Improving Accuracy of GNSS Devices in Urban Canyons*

Boaz Ben-Moshe† Elazar Elkin‡ Harel Levi§ Ayal Weissman¶

Abstract

This paper addresses the problem of calculating the accurate position of a *GNSS* device operating in an urban canyon, where lines of sight (*LOS*) with navigation satellites are too few for accurate trilateration calculation. We introduce a post-processing refinement algorithm, which makes use of a *3D* map of the city buildings as well as captured signals from all traceable navigation satellites. This includes weak signals originating from satellites with no line of sight (*NLOS*) with the device. We also address the dual problem - computing a *3D* map of the city buildings when the position of the device is given. This is achieved by storing *LOS/NLOS* rays to all navigation satellites sampled at multiple locations within a region of interest (*ROI*). These rays are then used to compute the *3D* shapes of buildings in the *ROI*.

A series of field experiments confirm that both algorithms are applicative. The position refinement algorithm significantly improves the device's accuracy and the mapping algorithm allows few users to map a complex urban region simply by walking through it.

1 Introduction

Receivers in Global Navigation Satellite Systems (*GNSS*) such as *GPS*, *GLONASS* or *GALILEO* tend to output inaccurate location estimations while operating in urban regions, mostly due to the density of tall buildings, which often block a receiver's line of sight (*LOS*) to the navigation satellites. Modern *GNSS* receivers are sensitive enough to receive the indirect signal reflected from the buildings. This *multi-path* effect is the major factor of poor performance of *GNSS* in urban canyons. A *GNSS* receiver approximates its position by interpolating the signal from each navigation satellite into a *pseudorange*, an approximation of the distance between the receiver and the navigation satellite, obtained by multiplying the speed of light by the time needed for the signal to travel the distance. Using four *pseudoranges* and their associated satellite locations, the *GNSS* receiver location can be computed simply by intersecting the four spheres (see [2, 4] for more information regarding *GNSS* principles). Figure 5 presents an actual positioning error caused by wrong *pseudoranges* in an urban region.

In all but rare cases, a rule of thumb, which correlates a strong signal to the existence of *LOS* is proven very effective. Therefore, a *GNSS* device operating in a non-urban area would simply sort captured signals according to their strength, then use four (or more) strong enough signals to compute its location. Since a receiver wandering around at the country-side typically has *LOS* with more than four satellites for most of its journey, the decisive majority of location computations in such areas are typically based on signals originating from *LOS* satellites, for which *pseudoranges* tend to be accurate (the error range is typically within 2-5 meters).

In urban canyons, however, the situation is fundamentally different. It is very common for a *GNSS* device operating in an area of this sort (e.g. downtown Manhattan) to be surrounded by obstacles such as tall buildings, which block *LOS* with most, and infrequently all, otherwise available satellites. Since at least four strong-enough signals, equivalent to four *LOS* satellites, are required for accurate positioning, the outcome of a narrowly available sky is inevitably a skewed computation, up to the point where the device is unable to perform its task.

Prior attempts to address limited *LOS* in urban areas, all of the while succeed to present reasonably-accurate results where satellites' signals are scarce and weak, are mostly based on approaches such as Map Matching (*MM*) [7, 10, 17, 18] and Dead Reckoning (*DR*) [4, 13]. A certain degree of improvement could arguably be obtained by assuming the *GNSS* receiver is located inside a car, which drives at some estimated speed on top of a road with a known path. The fact of the matter, however, is that most of these methods are evidently not sufficient in rough urban canyons, where lines of sight (*LOS*) can and do deteriorate up to the point where a receiver only captures multipath indirect reflections (zero *LOS*). In such circumstances, the decisive majority of *GNSS* devices become incompetent and cannot improve

*This research was partially supported by the MAGNET program of the Israel Ministry of Industry and Trade - RESCUE consortium (patent pending 61/426,541).

†Department of Computer Science, Ariel University Center, Ariel 40700, Israel. benmo@g.ariel.ac.il

‡Department of Computer Science, Ariel University Center, Ariel 40700, Israel

§Department of Computer Science, Ariel University Center, Ariel 40700, Israel

¶Department of Computer Science, Bar-Ilan University, Ramat-Gan, 52900 Israel

accuracy using these methods.

The novelty of the $GNSS-refinement$ method presented in this paper is based on two core concepts. The first is that unlike existing methods, which mostly rely on information external to the line of sight (LOS) problem, such as vehicle speed and road location, the discussed improvement is confronting the LOS problem in a more direct manner, by applying LOS-based algorithms. The second inventive aspect is an effective leverage of supposedly-useless weak signals originating from no line of sight ($NLOS$) satellites. By combining captured signals' strength with shading algorithms on a region of interest's *3D* map, our improved *GNSS* device is able to determine with a high degree of assurance in which parts of the region of interest (ROI) it could potentially be, and likewise, in which parts of the ROI it is certain not to be, thus significantly narrowing the problem's error range.

The above, however, merely segments a ROI into "can-be" and "cannot-be" partial regions. Therefore, to address the general case, in which intersecting the satellites' binary LOS maps yields more than one "can-be" region, we multiply each binary map with a "likelihood weight". These weights are from a continuous range, where each derives from the captured signal's strength of the respective satellite. We later discuss how summing weighted LOS maps for all satellites usually converges to a single "highest likelihood" location. We also explain the heuristics we use in case there are still several candidate locations subsequent to that summing procedure.

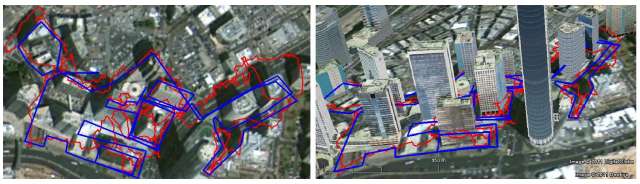

Figure 1: The Urban Canyon effect: In red, the *GPS* captured path. In blue, the actual path.

1.1 Related Work

Prior studies have shown that longer integration times and data wipe-off enable High Sensitivity *GPS* (*HS-GPS*) receivers to acquire and track signals at lower signal strengths [14, 9]. This increases satellite availability in weak signal environments, but in an urban canyon comes with a baggage of positioning errors resulting from signal cross-correlation, multipath and echo-only signals [14, 8]. Most attempts to improve *GNSS* devices' accuracy in urban canyons consider the typical in-vehicle situation. This narrows the estimation problem, since vehicles are generally restricted to travel on roads. Nevertheless, *GNSS* and other absolute positioning systems do not inherently locate vehicles onto roads [12, 15]. The process of coinciding the output of a sensor such as *GPS* with a road network map is called Map Matching (MM) and is often integrated with Dead Reckoning (DR), which is the process of estimating one's current position based upon a previously determined position [3, 19].

Unfortunately, the problem's narrowing achieved by MM techniques is not sufficient in complex urban canyons. This is mainly because limited LOS in such areas frequently causes initial location estimates that are off by tens of meters. Such deviations are too large for MM techniques, which often leads to placements onto wrong distant roads.

In this paper we demonstrate how *3D* maps of an area can be leveraged to acquire more information out of captured (both LOS and $NLOS$) *GNSS* signals. This additional information can then be used in conjunction with existing MM techniques, or as an alternative to such methods. Moreover, the concepts employed to narrow the estimation problem also form the basis of our algorithm for the dual *3D* modeling problem (see section 3).

1.2 Paper Structure

Following an introduction and a related work review, we turn to a detailed discussion of the *GNSS* refinement algorithm. We then present a framework algorithm to the dual (inverse) problem, namely the computing of the city buildings' *3D* maps by capturing the signal strength of all navigation satellites. Subsequently, we put the presented algorithms to the test and discuss results from field experiments conducted in rough urban canyons. We conclude with suggested future work.

2 GNSS Refinement Algorithm

2.1 Overview and Definitions

In this section we present the main algorithm for improving the *GNSS* receiver's accuracy in urban canyons. The algorithm transforms the signal strength of each traceable navigation satellite into a $LOS/NLOS$ value. This value is not boolean but a continuous value in the range of $[0, 1]$, representing the LOS clearance.

The algorithm's detailed description begins with formalization of (i) input parameters available from the *GNSS* device; (ii) some pre-defined constants (thresholds) and (iii) functions and data structures used throughout the refinement process (see Table 1). We then describe the mechanism by which the algorithm determines LOS status with each satellite. Subsequently, we explain the concepts behind $LOS/NLOS$ partial

maps and discuss the formation of an aggregated likelihood map out of them. We conclude the section with a high-level pseudo-code, which encapsulates the entire algorithm.

GNSS Algorithm Definitions	
Device	*DeviceOutput*: The current non-refined location estimation, position error range, and the set ($S(t)$) of all traceable satellite signals.
	LastPosition: The last recorded refined location, and the corresponding error ratio and confidence level.
Constants	*SigBench*: A signal-strength threshold, which determines visibility (assume *LOS* if higher; *NLOS* if lower).
	MaxSig: A surely visible signal-strength threshold (used for linear transformation to [0,1] range).
	MinorErr: An error estimation threshold (don't correct if smaller).
Functions	$S(t) = S_1 ... S_n$: The set of all satellite signals as captured by the *GNSS* receiver in time t.
	$S_v(t) \subset S(t)$: The set of visible satellites (subset of $S(t)$ obtained using *SigBench*).
	$S_u(t) \subset S(t)$: The set of invisible satellites (subset of $S(t)$ obtained using *SigBench*).
	Map: A 2.5D representation of the terrain - including both earth surface and buildings on top of it.
	ROI: The region of interest; a minimal polygon which contains both: (i) all buildings which may affect the user.(ii) all possible locations in which the user could potentially be.
	$LOS(S_i, ROI)$: A binary shading function from each point of the *ROI* to a $LOS, NLOS$ w.r.t. S_i location.
	$SIG(S_i, ROI, W_i)$: A refinement of LOS(Si, ROI) to a continuous range using a signal strength ($W_i \in [0, 1]$) weight.
	Approximate: The proposed *GNSS* refinement algorithm (i) Creates an aggregated likelihood map from all $SIG(Si, ROI, W_i)$ (ii) Picks most likely location with respect to the parameters *DeviceOutput* and *LastPosition*.
	Aggregated: An aggregation ad-hoc *2D* matrix with *ROI*'s boundaries.

Table 1: Formal definitions of parameters used throughout the proposed *GNSS* refinement algorithm.

2.2 Approximating Satellites' LOS Status

The strength of a signal as captured by *GNSS*-receiver depends on several factors ([5]):

- Global parameters: transmission frequency, transmission power - these parameters are mostly fixed.
- Position and time: atmosphere and ionosphere condition, the angle between the satellite and the receiver.
- *LOS* and multi-path status: the nature of propagate signal with respect to the possible "radio path" to the receiver [16, 5, 6].

For a given *GNSS* (e.g. GPS L1, L2), the global parameters are fixed. The position and time parameters can be approximated within a small error range, usually smaller then 5 dB. The typical *LOS* signal strength is at least 10dB stronger than the signal strength of a reflected signal (*NLOS*). It is therefore rather simple to classify captured signals into $S_v(t)(LOS)$ and $S_u(t)(NLOS)$ subsets. Moreover, the field experiments we conducted (see section 4) demonstrate that determining a signal's *LOS/NLOS* status is applicable even in highly complex urban regions.

2.3 Partial LOS Map

Computing a shading map of a city building map (*ROI*) w.r.t. a satellite position can be done by projecting the buildings on the surface (the satellite position can be thought as in infinity). Our implementation encapsulates the *LOS* map of each captured signal S_i in $S(t)$ as a *2D* matrix filled with (0 and 1) binary values, each indicating whether the receiver is likely to have *LOS* with the corresponding satellite within a one square meter spot. The computation of the map's values is a relatively straightforward shading algorithm, which makes use of the satellite's position and the *2.5D Map* of the area.

A fundamental, somewhat tricky, feature of the proposed algorithm concerns the leverage of weak multipath signals captured by the receiver to obtain meaningful information. A weak captured signal almost always indicates the absence of *LOS* with the respective satellite and is therefore classified (using *SigBench*) into the subset of invisible satellites $S_u(t)$. Knowing the receiver cannot see the satellite from its current location, we can conclude it is certainly not positioned in spots where this satellite can be seen. Derives from this observation is the applicability of using the complementary *LOS* maps (switching 1 and 0 values) of invisible satellites belonging to the subset $S_u(t)$ as likelihood layers, which are as informative as likelihood layers generated from visible satellite signals belonging to the $S_v(t)$ subset. Furthermore, since the discussed algorithm is targeted at urban canyons scenarios where *LOS* may deteriorate even to an empty $S_v(t)$ subset (all captured signals are weak), the complementary *LOS* maps of invisible satellites belonging to $S_u(t)$ become an essential source of information for the formation of the aggregated likelihood map discussed below.

2.4 Likelihood Weights

Likelihood weightening is a mechanism employed by the algorithm to improve determinism. In the absence of weights, which transform a *LOS* map from a binary *LOS/NLOS* representation into a more refined likelihood map, the algorithm could very well narrow the

problem by segmenting the *ROI* into "can be" and "cannot be" regions, but would not be able to resolve a scenario of multiple "can be" spots, and pick a single location to be presented on the *GNSS* device's screen. By introducing signal-strength derived weights, the *LOS* maps become differentiated from one another (each map is multiplied by a weight from a continuous range). This, in turn, results in (i) a further segmented aggregated likelihood map, which serves the algorithm's purpose of picking a single spot ; (ii) improved accuracy and reliability, since greater importance is granted to stronger signals.

2.5 Aggregated Likelihood Map

The likelihood map is implemented as a *2D* matrix filled with real numbers, each representing the likelihood of the receiver to be located within a location (e.g., within 1 square meter). The matrix is constructed by (a) multiplying each partial *LOS* matrix with a weight, which signifies the signal's strength of the corresponding satellite and (b) summing all partial weighted matrices. The outcome of this matrix addition is a single *2D* matrix, where each point represents a likelihood, and the matrix's highest value(s) is the most likely spot. Since this likelihood matrix is being constructed by summing a considerable number (roughly 8 to 18) of partial (already differentiated by weights) matrices, the max value of the matrix tends to have a relatively small number of appearances. In the empirically common case of a single unique max value, the algorithm concludes and the spot represented by that max value is being presented. Otherwise, if several max values are encountered, the algorithm's *Approximate* function is employing heuristic methods (e.g. nearest point to the non-intervened receiver's output - *DeviceOutput*) to choose the point to be presented.

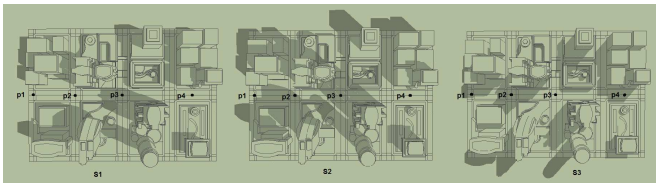

Figure 2: Construction of a likelihood map from *LOS* maps. In this example there are three satellites ($S_1 - S_3$) and four points ($p_1 - p_4$), p_1 and p_4 have the same aggregated shading maps (they both can see S_1, and S_2). Yet p_2 and p_3 aggregated shading maps differ from the aggregated shading map of p_1 (or p_4).

2.6 Algorithm Formalization

Using the definitions at the beginning of this section (see Table 1), the refinement algorithm's high-level pseudo-code would be:

Algorithm 1: High-Level pseudo-code for *GNSS* refinement algorithm

Result: $RefinedPosition$
if $PositionError < MinorErr$ **then**
  return $DevicePosition$;
Let *S(t)* be the set of all satellite signals as captured by the *GNSS* receiver in time *t*.
Let *Aggregated* be a *2D* matrix with *ROI*'s boundaries (initialized with 0 values).
for *each* S_i *in* $S(t)$ **do**
  Use *SigBench* to determine whether $S_i \in S_v$ or $S_i \in S_u$.
  Let W_i be S_i's signal strength weight [1]
  $L_i \leftarrow SIG(S_i, ROI, W_i)$ [2]
  if $S_i \in S_u$ **then**
    Inverse L_i [3]
  $Aggregated \leftarrow Aggregated + L_i$ [4]
Let *LikelyPoints* be the set of max values in *Aggregated*.
Refined Position = Nearest point in *LikelyPoints* to *DevicePosition*.
return *Refined Position*.

[1] W_i is transformed into $[0, 1]$ range using $MaxSig$ constant.
[2] Compute partial weighted map.
[3] For an *NLOS* satellite : $x = (1 - x)$ to each x in the matrix.
[4] Add partial weighted map to aggregated likelihood map.

3 3D Mapping Algorithm

3.1 Overview

In this section we address the problem of constructing a *3D* building map using the strength of the signals from navigation satellites. This task is the dual problem to the improved accuracy: instead of using the *3D* buildings' map to improve the receiver's position, we use its position to approximate the buildings' *3D* map.

Most *GNSS* devices are able to keep detailed log files, which contain information about captured satellites' signals along a device's journey. Thereafter, it is a straightforward process to track down a device's path and the satellites' position respective to that device at each point in time during the journey (in sampling rate of 1-10Hz). This ability to track down signals, when magnified by a number of *GNSS* devices covering a subjected urban canyon, is a preliminary enabler of our novel framework algorithm to the dual problem, which is the generation of buildings' *3D* models out of captured satellites signals.

The *3D* mapping problem can be defined as follows: given a *GNSS* log-file, which contains samples of: time,

position, accuracy and the signal strength to each traceable satellite, convert the log file into two sets of *3D* vectors: (i) Blue vectors: all *LOS* signals. (ii) Red vectors: all the *NLOS* signals. The goal is to construct the surface, which will block all the vectors in the red set and will not block the vectors from the blue set.

3.2 2.5D Mapping Heuristics

We limit this algorithm to compute a *2.5D* map representation, a surface of a terrain representation in which each (X, Y) location has a single Z value associated with it. For simplicity, we divide the mapping algorithm into two sequential steps: (i) computing the buildings' contours. (ii) approximating the height of each contour.

For the algorithm's first step we traverse time-consecutive samples of the satellite signals. We then compute distances between consecutive samples, using a weighted *xor* function (assuming *LOS* is 1 and *NLOS* is 0), a change in satellite status (*LOS/NLOS*) contributes to the distance function according to the satellite angle (high-angle satellites contribute more). If the distance is above some threshold, we consider this point to be an *edge-point*. *Edge-points* tend to appear in buildings' corners and next to buildings' walls (due to the amplifying distance of high-angle satellites). We then filter out *edge-points* with potential high position error, and aggregate all the relatively accurate *edge-points*. Lastly, we compute contours which are bounded by the *edge-points*. These contours may be general polygons or some constraint shapes (see Figure 6).

The algorithm's second step computes the z-value of each contour. The height value of each contour is bounded by all the *LOS* rays going over it. Yet because the *LOS/NLOS* data is "noisy" by nature, the actual z-value is computed as the height for which maximal weighted-constraints (*LOS/NLOS*) are satisfied. As in the first stage, the higher the ray-angle is (*LOS/NLOS*) the larger its weight becomes. Noteworthy is that samples from the same spot taken at different times of the day are beneficial for the algorithm. This is because for each triplet (spot, building, satellite) if there's a (time dependent) *LOS* ray from the spot to the satellite, which goes over the building's contour, then there exists a time of the day, which minimizes the vertical distance between that ray and the building's roof.

During our initial field experiments, the contours of the builds were slightly larger and shorter than in reality. In order to fix this effect we modified the algorithm to keep refining the *2.5D* map by updating the contours according to the building approximated height.

4 Experimental Results

We conducted a set of preliminary experiments to evaluate the suggested algorithms in practice. In order to evaluate the improved accuracy algorithm two types of experiments were conducted (see Figures 3-5): (i) locating a *GPS* receiver in a fixed position for few hours (at each position). (ii) walking along a fixed route. In both experiments the actual position was compared to the suggested location of both (a) the *GPS* device (b) the refinement algorithm. In order to evaluate the *2.5D* mapping algorithm we have walked through the university campus and computed the approximated building map. The algorithm was implemented in Java and tested on both Android and Linux. The algorithm was able to refine the location (in an area of 200*200 meters) in less than 1 second, which validated its applicability to run efficiently on mobile devices (equipped with a 1Hz *GPS*). The following *GPS* receivers were used: *Fastrax 1Hz*, *Wintec G-Rays 10Hz* and the internal *GPS* of the Android devices. All tests were made while walking. The *GPS* row data was accessed via the *NMEA* protocol.

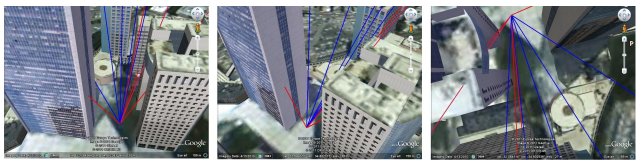

Figure 3: 3D mapping algorithm in action. In red are the rays which are blocked (*NLOS*), in blue are the rays with *LOS* to the corresponding navigation satellites.

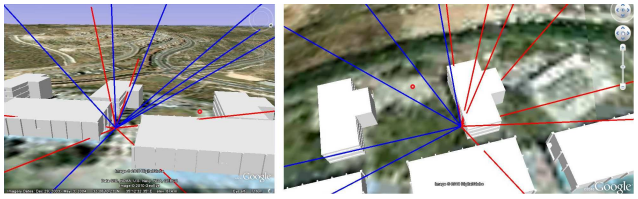

Figure 4: Improved position algorithm in action: In both above examples the *GPS* suggested a position with 12-18 meters error ratio. The refinement algorithm was able to fix the position to an error of less than 1 meter.

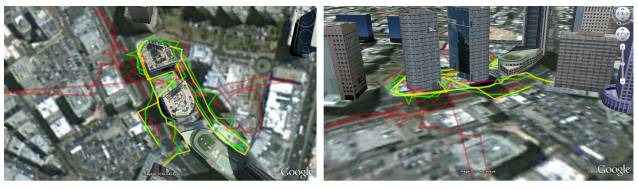

Figure 5: Improved Position: Left: above view. Right: *3D* perspective view. In red, the path computed by a 1Hz *fastrax GPS* with an average error of 31 meters and max error of 180 meters; In green, the refined position with an average error of 4 meters and max error of 11 meters; In yellow, the actual path.

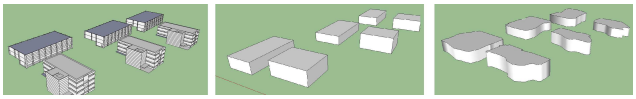

Figure 6: *2.5D* mapping example, Left: the actual buildings, Middle: the *2.5D* map of the buildings using axis parallel rectangles contours. Right: the *2.5D* map of the buildings with no contour constraints

5 Conclusion and Future Work

We have proposed a new framework for improving a GNSS-receiver accuracy in urban regions. We have also presented an algorithm for constructing a *2.5D* building map - using the receiver's position and the *LOS* status to it from each navigation satellite. In practical implementations, both algorithms could be running together - improved position accuracy assists in improving the $2.5D$ map accuracy, which in turn further improves positioning accuracy. Such approach for Simultaneous Localization And Mapping ($SLAM$) is proven to be an efficient method in many navigation and mapping tasks [1, 11]. The experimental results presented above show that even a basic implementation of the algorithms helps improving the *GNSS* accuracy significantly in urban canyons. Using few walking clients equipped with standard *GPS* devices we were able to construct a $2.5D$ buildings map of a complex downtown area; this map was then sufficient as the input source for the accuracy improvement algorithm. For future work we intend to use *GNSS-pseudoranges* to compute a more accurate map of the buildings. In particular, we would like to generalize the $2.5D$ mapping algorithm into a real *3D* mapping method.

Acknowledgment The authors wish to thank Eliyahu Ariel and Prof. Avner Kidar for introducing us to the real world of *GPS*.

References

[1] J. Artieda, J. M. Sebastián, P. Campoy, J. F. Correa, I. F. Mondragón, C. Martínez, and M. Olivares. Visual 3-d slam from uavs. *Journal of Intelligent and Robotic Systems*, 55(4-5):299–321, 2009.

[2] S. Gleason and D. Gebre-egiabher. *GNSS Applications and Methods*. 2009.

[3] J. S. Greenfeld. Matching GPS observations to locations on a digital map. In *Proceedings of the 81th Annual Meeting of the Transportation Research Board, Washington D. C.* 2002.

[4] P. D. Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. 2008.

[5] B. M. Hannah. Modelling and simulation of gps multipath propagation. 2001.

[6] Y.-W. Lee, Y.-C. Suh, and R. Shibasaki. A simulation system for gnss multipath mitigation using spatial statistical methods. *Comput. Geosci.*, 34:1597–1609, November 2008.

[7] S. Liu, Z. Shi, M. Zhao, W. Xu, and K. Zhang. An urban map matching algorithm using rough sensor data. *Power Electronics and Intelligent Transportation System, Workshop on*, 0:266–271, 2008.

[8] G. D. Macgougan. High sensitivity GPS performance analysis in degraded signal environments. *M. Sc. Thesis, UCGE Report No*, page 20176, 2003.

[9] B. Peterson, D. Bruckner, and S. Heye. *Measuring GPS Signals Indoors, Proceedings of ION GPS-1997, The Institute of Navigation, 16-19 September, Kansas City, Missouri, USA*. pp 389-398, 1997.

[10] M. A. Quddus, W. Y. Ochieng, and R. B. Noland. Integrity of map-matching algorithms. *Transportation Research Part C: Emerging Technologies*, 14(4):283 – 302, 2006.

[11] D. Schleicher, L. M. Bergasa, M. Ocaña, R. Barea, and E. L. Guillén. Real-time hierarchical gps aided visual slam on urban environments. In *EUROCAST*, pages 326–333, 2009.

[12] C. Scott. *Improved GPS Positioning for Motor Vehicles Through Map Matching, Proceedings of ION GPS-1994, The Institute of Navigation, 20-23 September, Salt Lake City, Utah, USA*. pp 1391-1400, 1994.

[13] J. Stephen and J. Stephen. Development of a multisensor gnss based vehicle navigation system, 2000.

[14] S. Syed. University of calgary development of map aided gps algorithms for vehicle navigation in urban canyons, 2005.

[15] G. Taylor and G. Blewitt. *Virtual Differential GPS and Reduction Filtering by Map Matching, Proceedings of ION GPS-1999, The Institute of Navigation, 20-23 September, Salt Lake City, Utah, USA*. pp 114-120, 1999.

[16] N. Viandier, D. F. Nahimana, J. Marais, and E. Duflos. Gnss performance enhancement in urban environment based on pseudo-range error model. In *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 377–382, 2008.

[17] C. E. White, D. Bernstein, and A. L. Kornhauser. Some map matching algorithms for personal navigation assistants. *Transportation Research Part C: Emerging Technologies*, 8(1-6):91 – 108, 2000.

[18] Y. Zhang and Y. Gao. A fuzzy logic map matching algorithm. *Fuzzy Systems and Knowledge Discovery, Fourth International Conference on*, 3:132–136, 2008.

[19] L. Zhao, W. Y. Ochieng, M. A. Quddus, Noland, and R. B. An Extended Kalman Filter algorithm for Integrating GPS and low-cost Dead reckoning system data for vehicle performance and emissions monitoring. *The Journal of Navigation*, 56:257–275, 2003.