

Article

Tightly-Coupled GNSS/Vision Using a Sky-Pointing Camera for Vehicle Navigation in Urban Areas

Paul Verlaine Gakne *  and Kyle O'Keefe 

Position, Location and Navigation (PLAN) Group, Department of Geomatics Engineering, Schulich School of Engineering, University of Calgary, 2500 University Drive, N.W., Calgary, AB T2N 1N4, Canada; kpgokeef@ucalgary.ca

* Correspondence: pvgakne@ucalgary.ca; Tel.: +1-403-210-9798

Received: 27 February 2018; Accepted: 9 April 2018; Published: 17 April 2018



Abstract: This paper presents a method of fusing the ego-motion of a robot or a land vehicle estimated from an upward-facing camera with Global Navigation Satellite System (GNSS) signals for navigation purposes in urban environments. A sky-pointing camera is mounted on the top of a car and synchronized with a GNSS receiver. The advantages of this configuration are two-fold: firstly, for the GNSS signals, the upward-facing camera will be used to classify the acquired images into sky and non-sky (also known as segmentation). A satellite falling into the non-sky areas (e.g., buildings, trees) will be rejected and not considered for the final position solution computation. Secondly, the sky-pointing camera (with a field of view of about 90 degrees) is helpful for urban area ego-motion estimation in the sense that it does not see most of the moving objects (e.g., pedestrians, cars) and thus is able to estimate the ego-motion with fewer outliers than is typical with a forward-facing camera. The GNSS and visual information systems are tightly-coupled in a Kalman filter for the final position solution. Experimental results demonstrate the ability of the system to provide satisfactory navigation solutions and better accuracy than the GNSS-only and the loosely-coupled GNSS/vision, 20 percent and 82 percent (in the worst case) respectively, in a deep urban canyon, even in conditions with fewer than four GNSS satellites.

Keywords: visual odometry; upward-facing camera; motion estimation; satellites; GNSS; tightly-coupled integration; vehicle navigation; image segmentation; clustering algorithms

1. Introduction

Autonomous vehicles rely on navigation sensors such as GNSS receivers, inertial navigation systems (INS), odometers, LiDAR, radar, etc. However, none of these sensors alone is able to provide satisfactory position solutions in terms of accuracy, availability, continuity and reliability all the time and in all environments. For example, INS are immune to interference. However, their performance degrades quickly when updates from other systems such as GNSS are not available. This is generally the typical scenario that is observed in deep downtown areas as presented in [1–3]. Moreover, integrating INS and GNSS might not be enough to ensure the availability of the position solutions with a certain accuracy and reliability. To avoid these kinds of situations, Gao et al. [4] proposed the integration of GNSS, INS and LiDAR. This is done to alternatively take advantage of the good performance of GNSS in open-sky and the rich LiDAR measurements in occluded outdoor environments for integration with the INS. The same reason drove [5] to combine GNSS, INS, an odometer and an omnidirectional camera. In open-sky scenarios (with no interference or spoofing), relying on GNSS for lane-level navigation is feasible. However, in challenging environments such as deep urban canyons, the GNSS signals are often blocked, reflected or shadowed. This significantly degrades the solution provided by the GNSS receiver. To overcome the disadvantages of GNSS in

urban canyons, fusion with map data and other sensors is often employed (e.g., the Google self driving car [6]). However, combining sensors is expensive both in terms of cost and the computational load.

Besides sensors fusion, many multipath mitigation methods have been implemented in the literature, including: (i) antenna-based multipath detection/mitigation (antenna design [7], choke rings, controlled reception pattern antenna [8], angle of arrival [9], multiple antennas, dual-polarization antenna); (ii) receiver based (code discriminator design [10], early-late correlator comparison, Doppler domain, carrier smoothing, vector tracking, etc.); (iii) C/N_0 -based multipath detection (C/N_0 -based selection and weighting [11], multi-frequency C/N_0 -based multipath detection, etc.); (iv) NLOS detection using a sky-pointing camera [12–14] and a 3D building model [15–17]. The research proposed in this paper uses a sky-pointing camera setup similar to what is used in [12–14]. The proposed method differs from the previous works in two ways. Previous works use a fish-eye camera in order to observe satellites at all elevations. However, low elevation satellites are not very beneficial for the position solution computation since most of the time they are corrupted by the tropospheric [18], ionospheric and multipath errors (causing the receiver to lose the lock on them frequently). Moreover, fish-eye cameras are not yet widely available on mass market products such as cellphones or car-mounted cameras. This paper uses a narrow field of view camera. This is practical since the method can be implemented on mass-market products that already incorporate such cameras. Furthermore, this setup helps to automatically exclude low elevation satellites. The second contribution regarding the multipath mitigation in this work is the segmentation algorithm proposed. A fast and robust segmentation algorithm tailored for navigation (with an upward-facing camera) is introduced.

Since almost all modern vehicles are equipped with multiple cameras, they constitute an attractive option for navigation. Some research has focused on using only the camera for navigation purposes. In [19], simultaneous localization and mapping (SLAM) methods are proposed using different camera configurations (monocular versus stereo), as well as different types of cameras (monocular/stereo visual spectral cameras versus RGB-D cameras). The proposed methods were tested on multiple publicly available datasets and showed consistency in terms of accuracy. An improved visual odometry and a robust visual localization using an omnidirectional camera are presented in [20,21], respectively. In the development of their improved omnidirectional visual odometry, the authors in [20,21] adapted the epipolar constraint to the omnidirectional camera sensor. They also improved the matching process by propagating the current uncertainty of the system. Overall, their method has proven to be robust because their adaptive matching process reduces false positives.

Another “stand-alone” solution available in the literature consists of using map data as presented in [22,23]. In order to facilitate the segmentation of the camera images into sky and non-sky, Petovello and He [23] use an infra-red camera, while a visual spectral camera is used in [22]. Moreover, Gakne and O’Keefe [22] proposed an improved image segmentation algorithm tailored for sky-pointing camera-based navigation. Practically, both methods consist of determining the vehicle position by matching skylines obtained from a 3D building model with the skylines observed with a camera mounted on the top of the vehicle. Using the skyline as a fingerprint reduces the vehicle localization problem to a place recognition problem. This approach could be useful in cases where it is difficult to obtain the position solution from any type of sensor since it can provide the user position even in the absence of GNSS. Readers can refer to the mentioned references for more details.

Integration between inertial measurement units (IMU) and cameras has also been extensively studied. A method of estimating robot attitude angles by fusing IMU data with a monocular vision system data is proposed in [24]. A six degrees of freedom (6-DOF) inertial sensor was fused with a low cost camera for an autonomous mobile robot navigation system using an extended Kalman filter. The proposed method has been shown to be effective in terms of computation load and attitude estimation accuracy.

More than two sensors can also be integrated. In [25], GNSS, INS and Ultra-Wideband (UWB) were integrated in a tightly coupled way using a robust Kalman filter that supported Vehicle to Infrastructure (V2I) communication. Furthermore, a monocular camera, IMU and GNSS were integrated for vehicle

navigation in harsh environments in [26,27]. In [2], the navigation performance of GNSS/INS/vision integration was evaluated via simulated data, where the number of the observed satellites was continually decreased from three to one GPS satellites. As expected, the position error increased as the number of satellites decreased.

Finally, GNSS/vision integration has been presented in [28,29]. The vision system was combined with a differential GPS (DGPS) in [28]. The presented work focused on evaluating the heading obtained from the combined solution for micro-Unmanned Aerial Vehicle (UAV) navigation. Both simulated and field experiments showed that the DGPS/vision approach outperformed the fused onboard data (autopilot, magnetometer). A deeply-coupled GNSS/vision approach for vehicle navigation was introduced in [29]. The GNSS/vision system was able to provide continuous navigation in challenging environments that outperformed the GNSS/INS system.

The research proposed in this paper fits in this category with the contrast that this work uses only line-of-sight satellites selected by segmenting an upward-facing camera image into sky and non-sky areas. Moreover, our tightly-coupled implementation takes care of the scale factor issue that affects monocular vision systems.

It should be noted that the above-mentioned approaches do not represent an exhaustive list of sensor fusion approaches that exist in the literature. An autonomous car such as Faraday's FF 91 fuses more than 30 sensors including cameras, radars and ultrasonic sensors. This list of approaches is given for illustration of what exists in the literature pertaining to the integration of optical sensors with various other types of sensors.

The objective of this paper is to evaluate the performance of a system that integrates monocular visual odometry (implemented with a camera pointing upward) and GNSS signals. Since GNSS signals are subject to severe multipath in urban canyons, an image-based multipath mitigation approach was developed. This approach consists of segmenting an image into sky and non-sky areas. Images taken with an upward-facing camera are complex and difficult to segment, we introduce a tailored segmentation method appropriate for vehicle navigation applications.

The remainder of this paper is organized as follows: Section 2 briefly presents the existing works related to this research; Section 3 describes the vehicle motion estimation steps using a camera; Section 4 implements the improved segmentation approach, as well as the Non-Line-Of-Sight (NLOS) mitigation algorithm; Section 5 presents the GNSS/vision integration strategy we propose; Section 6 describes the experiment, results and analysis; and Section 7 draws some conclusions.

2. Background and Related Works

The research presented in this paper involves many fields including image processing and computer vision, GNSS, as well as estimation for navigation (Kalman filtering).

The image-based NLOS satellites exclusion used in this research heavily relies on the accuracy of the segmentation algorithm [14,30,31]. Let us assume that due to a segmentation error, a building is classified as sky. A satellite that is projected in that area will not be excluded, i.e., will be considered as a Line-Of-Sight (LOS) satellite (but actually it is not). Given that even a single GNSS signal affected by multipath can cause a significant position error, this could significantly degrade our GNSS-only solution. Few segmentation algorithms have been evaluated in the literature [12,14]. In Attia et al. [12], a set of supervised and unsupervised (pixel) classifiers were compared, and it was concluded that the proposed method (i.e., the Geodesic Reconstruction by Dilatation (GRD) coupled to the Fisher clustering algorithm) was more efficient than the other evaluated classifiers. However, the Fisher classifier takes about 1.03 s to process one image. This is computationally heavy compared to the method proposed in [14], where it was found that the Otsu method outperforms the other considered algorithms (Meanshift, HMRF-EM, graph-cut) for this specific upward-pointing camera application. The Otsu method segments an image with a resolution of 1288×728 in 0.015 s and provides an accuracy of 94.7%. To increase the accuracy of the segmentation algorithms, Petovello et al. and Meguro et al. [23,30] used infrared cameras. However, to date, these types of cameras are not yet

available on mass market devices. This leads us to choose a visual spectrum standard camera with a narrow Field Of View (FOV) and suggests a need to improve the segmentation algorithm. A fast and accurate segmentation method is presented in Section 4.

Obtaining the position of a robot or vehicle using visual features is one of the most studied problems in the computer vision community. The subject is extensively discussed in [32]. There are several approaches that are usually classified based on how feature points are matched. These include 2D-to-2D matching, 3D-to-3D and 3D-to-2D [32]. An example of monocular visual odometry using 3D-to-2D matching is presented in [33] where a database is used to store a set of point clouds (feature database) registered with intensity information. 2D features are detected in an image captured by the camera, and the match is done with the content of the database. Then, by applying the PnP algorithm, the camera's rotation and translation are obtained. In contrast, in the present paper, we propose a 2D-to-2D approach where 2D feature points are detected on successive images and matching and tracking is used to obtain the position of the camera with respect to the tracked features. Other references such as [20,34–36] integrate visual odometry into SLAM applications. This not only provides an advantage of allowing the methods to be used for multiple purposes (positioning, navigation and mapping) but also increases the robustness of these methods. In contrast, in this work, visual odometry based on feature point detection, matching and tracking is used since there is no need in our application (so far) to build a map as the vehicle navigates.

GNSS fundamentals are highly documented in the literature [37]. In open-sky, GNSS performs well. However, in harsh environments, it performs poorly. In such areas where some weak GNSS signals are available, high sensitivity receivers using massive numbers of parallel correlators and longer integration times can be used. However, this approach involves higher processing, complexity and power costs [38,39]. Different GNSS receiver architectures that can improve the receiver performance are also introduced in [40,41].

Many works related to GNSS/vision vary from the type of the measurements used (pseudorange versus carrier phase) or the type of the integration (EKF, loosely coupled, tightly coupled or ultra-tightly coupled). Some examples of these types of integration can be found in [2,29,42,43]. In this work, selected line-of-sight satellite measurements are tightly integrated with the visual odometry. The satellite selection, discussed previously, helps to reduce the multipath effect, and tightly coupling the selected satellites with the vision ensures that the GNSS measurements are used even if fewer than four line-of-sight satellites are available.

3. Vehicle Motion Estimation Using an Upward-Facing Camera

This section discusses the steps required for vehicle ego-motion estimation.

3.1. Camera Calibration and Image Rectification

Similar to most computer vision applications, camera calibration and image rectification are two important steps in the development of the proposed method. The calibration method described in [44] is used in this paper as implemented in the OpenCV library [45]. It consists of determining the camera matrix, as well as the distortion coefficient matrix. In fact, there are two main lens distortions: the radial and the tangential distortions. The former are caused by the shape of the lens, while the later arise from manufacturing defects resulting from the lens not being exactly parallel to the imaging plane. Concerning the radial distortion rectification, given a current distorted pixel \tilde{p} with coordinates (\tilde{x}, \tilde{y}) , the radial location of \tilde{p} will be rescaled [46] on the undistorted output image as:

$$x = \tilde{x}(1 + r_1d^2 + r_2d^4 + r_3d^6) \quad (1)$$

$$y = \tilde{y}(1 + r_1d^2 + r_2d^4 + r_3d^6) \quad (2)$$

where:

r_i ($i=1,2,3$) denotes the radial lens distortion parameters;
 x and y are the new coordinates of the pixel as a result of the correction;
 $d = \sqrt{(\tilde{x} - x_c)^2 + (\tilde{y} - y_c)^2}$ is the distance of the distorted coordinates to/from the principal point. x_c and y_c are the coordinates of the principal point.

Regarding the tangential distortion, it is corrected by using:

$$x = \tilde{x} + 2t_1\tilde{x}\tilde{y} + t_2(d^2 + 2\tilde{x}^2) \quad (3)$$

$$y = \tilde{y} + t_1(r^2 + 2\tilde{y}^2) + 2t_2\tilde{x}\tilde{y} \quad (4)$$

where t_j ($j=1,2$) are the tangential distortion parameters.

Besides the lens distortion correction, the camera focal length f and the principal point coordinates (x_c, y_c) are determined to build the camera matrix. More details can be found in [46]. The algorithm successively estimates the parameters using the closed-form solution [44] and maximum-likelihood estimation using all detected corners on a chessboard in multiple views. It minimizes the error between the projected object space coordinates on the image with the estimated parameters and the measurements of the feature points [29], using the Levenberg–Marquardt algorithm [47]. Once the image is rectified and the intrinsic parameters of the camera obtained, further processing can be carried out.

3.2. Feature Detection, Description and Matching

Feature extraction and matching constitute essential processes for the platform (robot, vehicle) motion estimation. In this work, we use ORB (Oriented FAST and Rotated BRIEF) [48] as the feature detector and descriptor algorithm. ORB combines an improved version of the Feature from the Accelerated Segment Test (FAST) algorithm [49] for feature detection, with the Binary Robust Independent Elementary Features (BRIEF) [50] descriptor extraction process. Its main advantages are its low run-time and its ability to detect a relatively large number of features.

3.2.1. Feature from Accelerated Segment Test Feature Detector

FAST considers a pixel at the centre of a candidate corner p and performs a test on a circle of sixteen pixels around p as shown in Figure 1a. If twelve of those pixels are brighter than the pixel at the centre of the candidate point, then p is a corner. To speed up the processing, FAST starts by setting a threshold intensity value T (e.g., 20% of the intensity of p). Then, it compares the intensity of the pixels at Indexes 1, 5, 9 and 13, as highlighted in Figure 1b with T . If at least three out of four of these pixels' intensities are above or below the threshold (threshold test), the test continues with the other sixteen pixels, and at least twelve of these pixels intensities should satisfy the threshold test to declare p as an interest point. If at the first level there are not at least three (out of four) pixels' intensity satisfying the threshold test, FAST rejects the current candidate corner. Despite its computation efficiency, FAST features do not have an orientation component. The enhanced FAST used in the ORB algorithm addresses this issue by using the intensity centroid [48]. This method uses the vector representing the offset of a corner intensity from its centre. The enhanced FAST uses this vector to impute the orientation information.

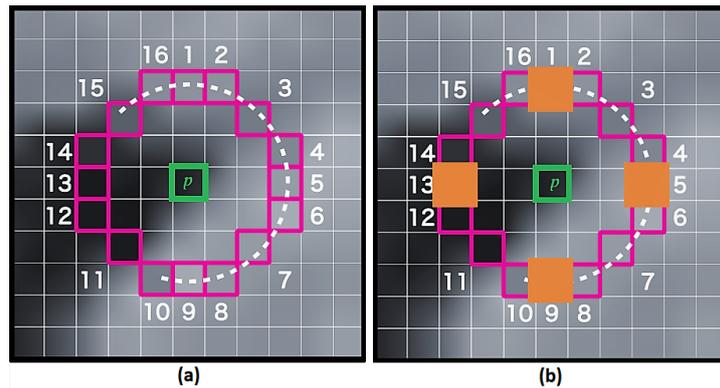


Figure 1. FAST feature detection illustration. (a) p is the candidate corner; pink-highlighted squares are sixteen pixels used in the feature detection; the dashed line passes through twelve consecutive pixels that are brighter than the pixel at p . (b) Same as (a) with the four pixels at Indexes 1, 5, 9 and 13 filled in dark orange used to speed up the threshold test.

3.2.2. Binary Robust Independent Elementary Features Descriptor

The BRIEF algorithm described in [50] is not designed to be rotationally invariant and tolerates small amounts of rotation only. In other words, if two images of the same viewpoint are taken with a large camera rotation ($\gtrsim 10^\circ$), the feature description using BRIEF will fail. Indeed, given n ($n = 128, 256$ or 512) predefined pixel pairs within an image patch, BRIEF uses local binary tests to produce n -bit strings. A so obtained vector representation is very sensitive to rotation variation. Thus, ORB uses a rotation-invariant version of BRIEF (rBRIEF (rotation-aware BRIEF)) presented in [48].

After applying the FAST and BRIEF algorithms, the Hamming distance [51] is used to match the obtained strings of descriptors. However, false matches might happen between certain very similar features in the viewpoints. These types of mismatches can cause severe ego-motion estimation error. Thus, the outliers need to be removed.

3.3. Outlier Rejection

Due to the dense traffic in the urban canyons, clutters and moving objects such as cars or pedestrians are captured by the camera. The detected corners on these objects will be considered by the tracking algorithm as fixed points, while in reality, the features are dynamic. This constitutes the major source of outliers in urban areas where humans move actively. Although our configuration (with a sky-pointing camera) is beneficial in the sense that it will capture very few moving objects (depending on the FOV of the camera), in order to make the developed method camera configuration independent, the random sample consensus (RANSAC) algorithm [52] is used. The advantage of this method is its robustness when facing a high number of outliers, as well as its low computational load. RANSAC randomly selects a subset of feature points from the full set of the tracked features. The non-considered feature points are then reprojected using the estimated states. The errors between the measured image coordinates and the computed reprojection coordinates are tested using the specified measurement accuracies, the estimated state accuracy, as well as the defined confidence level. The process is repeated to refine the remaining feature points that are roughly consistent with the previous estimates. The inliers are those feature points that are consistent, and the outliers are those that are not. This procedure is repeated a fixed number of times.

3.4. Motion Estimation Process

Once the correspondences are established between the detected features, the next step is the estimation of the essential matrix E . The matrix E represents the geometric relations between two

consecutive images I_{k-1} and I_k . It contains the camera motion parameters up to an unknown scale factor for the translation [32] in the form:

$$\mathbf{E}_k \simeq \hat{\mathbf{t}}_{c_k} \mathbf{r}_{c_k} \quad (5)$$

where:

k denotes the image frame number;

$$\hat{\mathbf{t}}_{c_k} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix};$$

\mathbf{r}_{c_k} is the rotation matrix;

and $\mathbf{t}_{c_k} = [t_x \ t_y \ t_z]^T$, where t_x , t_y and t_z are the relative translations following the camera axes.

Equation (5) is resolved based on the five-point algorithm solver [53], which provides a certain robustness in the presence of outliers.

The camera translation up to a scale, as well as its rotation are thus obtained for two consecutive images and can be used in the subsequent steps in our integration algorithm. One of the most important properties of the motion estimation in this paper is the epipolar constraint that can be described by the essential or the fundamental matrix between two camera positions. The epipolar constraint provides that the projection of a point on the actual image frame I_k must fall in the intersection of an epipolar plane with the image plane [53,54]. By this process, it is possible to verify if two 2D feature points correspond to the same 3D point. Thus, the epipolar constraint determines the line on which the feature point on the image I_k falls in the image I_{k-1} . This constraint is formulated as:

$$p'^T \mathbf{E} p = 0 \quad (6)$$

The step by step determination of the essential matrix is presented in [53]. Once \mathbf{E} is estimated from the matched feature points, the relative rotation and translation of the camera are recovered by using the cheirality check [53]. The essential matrix is first decomposed using Singular Value Decomposition (SVD) [54] to obtain four possible poses (rotation/translation), then the verification of the possible pose hypotheses by doing the cheirality check is performed. The cheirality check consists of verifying that the feature point lies in front of the camera (i.e., the triangulated feature point should have positive depth). By this process, the correct rotation \mathbf{r}_c and translation \mathbf{t}_c can be identified and used for the GNSS and vision integration (see Section 5). However, as mentioned previously, using a monocular system, the magnitude of the translation between two consecutive images suffers from scale ambiguity. The integration with GNSS helps to overcome this issue. The visual odometry steps [55] are depicted in Figure 2.

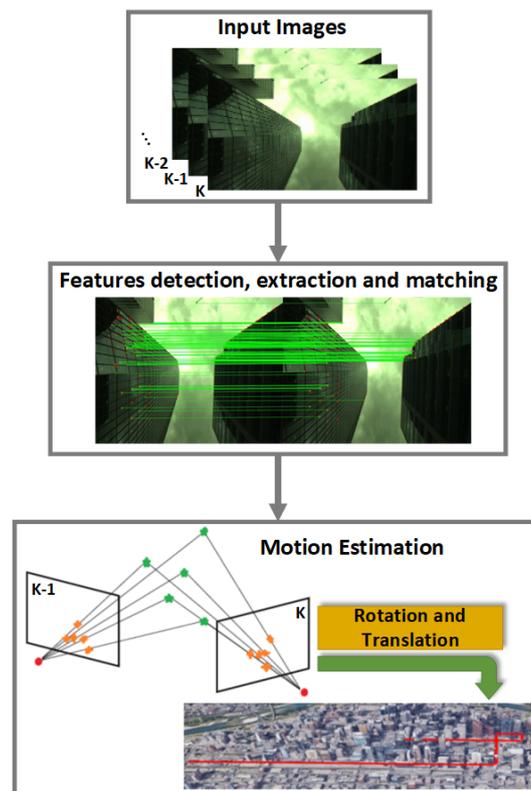


Figure 2. Feature points-based visual odometry.

3.5. Computing the Rotation and Translation Using the Singular Value Decomposition Approach

Given two sets of feature points $f_{p_1} = \{p_0, p_2, \dots, p_{M-1}\}$ and $f_{p_2} = \{p'_0, p'_2, \dots, p'_{M-1}\}$ (M is the total number of feature points), the process described here consists of determining the rotation and translation based on the least squares approach using the SVD. Having the sets of the feature points, we are looking for the rotation \mathbf{r}_c and translation \mathbf{t}_c such that:

$$(\mathbf{r}_c, \mathbf{t}_c) = \arg \min_{\mathbf{r}_{c_{min}}, \mathbf{t}_{c_{min}}} \sum_{j=0}^{M-1} w_j \|(\mathbf{r}_{c_{min}} p_j + \mathbf{t}_{c_{min}}) - p'_j\|^2 \quad (7)$$

where $w_j > 0$ is the weight of each point pair.

The algorithm for computing \mathbf{r}_c and \mathbf{t}_c is summarized in Algorithm 1. More details can be found in [56].

From the rotation matrix, the pitch, roll and yaw of the platform can be determined by a simple conversion of the rotation matrix to Euler angles (for example). It should be noted that, instead of using the visual odometry to determine the platform orientation, the concept of vanishing points can also be used. This is done in [55,57].

Algorithm 1: Rotation and Translation Computation Algorithm.

Input : $f_{p_1} = \{p_0, p_2, \dots, p_{M-1}\}$ and $f_{p_2} = \{p'_0, p'_2, \dots, p'_{M-1}\}$
Output: \mathbf{r}_c and \mathbf{t}_c

// initialization
 $\mathbf{r}_c = \mathbf{I}_{3 \times 3}$
 $\mathbf{t}_c = \mathbf{0}_{3 \times 1}$

// iterate to the total number of feature points
for $j = 0, 1, 2, \dots, M - 1$ **do**

// compute the weighted centroids for both inputs f_{p_1} and f_{p_2}
 $\bar{p} = \frac{\sum_j w_j p_j}{\sum_j w_j}$ and $\bar{p}' = \frac{\sum_j w_j p'_j}{\sum_j w_j}$

// compute the centred vectors
 $c_j = p_j - \bar{p}$ and $c'_j = p'_j - \bar{p}'$

end

// compute the covariance matrix
 $\text{Cov} = \mathbf{C} \times \text{diag}(w_0, w_1, \dots, w_{M-1}) \times \mathbf{C}'^T$, \mathbf{C} and \mathbf{C}' = matrices with c_j and c'_j as their columns

// determine the SVD of Cov
 $\text{Cov} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$

// compute the rotation

$$\mathbf{r}_c = \mathbf{V} \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & \det(\mathbf{V}\mathbf{U}^T) \end{bmatrix} \mathbf{U}^T$$

// compute the translation
 $\mathbf{t}_c = \bar{p}' - \mathbf{r}_c \bar{p}$

return: \mathbf{r}_c , \mathbf{t}_c

4. Camera-Based Non-Line-Of-Sight Effect Mitigation

This section presents the vision-based method we used to reduce and mitigate the NLOS GNSS effects.

4.1. Image Segmentation-Based NLOS Mitigation Algorithm

Image segmentation is a process that consists of partitioning the image into two or more portions called segments. The resulting output image is a simplified version of the input image that depicts “meaningful” regions or objects only. The NLOS mitigation method used in this paper consists of detecting and rejecting the obstructed satellites that are the principal sources of the multipath and NLOS errors. Our method uses a sky-pointing camera to acquire the GNSS receiver antenna’s surrounding environment [14,31,58]. The captured images need to be segmented into sky and non-sky. The tracked satellites are then reprojected onto the segmented image. Finally, the satellites that lie on the non-sky areas in the image are removed before computing the final position solution. The concept has been presented in our previous work [14] where we evaluated various segmentation algorithms in different weather conditions. The steps of this method are depicted in Figure 3.

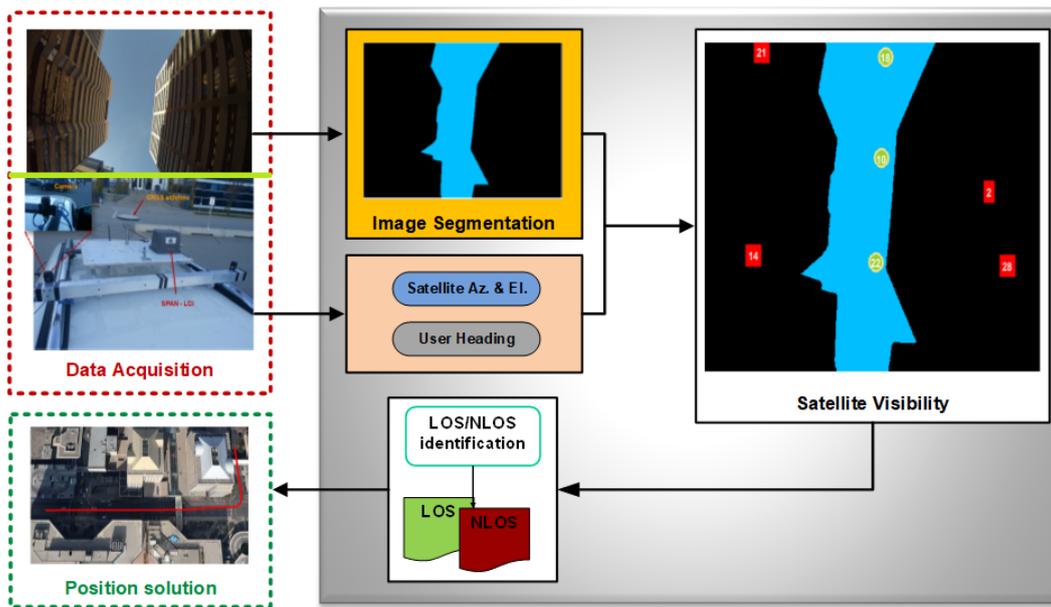


Figure 3. NLOS effect mitigation algorithm steps: “Data Acquisition” box: synchronized GNSS receiver and monocular system (down) and an example of the captured image (top); “Image Segmentation” box: the blue area in the segmented image represents the sky and the black area is the non-sky area (obstacle); “Satellite Visibility” box: the green circles are the non-obstructed satellites, and the red rectangles are the obstructed satellites that will be ignored.

However, segmenting an image into sky and non-sky is not an easy task due to lighting variations, weather conditions and the facades of buildings (e.g., buildings with glass). Figure 4A,B,D, shows few challenging images to segment (in sunny and cloudy weather conditions).

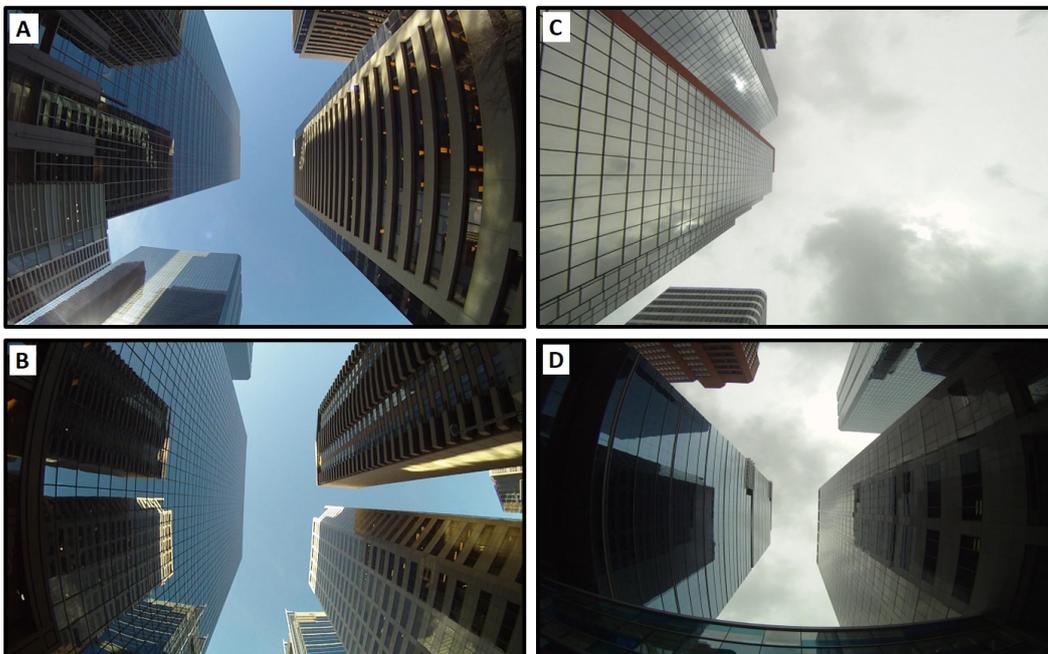


Figure 4. Example of complex downtown images taken using an upward-facing camera. Correct image segmentation can be affected by the environment illumination, building facades (reflections observed on the glasses facades). Left (A,B): sunny weather condition; right (C,D): cloudy weather condition.

Using conventional image segmentation algorithms, most of them initially developed for different purposes than outdoor navigation (e.g., biomedical applications), will produce poor results. Figure 5 shows the Otsu method segmentation applied to binarize an image. Figure 5b shows the Otsu method without pre-processing. Figure 5c shows the Otsu method with pre-processing that we introduced in [14].

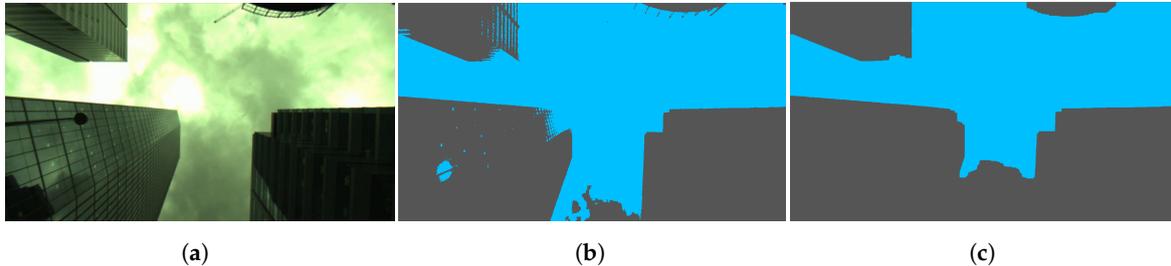


Figure 5. Image segmentation using the Otsu methods. (a) Original; (b) Otsu; (c) Improved Otsu [14].

It can clearly be seen that accurately segmenting this image is challenging, and segmentation errors are observed both in Figure 5b and Figure 5c. Surrounding buildings and sky are reflected and visible on the buildings with glass facades. Partitioning such images in a way that pixels belonging to sky and buildings share the same label with regard to features such as colour or texture is generally difficult to achieve.

In this work, we introduce an improved segmentation approach tailored for vision-based outdoor vehicular navigation applications. The algorithm consists of using edges in order to partition an image into sky and non-sky segments. Objects (buildings, sky, trees, etc.) in an image visually occupy regions. These regions are separated by using algorithms such as the canny edge detector [59]. The centre of the image is assumed to be a sky segment. This assumption makes sense, since in normal conditions, a sky-pointing camera mounted on the top of a land vehicle driven on the road will not be inclined in a such way that the centre of the image is obstructed by a building. From the middle of the image, the following pixels are “filled” as sky until we reach an edge. Then, anything beyond the edge is non-sky. However, as shown in Figure 4C, clouds (or sun in some other cases (not shown)) can cause the sky area to appear non-homogeneous. The canny edge detector in this case will detect edges within sky area and thus leads to segmentation errors. To avoid these kinds of issues, the developed algorithm includes four steps:

- Image filtering: Given that the images captured using a sky-pointing camera are highly corrupted by bright (e.g., sun light) and dark (e.g., buildings or clouds) objects/structures, we adopted a sequential application of open-close filters denoted Alternate Sequential Filter (ASF) [60]. Indeed, we observed that when the noise is wide-spread over an image, using a single open-close filter with a large structuring element leads to segmentation errors (bright objects tend to be lost and the difference between the sky segment and the other structures in the image becomes hard to observe). ASF provides efficient results since it alternates openings and closings proceeding in an incremental way from small to a given size of the structuring element m , $m \geq 1$ [60]. Consider Ω_m and ζ_m the morphological opening and closing of size m , respectively. The ASF is a sequential combination of Ω_m and ζ_m such as $\gamma_m = \Omega_m \zeta_m$ is a morphological filter. Thus, we have:

$$ASF_m = \gamma_m \dots \gamma_2 \gamma_1 \quad (8)$$

For illustration, if $m = 2$, we have $ASF_2(I) = \Omega_2[\zeta_2(\Omega_1[\zeta_1(I)])]$, where I is the image to filter. The result obtained by using the filter defined in Equation (8) is a less noisy image than the original image. The different portions of the image are more obvious after this step. However, the output of the ASF still provide edges within the sky areas. Since our approach uses edges to

categorize sky and non-sky, it is important to remove such edges. For this reason, the levelling algorithm [61] is used along with ASF to find a good trade-off between good identification of edges between sky and buildings and suppression of edges within the same structures (sky and other objects in the image).

- Colour space conversion: once the image is filtered, we determine that the perceptual brightness (luminance) of the image is enough to accurately distinguish the main partitions contained in the image since it depicts sharp details. For this reason, the RGB (Red, Green and Blue) image is converted to the Luv colour space. The luminance channel L is then extracted for further processing.
- Edge detection: the luminance channel extracted from the filtered image is smooth and suitable for edge detection with limited errors. The edge detection here consists of finding discontinuity in the luminance of the pixels within the image. The well-known canny edge detector [59], which consists of smoothing the image with a Gaussian filter, computing horizontal and vertical gradients, computing the magnitude of the gradient, performing non-maximal suppression and performing hysteresis thresholding, is used in this paper.
- Flood-fill algorithm application: At this point, the decision should be made on which edges mark the limit between sky and non-sky areas. The flood-fill step is initialized by assuming the pixel at the centre of the image as belonging to the sky category. Then, the pixels from the centre of the image are filled until we reach an edge. In other words, we used the output of the edge detector algorithm as a mask to stop filling at edges. This process is illustrated in Figure 6.

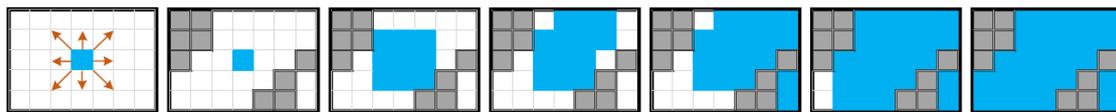


Figure 6. Flood-fill algorithm illustration. Pixels are filled from the centre of the image until an edge is reached following the eight arrows indicated in the first (left-most) box.

Figure 7 shows the steps of the image segmentation algorithm we implemented. The original image is a relatively complex image containing sun light, clouds, reflective building facades, etc. The second box represents a smoother (filtered) image. The small bright spots observed on the building facades are filtered out, and structures look sharper and less influenced by the sun. The third box depicts the luminance channel of the filtered image represented in the Luv colour space. Dominant segments are kept, and lighting effects decrease, while less dominant segments look smoother. This reduces/eliminates edges within the same structures, which leads to accurate edge detection between segments, as depicted in the fourth box. Accurate edge detection naturally leads to accurate image segmentation (fifth box). In the segmented image (fifth box), the sky is represented in blue colour, and the non-sky or obstacles are represented in dark grey colour.

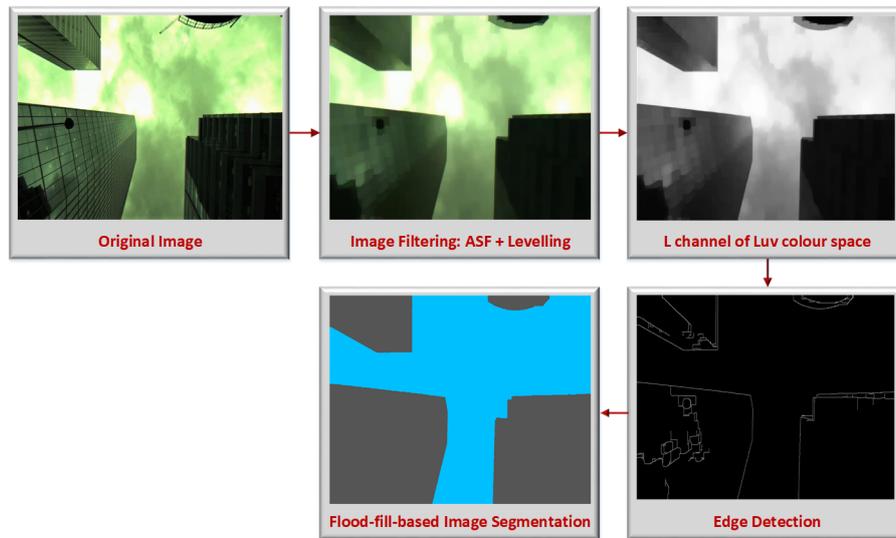


Figure 7. Flood-fill-based image segmentation steps. Following the arrow direction we have the following: first box: original image; second box: filtered RGB image; third box: luminance channel extracted from the RGB to Luv colour space conversion; fourth box: detected edges; fifth box: image segmentation result.

4.2. Satellite Projection and Rejection

The method used in this paper consists of projecting all the tracked satellites onto the successive images and classifying them into LOS satellites (in the sky area) and NLOS satellites (in non-sky area). To project the satellites on an image, we need:

- The distance from the centre of the image in pixels ($d_{c_{pix}}$): this corresponds to the elevation angle of the satellite (ϵ_{sat}),
- The azimuth within an image: for this, the heading of the platform is required.

The geometry shown in Figure 8 is used to determine the distance ($d_{c_{pix}}$, in pixels) from the centre of a calibrated image to a satellite's projection onto the image plane.

To determine $d_{c_{pix}}$ corresponding to ϵ_{sat} , we defined a mapping function, which is a function of the focal length f and the angle Ψ (see Figure 8). Assuming that the optical centre is constantly zenith pointing, the angle Ψ is defined as:

$$\Psi = \frac{\pi}{2} - \epsilon_{sat} \quad (9)$$

Having a calibrated camera and from Figure 8 and Equation (9), we obtain:

$$d_{c_{pix}} = f_{pix} \cdot \tan\left(\frac{\pi}{2} - \epsilon_{sat}\right) \quad (10)$$

where f_{pix} is the focal length in pixels.

Thus, given the centre of the image coordinates (x_c, y_c) , the platform heading, as well as the satellite azimuth, the coordinates of the projection of the satellite on the image (in pixel) are given by Equations (11a) and (11b) below:

$$x_{Sat_{img}} = x_c + d_{c_{pix}} \cdot \cos(\alpha_{platform} + \alpha_{sat}) \quad (11a)$$

$$y_{Sat_{img}} = y_c - d_{c_{pix}} \cdot \sin(\alpha_{platform} + \alpha_{sat}) \quad (11b)$$

where:

$(x_{Sat_{img}}, y_{Sat_{img}})$ are the projected satellite coordinates on the image plane;

α_{platform} is the heading of the platform;
 α_{sat} is the satellite azimuth.

An example of the satellite projection result at a given vehicle location is depicted in Figure 9. The non-obstructed satellites, also called LOS satellites, are those that lie in the sky area represented with the green circles. In Figure 9, they are PRN6, PRN17, PRN19 and PRN28. The obstructed satellite (NLOS satellite, red circle) in this case is PRN9.

Thus, at this specific epoch, PRN9 will be rejected, and PRNs 6, 17, 19 and 28 will be used for further processing.

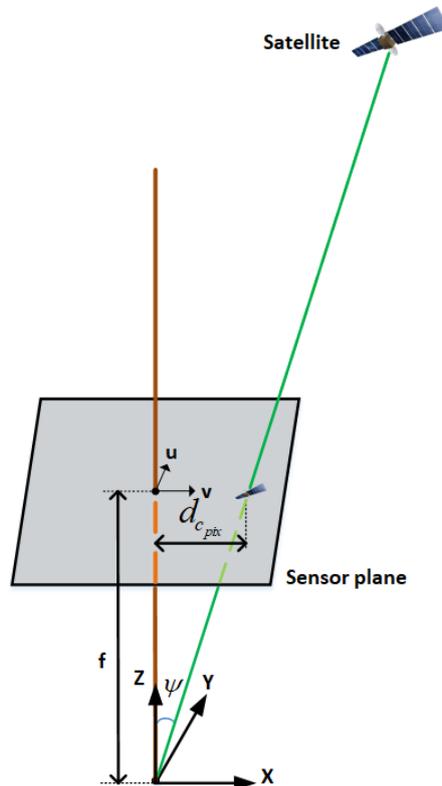


Figure 8. Modelling of the satellite projection on the image.

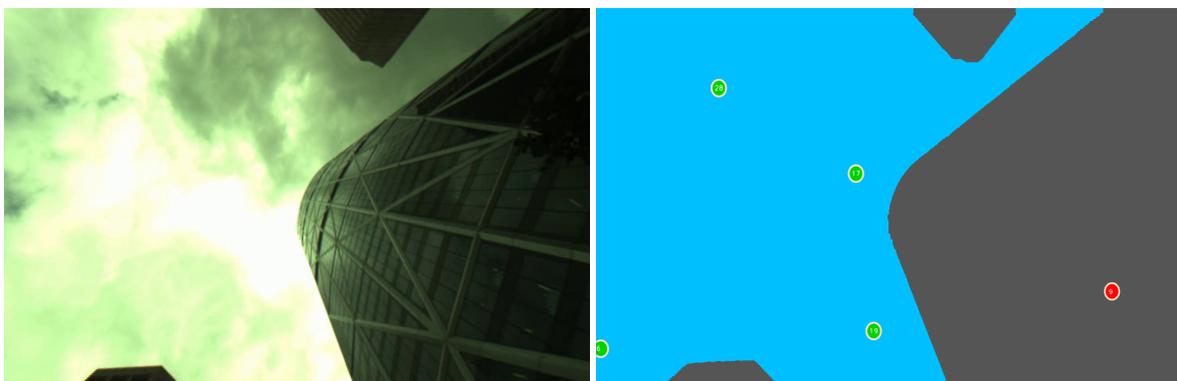


Figure 9. Example of satellites' classification based on their obstruction status. Left: original image; right: satellites projection and classification. Obstructed, NLOS (PRN9), and non-obstructed, LOS (PRN6, PRN17, PRN19, PRN28), satellites are represented in red and green, respectively.

5. Algorithm Development: GNSS/Vision Integration

In this paper, we are proposing the integration of the GNSS raw measurements with the vision motion estimation in a tightly-coupled manner. The proposed approach diagram architecture is depicted in Figure 10.

The acquired images (from the vision equipment box) are rectified and used in two different ways: firstly, the images are not modified in order to implement the visual odometry and obtain the camera pose, as well as its relative position; secondly, the “copy” of these images is processed (segmented following the steps presented in Figure 7) to implement the multipath mitigation algorithm. Only the selected satellites’ (LOS satellites) raw measurements are used in the filter.

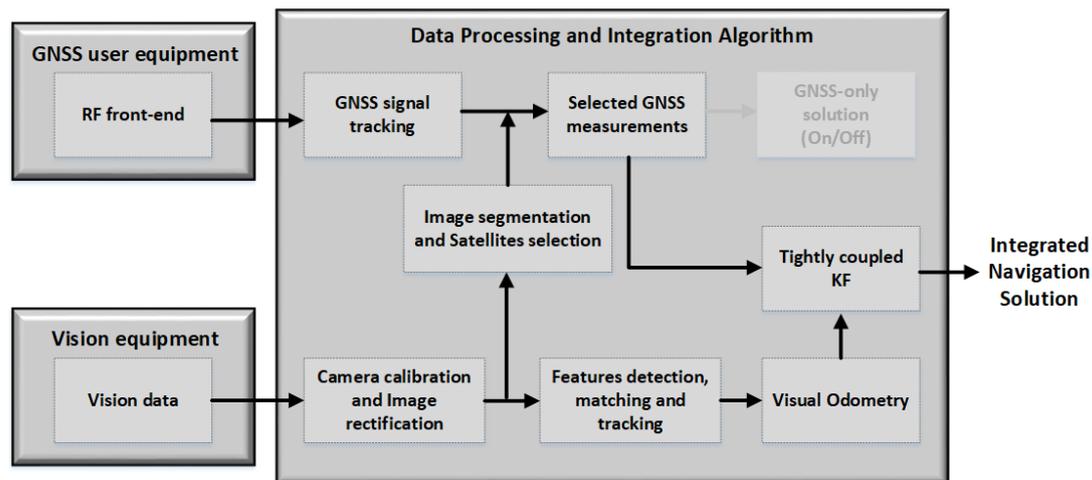


Figure 10. The proposed GNS/vision tightly-coupled integration’s general architecture.

5.1. Global Navigation Satellite Systems

From all the GNSS measurements [37] available, in this paper, we use the pseudorange and pseudorange rate (Doppler). The carrier phase and the accumulated delta range are left for future work.

5.2. Pseudorange Observation

The pseudorange observation between the user and a satellite is related to the user position and clock states as:

$$\rho_i = |r_i - r_u| + cdt + \mu_{\rho_i} \quad (12)$$

where:

- ρ_i is the pseudorange of the i^{th} satellite;
- r_i denotes the i^{th} satellite’s position at the transmission time;
- r_u represents the user position at the reception time;
- cdt is the receiver clock bias;
- μ_{ρ_i} denotes the sum of all errors on the measurement;
- $|\bullet|$ denotes the magnitude of a vector.

Given an a priori estimate of the state, an estimate of the clock bias and the errors, the estimate of the pseudorange measurement is given by:

$$\hat{\rho}_i = |r_i - \hat{r}_u| + c\hat{d}t + \hat{\mu}_{\rho_i} \quad (13)$$

where all $(\hat{\cdot})$ represent the corresponding estimates of the parameters as defined in Equation (12).

Thus, the residual is expressed as:

$$\delta\rho_i = \rho_i - \hat{\rho}_i = \begin{bmatrix} \hat{\mathbf{1}}_i^T & 1 \end{bmatrix} \begin{bmatrix} r_u - \hat{r}_u \\ c(dt - \hat{d}t) \end{bmatrix} + \mu_{\rho_i} - \hat{\mu}_{\rho_i} \quad (14)$$

where $\hat{\mathbf{1}}_i^T = \frac{r_i - \hat{r}_u}{|r_i - \hat{r}_u|}$ is the unit vector from the estimated user position to the satellite.

Likewise, the pseudorange rate observation is given by the following:

$$\dot{\rho}_i = (v_i - v_u) \bullet \frac{r_i - r_u}{|r_i - r_u|} + d + \mu_{\dot{\rho}_i} \quad (15)$$

where:

v_i and v_u are the i th satellite and user velocities, respectively, expressed in the Earth-Centred Earth-Fixed (ECEF) coordinate frame;

d is the receiver clock drift in m/s;

$\mu_{\dot{\rho}_i}$ represents the ensemble of errors in the measurement in m/s;

(\bullet) denotes the dot product.

The estimate of the pseudorange rate is expressed as:

$$\hat{\dot{\rho}}_i = (v_i - \hat{v}_u) \bullet \hat{\mathbf{1}}_i + \hat{d} + \hat{\mu}_{\dot{\rho}_i} \quad (16)$$

where $(\hat{\cdot})$, as previously, represents the estimates of the corresponding parameters.

From Equations (15) and (16), we obtain the linearized pseudorange rate measurement given by the equation:

$$\delta\dot{\rho}_i = \dot{\rho}_i - \hat{\dot{\rho}}_i = \begin{bmatrix} -\hat{\mathbf{1}}_i^T & 1 \end{bmatrix} \begin{bmatrix} v_u - \hat{v}_u \\ d - \hat{d} \end{bmatrix} + \mu_{\dot{\rho}_i} - \hat{\mu}_{\dot{\rho}_i} \quad (17)$$

From the above equations, the set of GNSS observations (\mathbf{z}_{GNSS}) is used in the form of measurements:

$$\delta\mathbf{z}_{GNSS} = \begin{bmatrix} \delta\rho_0 \\ \delta\rho_1 \\ \vdots \\ \delta\rho_n \\ \delta\dot{\rho}_0 \\ \delta\dot{\rho}_1 \\ \vdots \\ \delta\dot{\rho}_n \end{bmatrix} \quad (18)$$

5.3. Visual Odometry

The visual odometry estimate vector \mathbf{x}_c consists of rotations \mathbf{r}_c and translations \mathbf{t}_c (where the subscript c stands for the camera) that can be represented as:

$$\mathbf{x}_c = \begin{bmatrix} \mathbf{r}_c & \mathbf{t}_c \end{bmatrix}^T \quad (19)$$

In fact, the camera and the land vehicle coordinate frames are not the same. Thus, the rotations and the translations of the camera should be transformed into the vehicle coordinate frame following Equations (20) and (21).

$$\mathbf{r}_{c_b} = \mathbf{R}_c^b \mathbf{r}_c \mathbf{R}_b^c \quad (20)$$

where \mathbf{R}_c^b is the matrix used for the rotation from the camera to the body frame, \mathbf{R}_b^c the rotation matrix from the body frame to the camera frame and \mathbf{r}_{c_b} represents the rotated \mathbf{r}_c into the body frame. Likewise, the translation should also be rotated into the body frame as follows:

$$\mathbf{t}_{c_b} = \mathbf{R}_c^b (\mathbf{t}_c + \mathbf{r}_c l_b^c - l_b^c) \quad (21)$$

where l_b^c denotes the lever arm from the camera to the body frame.

5.4. Tightly-Coupled GNSS/Vision

After applying our multipath mitigation algorithm, there are still a few GNSS satellites (mostly less than four) that are available. Theoretically, at least four satellites are needed to compute the GNSS-only navigation solution [37]. This situation leads us to make a choice of the tightly-coupled GNSS/vision Kalman filter since even with fewer than four satellites, it is possible to use the raw GNSS measurement to improve the efficiency of vision-based navigation.

The system model used is given by:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{G}(t)w(t) \quad (22)$$

where:

- \mathbf{x} and \mathbf{F} are the state vector and the dynamics matrix, respectively;
- \mathbf{G} represents the shaping matrix;
- w is a vector of zero-mean, unit variance white noise.

Thus, the dynamic system is expressed in the form:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) \quad (23)$$

The dynamic matrix is approximately constant between image frames. Its equivalent in discrete-time is called the state transition matrix $\Phi_{k,k+1}$ that converts the state from epoch k to $k+1$, given by:

$$\Phi_{k,k+1} = \exp(\mathbf{F}\Delta t) \quad (24)$$

Given that the Taylor series [62] of the exponential function is computed as $\exp(A) = \mathbf{I} + A + \frac{A^2}{2} + \frac{A^3}{6} + \dots$, the transition matrix in Equation (24) becomes:

$$\Phi_{k,k+1} = \mathbf{I} + \mathbf{F}\Delta t + \frac{\mathbf{F}^2\Delta t^2}{2} + \frac{\mathbf{F}^3\Delta t^3}{6} + \dots \quad (25)$$

The state vector incorporates the vehicle position, velocity, acceleration, attitude and the receiver clock as shown in Equation (26).

$$\mathbf{x} = \left[\phi \quad \lambda \quad h \quad v \quad a \quad p \quad \dot{p} \quad r \quad \dot{r} \quad A \quad \dot{A} \quad cdt \quad \dot{cdt} \right]^T \quad (26)$$

where:

- ϕ , λ and h represent the position components;
- v and a stand for speed and acceleration, respectively;
- A , p and r are the azimuth, pitch and roll respectively;

(\bullet) represents their corresponding rates.

The error states are:

$$\delta \mathbf{x} = \begin{bmatrix} \delta \mathbf{x}_{vehicle} & \delta \mathbf{x}_{vision} & \delta \mathbf{x}_{clock} \end{bmatrix}^T \quad (27)$$

which is equivalent to:

$$\delta \mathbf{x} = \begin{bmatrix} \delta \phi & \delta \lambda & \delta h & \delta v & \delta a & \delta p & \delta \dot{p} & \delta r & \delta \dot{r} & \delta A & \delta \dot{A} & \delta cdt & \delta \dot{c}dt \end{bmatrix}^T \quad (28)$$

and the receiver clock model is given by:

$$\delta \mathbf{x}_{clock}(t) = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \delta \mathbf{x}_{clock}(t-1) + \mu_{clk} \quad (29)$$

where $\delta \mathbf{x}_{clock} = \begin{bmatrix} \delta cdt & \delta \dot{c}dt \end{bmatrix}^T$

The prediction of the state and the corresponding error covariance matrix are given by Equations (30) and (31), respectively.

$$\hat{\mathbf{x}}_{k+1}^- = \Phi_{k,k+1} \hat{\mathbf{x}}_k^+ \quad (30)$$

$$\mathbf{P}_{k+1}^- = \Phi_{k,k+1} \mathbf{P}_k^+ \Phi_{k,k+1}^T + \mathbf{Q}_k \quad (31)$$

where the superscript minus and plus respectively denote the state before and after an update; and \mathbf{Q}_k is the process noise matrix.

The state and the corresponding error covariance matrix after the update stage are given in Equations (32) and (33).

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \quad (32)$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (33)$$

where:

\mathbf{H} denotes the design matrix, which is the derivative of the measurements with respect to the states;

$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$ represents the Kalman gain.

The term in the Kalman filter state update in the brackets (in Equation (32)) is called the innovation. It is defined as:

$$\mathbf{v}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^- \quad (34)$$

The innovation is interpreted as the new information brought from the measurements and is used in the context of statistical testing and reliability [63] to prevent measurement errors from corrupting the solution.

5.4.1. GNSS Observables

Because tight coupling is used, it is necessary to estimate the receiver clock offset and clock drift in addition to the position and velocity states. Having this in mind, the GNSS measurements that fit the Kalman filter are the error in the range and the error in the range rate (to each satellite), expanded at the current estimate of the states. This is expressed as:

$$\delta \mathbf{z}_{GNSS} = \begin{bmatrix} \rho_i - (\hat{\rho}_i + cdt) \\ \dot{\rho}_i - (\hat{\dot{\rho}}_i + c\dot{d}t) \\ \vdots \end{bmatrix} \quad (35)$$

where the parameters are defined as in Section 5.1.

5.4.2. Vision Observables

The vision observables that are added to the tight integration filter are the rotation and translation around and along the axes, as depicted in Figure 11.

However, as presented in [42], it is also possible to consider the distance and azimuth rate as the vision measurement, by considering the elapsed time between corresponding frames. The vision system measurement vector is given as:

$$\delta \mathbf{z}_{Vision} = \begin{bmatrix} \mathbf{r}_{c_b,k} \\ \mathbf{t}_{c_b,k} \end{bmatrix} \quad (36)$$

where k corresponds to the image frame number.



Figure 11. Upward-pointing camera coordinates system.

5.5. State Estimation and Data Integration

The filter is initialized by the first epoch GNSS-only solution. Monocular visual odometry directly provides the attitude and the translation of the platform. The scale factor can be initialized by considering the two first image frames. In this case, the baseline corresponding to the two locations is synthesized and measured. This information can then be used to derive the depth information in a similar way used for stereo vision-based depth estimation [26,29].

The scale ambiguity was resolved in [26] by integrating the GPS carrier phase measurements with the vision system. In our case, we are using pseudorange and pseudorange rate as the measurement. For simplicity and illustration, let us consider a case where the vehicle moves forward without any rotation (translation only). The vehicle position vector and feature points ranges [26] are related as:

$$H_j \cdot \begin{bmatrix} \Delta r \\ s_j \end{bmatrix} = \mathbf{0}_{2 \times 1} \quad (37)$$

where:

$\mathbf{0}_{2 \times 1}$ denotes a 2×1 vector of zeros;

$j = 1, \dots, M$, with M the total number for features;

H_j is the matrix defining the homogeneous coordinates of the feature points from consecutive image frames;

Δr is the position change vector between consecutive frames;
 s_j denotes the unknown range.

On the other hand, given the pseudorange defined as in Equation (12), the changes in pseudorange measurements between two consecutive images are related to Δr :

$$\Delta \rho_j = \Delta r + c\Delta t + \Delta \mu \quad (38)$$

Combining Equations (37) and (38) defines a system of linear equations that can be used to unambiguously resolve the position changes, as well as the range estimates [26]. This procedure can be used for the vision system ranges and position changes' initialization. The assumption here is that the vehicle frame is aligned with the world frame.

Once the system is initialized, the scale factor can be estimated based on the actual and previous estimated platform position. Thus, we have:

$$\hat{s}_k = \sqrt{(\hat{x}_k - \hat{x}_{k-1})^2 + (\hat{y}_k - \hat{y}_{k-1})^2 + (\hat{z}_k - \hat{z}_{k-1})^2} \quad (39)$$

where:

$(\hat{x}, \hat{y}, \hat{z})$ represent the estimated platform's position.

The filter is implemented with prediction performed at a rate of 10 Hz. This means that at an interval of 100 ms, the system checks if a new GNSS observation or visual odometry estimated parameter is available. As for the method applied in [42], the vision system's updates are processed at 10 Hz, and the update from the GNSS receiver (depicted in Figure 12) is performed at 1 Hz.



Figure 12. The consumer-grade GNSS receiver used.

In the implementation presented herein, the vision measurements are integrated in the form of a filter update. The platform's attitude is obtained by using the velocity estimates, i.e., by taking the arctangent of the ratio between the velocities in the east and north directions, respectively, for the azimuth for example.

6. Experiment and Results

The performance of the proposed approach is demonstrated in this section via a real-world experiment conducted in downtown Calgary.

6.1. Hardware

The experimental system consisted of a camera synchronized with a u-blox 6T GPS receiver rigidly mounted on the top of the car and driven around downtown Calgary (Alberta, Canada). The camera

has a field of view of 90° , and images with a resolution of 1288×728 were taken at a frequency of 10 frames per second. A NovAtel SPAN[®] LCI system that includes a GNSS receiver and an LCI inertial navigation system was used as the reference system. The reference solution is generated post-mission, and its accuracy is at the decimetre-level or better [14]. Figure 13 shows the test vehicle, as well as a zoomed-in view of the equipment mounted on the roof of the vehicle.

The NovAtel GNSS module, as well as the consumer-grade GNSS receiver and the computers that record the images and GNSS data are inside the vehicle and not shown.

The vehicle was driven around areas with tall buildings as well as areas with short buildings (and with buildings on one side of the road). The travelled path and the buildings' height in downtown Calgary are shown in Figure 14. The buildings' height is obtained from the commercially available 3D building model presented in [17,22,64]. Since the camera is pointed upwards, both the NLOS mitigation algorithm and the visual odometry are influenced by the building height. Tall buildings captured by the camera mean that the path between the satellites and the GNSS antenna is highly obstructed. This suggests that the GNSS may perform poorly. Inversely, high buildings in an image provide more feature points to be detected in successive images and, thus, leads to more accurate visual odometry.



Figure 13. Experiment setup. Left: the used vehicle with the mounted equipment; Right: top view of the vehicle.

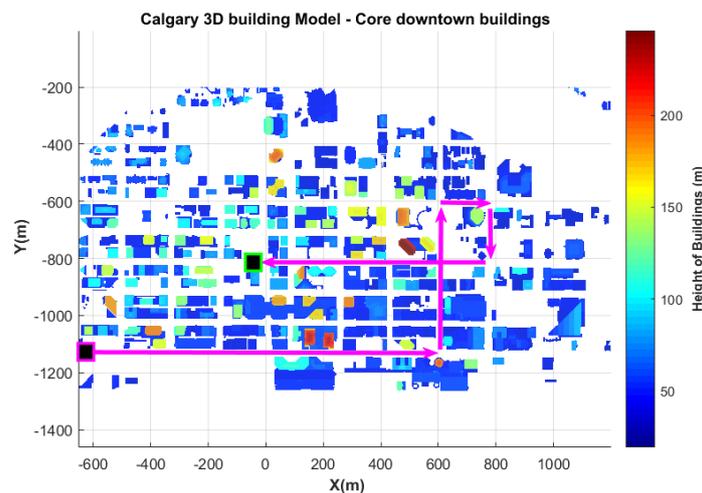


Figure 14. Data collection path travelled. The vehicle starts at the bottom black box (with purple edges), travels following the path indicated with the arrows and stops at the top box (with green edges).

The hardware specifications of the equipment used in this paper are provided in Table 1.

Table 1. Data collection hardware specifications. VFOV, Vertical Field of View.

Reference and GNSS	
SPAN-SE Combination	dual-frequency L1/L2 GPS + GLONASS GPS + GLONASS combined with UIMU-LCI
Camera (Lens Specification, Intrinsic Parameters) and Images	
Aperture	f/1.2—closed
Focal length	2.2 mm
VFOV (1/3°)	90%
Image resolution	1288 × 728
Image frame rate	10 fps
Image centre	(643.5, 363.5)
r_1	−0.26
r_2	0.08
r_3	−0.01
t_1	0
t_2	0

where:

VFOV stands for Vertical Field of View;

r_i ($i=1,2,3$) and t_j ($j=1,2$) are the radial lens and the tangential distortions obtained from the calibration matrix as defined in Section 3.1.

6.2. Results and Interpretations

Our results will be evaluated by using the outputs of three integration approaches that will be compared with the reference trajectory:

- The GNSS-only navigation solution: For this, the PLAN-nav (University of Calgary’s module of the GSNRx™ software receiver) was used. As with most consumer-grade receivers, it uses local level position and velocity states for its GPS-only Kalman filter. In the results, this solution is referred to as GNSS-KF;
- The tightly-coupled GNSS/vision solution: The Line Of Sight (LOS) satellites are first selected. Then, this solution tightly couples the vision-based relative motion estimate to the GNSS. In the results, this is referred to as Tightly-Coupled (TC) GNSS/vision;
- The loosely-coupled GNSS/vision solution integrates measurements from the vision system with the GNSS least squares PVTsolution obtained by using range and range rate observations. Both systems independently compute the navigation solutions, and they are integrated in a loosely-coupled way. This means that if one of the system is unable to provide the solution (e.g., GNSS), then no update from that system is provided to the integration filter. This solution will help to clearly see how beneficial the proposed integration method is, especially when there are fewer than four (LOS) satellites. We refer to this as Loosely-Coupled (LC) GNSS/vision in the text. More details on integration strategies can be found in [10,65].

In this work, only GPS L1 C/A pseudoranges and range rates are used. The entire path travelled during the data collection and the path considered for evaluating our proposed method are depicted in Figure 15 (in red and green, respectively, connecting Points 1, 2, 3, 4 and 5, as indicated by the arrows in Figure 14).

The reason why we choose the green path for our experiment has been discussed in Section 6.1 and is shown in Figure 14.

After the equipment calibration (in the circled area depicted in Figure 15), the vehicle stayed stationary for a few minutes before the experiment started. As observed in [42], at the beginning of the experiment, the initial heading of the GNSS/vision integrated solution can be very inaccurate due to

the fact that the absolute heading is not available when the car is static. However, once the vehicle starts moving, the heading is quickly recovered.

Figure 16 shows the reference (true) trajectory and the position estimation for GNSS-KF, Tightly-Coupled GNSS/vision (TC GNSS/vision), as well as the Loosely-Coupled GNSS/vision (LC GNSS/vision) integrated navigation.



Figure 15. Reference path. Red: the entire travelled path when data were collected; green: the considered path in this paper. 1, 2, 3, 4, 5: portions of the travelled path before turns.

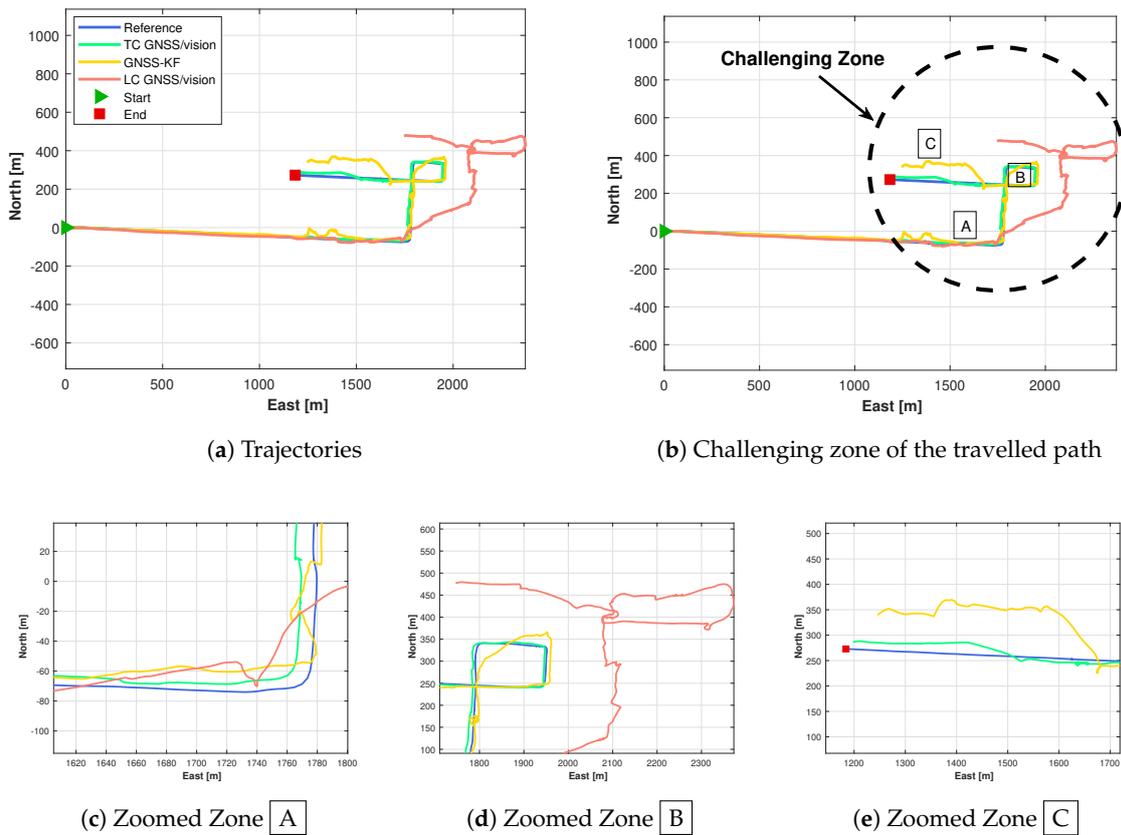


Figure 16. Estimated positions results. (a) Estimated position; (b) highlighted challenging zones; (c) zoomed challenging Zone A; (d) zoomed challenging Zone B; (e) zoomed challenging Zone C. TC, Tightly-Coupled; LC, Loosely-Coupled.

From the plotted trajectory, it can be seen that the presented integration approaches closely follow the reference trajectory during the portion of the path where buildings are low (Path 1, shown in Figure 15). During this portion of the trajectory (mostly with one side of the road non-obstructed), after excluding NLOS satellites, four satellites remain (see Figure 17), and the TC GNSS/vision integrated solution is relatively accurate. However, the second portion of the trajectory that contains a mixture of short and tall (≥ 100 m tall; see Figure 14) buildings (Paths 2, 3, 4 and 5 in Figure 15) is a bit more challenging for two reasons: (i) the first reason is the height of the buildings. Downtown Calgary buildings in general range from 20–50 stories with roads about 15 m wide [64], and even if measurements pass the NLOS selection process, the multipath effect is expected to be large. (ii) The second reason is that the vehicle “stops”. There was severe traffic during this portion of the test. The vehicle was frequently stopping and starting. Since feature points were detected on the sky (and “non-static” clouds), this influenced the accuracy of the visual odometry outputs. When these two phenomena were occurring at the same location, the integrated solution was degraded as shown in Figures 16 and 18. From around Zone A, to the end of the travelled path, there are less than four satellites (after NLOS exclusion) 75.8% of the time. For this reason, the LC GNSS/vision integration, which in this situation relies on the vision system only, performs poorly, while the proposed TC GNSS/vision closely follows the reference solution.

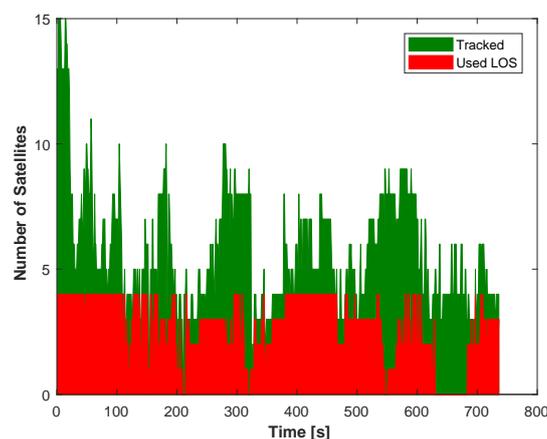


Figure 17. Number of tracked and LOS satellites over time.

Overall, the proposed method provides consistent accuracy over the travelled path with a few exceptions at the turns. At Zones A and B, we can observe that the proposed method does not drift too much while the GNSS-KF provides poor solutions. From Figure 18, it is seen that the solution obtained from the LC GNSS/vision outputs larger error compared to the two other approaches. In Zone A (Figure 16), after NLOS satellites exclusion, there are constantly less than four satellites (Figure 16). As a result, GNSS-aiding is not available, and the obtained navigation solution is poor. The Cumulative Distribution Functions (CDF) of the horizontal errors are provided in Figure 19.

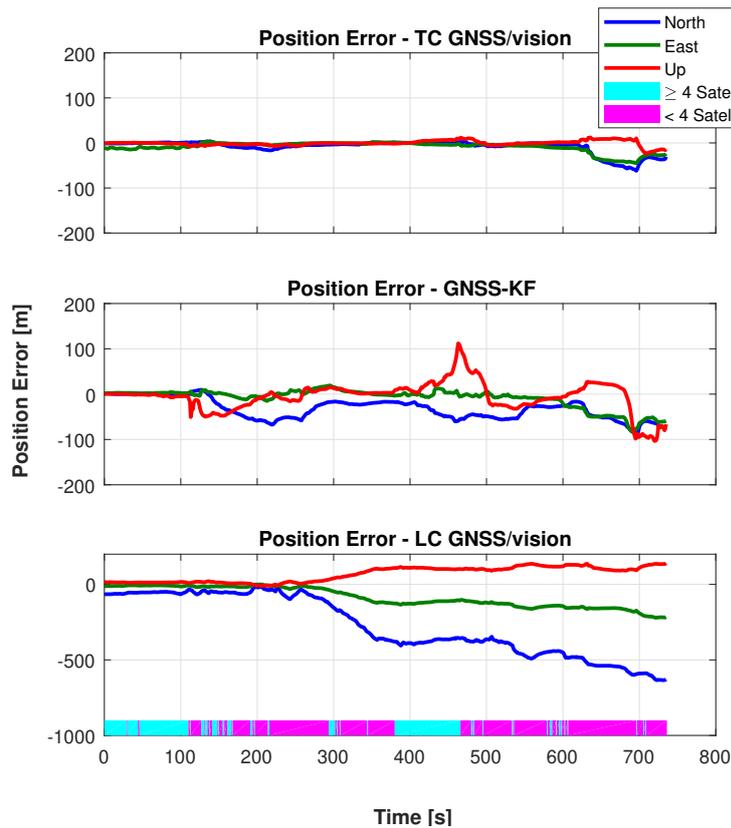


Figure 18. Comparison of position error characteristics between GNSS-KF and the proposed TC GNSS/vision.

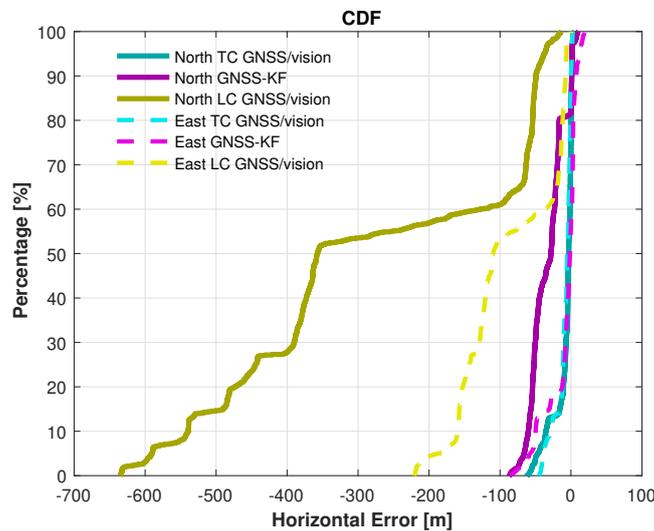


Figure 19. Cumulative distribution functions of the horizontal errors.

From Figure 19, considering the north direction, the TC GNSS/vision performs better than the GNSS-KF approach 82% of the time and 100% of the time when compared with the LC GNSS/vision. In the east direction, the proposed method performs better than the GNSS-KF 20% of the time and 99% of the time than the LC GNSS/vision. The performance of the proposed method over the LC GNSS/vision indicates the benefit of coupling the LOS satellites with the visual odometry. It can clearly be seen that when the GNSS PVT solution is not available (which happens 65.5% of the time

during the data collection), tight coupling of the small number of available GNSS observations is still able to improve the accuracy compared to using no GNSS observations (in the loose coupling case). The proposed method provides a significant improvement over the GNSS-only solution, demonstrating the benefit of fusing visual odometry with the GNSS solution.

Returning back to Figure 16, Zone C depicts a straight moving zone where both TC GNSS/vision and GNSS-KF provide a poor position estimation. This case, although not alarming for the proposed method, requires further investigation. The two reasons mentioned previously that cause poor integrated solutions occur at this location. Due to a traffic jam, the vehicle stopped under a bridge where LOS satellites were not observed, and the visual odometry was very poor, caused by the fact that very few feature points were detected (the very dark captured images are shown in Figure 20b).

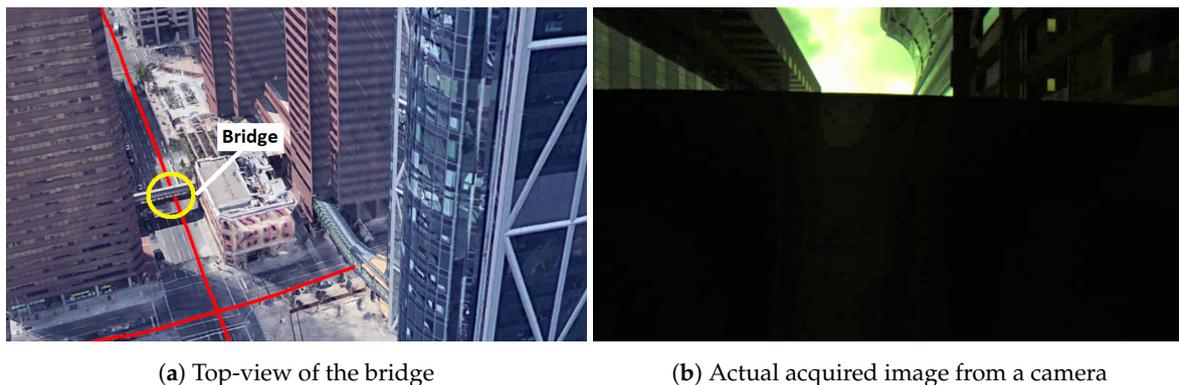
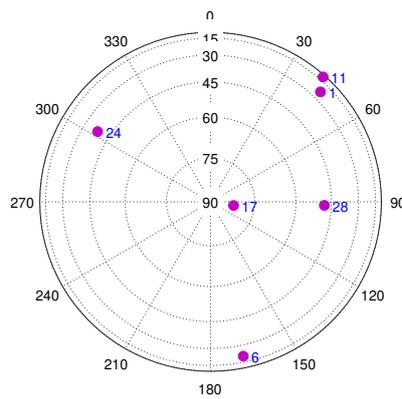


Figure 20. Bridge causing position errors.

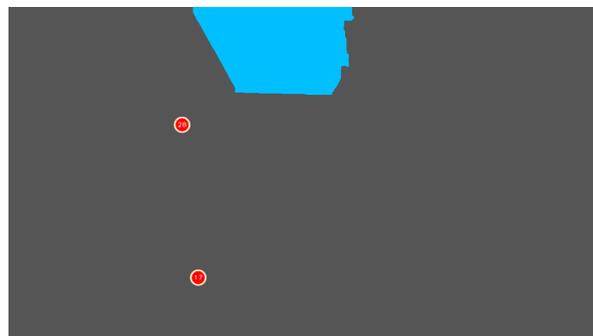
In this situation a GNSS/vision/INS integrated system, for example (or GNSS/vision/“appropriate sensor”), would be beneficial. The GPS constellation at this time is shown in Figure 21a. Due to the camera’s FOV, the high elevation satellites that appear to be obstructed by the bridge are in fact tracked, but are excluded from both solutions. This is shown in Figure 21b. The remaining satellites that fall out of the image are considered as obstructed.

From the number of the tracked and LOS satellites over time depicted in Figure 17, it can be seen that when the vehicle is stopped under the bridge, although a few satellites were still tracked, no LOS was observed (see Figures 22b and 17, Epochs 628 to 684). To see the performance in terms of the reliability, the residuals of the satellites are shown in Figure 22.

These represent the difference between the actual and the solved measurements and thus indicate the level of agreement of the observations. In open-sky (during the first 100 s), the innovation values are bounded between ± 5 m. In the challenging zone, they are larger. The end circle markers in Figure 22 indicate values in excess of ± 30 m. By applying statistical testing, satellites with larger innovation values, i.e., the outliers, are detected, and their measurements are prevented from corrupting the solution in the GNSS-KF case. For example, at Epoch 413179 (Figure 21), PRN 19 has a high elevation, but has very large innovation values (Figure 22); thus, its measurements are not used in the GNSS-only case. Even if not the case, our NLOS mitigation method had discarded this specific PRN since the projected satellite falls out of the image “scope” and is considered as obstructed. However, in some cases, satellites with small innovation values can be obstructed and thus degrade the final position solution. PRN17 is a good example of this. PRN17 has a very high elevation and small innovation values, most of the time. Its measurements are thus trusted and used in the GNSS-KF even if obstructed (“slightly” degrading the solution). In this case, the NLOS mitigation algorithm is able to determine this PRN as obstructed, and its measurements are not used, leading to an improved solution. Throughout the test, this occurs with other high satellites that are briefly obstructed.



(a) Sky plot



(b) Satellites reception condition

Figure 21. Sky plot of the GPS constellation at Epoch 413179 and its reception conditions.

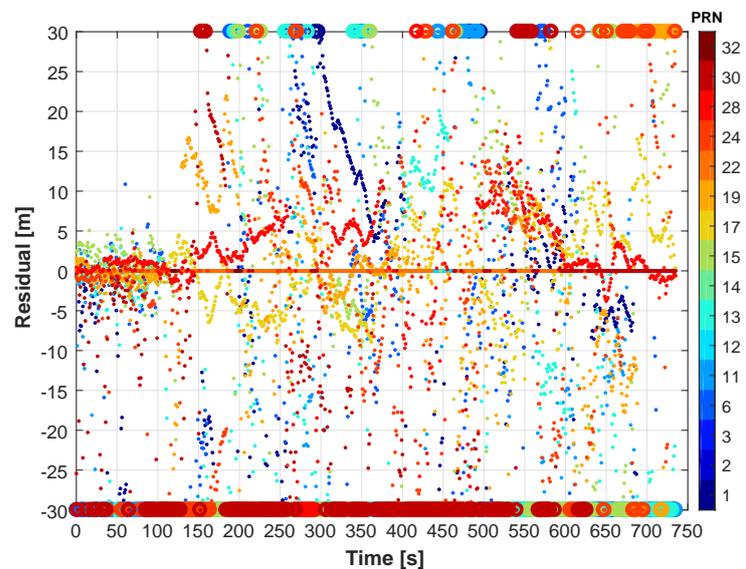


Figure 22. GNSS-only PR residuals.

Surprisingly, as shown in Figure 18, during the first 100 s, the GNSS-KF strategy outperforms the TC GNSS/vision approach in the east direction. The obvious reason behind this is the poor vision performance during this time span (that was close to an open-sky scenario: one side of the road

unobstructed, thus, less and bad feature point distribution). However, once the receiver's antenna is obstructed, the TC GNSS/vision approach becomes more accurate than GNSS-KF.

The statistics of the 2D position errors of the considered path are shown in Table 2.

Table 2. Statistics of the considered trajectory.

Estimator	2D rms Error (m)	2D Maximum Error (m)
GNSS-KF	39.8	113.2
LC GNSS/vision	56.3	402.7
TC GNSS/vision	14.5	61.1

These statistics are comparable or better than the results obtained in [29] for urban canyon tightly-coupled GNSS/vision (with similar equipment) where a stereo camera system is used instead. This is expected because NLOS satellites are not rejected in this work, which degrades their solution compared to the method presented in this paper.

7. Conclusions

This paper presents a tightly-coupled GNSS/vision integration approach. Vehicle motion was estimated by using the visual odometry technique implemented based on feature points detected in successive images acquired by a camera system mounted on the top of the car. An NLOS effect mitigation algorithm tailored for harsh environments and based on LOS satellite selection was also introduced. To improve the LOS satellite selection method, an appropriate image segmentation method using ASF filtering and the flood-fill approach was used to classify image contents into sky and non-sky. Finally, the selected LOS satellites, as well as the vision measurements were integrated in a tightly-coupled manner, taking into account the scale factor since a monocular vision system was used and these types of systems are able to determine the relative translation up to a scale factor only. Results show improvements of 82% and 20% compared to loose coupling of GNSS and visual odometry and to a GNSS-only Kalman filter navigation system, respectively, suggesting that both LOS satellites' selection and the visual odometry were beneficial. However, some portions of the travelled path were challenging, and the reasons why the algorithm could be influenced were discussed. One of the solutions to stops under bridges where our vision system setup could face challenges (as most of the systems using a camera in a dark environment) could be adding a sensor that is able to provide relative position such as INS or absolute position such as a 3D Building Model (3DBM). In future work, we will use a 3DBM to improve the proposed method. Moreover, instead of employing the commonly-used Kalman filtering technique, the Multiple Model Particle Filter (MMPF), which has proven to better characterize the position errors and estimate the multipath effects [66,67], will be investigated.

Acknowledgments: The first author would like to thank Mark Petovello for his guidance and suggestions during the early development of this work. The data used in this paper can be found in [68].

Author Contributions: P.V.G. conceived of and designed the algorithms and performed the related experiments. P.V.G. and K.O. analysed the data and wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

3DBM	3D Building Mode
6-DOF	Six Degrees of Freedom
ASF	Alternate Sequential Filter
CDF	Cumulative Distribution Function
DGPS	Differential Global Positioning System

ECEF	Earth-Centred, Earth-Fixed
EKF	Extended Kalman Filter
FOV	Field of View
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
GRD	Geodesic Reconstruction by Dilatation
IMU	Inertial Measurement Unit
KF	Kalman Filter
LOS	Line Of Sight
MMPL	Multiple Model Particle Filter
NLOS	Non-Line of Sight
PnP	Perspective-n-Point
PR	Pseudorange
RGB	Red, Green, Blue (colour space)
SLAM	Simultaneous Localization and Mapping
SVD	Singular Value Decomposition
UAV	Unmanned Aerial Vehicle
UWB	Ultra-Wideband
V2I	Vehicle to Infrastructure
VO	Visual Odometry

References

- Petovello, M. Real-Time Integration of a Tactical-Grade IMU and GPS for High-Accuracy Positioning and Navigation. Ph.D. Thesis, University of Calgary, Calgary, AB, Canada, 2003.
- Won, D.H.; Lee, E.; Heo, M.; Sung, S.; Lee, J.; Lee, Y.J. GNSS integration with vision-based navigation for low GNSS visibility conditions. *GPS Solut.* **2014**, *18*, 177–187, doi:10.1007/s10291-013-0318-8.
- Ben Afia, A.; Escher, A.C.; Macabiau, C. A Low-cost GNSS/IMU/Visual monoSLAM/WSS Integration Based on Federated Kalman Filtering for Navigation in Urban Environments. In *ION GNSS+ 2015, Proceedings of the 28th International Technical Meeting of The Satellite Division of the Institute of Navigation, Tampa, FL, USA, 14–18 September 2015*; ION, Institute of Navigation: Tampa, FL, USA, 2015; pp. 618–628.
- Gao, Y.; Liu, S.; Atia, M.M.; Noureldin, A. INS/GPS/LiDAR Integrated Navigation System for Urban and Indoor Environments Using Hybrid Scan Matching Algorithm. *Sensors* **2015**, *15*, 23286–23302, doi:10.3390/s150923286.
- Kim, S.B.; Bazin, J.C.; Lee, H.K.; Choi, K.H.; Park, S.Y. Ground vehicle navigation in harsh urban conditions by integrating inertial navigation system, global positioning system, odometer and vision data. *IET Radar Sonar Navig.* **2011**, *5*, 814–823, doi:10.1049/iet-rsn.2011.0100.
- Levinson, J.; Thrun, S. Robust vehicle localization in urban environments using probabilistic maps. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010*; pp. 4372–4378, doi:10.1109/ROBOT.2010.5509700.
- Braasch, M.S. Multipath Effects. In *Global Positioning System: Theory and Applications*; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 1996; Volume I, pp. 547–568.
- Brown, A.; Gerein, N. Test Results from a Digital P(Y) Code Beamsteering Receiver for Multipath Minimization. In *Proceedings of the ION 57th Annual Meeting, Albuquerque, NM, USA, 11–13 June 2001*; pp. 872–878.
- Keshvadi, M.H.; Broumandan, A. Analysis of GNSS Beamforming and Angle of Arrival estimation in Multipath Environments. In *Proceedings of the 2011 International Technical Meeting of The Institute of Navigation, San Diego, CA, USA, 24–26 January 2011*; pp. 427–435.
- Groves, P.D. Principles of GNSS, inertial, and multisensor integrated navigation systems, 2nd edition [Book review]. *IEEE Aerosp. Electron. Syst. Mag.* **2015**, *30*, 26–27, doi:10.1109/MAES.2014.14110.
- Groves, P.D.; Jiang, Z. Height Aiding, C/N0 Weighting and Consistency Checking for GNSS NLOS and Multipath Mitigation in Urban Areas. *J. Navig.* **2013**, *66*, 653–669, doi:10.1017/S0373463313000350.
- Attia, D.; Meurie, C.; Ruichek, Y.; Marais, J. Counting of Satellite with Direct GNSS Signals using Fisheye camera: A Comparison of Clustering Algorithms. In *Proceedings of the ITSC 2011, Washington, DC, USA, 5–7 October 2011*, doi:10.1109/ITSC.2011.6082955.

13. Suzuki, T.; Kubo, N. N-LOS GNSS Signal Detection Using Fish-Eye Camera for Vehicle Navigation in Urban Environments. In Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014), Tampa, FL, USA, 8–12 September 2014.
14. Gakne, P.V.; Petovello, M. Assessing Image Segmentation Algorithms for Sky Identification in GNSS. In Proceedings of the 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Calgary, AB, Canada, 13–16 October 2015; pp. 1–7, doi:10.1109/IPIN.2015.7346956.
15. Obst, M.; Bauer, S.; Wanielik, G. Urban multipath detection and mitigation with dynamic 3D maps for reliable land vehicle localization. In Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium, Myrtle Beach, SC, USA, 23–26 April 2012; pp. 685–691, doi:10.1109/PLANS.2012.6236944.
16. Peyraud, S.; Bétaille, D.; Renault, S.; Ortiz, M.; Mougél, F.; Meizel, D.; Peyret, F. About Non-Line-Of-Sight Satellite Detection and Exclusion in a 3D Map-Aided Localization Algorithm. *Sensors* **2013**, *13*, 829–847, doi:10.3390/s130100829.
17. Kumar, R.; Petovello, M.G. 3D building model-assisted snapshot positioning algorithm. *GPS Solut.* **2017**, *21*, 1923–1935, doi:10.1007/s10291-017-0661-2.
18. Ulrich, V.; Alois, B.; Herbert, L.; Christian, P.; Bernhard, W. Multi-Base RTK Positioning Using Virtual Reference Stations. In Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2000), Salt Lake City, UT, USA, 19–22 September 2000; pp. 123–131.
19. Mur-Artal, R.; Tardós, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262, doi:10.1109/TRO.2017.2705103.
20. Valiente, D.; Gil, A.; Reinoso, S.; Juliá, M.; Holloway, M. Improved Omnidirectional Odometry for a View-Based Mapping Approach. *Sensors* **2017**, *17*, 325, doi:10.3390/s17020325.
21. Valiente, D.; Gil, A.; Payá, L.; Sebastián, J.M.; Reinoso, S. Robust Visual Localization with Dynamic Uncertainty Management in Omnidirectional SLAM. *Appl. Sci.* **2017**, *7*, 1294, doi:10.3390/app7121294.
22. Gakne, P.V.; O’Keefe, K. Skyline-based Positioning in Urban Canyons Using a Narrow FOV Upward-Facing Camera. In Proceedings of the 30th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2017), Portland, OR, USA, 25–29 September 2017; pp. 2574–2586.
23. Petovello, M.; He, Z. Skyline positioning in urban areas using a low-cost infrared camera. In Proceedings of the 2016 European Navigation Conference (ENC), Helsinki, Finland, 30 May–2 June 2016; pp. 1–8, doi:10.1109/EURONAV.2016.7530554.
24. Alatise, M.B.; Hancke, G.P. Pose Estimation of a Mobile Robot Based on Fusion of IMU Data and Vision Data Using an Extended Kalman Filter. *Sensors* **2017**, *17*, 2164, doi:10.3390/s17102164.
25. Wang, J.; Gao, Y.; Li, Z.; Meng, X.; Hancock, C.M. A Tightly-Coupled GPS/INS/UWB Cooperative Positioning Sensors System Supported by V2I Communication. *Sensors* **2016**, *16*, 944, doi:10.3390/s16070944.
26. Soloviev, A.; Venable, D. Integration of GPS and Vision Measurements for Navigation in GPS Challenged Environments. In Proceedings of the IEEE/ION Position, Location and Navigation Symposium, Indian Wells, CA, USA, 4–6 May 2010; pp. 826–833, doi:10.1109/PLANS.2010.5507322.
27. Chu, T.; Guo, N.; Backén, S.; Akos, D. Monocular camera/IMU/GNSS integration for ground vehicle navigation in challenging GNSS environments. *Sensors* **2012**, *12*, 3162–3185, doi:10.3390/s120303162.
28. Vetrella, A.R.; Fasano, G.; Accardo, D.; Moccia, A. Differential GNSS and Vision-Based Tracking to Improve Navigation Performance in Cooperative Multi-UAV Systems. *Sensors* **2016**, *16*, 2164, doi:10.3390/s16122164.
29. Aumayer, B. Ultra-Tightly Coupled Vision/GNSS for Automotive Applications. Ph.D. Thesis, Department of Geomatics Engineering, The University of Calgary, Calgary, AB, Canada, 2016.
30. Meguro, J.I.; Murata, T.; Takiguchi, J.I.; Amano, Y.; Hashizume, T. GPS Multipath Mitigation for Urban Area Using Omnidirectional Infrared Camera. *IEEE Trans. Intell. Transp. Syst.* **2009**, *10*, 22–30, doi:10.1109/TITS.2008.2011688.
31. Marais, J.; Ambellouis, S.; Flancquart, A.; Lefebvre, S.; Meurie, C.; Ruichek, Y. Accurate Localisation Based on GNSS and Propagation Knowledge for Safe Applications in Guided Transport. *Procedia Soc. Behav. Sci.* **2012**, *48*, 796–805, doi:10.1016/j.sbspro.2012.06.1057.
32. Scaramuzza, D.; Fraundorfer, F. Visual Odometry [Tutorial]. *IEEE Robot. Autom. Mag.* **2011**, *18*, 80–92, doi:10.1109/MRA.2011.943233.
33. Deretey, E.; Ahmed, M.T.; Marshall, J.A.; Greenspan, M. Visual indoor positioning with a single camera using PnP. In Proceedings of the 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Banff, AB, Canada, 13–16 October 2015; pp. 1–9, doi:10.1109/IPIN.2015.7346756.

34. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 834–849.
35. Engel, J.; Stückler, J.; Cremers, D. Large-scale direct SLAM with stereo cameras. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 1935–1942, doi:10.1109/IROS.2015.7353631.
36. Caruso, D.; Engel, J.; Cremers, D. Large-scale direct SLAM for omnidirectional cameras. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 141–148, doi:10.1109/IROS.2015.7353366.
37. Misra, P.; Enge, P. *Global Positioning System: Signals, Measurements, and Performance*; Ganga-Jamuna Press: Lincoln, MA, USA, 2011.
38. Peng, X. Improving High Sensitivity GNSS Receiver Performance in Multipath Environments for Vehicular Applications. Ph.D. Thesis, University of Calgary, Calgary, AB, Canada, 2013.
39. He, Z.; Petovello, M.; Lachapelle, G. Indoor Doppler error characterization for High Sensitivity GNSS receivers. *IEEE Trans. Aerosp. Electron. Syst.* **2014**, *50*, doi:10.1109/TAES.2014.130111.
40. Xie, P.; Petovello, M.G. Improved Correlator Peak Selection for GNSS Receivers in Urban Canyons. *J. Navig.* **2015**, *68*, 869–886. doi:10.1017/S037346331500017X.
41. Ren, T.; Petovello, M. Collective bit synchronization for weak GNSS signals using multiple satellites. In Proceedings of the 2014 IEEE/ION Position, Location and Navigation Symposium—PLANS 2014, Monterey, CA, USA, 5–8 May 2014; pp. 547–555, doi:10.1109/PLANS.2014.6851416.
42. Aumayer, B.M.; Petovello, M.G.; Lachapelle, G. Development of a Tightly Coupled Vision/GNSS System. In Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014), Tampa, FL, USA, 8–12 September 2014; pp. 2202–2211.
43. Schreiber, M.; Königshof, H.; Hellmund, A.M.; Stiller, C. Vehicle localization with tightly coupled GNSS and visual odometry. In Proceedings of the 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, Sweden, 19–22 June 2016; pp. 858–863, doi:10.1109/IVS.2016.7535488.
44. Zhang, Z. A Flexible New Technique for Camera Calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334, doi:10.1109/34.888718.
45. Bradski, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 1 November 2000.
46. Bradski, G.; Kaehler, A. *Learning OpenCV: Computer Vision with the OpenCV Library*; O'Reilly: Cambridge, MA, USA, 2008.
47. Levenberg, K. A Method for the Solution of Certain Problems in Least Squares. *Q. Appl. Math.* **1944**, *2*, 164–168.
48. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An Efficient Alternative to SIFT or SURF. In Proceedings of the 2011 International Conference on Computer Vision (ICCV'11), Barcelona, Spain, 6–13 November 2011; IEEE Computer Society: Washington, DC, USA, 2011; pp. 2564–2571, doi:10.1109/ICCV.2011.6126544.
49. Rosten, E.; Drummond, T. Fusing Points and Lines for High Performance Tracking. In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05), Beijing, China, 17–21 October 2005; Volume 2, pp. 1508–1515, doi:10.1109/ICCV.2005.104.
50. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. BRIEF: Binary Robust Independent Elementary Features. In *European Conference on Computer Vision: Part IV*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 778–792.
51. Hamming, R. Error Detecting and Error Correcting Codes. *Bell Syst. Tech. J.* **1950**, *29*, 147–160.
52. Fischler, M.A.; Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* **1981**, *24*, 381–395, doi:10.1145/358669.358692.
53. Nister, D. An Efficient Solution to the Five-point Relative Pose Problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 756–770, doi:10.1109/TPAMI.2004.17.
54. Hartley, R.I.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2004; ISBN 0521540518.
55. Gakne, P.V.; O'Keefe, K. Monocular-based pose estimation using vanishing points for indoor image correction. In Proceedings of the 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sapporo, Japan, 18–21 September 2017; pp. 1–7, doi:10.1109/IPIN.2017.8115954.
56. Horn, A. Doubly Stochastic Matrices and the Diagonal of a Rotation Matrix. *Am. J. Math.* **1954**, *76*, 620–630.

57. Ruotsalainen, L. Vision-Aided Pedestrian Navigation for Challenging GNSS Environments, vol. 151. Ph.D. Thesis, Publications of the Finnish Geodetic Institute, Masala, Finland, November 2013.
58. Meguro, J.; Murata, T.; Takiguchi, J.; Amano, Y.; Hashizume, T. GPS Accuracy Improvement by Satellite Selection using Omnidirectional Infrared Camera. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 Septemebr 2008; pp. 1804–1810, doi:10.1109/IROS.2008.4650709.
59. Canny, J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 679–698, doi:10.1109/TPAMI.1986.4767851.
60. Soille, P. *Morphological Image Analysis: Principles and Applications*; Springer Science & Business Media: New York, NY, USA, 2013.
61. Meyer, F. Levelings, Image Simplification Filters for Segmentation. *J. Math. Imaging Vis.* **2004**, *20*, 59–72.
62. Datta, B.N. Linear state-space models and solutions of the state equations. In *Numerical Methods for Linear Control Systems*; Datta, B.N., Ed.; Academic Press: San Diego, CA, USA, 2004; Chapter 5, pp. 107–157, doi:10.1016/B978-012203590-6/50009-4.
63. Koch, K.R. *Parameter Estimation and Hypothesis Testing in Linear Models*; Springer Inc.: New York, NY, USA, 1988.
64. Petovello, M.G.; Zhe, H. Assessment of Skyline Variability for Positioning in Urban Canyons. In Proceedings of the ION 2015 Pacific PNT Meeting, Honolulu, HI, USA, 20–23 April 2015; pp. 1059–1068.
65. Noureldin, A.; Tashfeen, B.K.; Georgy, J. *Fundamentals of Inertial Navigation, Satellite-Based Positioning and Their Integration*; Springer: New York, NY, USA, 2013.
66. Toledo-Moreo, R.; Colodro-Conde, C.; Toledo-Moreo, J. A Multiple-Model Particle Filter Fusion Algorithm for GNSS/DR Slide Error Detection and Compensation. *Appl. Sci.* **2018**, *8*, 445.
67. Giremus, A.; Tourneret, J.Y.; Calmettes, V. A Particle Filtering Approach for Joint Detection/Estimation of Multipath Effects on GPS Measurements. *IEEE Trans. Signal Process.* **2007**, *55*, 1275–1285, doi:10.1109/TSP.2006.888895.
68. Gakne, P.V.; O’Keefe, K. Replication Data for: Tightly-Coupled GNSS/Vision using a Sky-pointing Camera for Vehicle Navigation in Urban Area. *Scholars Portal Dataverse.* **2018**, doi:10.5683/SP/CWJ2VH.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).