POLITECNICO DI TORINO
Repository ISTITUZIONALE

URBAN DATA COLLECTION USING A BIKE MOBILE SYSTEM WITH A FOSS ARCHITECTURE

Original
URBAN DATA COLLECTION USING A BIKE MOBILE SYSTEM WITH A FOSS ARCHITECTURE / ABDUL JABBAR, A., Aicardi, I., Grasso, N., Piras, M.. - In: INTERNATIONAL ARCHIVES OF THE PHOTOGRAMMETRY, REMOTE SENSING AND SPATIAL INFORMATION SCIENCES. - ISSN 2194-9034. - ELETTRONICO. - XLII-4/W2:(2017), pp. 3-9. (FOSS4G-Europe 2017 Marne La Vallée, France 18–22 July 2017) [10.5194/isprs-archives-XLII-4-W2-3-2017].

Availability:
This version is available at: 11583/2676966 since: 2021-10-05T12:14:16Z

Publisher:
Copernicus Publisher

Published
DOI:10.5194/isprs-archives-XLII-4-W2-3-2017

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

27 November 2026

URBAN DATA COLLECTION USING A BIKE MOBILE SYSTEM WITH A FOSS ARCHITECTURE

A. Abdul Jabbar, I. Aicardi, N. Grasso, M. Piras*

Politecnico di Torino, DIATI - Department of Environmental, Land and Infrastructure Engineering, C.so Duca degli Abruzzi 24, 10129, Torino, Italy, (abdul.ansar, irene.aicardi, nives.grasso, marco.piras)@polito.it

Commission IV, WG IV/4

KEY WORDS: Low cost sensors, Urban area, 3D model, Raspberry system, Python, GNSS

ABSTRACT:
European community is working to improve the quality of the life in each European country, in particular to increase the quality air condition and safety in each city. The quality air is daily monitored, using several ground station, which do not consider the variation of the quality during the day, evaluating only the average level. In this case, it could be interesting to have a "smart" system to acquire distributed data in continuous, even involving the citizens. On the other hand, to improve the safety level in urban area along cycle lane, road and pedestrian path, exist a lot of algorithms for visibility and safety analysis; the crucial aspect is the 3D model considered as "input" in these algorithms, which always needs to be updated.
A bike has been instrumented with two digital camera as Raspberry PI-cam. Image acquisition has been realized with a dedicated python tool, which has been implemented in the Raspberry PI system. Images have been georeferenced using a u-blox 8T, connected to Raspberry system. GNSS data has been acquired using a specific tool developed in Python, which was based on RTKLIB library. Time synchronization has been obtained with GNSS receiver. Additionally, a portable laser scanner, an air quality system and a small Inertial platform have been installed and connected with the Raspberry system.
The system has been implemented and tested to acquire data (image and air quality parameter) in a district in Turin. Also a 3D model of the investigated site has been carried. In this contribute, the assembling of the system is described, in particular the dataset acquired and the results carried out will be described. different low cost sensors, in particular digital camera and laser scanner to collect easily geospatial data in urban area.

1. INTRODUCTION

1.1 State of the art

In the last decades, European community is working to improve the quality of the life in each European country, in particular, to increase the quality air condition and safety in each city. The quality air is daily monitored and analyzed, using several ground station devoted to acquire data about the air quality as PM10, PM5, ect.
Each European country is working to define some actions dedicated to limit or reduce the air pollution emission and to increase the quality of the life, first of all stimulating the mobility with green vehicle, like electric car or bicycle.
As demonstrated, mobility and life quality are strongly correlated and it is important the planning in the urban area, in order to facilitate the use of the green vehicle and to analyse the critical area on the pollution point of view.
In addition, the road safety is important, especially in the infrastructure dedicated to VRU (vulnerable Road User), as bicycle, wheel chairs, etc. In fact, it is important to guarantee an adequate level of safety, whit purpose to avoiding the road accidents (AASHTO, 2011).
In Figure 1, is described the comparison between road accidents and coal pollution in some European city. As we denote, in some case, the coal pollution is higher than the road accident, but the two phenomena are strongly correlated.

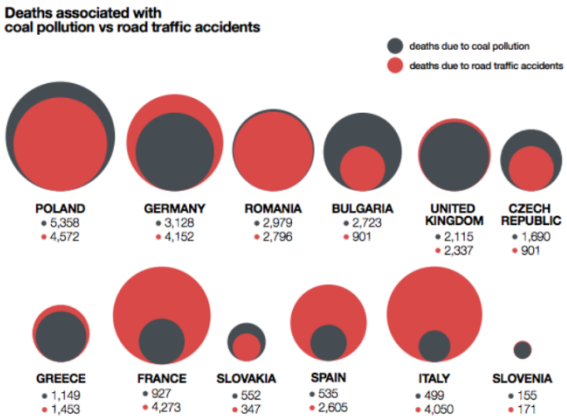

Figure 1. Coal pollution vs Road traffic accidents in EU [source Internet]

On the other hand, to improve the safety level in urban area along cycle lane, road and pedestrian path, exist a lot of algorithms for visibility and safety analysis (Bassani et al., 2015).
Considering this aspect, it is important to have some tools, which allow to realize a 3D city model, with a good geometrical accuracy, and to keep updated this model.
There are several commercial solutions like the mobile mapping system, but unfortunately they are very expensive and are not available to acquire data along bike path.
The use of mobile mapping system could be suggested to create a 3D model for vehicular infrastructure, but the local administrations (municipality) are not able to buy one. 3D city

* Corresponding author

model is usually generated by a specific survey, but it is very difficult to update it with low-cost solution.
An idea comes to Google Street view, where a bicycle has been used to acquire the data in several cities, where it has not been possible to use the Google car.



Figure 2. Coal pollution vs Road traffic accidents in EU [source Internet]

1.2 The case study

The area chosen as a test site is the 'Campidoglio' District in Turin, northwest of Italy (Figure 3). This area is predominantly residential, located in the semi-central location of the City, and characterized by the presence of commercial facilities, as well as public places and services, such as schools. The activities were aimed at the implementation of a medium-scale 3D metric documentation of a block of the 'Campidoglio' district by low-cost devices with the capabilities to associate pollution data. The generated 3D model can be used as reference for the spatial information system, which will be structured in an interoperable database.
This test site was selected for the presence of different kind of features like buildings, schools, roads, cycle and public transport paths. This area is also rounded by streets having different kind of traffic intensity, consequently, the air pollution can change significantly during the day according to the work and school time. So it can be very useful to monitor the pollution behavior at different times of the day.



Figure 3. The Campidoglio district (left) and the buildings block selected for the air quality analyses (right).

The activities described below refer to a fist acquisition made to test the system, the available sensors and the implemented structure for the storage and management of the data in a spatial database.

2. THE ACQUISITION METHODOLOGY

2.1 The bike mobile system

The bike mobile system can be an alternative to other mobile mapping systems (Ajmar et al., 2011;Bassani et al.,2012) due to its lower cost, smaller size and easier usage. This system can be used in areas where other vehicle based mapping system have difficulty of access. It is also designed as an automatic system which implies that all the sensors connected to this system start automatically without any further command. All the sensors connected to the bike can be seen in the Figure 4.

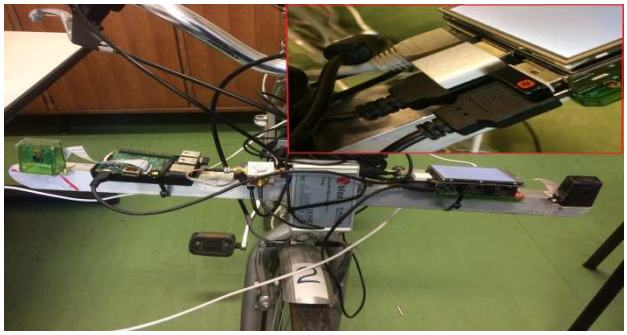

Figure 4. Bike mobile system with all sensors installed

2.2 Sensors

The Raspberry Pi (RPi) 3 system is used to control and process data collected from all the sensors. Five different sensors are connected to this bike mobile system.

- Raspberry Pi Camera RGB
- Raspberry Pi Camera IR
- Microstrain Inertial Measurement Unit (IMU)
- Low cost U-blox GNSS receiver
- HOKUYO Laser Scanner (LIDAR)

Two Raspberry Pi cameras are installed on the bike mapping system; these devices are both connected to the end of a 60 cm long rod, as shown in Figure 4. One camera collects images in RGB band, whereas, the other camera, acquires images in the infrared band. Both sensors have 8 MP resolution and they can be set up to 60 frames per second for low resolution images. For this project, they are set to 8 frames per second to capture high resolution images. It can also be recorded a video with 1080P (30 frames per second). An app in python is developed to control and capture images with both cameras and save them with a GPS timestamp automatically, so that these images can be integrated with other sensors data.

An inertial measurement unit 3DM-GX3-35 from Microstrain is also attached to the system to collect accelerometer, gyroscope and magnetometer data in the three dimensions x,y and z. There was no app available for Linux system to collect data from this kind of sensor so a new app has been developed in python language.

A GPS antenna is also connected to this IMU so that data from all inertial sensors can be synchronized with GPS time. GPS data packet and inertial sensors data packet can be collected with this python app. The specifications of all the IMU sensors are given in Table 1.

	Accelerometer	Gyroscope	Magnetometer
Measurement Range	±5 g	±300°/sec	±2.5 Gauss
Non-Linearity	±0.1 % fs	±0.03 % fs	±0.4 % fs
Initial Bias Error	±0.002 g	±0.25°/sec	±0.003 Gauss
Scale Factor Stability	±0.05 %	±0.05 %	±0.1 %
Noise Density	80 µg/√Hz	0.03°/sec/√Hz	100 µGauss/√Hz
Sampling Rate	30 kHz	30 kHz	7.5 kHz max

Table 1. Main specifications of IMU sensors

A low cost GNSS receiver, model u-blox M8T is also attached to this mobile mapping system to get position of the antenna attached to the bike and an open source software RTKLIB from Tomoji Takasu is used for standard and precise positioning with GNSS. It consists of a portable program library and several AP's in which some are graphical user interfaces (GUI) and some are command line interfaces (CUI). RTKRCV and STR2STR are the two AP's used in this system to get real time position and store raw data for post processing. RTKLIB supports all constellations like GPS, GLONASS, GALILEO, QZSS, BeiDou and SBAS with standard and precise positioning algorithms. Various positioning modes like single, DGPS/DGNSS, Kinematic, Static, Fixed, PPP-Kinematic, PPP-Static and moving baseline are supported for both real time and post processing modes. The correction from an external server can also be collected with this software by using TCP/IP, NTRIP, local log file and FTP/HTTP protocols.

Different configuration files are developed for different positioning modes and these files can be loaded in RTKRCV to get position in real time. The system is tested for static and kinematic modes to get the real time position with corrections from the nearest GNSS permanent station through NTRIP server.

A scanning laser range finder UTM-30LX from HOKUYO is also connected to this bike mobile system to estimate ranges. It is a small, accurate and high speed sensor which can be used for real time distance data collection. Due to its low power consumption, it can be used on any battery operated platform. The specification of this sensor are given below:

- Class 1 infrared laser scanner
- Range from 100mm to 30,000mm
- +/- 30mm accuracy
- 270° area scanning range with 0.25° angular resolution
- 25msec scanning time

3. DATA COLLECTION

The bike mapping system is tested around the area of Politecnico Di Torino with all the sensors working at the same time. Two Raspberry Pi 3 modules are connected on a rod in the front of the bike and both are synchronized with each other in time, which enables images and all the other data to be synchronized later. A power bank with the capacity of 15000mAh is also connected to the bike with two ports to power both the Raspberry Pi systems. The current consumption is around 800mA and hence the system can work for more than 15 hours with a fully charged power bank. The basic architecture of the bike mobile system can be seen in Figure 5.

Both Raspberry Pi cameras are set to 8 frames per second with 2592*1944 resolution and saves images with a GPS timestamp automatically. These Raspberry Pi cameras are connected to the Raspberry Pi system through a camera serial interface (CSI) and there is the possibility to connect more cameras to the system through USB ports.

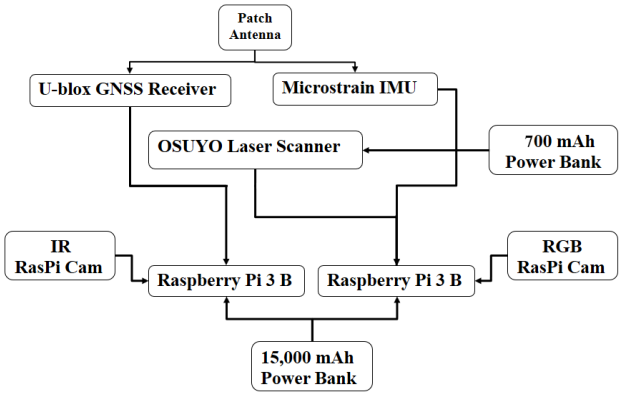

Figure 5. Basic architecture of bike mobile system

Then Inertial measurement unit (IMU) is connected through a USB interface and it is set to collect only inertial sensors packet with GPS timestamp and, also, internal timestamp of IMU. The frequency of data collection can be set from 1Hz to 1000 Hz but, during the tests, it was set to 2Hz. The system starts recording data automatically and save the data in csv format, which can be read through Microsoft, excel. Since the inertial sensor data is highly accurate for shorter time periods, it can be integrated with GNSS data collected during post processing to improve the positioning. Figure shows the data packet collected from inertial sensors with the bike mapping system.

```
Packet Received:  62
GPS Correlation Timestamp-Time of Week :  124031.507760
GPS Correlation Timestamp-Week Number :  1941
GPS Correlation Timestamp-Flags :  5
Internal Timestamp-Tick :  96126793
Raw Accelerometer Vector-X :  66425.734375
Raw Accelerometer Vector-Y :  73086.398438
Raw Accelerometer Vector-Z :  63195.199219
Raw Gyro Vector-X :  64699.535156
Raw Gyro Vector-Y :  66346.000000
Raw Gyro Vector-Z :  65314.800781
Raw Magnetometer Vector-X :  -3354.235352
Raw Magnetometer Vector-Y :  -1020.705872
Raw Magnetometer Vector-Z :  4910.176270
Euler Angles-Roll :  -0.123545
Euler Angles-Pitch :  0.287501
Euler Angles-Yaw :  -2.633861
Scaled Accelerometer Vector-X :  0.283568
Scaled Accelerometer Vector-Y :  0.115364
Scaled Accelerometer Vector-Z :  -0.956997
Scaled Gyro Vector-X :  -0.004931
Scaled Gyro Vector-Y :  -0.006507
Scaled Gyro Vector-Z :  -0.001298
Scaled Magnetometer Vector-X :  -0.151185
Scaled Magnetometer Vector-Y :  0.016041
Scaled Magnetometer Vector-Z :  0.236851
Orientation Matrix-Matrix :  [[-0.837983,-0.466241,-0.283556],[0.513025,-0.8502,
-0.118173],[-0.185983,-0.244498,0.951647]]
```

Figure 6. Data Packet received from IMU

The low cost GNSS receiver is also connected to the Raspberry Pi 3 through USB serial interface. A patch antenna is connected to the receiver, which collects signals from all available constellations. RTKRCV and STR2STR are the two AP's installed on the RPi system to process raw data collected from u-blox receiver and then get position in real time. STR2STR takes raw data from the receiver and divides it into two streams. One stream is saved to a file in ubx format to be processed later after applying corrections downloaded from nearest permanent base station. The other data stream is given to RTKRCV, which

performs real time positioning after applying corrections taken from SPIN GNSS network through NTRIP protocol. The system is tested with static mode for two baselines i-e zero baseline and 100 meters baseline with known coordinates. The accuracy of real time position was up to mm level on North and east coordinates whereas up to cm level for up coordinate as shown in table.

	Zero Baseline	100 meters Baseline
MSDN	6.8 mm	6.8 mm
MSDE	4.8 mm	5.6 mm
MSDU	1.13 cm	1.14 cm

Table 2. Mean standard deviations for Two Baselines

The real time data is plotted using RTKPLOT, which is another AP of RTKLIB, and the solution could either be single, float or fix as seen in figure. RTKLIB calculates the exact number of radio wavelengths between the available GNSS satellites and the base station antenna using a complicated algorithm for ambiguity resolution and it yields either float or fixed solution. Poor satellite constellation, low number of satellites and a poor radio link between rover and base station may prevent a fixed solution.

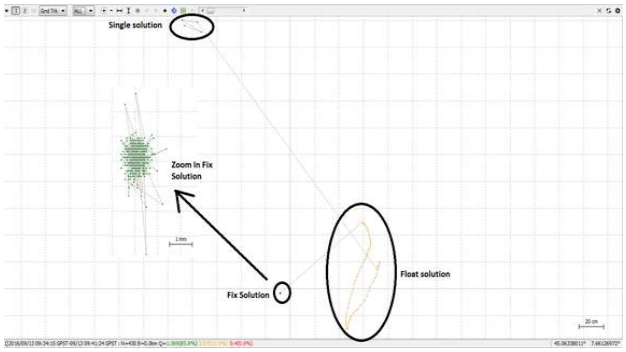

Figure 7. Real time position with fix, float and single solution

The range laser scanner is connected to the RPi system through an Ethernet cable interface but it can also be connected with USB 2.0 interface. One scan returns 1080 distance values between the angles -135 to +135 degrees and it can perform a total of 40 scans per second. The angle 0 corresponds to the Y-axis, the scanned data values can be converted to x and y coordinates. Along with the distance values, it is also possible to get intensity values to categorize different objects based on the different strength of reflection from each object. There is also a possibility to have more than one reflection and a multi-epoch option is also available to collect both reflection and intensity values as shown in Figure 8. An app has been developed in C language to collect all distance, multi-epoch and intensity data values from LIDAR and save it in ubh format, which can be read in UrgBenry software and processed later to integrate it with GNSS and IMU data. In Figure 8, the data collected from the Laser Scanner is displayed in a plot from the UrgBenri software, where the intensity values are in pink and the distance values are in green.

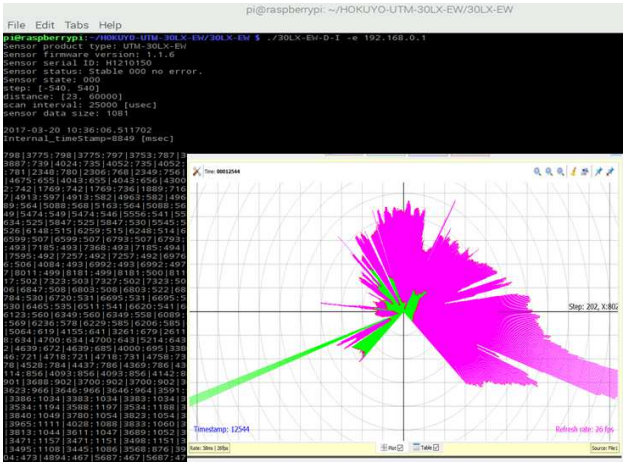

Figure 8. Data collection from Laser range Scanner and then plotting it in UrgBenri software

The next step after the collection of data from all the sensors is to geotag all collected digital images with the geographic coordinates from the GNSS receiver by comparing the timestamp.

Geotagging is the process of adding geographical identification data to digital images. An open source software EXIFTOOL is installed on the RPi system to geotag all collected images from the RGB and IR cameras. The software supports many formats and in this work, positioning data in the nmea format is compared to the time stamp of the images to geotag each image. Then all these geotagged images are exported in google earth to see the path of the data collection, the images can also be seen as shown in Figure 9.



Figure 9. Data Collection Path along with geotagged photos in google earth

4. DATA PROCESSING

4.1 Management of Dynamic Data in GIS

The data obtained from the used sensors were structured on a dynamic spatial database for the management of air quality data. It is based on a data model compliant with the OGC CityGML standard (Prandi et al., 2013, Soave et al., 2013), with a proposed extension for what concerns the dynamic environmental parameters: the CityGML Air Quality Application Domain Extension (AQADE) (Arco et al., 2016). Analyzing the monitored data in association with their location

in the 3D city model and further parameters (such as traffic at the moment of the survey, building usage, heating periods and so on) it could be possible to do some considerations (e.g. what are the main sources of air quality pollution and what are the main affected objects).

These data can be managed translating the model in a SQL database. The passage to an object-relational model permits to manage in an easier way the data, maintaining some important relations. To these purposes PostgreSQL-PostGIS was used. The GIS software tool, QGIS, was used as a support for both a graphical interface and analysis tools.

The data can be managed, visualized and analyzed in GIS for obtaining useful information and perform many analysis (eg. statistics, spatial queries). Moreover, a map of the pollution of the city can be generated by the interpolation of the measured points (Arco et al., 2016) (Figure 10).

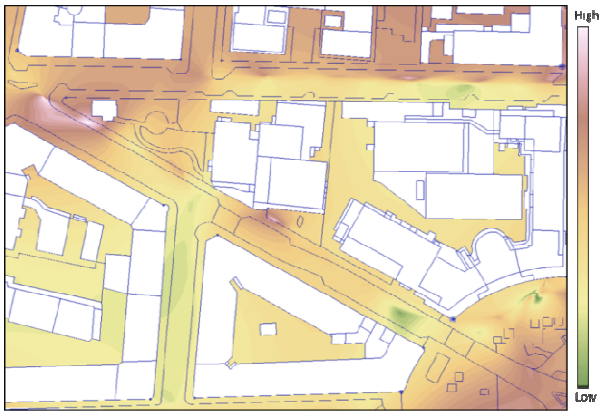

Figure 10. Generation of the map of the presence of particulate (PM 2.5) in the study area, deriving from the interpolation of the measured values conducted on the GIS platform.

4.2 Imaging Sensors Calibration

Before using the system, it is necessary to calibrate it. This means to know the interior parameters (focal length, principal point and lens distortions), which characterizes the cameras involved. To get this data, a free calibration tool, Calib 3V, developed by the IUAV (Istituto Universitario di Architettuta di Venezia) (Balletti et al., 2014) was used. This software allows to obtain the intrinsic parameters from images taken from the camera. For the procedure, a regular grid of circles (5 rows × 7 columns) with known dimensions need to be used (Figure 11)
The procedure then suggests carrying out a series of photos in such a way that the calibration page will cover almost the total width and height of the sensor from different locations above of the planar surface of the calibration field.

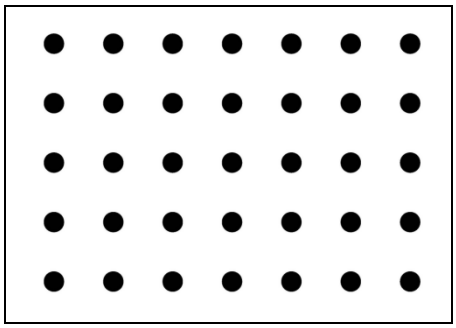

Figure 11. Grid of circles printed and used for the camera calibration.

These pictures can, then, be imported in Calib 3V. 10 photos were acquired for each camera and the application allowed to extract the parameters (Table 3):

Cam. Param.	Camera 1	Camera 2
f [mm]	1.457	1.463
X_o [mm]	0.68	0.593
Y_o [mm]	0.475	0.533
k_1	-0.07	0.024
k_2	2.297	3.187
k_3	-13.114	-8.779
p_1	-0.004	-0.004
p_2	-0.003	-0.002

Table 3. Estimated values of intrinsic parameters.

4.3 Photogrammetric data processing

In the present work, some tests are presented, performed with the purpose to analyse the possibility of acquire images with Raspberry Camera Module and use them for 3D modelling, to obtain a more detailed description of the area. Images have been georeferenced using a ublox 8T which was connected to the Raspberry system.
The alignment of images sequences, 813 from each camera (1626 in total), was realized. The acquired data were processed using the Structure from Motion (SfM) approach implemented in different software tools, VisualSFM and Agisoft Photoscan Professional. The process is carried out almost automatically, and in a fast and easy way, by these software tools, based on algorithms of photogrammetry and computer vision. The input data required by these tools to perform the 3D dense point cloud reconstruction process are only the acquired images, since it is not even necessary to know a-priori the exterior orientation parameters of the cameras. To georeference the model is possible to add the coordinates of the images or, if known, that of some control points in the environment.
A first test was performed using the open source software VisualSfM (Wu, 2011) with the integrated algorithm CMVS/PMVS (Furukawa, 2010). With this tool the images were easily aligned and a first point cloud was generated, nevertheless, the software doesn't allow to import the coordinates of the images, so the model cannot be georeferenced.

On the other and, Agisoft Photoscan Professional, which is a commercial tool, produced by the Russian-based company Agisoft LLC, is able to directly read the exif file from the images and their coordinates. The result of processing is therefore a georeferenced cloud (Figure 12).



Figure 12. The aligned images (in blue) and the dense point cloud generated by the software Agisoft Photoscan.

This data could be further processed to obtain a 3D mesh, which could be used to update or replace existing 3D model of the environment.
According to the acquired data, the obtained point clouds offer an almost complete 3D model of the urban environment, excluding the roofs of the buildings.

4.4 Results and discussion

The proposed system allows to survey the 3D geometry of the city using mobile and low-cost sensors.
To georeference the model, the coordinates of the point of acquisition of the images (Figure 13), measured with the U-blox 8T, were used.



Figure 13. Camera positions measured by the GNSS device.

Agisoft Photoscan Professional uses these informations as input data and, after the images alignment, it recalculates the corrected camera positions (Figure 14).



Figure 14. Camera positions estimated by Agisoft Photoscan Professional.

Some first analysis has been performed to evaluate the accuracy of the model. Not having at disposal known coordinates of control points in order to asses the accuracy of the product, the estimated positions of the cameras were compared with the measured coordinates (GNSS track) (Figure 15). The results are also summarized in Table 4.

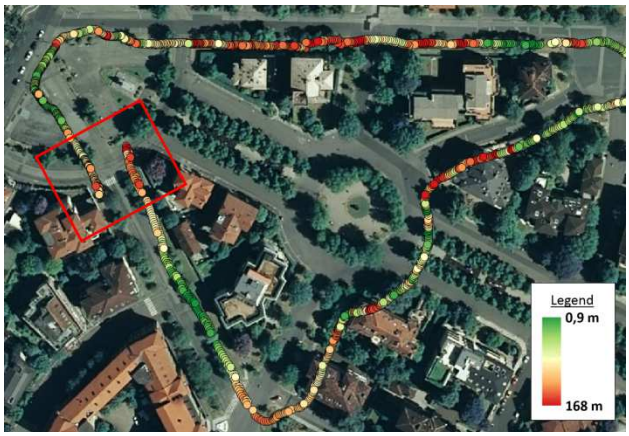

Figure 15. Difference between image coordinates measured by the U-blox 8T and the camera positions estimated by the software Agisoft Photoscan.

	Min [m]	Max [m]	Average [m]
Difference between coordinates	0.93	167.5	38,65

Table 4. Results of the difference between image coordinates measured by the U-blox 8T and the camera positions estimated by the software Agisoft Photoscan.

Observing the differences between the coordinates some problems could be ascribed to the characteristics of the surveyed environment. The tall buildings on the sides of the road and the dense vegetation have affected the GNSS coverage causing a large number of outliers. Therefore, the system for the position data acquisition could be improved by using alternative instruments such as inertial measurement units (UMBs) or ultra-wide band (UWB). Also, a certain number of control points can be measured, in order to better verify the accuracy of the result.

According to the achieved results, the found discrepancies, may be also generated by an incorrect alignment of the images. Looking at the Figure 15 it is evident that, in the area highlighted by the red box, the two points of the track do not match between them. These two points represent the position of two subsequent images.
As general remark, the image alignment could be also influenced by the presence of certain kind of features which characterize the environment, as the vegetation, but also by moving object. During the acquisition phase, a lot of cars and people were moving along the street. Also in this case, the dense vegetation influenced the product; probably, performing the data acquisition in a season in which the trees have no leafs could improve the result.
Finally, for the future tests, further analysis can be conducted on these particular kind of imaging sensors, which, as known, are quite instable, to optimize the estimation of their intrinsic parameters performed by the software and improve the image alignment.

5. CONCLUSIONS AND FUTURE WORKS

The developed and tested system shown several interesting aspect, in particular the performance of the sensors (IMU, GNSS, digital camera) are very impressive.

Is, therefore, possible to affirm that the proposed system allows to survey the 3D geometry of the city using mobile and low-cost sensors. The system, composed by the GNSS receiver connected to the Raspberry, which allows to georeference the images, still needs to be improved to obtain a 3D model with a metrical level of accuracy, which would make it suitable to perform analysis on air quality.
Generally, under open sky condition, the accuracy of the 3D model is locally correct and it can be used to make the visibility analysis in some limited area and to updating the map.
Moreover, the acquired information can be effectively managed in standardized GIS, useful for supporting a number of queries and analysis for different aims, as that focused on the analysis of the air quality.

Using FOSS platform, it is possible to developing a smart system, which is able to make the data acquisition from different sources, especially to synchronize the time and to georeferencing the data using the GNSS positioning, even applying a RTK solution.
Air quality model describes only the behaviour of the pollution in the test site, without a realistic correspondence to the real pollution value. This model is also important, because it allows to describe locally and relatively the different air quality zone and to know the correlation between traffic and pollution, even considering the geometry of the urban landscape.

The quality of RaspPi cam is quite good and it can improve applying a correct calibration model. The integration is very simple and it is easy to collect the images, but time synchronization requires to developing a specific tools.

Possible future activities would be the investigation about the GNSS performance, changing the antenna or working with a real multi-constellation system composed by GPS, GLONASS and BEIDOU. Moreover, it will be also interesting to analyse the performance making a combined data processing IMU-GNSS, with purpose to have a continuous tracking and solution, even on urban canyon or tree-lined area, both in real time and post-processing.

REFERENCES

Ajmar A., Balbo S., Boccardo P., Giulio Tonolo F., Piras M., Princic, 2011. *A Low-Cost Mobile Mapping System (LCMMS) for Field Data Acquisition: a Potential Use to Validate Aerial/Satellite Building Damage Assessment. International Journal of Digital Earth*, 1(21), pp. 1-21

American Association of State Highway and Transportation Officials, AASHTO, 2011. *A Policy on Geometric Design of Highways and Streets*. 4th edition, ISBN: 1-56051-156-7.

Arco, E.; Boccardo, P.; Gandino, F.; Lingua, A.; Noardo, F.; Rebaudengo, M., 2016. An integrated approach for pollution monitoring: smart acquirement and smart information. In: *1st International Conference on Smart Data and Smart Cities*, 30th UDMS, Split (Croatia), 7–9 September 2016. pp. 67-74.

Balletti, C., Guerra, F., Tsioukas, V., & Vernier, P. (2014). Calibration of action cameras for photogrammetric purposes. *Sensors*, 14(9), 17471-17490.

Bassani, M., Lingua, A., Piras, M., De Agostino, M., Marinelli, G. and G. Petrini, 2012. Alignment Data Collection of Highways using Mobile Mapping and Image Analysis Techniques. Transportation Research Board of the National Academies, 91st Annual Meeting, No. 12-0312, Washington D.C.

Bassani, M., Grasso, N., & Piras, M. (2015). 3D GIS based evaluation of the available sight distance to assess safety of urban roads. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *40*(3), 137.

Furukawa, Y., Curless, B., Seitz, S. M., Szeliski, R., 2010. Towards internet-scale multi-view stereo. In: *Computer Vision and Pattern Recognition (CVPR)*, pp. 1434-1441.

Prandi, F., De Amicis, R., Piffer, S., Soavea, M., Cadzowb, S., Boix, E. G., D'Hondt, E., 2013. Using CityGML to deploy smart-city services for urban ecosystems. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4.

Soave, M., Devigili, F., Prandi, F., de Amicis, R., 2013. Visualization and analysis of CityGML dataset within a client sever infrastructure. In: *Proceedings of the 18th International Conference on 3D Web Technology,* pp. 215-215.

Wu, C., 2011. *VisualSFM: A visual structure from motion system.* http://ccwu.me/vsfm/doc.html