Toward Map Updates with Crosswalk Change Detection Using a Monocular Bus Camera

Tom Bu, Christoph Mertz*, John Dolan*
Robotics Institute
Carnegie Mellon University, USA
{tomb, mertz, jdolan}@cs.cmu.edu
https://tom-bu.github.io/BusCamCrosswalkCD

2023 IEEE Intelligent Vehicles Symposium (IV) | 979-8-3503-4691-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/IV55152.2023.10186622

***Abstract*—Detecting when road maps change is useful for autonomous vehicles to drive safely and legally, for city planners to make more educated decisions, and web maps to better serve consumers. Many public vehicles drive around the city on a regular basis and collect road data for security and safety purposes through dash cams, yet few cities and companies have considered this as a data source for city monitoring. We present an automatic method and system for crosswalk change detection at city intersections using a monocular camera on a city bus and analyze longitudinal results over the course of a year. Using images recorded by a bus two years ago as reference, multiple city intersections are reconstructed, fitted for ground planes, and labelled for crosswalks. Subsequent images from the bus are imported and processed to detect if changes have occurred since intersections were first seen by first localizing current images with respect to the reference images, detecting for crosswalks, and computing detection overlaps in the bird's-eye-view. Our method makes improvements upon baseline methods by checking for crosswalk visibility and localization errors, is able to generate results typically seen by using more expensive LiDAR sensors, and has been successfully deployed live for one month.**

I. Introduction

Maps have made everyday navigation easier and safer. More recent advances in digital maps, such as Google Maps, include semantic information at a given location such as lane directions and nearby stop signs. While having this semantic knowledge in a map is useful for humans, autonomous vehicles depend on these maps for navigation. High definition (HD) maps enable many autonomous vehicles (AVs) and include both geometric and semantic information about crosswalks, lanes, and driveable areas.

Currently, most AVs are geofenced inside areas that are mapped to ensure safety. However, efficiently detecting changes in an HD map is a challenge. Lambert et al. [1] analyzed the frequency of map changes over a period of 5 months. Subdividing a city map into 30x30 meter square tiles, they estimated that there is a probability of a changed lane geometry or crosswalk in 7 out of 1000 map tiles in a 5-month span. Applying these statistics to the city of Pittsburgh, which has an area of 140 km^2, leads to an expected 7.4 tile changes per day.

This research was supported in part by the CMU Argo AI Center for Autonomous Vehicle Research, by NSF under Award No 2038612, and by Carnegie Mellon University's Mobility21 National University Transportation Center, which is sponsored by the US Department of Transportation.

*Authors share senior authorship.

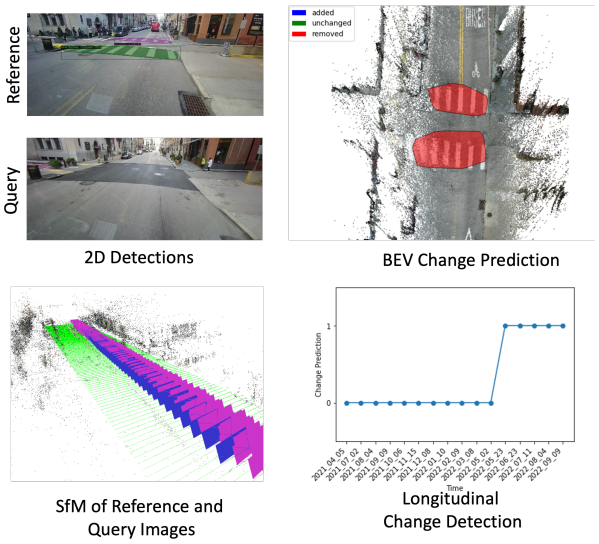

Fig. 1: 2D crosswalk detectors are used to analyze crosswalk changes at intersections over time (top left). Detections are represented in BEV (top right) by localizing images of reference images and query images (bottom left) and transforming detections onto a ground plane. Analysis of detected changes has been performed for the period of one year (bottom right).

The challenge with change detection is that the location and time of changes are unknown. Because road changes can occur any day, the operational costs of deploying specialized vehicles to perform daily mapping can be expensive. Crowd-sourced photographs have recently shown success in mapping and displaying changes in the environment [2]–[4]. In this paper, we leverage a commuter bus that travels daily around the city with camera and GPS sensors. In contrast to [2], [3], which used photos scraped from the internet, images from the bus provide regular crowd-sourced data. Though a commuter bus travels a limited route and has a narrow coverage of the city, the system requirements are minimal and can work for other service vehicles like garbage trucks and postal cars to expand the map coverage. Given the need for regular monitoring of the environment, this research shows how using a crowd-sourced vehicle-mounted camera can be used to efficiently detect map-relevant changes in noisy and high-traffic environments, as seen in Fig. 1. In this work we argue that rather than relying on LiDAR data, a similar bird's-

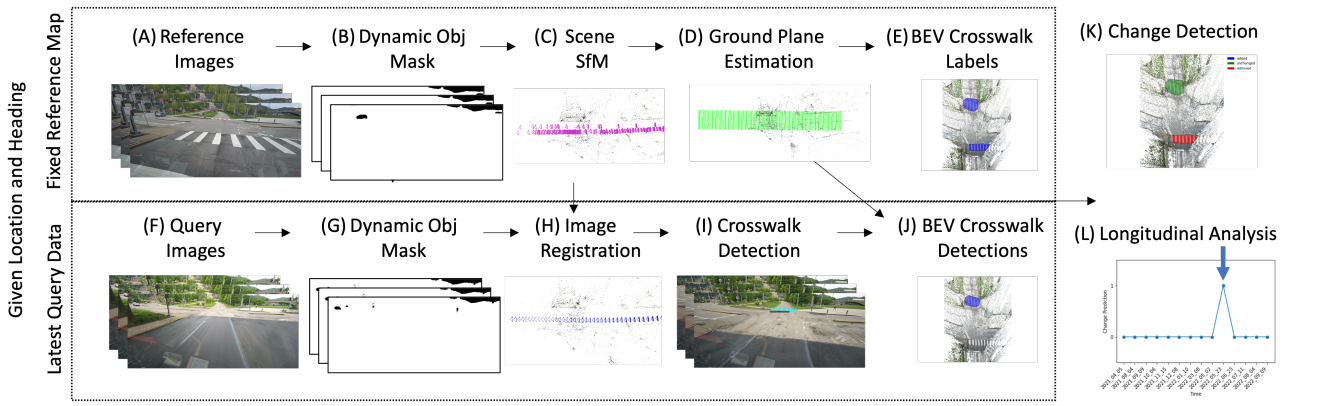

Fig. 2: The pipeline for monocular change detection of crosswalks at intersections. First a reference map is created by collecting several videos of images at a location (A). The scene is reconstructed using structure-from-motion (C), using an off-the-shelf panoptic segmentation model to mask out 2D keypoints from dynamic objects (B). With 3D point clouds of the scene, a plane is fitted to the ground points (D). Crosswalk labels (E) are created for the reference map, and they can either be from crosswalk detection models inferenced on the reference images or hand-labeled in the BEV. Next, any subsequently recorded images (F) can be registered to the reference map (H) by first also masking out 2D features of dynamic objects (G). Then, crosswalks can be detected in each image with an object detection model (I), transformed into the BEV representation (J), and compared against the reference labels. The pipeline produces change predictions (K) at a given time as well as longitudinal information of change (L).

eye view (BEV) analysis of road maps, specifically crosswalks, can be done using only image data through the pipeline shown in Fig. 2. Crosswalks trained and detected by off-the-shelf 2D object detectors can be transformed into BEV; images can be localized using structure-from-motion (SfM); and point clouds of the scene can be obtained by triangulating points from images to fit a ground plane in 3D. The contributions of our paper are as follows:

- a monocular crosswalk change detection framework that aggregates information across image frames and addresses localization and occlusion challenges;
- a new dataset of street view images spanning one year with crosswalk changes to assess change detection consistency and accuracy;
- a crowd-sourced data collection method to monitor crosswalks with daily feedback;
- an extensible system that can be used for other road markings and can be installed on other vehicles.

II. Related Works

A. Online Map Generation

Some recent works attempt to use deep learning networks to generate maps and road semantics on the fly [5], [6]. They both use surround view cameras and predict a BEV of the road semantics directly. This is in contrast to our approach of first learning features and objects in the image space and then transforming them into BEV space through an inverse perspective mapping [7]. Although directly learning the BEV map is ideal, training data for this is particularly limited. As of this writing, NuScenes [8] and Argoverse [9] are the only datasets that contain bird's-eye-view annotations of drivable areas, crosswalks, and lanes paired with sensor data. Because these datasets are only labeled for a few cities, the map generation results are difficult to generalize to other locations and environments. Meanwhile, 2D annotations of objects in images like crosswalks are abundant, diverse, and easier to obtain.

B. 2D Change Detection

Various groups tackle change detection through a learned approach. Sakurada et al. [10] uses aligned omnidirectional images of a scene to perform change detection before and after a natural disaster. They use superpixel segmentations and features encoded by a convolutional neural network for change comparison. Alcantarilla et al. [11] uses deconvolutional networks to perform pixel-wise change detection, training a model that detects relevant structural changes such as construction, building demolition and traffic signs between aligned image pairs. One drawback of [10] and [11] is that they only perform change detection at the image level and do not aggregate change predictions across images, which our work includes to address temporary occlusions of the scene from something like an oncoming vehicle. In addition, in order to generate aligned images, they need a time-consuming dense reconstruction, where the dense point cloud is projected into a virtual camera, an imperfect process that can generate artifacts.

C. 3D Change Detection

Matzen et al. [3] illustrates changes across billboards in Times Square and performs accurate structure-from-motion and grouping of 3D points in time, based on temporal and spatial clustering. They reconstruct the scene from noisy images taken from a large, crowd-sourced internet collection of photos. Our work is similar; however, we provide an application for change detection, and our method contains additional semantic understanding of change events. For example, using their methods, a crosswalk that is removed and repainted at the same location might be considered a change, but ours would not because the underlying meaning on the road stayed the same. An example is shown in Fig. 3.

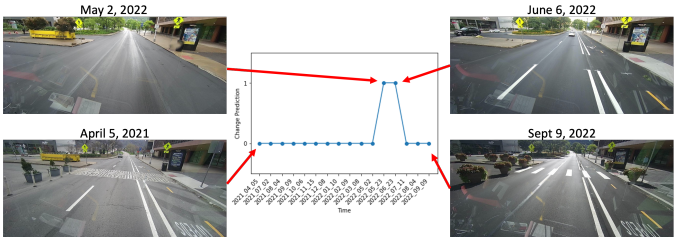

Fig. 3: Change over time at a given location. The bottom left image shows what the scene was in the beginning when the reference map was made, with a crosswalk. The top left shows the scene after the road was repaved before any paint was applied. The top right shows when some road markings were painted in but not the crosswalk. The bottom right shows all road markings painted in, and a crosswalk with a different pattern from the original. Our change detection method (middle) detects the change and also recognizes a crosswalk was painted back at the same location and stops indicating a detected change afterwards.

Lambert et al. [1] compares current sensor inputs and the latest map information to predict if change has occurred. They use LiDAR and RGB data, experiment with ego-view and BEV viewpoints, and make a binary classification of change for a given image. The drawback of this approach is that it relies on an expensive autonomous vehicle's sensor suite for change detection, which would be difficult to crowd-source and relies on precise localization of the vehicle. Another drawback is that the method does not evaluate location of the changes in the sensor data, the number of changes at each time, nor description of the change at each location. In our work, detectors pinpoint locations in BEV where crosswalks have changed, how many changed, and what kind of change occurred, i.e., added or removed. Furthermore, because they train their model end-to-end, their model requires examples of change data, which is hard to obtain and forces them to simulate change. In our work, we rely on the ease of collecting 2D instance segmentations of crosswalks in images to train our model. Lastly, in their detection pipeline, they rely on rasterizing the HD map into an image, but the rasterization process may not represent the real scene correctly. For example, when road markings are partially faded, there is a discrepancy between the reference sensor data and the map which can lead to false positive change detections because the map is an ideal representation of the world and does not capture all edge cases. In contrast, our work uses sensor detections at reference time to compare with current detections; however, we also provide hand-labeled BEV crosswalks as a baseline.

III. Methods

A. Bus-mounted Camera

Data collected in this paper come from a metro commuter bus running between downtown Pittsburgh and Washington, Pennsylvania. A photo of the transit bus is shown in Figure 4. As of this writing, the bus has collected more than a year's worth of data, continues to collect data daily, and is therefore a valuable means to deploy the proposed method for live change detection. The bus makes at least two round-trips every weekday and contains a computing and storage device to carry out preliminary processing of data from its cameras and GPS


Fig. 4: The left four images show the commuter bus, an image of the computer, the cabinet that contains the computer and electronics, and one of the cameras. The right diagram shows the field of view of each installed camera, where one faces forward and four side cameras face opposite directions from each other.

sensor. The computer on the bus is an Intel Core i7-8700t CPU at 2.40GHz with 16 GB RAM. Four waterproof cameras are installed on the exterior four corners of the bus and one in the interior positioned behind the windshield; however, only the front center camera is used in this paper. Two cellular antennas and two dual-band WiFi antennas exist for data transfer.

B. BEV Crosswalk Change Detector

Previous works [10], [11], which used camera sensors, only performed change detection in the image plane. Other works [1], which performed BEV change detection, had the benefit of LiDAR sensors and detailed surface estimates to obtain BEV images. Our work combines the advantages of each to perform BEV change detection with only camera sensors, and there are four components to our method: structure-from-motion, learning-based object detection, ground plane fitting, and BEV comparison.

- Structure-from-Motion (SfM): The COLMAP [12], [13] software is used to perform structure-from-motion, to reconstruct scenes in 3D point clouds and to estimate poses of images taken. Having information of an image's pose provides information as to what is visible in the scene and how to back-project detections or semantic segmentations from the image plane onto a ground plane.
- Learning-Based Object Detection: Models provided by the Detectron2 [14] library were used to create masks for dynamic objects, detect crosswalks, and segment ground points. The panoptic segmentation model [15] trained on MS COCO [16] was used to output masks for dynamic objects like vehicles, pedestrians, and clouds, while also outputting a mask for the ground. A separate Mask-RCNN model [17] with a 50-layer ResNet network [18] and a Feature Pyramid Network [19] backbone was used to detect zebra crosswalks, and was trained on the Mapillary Vistas Dataset [20] and additional Pittsburgh data to detect crosswalks.
- Ground Plane Estimation: A ground plane is fitted by RANSAC to the 3D ground points whose 2D correspondences lie in the ground mask for each image. In some cases, where there is a slope, the ground plane was fitted only in the location where crosswalks existed, by using 3D crosswalk points.

- BEV Comparison and Change Detection: Once crosswalk detections and labels are in the BEV representation, they can be compared against each other by the metric of intersection over union (IoU) to determine if crosswalks are still present or have been added or removed.

Using the above components, the change detection pipeline is as follows. A reference map is generated through SfM using images taken at the time of reference, i.e., the starting time. A large enough collection of images is required to produce an accurate reconstruction. In the case of the bus, this meant using images from the bus' center camera at different times that the bus visited a given location. Because images are taken at different times, dynamic objects (e.g., vehicles, pedestrians, and the sky) appear differently in images in one day versus another. The dynamic object segmentation model creates masks so that features contained inside these areas are ignored. This helps the SfM converge. Once the scene is reconstructed, the ground plane estimation occurs, where the ground segmentation model is used per image to determine which 3D points lie in the ground and can be fitted against. Next, in all of the images the crosswalk object detector is applied, and each instance segmentation of a crosswalk is transformed out of the image plane onto the ground plane through a homography transformation. Each detection is accumulated across frames. Each BEV detection is verified in three or more frames as a multi-frame consistency check to reduce false positives. For detections with an overlapping IoU of 0.1 or greater, non-maximum suppression is applied. This process creates the reference "map."

Subsequent "query" images that are recorded after the reference map is made are registered to the reference reconstruction and localized by matching 2D features and minimizing reprojection errors. In this registration, dynamic objects are similarly masked out. For each image, crosswalks are similarly detected and transformed out of the image plane onto the ground plane. Note that the reference images and query images could undergo the SfM process together; however, creating a pre-computed reconstruction of the scene and a pre-computed ground plane saves time for each query input and ensures that new query images do not influence the geometry of the reference map.

Finally, the change prediction is made by comparing the query detections and the reference detections or labels by their IoU. If there is an IoU of 0.1 or greater, then that reference label is a confirmed "no change". If there is no IoU of 0.1 or greater in either of the reference or query images then a removed or added prediction is made, respectively. Though an IoU of 0.10 is low, it allows for flexibility in the detection location since the road is not exactly a planar surface and this approximation can lead to small inaccuracies in the 2D to 3D mapping.

C. Live Deployment

We follow the works of Gabriel [21], [22] in combining edge computing with servers for low-latency, same-day change results. The data flow follows the same pattern: an edge

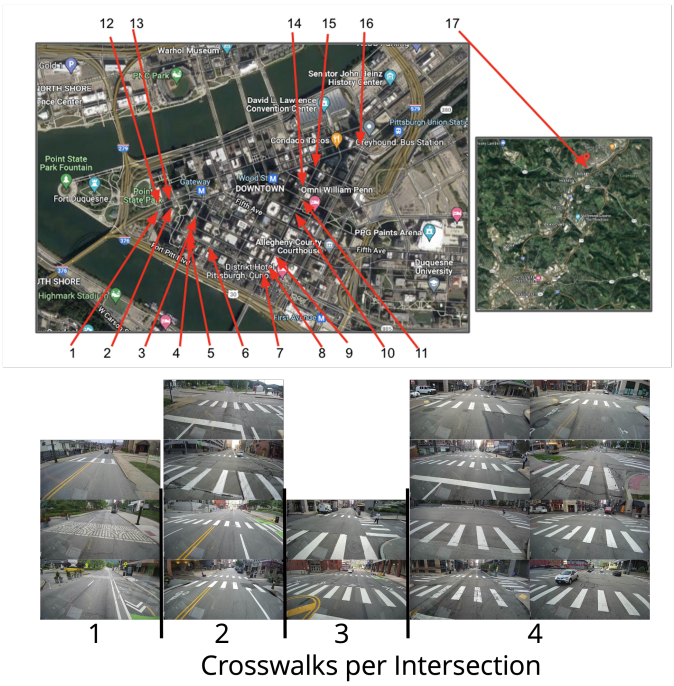

Fig. 5: **Top**: Locations of 17 zebra crosswalk locations regularly observed by the bus. Most occur in downtown Pittsburgh with one occurring in Washington County. **Bottom**: Images of the 17 locations and the diversity of each. They are grouped by the number of crosswalks at each intersection.

computer exists on an edge device where data are collected, filtered, and transferred to a remote server reserved for computationally intensive tasks. First, information is recorded by sensors on the edge device and then processed by a filter on the edge computer, i.e., the bus computer. This filter on the bus finds relevant data and reduces the amount of data needed to be sent to the server, thereby saving network bandwidth. In this paper, the filter on the bus determines images to be saved by its GPS proximity to intersections of interest. Saved images have an intersection label assigned to them. The images are grouped by intersection labels and sent to the server as they are collected as a packet or a video. This packet contains the information of the GPS as well as the heading of the bus when the images were taken. When the packet is received by the server, the corresponding reference SfM reconstruction is retrieved and the images are processed as described in section III-B. The remote server we use is an Intel Core i7-7700 CPU @ 3.60GHz with 32GB Ram and two GeForce GTX 1070 GPUs.

The main idea is that we take advantage of the small computation available on the edge to perform simple tasks like GPS filtering and find relevant images immediately, in contrast to waiting until the end of the day to offload the data and obtain delayed results. Meanwhile, the larger computer on the server can be used to perform complicated tasks that require GPUs and a steady power source like SfM. Although data filtering is simple, it serves an important function of trimming 99% of the bus data, which are often redundant or uninteresting images taken while on highways or at bus depots.

IV. Experiments

A. Data

We analyze the crosswalks at 17 locations along the bus route. These locations contain zebra crosswalks and range from 1 to 4 crosswalks per location, as seen in Fig. 5. 13 locations are seen from two opposed viewing angles, since the bus traverses them in each direction, leading to a total of 30 different SfM maps being created. These different view angles are treated as separate maps and reference images due to the difficulty of matching images with large angle differences using scale-invariant feature transform (SIFT) features [23] in COLMAP. Developing a more robust map at each location so that different maps are not needed for different viewing angles is left for future work. Each reference map uses an average of 183 images in its SfM process, with images taken from the bus on multiple days. The images were taken in May and June 2021. The labels for changes are recorded at an intersection level as well as an individual crosswalk level at each time the query images are taken. We annotate crosswalk-level changes because some crosswalks at an intersection can change while others remain the same. This is not considered in [1], where if any change occurs in the scene, there is only one output value. Having more specificity of the change and the localization of the change is helpful for the vehicle as well as for updating the map.

Offline Dataset: We collect a dataset of bus images from April 2021 to September 2022, covering more than a year's worth of data. Each month has images collected from the bus and assigned to each location and heading angle. However, some days the bus takes a different route and certain angles of a location are not observed, so evaluation at those locations and times are ignored. During this one year of recording, there are 427 query logs, with each map averaging 14 query logs, and each query log averaging 85 images. Query logs are manually sorted into their respective maps, with start and stop frames visually inspected for bad weather and camera issues. While manual intervention is used here in order to remove data outliers and evaluate the change detection pipeline, image sorting is automated in the live deployment where the complete end-to-end system is evaluated. Changes occur in six locations, two of which are seen at different angles. Four locations experienced crosswalk removal due to repaving of the road, where after one or two weeks the crosswalks are painted back, as shown in Fig. 3. One location experienced a change of crosswalk type from a plain crosswalk to a zebra crosswalk. Another location experienced two crosswalk removals for a period of one month before only one crosswalk was repainted in. This dataset is the first of its kind to analyze locations for crosswalk changes over an extended period of time and evaluate the consistency of change predictions given slight changes of the camera and different dynamic objects in view.

Online Dataset: Another way we evaluated the effectiveness of our change predictor was to deploy it live for a month in January 2023 with communication from a bus where images are received and assigned to each intersection by location and angle automatically without human supervision. This important difference tests the robustness of the method in dealing with images that are not visually inspected for their correctness with respect to their reference map assignments. In this online dataset the images are assigned by the GPS proximity filter, which can be erroneous if the GPS is noisy, such as in the downtown region where tall buildings exist. Furthermore, live deployment does not consider the weather and lighting of the images in its collection, which poses additional challenges for the change prediction. There are also other factors, such as different bus routes taken, etc. In total, the online dataset has 1347 logs, with each map having on average of 45 logs and each log having 82 images. Logs from the bus that have unsuccessful image registration are ignored.

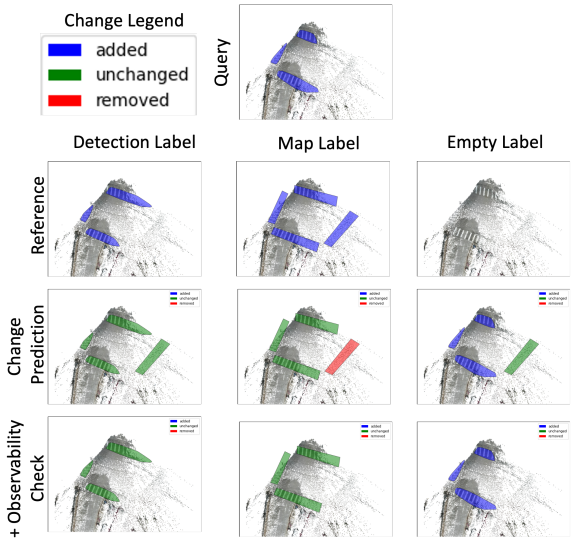

Fig. 6: Different reference labels (detection, map, and empty) by columns and the observability check (last row) are shown. The query (first row) is compared with the reference (second row). These are the results at one intersection with four crosswalks. However, only three are clearly visible. The third row indicates the change predictions for each crosswalk. The left column shows an implied no change for the unobserved crosswalk, the middle column shows a removed crosswalk, and the right column shows a no change. When the observability check is applied, the right crosswalk determination is removed because we identify it as not visible.

B. Additional Checks

When evaluating the change prediction method there are many factors that can affect its result and accuracy. In this paper, we recognize two important factors: crosswalk observability and image localization quality.

Observability Check: For observability, even though we know a crosswalk exists at a location, if the crosswalk is occluded by a vehicle or is not fully seen in an image due to the field of view, such that it makes detection difficult, then that crosswalk prediction can be ignored. This provides some flexibility in our evaluation. This also follows the work of [1], which considers change only for visible crosswalks. Our observability check is determined by projecting crosswalk labels in the reference map into query images and seeing if the entire crosswalk is seen in the image. If the entire crosswalk

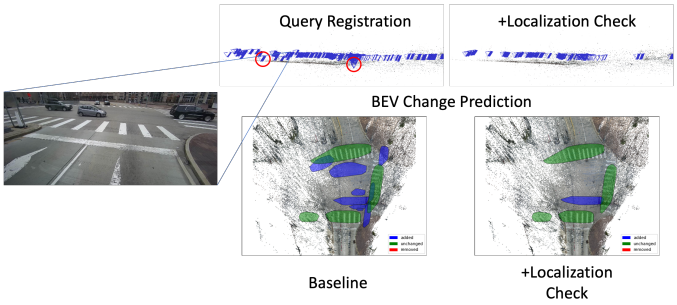

Fig. 7: The top row shows localized query images (blue) in the SfM map. The original registration has several bus images being localized poorly. While the images should have a fixed height with respect to the ground, some images are seen scattered vertically. This creates multiple false positive changes in the bottom left prediction. Many of these false positives can be removed if poorly localized images are removed, circled in red. The bottom right prediction is after the localization check is applied and only one false positive detection remains.

is not visible in three or more frames, we indicate that the crosswalk is not observable. An example is shown in Fig. 6.

Localization Check: Another factor is the localization quality of the images. If image localization is poor, then the change prediction results will also be poor. This is because if there is a misalignment of the detections coming out of the images onto the bird's eye view, a single crosswalk in the real world would appear multiple times in our BEV representation, causing false positive change predictions. Localization quality is determined by measuring the distance between a registered query image and its nearest neighbor reference image. If the difference is too large, that image is removed from the list of query images. Because each intersection uses images from the bus coming from the same angle and trajectory, we can assume that the query images should lie close to the locations of images in the reference map. The localization check is demonstrated in Fig. 7.

C. Hand Labels

Another factor we consider is the label used for the crosswalks in the reference map. [1] uses hand-labeled HD map polygons of crosswalks. This can be problematic if the crosswalk's appearance does not match the hand label. A situation like this can occur, for example, when the crosswalk is faded and is difficult to detect using an object detection model. Therefore, we experiment with hand labels for the reference crosswalks by creating dense reconstructions of each scene and labeling the crosswalks from the bird's-eye view. An example is shown in Fig. 6 as map labels.

D. Pretending an Empty Map

The last experiment that is done is pretending that there is nothing in the map, so that the expected change is for all crosswalks to be newly added crosswalks. This experiment is conducted because the existing changes in the dataset involve mostly removed crosswalks, with few added crosswalks. This imbalance is due to the limited range of the bus route and the bus primarily driving on main roads where necessary crosswalks already exist. However, in theory, our methods

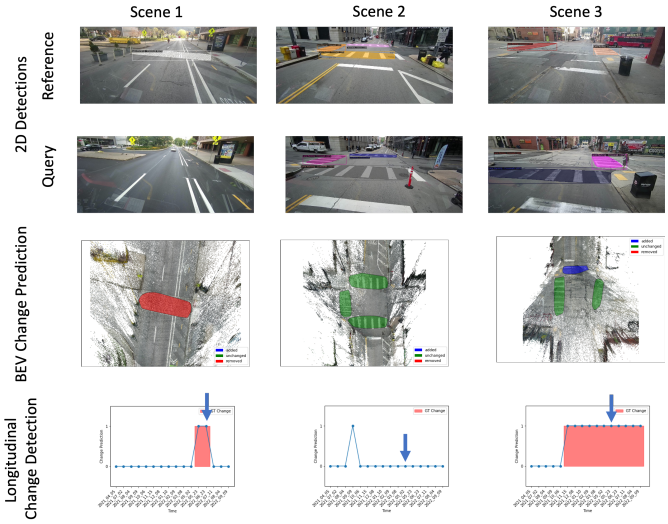

Fig. 8: Three examples of change detection from the offline dataset where reference and query images are shown in the first two rows, the change detection result for that time instance in the third row, and the longitudinal result over one year where the arrow indicates when the third row prediction occurs and the red filling is when the ground truth change occurs. Scene 1 demonstrates a crosswalk being removed. Scene 2 demonstrates no change in the crosswalks; however, there is one instance of a false positive detection. Scene 3 shows one instance of an added zebra crosswalk, which persists.

could be tested for added crosswalks if the reference map was created in the middle of road renovations when the pavement is entirely empty and no road markings have been painted. But instead, we pretend the map is empty and no existing crosswalk labels are yet in the reference map, which can be useful if the map is yet to be annotated.

V. Results

We measure the results in accuracy, F1 score, precision, and recall for the change detection system on the offline and online data at the individual crosswalk level and intersection level. The most extensive experimental results are for the offline data, which include ablation results, using the detections from reference images, hand labels from dense reconstructions, and an empty map.

Offline Dataset: Our results in Table I show that observability and localization checks increase the performance in all metrics. When comparing the use of detection vs. the map as reference labels, the detection labels perform better; however, the localization and observability checks make the performance gap smaller. If we assume the map is empty, we can achieve high change precision, which indicates the method is not overfitted to a low probability of change and indicates our crosswalk detector is very accurate even over the period of a year. We also look at the intersection level change prediction and see similar patterns. It is useful to note that despite a large imbalance of no change occurring, high precision and F1 scores can be achieved. Fig. 8 shows qualitative change detection results of three intersections and the longitutinal change detection over the year.

Online Dataset: We also show the results using the best performing method from the offline dataset, using the

TABLE I: Change detection results for the **offline** dataset. The query predictions are compared against different reference labels: detections from the reference images (det), hand drawn map labels (map), and no labels (empty). Ablation for the observability (obs) and localization (loc) checks are also shown.

ref	+obs	+loc	precision	accuracy	recall	F1
Individual Crosswalk Performance						
det			0.19	0.86	0.98	0.32
det	x		0.27	0.91	0.98	0.42
det		x	0.23	0.89	0.98	0.37
det	x	x	**0.38**	**0.94**	0.98	**0.54**
map			0.16	0.83	0.98	0.27
map	x		0.26	0.90	0.98	0.41
map		x	0.18	0.85	0.98	0.30
map	x	x	0.36	0.94	0.98	0.52
emp			0.94	0.84	0.88	0.91
emp	x		0.94	0.91	0.97	0.96
emp		x	0.98	0.86	0.87	0.92
emp	x	x	**0.98**	**0.95**	0.97	**0.97**
Intersection Performance						
det			0.25	0.75	1.00	0.40
det	x		0.36	0.85	0.94	0.52
det		x	0.25	0.75	1.00	0.40
det	x	x	**0.37**	**0.86**	0.94	**0.53**
map			0.22	0.70	1.00	0.36
map	x		0.34	0.84	0.94	0.50
map		x	0.22	0.70	1.00	0.36
map	x	x	0.35	0.85	0.94	0.51
emp			1.00	0.98	0.98	0.99
emp	x		1.00	0.98	0.98	0.99
emp		x	1.00	0.98	0.98	0.99
emp	x	x	1.00	0.98	0.98	0.99

TABLE II: Change detection results for the **online** dataset.

ref	+obs	+loc	precision	accuracy	recall	F1
Individual Crosswalk Performance						
det	x	x	0.22	0.85	0.98	0.35
Intersection Performance						
det	x	x	0.28	0.66	0.84	0.42

sensor detections as labels, in the live deployment and show accuracy, precision, recall and F1 scores of predictions in Table II. We observe a performance gap between the online and offline in all metrics, for example a 0.11 difference in the F1 score for intersection level performance. We also track the accuracy over time to indicate the consistency of the change prediction results in Fig. 9. However, we do notice that there are fluctuations, which indicates that there are environment factors that can affect the change prediction even at the same location. This can be a form of anomaly detection in the environment or data, because in the plot there is a large dip in accuracy on the last day and this is due to large snow precipitation from the previous night, which might have affected detection performance.

Influence of Weather: We explore the effect of weather on the change detection performance using the online dataset shown in Fig. 10. For the hour that the query images are taken, the metadata of the weather is also recorded. We explore the change detection accuracy as well as the rate of success of SfM in registering the new images. For change detection accuracy, there is a drop in performance on three notable occasions, clear and foggy days and night time. Performance difficulty at night is reasonable because darkness can hinder detection

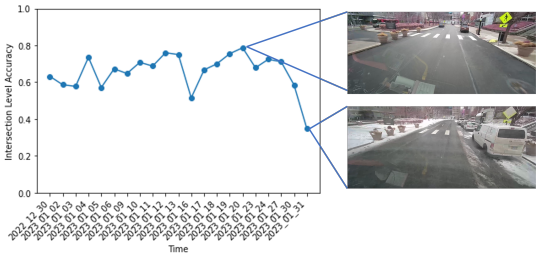

Fig. 9: Accuracy of the method over all intersections on a given day through the online dataset. Fluctuations exist in accuracy due to external factors such as weather, dynamic objects, etc. Example images from the best and worst performing days are shown on the right.

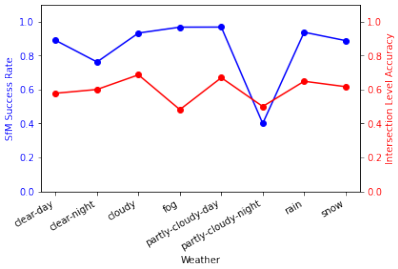

Fig. 10: Accuracy of the method and success rate of query image localization over different weather conditions in the online dataset. Drops in performance for both metrics occur at night. Additionally, change detection performance decreases on clear and foggy days.

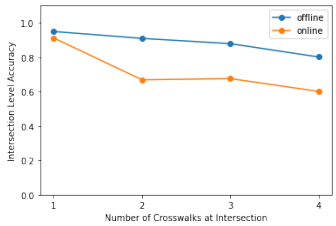

Fig. 11: Accuracy of the method over intersections with different number of crosswalks on the online and offline datasets. Decrease in performance as crosswalk numbers increase and a larger performance gap between the online and offline datasets.

performance and change the lighting of the environment. Surprisingly then is that a clear day also results in lower performance. The reason, however, is that on clear days, there are more cases of shadows, sun glare, image overexposure, which can also hinder detection. Foggy weather will limit visibility and also hinder detections. We also look at the registration success rates of query images and see similar trends, except clear days see an improvement while night time imagery remains low. Another surprise is how there is little performance decrease on snow weather, but it may be dependent on the amount of precipitation that occurs. Snow days have diffuse light like cloudy days which is beneficial for localization and crosswalk detection, but when snow begins to cover the ground, it will affect performance. Therefore, this analysis shows that it is best to perform change detection during the day time, with little precipitation and with diffuse light.

Influence of Number of Crosswalks: We explore the effect of the type of intersection on the change detection

result as shown in Fig. 11. Here we see analysis on both the online and offline datasets where accuracy is plotted against the number of crosswalks at the intersection. There is a downward trend of accuracy as more crosswalks exist, with a performance gap between the offline and online change detection for intersections with more than one crosswalk. This drop in performance is likely because intersections with more crosswalks tend to be larger, having more crosswalks to account for, and more traffic that can disrupt the normal change detection process.

Latency: Since the purpose is for high frequency monitoring of HD maps, the general time consumption and latency for the whole process is important to consider. The cellular internet bandwidth of the bus of 200KB-1MB/s allows for an image transfer rate of 2-10 images/s with an image size of 151KB/image. A single intersection with 80 images can, therefore, take 8-40s to be received on the server. The server analysis takes 107 ± 52s per intersection. Thus, the total processing time of 4 mins per intersection is suitable for daily map status updates; however, there is an upper bound of the number of intersections that can be monitored given the slight latency.

VI. Conclusion and Future Work

This paper demonstrates a way to obtain accurate crosswalk change detections from a low-cost sensor suite of a front camera and a GPS on a bus. We show how images contain enough information to perform complex scene analysis in 2D and 3D and are more easily crowd-sourced compared to other sensor modalities. The system performs with 86% and 66% accuracy of change detection at the intersection level on the offline dataset and online dataset, respectively, and 94% and 85% at the crosswalk level, respectively. Though not perfect, our methods can be used to feed suggestions to map maintainers who can verify if a change has occurred. This would improve their workflow and save their time in searching for changes. The paper improves on past work that used map labels as reference and addresses sources of error such as observability and localization issues when using a monocular camera approach. This paper, furthermore, shows that buses and other public vehicles are valuable sources of data for road monitoring and present an alternative solution to designated mapping fleets.

In the future, it is important to consider the plain crosswalk, which is currently excluded in this paper. The plain crosswalk is defined by only two lines and can easily be confused with lane dividers by modern object detectors. Furthermore, other map changes exist such as lane changes, added bike lanes, or lane shifting. These are additional detection tasks that should be incorporated and considered for map updates. Lastly, the number of intersections this paper addresses is limited and covering a larger area should be considered.

VII. Acknowledgment

We thank Canbo Ye, William Pridgen, Anurag Ghosh, and Khiem Vuong for their useful feedback and assistance.

References

[1] J. Lambert and J. Hays, "Trust, but verify: Cross-modality fusion for HD map change detection," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[2] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, "Building rome in a day," in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 72–79.

[3] K. Matzen and N. Snavely, "Scene chronology," in *Proc. European Conf. on Computer Vision*, 2014.

[4] S. M.S., H. Grimmett, L. Platinský, and P. Ondrúška, "Visual vehicle tracking through noise and occlusions using crowd-sourced maps," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4531–4538.

[5] Q. Li, Y. Wang, Y. Wang, and H. Zhao, "Hdmapnet: An online hd map construction and evaluation framework," 2021.

[6] T. Roddick and R. Cipolla, "Predicting semantic map representations from images using pyramid occupancy networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[7] S. A. Abbas and A. Zisserman, "A geometric approach to obtain a bird's eye view from an image," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE, 2019, pp. 4095–4104.

[8] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *CVPR*, 2020.

[9] M.-F. Chang, J. W. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, "Argoverse: 3d tracking and forecasting with rich maps," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[10] K. Sakurada and T. Okatani, "Change detection from a street image pair using cnn features and superpixel segmentation," in *BMVC*, 2015.

[11] P. F. Alcantarilla and S. Stent, "Street-View Change Detection with Deconvolutional Networks," p. 10.

[12] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision (ECCV)*, 2016.

[13] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[15] A. Kirillov, K. He, R. B. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," *CoRR*, vol. abs/1801.00868, 2018.

[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and L. Zitnick, "Microsoft COCO: Common Objects in Context," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Sept. 2014.

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[19] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.

[20] G. Neuhold, T. Ollmann, S. Rota Bulò, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *International Conference on Computer Vision (ICCV)*, 2017.

[21] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards wearable cognitive assistance," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 68–81.

[22] C. Ye, "Busedge: Efficient live video analytics for transit buses via edge computing," Master's thesis, Carnegie Mellon University, Pittsburgh, PA, July 2021.

[23] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157 vol.2.