

Research Article

Simulation of GNSS Availability in Urban Environments Using a Panoramic Image Dataset

Sakpod Tongleamnak¹ and Masahiko Nagai²

¹Remote Sensing and GIS, School of Engineering and Technology, Asian Institute of Technology, 58 Moo 9, Paholyothin Highway, Klong Luang, Pathumthani 12120, Thailand

²Graduate School of Sciences and Technology for Innovation, Yamaguchi University, 2-16-1 Tokiwadai, Ube, Yamaguchi 755-8611, Japan

Correspondence should be addressed to Sakpod Tongleamnak; st113073@ait.ac.th

Received 4 August 2016; Revised 7 December 2016; Accepted 28 December 2016; Published 26 January 2017

Academic Editor: David Akopian

Copyright © 2017 Sakpod Tongleamnak and Masahiko Nagai. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Performance of Global Navigation Satellite System (GNSS) positioning in urban environments is hindered by poor satellite availability because there are many man-made and natural objects in urban environments that obstruct satellite signals. To evaluate the availability of GNSS in cities, this paper presents a software simulation of GNSS availability in urban areas using a panoramic image dataset from Google Street View. Photogrammetric image processing techniques are applied to reconstruct fisheye sky view images and detect signal obstacles. Two comparisons of the results from the simulation and real world observation in Bangkok and Tokyo are also presented and discussed for accuracy assessment.

1. Introduction

Currently, the Global Navigation Satellite System (GNSS) plays a crucial role in providing positioning services for various applications such as navigation and Location Based Services (LBS) [1]. The demand for these services has grown continuously and with the introduction of new constellations such as BeiDou and Galileo, the number of GNSS satellites has also increased continuously. It is known that GNSS positioning availability and accuracy depend largely on the number and geometry of satellites which are visible to a GNSS receiver. Several tools such as Trimble Planner [2] and GNSS-Radar [3] have been created to predict the availability and geometry of GNSS satellites at a given location and time.

However, evaluation of the ground environment which affects the visibility of GNSS satellites is very difficult especially in urban areas where various features, both man-made and natural, are present. The visibility of satellites is significantly reduced in urban environments because signals from the satellites are blocked by tall buildings, elevated structures such as overpasses, and elevated railways. This phenomenon is called the urban canyon effect. Tools that do not consider the effect of urban canyons cannot accu-

rately predict availability and geometry of GNSS satellites. Currently, evaluations of the urban canyon effect are usually conducted directly by measurement with a receiver, which is time-consuming and requires a site visit, and this might not be feasible. Another method is using 3D GIS or 3D city models to simulate the availability of GNSS [4–6]. However, such data is not widely available and costly to create. Google Earth could be used to solve the availability and the costs associated with 3D city models [7] but the quality of models is not guaranteed and they are currently only available for major cities. A third approach would be to use a camera and image processing techniques to capture geometries of urban canyons [8–14]. This approach requires a camera and in kinematic applications an Inertial Measurement Unit (IMU). It also introduces the disadvantage of sensitivity to light and weather conditions during operation. Thus this paper proposes an alternative method using panoramic image datasets such as Google Street View as an input to evaluate urban canyon geometry.

Google Street View is widely available and covers most of the world's urban areas [15]. It has been used for many applications such as tourism and education. In scientific research, Google Street View has been used for environmental studies

[16] to obtain the geometry of urban canyons. This study created a simulation that calculates satellite geometry and then compares it with urban canyon geometry from Google Street View. The result of the simulation is the number of visible satellites and the value of DOP (Dilution of Precision) in a given location and time. The simulation accuracy is evaluated based on two actual survey experiments conducted in central Bangkok and Tokyo by a vehicle equipped with a GNSS receiver.

2. Methods

The simulation starts by acquiring multiple images from Google Street View at a given location and combining them to create one fisheye sky view image. The created fisheye image would cover a full view of the upper hemisphere at that location. The geometry of urban canyons can be determined by separating sky from nonsky areas in the sky view image using image processing techniques. The result is a binary sky mask image used to determine satellite visibility. A satellite's position in the sky is calculated using an orbital model and its orbital parameters. Finally, the satellite is considered visible if its position in the sky mask image is in the sky area; otherwise, it is considered not visible.

2.1. Google Street View and Sky View Generating. Google Street View is a technology to capture 360-degree spherical images at street level. Since its launch in 2007, Google has collected Street View images for about 3,000 cities in 63 countries [17] and the coverage has expanded regularly. By using Google Street View, our system can simulate GNSS satellite availability at any location where Street View is available.

Google does not allow direct access to Street View panoramas but users can request Street View images through the Application Program Interface (API). To acquire an image, the user must pass the following parameters to Google's API: location, heading, pitch, horizontal field of view (HFOV), and image resolution. Heading indicates the compass heading of the camera relative to north. Pitch indicates the up or down angle of the camera relative to the Street View vehicle, which most of the time is flat horizontal. The API will return a rectilinear image according to the parameters including the metadata such as latitude, longitude, and month-year of which the image was taken.

To limit the distortion in the rectilinear image, all of the rectilinear images from Google Street View have HFOV of 90°. In order to have enough images to cover the upper hemisphere, a total of 9 images are requested per location. Figure 1 shows an example of requested Street View images at one location with their camera direction parameters. Each image has a 640 × 640 pixels resolution.

For this study, rectilinear images from Google Street View are assumed to be free of lens distortion and pitch angles are relative to flat horizontal. To create a fisheye sky view image, every pixel in the rectilinear images is remapped to a fisheye image from a virtual camera pointing up and north at the top. Figure 2 gives representations of rectilinear projection and equidistant fisheye projection.

In Figure 2(a), there are three coordinate systems: world coordinates east, north, uP (*ENU*), camera coordinates (*xyz*), and image coordinates (*uv*). Image coordinates have an origin at top left of the image with *u*- and *v*-axis parallel to *x*- and *y*-axis of camera coordinates. *z*-axis of camera coordinates is a principal axis passing through the center of the image. The movement of a Street View camera is restricted to be rotational only; therefore the origin of camera coordinates and world coordinates is at the same location (*O*). For a point *P* on the rectilinear image, camera coordinates can be derived from image coordinates by

$$P_{\text{camera}} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{w}{2} - u \\ \frac{h}{2} - v \\ f \end{bmatrix}, \quad (1)$$

where *w* and *h* are width and height of the image and *f* is the focal length which can be determined by

$$f = \frac{w}{2 \tan(\rho/2)}, \quad (2)$$

where ρ is horizontal field of view of the image. The coordinates of *P* in world coordinates are

$$P_{\text{world}} = \begin{bmatrix} E \\ N \\ U \end{bmatrix} = R \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad (3)$$

where *R* is a 3 × 3 rotation matrix computed from heading and pitch of the camera. After world coordinates of *P* are calculated, it can be remapped on to the fisheye image (Figure 2(b)). The mapping function of equidistant fisheye projection is given as [18]

$$r = f\theta, \quad (4)$$

where *r* is the radial distance of the projected point from the center of the projection (*O*), *f* is the equidistant parameter of the virtual fisheye camera, and θ is the incident angle of the ray from the projected point to the optical axis. Since the optical axis of the virtual fisheye camera is the same as *U* axis of world coordinates. The incident angle θ of *P* is

$$\theta = \tan^{-1} \frac{\sqrt{N^2 + E^2}}{U}. \quad (5)$$

Finally, the azimuth or angular distance (\emptyset) of \overrightarrow{OP} to *N* axis on *NE* plane can be determined by

$$\emptyset = \tan^{-1} \frac{E}{N}. \quad (6)$$

After \emptyset and θ being known, (4) can be used to remap a pixel at *P* on the rectilinear image to the fisheye sky view image.

Each of the rectilinear images from Google Street View is remapped to an equidistant fisheye image using the above method. Figure 3(a) shows the result of remapping



FIGURE 1: Images from Google Street View with the camera direction.

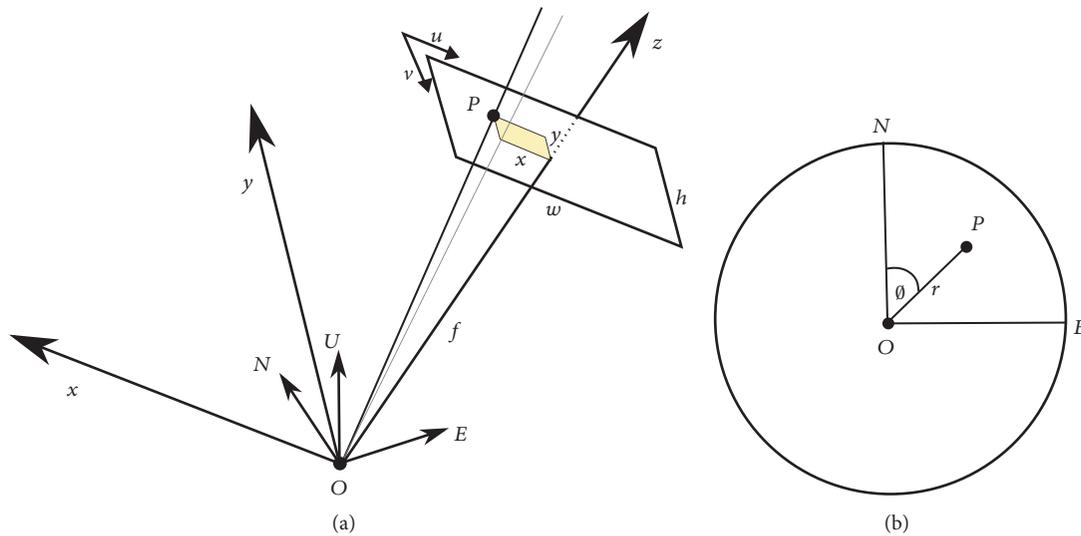


FIGURE 2: (a) Rectilinear projection. (b) Equidistant fisheye projection.

with images (2) to (9) moved outward from the center of projection to display individual images. When remapping images, there are some areas where images are overlapping. A multiresolution spline [19, 20] is used to seamlessly blend them together. The basic principle is to define the size of transition zone proportionally to the spatial frequency of the overlapping area. Low spatial frequency objects such as sky or cloud should have a wide transition zone to blend

together, while high spatial frequency objects such as trees or buildings should have a narrow transition zone. Figure 3(b) shows the sky view image after blending all 9 images together. The overlay shows the scale of azimuth and elevation angle in the sky view image.

Both remapping and blending are implemented in a set of open source software called Panorama Tools [21] which is used in this paper.

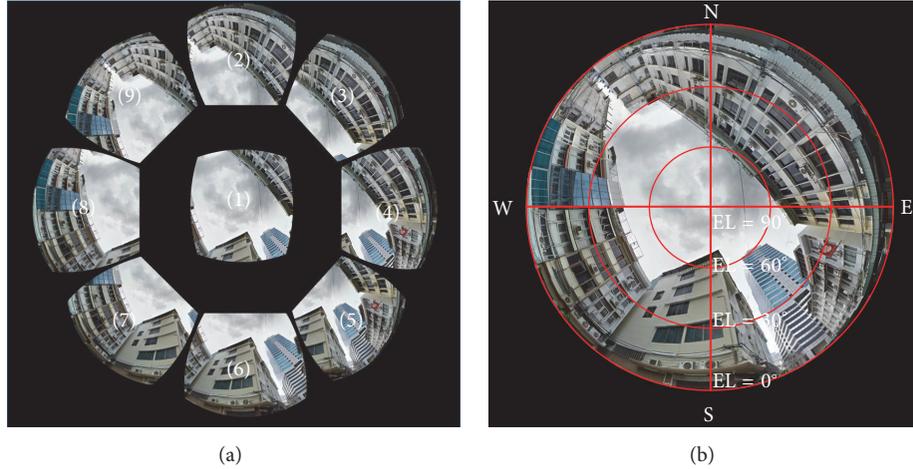


FIGURE 3: (a) Remapped Google Street View images. (b) Sky view image.

2.2. Automatic Sky Segmentation. In order to determine whether a satellite is visible from the image location, the sky view image needs to be segmented into two groups: sky areas and nonsky areas. The sky usually has more brightness than other objects in the image, and a simple image processing algorithm is developed to separate the sky from other objects in the image. The first step is to convert an image to grayscale then apply automatic thresholding based on the Otsu algorithm [22]. The result is a binary image which separates the high brightness areas and low brightness areas of the image. However, the binary image usually has a lot of noise, and in order to reduce it, morphological closing and a median filter are applied to the image. The results of each step are shown in Figure 4. The final binary image will be used as a sky mask image where the white regions are considered as sky areas and black regions are considered as nonsky areas.

To assess the accuracy of the sky segmentation algorithm, 25 sky view images from Google Street View were randomly selected and the sky areas were manually segmented, after which the result from manual segmentations were compared to the results from the automatic sky segmentation algorithm. The comparisons showed that on average 91.33% of the pixels in the sky view image were identified as the same class by both manual and automatic segmentation. The detailed comparisons are shown in Appendix.

2.3. Orbital Model. GNSS satellites broadcast Almanac data and a receiver can use an orbital model for approximate calculation of satellite positions from this data. Each GNSS system has its own Almanac and orbital model which employ different calculation processes and orbital parameter. In order to simplify the process of estimating the satellites position, this simulation used the Simplified General Perturbations (SGP) models instead. The SGP model was first developed in the 1960s and has been used by organizations such as NORAD and NASA. The theory and mathematical model of the SGP4, current implementation of SGP, are described in [23]. The SGP model can use orbital data in Two-Line Elements (TLE) format which is the format provided by USAF Space Command. All objects that orbit the earth including GNSS

satellites are monitored and tracked by the US government and their TLE data is routinely published [24].

TLE and SGP4 use the “true equator, mean equinox” (TEME) coordinate system. The output satellite position needs to be transformed before use. A method to transform TEME coordinates to Earth Center Earth Fixed (ECEF) coordinates is described in [25]. Firstly, TEME coordinates are rotated to Pseudo Earth Fixed (PEF) coordinates using Greenwich Mean Sidereal Time (GMST) then earth polar motion is applied to PEF coordinates to transform it to ECEF coordinates. The satellite ECEF coordinates can be transformed into local *ENU* coordinates by

$$\begin{bmatrix} E \\ N \\ U \end{bmatrix} = \begin{bmatrix} -\sin \lambda & \cos \lambda & 0 \\ -\cos \lambda \sin \varphi & -\sin \lambda \sin \varphi & \cos \varphi \\ \cos \lambda \cos \varphi & \sin \lambda \cos \varphi & \sin \varphi \end{bmatrix} \begin{bmatrix} x_{\text{ECEF}} \\ y_{\text{ECEF}} \\ z_{\text{ECEF}} \end{bmatrix}, \quad (7)$$

where λ and φ are geodetic coordinates. After *ENU* coordinates of the satellites are calculated, (5) and (6) can be used to determine the azimuth (θ) and elevation ($90^\circ - \theta$) angle.

The accuracy of the SGP4 model is comparable to the GPS Almanac as shown in a paper by Kelso [26]. They estimated error of SGP4 and GPS Almanac by comparing their results with the high accuracy GPS ephemerides. Within the period of 15 days from the epoch, the maximum orbital errors from both methods were in the comparable level of tens kilometers. Since GNSS satellites are relatively far away from the earth ($>20,000$ km), this level of accuracy is enough to predict satellites’ elevation and azimuth angles with less than 1-degree error.

2.4. Satellite Visibility. Once a satellite’s azimuth and elevation are calculated, its line of sight visibility can be predicted by using (4) to find the satellite’s position in the sky mask image from the sky segmentation. The satellite is considered

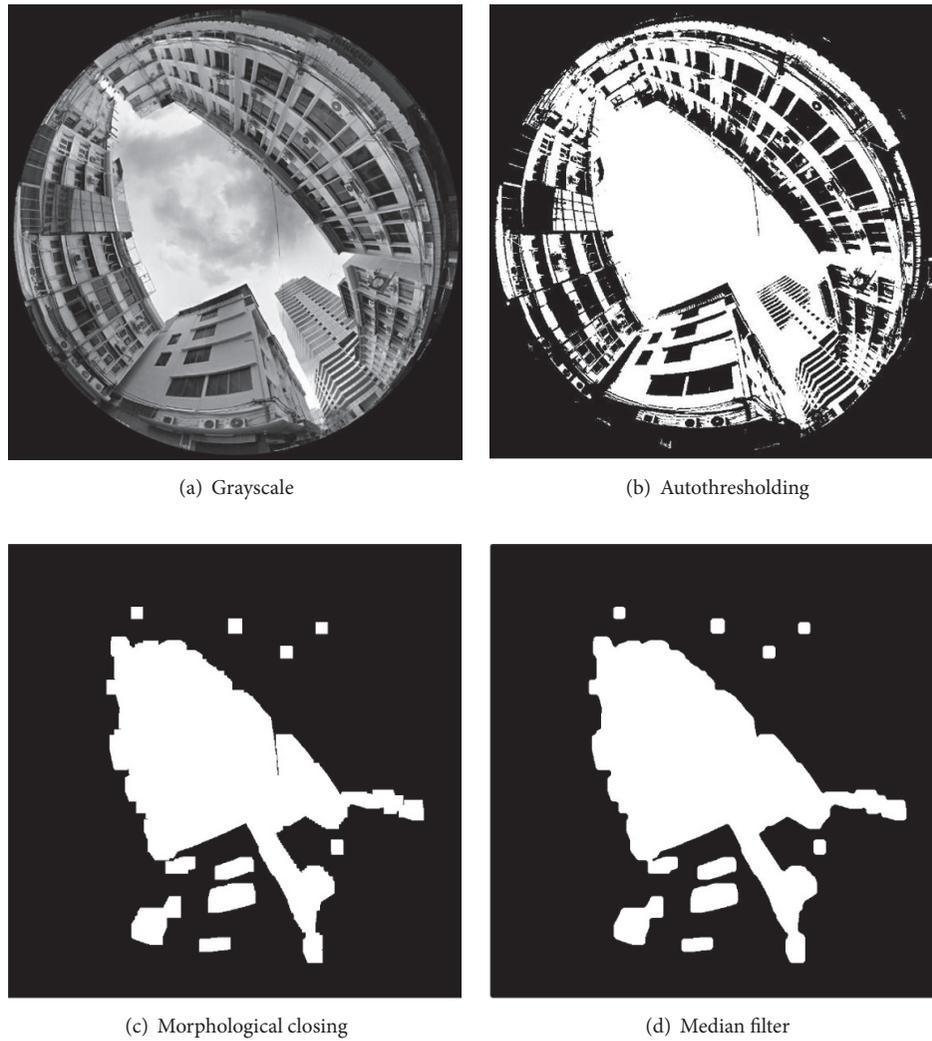


FIGURE 4: Automatic sky segmentation.

visible when the value of the sky mask image at its position is white (sky) and not visible when the value of the sky mask image at its position is black (nonsky).

3. Comparison of Simulation Result with Observation Data

To validate the simulation, two experiments were conducted to compare the simulated output with real world observation data. The first experiment in Bangkok used a van as a mobile platform equipped with a Javad Sigma GNSS receiver and a D-Link DCS-6010L fisheye camera. The camera was calibrated using omnidirectional camera calibration technique [27] and the images from the camera were adjusted to be equidistant fisheye images. Figure 5 shows the Javad GrAnt antenna and the D-Link camera on van's roof. The van traveled around the central Bangkok area to collect GNSS signals and sky view video. Figure 6 shows comparison between the collected data and the simulation output at approximately $13.721837^{\circ}\text{N}$, $100.529752^{\circ}\text{E}$. In the observation

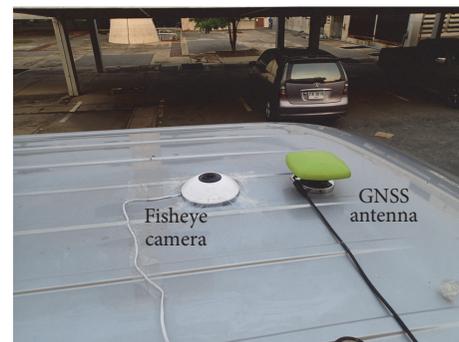


FIGURE 5: The van equipped with Javad antenna and D-Link camera.

image (Figure 6(a)), the sky view images from the fisheye camera were rotated according to van's direction and horizontally flipped to keep the north direction on the top and east direction on the right of the images. The Javad Sigma receiver can track GPS and GLONASS satellites. Positions of

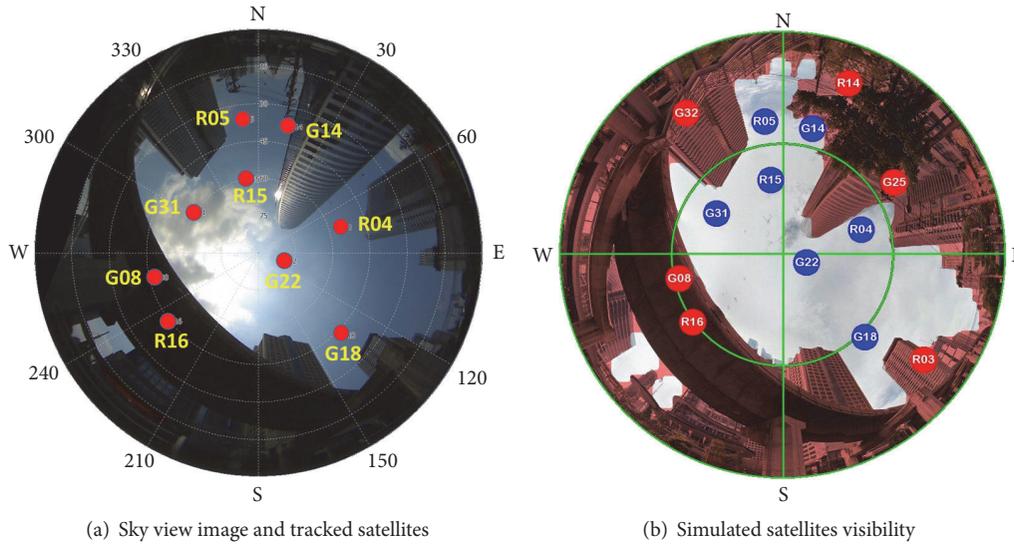


FIGURE 6: Bangkok observation and simulation results.

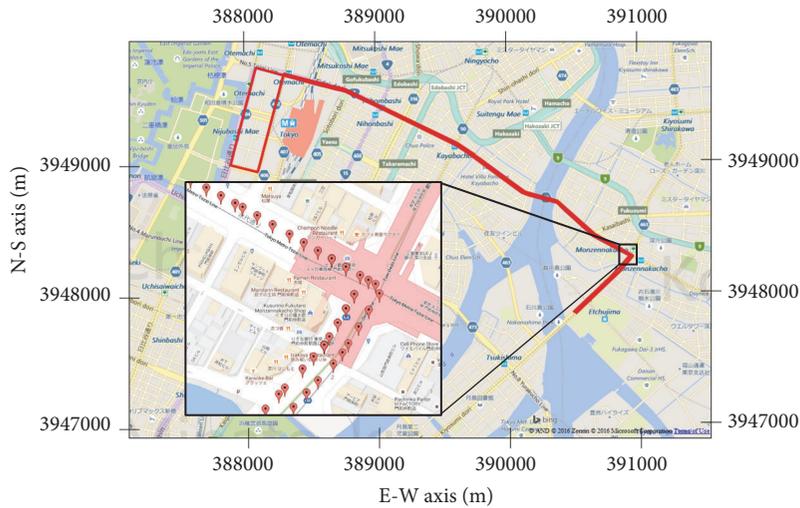


FIGURE 7: Study area in central Tokyo.

the satellites for which signals were tracked by the receiver are overlaid on the image. It can be seen that the positions of GPS PRN 8 and GLONASS 16 were behind a sky train track but the receiver could still track their signal. This is likely due to a multipath signal that refracted from the track or was reflected from a nearby building. In the simulated output (Figure 6(b)), the sky view image from Google Street View is shown with red tint applied to the nonsky area. Since Google collects Street View data discretely, a Street View point closest to van's location was used to generate the sky view image. Satellites in nonsky areas are displayed in red and considered as not visible at that location. The visible satellites in sky area are displayed in blue. Calculated DOP are 3.0 m and 3.8 m for observation and simulation, respectively.

The second experiment was conducted in central Tokyo on the road from Tokyo University of Marine Science and

Technology (TUMSAT) to the Tokyo train station and back as shown in Figure 7. This area is a dense urban area with a lot of tall buildings, several overpasses, and bridges. The total length is about 9 kilometers. The close-up inset on the map shows the locations of Google Street View panoramas; the distance between adjacent Street View locations is approximately 10 m.

The data was collected by a car equipped with a GNSS receiver, Trimble Net R9, and a Trimble Zephyr antenna at 1 Hz in total of 3404 epochs. The receiver mask angle was set to 15° and it could receive signals from the following GNSS constellations: GPS, QZSS, and BeiDou. Since the simulation only considers a satellite visible when it has direct LOS to the receiver, multipath signals need to be mitigated from the observed data. In order to avoid multipath problems like that in Bangkok experiment, an algorithm developed by Kubo et

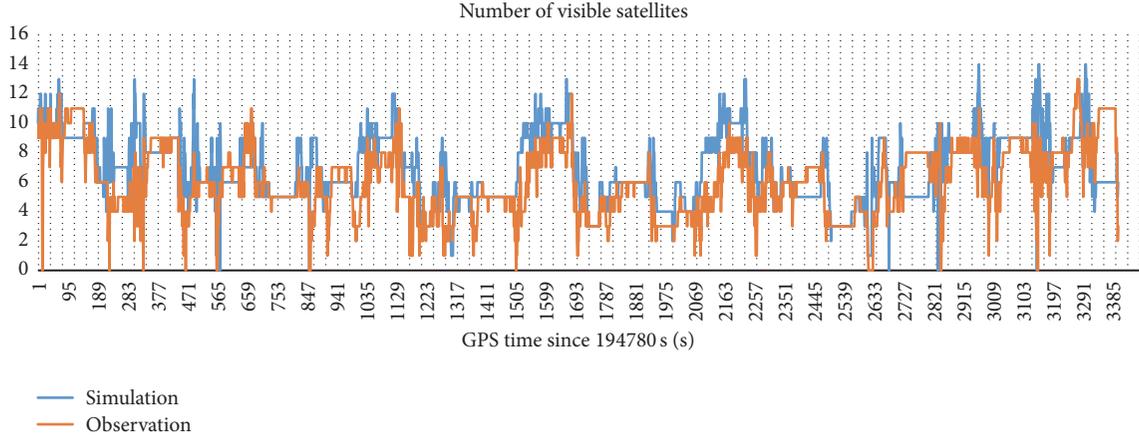


FIGURE 8: Observed and simulated number of visible satellites.

TABLE 1: Accuracy assessment detail by satellites.

Satellite name	Number of predictions and observations			
	Tracked	Observation		Tracked
		Not tracked	Not tracked	
	Visible	Simulation		Not visible
Not visible		Visible		
GPS PRN G02	101	298	164	56
GPS PRN G04	1870	984	310	240
GPS PRN G10	469	2177	709	49
GPS PRN G13	744	2147	378	135
GPS PRN G17	3081	57	62	204
GPS PRN G20	911	1918	344	231
GPS PRN G23	1662	1031	455	256
GPS PRN G28	769	1405	300	77
QZSS PRN J01	2872	130	116	286
BEIDOU PRN C01	379	1645	1239	141
BEIDOU PRN C03	500	2353	501	50
BEIDOU PRN C04	575	1873	675	281
BEIDOU PRN C08	2936	35	361	72
BEIDOU PRN C09	547	2563	251	43
BEIDOU PRN C12	149	1513	286	57
Total	17565 (38.16%)	20129 (43.74%)	6151 (13.37%)	2178 (4.73%)

al. [28] to mitigate multipath signals based on the received signal strength was applied to the data before comparison with the simulation result.

As in the previous experiment, the location of the car in each epoch was used to find the nearest Google Street View panorama to generate sky view images. The total number of sky view images generated from Google Street View was 980. This number is significantly lower than the number of epochs because when the car moved slowly or stopped, the closest Street View location would not change. In addition, in the area where Street View panoramas were only available in one lane such as in the top part of the close-up map, the same sky view images were used twice in both of the passages. The

average distance between receiver positions and the closest Street View locations was 4.12 m.

Throughout the data collection period, the receiver was able to receive signals from 8 GPS satellites, 1 QZSS satellite, and 6 BeiDou satellites. The graph in Figure 8 shows the numbers of visible satellites at each epoch from observed data and the predicted number from the simulation. In terms of total number of visible satellites, a decent agreement can be seen between the two sets of data in the graph with a root-mean-square deviation (RMSD) of 2.32 satellites.

$$\text{RMSD} = \sqrt{\frac{\sum_{t=1}^n (\text{NVIS}_{t,\text{sim}} - \text{NVIS}_{t,\text{obs}})^2}{n}}, \quad (8)$$

TABLE 2

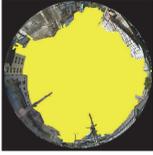
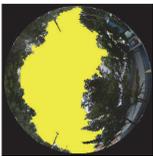
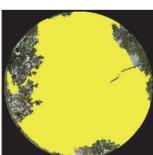
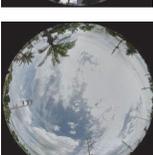
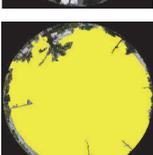
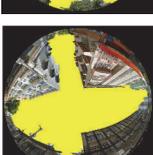
Number	Sky view image	Manual segmentation	Automatic segmentation	Number of correct classification pixels	Number of wrong classification pixels
1				333915 (75.58%)	107873 (24.42%)
2				406325 (91.97%)	35463 (8.03%)
3				370355 (83.83%)	71433 (16.17%)
4				409936 (92.79%)	31852 (7.21%)
5				425131 (96.23%)	16657 (3.77%)
6				408077 (92.37%)	33711 (7.63%)
7				425657 (96.35%)	16131 (3.65%)
8				333225 (75.43%)	108563 (24.57%)
9				389741 (88.22%)	52047 (11.78%)

TABLE 2: Continued.

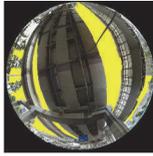
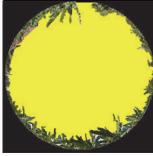
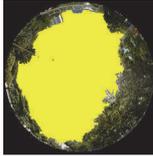
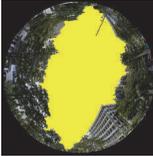
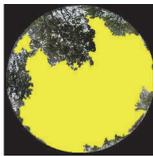
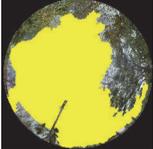
Number	Sky view image	Manual segmentation	Automatic segmentation	Number of correct classification pixels	Number of wrong classification pixels
10				424565 (96.10%)	17223 (3.90%)
11				408450 (92.45%)	33338 (7.55%)
12				433459 (98.11%)	8329 (1.89%)
13				388129 (87.85%)	53659 (12.15%)
14				421752 (95.46%)	20036 (4.54%)
15				401950 (90.98%)	39838 (9.02%)
16				424003 (95.97%)	17785 (4.03%)
17				394398 (89.27%)	47390 (10.73%)
18				391929 (88.71%)	49859 (11.29%)

TABLE 2: Continued.

Number	Sky view image	Manual segmentation	Automatic segmentation	Number of correct classification pixels	Number of wrong classification pixels
19				426232 (96.48%)	15556 (3.52%)
20				407263 (92.19%)	34525 (7.81%)
21				416390 (94.25%)	25398 (5.75%)
22				423570 (95.88%)	18218 (4.12%)
23				408617 (92.49%)	33171 (7.51%)
24				390650 (88.42%)	51138 (11.58%)
25				423489 (95.86%)	18299 (4.14%)
Average				91.33%	8.67%

where n is the number of epochs, $NVIS_{t,sim}$ is the number of predicted visible satellites at epoch t , and $NVIS_{t,obs}$ is the number of observed visible satellites at epoch t .

A detailed comparison of simulation and observation data for each satellite is shown in Table 1. There are 4 possible cases: the first case is that the simulation correctly predicted that the satellite was visible and the receiver could track the satellite signal (2nd column); the second case is that the simulation correctly predicted that the satellite was not visible and the receiver could not track the satellite signal (3rd

column); the third case is that the receiver could not track the satellite's signal but the simulation incorrectly predicted that the satellite was visible (4th column); and the last case is that the receiver could track the satellite's signal but the simulation incorrectly predicted that the satellite was not visible (5th column). For example, in the case of QZSS, out of the total 3404 epochs, there were 2872 epochs for which the simulation correctly predicted that it was visible; 130 epochs for which the simulation correctly predicted that it was not visible; 116 epochs for which the simulator predicted it was visible but the

receiver could not track its signal; and 286 epochs for which the simulator predicted that it was not visible but the receiver could track its signal. An interesting example is that of GPS G02, which only had a total number of observations and predictions of 619 epochs. This number is significantly fewer than that of other satellites because when the observation started, its elevation angle was below the mask angle. Only the last 619 epochs when its elevation angle was above the mask angle are included in the data.

Further investigation reveals two main sources of errors: the automatic sky segmentation and the distance between Street View's location and the receiver's location. The automatic sky segmentation has difficulty in separating bright and glass window buildings from the sky. It also tends to exaggerate the size of long thin obstacles such as power lines. As for distances, the nearer a Street View's location was, the fewer the errors there were since the urban canyon geometry would be similar to the receiver's; in other words, errors increased with distance.

To sum up, the simulation correctly predicted (both visible and not visible) 81.90% of satellite visibilities for these tests. These results indicate that the simulation can use Google Street View to predict GNSS satellites visibility for nearby locations with good accuracy.

4. Conclusion and Future Works

This paper described a simulation of GNSS satellite visibility in urban areas by using widely available Google Street View instead of a 3D city model as an input. The simulation started from generating sky view images from multiple Street View images and then separating the sky area and nonsky areas using digital image processing techniques to create a binary sky mask image. In terms of area, the automatic sky extraction can separate sky and nonsky areas with relatively high accuracy compared to manual segmentation. GNSS satellite positions in the sky can be calculated using the SGP4 orbital model and TLE data. The visibility of a satellite can be determined by testing its position with the sky mask image.

Two experiments were conducted to validate the simulation with real world observation data. One was in central Bangkok and the second one in central Tokyo. Both areas were very challenging environments for GNSS because of dense tall buildings. The Bangkok experiment showed that the simulation could work and the model was feasible. The Tokyo experiment mitigated multipath problems in observed data and showed in detail that the simulation output and real world observation could be comparable.

The simulation can be used for prediction of visibilities and DOPs of GNSS satellites where Google Street View is available without the need for prior visits to a site, which might be useful for planning the operations. Since the simulation can be automated, a large area simulation is possible. This in turn could be useful for studying the effect of a new GNSS satellite or a constellation. The simulation can also be used to improve the accuracy of positioning results by providing satellite visibility data to the positioning process. This data can also be used to remove predicted NLOS satellites from positioning calculations to improve the positioning accuracy.

The effect of this method on the accuracy of positioning results will be a future work.

Appendix

Detailed Comparison of Manual and Automatic Sky Segmentation Results

Table 2 shows the comparison between manual sky segmentation and automatic sky segmentation. Every pixel in the sky view images was segmented into two groups: sky and nonsky. A pixel was counted as a correct classification when it was classified as the same group by both manual and automatic sky segmentation; otherwise, it was counted as a wrong classification. There were 441788 pixels in a sky view image. The sky area in manual segmented images is displayed in yellow.

Competing Interests

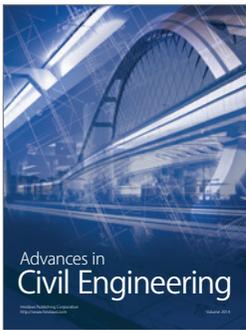
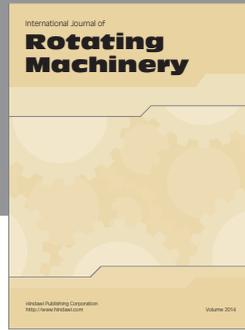
The authors declare that there are no competing interests regarding the publication of this paper.

References

- [1] "The European Commission and The European GNSS Agency," GNSS Market Report 4, 2015.
- [2] "Trimble GNSS Planning," 2016, <http://www.trimble.com/gnss-planningonline/>.
- [3] "GNSS-Radar," 2016, <http://www.taroz.net/GNSS-Radar.html>.
- [4] Y. Lee, Y. Suh, and R. Shibasaki, "Ajax GIS application for GNSS availability simulation," *KSCE Journal of Civil Engineering*, vol. 11, no. 6, pp. 303–310, 2007.
- [5] J. Bradbury, "Prediction of urban GNSS availability and signal degradation using virtual reality city models," in *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS '07)*, pp. 2696–2706, Fort Worth, Tex, USA, September 2007.
- [6] L. Wang, P. D. Groves, and M. K. Ziebart, "GNSS shadow matching: improving urban positioning accuracy using a 3D city model with optimized visibility prediction scoring," in *Proceedings of the 25th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS '12)*, pp. 423–437, September 2012.
- [7] T. Suzuki and N. Kubo, *Simulation of GNSS Satellite Availability in Urban Environments Using Google Earth*, 2012.
- [8] T. Suzuki, M. Kitamura, Y. Amano, and T. Hashizume, "Multipath mitigation using omnidirectional infrared camera for tightly coupled GPS/INS integration in urban environments," in *Proceedings of the 24th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS '11)*, pp. 2914–2922, Portland, Ore, USA, September 2011.
- [9] T. Suzuki and N. Kubo, "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 '14)*, vol. 3, pp. 1897–1906, September 2014.
- [10] J. S. Sánchez, A. Gehrmann, P. Thevenon, P. Brocard, A. B. Afia, and O. Julien, "Use of a FishEye camera for GNSS NLOS exclusion and characterization in urban environments," in

Proceedings of the Institute of Navigation International Technical Meeting (ITM '16), vol. 1, 2016.

- [11] J. Marais, C. Meurie, D. Attia, Y. Ruichek, and A. Flancquart, "Toward accurate localization in guided transport: combining GNSS data and imaging information," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 188–197, 2014.
- [12] D. Attia, C. Meurie, Y. Ruichek, and J. Marais, "Counting of satellites with direct GNSS signals using Fisheye camera: a comparison of clustering algorithms," in *Proceedings of the 14th IEEE International Intelligent Transportation Systems Conference (ITSC '11)*, pp. 7–12, October 2011.
- [13] J.-I. Meguro, T. Murata, J.-I. Takiguchi, Y. Amano, and T. Hashizume, "GPS multipath mitigation for urban area using omnidirectional infrared camera," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 22–30, 2009.
- [14] J.-Y. Han and T.-H. Juan, "Image-based approach for satellite visibility analysis in critical environments," *Acta Geodaetica et Geophysica*, vol. 51, no. 1, pp. 113–123, 2016.
- [15] "Explore Street View," 2016, <https://www.google.com/maps/streetview/explore/>.
- [16] R. Carrasco-Hernandez, A. R. D. Smedley, and A. R. Webb, "Using urban canyon geometries obtained from Google Street View for atmospheric studies: potential applications in the calculation of street level total shortwave irradiances," *Energy and Buildings*, vol. 86, pp. 340–348, 2015.
- [17] "Google Launches Street View in Dubai," 2014, <http://blogs.wsj.com/digits/2014/12/09/google-launches-street-view-in-dubai/>.
- [18] K. Miyamoto, "Fish eye lens," *Journal of the Optical Society of America*, vol. 54, no. 8, pp. 1060–1061, 1964.
- [19] P. J. Burt and E. H. Adelson, "A multiresolution spline with application to image mosaics," *ACM Transactions on Graphics*, vol. 2, no. 4, pp. 217–236, 1983.
- [20] P. J. Burt and E. H. Adelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, 1983.
- [21] Panorama Tools, 2016, <http://www.panoramatools.com/>.
- [22] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetic*, vol. 9, no. 1, pp. 62–66, 1979.
- [23] D. Vallado and P. Crawford, "SGP4 orbit determination," in *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Honolulu, Hawaii, USA, 2008.
- [24] SPACE-TRACK.ORG, 2016, <https://www.space-track.org/>.
- [25] N. Panigrahi and R. Gaurav, "Tracking satellite footprints on Earth's surface," *Coordinates*, vol. 11, no. 2, pp. 23–28, 2015.
- [26] T. S. Kelso, "Validation of SGP4 and IS-GPS-200D against GPS precision ephemerides," in *Proceedings of the 17th Annual Space Flight Mechanics Meeting*, pp. 427–440, Sedona, Ariz, USA, February 2007.
- [27] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A flexible technique for accurate omnidirectional camera calibration and structure from motion," in *Proceedings of the Fourth IEEE International Conference on Computer Vision Systems (ICVS '06)*, New York, NY, USA, January 2006.
- [28] N. Kubo, T. Suzuki, A. Yasuda, and R. Shibazaki, "An effective method for multipath mitigation under severe multipath environments," in *Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS '05)*, pp. 2187–2194, Long Beach, Calif, USA, September 2005.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

