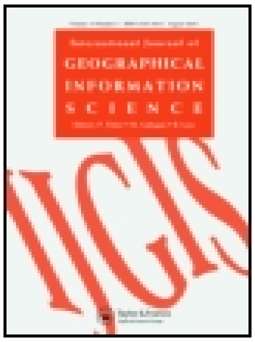


International Journal of Geographical Information Science

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tgis20

Exploring the vertical dimension of street view image based on deep learning: a case study on lowest floor elevation estimation

Huan Ning, Zhenlong Li, Xinyue Ye, Shaohua Wang, Wenbo Wang & Xiao Huang

INTERNATIONAL JOURNAL OF GEOGRAPHICAL INFORMATION SCIENCE
https://doi.org/10.1080/13658816.2021.1981334

Taylor & Francis
Taylor & Francis Group

RESEARCH ARTICLE

Exploring the vertical dimension of street view image based on deep learning: a case study on lowest floor elevation estimation

Huan Ning[a,b], Zhenlong Li[a], Xinyue Ye[c], Shaohua Wang[b], Wenbo Wang[b] and Xiao Huang[d]

[a]Geoinformation and Big Data Research Laboratory, Department of Geography, University of South Carolina, Columbia, USA; [b]Department of Informatics, New Jersey Institute of Technology, Newark, USA; [c]Department of Landscape Architecture and Urban Planning, Texas A&m University, College Station, USA; [d]Department of Geosciences, University of Arkansas, Fayetteville, Arkansas, USA

ABSTRACT

Street view imagery such as Google Street View is widely used in people's daily lives. Many studies have been conducted to detect and map objects such as traffic signs and sidewalks for urban built-up environment analysis. While mapping objects in the horizontal dimension is common in those studies, automatic vertical measuring in large areas is underexploited. Vertical information from street view imagery can benefit a variety of studies. One notable application is estimating the lowest floor elevation, which is critical for building flood vulnerability assessment and insurance premium calculation. In this article, we explored the vertical measurement in street view imagery using the principle of tacheometric surveying. In the case study of lowest floor elevation estimation using Google Street View images, we trained a neural network (YOLO-v5) for door detection and used the fixed height of doors to measure doors' elevation. The results suggest that the average error of estimated elevation is 0.218 m. The depthmaps of Google Street View were utilized to traverse the elevation from the roadway surface to target objects. The proposed pipeline provides a novel approach for automatic elevation estimation from street view imagery and is expected to benefit future terrain-related studies for large areas.

ARTICLE HISTORY
Received 15 March 2021
Accepted 13 September 2021

KEYWORDS
Street view; image; lowest floor elevation; vertical dimension; deep learning

1. Introduction

Street view imagery (SVI) became well known to the public since the launch of Google Street View (GSV). Usually, SVI is captured by mapping vehicles that record visual information of the surroundings via panoramas. After years of development and exploration, people now use SVI for various purposes, including navigation and travel planning. For example, SVI services provide visitors an intuitive way to understand their destination environments without being on the scene physically.

CONTACT Xinyue Ye ✉ xinyue.ye@tamu.edu

SVI's growing popularity is primarily due to its accessibility. Many SVI providers (e.g. Google Map, Bing Map, and Mapillary.com) have collected massive data covering major cities worldwide. People can access these images easily via a browser or an application on a smartphone. Notably, the cameras and LiDAR (Light Detection and Ranging) sensors equipped on autonomous vehicles also capture massive street images and point clouds, and these datasets are another emerging source of SVI. As a novel data source, SVI attracts scholars to explore professional applications. In the urban planning domain, city managers take advantage of SVI to collect municipal information, such as street trees and traffic facilities (Campbell *et al*. 2019). School district offices assess the safety of the walking routes to schools by analyzing SVI (Odgers *et al*. 2012, Mooney *et al*. 2016). Public health researchers mine the physical and mental health impacts of built environments (Kang *et al*. 2020). Most of these studies rely on the accurate identification of target objects geographically. Therefore, mapping is one of the most important procedures in SVI-based studies.

Essentially, mapping objects in SVI is a photogrammetric problem that aims to derive three-dimensional (3D) coordinates of physical objects from two-dimensional (2D) photos. Traditionally, photogrammetric methods use stereo images to model the physical space and then scale measurements on stereo images to the real world (Micusik and Kosecka 2009, Klingner *et al*. 2013, Bruno and Roncella 2019). The trigonometrical approach is also commonly used to map objects in SVI, and the solution of trigonometry varies with the known angles and side lengths (Agarwal *et al*. 2015, Krylov *et al*. 2018, Campbell *et al*. 2019, Lumnitz *et al*. 2021). Specifically, the directional angles of light rays are widely used as known elements in geometrical computation. Given the orientation information of SVI, the altitude and azimuth angle of a light ray can be converted from the position of its intersected pixel in a street view image. The known sides can be the lengths of light rays, the length of the object in real-world/image-space, or the baseline of the stereo images.

Despite these aforementioned efforts on mapping SVI objects, we observed that most mapping methods focus only on the horizontal dimension while neglecting the vertical dimension. Indeed, in many SVI applications such as estimation of greenness in the street, the vertical coordinate plays a trivial role, thus is often ignored. Another reason is that vertical coordinates are less accurate than horizontal coordinates in SVI's metadata. For example, vertical and horizontal accuracies are ±6 meters and ±2 meters in GSV (Tsai and Chang 2013). This vertical inaccuracy poses a challenge for terrain analysis and topography mapping using SVI. One solution to obtain reliable vertical dimension measurements is to conduct photogrammetric triangulation or 3D reconstruction (Bruno and Roncella 2019) using control points. The collection of control points, however, may not be feasible for large areas.

Taking Lowest Floor Elevation (LFE) estimation as a case study, we explored GSV's vertical dimension to facilitate the assessment of building flood vulnerability. LFE is a required measurement of the Elevation Certificate (E.C.) that is used for the National Flood Insurance Program. According to the definition from Federal Emergency Management Agency (FEMA 2010), the lowest floor refers to the lowest enclosed living area other than building access, parking, or storage. Given the water height of a flood event, the LFE determines whether floods can inundate the lowest living floor and cause property loss. Therefore, the premium for building flood insurance subjects to LFE (FEMA

2021). An E.C., completed by a licensed surveyor in the field, is required for the National Flood Insurance Program enrollment. The E.C. helps insurance agents accurately rate a flood insurance policy and assists FEMA and local neighborhoods with floodplain management compliance issues. Scholars attempt to explore scalable approaches to estimate LFE to avoid labor-intensive field survey. Multivariate regression analysis can be applied if necessary building characteristics are available (Gordon and Benjamin 2019).

Some researchers attempt to use street view imagery to measure LFE. They view the porch surface as the lowest floor surface and estimate the height from the porch surface to the ground visually. By adding the height to the ground DEM value, the LFE would be obtained. Essentially, these visual approaches use referencing length, such as roof edges and doorsteps, to estimate the height from the porch surface to the ground (Needham and Nick 2018, Gordon and Benjamin 2019). On most occasions, the bottom of the front door is close to the porch surface, though thresholds may rise up about 2 centimeters ("2018 International Building Code" 2018). Therefore, the elevation measurement of the front door bottom can be used as LEF (Gordon and Benjamin 2019).

In this paper, we propose a pipeline to estimate LFE via street view imagery following the principle of tacheometric surveying. Specifically, we use a side of known length and the apex angle against the side to compute the distance from the apex to the side. A front door is viewed as a subtense bar with a known height, and the light rays from the camera center to a door's top and bottom edge form the apex angle. Given these known elements, we apply trigonometry to restore the door's 3D coordinates related to the camera center. Combining the depthmap in GSV and the external DEM, the elevation of the door bottom is obtained. Our results suggest that the average error of LEF estimation in the study area is 0.218 m. The proposed method applies the deep learning technique (i.e. multi-layer neural network) to detect doors, then computes the door's elevation by traversing roadway surface elevation to the door via trigonometry. The proposed measuring method uses single images other than stereo or multiple images for triangulation, which is straightforward, automatic, and can be scaled out for large areas covered by SVI and DEM data.

Our study also shows how to utilize the object dimension information other than only the category information returned by deep learning object detectors, inspiring other surveying applications based on deep learning. The contributions of this study are threefold: (1) This study is among the first efforts on investigating the vertical measurement using single street view images; (2) we develop a method to link vertical measurements on SVI to vertical coordinate systems and provide accuracy assessment; (3) we propose a scalable method to estimate LFE using publicly accessible DEM data and street view data.

2. Related work

2.1. Surveying based on street view imagery

Online SVI services offer continuous image serials with geometric restrictions, which ensure SVI aligns to the physical streets seamlessly. Therefore, the measurement of SVI space can be converted to physical space, and such a conversion is essentially an application of photogrammetry, a well-established and widely used discipline in the surveying domain. The core of photogrammetry is triangulation. It is a technique that uses known angles and distances to measure unknown ones with several implementations considering the selection of known elements. In most cases, the altitude and azimuth angle of an object are known elements because they can be obtained from its position in an oriented SVI. The locations of SVIs are usually known elements.

Based on these known angles and lengths, Hebbalaguppe *et al.* (2017) conducted photogrammetric triangulation to obtain the position of telecom assets from GSV. Yan *et al.* (2013) applied Direct Linear Transformation (DLT), a topical close-range photogrammetric method, on GSV to derive the locations of traffic signs. They recommended using more than three images and 10 control points to obtain a root mean square error (RMSE) of less than 1 meter vertically and horizontally. Tsai and Chang (2013) developed an approach based on GSV to measure Points of Interests (POIs). Users obtained the 3D position of POIs by the forward intersection method using two overlapping street view images. The evaluation shows that the extracted vertical and horizontal coordinates derived from GSV have an error of 6 meters and 1–2 meters, respectively. In comparison, the method proposed in this study obtained a vertical error of 0.218 m, which is a significant improvement.

SVI can be used to construct 3D city models, in which the point cloud generation from SVI is a key step. After deriving point clouds with customized image matching techniques for SVI (Klingner *et al.* 2013), 3D models of buildings and the ground (Micusik and Kosecka 2009) can be generated. Bruno and Roncella (2019) conducted an accuracy assessment, suggesting that the 3D location error for such models was a few meters without using control points. Hara *et al.* (2013) suggested that the LiDAR point clouds collected by mapping vehicles simultaneously with the SVI can also be used for measuring purposes. GSV provides such point clouds in a sparse depthmap format, as well as other competitors such as Baidu Map. These depthmaps associated with panoramas can be used to reconstruct 3D city models (Cavallo 2015). Our study uses them to traverse the elevation of the roadway surface to the objects, connecting the vertical measurement in image space to physical space (see Section 3).

2.2. *Object detection and mapping from street view imagery*

The emerging deep learning technique in the computer vision domain has obtained remarkable progress in recent years (Zhao *et al.* 2019, Liu *et al.* 2020). Researchers started to utilize these techniques on SVI to detect and map objects for built environment analysis, such as street trees detection (Branson *et al.* 2018, Lumnitz *et al.* 2021), traffic signs mapping (Campbell *et al.* 2019, Balado *et al.* 2020), and sidewalk extraction (Koo *et al.* 2021, Ning *et al.* 2021). These studies used deep learning-based object detection approaches to mark target objects in SVI then map them according to their position in images and the geolocation of SVI.

Campbell *et al.* (2019) used the focal length, pixel size of the panorama camera, and height of traffic signs as known elements in triangulation to compute the distance from the camera to traffic signs. They further mapped the signs according to their azimuths and the locations of panoramas. This approach requires accurate internal parameters (i.e. focal length and pixel dimension) of cameras, which may not be feasible for undocumented SVI. External location information can be introduced to SVI object mapping. In a study to extract street trees (Branson *et al.* 2018), the georeferenced overhead satellite image was combined with GSV to train a multi-view tree detection model. The authors applied a Conditional Random Field (CRF) framework to improve the detection results and simplified an object detecting model to output a tree's longitude and latitude instead of a bounding box. Similarly, Laumer *et al.* (2020) detected street trees from GSV then matched results to the tree inventories without geographic coordinates. Ning *et al.* (2021) combined aerial images and SVI to extract sidewalks. However, the vertical dimension was not investigated in the above efforts.

The critical step in SVI-based surveying is to determine the distance from the camera to the target. Most deep learning approaches can estimate these distances based on stereo images, but camera parameters and rectified images are also required (Laga *et al.* 2020), usually lacking in SVI. Studies on monocular depth estimation can estimate distance using a single image instead of stereo images, and it can achieve a relative accuracy of 0.07 and an RMSE of 2.3 meters (Ming *et al.* 2021). Krylov *et al.* (2018) used a monocular depth estimation approach to obtain the camera-to-object distance. They applied a pre-trained CNN monocular depth estimation model to extract depthmaps from SVI, then combined a customized Markov random field (MRF) and triangulation to refine objects' location. In their case study of traffic lights and telegraph poles, the vertical coordinate was ignored. Lumnitz *et al.* (2021) use Mask R-CNN (He *et al.* 2017), monocular depth estimation, and triangulation to detect and locate street trees from SVI. The mean error of the horizontal position is 4–6 meters, and the authors suggested improving geolocation performance on the vertical dimension.

Though surveying using a single image with monocular depth estimation is possible, many studies utilized multiple images and triangulation to improve location accuracy (Krylov *et al.* 2018, Branson *et al.* 2018, Lumnitz *et al.* 2021). We propose a method to use single images for vertical measurement, and the accuracy assessment shows that the result is competitive (RMSE = 0.319 meters) compared with the best results reported using dense Ground Control Points (GCPs), whose distance RMSE (combination of horizontal and vertical dimension) is 0.228–0.254 meters (Bruno and Roncella 2019).

2.3. Lowest floor elevation (LFE) estimation

The lowest floor refers to the lowest enclosed living area other than building access, parking, or storage (FEMA 2010). Given the water height of a flood event, a building's LFE determines whether floods will inundate the lowest living floor and cause property loss. Therefore, the premium for building flood insurance subjects to LFE (FEMA 2021). An E.C. completed by a licensed surveyor is a required document for the enrollment in insurance programs.

E.C. inventories contain LFE records of associated buildings so that hazard responders can use these records to assess the flooding damage given a floodwater height (Cawood 2005). However, E.C. inventories are only available for newly built or renovated homes in most cities (Taghinezhad *et al.* 2020). The traditional method to obtained LFE is field surveying using optical and GPS instruments, which is accurate but labor-intensive (Cawood 2005). Ground-level LiDAR is an alternative to extract LFE. For instance, Ibrahim and Lichti (2012) used a mobile terrestrial laser scanner to produce point clouds of streets then model street curbs and street floors. However, it is difficult to find the lowest floor using the point cloud data without texture information.

Many studies tried to estimate LFE using existing datasets to avoid labor- and time-consuming field surveys. For example, Gordon and Benjamin (2019) developed a method using multivariate regression and random forest to estimate the LFE for residential structures using building characteristics as input variables, e.g. foundation type, built-year, and flood-zone. These models are subject to building characteristics inventories and associated LFE recorded as the training dataset. There are no common models for jurisdictions that used different building characteristics. For example, a city may code foundation types as categories (e.g. concrete or brick wall) while others may use numeric systems (e.g. the height of crawl space). Our method is relatively universal for large-area estimation covering several jurisdictions.

In contrast, Needham and Nick (2018) directly measured LFE from GSV and Google Earth. For each building, they manually measured the length of a roof edge in the façade in meters based on the overhead images from Google Earth, then measured the same roof edge in pixels on GSV to establish the conversion formula from pixels to meters. Next, they measured the distance in pixels from the lowest floor to the ground in the same façade on GSV then converted the measurement into meters. Though field surveying has been avoided, establishing the conversion between pixels and meters for each building is also time-consuming and labor-intensive (4 hours for about 12 buildings) as appropriate roof edges that can be measured both on Google Earth and GSV need to be carefully selected. In addition, the ground under the lowest floor is sometimes occluded by cars or plants, leading to the failure to measure the distance from the lowest floor to the ground. Similarly, the U.S. Army Corps of Engineers counted the risers of the doorstep leading up to a front door, then used the product of riser number and its height to estimate LFE (Gordon and Benjamin 2019). Given the variance of height of risers, significant errors might occur using such an approach. Moreover, the LFE of houses without stairs cannot be estimated.

These two GSV-based methods, requiring an object with a known length as a reference (i.e. roof and doorstep), can be applied to places where GSV data are available. Following a similar principle, we treat the front door as a length reference. Since the minimum height of an external door is 2.03 meters in the U.S. ("International Building Code" 2018), we assume the front door heights of residential houses are 2.03 meters in this paper.

3. Methodology

This section describes the proposed method to estimate the vertical coordinate of objects with a known height. We take the estimation of LFE as a case study to explore the utility and applicability of the vertical dimension of SVI. Street view images used in the training dataset and LFE estimation can be obtained from Representational State Transfer Application Program Interface (REST API) of street view services such as Google Map, Bing Map, Baidu Map, and Mapillary. Images and LiDAR data from sensors of autonomous vehicles can be used as well. When preparing images, their altitudes, azimuths, and angles of the field of view should be recorded for further door localization. In this study, we applied images from GSV and DEM data of the study area (Hampton Roads metropolitan area, USA) to demonstrate the workflow. Our method conducts the following steps to estimate the LFE for individual houses (see Figure 1, Section 3.2, and 3.3 for more details):

(1) Detect the door in SVI using a trained object detection model.
(2) Calculate the vertical coordinate of the door bottom to the camera center using its height and apex in the panorama.
(3) Calculate the door bottom elevation according to its vertical coordinate, the extracted roadway elevation, and the height from the camera to the roadway.

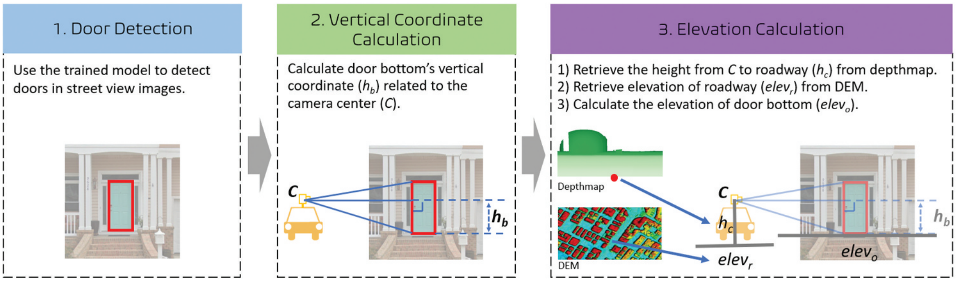

1. Door Detection
Use the trained model to detect doors in street view images.
2. Vertical Coordinate Calculation
Calculate door bottom's vertical coordinate (h_b) related to the camera center (C).
C
h_b
3. Elevation Calculation
1) Retrieve the height from C to roadway (h_c) from depthmap.
2) Retrieve elevation of roadway ($elev_r$) from DEM.
3) Calculate the elevation of door bottom ($elev_o$).
Depthmap
DEM
C
h_c
$elev_r$
$elev_o$
h_b

Figure 1. Pipeline of the proposed method.

3.1. Door detection

A training dataset is required to train an object detection model, which is YOLO-v5[1] in the case study. We annotated the bounding boxes of doors in the images as training samples and ensure that the bounding boxes have a position error of less than 1 pixel for accurate detection. YOLO (You Only Look Once) detectors (Redmon *et al.* 2015, Redmon and Farhadi 2016, 2018) are widely used for various visual tasks due to their good balance between speed and performance. YOLO detectors return bounding boxes of targeting objects in real-time with practicable accuracy. A YOLO model divides the input image into grids and predicts object information for each grid, including the bounding box, object category, and associated confidence. Note that the door height returned by the YOLO model is a normalized door height by dividing the image height. This height should be stored as a float number (e.g. 190.31 pixels) to avoid accuracy loss from quantization or rounding operation.

3.2. Vertical coordinate calculation

The bounding boxes of doors from the object detecting model need to be restored to the vertical dimension for elevation estimation. The geometrical relationship between the camera and the target object is used to estimate elevation (Figure 2). Based on the principle of tacheometric surveying, the object's height *AB* (i.e. a subtense bar) and apex angle ($\angle ABC$) to camera center (point *C*) are known elements and used to compute horizontal camera-to-object distance (d_{hor}). We set the camera center as the origin in the vertical axis, and the vertical coordinate of object bottom can be calculated by Equation (1),

$$h_b = \frac{h_o \cdot sin(\theta_b) \cdot cos(\theta_t)}{sin(\theta_t + \theta_b)} \tag{1}$$

where:

$\theta_t,\ \theta_b$: altitude angles from the camera to the top/bottom edge of the target object. $\theta_t +\ \theta_b$ is the apex angle against the object height *AB*. Appendix A shows the method to convert image coordinates to altitude and azimuth angles in a spherical coordinate system.

d_{hor}: horizontal distance from the camera to the target object, $d_{hor} = ctg(\theta_t) \cdot h_t = \frac{cos(\theta_t)}{sin(\theta_t)} \cdot h_t$;

h_o: height of the target object, i.e. the length of *AB*, $h_o = h_t + h_b$;

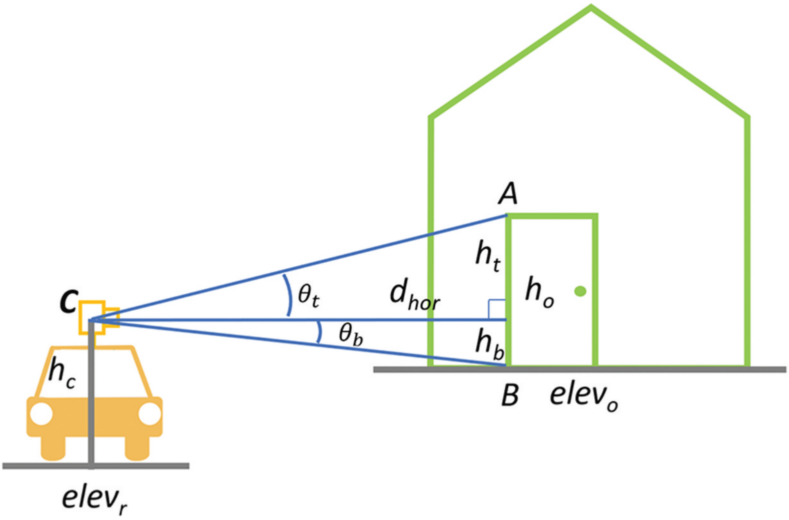


Figure 2. Vertical measurement based on the principle of tacheometric surveying.

h_t: the height from the camera to the top of the target object, $h_t = tan(\theta_t) \cdot d_{hor} = \frac{sin(\theta_t)}{cos(\theta_t)} \cdot d_{hor}$, $h_t = h_o - \ h_b$;

h_b: the height from the camera to the bottom of the target object, $h_b = tan(\theta_b) \cdot d_{hor} = \frac{sin(\theta_b)}{cos(\theta_b)} \cdot d_{hor}$, and by plugging $d_{hor} = \frac{cos(\theta_t)}{sin(\theta_t)} \cdot h_t$ and $h_t = h_o - \ h_b$, we can obtain Equation (1).

3.3. Elevation calculation

3.3.1. Calculate the bottom elevation of the target object

According to Figure 2, the elevation of object bottom, i.e. the lowest floor elevation in the case study (point *B*), can be calculated by Equation (2):

$$elev_o = elev_r + h_c - h_b \tag{2}$$

where:

$elev_o$: elevation of the bottom of the target object, i.e. the LFE in the case study;

$elev_r$: elevation of the roadway surface, extracted from DEM according to the geolocation of the panorama;

h_c: camera height, i.e. the vertical distance from the camera center to the roadway surface. See Section 3.3.2 for more details to extract h_c from the GSV depthmap.

3.3.2. Extract camera height from the depthmap

The depthmap from SVI services contains the distances from the camera to the planes of the scene used for interactive navigation. For example, locations with valid data of a depthmap indicate passable areas, and users can sense these passable areas and distances to the viewpoint according to the cursor's shape, size, and tip. Usually, the depthmap associated with a panorama can be converted to raster format in equirectangular projection as the panorama. Figure 3 shows a GSV depthmap, its corresponding panorama, and the derived point cloud. The dimension of a GSV depthmap is 512 × 256 pixels. Figure 3 (d) demonstrates the spherical coordinate system used in the panorama and depthmap. The central column of a depthmap points to the forward direction of the mapping vehicle, while the leftmost and rightmost column toward the rear of the mapping vehicle. The bottom row stores the vertical distance from the camera center to the roadway, i.e. the camera height, h_c, which can be extracted from the central point of the bottom line of the depthmap. The red dots in Figure 3 are the projection point of the camera center on the roadway surface.

3.3.3. *Error assessment of camera height*

The camera height (h_c) is critical in elevation traverse. Its error impacts the final estimation. Due to the lack of validation data, it is difficult to assess the absolute error of h_c. However, the precision of h_c can be observed by computing the standard deviation of all successive depthmaps in a trajectory of the mapping vehicle.

We downloaded 1,863 trajectories of the GSV mapping car in the study area (Figure 4). The length of these trajectories varies from 21 to 200 panoramas obtained in the same month, and the spacing distance between two adjacent panoramas is around 10 m. There are 57,797 depthmaps in these trajectories. We extracted each panorama's h_c from their associated depthmap and grouped them by trajectory lengths to compute the *mean standard deviation* (*mean std*). Table 1 lists the statistics of *mean stds* and the trajectory length. We noticed that the *mean std* is about 0.042 to 0.046

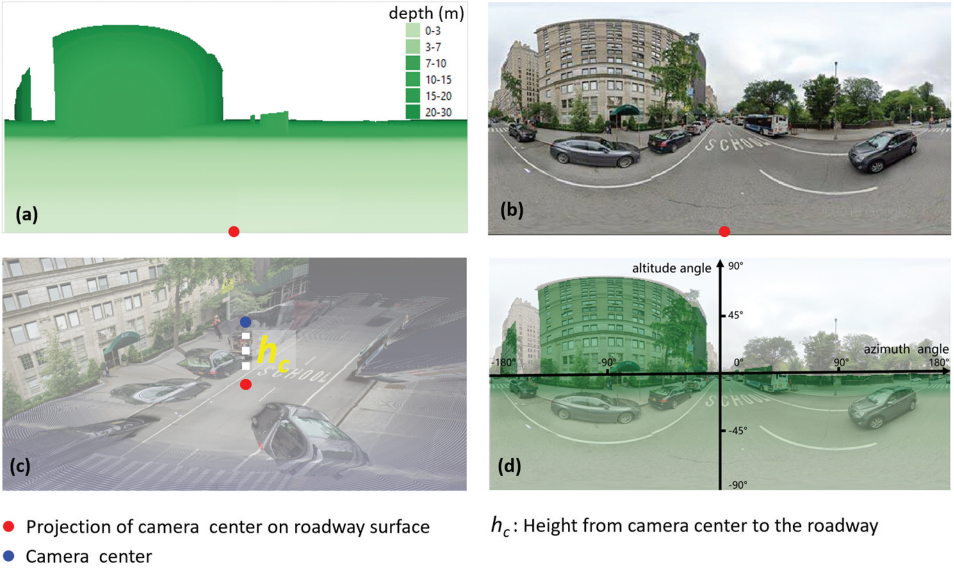


Figure 3. An example of depthmap (a), its corresponding panorama (b), and the point cloud (c). (d) shows the spherical coordinate system used in the equirectangular projection (d); the base image is the composition of (a) and (b). Image © Google.

Table 1. The *mean standard deviation* of camera height (h_c) grouped by trajectory lengths.

Trajectory Length (number of panoramas)	Number of trajectories	*Mean std* (meter)
21–50	847	0.046
51–100	591	0.046
101–150	316	0.042
151–200	109	0.042
Total	1863	0.045

meters regardless of the trajectory length and concluded that h_c is considerably stable (varying less than 0.05 m) when the vehicle conducting mapping tasks along the streets.

4. Experiment

4.1. Detect doors

4.1.1. Build the training dataset

A panorama covers 360° of a view while most objects, such as doors, taking up only a few proportions in the image. Therefore, high-resolution SVIs containing detailed textural and shapes of the target objects are needed for the training dataset. We downloaded the full-size GSV panoramas (16,384 ×8192 or 13,312 ×6656 pixels) and then reprojected the street sides into perspective images. The image size is one-fourth of the panorama (4096 ×4096 or 3328 ×3328) with a field of view (FoV) of 90° in both horizontal and vertical dimensions. The images used in the training dataset were obtained in Ocean City, New Jersey, U.S., without overlapping the images used in the case study.

Further, we annotated doors using Labelme[2] as the annotating tool. The bounding boxes for doors were drawn precisely to cover the moving panels, excluding door frames. Figure 5 shows four annotation examples. We annotated a total of 610 GSV images, among which 490 were used as the training set, and the remaining 120 were used as testing samples.

4.1.2. Train the object detection model

The YOLO-v5 model was trained on the annotated training set. YOLO-v5 has four different settings given the number of the learnable parameters: small, medium, large, and extremely large. A model with a small number of parameters is able to infer faster but in a less accurate manner, and vice versa. We used a GTX 1080 TI NVIDIA GPU to train YOLO-v5 with the extremely large parameter setting for 50 epochs.

4.2. LFE estimation

We downloaded the E.C. dataset of the Hampton Roads metropolitan area.[3] This dataset contains 3,611 polygons of building footprints with LFE records. Thumbnail images in perspective projection from the nearest panoramas to these building polygons are downloaded using a GSV REST API.[4] The key parameters of this API include: (1) **ll**: decimal latitude and longitude; (2) **panoid**: the identification number of a panorama,

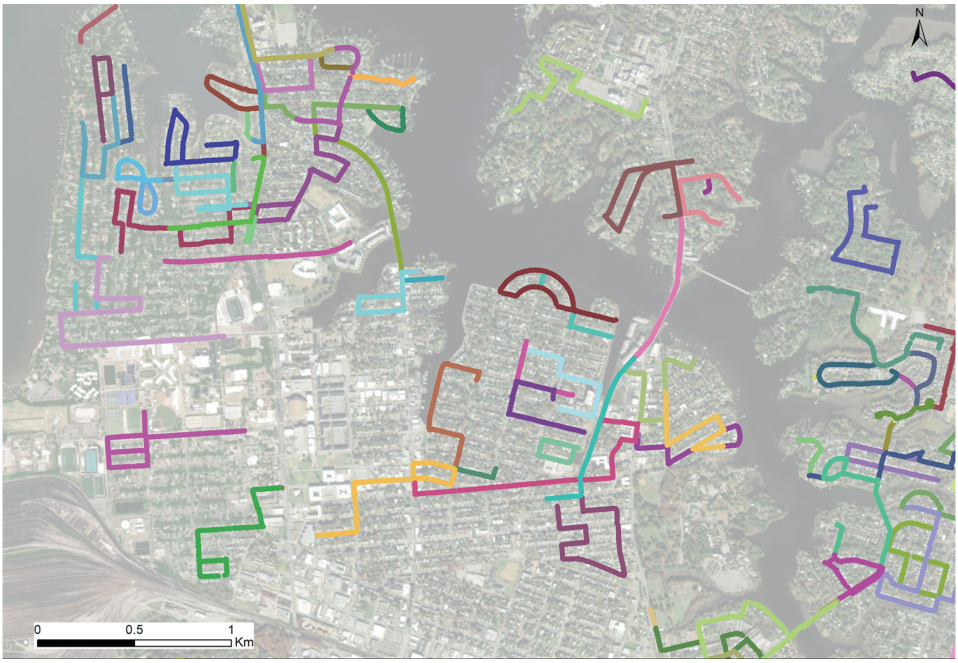


Figure 4. Examples of trajectories used to assess the stableness of camera height. Colors indicate different trajectories. Image © Mapbox.

either **ll** or **panoid** is satisfied, no need to use both; (3) **yaw**: the bearing angle of the camera (0–360°); (4) **pitch**: pitch angle (−8–8°); (5) **thumbfov**: the field of view angle (30–120°); and 6) **w, h**: the width and height of thumbnail with the maximum of 1024 ×768 pixels. Due to the limitation of thumbnail size, we set the **thumbfov** to a small angle of 30° to retrieve high-resolution images. Five overlapped thumbnails were used to cover the entire house, and each thumbnail has an overlap of 15° to its neighbors. Figure 6 presents the sliding window scheme mentioned above with the horizontal angle ranging from 60° to 120°. We employed the two postprocessing means to extract appropriate measurements from multiple detections among overlapped thumbnails: 1) choose the lowest measurement, and 2) remove detection touching the image edges. Doors above the first door will not be selected in the postprocessing if doors on the first floor are detected.

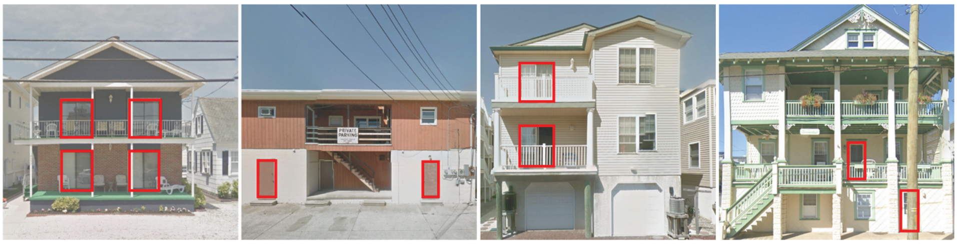


Figure 5. Examples of door annotation. Image © Google.

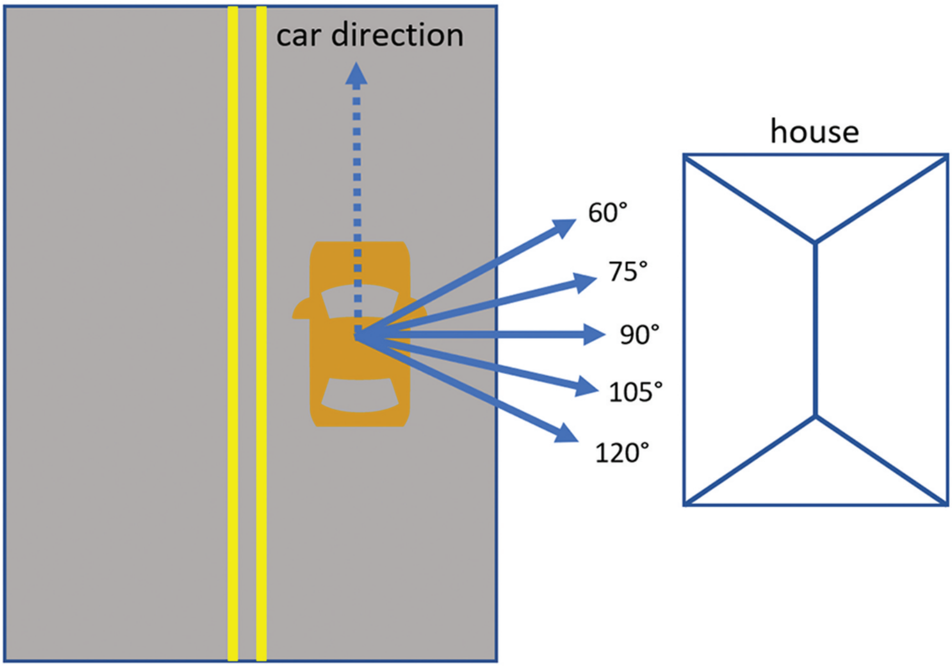


Figure 6. The scheme of shooting angles of the five overlapping thumbnails for target houses.

Most of the buildings in the E.C. dataset are residential houses. 2,500 out of 3,611 records can find a panorama within 30 meters, but many front doors were occluded by trees or cars. Some homeowners plant trees in front of their houses for privacy, and some front doors are not toward public roads. Panoramas captured before the year 2014 were not used due to their low resolution and quality. The 10-meter space between adjacent panoramas leads to some failures to shoot the intact front doors in an appropriate viewpoint. Regarding the reason mentioned above, we removed the GSV images without observable front doors, and 798 records were left for our experiment. The DEM of the study area was downloaded from the NOAA Sea Level Rise Viewer DEM dataset[5] derived from LiDAR data with a grid size of 3-meter. According to the metadata, the vertical accuracy is 10 cm for open terrain.

5. Results and evaluation

The precision and recall from the trained YOLO-v5 are 0.8000 and 0.5984, respectively. The IoU (Intersection of Union) is 0.8146. Because the case study focuses on the vertical dimension, we calculate the vertical IoU (*v–IoU*). Instead of areas, the intersection and union used to calculate *v–IoU* are heights of the intersected and union rectangles (I_v and U_v in Figure 7), $\mathrm{v} - \mathrm{IoU} = I_v/U_v$. Vertical errors of top edge ($error_t$) and bottom edge ($error_b$) of the detected doors are also measured and normalized by dividing the door height ($error_t = E_t/H$, $error_b = E_b/H$, where $E_t$and E_b are the vertical errors of the door's top and bottom edge respectively, see Figure 7). The trained YOLO-v5 model obtained an average *v–IoU* of 0.9501 on the test set, indicating the height of the detected

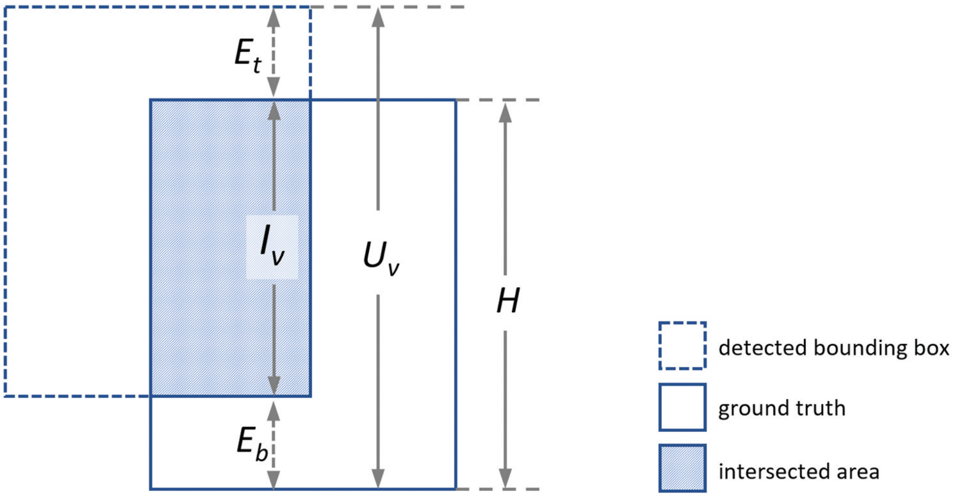


Figure 7. Calculation of vertical IoU ($\mathrm{v}-\mathrm{IoU}=I_v/U_v$) and vertical errors of the door's top and bottom edge.

bounding box of a door is close to the ground truth. The average *error*$_t$ and *error*$_b$ are 0.0469 and 0.0274, respectively, which mean the errors of detected top and bottom edge are about 9.52 cm (0.0469 × 2.03 meter) and 5.56 cm (0.0274 × 2.03 meter) respectively.

The trained YOLO-v5 model successfully detected front doors from 703 houses (88.1%) while failed for the remaining 95 ones. Most missing doors are occluded partially or in poor ambient light. Figure 8 shows examples of detected and missing doors. According to manual inspection, the model mistakenly marked 21 windows and one garage as doors for 20 houses, but most of these mistakes (16 windows and one garage) were removed by postprocessing (choosing the lowest measurements and removing measurements touching image boundaries). Overall, these false door detections caused wrong LFE estimations for three houses (errors range from 0.301 to 1.117 meters) and missing estimation for three houses. The precision for houses is 0.9957 (1–3/703), indicating that most detected doors are correct.

We consider the LFE of these houses in the E.C. dataset as ground truth and used them to evaluate the derived LFE (*GSV_LFE*). Figure 9 (a) shows the error distribution: about two-thirds (65.2%) errors are less than 0.2 m, with errors larger than 0.3 meters accounting for 22.9%. The mean error of *GSV_ LFE* is 0.218 m, the median error is 0.146 m, the RMSE is 0.319 m, and the maximum error is 1.36 m. Figure 9 (b) reveals a strong correlation between the *GSV_LEF* and its ground truth LFE with an R^2 of 0.861. The trend line in the linear regression has a slope of 0.985 and an intercept of −0.008, indicating a stable performance of the model with high accuracy. The spatial patterns of the *GSV_LEF* error distribution are shown in Figure 10. Densely built-up areas have slightly high errors through a visual inspection.

The case study results are promising, suggesting the vertical dimension of GSV can be extracted via the proposed method and applied in various domains, especially the hydrological analysis in small areas such as flood risk assessments. To better understand the error of the vertical measurement of GSV via trigonometry, we have Equation (3) and (4) according to the error propagation principle (Ghilani 2017) and Equation (2),

$$DEM_{rmse}^2 + GSV_{rmse}^2 = ELE_{rmse}^2 \tag{3}$$

$$GSV_{rmse}^2 = hc_{rmse}^2 + tri_{rmse}^2 \tag{4}$$

where:

ELE_{rmse}: the RMSE of the object elevation, i.e. the composition of independent variables of DEM_{rmse} and GSV_{rmse};

DEM_{rmse}: the RMSE of DEM;

GSV_{rmse}: the RMSE of the elevation traversing using GSV, i.e. the composition of independent variables of hc_{rmse} and tri_{rmse};

hc_{rmse}: the RMSE of h_c;

tri_{rmse}: the RMSE of the trigonometrical process to obtain h_b.

Given that ELE_{rmse} is 0.319 m, hc_{rmse} is about 0.05 meters estimated from 1,863 trajectories, and DEM_{rmse} is 0.1 meters according to the DEM metadata, we can solve tri_{rmse}, which reflects the RMSE for the measurement derived from GSV, as 0.299 m. This error provides a valuable reference for future research on vertical measurement using trigonometry based on GSV.

Our method is competitive in vertical surveying compared with previous research with and without GCPs. Per our knowledge, an accuracy assessment of 3D models generated from GSV (Bruno and Roncella 2019) reported the best results: an RMSE

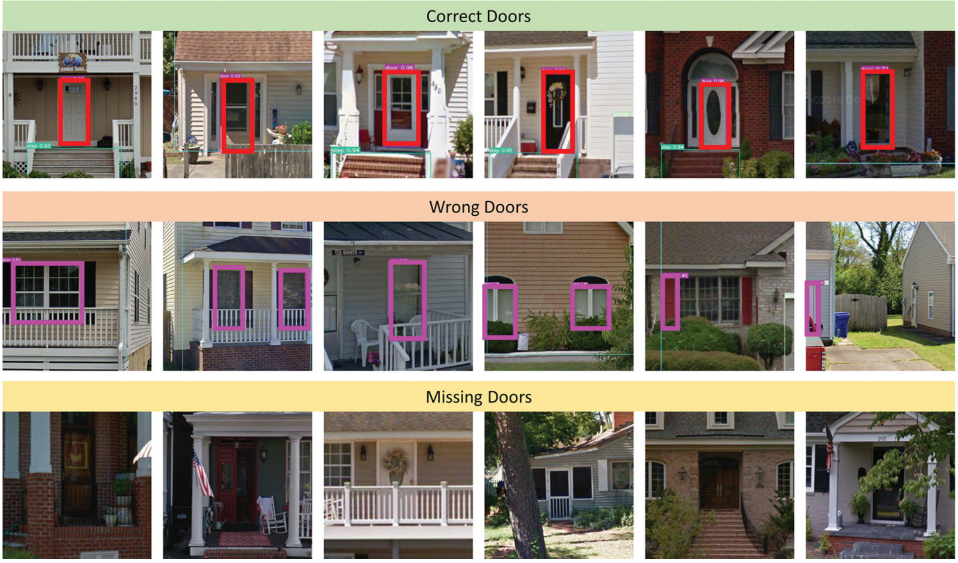


Figure 8. The correct, wrong, and missing detections in the detected results of the case study. Red (correct) and purple (wrong) rectangles are detected doors returned by the trained model. Image © Google.

Table 2. Errors of estimated LFEs reported from different methods. Note that these errors are obtained from different datasets and may not be comparable.

Method	Average (m)	Median (m)	Limitations
Random Forest, (Gordon and Benjamin 2019)	0.305	0.189	Required building characteristics vary among cities.
Integrated Surveying of Google Earth and GSV, (Needham and Nick 2018)	0.100*	Not reported	Manual, laborious, and difficult to link to the vertical coordinate system.
Step Counting, (Needham and Nick 2018)	0.100**	Not reported	
Ours	0.218	0.147	Requires observations of intact front doors.

*The authors report the average error of the height of the lowest floor above ground only, not the lowest floor elevation, and the Step Counting method in the next row uses the same metric.

**A half of the height of a stair, about 0.1 m.

0.228–0.254 meters using 4 GCPs in a structure-from-motion block (Schonberger and Frahm 2016). However, the report also revealed that they need to use a dataset containing three years of images to produce results for a typical narrow street because the 3D reconstruction failed when using images from an individual year only. The authors think that the high perspective distortion caused by wide base-lengths and short camera-to-object distances leads to such failures. The requirement of GCPs and potential failures of 3D reconstruction limit the application of the method developed by Bruno and Roncella (2019).

While with a slightly higher RMSE of 0.299 meters, our method is scalable, automatic, and it is only subjected to data availability. More importantly, it does not need GCPs and 3D reconstruction. One should note that the RMSE comes from the research of Bruno and Roncella (2019) is based on a different dataset and may not be comparable with our results.

Errors of estimated LFEs reported from different methods are listed in Table 2. Our approach obtained better results than Random Forest (Gordon and Benjamin 2019) but worse than the manual measurements from the Google Earth and GSV and step counting (Needham and Nick 2018). However, the latter two methods are labor-intensive, time-consuming, and challenging to be automatized. In addition, the step counting method needs to add the DEM value of the ground under the front door to obtain LFE, which is usually associated with higher uncertainty than the DEM values in open terrain (applied in our method). Despite their feasibility, these drawbacks limit their capability in measuring LFE from GSV for large areas. Our method does not require manual operation once the workflow and data have been set up.

For the vertical measuring error brought by doors with a height other than 2.03 meters, we can estimate it as $\frac{\Delta h_o \cdot sin(\theta_b) \cdot cos(\theta_t)}{sin(\theta_t + \theta_b)}$ according to Equation (1), where h_o is the difference between the actual door height and assumed height (2.03 meters). When a door is higher than 2.03 meters, h_b will have a smaller absolute value (closer to the camera center vertically) than its actual position; while a door is shorter than 2.03 meters, h_b has a larger absolute value (farther from the camera center vertically).

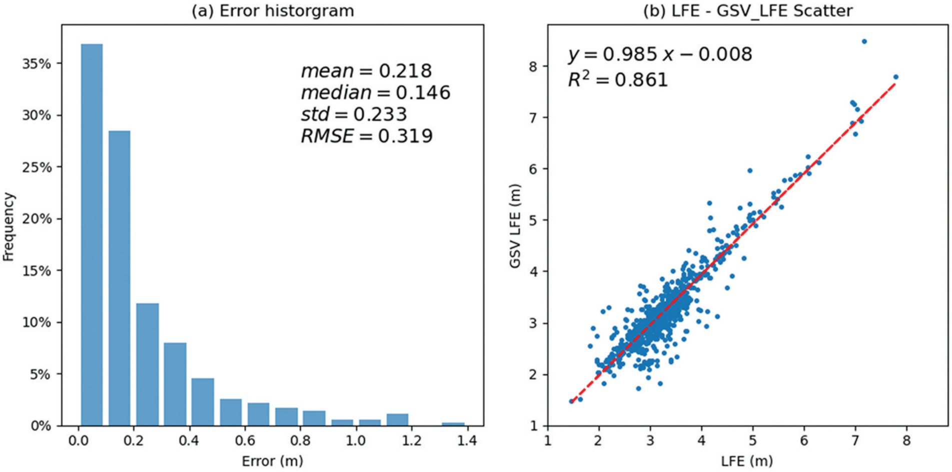


Figure 9. RMSE distribution. (a): most samples (65.2%) have an error of less than 0.2 m. (b): *GSV_LFE* (vertical axis) presents a strong positive correlation to ground truth (horizontal axis).

6. Discussions

6.1. Vertical coordinates measurement

The main idea of the proposed method is to use an object with a known height as a subtense bar to conduct tacheometric surveying. Candidates can be any object with widely accepted standards in size, such as house doors, garage doors, satellite dishes, lanes, traffic signs, traffic lights, bus stop shelters, parking spots, parking meters, crosswalks, road drains, utility equipment, streetlights, manholes, post boxes, trash bins, intermodal containers, wheelbases, and tires, to list a few. In our case study of using doors to estimate LEF, the tacheometric surveying archives an RMSE of 0.299 meters – about 15% of the door height (2.03 meters), which is competitive among methods without GCPs. Measurements in the horizontal dimension are also viable using azimuth and the horizontal distance of camera-to-object, which is d_{hor} in Eq. 1. Accurate detection of bounding boxes of these objects is needed in our method.

The accuracy of SVI-based trigonometrical is subject to the imaging of panoramas. If more details of imagining are known, such as camera parameters and image stitching, theoretical accuracy assessments and better results of high accuracy are possible. Empirical evaluations using more samples (e.g. buildings of a city) are necessary to fully investigate the accuracy of SVI-based measurements.

6.2. Panorama reprojecting distortion

The reprojection from panorama to perspective image has distortion. We analyzed the decay of image discriminability of this reprojection (i.e. from equirectangular projection to gnomonic projection) in Appendix B and found that it has no significant influence in the central area of the perspective image. In the projecting process, a pixel on the sphere surface far away from the tangent point (B in Figure A.1) will be magnified, covering several cells in the perspective images, leading to blurs in the perspective image (i.e. low

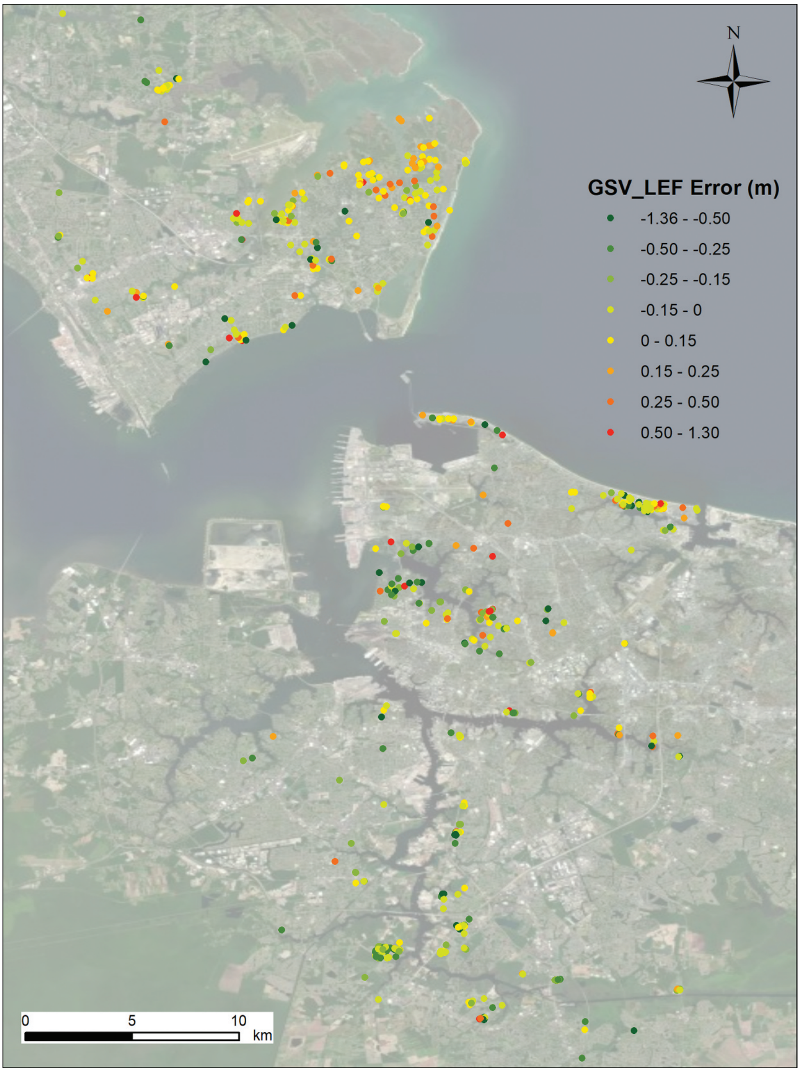


Figure 10. Spatial distribution of *GSV_LEF* error. Image © Mapbox.

discriminability). Bilinear- or cubic-sampling may compensate the blurs to some degree, but this complement is weak for large blurs in the margin where furthest away from the tangent point. These blurs are caused by the coarser projected height and width in the image margin (i.e. large altitude angle θ or azimuth angle φ) than in the center.

In our case study, the discriminability decay is minor because of the small FoV of 30°. In the corners of the perspective image, $\theta = 14.5^\circ$ and $\varphi = 15^\circ$, the projected area is 1.20 times to the central point. We do not find obvious correlations between the error and the location of the door top and bottom (Figure 11). Whether this decay of discriminability in the image margin affects the accuracy of the detected bounding box needs further investigation.

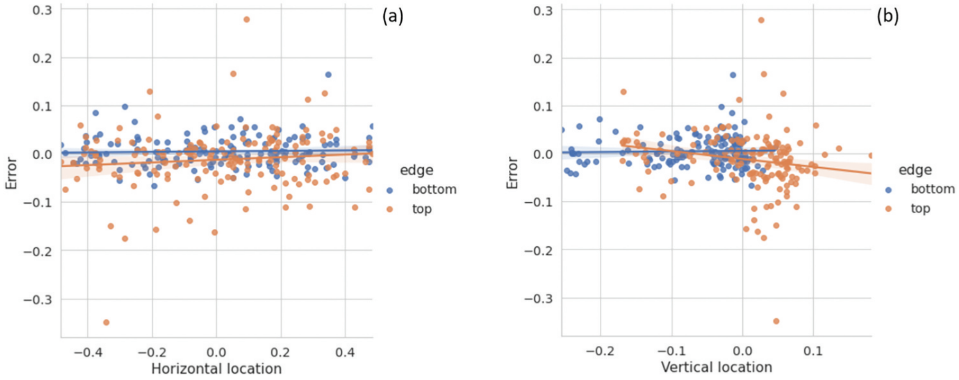


Figure 11. Edge location and predicted error have no obvious relationship. The origin of location is the central point of the perspective image, and both image height and width are normalized as 1. $Error = \frac{predict\ row - ground\ truth}{door\ height}$.

6.3. Recall and precision of door detection

On the door test set, the performance of YOLO-v5 is not very high (recall: 0.5984, precision: 0.8000). In the case study, the recall and precision of houses are 0.8810 and 0.9957, respectively, which are higher than the test set. The reasons to lower the model performance on the test set include: (1) Occlusion. Because of decoration, porch, and street trees, front doors are prone to be occluded by the mentioned objects, especially when the view angle is not perpendicular to the front door. The average interval of panoramas is 10 meters, so most view angles to front doors are not perpendicular. (2) The small size of the training set. There are many door types and decorations, and they vary in geographic regions. Our training set has 490 images only, which is relatively tiny for visual tasks based on deep learning. The model performance and door variance are under representative. (3) The small doors' relative area (door pixel number/image pixel number) in the test set. The mean area of doors in the test set is 0.0029 (FoV = 90°), and the detected mean door area is 0.0033, while the miss-detected is 0.0023. In contrast, by applying a small FoV (30°), the mean door area in the case study is 0.0185, which is over six times than test set. The correlation between recall and area reveals that a small FoV is beneficial to door detection.

The following approaches can improve the model performance: (1) Use historical panoramas obtained in multiple periods.[6] Sometimes the latest panoramas have clarity issues due to weather or lighting condition (e.g. heavy rain or shadow). Using historical panoramas is an alternative. (2) Add more training door images that cover typical styles in the area of interest. (3) Increase the area of the target object in the model input, or improve the model capability to detect small objects. The convolutional layer applied in YOLO-v5 and many other object detectors aggregate pixels to form features; thus, the information of the small object is prone to be overridden by neighbors. Our case study decreased the FoV to enlarge the doors in the perspective image while applying a sliding window to capture five overlapping thumbnail images to cover the entire target house. Front doors appear more than once in these images may help to increase the recall of houses.

A general object detected model, YOLO-v5, was used in this study to detect doors. Further modification of YOLO-v5 can be made to fit our scenario more to increase performance: (1) Design loss functions that focus on the horizontal edge (i.e. top/bottom) of the object bounding box. Our elevation shows that the vertical IoU (*v–IoU*) on the test set of 0.95, which results in errors of 5–10 cm in the top/bottom edge. The error should be decreased by improving the loss function. (2) Enhance the capability of processing large images. Many deep learning-based object detectors have many parameters then results in high memory consumption. The number of parameters is proportional to the number of input pixels. Thus, the dimension of the input image is usually limited. Street view panoramas have large dimensions (e.g. 16,384 × 8192 in GSV) to capture the entire scene, and it is challenging to feed a full-size panorama into the model. Sliding windows on multiscale are common techniques to tackle large images but will complicate the workflow. In our case study, due to memory limitation (11 GB in our GPU), YOLO-v5 down-sampled input images to 1600 × 1600 pixels. Some doors far away from the camera became 'small' (occupy few pixels) then lose their discriminability. Increasing the model input dimension can benefit small object detection and simplify the workflow (e.g. using a large FoV to avoid sliding windows).

For the miss-detected doors, labeling them manually is effective for the remaining workflow. The labeling results can also enrich the training dataset. However, the human worker needs to accurately delineate the top/bottom edge of a door, which is laborious and time-consuming. A customized training policy or loss function focusing on top/bottom edge accuracy may improve precision and recall, then minimize manual work. In our case study and similar photogrammetric applications, location accuracy at pixel level and sub-pixel level is critical. Per our knowledge, however, the literature on edge accuracy analysis at the pixel level is rare. It is worthy of investigating the techniques to obtain accurate bounding boxes for photogrammetric applications, as well as new metrics beyond *IoU* pixel-based accuracy.

6.4. Depthmaps of GSV

We applied GSV depthmaps to connect vertical measurements to the elevation system used in the DEM. The distance from the camera to the ground (h_c), or the nadir point, were be obtained from a depthmap. The analysis of 57,797 depthmaps from 1,863 trajectories shows that h_c has a relatively small RMSE of fewer than 0.05 meters. This characteristic can reference future applications. However, the precision and accuracy of the depthmap are still unknown. A full assessment of the accuracy of SVI depthmaps is worthy of further investigation. If the accurate vertical coordinate of the camera can be restored from metadata, the SVI-based vertical measurements can be introduced into a vertical coordinate system directly without using external data such as DEMs. More exploration of depthmaps from GSV or other SVI services can be conducted in future experiments.

6.5. Limitations

Due to the limitation of the E.C. dataset used in this case study, some potential biases are worth noting. The number of E.C. records (798) is small, considering the 1.7 million population in the Hampton Roads metropolitan area. The dataset mainly covers single houses in residential areas. The types and styles of buildings involved in this dataset are limited, so the detection of front doors for various types of buildings needs further investigation. For example, houses in the countryside may be far away from roads, then the small projection of doors in panoramas will lower the accuracy. Moreover, the door detected in the case study is not necessarily the measuring spot of the associated E. C. record, so that the evaluation may contain unknown errors. A large dataset containing precise vertical dimension information is needed to assess the ability of SVI in vertical measurement more accurately, such as 3D city models or larger E.C. datasets.

The case study of LFE extraction estimated 703 buildings out of 3,611 records in the E. C. dataset, meaning that about 20% of the total houses can be measured. Many doors were occluded by trees or vehicles. Other machine learning methods (Gordon and Benjamin 2019) using building characteristics can predict LFE for every building if necessary characteristics are collected. We believe SVI obtained from other periods may capture more doors then measure more buildings. Parallax is another potential way to obtain an object's 3D coordinates. Once an object's positions and sizes in different SVIs are measured, its physical size can be obtained based on the exterior orientation elements of SVI. One major challenge is establishing accurate correspondence of boundaries between multiple-view images in which the same object may have large shape distortion due to the change of perspective.

FEMA has detailed technical instructions to measure LFE (FEMA 2021) based on building structure and flood zones. Basements are usually considered as the lowest floor LFE. In our study, we suppose the lowest door in the facade leads to the lowest floor. In addition, the Hampton Roads E.C. dataset has all LEFs above their highest adjacent grade, so our experiment cannot reflect the error of LEF estimation for houses having basements below ground level. If the street view image cannot capture the basement entrance, the LEF estimation will be incorrect. Without a field survey, we have not found a way to measure or estimate the elevation of basements with no observable entrances in the street view images. Novel methods are needed to overcome this limitation.

7. Conclusion

In this study, we demonstrated an automatic workflow to measure the vertical coordinates of target objects from SVI. Based on the principle of tacheometric surveying, the height of an object is viewed as a subtense bar, and the apex angle is measured from SVI. Thus, the vertical coordinates of objects can be computed using trigonometry. In the case study, we used the fixed height of doors to estimate the lowest floor elevations of residential houses and obtained an average error of 0.218 meters, which is competitive compared to existing approaches. Trained on a door dataset built from SVI, a YOLO-v5 model is used to detect doors, then door vertical coordinates are calculated based on the position of the bounding boxes. Depthmaps provided by GSV were introduced to traverse elevations of road surfaces to the bottoms of front doors. The results suggest that the RMSE of trigonometric vertical measurement is 0.299 m, which references future applications.

We assessed the stability of the distance from the camera to the ground in GSV depthmaps, then obtained an RMSE of about 0.05 m. To our knowledge, this is the first attempt to report such characteristics of GSV depthmaps. Overall, our study presented a method to use GSV's vertical information. The proposed pipeline explored the vertical measurements for detected objects and is expected to benefit future GSV-related studies.

One should note the potential biases in our case study. The number of E.C. records used in the experiment is limited (789), and these records are formed by the residential houses in Hampton Roads metropolitan area. Moreover, the measuring spots of E. C. records are not necessarily to be the detected door bottom. Datasets of fewer limitations regarding representative, qualitative, and quantitative are needed to assess SVI's vertical measurement fully.

The proposed method is inapplicable to scenarios without objects of known size in the vertical dimension. The error of GSV-based trigonometry (RMES = 0.299 meters in our case study) also limits the application of our method in high precise surveying. In the future, potential studies include (1) vertical accuracy assessments of data from various SVI services and autonomous vehicle sensors and (2) other SVI-based measurements, such as sidewalk widths and street tree diameters.

Notes

1. YOLOv5, 25 October 2020. https://github.com/ultralytics/yolov5.
2. Kentao Wada, labelme: Image Polygonal Annotation with Python. https://github.com/wkentaro/labelme.
3. Elevation Certificates, data downloaded on 22 October 2020. https://www.hrgeo.org/datasets/elevation-certificates-building-footprints-navd88.
4. An example: https://geo0.ggpht.com/cbk?cb_client=maps_sv.tactile&authuser=0&hl=en&gl=us&output=thumbnail&thumb=2&w=768&h=768&pitch=0&ll=40.710359%2C-74.2535399&panoid=3hMZyKnDFRovyUuoXNK_ng&yaw=129&thumbfov=90.
5. NOAA Sea Level Rise Viewer DEM, 24 October 2020. https://coast.noaa.gov/htdata/raster2/elevation/SLR_viewer_DEM_6230/.
6. This Python module shows how to obtain historical panoramas: Roboyst, Streetview, https://github.com/robolyst/streetview.

Acknowledgements

The research was supported by National Science Foundation (NSF) under grants SMA- 2122054, University of South Carolina ASPIRE program under grant 135400-20-53382, USDOT/North Jersey Transportation Planning Authority, Texas A&M University Harold Adams Interdisciplinary Professorship Research Fund, and College of Architecture Faculty Startup Fund. The funders had no role in the study design, data collection, analysis, or preparation of this article.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

Huan Ning is a Ph.D. student in the Department of Geography at the University of South Carolina. His research areas include GeoAI and big data analysis with publication on image understanding and data mining for city management using advanced computing technology.

Zhenlong Li is an Associate Professor in the Department of Geography at the University of South Carolina, where he established and leads the Geoinformation and Big Data Research Laboratory (GIBD). He received B.S. degree (2006) in Geographic Information Science from Wuhan University, and Ph.D. (2015, with distinction) in Geography and Geoinformation Sciences from George Mason University. His primary research field is GIScience with a focus on geospatial big data analytics, spatial computing, cyberGIS, and geospatial artificial intelligence with applications to disaster management, public health, and human dynamics.

Xinyue Ye is a Harold Adams Endowed Associate Professor at the Department of Landscape Architecture and Urban Planning at Texas A&M University. He holds a Ph.D. degree in Geographic Information Science from Joint Program between University of California at Santa Barbara and San Diego State University, a M.S. in Geographic Information Systems from Eastern Michigan University, and a M.A. in Human Geography from University of Wisconsin at Milwaukee. His research focuses on geospatial artificial intelligence, smart cities, spatial econometrics and urban computing.

Dr. Shaohua Wang is an assistant professor in the Department of Informatics, College of Computing, New Jersey Institute of Technology. His research interests include software engineering, program analysis, and artificial intelligence. Dr. Wang has published research on top computer science conferences and journals, such as ICSE, FSE, ASE, OOPSLA and TSC.

Wenbo Wang is a Ph.D. student in the Department of Informatics, College of Computing, New Jersey Institute of Technology. His research interests include Natural Language Processing, Machine Learning, Vulnerability Detection, published in Transactions in GIS.

Xiao Huang is an Assistant Professor at the Department of Geosciences at the University of Arkansas. He holds a Ph.D. degree in Geography from the University of South Carolina and a Master's degree in Geographic Information Science and Technology from Georgia Institute of Technology. His research interests involve GeoAI, Remote Sensing, and Social Science.

ORCID

Xinyue Ye http://orcid.org/0000-0001-8838-9476
Xiao Huang http://orcid.org/0000-0002-4323-382X

Data and codes availability statement

The data and code that support the findings of this study are openly available at github.com/gladcolor/lowest_floor_elevation.

Author contributions

Conceptualization: H.N, Z.L., and X.Y; Investigation: H.N. and Z.L.; Funding acquisition, X.Y, S.W., and Z.L; Validation, X.H. and W.W.; Writing: H.N., Z.L, and X.H.; Review, S.W. and W.W.

References

2018 international building code. 2018. International Code Council, INC.

Agarwal, P., Burgard, W., and Spinello, L, 2015. Metric Localization Using Google Street View. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg Germany, 3111–3118. 10.1109/IROS.2015.7353807

Balado, J., *et al*., 2020. Novel approach to automatic traffic sign inventory based on mobile mapping system data and deep learning. *Remote Sensing*, 12 (3), 442. doi:10.3390/rs12030442

Branson, S., *et al*., 2018. From google maps to a fine-grained catalog of street trees. *ISPRS Journal of Photogrammetry and Remote Sensing*, 135 (January), 13–30. doi:10.1016/j.isprsjprs.2017.11.008

Bruno, N. and Roncella, R., 2019. Accuracy assessment of 3D models generated from google street view imagery. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W9 (January), 181–188. doi:10.5194/isprs-archives-XLII-2-W9-181-2019

Campbell, A., Both, A., and (Chayn) Sun, Q., 2019. Detecting and mapping traffic signs from google street view images using deep learning and GIS. *Computers, Environment and Urban Systems*, 77 (September), 101350. doi:10.1016/j.compenvurbsys.2019.101350

Cavallo, M., 2015. 3D city reconstruction from google street view, https://evl.uic.edu/documents/3drecomstrictionmcavallo.pdf. Accessed 30 January 2021

Cawood, T.J., 2005. Evaluating survey methods for obtaining first floor structure elevations for use in federal shore protection studies. *Solutions to Coastal Disasters*, 2005, 386–394.

FEMA, 2010. Home Builder's guide to coastal construction: technical fact sheet series. Federal Emergency Management Agency, https://campbellriver.ca/docs/default-source/planning-building-development/fema_buildersguide_coastalconstruction.pdf. Accessed 30 January 2021

FEMA, 2021. National Flood Insurance Program Flood Insurance Manual. Federal Emergency Management Agency, https://www.fema.gov/sites/default/files/2020-09/fema_flood-insurance-manual-full-edition_april-oct2020.pdf. Accessed 30 January 2021

Ghilani, C.D., 2017. Adjustment computations: spatial data analysis. 6th. John Wiley & Sons, Inc., John Wiley & Sons, Inc., Hoboken, New Jersey

Gordon, A. and Benjamin, M., 2019. Developing first floor elevation data for coastal resilience planning in Hampton Roads. Hampton Roads Planning District Commission. Chesapeake, Virginia

Hara, K., Le, V., and Froehlich, J., 2013. Combining crowdsourcing and google street view to identify street-level accessibility problems. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Paris France, 631–640

He, K., et al, 2017. Mask R-CNN. In 2017 IEEE International Conference on Computer Vision (ICCV), Venice Italy, 2980–2988. 10.1109/ICCV.2017.322

Hebbalaguppe, R., et al, 2017. Telecom inventory management via object recognition and localisation on google street view images. In 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). Santa Rosa, US. 725–733. 10.1109/WACV.2017.86

Ibrahim, S. and Lichti, D., 2012. Curb-based street floor extraction from mobile terrestrial LiDAR point cloud. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 39, B5.

Kang, Y., *et al*., 2020. A review of urban physical environment sensing using street view imagery in public health studies. *Annals of GIS*, 26 (3), 261–275. doi:10.1080/19475683.2020.1791954

Klingner, B., Martin, D., and Roseborough, J, 2013. Street view motion-from-structure-from-motion. In Proceedings of the IEEE International Conference on Computer Vision. Sydney, AU. 953–960

Koo, B.W., Guhathakurta, S., and Botchwey, N., 2021 May. How are neighborhood and street-level walkability factors associated with walking behaviors? A big data approach using street view images. *Environment and Behavior*, 00139165211014609. doi: 10.1177/00139165211014609.

Krylov, V.A., Kenny, E., and Dahyot, R., 2018. Automatic discovery and geotagging of objects from street view imagery. *Remote Sensing*, 10 (5), 661. doi:10.3390/rs10050661

Laga, H., *et al*., 2020. A survey on deep learning techniques for stereo-based depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1. doi:10.1109/TPAMI.2020.3032602

Laumer, D., *et al*., 2020. Geocoding of trees from street addresses and street-level images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 162 (April), 125–136. doi:10.1016/j.isprsjprs.2020.02.001

Liu, L., *et al*., 2020. Deep learning for generic object detection: A survey. *International Journal of Computer Vision*, 128 (2), 261–318. doi:10.1007/s11263-019-01247-4

Lumnitz, S., *et al*., 2021. Mapping trees along urban street networks with deep learning and street-level imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 175 (May), 144–157. doi:10.1016/j.isprsjprs.2021.01.016

Micusik, B. and Kosecka, J, 2009. "Piecewise planar city 3D modeling from street view panoramic sequences." In 2009 IEEE Conference on Computer Vision and Pattern Recognition. Miami, US. 2906–2912. 10.1109/CVPR.2009.5206535

Ming, Y., *et al*., 2021. Deep learning for monocular depth estimation: a review. *Neurocomputing*, 438 (May), 14–33. doi:10.1016/j.neucom.2020.12.089

Mooney, S.J., *et al*., 2016. Use of google street view to assess environmental contributions to pedestrian injury. *American Journal of Public Health*, 106 (3), 462–469. doi:10.2105/AJPH.2015.302978

Needham, H. and Nick, M., 2018. Analyzing the vulnerability of buildings to coastal flooding in Galveston, Texas.

Ning, H., et al., 2021. Sidewalk extraction using aerial and street view images. Environment and Planning B: Urban Analytics and City Science. doi:10.1177/2399808321995817

Odgers, C.L., *et al*., 2012. Systematic social observation of children's neighborhoods using google street view: a reliable and cost-effective method. *Journal of Child Psychology and Psychiatry*, 53 (10), 1009–1017. doi:10.1111/j.1469-7610.2012.02565.x

Redmon, J., et al, 2015. You only look once: unified, real-time object detection. ArXiv:1506.02640 [Cs], June. http://arxiv.org/abs/1506.02640. Accessed 30 January 2021

Redmon, J. and Farhadi, A, 2016. YOLO9000: better, faster, stronger. ArXiv:1612.08242 [Cs], December. http://arxiv.org/abs/1612.08242. Accessed 30 January 2021

Redmon, J. and Farhadi, A. 2018. "YOLOv3: an incremental improvement." ArXiv:1804.02767 [Cs], April. http://arxiv.org/abs/1804.02767. Accessed January 30, 2021

Schonberger, J.L. and Frahm, J.-M, 2016. Structure-from-motion revisited. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, US. 4104–4113

Taghinezhad, A., *et al*., 2020. An imputation of first-floor elevation data for the avoided loss analysis of flood-mitigated single-family homes in Louisiana, United States. *Frontiers in Built Environment*, 6. doi:10.3389/fbuil.2020.00138

Tsai, V.J.D. and Chang, C., 2013. Three-dimensional positioning from google street view panoramas. *IET Image Processing*, 7 (3), 229–239. doi:10.1049/iet-ipr.2012.0323

Yan, W.Y., Shaker, A., and Easa, S., 2013. Potential accuracy of traffic signs' positions extracted from google street view. *IEEE Transactions on Intelligent Transportation Systems*, 14 (2), 1011–1016. doi:10.1109/TITS.2012.2234119

Zhao, Z.-Q., *et al*., 2019. Object detection with deep learning: a review. *IEEE Transactions on Neural Networks and Learning Systems*, 30 (11), 3212–3232. doi:10.1109/TNNLS.2018.2876865

Appendix A: Convert image coordinate to spherical coordinate

Coordinates in perspective images from a panorama can be converted to spherical coordinates to extract pixel values from the panorama. Figure A.1 shows the geometric relationship between these two coordinate systems. We use a perspective image shot toward the panorama center (origin O in Figure 4) and parallel to the level plane for simplification. According to Figure A.1, for any pixel $P(x,y)$ in the perspective images, its altitude angle θ and azimuth angle φ in a spherical coordinate system can be calculated by Equation (A.1) and (A.2).

$$\theta = atan(CP/CO) \tag{A.1}$$

$$\varphi = atan(BC/BO) \tag{A.2}$$

where:

$\omega = \frac{fov_h}{2}$, fov_h is the horizontal FoV;

$CP = |y|$;

$BC = |x|$;

$BO = \frac{BE}{\tan(\omega)} = \frac{w/2}{\tan(\omega)}$

$CO = \sqrt{BO^2 + BC^2} = \sqrt{\left(\frac{w/2}{\tan(\omega)}\right)^2 + x^2}$

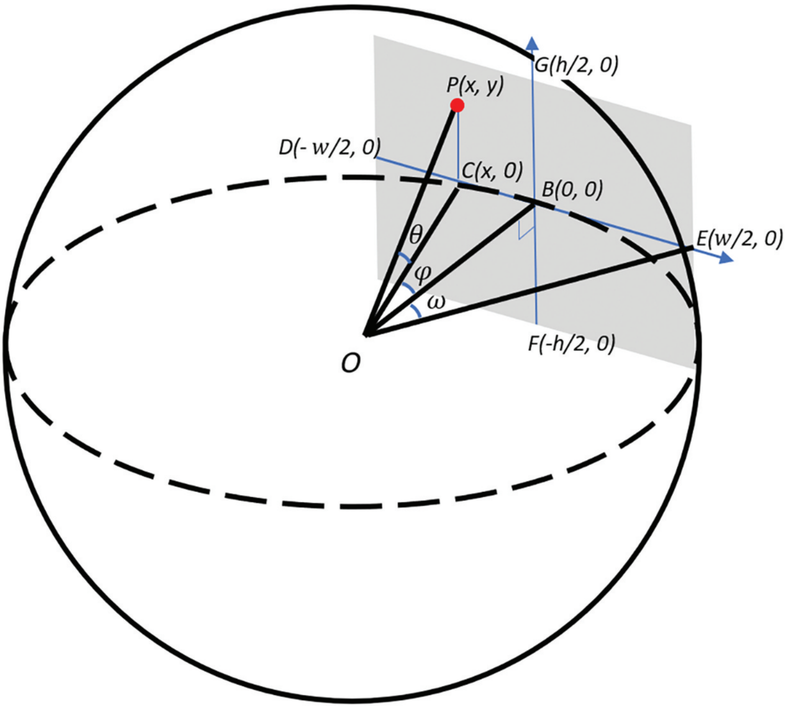


Figure A.1. Conversion from image coordinates to spherical coordinates. Shaded area: a rectangular perspective image of the panorama. O: the camera center of the panorama. B: tangent point of the perspective image plane and spherical surface. D, E, F, G: centers of four edges of the perspective image. $P(x,y)$: P is a pixel in the perspective image, and x, y are its coordinate in the image coordinate system from origin B. The unit of axes $\overline{BE}$ and $\overline{BG}$ is pixel. w, h: width and height in pixels of the perspective image.

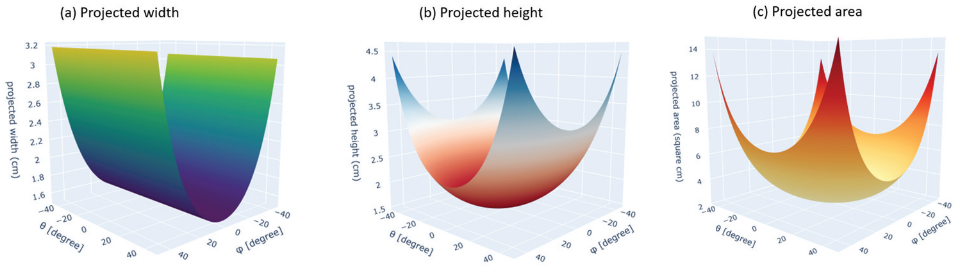
Appendix B: Calculate the projected distortion for the panorama pixels

To quantitatively analyze this discriminability decay, we calculate the projected area (in a perspective image frame) of each panorama pixel. According to Figure B.1 (denotation as Figure A.1), Δ is the angular resolution of panorama, *PP′* is the projected height (*Δh*) in the perspective image from the pixel in the location of (*θ*, *φ*) in panorama, and it can be obtained by Equation (B.1). Similarly, the projected width *Δw* can be computed by Equation (B.2). The projected area *Δa* is calculated by *Δh* × *Δw*.

$$\begin{aligned} \Delta h = PP' = CP' - CP &= tg(\theta + \Delta) \cdot CO - tg\theta \cdot CO \\ &= (tg(\theta + \Delta) - tg\theta) \cdot CO = (tg(\theta + \Delta) - tg\theta) \cdot \tfrac{BO}{cos\varphi} \end{aligned} \qquad (B.1)$$

$$\begin{aligned} \Delta w = CC' = BC' - BC &= tg(\varphi + \Delta) \cdot BO - tg\varphi \cdot BO \\ &= (tg(\theta + \Delta) - tg\theta) \cdot BO \end{aligned} \qquad (B.2)$$

Figure B.2 shows the dilation of projected width (Δ*w*), height (Δ*h*), and area of panoramas (Δ*a*) along with the increase of *θ* and *φ*. In the perspective image's four corners, the dilation increases dramatically, so the discriminability in the margin is lower than the image center. For example, in Figure B.2, when $\theta = 0$ and $\varphi = 0$, $\Delta h = 1.5\ cm$, $\Delta w = 1.5\ cm$, $\Delta a = 2.4\ cm^2$; when *P* is in the corner of the perspective images, $\theta = 35.5°$ and $\varphi = 45°$, $\Delta h = 3.2\ cm$, $\Delta w = 2.3\ cm$, $\Delta a = 7.5\ cm^2$, the area increases to 3.2 times, meaning the margin areas contain less textural information than the center area and may affect the localization of bounding box edges. Future studies are needed to fully understand the relationship between the discriminability decay and the location error of the bounding box.


(a) Projected width

(b) Projected height

(c) Projected area

Figure B.1. Calculate the projected height and width for the panorama pixels, whose angular resolution is Δ.

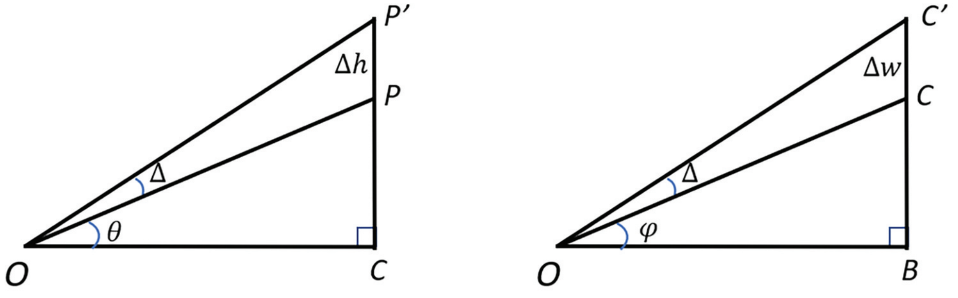


Figure B.2. Panorama pixel's projected height, width, and area in the perspective image. The distance from the camera (*BO* in Figure A.1) to the projected plane is set to 20 m, FoV is 90°, panorama height is 4092 pixels, and panorama angular resolution $\Delta = 180° / 4092\text{pixel} = 0.044°$.