy to add Internet access)

2-3 cm (real-time) ⊢—— *nothing in between?* ——⊣ 3-5 m (real-time)

DGPS vs stand-alone

US $ 1000 - 1500

US $ 70 - 150



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

Stand-alone
(but still they cost much
less)

15-30 cm (real-time) |—— *nothing in between!* ——| 3-5 m (real-time)

goGPS niche

US $ 15000 - 25000

2-3 cm

RTK

US $ 1000 - 1500

15-30 cm

DGPS

US $ 70 - 150

3-5 m

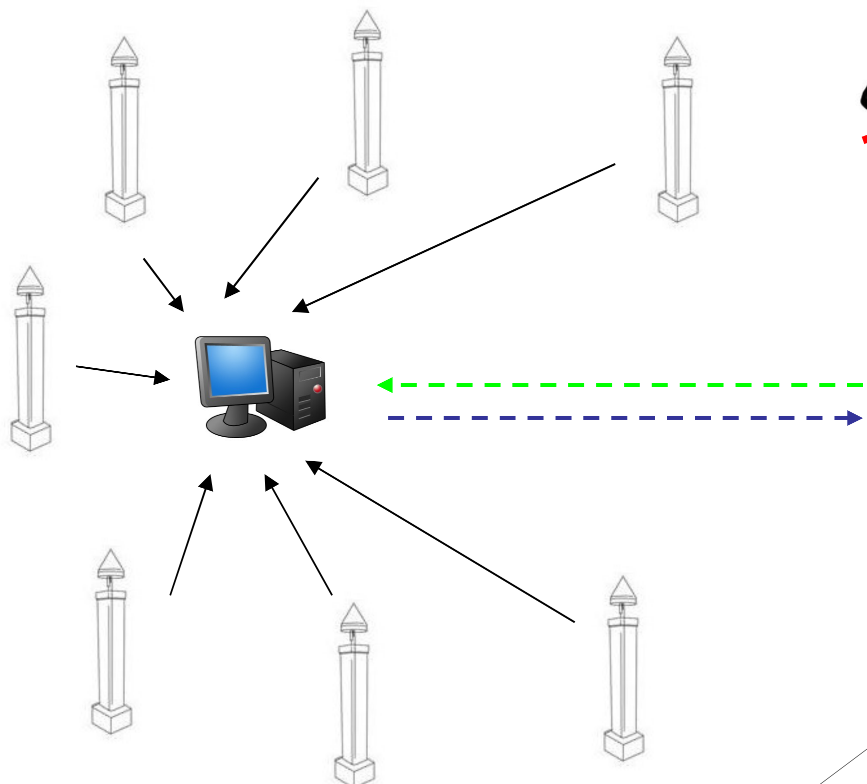
Stand-alone

40 cm - 1 m

goGPS
L1 RTK positioning

Raw GPS data!

OSAKA CITY UNIVERSITY

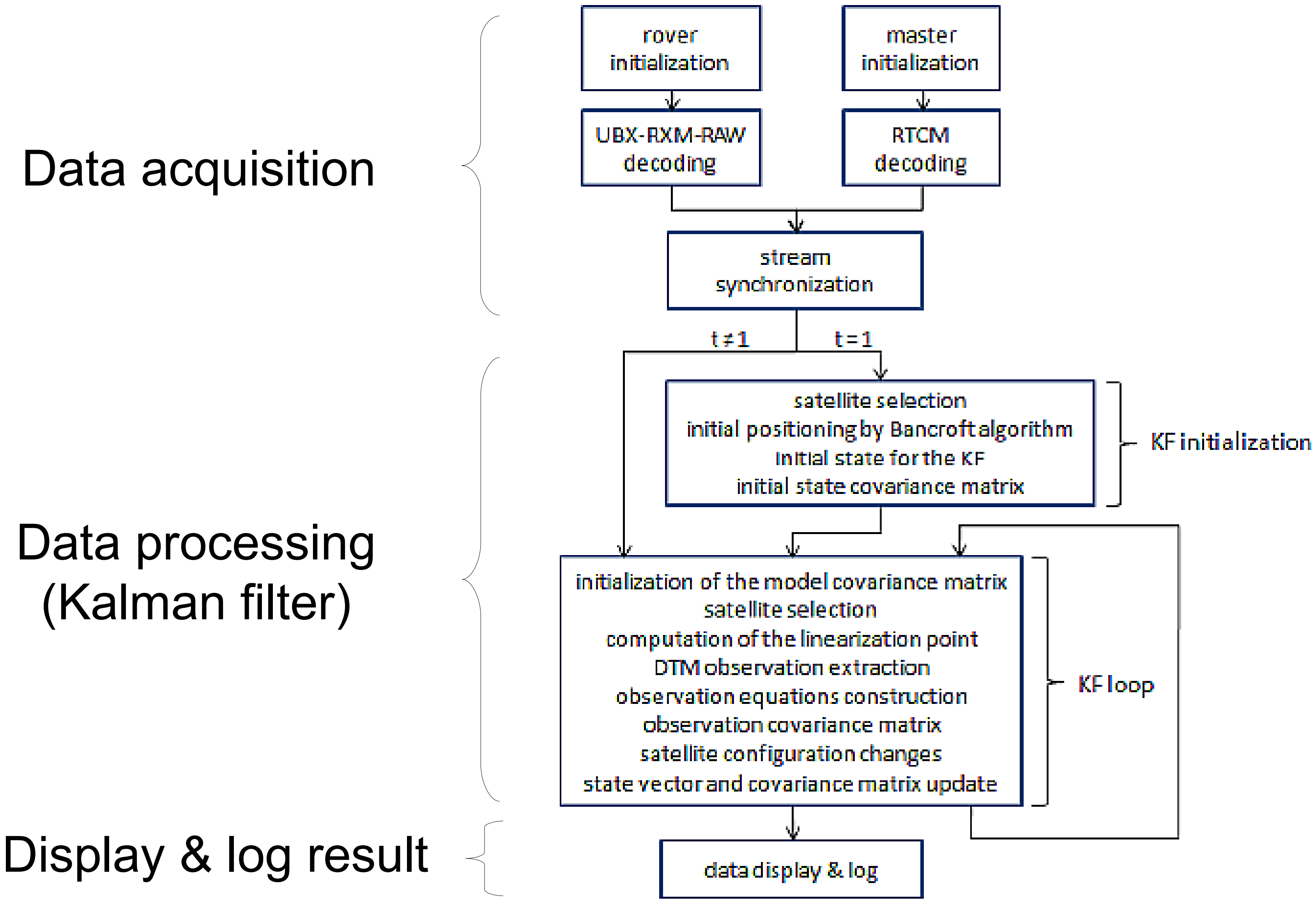


u-blox Evaluation Kits
(AEK-4T / EVK-5T)

To apply RTK
raw GPS observations
are needed!

goGPS: L1 RTK positioning
(via Internet connection)

goGPS system design



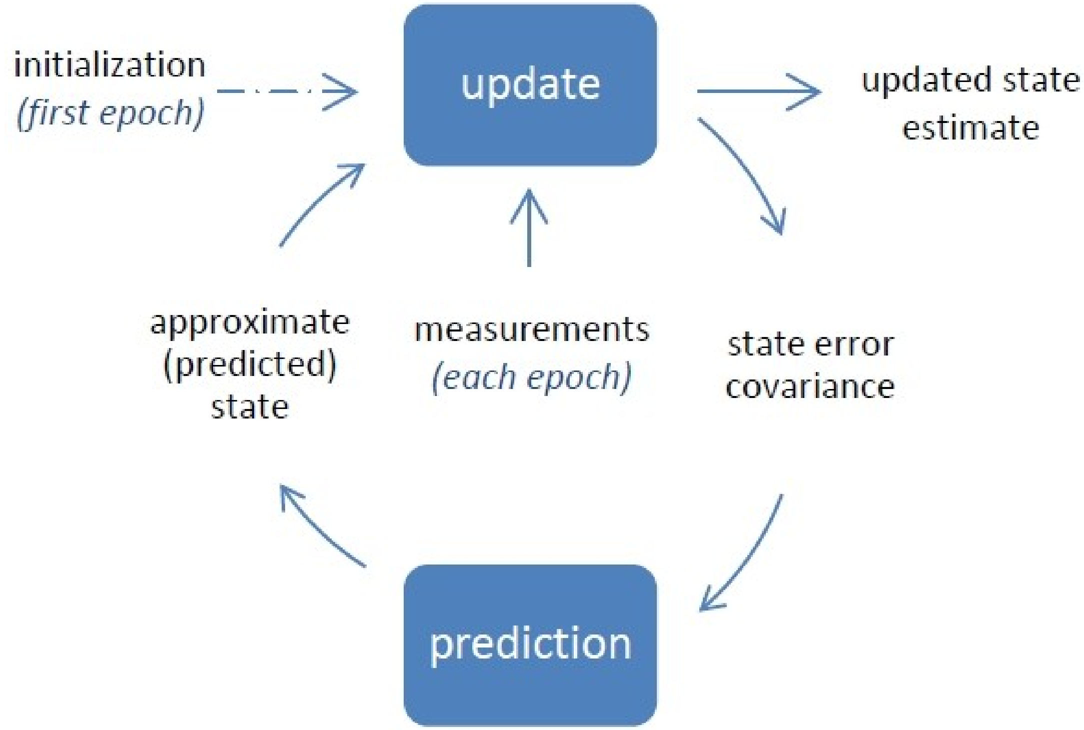
Kalman filter/1

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

DTM observation

In order to improve the heigth positioning quality, a new observation from a DTM is introduced:

$$h_{DTM} = h(x_r, y_r, z_r) + v_{DTM}$$

$\sigma_v \approx 30$ 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.

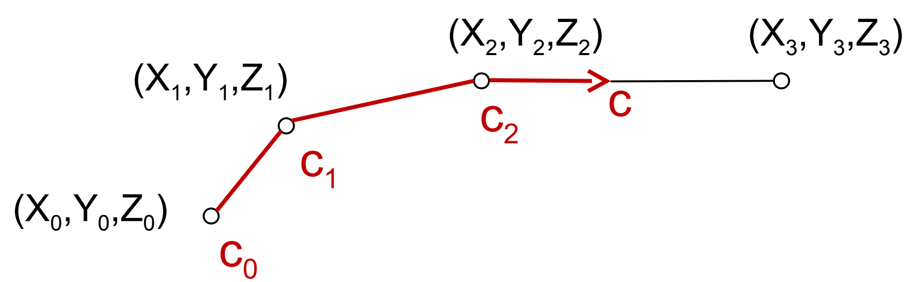
approx. position → Tile search → detection of the 4 nearest vertices → Interpolation → KF

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):

new state variable: $\underline{X}_t = \begin{bmatrix} c_r \\ \dot{c}_r \\ \vdots \\ N_{rm}^{p1} \\ \vdots \\ N_{rm}^{p32} \end{bmatrix}$



Software/1

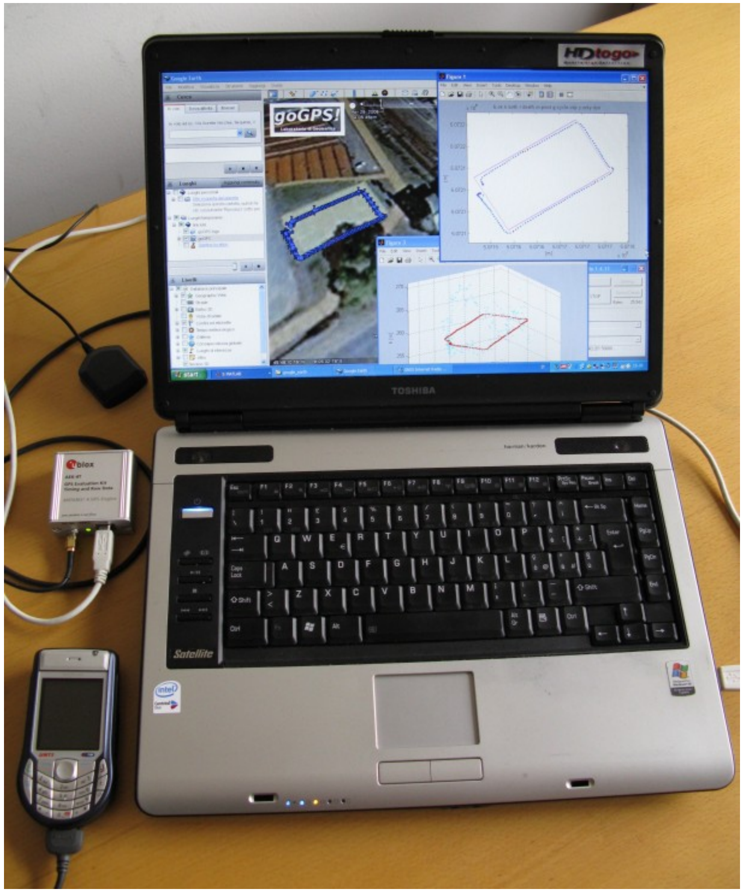
- developed in MATLAB environment

- 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

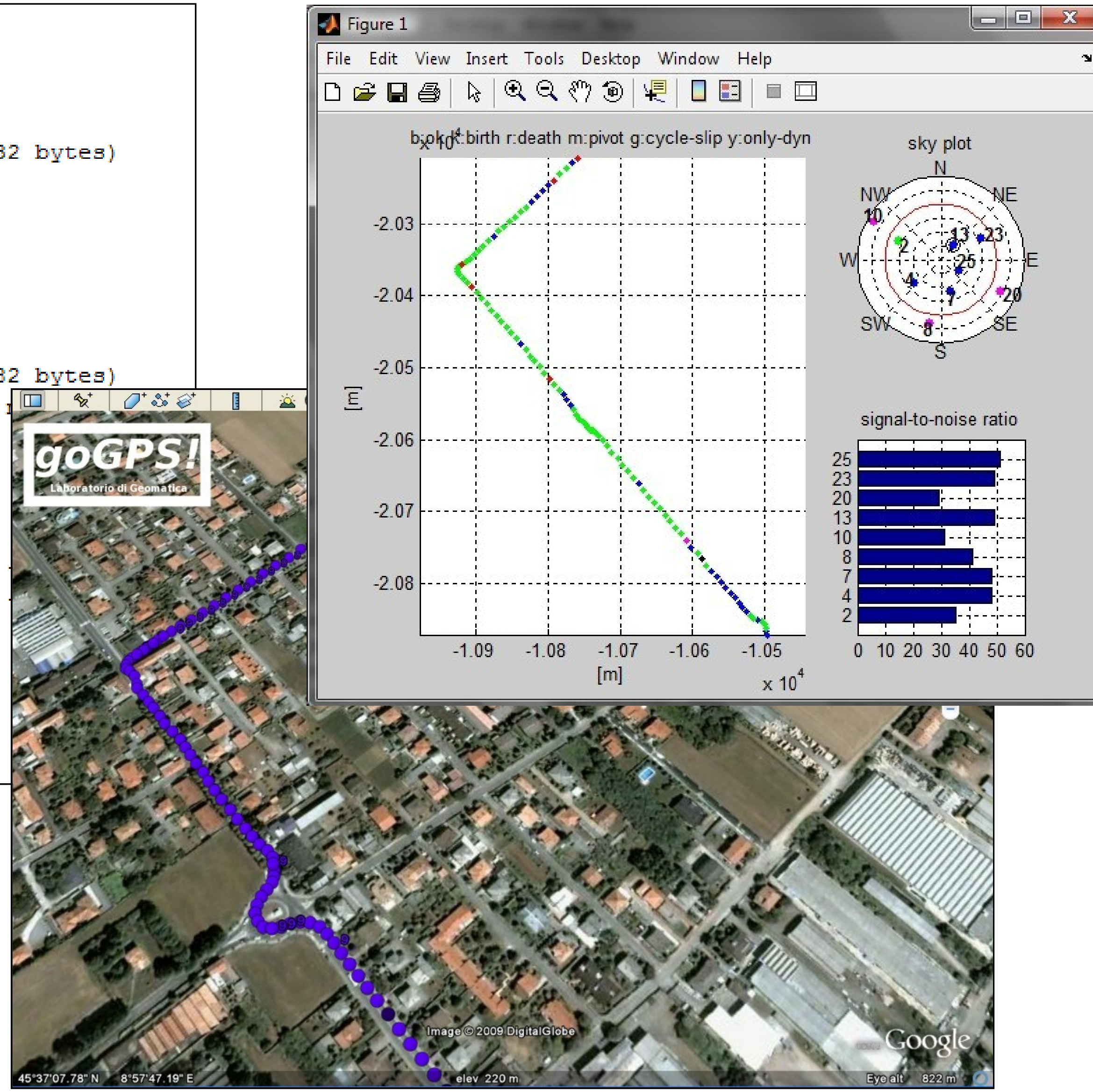
- Post-processing (post-mission) analysis by means of RINEX files or goGPS data saved during a real-time test



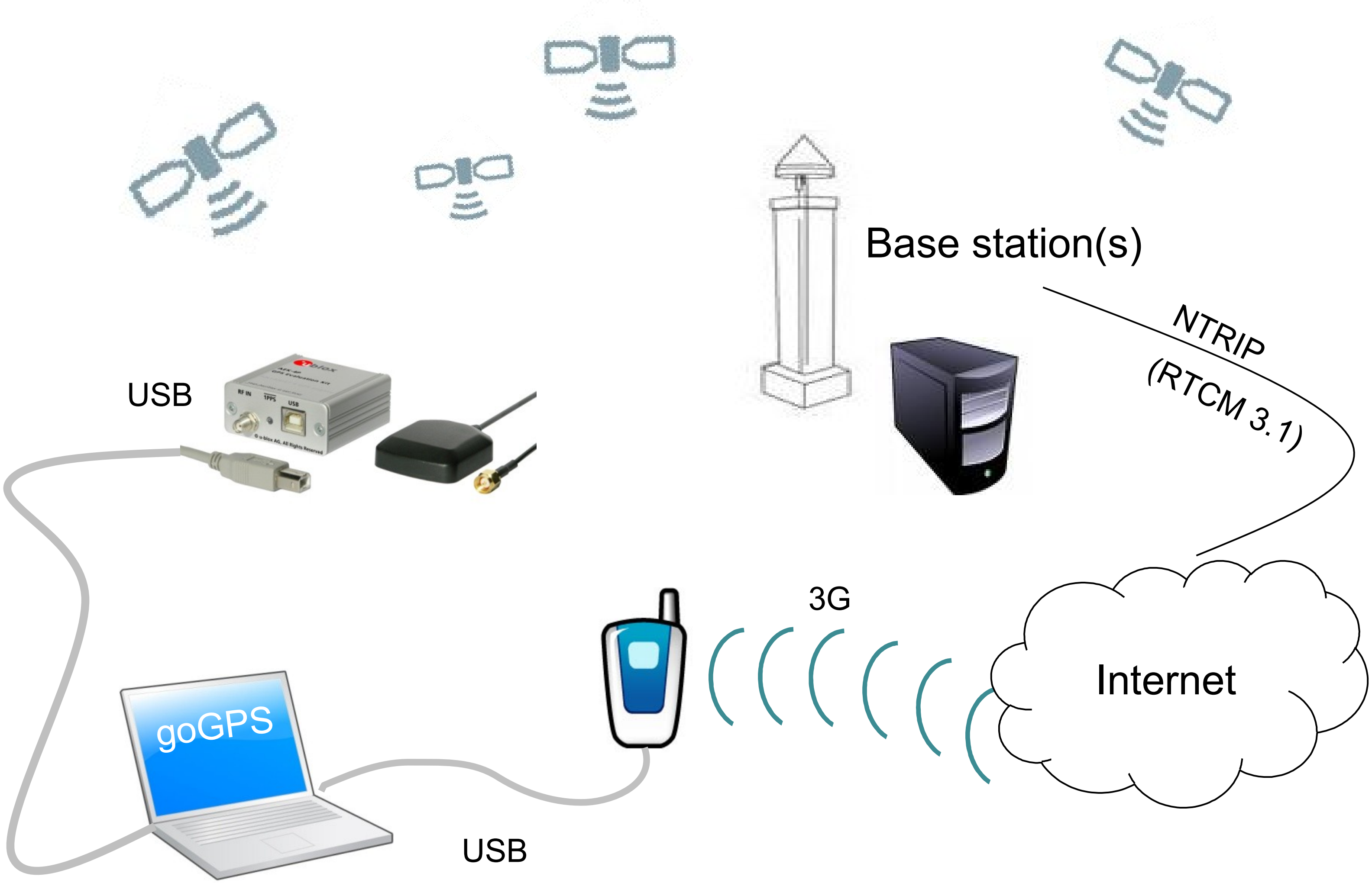
Software/2

OSAKA CITY UNIVERSITY

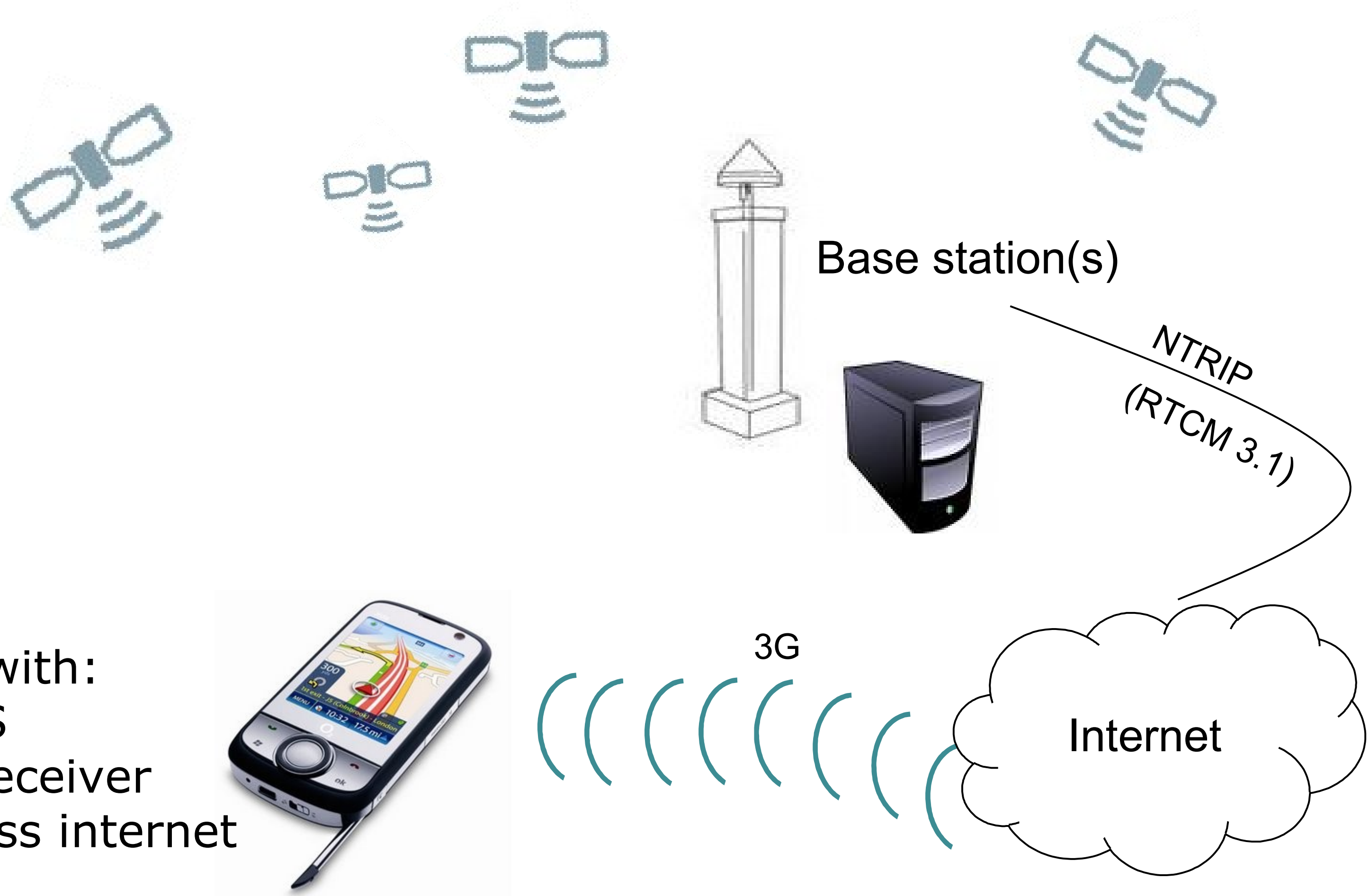
```
TIMING
epoch 1: GPStime=1517:322088

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): ooooooooooooooooooox
                                    ^
POSITIONING
no position/velocity are computed
EPH SAT: 13
OBS SAT: 13
```



Hardware/1



Hardware/2



device with:
- goGPS
- GPS receiver
- wireless internet

Accuracy test/1

<u>eBonTek egps597</u>
chipset: ANTARIS 4
signal: C/A, L1

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

<u>u-blox AEK–4T</u>
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.

Accuracy test/2

<u>Leica GPS1200</u>
signal: C/A, P
L1, L2

Double frequency receiver
with RTK capabilities.

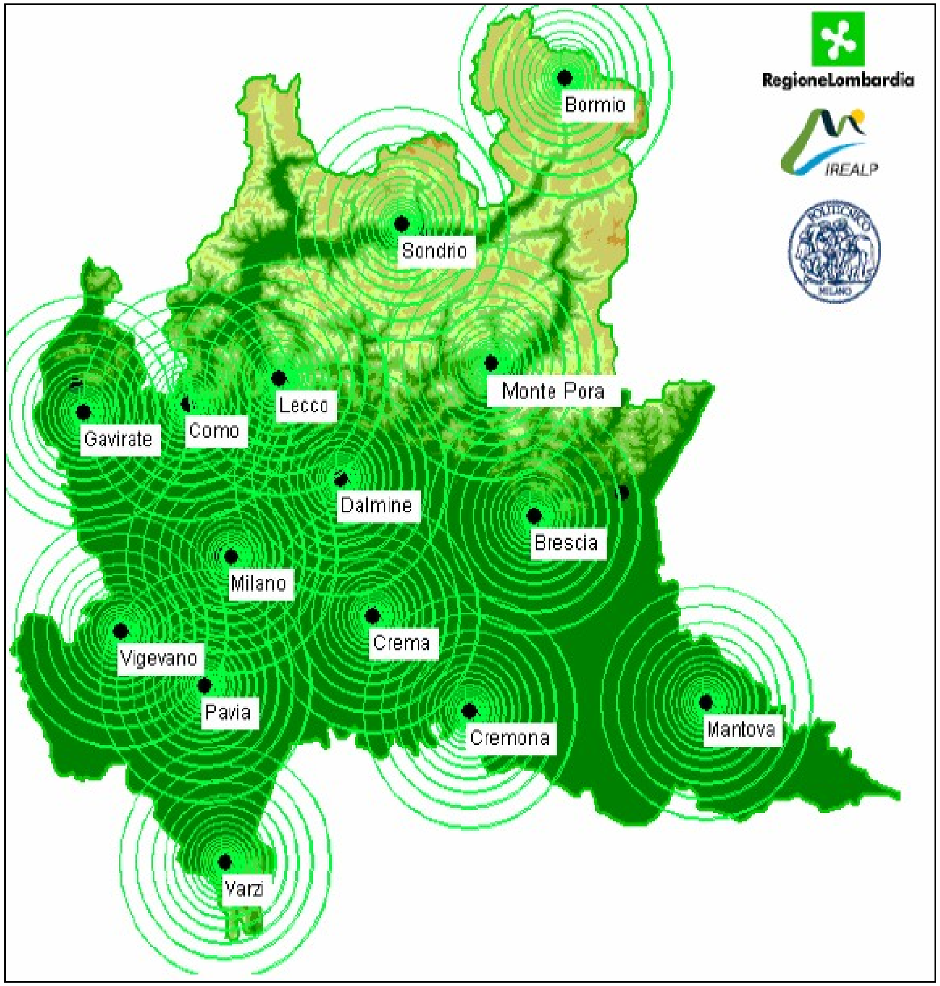
<u>Leica GS20</u>
signal: C/A, L1

Mid-level receiver (single freq.), designed
for cartographic update and quick
decimeter-level surveys. It supports
DGPS positioning.

Accuracy test/3

O S A K A C I T Y U N I V E R S I T Y

Como permanent station, used as base station (through GPSLombardia positioning service)



Accuracy test/4

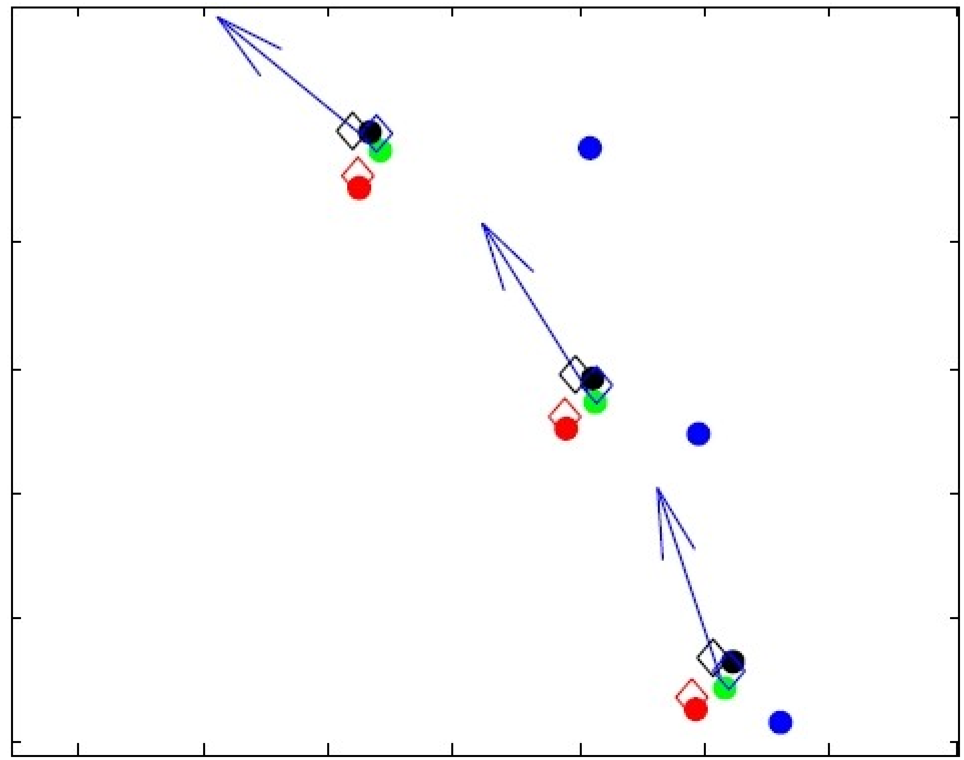
Devices:
- Leica GPS1200
- Leica GS20
- eBonTek eGPS 597
- ev. kit u-blox + goGPS

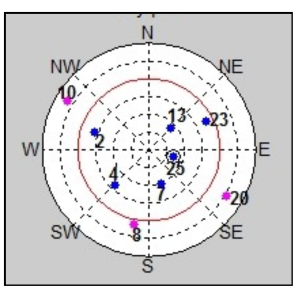
Fixed on the rooftop of a car driven on a road with good sky visibility.

Accuracy test/5

OSAKA CITY UNIVERSITY

Devices:
- Leica GPS1200 (green)
- Leica GS20 (red)
- eBonTek eGPS 597 (blue)
- ev. kit u-blox + goGPS (black)





goGPS
(cutoff = 10°)

mean	1.13 m
std	0.67 m

goGPS
(cutoff = 30°)

mean	0.78 m
std	0.47 m

Leica GS20
(mod. “Max Accuracy”)

mean	0.30 m
std	0.15 m

eBonTek

mean	4.03 m
std	1.70 m

goGPS & RTKLIB

goGPS conversion from MATLAB to C/C++

new (C++)	goGPS GUI	→	GUI
MATLAB → C	goGPS core	→	Kalman filter
already existing (C)	RTKLIB by T.Takasu (GPLv3)	→	- positioning func. - read/write - plotting - (...)

http://gpspp.sakura.ne.jp/rtklib/rtklib.htm

goGPS & WPS

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

DTM height

rough position

Rovers running goGPS

Server running goGPS with WPS functionality

accurate positioning

raw code and phase observations

Rovers just acquiring raw data

goGPS@Sourceforge

goGPS MATLAB code:

http://sourceforge.net/projects/gogps

Thank you!