

COMPARISON BETWEEN RGB AND RGB-D CAMERAS FOR SUPPORTING LOW-COST GNSS URBAN NAVIGATION

L. Rossi^{1,*}, C.I. De Gaetani¹, D. Pagliari¹, E. Realini², M. Reguzzoni¹, L. Pinto¹

¹ Politecnico di Milano, DICA - Geodesy and Geomatics section, Piazza Leonardo da Vinci 32, 20133 Milan, Italy
(lorenzo1.rossi, carloiaipige.degaetani, diana.pagliari, mirko.reguzzoni, livio.pinto)@polimi.it

² Geomatics Research & Development s.r.l., via Cavour 2, 22074 Lomazzo (CO), Italy
eugenio.realini@gred.eu

Commission II, WG II/1

KEY WORDS: Urban Navigation, Relative Orientation, Kalman filter, RGB-D, Kinect, GNSS, Low-cost

ABSTRACT:

A pure GNSS navigation is often unreliable in urban areas because of the presence of obstructions, thus preventing a correct reception of the satellite signal. The bridging between GNSS outages, as well as the vehicle attitude reconstruction, can be recovered by using complementary information, such as visual data acquired by RGB-D or RGB cameras. In this work, the possibility of integrating low-cost GNSS and visual data by means of an extended Kalman filter has been investigated. The focus is on the comparison between the use of RGB-D or RGB cameras. In particular, a Microsoft Kinect device (second generation) and a mirrorless Canon EOS M RGB camera have been compared. The former is an interesting RGB-D camera because of its low-cost, easiness of use and raw data accessibility. The latter has been selected for the high-quality of the acquired images and for the possibility of mounting fixed focal length lenses with a lower weight and cost with respect to a reflex camera. The designed extended Kalman filter takes as input the GNSS-only trajectory and the relative orientation between subsequent pairs of images. Depending on the visual data acquisition system, the filter is different because RGB-D cameras acquire both RGB and depth data, allowing to solve the scale problem, which is instead typical of image-only solutions. The two systems and filtering approaches were assessed by ad-hoc experimental tests, showing that the use of a Kinect device for supporting a u-blox low-cost receiver led to a trajectory with a decimeter accuracy, that is 15% better than the one obtained when using the Canon EOS M camera.

1. INTRODUCTION

The task of precise navigation in urban areas is quite challenging, mainly because the complexity of urban environments prevents the use of a single sensor for recovering trajectories. In the last decades, the use of Global Navigation Satellite System (GNSS) has been widespread, mainly thanks to its easiness of use. However, a pure GNSS navigation is often unreliable in urban areas because of the presence of obstructions (e.g. tall buildings, narrow streets, trees, tunnels) that could prevent a correct reception of the satellite signal. To overcome the problem of sky visibility, solving at the same time the navigation and the mapping tasks, the so-called Mobile Mapping Systems have been employed for decades. These systems are usually very expensive, especially when high accuracies are required. Nowadays, a plenty of possible low-cost solutions are available, e.g. supporting GNSS receiver with Micro Electro-Mechanical Systems (MEMS) (Noureddin et al., 2009), reduced inertial sensor systems (Georgy et al., 2010) or radio-frequency wireless technologies (Nur et al., 2013).

Interesting low-cost solutions are also represented by image-based techniques, useful to overcome GNSS leakages in urban areas. The use of photogrammetric techniques for navigation purposes has been widely discussed by several authors (see e.g. Chaplin, 1999; Da Silva et al., 2003; Pagliari et al., 2018). Also Computer Vision (CV) techniques are quite common to solve navigation tasks concerning autonomous robots and Unmanned Aerial Systems (UAS) in GNSS denied environments. In this context, the

problem of recovering simultaneously the relative camera position and the 3D model from a set of images (calibrated or uncalibrated) is known as Simultaneous Location And Mapping (SLAM). SLAM solutions are usually computed integrating data acquired by different sensors. On the other hand, when the solution to the problem is computed using only input images from a single or multiple cameras, the term Visual Odometry (VO) is used (Scaramuzza and Fraundorfer, 2011). VO approaches can be classified in stereo VO and monocular VO. In the latter case, the solution is recovered up to a scale factor, which has to be externally measured (e.g. using other sensors) or estimated. A common approach to overcome this problem is the integration of RGB images with data acquired by depth cameras. Time-of-flight (ToF) cameras deliver images in which every pixel contains range information, meaning that a single image is sufficient for creating a correctly scaled 3D model. Usually, these systems integrate also RGB images and are known as RGB-D sensors. Several RGB-D sensors have been launched on the market in the last years and thanks to their high frame rate, low weight and reduced power consumption they have been employed in several research areas, such as robotics and machine vision (see e.g. Prusak et al., 2008; May et al., 2006). In this sense, the Microsoft Kinect device and analogous instruments, such as PrimeSense Carmine and Asus Xtion ProLive, are very attractive since they integrate both active and passive sensors. In literature, a high number of navigation solution based on the use of RGB-D cameras can be found. See, among others, Fankhauser et al. (2015), Oliver et al. (2012) and Henry et al. (2012).

It is important to point out that the estimation of the camera 3D motion is performed sequentially in SLAM. This frame-to-frame

*Corresponding author

approach inevitably leads to accumulate drift errors over time. Therefore, coupling GNSS and cameras for outdoor navigation applications is mutually beneficial for these sensors, the GNSS overcoming possible signal outages and the lack of attitude information, the cameras controlling trajectory drift errors.

This paper exploits the possibility of integrating low-cost GNSS observations with different kinds of imaging data by means of an extended Kalman filter, focusing on the comparison between the use of RGB-D and RGB cameras. In particular, we compared the solution obtained by integrating a Microsoft Kinect for XboxOne (second generation) or a Canon EOS M, considering these two sensors as representatives of their categories. From one hand, the Kinect is a low-cost RGB-D camera, meaning that it delivers an already scaled solution. On the other hand, the Canon EOS M acquires high quality images, coupled with the possibility of using fixed focal length lenses, but guaranteeing lower weight and cost with respect to a reflex camera. Nevertheless, the length of the baseline between two subsequent acquisitions, i.e. the scale factor, needs to be estimated separately. This implies to design and to implement two different versions of the extended Kalman filter for the data integration. Furthermore, this filter represents an improvement with respect to that presented in Pagliari et al. (2016) because it takes into account also the estimation of the vehicle attitude. In fact, the GNSS antenna and the image acquisition systems (RGB and RGB-D cameras) are not necessarily located in the centre of the vehicle, thus making crucial the knowledge of the attitude to correctly compute the vehicle trajectory. After implementing the filters, the comparison between RGB-D and RGB cameras in supporting low-cost GNSS receivers is performed on an experimental level.

The paper is organized as follows. In Section 2 the experimental test configuration and the system calibration are discussed. In Sections 3 and 4 a brief introduction to the Kinect device and the Canon EOS M camera, including their characteristics and calibration is presented, respectively. In Section 5 the low-cost GNSS receiver used and the goGPS software are introduced. Section 6 focuses on the proposed solution, with a detailed description of the implemented Kalman filter, while in Section 7 the results of the experimental test are presented and discussed. Conclusions and recommendations are drawn in Section 8.

2. EXPERIMENTAL TEST CONFIGURATION AND CALIBRATION

The test vehicle was realized by equipping a wooden cart with the instruments shown in Figure 1, namely a Kinect (K), a Canon EOS M (C) mounting an EF-M 22 mm f/2 STM lens and a u-blox EVK-6T receiver connected to a Tallysman TW-3070 antenna (G). Moreover, a 360° reflective prism (P1) and three double-frequency Leica GNSS receivers (G1, G2, G3) were installed on the cart and used to reconstruct a reference trajectory for the result validation. The prism was tracked by means of a Leica Nova MS-60 Multistation (MS).

During the survey, the RGB and depth images of the Kinect were acquired at about 0.3 s sampling rate, the RGB images of the Canon EOS M at about 0.5 s, and the GNSS data of the u-blox receiver at about 1 s. As for their synchronization, the GPS time was selected as the reference one. Since the Kinect and the u-blox were controlled by the same PC, their observations could be synchronized through the PC internal time, which is then converted into GPS time. The Canon EOS M synchronization with the other sensors was possible thanks to the Canon GP-E2 device that synchronizes the internal camera time with the GPS time.

A remark is that the three instruments were mounted in differ-

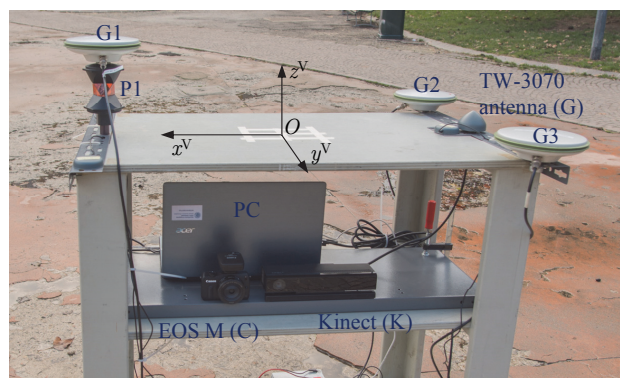


Figure 1. Cart setup. The instruments mounted on-board and the materialized vehicle reference frame V are highlighted. The photo was taken during the calibration phase.

ent locations on the cart, namely their observations are referred to different points, i.e. C , G and K . This translates into the fact that each instrument follows a different trajectory when the cart is moving. Therefore, as discussed in Section 6, the position and the attitude of each instrument with respect to a vehicle reference frame V are necessary. The vehicle reference frame has to be materialized, thus we assumed its origin at the barycentre O of the top plane of the cart, with the x^V and y^V axes parallel to its two main sides and the z^V axis orthogonal to the plane itself (see Figure 1). Furthermore, to initialize the extended Kalman filter, the starting position of the barycentre O and the attitude with respect to the reference frame in which the cart trajectory were computed, e.g. a local East-North-Up (ENU) reference frame. Actually, the cart trajectory was estimated in a 2D local East-North (EN) reference frame, assuming that the vehicle z^V axis remained parallel to the Up direction and that we were in presence of a flat field.

All the quantities above introduced were estimated by means of photogrammetric techniques, performing an on-field calibration. In particular, by parking the cart at its initial state, we estimated the position and the attitude of the Kinect and Canon EOS M centres of projection by introducing their images (taken at the initial position) into a bundle block adjustment together with a set of images acquired by a Nikon D800 reflex camera with a fixed focal length equal to 20 mm looking at the cart and the surrounding environment. The photogrammetric block relies on 12 Ground Control Points (GCPs) acquired by using the MS and referred to the ENU reference frame, since the station points were surveyed by double-frequency GNSS receivers too. The output of the bundle block adjustment were the position and attitude of Kinect and Canon EOS M with respect to the ENU reference frame and the coordinates of some corners of the cart. From these coordinates we determined the position of the barycentre O and the attitude of the cart in the ENU reference frame. During this phase, we measured also the position of the prism P1 and the four GNSS antennas (G, G1, G2, G3) in the ENU frame. The former is observed by means of the MS, while the latter are estimated by 15 minutes static sessions double-differenced with respect to a near permanent station (MILA station from the SPIN GNSS network). Combining the position and attitude of the instruments in the ENU frame and the origin and attitude of the vehicle reference frame V , we determine also the position and attitude of each instrument with respect to the latter.

Finally, the reference trajectory of the cart barycentre O and the reference attitude were retrieved by combining the trajectory observed by G1, G2, G3 and P1. This is possible since during the calibration phase also the relative position of these instruments

with respect to the cart reference system V were estimated.

3. MICROSOFT KINECT RGB-D DATA PROCESSING

The Microsoft Kinect device was firstly released on the market in 2010, and since then it has appeared as a great innovation, not only for gaming and entertainment, but also for research applications. It is composed by an RGB camera, an IR camera (co-registered with a depth camera) and an IR projector. It is capable of acquiring coloured and range images with a frame rate up to 30 fps. Since it allows to access at a low-cost (less than \$200) data with a complementary nature, it has been widely studied and used for both autonomous navigation and 3D modelling purposes. The second generation (v2) of the device, released in 2014, represents a huge improvement over the previous one because of a higher resolution (1280×1090 pixels for the RGB camera and 512×424 pixels for the IR camera) and the possibility of acquiring depth information even outdoor (with an operative range between 0.5 m and 4 m), under the influence of sunlight radiation, thanks to a different range measurement principle, i.e. Time of Flight (ToF) instead of triangulation based on structured light speckle pattern techniques.

The Kinect IR camera delivers different output images: (1) gray scale images dependent on the ambient lighting, (2) active gray scale images independent from the ambient lighting and (3) depth images, namely images where the value of the distance between the object and the sensor is stored into each pixel.

To use the Kinect for metric applications it is important to geometrically calibrate all the imaging sensors. First of all, the intrinsic camera parameters (together with the radial and tangential distortion) were estimated following the standard calibration procedure of the Matlab embedded Camera Calibrator app, by using the provided black and white checkerboard. The same image dataset were also used for the estimation of the relative orientation parameters between the IR and the RGB cameras, whose knowledge is fundamental for integrating the data delivered by the Kinect. Last, but not least, the systematic errors that affect the depth measurements (for a review of ToF cameras error source see e.g. Foix et al., 2011) were corrected and removed following the procedure for depth calibration by Pagliari and Pinto (2015). In our experiment the Kinect was used for acquiring both RGB and depth images. The high frame rate of the device produces a large amount of data, therefore the images were down-sampled at about 1.5 s sampling rate. The down-sampling frame rate was selected considering the trade-off between a high overlapping between subsequent frames and the presence of a sufficiently long baseline to guarantee a good intersection between the projection rays. Then, the images were preprocessed by applying the calibration parameters explained above. Moreover, the depth images were interpolated (sampling rate 4:1) to obtain a resolution similar to the RGB one, ensuring that all the channels of the RGB-D images generated from the RGB and depth ones are characterized by the same level of information. In fact, each RGB-D image is composed by 6 channels: the first 3 channels correspond to the RGB image, while the other 3 contains the coordinates of the point cloud generated from the depth images. The great innate advantage of RGB-D data is that they exploit the complementary nature of visual images and point clouds, for recovering both camera poses and 3D modelling.

To introduce the Kinect observations into the extended Kalman filter presented in Section 6, the RGB-D images were preprocessed, in order to determine the angular variation and the displacement between two subsequent epochs. The core of the algorithm used to compute these quantities is the function presented

in Xiao et al. (2013) applied between each couple of subsequent RGB-D images. In particular, the RGB channels were used for extracting Scale Invariant Feature Transform (SIFT) keypoints (Lowe, 2004), which were used for computing a first approximation of relative orientation parameters, using a 3-points algorithm combined with RANSAC (Fischler and Bolles, 1987) for outliers removal. Then, this solution was refined by using the Iterative Closest Point (ICP) algorithm on the remaining channels (namely from 4 to 6, corresponding to the depth generated point clouds). Note that an RGB-D image-only trajectory can be computed incrementally, by sequentially adding the estimated rototranslation to the previous step solution.

4. CANON EOS M RGB DATA PROCESSING

The Canon EOS M is a mirrorless interchangeable lens camera (MILC). This kind of cameras are equipped with a single and removable lens and uses a digital display system rather than an optical viewfinder (OVF). The term "mirrorless" is used to highlight the fact that the camera does not have an OVF linked to a mirror-box, a movable mirror or a prism, like in conventional Digital Single Lens Reflex camera (DSLR). Therefore, compared to DSLR, MILCs are mechanically simpler and often smaller, lighter and quieter due to the elimination of the moving mirror for OVF viewing. Because of fewer moving parts the camera is generally more durable. Nevertheless, like a DSLR, a MILC accepts any of a series of interchangeable lenses compatible with the lens mounted of that camera, usually including also fixed focal length ones.

In our experiment we used a Canon EOS M, equipped with a fixed focal length lens. This camera was launched on the market in 2012 and is equipped with an APS-C sensor (22.3×14.9 mm) with 5184×3456 pixels, guaranteeing an image quality comparable to a DSLR, but at lower weight and cost (about €500). The Canon EOS M is able to continuously acquire images at a sampling rate of 2 fps in JPEG format. Moreover, it is able to synchronize the camera time with the GPS one by using a proper external tool (Canon GP-E2). As explained in Section 1, we relied on this device to guarantee the time synchronization with respect to the GNSS data. To avoid blurred images when moving the cart, the shutter time was set to $1/1000$ s and the diaphragm aperture to $f/8$. By fixing these parameters, we deactivated the auto-focus and we manually fixed the focus to the hyperfocal distance, obtaining the maximum depth of field (from about 2 m up to the infinity). Notice that the correct exposure is guaranteed by automatic varying of the sensor sensibility at each shot.

The imaging sensor has to be geometrically calibrated to use the camera for metric applications. This calibration consists in estimating the intrinsic camera parameters, together with radial and tangential distortions parameters. This is performed by the Camera Calibrator app, which is embedded into the Matlab Computer Vision System Toolbox, and by using the provided black and white checkerboard.

To introduce the Canon EOS M observations into the extended Kalman filter presented in Section 6, the RGB images were preprocessed, in order to determine the angular variation and the direction of the displacement between two subsequent epochs. In fact, without using any GCP it is not possible to determine the scale factor of the baseline between each couple of images. The relative orientation between each subsequent couple of images is computed by first detecting SURF keypoints (Bay et al., 2008) and then deriving the essential matrix using the Structure from Motion Matlab libraries embedded in the Computer Vision System Toolbox (Torr and Zisserman, 2000). From the estimated

essential matrix the relative rotation and translation between the two camera poses are then computed (Longuet-Higgins, 1981). Note that the translation vector is given with respect to the reference system defined by the first image and up to a scale factor, assuming its modulus unitary.

5. U-BLOX GNSS DATA PROCESSING

Among various evaluation kits currently available on the market, u-blox ones stand out for the completeness and clarity of the documentation, as well as for the broad configuration options. For this reason, the u-blox evaluation kit EVK-6T was used in this work. In order to increase the quality of the signal reception, the patch antenna ANN-MS-0-005 included in the kit was replaced by a Tallysman TW-3070 single frequency cone antenna. The latter offers much better performances without increasing the overall cost too much, since the price of the antenna is of the order of €100. The LEA-6T u-blox receiver included in the kit is a GPS L1-only receiver providing raw code and phase data in u-blox UBX binary format, through a COM port. The free and open-source goGPS software (Realini and Reguzzoni, 2013; Herrera et al., 2016) was used to record the data stream from the receiver and to process it. goGPS is a software package designed and initially developed at Politecnico di Milano. The development is now carried out mostly by GREd srl, with the contributions of users from several institutions, at international level. goGPS has applications in navigation (Realini et al., 2012), but also in other fields such as building monitoring (Caldera et al., 2016) and troposphere estimation (Barindelli et al., 2018).

To introduce the GNSS observations into the extended Kalman filter presented in Section 6, they were preprocessed to determine the position of the antenna at each epoch. The u-blox data processing by goGPS consists in applying code and phase double differences with respect to a permanent station. In our experiment we used the MILA station located about 500 m far from the experiment area. The adjustment procedure was carried out by means of an extended Kalman filter on double-differenced L1 observations, at 1 Hz, with float phase ambiguities (Realini and Reguzzoni, 2013). The effect of the filter dynamics is disabled by setting the model error standard deviation to a value significantly higher than the observation error standard deviation. This is basically equivalent to performing a kinematic solution based on phase-smoothed code observations. Satellite orbits and clocks were modelled by broadcast ephemeris data, the elevation angle cut-off was set to 15° and observations were weighted based on the sine of the elevation, squared.

6. EXTENDED KALMAN FILTER FOR NAVIGATION

Six degrees of freedom describe the motion of a vehicle, e.g. the position of its barycentre O and its attitude at the epoch t_i . The degrees of freedom are reduced to three if the motion occurs in a plane, e.g. a flat field. In this case, the position of O can be expressed by Cartesian coordinates in a local East-North frame (EN) and the attitude of the vehicle by a single angle. Regarding the dynamics of the system between two consecutive epochs, it can be modelled assuming that the barycentre is moving at constant velocity and the attitude is varying at constant angular rate. Therefore, at discrete epochs t_i ($i = 1, 2, \dots$) the following variables describe the state of the system:

- $\mathbf{x}_O^{\text{EN}}(t_i) = [E_O(t_i), N_O(t_i)]^T$, the position of O in the EN frame,

- $\dot{\mathbf{x}}_O^{\text{EN}}(t_i) = [\dot{E}_O(t_i), \dot{N}_O(t_i)]^T$, the velocity of O in the EN frame,
- $\alpha_V^{\text{EN}}(t_i)$, the attitude angle of the vehicle in the EN frame (see Figure 2),
- $\omega_V^{\text{EN}}(t_i)$, the angular rate of the vehicle in the EN frame around the Up axis (see Figure 2).

The system dynamics is given by the following equations:

$$\mathbf{x}_O^{\text{EN}}(t_{i+1}) = \mathbf{x}_O^{\text{EN}}(t_i) + (t_{i+1} - t_i) \dot{\mathbf{x}}_O^{\text{EN}}(t_i) \quad (1)$$

$$\dot{\mathbf{x}}_O^{\text{EN}}(t_{i+1}) = \dot{\mathbf{x}}_O^{\text{EN}}(t_i) + \boldsymbol{\varepsilon}_{\dot{\mathbf{x}}}(t_{i+1}) \quad (2)$$

$$\alpha_V^{\text{EN}}(t_{i+1}) = \alpha_V^{\text{EN}}(t_i) + (t_{i+1} - t_i) \omega_V^{\text{EN}}(t_i) \quad (3)$$

$$\omega_V^{\text{EN}}(t_{i+1}) = \omega_V^{\text{EN}}(t_i) + \varepsilon_{\omega}(t_{i+1}) \quad (4)$$

where the introduction of suitable model errors $\boldsymbol{\varepsilon}_{\dot{\mathbf{x}}}$ and ε_{ω} allow smooth changes in velocity and angular rate, respectively. In matrix notation, the state vector is:

$$\mathbf{X}(t_i) = [\mathbf{x}_O^{\text{EN}}(t_i), \dot{\mathbf{x}}_O^{\text{EN}}(t_i), \alpha_V^{\text{EN}}(t_i), \omega_V^{\text{EN}}(t_i)]^T \quad (5)$$

and the system dynamics is:

$$\mathbf{X}(t_{i+1}) = \mathbf{T}(t_i, t_{i+1}) \mathbf{X}(t_i) + \boldsymbol{\varepsilon}(t_{i+1}) \quad (6)$$

where \mathbf{T} is the non-stationary transition matrix:

$$\mathbf{T}(t_i, t_{i+1}) = \begin{bmatrix} 1 & 0 & t_{i+1} - t_i & 0 & 0 & 0 \\ 0 & 1 & 0 & t_{i+1} - t_i & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & t_{i+1} - t_i \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

and $\boldsymbol{\varepsilon}$ is the model error vector:

$$\boldsymbol{\varepsilon}(t_{i+1}) = [0, 0, \varepsilon_{\dot{E}}(t_{i+1}), \varepsilon_{\dot{N}}(t_{i+1}), 0, \varepsilon_{\omega}(t_{i+1})]^T. \quad (8)$$

In this work, we propose an improvement of the extended Kalman filter for navigation to combine the GNSS and the image-based information presented in De Gaetani et al. (2018), since we include into the state variables also the angular rate ω_V^{EN} . Moreover, we tested the use of RGB or RGB-D cameras, thus two versions of the improved extended Kalman filter are developed, in order to consider the scale factor that is missing in the RGB observations. The filter allows to introduce observations from the different kinds of instruments with different sampling rates. Therefore, calling t_i^G ($i = 1, 2, \dots$) the GNSS receiver observation epochs, t_i^K ($i = 1, 2, \dots$) the RGB-D acquisition epochs and t_i^C ($i = 1, 2, \dots$) the RGB ones, the union of t_i^G and t_i^K (or t_i^C) is the set of epochs t_i at which the state vector is evaluated. As already introduced in Section 2, the acquisition points of the instruments are at different positions with respect to the barycentre O of the vehicle, i.e. the point which the state variables refer to (see Figure 2). Hence, the following vectors are introduced:

$$\mathbf{x}_{GO}^V = \mathbf{x}_O^V - \mathbf{x}_G^V = [\mathbf{x}_{GO}^V, y_{GO}^V]^T \quad (9)$$

$$\mathbf{x}_{KO}^V = \mathbf{x}_O^V - \mathbf{x}_K^V = [\mathbf{x}_{KO}^V, y_{KO}^V]^T \quad (10)$$

$$\mathbf{x}_{CO}^V = \mathbf{x}_O^V - \mathbf{x}_C^V = [\mathbf{x}_{CO}^V, y_{CO}^V]^T \quad (11)$$

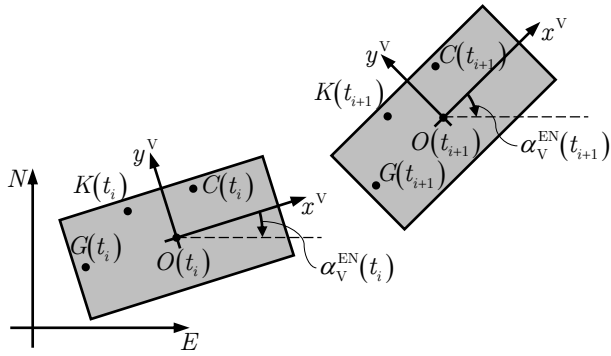


Figure 2. Vehicle at epochs t_i and t_{i+1} . The observation points C , G , K and the vehicle barycentre O are highlighted.

linking the barycentre of the vehicle O to the acquisition points G , K and C referring to the GNSS, RGB-D and RGB devices, respectively. These vectors are given in the vehicle reference system V . Note that they are constant in time and are estimated during the calibration phase described in Section 2.

Regarding the GNSS observations, sampled at the epochs t_i^G , the antenna coordinates estimated by goGPS in the EN frame \bar{x}_G^{EN} are considered as observations, instead of the actually observed pseudo-ranges. Therefore, the GNSS observation equation is:

$$\bar{x}_G^{EN}(t_i^G) = x_O^{EN}(t_i^G) + x_{GO}^{EN}(t_i^G) + \nu_x(t_i^G) \quad (12)$$

where ν_x is the observation error described by the covariance matrix of the estimated coordinates of \bar{x}_G^{EN} . The components of the vector x_{GO}^{EN} , i.e. the vector linking O and G in the EN frame, change in time according to changes of the vehicle attitude. Therefore, Eq. 12 can be rewritten as:

$$\bar{x}_G^{EN}(t_i^G) = x_O^{EN}(t_i^G) - \mathbf{R}(\alpha_V^{EN}(t_i^G)) x_{GO}^V + \nu_x(t_i^G) \quad (13)$$

where $\mathbf{R}(\alpha)$ is the two-dimensional rotation matrix depending on an attitude angle α . Note that Eq. 13 is not linear with respect to the angle $\alpha_V^{EN}(t_i^G)$. Linearizing around $\tilde{\alpha}_V^{EN}(t_i^G) = \alpha_V^{EN}(t_\ell)$ with $t_\ell = \max(t_j : t_j < t_i^G, j = 1, 2, \dots)$, Eq. 13 becomes:

$$\bar{x}_G^{EN}(t_i^G) - h^G(t_i^G) = \mathbf{H}^G(t_i^G) X(t_i^G) + \nu_x(t_i^G) \quad (14)$$

where h^G is the known term:

$$h^G(t_i^G) = -\mathbf{R}(\tilde{\alpha}_V^{EN}(t_i^G)) \left\{ x_{GO}^V + \tilde{\alpha}_V^{EN}(t_i^G) \begin{bmatrix} y_{GO}^V \\ -x_{GO}^V \end{bmatrix} \right\} \quad (15)$$

and \mathbf{H}^G is the GNSS non-stationary transformation matrix:

$$\mathbf{H}^G(t_i^G) = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{R}(\tilde{\alpha}_V^{EN}(t_i^G)) \begin{bmatrix} y_{GO}^V \\ -x_{GO}^V \end{bmatrix} & \mathbf{0} \end{bmatrix} \quad (16)$$

Regarding the RGB-D camera acquisitions, the displacement and the rotation between two subsequent epochs t_{i-1}^K and t_i^K in the vehicle reference frame V defined at t_{i-1}^K are observed. Dividing them by the time lag $\Delta t_i^K = t_i^K - t_{i-1}^K$ and rotating the displacement by the angle $\alpha_V^{EN}(t_{i-1}^K)$, the velocity \dot{x}_K^{EN} and the angular velocity ω_V^{EN} in the EN frame are obtained, referred at the epoch t_i^K . Since they are assumed as direct observations, they are indicated as \bar{x}_K^{EN} and $\bar{\omega}_V^{EN}$, respectively. Note that, since Δt_i^K is the time difference between two RGB-D acquisitions, the average

velocities and not the instantaneous ones at time t_i^K are actually computed. The RGB-D observation equations can be written as:

$$\bar{x}_K^{EN}(t_i^K) = \bar{x}_O^{EN}(t_i^K) + \frac{[\mathbf{I} - \mathbf{R}(\beta_V^{EN}(t_i^K))]}{\Delta t_i^K} x_{KO}^V + \nu_{\dot{x}}(t_i^K) \quad (17)$$

$$\bar{\omega}_V^{EN}(t_i^K) = \omega_V^{EN}(t_i^K) + \nu_{\omega}(t_i^K) \quad (18)$$

where $\beta_V^{EN}(t_i^K) = \omega_V^{EN}(t_i^K) \Delta t_i^K$. Eq. 17 is not linear with respect to the angular velocity $\omega_V^{EN}(t_i^K)$ describing the change of attitude of the vehicle. After linearization around $\tilde{\omega}_V^{EN}(t_i^K) = \omega_V^{EN}(t_\ell)$ with $t_\ell = \max(t_j : t_j < t_i^K, j = 1, 2, \dots)$, Eq. 17 becomes:

$$\bar{x}_K^{EN}(t_i^K) + h^K(t_i^K) = \bar{x}_O^{EN}(t_i^K) + \omega_V^{EN}(t_i^K) \dot{\mathbf{R}}(\beta_V^{EN}(t_i^K)) x_{KO}^V + \nu_{\dot{x}}(t_i^K) \quad (19)$$

where $\tilde{\beta}_V^{EN}(t_i^K) = \tilde{\omega}_V^{EN}(t_i^K) \Delta t_i^K$, $\dot{\mathbf{R}}(\beta) = \partial \mathbf{R} / \partial \beta$, and h^K is the known term:

$$h^K(t_i^K) = \tilde{\omega}_V^{EN}(t_i^K) \dot{\mathbf{R}}(\tilde{\beta}_V^{EN}(t_i^K)) x_{KO}^V + \frac{[\mathbf{I} - \mathbf{R}(\tilde{\beta}_V^{EN}(t_i^K))]}{\Delta t_i^K} x_{KO}^V \quad (20)$$

Eqs. 18 and 19 can be rewritten in matrix notation as:

$$\bar{y}^K(t_i^K) = \mathbf{H}^K(t_i^K) X(t_i^K) + \nu_{y^K}(t_i^K) \quad (21)$$

where $\bar{y}^K(t_i^K)$ is the observation vector given by:

$$\bar{y}^K(t_i^K) = \begin{bmatrix} \bar{x}_K^{EN}(t_i^K) + h^K(t_i^K) \\ \bar{\omega}_V^{EN}(t_i^K) \end{bmatrix} \quad (22)$$

the observation error vector $\nu_{y^K}(t_i^K)$ is:

$$\nu_{y^K}(t_i^K) = \begin{bmatrix} \nu_{\dot{x}}(t_i^K) \\ \nu_{\omega}(t_i^K) \end{bmatrix} \quad (23)$$

and the non-stationary transformation matrix $\mathbf{H}^K(t_i^K)$ is:

$$\mathbf{H}^K(t_i^K) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & a_1^K(t_i^K) \\ 0 & 0 & 0 & 1 & 0 & a_2^K(t_i^K) \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (24)$$

with:

$$\begin{bmatrix} a_1^K(t_i^K) \\ a_2^K(t_i^K) \end{bmatrix} = \dot{\mathbf{R}}(\tilde{\beta}_V^{EN}(t_i^K)) x_{KO}^V \quad (25)$$

Differently to the aforementioned RGB-D acquisitions, the RGB ones suffer the lack of information on the scale of the observed displacements between two subsequent acquisition epochs. In this case only the direction of the motion of the point C is observed, i.e. the unit vector \bar{e}_C^{EN} . The missing scale factor is nothing else than the modulus of the velocity vector of C in the EN frame. Assuming small variations of the attitude angles between two subsequent epochs, the velocity vectors at point O and C can be approximated as parallel. Then, the RGB observation equa-

tions can be written as:

$$\begin{aligned} \|\dot{\mathbf{x}}_O^{\text{EN}}(t_i^C)\| \bar{\mathbf{e}}_C^{\text{EN}}(t_i^C) &= \\ &= \dot{\mathbf{x}}_O^{\text{EN}}(t_i^C) + \frac{[\mathbf{I} - \mathbf{R}(\beta_V^{\text{EN}}(t_i^C))]\mathbf{x}_{CO}^V}{\Delta t_i^C} + \boldsymbol{\nu}_{\dot{\mathbf{x}}}(t_i^C) \end{aligned} \quad (26)$$

$$\bar{\omega}_V^{\text{EN}}(t_i^C) = \omega_V^{\text{EN}}(t_i^C) + \nu_{\omega}(t_i^C). \quad (27)$$

Note that Eq. 26 is not linear with respect to the variable $\dot{\mathbf{x}}_O^{\text{EN}}(t_i^C)$ describing the velocity of the vehicle barycentre, as well as in the angular velocity $\omega_V^{\text{EN}}(t_i^C)$. After linearization around $\tilde{\mathbf{x}}_O^{\text{EN}}(t_i^C) = \mathbf{x}_O^{\text{EN}}(t_\ell)$ and $\tilde{\omega}_V^{\text{EN}}(t_i^C) = \omega_V^{\text{EN}}(t_\ell)$ with $t_\ell = \max(t_j : t_j < t_i^C, j = 1, 2, \dots)$, Eq. 26 becomes:

$$\begin{aligned} \mathbf{h}^C(t_i^C) &= [\mathbf{I} - \mathbf{P}(t_i^C)] \dot{\mathbf{x}}_O^{\text{EN}}(t_i^C) + \\ &+ \omega_V^{\text{EN}}(t_i^C) \dot{\mathbf{R}}(\tilde{\beta}_V^{\text{EN}}(t_i^C)) \mathbf{x}_{CO}^V + \boldsymbol{\nu}_{\dot{\mathbf{x}}}(t_i^C) \end{aligned} \quad (28)$$

where:

$$\mathbf{P}(t_i^C) = \begin{bmatrix} p_{11}(t_i^C) & p_{12}(t_i^C) \\ p_{21}(t_i^C) & p_{22}(t_i^C) \end{bmatrix} = \frac{\tilde{\mathbf{x}}_O^{\text{EN}}(t_i^C)}{\|\tilde{\mathbf{x}}_O^{\text{EN}}(t_i^C)\|} \bar{\mathbf{e}}_C^{\text{EN}}(t_i^C)^\top \quad (29)$$

and \mathbf{h}^C is the known term:

$$\begin{aligned} \mathbf{h}^C(t_i^C) &= \tilde{\omega}_V^{\text{EN}}(t_i^C) \mathbf{R}(\tilde{\beta}_V^{\text{EN}}(t_i^C)) \mathbf{x}_{CO}^V + \\ &+ \frac{[\mathbf{I} - \mathbf{R}(\tilde{\beta}_V^{\text{EN}}(t_i^C))]\mathbf{x}_{CO}^V}{\Delta t_i^C}. \end{aligned} \quad (30)$$

In matrix notation, Eqs. 27 and 28 can be rewritten as:

$$\bar{\mathbf{y}}^C(t_i^C) = \mathbf{H}^C(t_i^C) \mathbf{X}(t_i^C) + \boldsymbol{\nu}_{\mathbf{y}^C}(t_i^C) \quad (31)$$

where the observation vector $\bar{\mathbf{y}}^C(t_i^C)$ is given by:

$$\bar{\mathbf{y}}^C(t_i^C) = \begin{bmatrix} \mathbf{h}^C(t_i^C) \\ \bar{\omega}_V^{\text{EN}}(t_i^C) \end{bmatrix}, \quad (32)$$

$\boldsymbol{\nu}_{\mathbf{y}^C}(t_i^C)$ is the observation error vector:

$$\boldsymbol{\nu}_{\mathbf{y}^C}(t_i^C) = \begin{bmatrix} \boldsymbol{\nu}_{\dot{\mathbf{x}}}(t_i^C) \\ \nu_{\omega}(t_i^C) \end{bmatrix} \quad (33)$$

and $\mathbf{H}^C(t_i^C)$ is the non-stationary transformation matrix:

$$\mathbf{H}^C(t_i^C) = \begin{bmatrix} 0 & 0 & 1 - p_{11}(t_i^C) & -p_{12}(t_i^C) & 0 & a_1^C(t_i^C) \\ 0 & 0 & -p_{21}(t_i^C) & 1 - p_{22}(t_i^C) & 0 & a_2^C(t_i^C) \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (34)$$

with:

$$\begin{bmatrix} a_1^C(t_i^C) \\ a_2^C(t_i^C) \end{bmatrix} = \dot{\mathbf{R}}(\tilde{\beta}_C^{\text{EN}}(t_i^C)) \mathbf{x}_{CO}^V. \quad (35)$$

Once the transition and transformation matrices are derived for any acquisition time, along with the model and observation error covariance matrices, the Kalman filter can iteratively update the state vector providing the estimated trajectory and its error

estimate (Kalman, 1960). A comment is due about the proposed Kalman filter. Displacements and attitude variations of the vehicle are directly observed by the RGB-D device and therefore an additional and predefined dynamics, like the one introduced in Eqs. 1, 2, 3, 4 is not strictly required. This dynamics is useful when only GNSS observations are available or when the RGB-D errors are larger than the expected variability of the vehicle position and attitude. It can be also used to reduce the effect of possible outliers in the observations. On the other hands, the RGB device is able to provide information only on the attitude variations and the direction of motion. In this case, the barycentre constant velocity dynamics plays a key role in the trajectory estimation.

7. TEST RESULTS

The experimental test was performed by moving the cart described in Section 2 at crawl velocity, realizing several closed trajectories along Andrea Cascella's Fountain (located near Piazza Leonardo da Vinci in Milan, Italy). Applying the developed Kalman filter, the trajectory of the cart barycentre O and cart attitude were estimated. In order to assess the performances when combining different kinds of observations, four solutions were computed. In particular, the following observation combinations were considered: GNSS-only, RGB-D-only, GNSS/RGB-D and GNSS/RGB. The variances (and covariances) of the model errors and the observation noise, required by the Kalman filter, were empirically set. This is crucial to control the integration of the data coming from the different sensors and, consequently, the quality of the final result. In the proposed strategy, the required variances are evaluated exploiting the information coming from the reference trajectory, dividing it in rectilinear and curved sections.

Regarding the standard deviation of the GNSS observations (Eq. 12), this is different for each acquisition epoch. The corresponding error covariance matrix is given by the goGPS data processing. However, the overall standard deviation of the u-blox trajectory can be computed by comparing it with the reference one, obtaining a value of about 20 cm. This results required to apply a scale factor of 5 to the formal error covariance matrices estimated by goGPS. The RGB-D camera noise standard deviations of the velocity (Eq. 17) and the angular rate (Eq. 18) have been estimated on the basis of the rectilinear stages of the trajectory. In fact, during these stages a constant velocity and a constant attitude are expected, hence the observation variability can be attributed to the observation noise only. Consequently, they have been set to 8 cm/s and 0.85°/s, respectively. With the same approach, the RGB camera angular velocity error standard deviation (Eq. 27) has been set to 2°/s. This value is larger than the RGB-D one because of the shorter baseline between two consecutive frames due to an higher acquisition frame rate of the RGB camera. Moreover, the RGB camera observes the direction of motion with a noise standard deviation of about 1.2°, empirically evaluated along rectilinear stages. This corresponds to a velocity error standard deviation of the order of 1 cm/s at cruise velocity that is about 35 cm/s. Since the GNSS position is known with an error of about 20 cm and the modulus of the velocity is basically derived from the GPS information, the standard deviation of the noise in Eq. 26 has been set to 20 cm/s.

Regarding model errors of the dynamics used in the Kalman filter, they have been determined on the basis of the reference trajectory and attitude. The standard deviation of the velocity and the angular rate have been evaluated, differently for straight and curved sections. According to the reference trajectory, the velocity error standard deviation (Eq. 2) has been set to 7 cm/s, while the an-

gular rate error standard deviation (Eq. 3) is obviously different for the straight and the curved sections and has been set to $2^\circ/\text{s}$ and $5^\circ/\text{s}$, respectively.

The results of the tests are shown in Figures 3 and 4. As expected, the GNSS-only solution is not able to reconstruct the attitude variation of the cart during its motion (see Figure 4) because the acquisition of the position of a single antenna does not provide any information about it. Therefore, the attitude predictions of the Kalman filter are computed by minimizing the variations of both the cart barycentre velocity and angular rate.

The RGB-D-only solution is instead able to well estimate the attitude of the cart (see Figure 4). However, a drift error causes an estimated trajectory getting farther away from the reference one epoch by epoch (see Figure 3). Nevertheless, the estimated trajectory is smooth and the Kalman filter plays only the role of limiting strong variations between consecutive epochs.

Introducing also GNSS observations and computing the GNSS/RGB-D solution the drift present in the RGB-D-only trajectory is corrected (see Figure 3), at the cost of obtaining a less smooth trajectory due to the irregularities introduced by the GNSS (that is a low-cost device in our experiment). Moreover, this combination leads to an increment of the drift in the estimated cart attitude (see Figure 4) that could be prevented by introducing some GCPs or a dedicated measuring device.

Regarding the GNSS/RGB solution, the estimate of the modulus of the velocity depends from the GNSS data. Therefore, in this solution, the information provided by the GNSS plays a key role, leading to an estimated trajectory closer to the GNSS-only solution than the GNSS/RGB-D one. This implies a worse cart attitude estimation (see Figure 4), producing a larger drift error with respect to the solutions containing RGB-D data. This also implies that at some point the information on the motion provided by the two instruments (i.e. GNSS and RGB camera) becomes slightly inconsistent and, therefore, the alternation between GNSS and RGB observations can produce a less smooth trajectory with a zig-zag behaviour (see e.g. the bottom-left stage in Figure 3) of the order of some centimetres.

Statistics of the difference between the estimated trajectories and the reference one, in terms of Euclidean distance between points at the same epoch, are reported in Table 1.

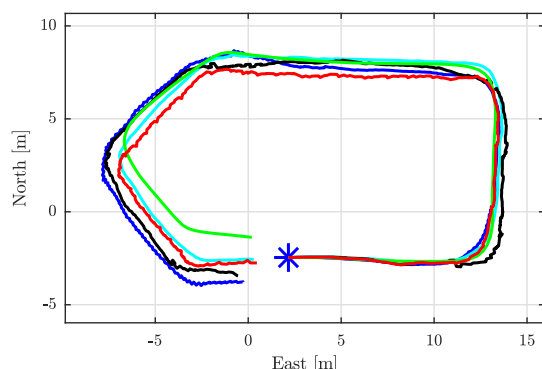


Figure 3. Estimated and reference cart trajectory: GNSS-only solution (black), RGB-D-only solution (green), GNSS/RGB-D combination (red), GNSS/RGB combination (blue), reference (cyan). The blue star represents the starting point of the trajectory.

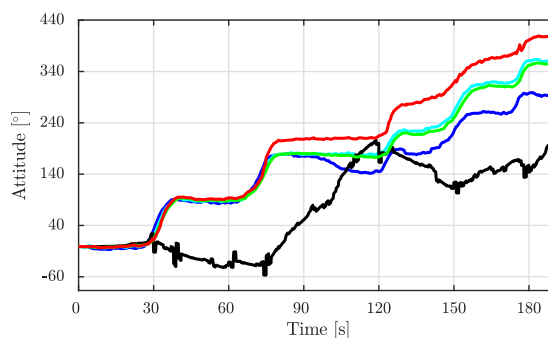


Figure 4. Estimated and reference cart attitude: GNSS-only solution (black), RGB-D-only solution (green), GNSS/RGB-D combination (red), GNSS/RGB combination (blue), reference (cyan).

	GNSS	RGB-D	GNSS/RGB-D	GNSS/RGB
mean [m]	0.71	0.87	0.57	0.69
std [m]	0.36	0.48	0.37	0.40
rms [m]	0.80	0.99	0.68	0.79

Table 1. Statistics of the differences between the estimated trajectories with respect to the reference one, see also Figure 3.

8. CONCLUSIONS

In this work, different versions of a Kalman filter were developed with the aim of estimating a vehicle trajectory and attitude by integrating GNSS and visual data. In particular, low-cost instruments were considered, such as a u-blox GNSS receiver, a Microsoft Kinect RGB-D camera and a Canon EOS M RGB camera. The filter was tested on data acquired during a field test consisting in a moving cart with the above mentioned devices mounted on-board. In addition, more sophisticated and expensive instruments were simultaneously used, in order to get the reference trajectory and attitude to be compared. In particular, the focus of the work was on the comparison between RGB-D and RGB devices. Both of them are able to observe the changes in attitude of the cart during its motion, but the former provides also information about its velocity, while the latter furnishes observations just on the direction of motion. Accordingly to this difference between the device acquisitions, the Kalman filter was modified and different estimated trajectories were obtained. When RGB-D and RGB observations are integrated to GNSS ones, the GNSS/RGB-D solution reconstructs the cart trajectory slightly better than the GNSS/RGB one. In terms of rms of the distances between the reference and the estimated trajectory, the RGB-D solution is about 15% better than the RGB one. However, it is worth to notice that the RGB camera is simpler to manage and does not have constraints regarding the distance between the camera and the object, as the used RGB-D camera has. This partly compensates the slightly worse performance in the trajectory reconstruction. Improvements could come from the integration of other devices that directly observe the vehicle attitude and not only its variation, thus controlling the attitude drift error. Furthermore, the generalization to a 3D motion is a fundamental step to apply this methodology to vehicles commonly used in surveying, such as UAS. Finally, a full integration of the proposed approach into the goGPS free and open source software package would make it usable in many applications.

REFERENCES

- Barindelli, S., Realini, E., Venuti, G., Fermi, A. and Gatti, A., 2018. Detection of water vapor time variations associated with heavy rain in northern Italy by geodetic and low-cost GNSS receivers. *Earth, Planets and Space* 70(1), pp. 28.
- Bay, H., Ess, A., Tuytelaars, T. and Van Gool, L., 2008. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding* 110(3), pp. 346–359.
- Caldera, S., Realini, E., Barzaghi, R., Reguzzoni, M. and Sansò, F., 2016. Experimental study on low-cost satellite-based geodetic monitoring over short baselines. *Journal of Surveying Engineering* 142(3), pp. 04015016.
- Chaplin, B. A., 1999. *Motion estimation from stereo image sequences for a mobile mapping system*. MSc Thesis, Dept. of Geomatics Engineering, University of Calgary.
- Da Silva, J. F. C., Camargo, P. d. O. and Gallis, R., 2003. Development of a low-cost mobile mapping system: a South American experience. *The Photogrammetric Record* 18(101), pp. 5–26.
- De Gaetani, C. I., Pagliari, D., Realini, E., Reguzzoni, M., Rossi, L. and Pinto, L., 2018. Improving Low-Cost GNSS Navigation in Urban Areas by Integrating a Kinect Device. In: *International Association of Geodesy Symposia*.
- Fankhauser, P., Bloesch, M., Rodriguez, D., Kaestner, R., Hutter, M. and Siegwart, R., 2015. Kinect v2 for mobile robot navigation: Evaluation and modeling. In: *Advanced Robotics (ICAR), 2015 International Conference on*, IEEE, pp. 388–394.
- Fischler, M. A. and Bolles, R. C., 1987. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: M. A. Fischler and O. Fischler (eds), *Readings in Computer Vision*, Elsevier, pp. 726–740.
- Foix, S., Alenya, G. and Torras, C., 2011. Lock-in time-of-flight (ToF) cameras: A survey. *IEEE Sensors Journal* 11(9), pp. 1917–1926.
- Georgy, J., Noureldin, A., Korenberg, M. J. and Bayoumi, M. M., 2010. Low-cost three-dimensional navigation solution for RISS/GPS integration using mixture particle filter. *IEEE Transactions on Vehicular Technology* 59(2), pp. 599–615.
- Henry, P., Krainin, M., Herbst, E., Ren, X. and Fox, D., 2012. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *The International Journal of Robotics Research* 31(5), pp. 647–663.
- Herrera, A. M., Suhandri, H. F., Realini, E., Reguzzoni, M. and de Lacy, M. C., 2016. goGPS: open-source MATLAB software. *GPS solutions* 20(3), pp. 595–603.
- Kalman, R. E., 1960. A new approach to linear filtering and prediction problems. *Transactions of ASME, Series D, Journal of Basic Engineering* 82(1), pp. 35–45.
- Longuet-Higgins, H. C., 1981. A computer algorithm for reconstructing a scene from two projections. *Nature* 293(5828), pp. 133–135.
- Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2), pp. 91–110.
- May, S., Werner, B., Surmann, H. and Pervolz, K., 2006. 3D time-of-flight cameras for mobile robotics. In: *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, IEEE, pp. 790–795.
- Noureldin, A., Karamat, T. B., Eberts, M. D. and El-Shafie, A., 2009. Performance enhancement of MEMS-based INS/GPS integration for low-cost navigation applications. *IEEE Transactions on Vehicular Technology* 58(3), pp. 1077–1096.
- Nur, K., Feng, S., Ling, C. and Ochieng, W., 2013. Integration of GPS with a WiFi high accuracy ranging functionality. *Geospatial Information Science* 16(3), pp. 155–168.
- Oliver, A., Kang, S., Wünsche, B. C. and MacDonald, B., 2012. Using the Kinect as a navigation sensor for mobile robotics. In: B. McCane, S. Mills and J. Deng (eds), *Proceedings of the 27th Conference on Image and Vision Computing New Zealand*, ACM, New York, NY, USA, pp. 509–514.
- Pagliari, D. and Pinto, L., 2015. Calibration of Kinect for Xbox One and comparison between the two generations of Microsoft sensors. *Sensors* 15(11), pp. 27569–27589.
- Pagliari, D., Cazzaniga, N. E., Pinto, L., Reguzzoni, M. and Rossi, L., 2018. Joint Use of Image-Based and GNSS Techniques for Urban Navigation. In: R. Cefalo, J. B. Zieliński and M. Barbarella (eds), *New Advanced GNSS and 3D Spatial Techniques*, Springer, pp. 139–153.
- Pagliari, D., Pinto, L., Reguzzoni, M. and Rossi, L., 2016. Integration of Kinect and low-cost GNSS for outdoor navigation. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 41(B5), pp. 565–572.
- Prusak, A., Melnychuk, O., Roth, H., Schiller, I. and Koch, R., 2008. Pose estimation and map building with a time-of-flight-camera for robot navigation. *International Journal of Intelligent Systems Technologies and Applications* 5(3-4), pp. 355–364.
- Realini, E. and Reguzzoni, M., 2013. goGPS: open source software for enhancing the accuracy of low-cost receivers by single-frequency relative kinematic positioning. *Measurement Science and Technology* 24(11), pp. 115010.
- Realini, E., Yoshida, D., Reguzzoni, M. and Raghavan, V., 2012. Enhanced satellite positioning as a web service with goGPS open source software. *Applied Geomatics* 4(2), pp. 135–142.
- Scaramuzza, D. and Fraundorfer, F., 2011. Visual odometry [tutorial]. *IEEE Robotics & Automation Magazine* 18(4), pp. 80–92.
- Torr, P. H. and Zisserman, A., 2000. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding* 78(1), pp. 138–156.
- Xiao, J., Owens, A. and Torralba, A., 2013. SUN3D: a database of big spaces reconstructed using SfM and object labels. In: *Computer Vision (ICCV), 2013 IEEE International Conference on*, IEEE, pp. 1625–1632.