The World is Changing: Finding Changes on the Street

Kuan-Ting Chen, Fu-En Wang, Juan-Ting Lin, Fu-Hsiang Chan, and Min Sun

Department of Electrical Engineering, National Tsing Hua University
{winterdaphne104,tdk356ubuntu,brade31919,corgi1205}@gmail.com,
sunmin@ee.nthu.edu.tw

Abstract. We propose to find changes in the constantly changing world, given visual observations at street-level. In particular, we identify "long-term" changes between Google Street View images and dashcam videos captured at different months or even years. This is a challenging task, since (1) dashcam frames are not localized in world coordinate, and (2) there are many changes introduced by moving objects. We propose a robust sequence alignment method to align dashcam sequence to Street View images. Our method outperforms a strong baseline method [1] by 12% mean Average Precision (AP). We also propose a novel change detection method designed to detect long-term changes. Our change detection method (13.54%) outperforms a baseline method without handling car interior and moving objects (11.70%) by 1.84% (relatively 13.6%) in mean AP. In a controlled experiment, given manually aligned high quality Street View images, our change detection method achieves a significantly better mean AP (45.57%).

1 Introduction

Geotagged image collections, such as Google Street View, allow users to experience the street scene of a remote location. This service allows users to find stores, offices, etc. before physically visiting a place. Google Street View is gleaned by a fleet of vehicles equipped with expensive cameras and sensors. This approach can cover many cities at a specific moment in time (e.g., panorama captured on Nov., 2012 in Fig. 1). However, the fleet is not large enough to keep up with the changes in the real world. For instance, there is a new building and two billboard ads in the image captured by a dashcam on Feb., 2014 in Fig. 1.

Another popular system which also captures the street scene is a dashboard camera (later referred to as dashcam). A dashcam is a consumer device which can be easily installed by users themselves. Due to its low cost and ease of use, it has become popular in many countries like Taiwan, Russia, etc. As a result, the number of vehicles with dashcam is significantly larger than the size of the Google Street View fleet. Theoretically, it is possible that most changes in a street scene will be recorded by a dashcam. This implies that fusing the information from dashcam and Google Street View could be mutually beneficial.

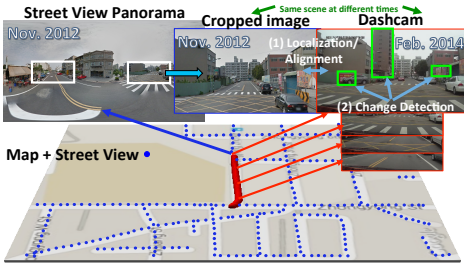


Fig. 1: Illustration of our system. We compare Street View images (taken on Nov., 2012) with dashcam videos (taken on Feb., 2014) to identify changes: a new building and two billboard ads (green boxes). Our system first (1) localizes the dashcam video on the map, then (2) identifies changed rectangle regions.

We propose to identify "long-term" changes between Google Street View images and dashcam videos captured at different months or even years (Fig.1). This is a challenging task, since (1) dashcam frames are not precisely localized in world coordinate, and (2) there are many changes introduced by moving objects. To achieve this, we first propose a robust sequence alignment method to localize dashcam frames on the coordinate of Street View images. Our method needs to handle significant structural changes in the scene accumulated through months or years, whereas most localization methods (such as [2]) assume that the changes are mainly due to lighting condition and slight viewpoint changes. We address the challenge by utilizing reliable matches from frames with less changes to geometrically align the whole video sequence to the world coordinate. Given pairs of Street View images and frames, a novel change detection method is proposed to predict a number of rectangle changed regions, which likely correspond to a new building, a billboard ads, etc. (Fig. 1). The change detection method is explicitly designed to ignore (1)"short-term" changes introduced by moving objects (e.g., cars) and different weather conditions (Fig. 6(a)), and (2) changes corresponding to the interior of a vehicle (Fig. 6(a)-Right).

Our system aims to handle dashcam videos in the wild. Therefore, we harvest dashcam videos captured at a diverse set of location (across a radius of 155 km) and time (across about three years) for evaluating our system. Our sequence alignment method is shown to outperform a baseline method [1] by 12% in mean AP. Moreover, our change detection method, which considers the properties of dashcam videos, is also 1.83% (relatively 13.6%) better in mean AP than a generic baseline method based on dense SIFT matching [3]. By inspecting failure examples, we found that failures are typically due to severe viewpoint difference between Street View images and dashcam frames, or less ideal dashcam frame quality. Hence, we further conduct a controlled experiment using manually aligned high quality Street View images. In the control experiment, our change detection method achieves a significantly better mean AP (45.57%).

2 Related Work

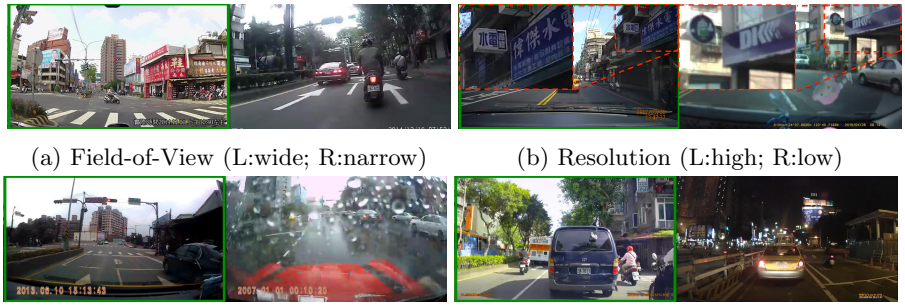
In Computer Vision, visual localization of images has been widely studied. We summarize the related work in three groups.

Landmark localization. [4,5] are early work showing the ability to match query images to a set of reference images. Schindler et al. [6] improve the performance to

handle city-scale localization. Hays and Efros [7] further demonstrate that query images can be matched to a collection of 6 million GPS-tagged images dataset (within 200km) at a global scale. Zamir and Shah [8] propose to use Street View images as reference images with GPS-tags, and match SIFT keypoints in a query image efficiently to SIFT keypoints in reference images by using a tree structure. A voting scheme is also introduced to jointly localize a set of nearby query images (within 300 meters). As a result of voting, it is able to outperform [6]. Vaca-Castano et al. [1] propose to estimate the trajectory of a moving camera in the longitude and latitude coordinate using Bayesian filtering to incorporate the map topology information. Similar to [8,1], we also use Street View images as reference images. However, unlike [8], we assume a video sequence is captured across an arbitrary distance (not restricted to within 300 meters). Unlike [1], we use not only the topology of the map, but also the relative 3D position of the dashcam frames to improve the localization accuracy. Cao and Snavely [9] propose to match a query image to reference images with a graph-based structure to reliably retrieve a sub-group of images corresponding to a representative landmark. This method can be used to improve single image matching accuracy at the first stage of our method. Bettadapura et al. [10] also use Street View images as reference images to match the point-of-view images captured by a cellphone camera. Moreover, the method utilizes accelerometers, gyroscopes and compasses on the cellphone to improve the matching accuracy. The final estimated point-of-view is used as an approximation of the users' attention in applications such as egocentric video tours at museums.

Localization from point clouds. Accurate image-based camera 6 Degrees of Freedom (DoF) pose estimation can also be achieved by utilizing 3D point clouds of a scene [11,12,13,14]. However, it is challenging to obtain 3D point clouds representation given a sparse dataset like Google Street view. Moreover, for change detection, 6DoF pose estimation is not required. Our method essentially estimates a 2D rigid transformation to align dashcam with Street View, since dashcam videos typically have common pitch and roll angles (i.e., parallel to the ground plane), a restricted yaw angle (i.e., tangent to the vehicle trajectory), and a fixed height.

Vehicle Localization. For egocentric vehicle localization, many methods combining image-based sensors with other sensors or map information have been proposed. Given the vehicle speed at all time and the rough initial vehicle location, Badino et al. [2] use Bayesian filter and a per-frame-based visual feature to align a current frame to pre-recorded frames in the database while considering the candidate location of previous frames. Taneja et al. [15] propose a similar but lightweight method requiring images sparsely sampled in space (in average every 7 meters). Lategahn et al. [16] combine frame-based visual matching and Inertial Measurement Unit (IMU) for localization. However, it also assumes the GPS information of the first frame is also given. Both [17,18] utilize relative position information from visual odometry with map information from OpenStreetMaps to globally localize a vehicle. However, these methods require the vehicle trajectory to be complex enough to be uniquely identified on the map



(a) Field-of-View (L:wide; R:narrow) (b) Resolution (L:high; R:low)

(c) Weather Condition (L:sunny; R:rainy) (d) Lighting Condition (L:day; R:night)

Fig. 2: Examples of dashcam videos in the wild. We focus on harvesting videos with good quality (highlighted by green bounding boxes).

(i.e., many turns, etc.). Dashcam videos on YouTube do not come with additional speed or initial location information. Moreover, a video is typically less than 5 minutes with simple trajectories. In contrast, our method does not require an initial location or any extract sensor, and it can even localize vehicles with simple trajectories using both map information and Street View images.

Our main application for dashcam localization is to detect changes between Street View images and dashcam frames. We summarize the related work of change detection below.

Change Detection. Many works have been proposed to compare images of a scene captured at different times [19]. However, their results are typically pixel-wise change map, which is often noisy. When additional information of the scene is given, different change detection methods have been proposed. Pollard and Mundy [20] propose a change detection method for 3D scenes observed from an aerial vehicle. Taneja et al. [21] propose geometric change detection by comparing 3D reconstructions built from videos. Recently, Matzen and Snavely [22] propose to detect rectangle regions corresponding to changed billboard and Graffiti from a large number of internet images. Their results are impressive, since they utilize the 3D structure of the scene and assume images are captured densely in time. In our case, the task is much more challenging, since neither the 3D reconstruction of the Street View scene is available nor the dashcam videos is captured densely in time at the same location. In fact, we typically match images captured at different years.

3 Our methods

We first describe how we harvest many dashcam videos in the wild. Then, we mention how to obtain relative camera position in dashcam video, and how to query Street View images as reference images for localizing cameras in world coordinate (i.e., longitude and latitude). Next, we introduce our robust sequence alignment method. Finally, we describe our novel change detection approach which identifies changed rectangle regions by considering the properties of dashcam videos.

3.1 Data Harvesting

In countries like Taiwan, Russian, etc., dashcam are commonly installed on cars. Through crawling YouTube, we can locate many dashcam videos with rough longitude and latitude positions annotated by the users who uploaded the videos. These videos are extremely diverse. However, not all of them are equally suitable for applying computer vision algorithm due to the following reasons:

Hardware Spec.

- *Field of View.* The field of view of the cameras could range from 70 degrees narrow angle to 200 degrees wide angle (Fig. 2a).
- *Resolution.* The resolution of the videos could range from low resolution such as 640x480 pixels to high resolution such as 1280x720 pixels (Fig. 2b).

Environmental Condition.

- *Weather Condition.* Dashcam videos recorded in raining and foggy weather have very low quality (Fig. 2c).
- *Lighting Condition.* There are also many videos recorded in low-light condition at night or in tunnels (Fig. 2d).

In order to harvest dashcam videos suitable for camera localization and change detection, we develop several semi-automatic methods to efficiently identify different conditions. Low resolution and small horizontal field-of-view videos can be approximately identified by checking the image dimension and aspect ratio (i.e., a large horizontal field-of-view video typically has width much larger than height), respectively. Low-light condition videos can be removed according to the color histogram with modest precision and high recall. As a result, we efficiently select many videos with good lighting condition and at least 1280x720 pixels (later referred to as high quality videos). Nevertheless, our selected videos are not guaranteed to have high signal to noise ratio or low image distortion. Hence, we still need to address these challenges while developing our method. The videos will be publicly available once the paper is published.

3.2 Relative Camera Position from Dashcam

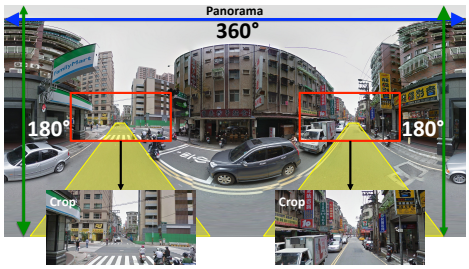
A dashcam video can be interpreted as sequential images captured by a moving camera. Hence, we could apply classical sequential Structure-from-Motion (SfM) techniques to obtain relative camera positions with respect to the first frame. However, our problem is not a classical Visual Odometry (VO) problem, since we do not have the camera intrinsic information of the harvested dashcam video. Moreover, most of the dashcam videos are captured by a wide-angle or even a fish-eye camera, which introduces an unknown camera distortion in raw video frames. Therefore, it is critical to estimate the camera intrinsic including parameters to mitigate the distortion. We applied an efficient sequential Structure-from-Motion (SfM) method [23,24] to jointly estimate the camera intrinsic and extrinsic, and sparse 3D point cloud representation of the scene. The method estimates the first-order radial distortion parameter which significantly improves the quality of the camera's 3D trajectory reconstruction compared to the reconstruction from camera assuming zero distortion (Fig. 3). For consistency, we refer the coordinate system of the reconstruction as the "model coordinate" in the following sections.

(a) Bird-eye's View

(b) Side View

Fig. 3: Comparison between 3D trajectory of the camera with and without distortion estimation from bird-eye's and side view in panal (a) and (b), respectively. Note that without distortion estimation, the estimated 3D trajectory is disconnected and erroneously scattered (large errors indicated by red bounding boxes).

Fig. 4: Illustration of the queried Street View panoramic image with a 180° vertical field-of-view and a 360° horizontal field-of-view. The road information is illustrated by overlaid yellow paths. We crop two images (later referred to as reference images) along the road direction indicated by the red rectangles.

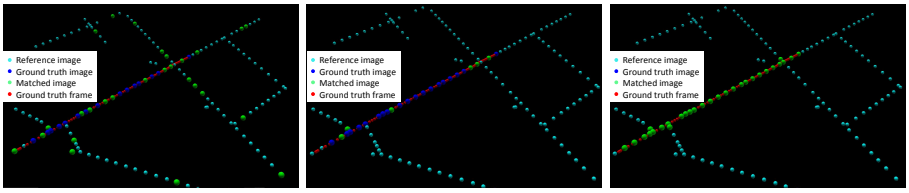


3.3 Reference Images from Street View

Similar to [8], we use Street View images as reference images for localizing the path of a dashcam video. For each dashcam video, we query Street View images within a rectangle region centered at the user provided rough longitude and latitude position[1].

Cropping Street View Images. Note that raw Street View images are panoramic images with a 180° vertical field-of-view and a 360° horizontal field-of-view. In order to efficiently match Street View images to dashcam videos, we propose to crop a panoramic image so that the cropped image is likely to be similar to a frame in the dashcam video. Unlike [8] which cannot assume a common camera orientation, we can assume that the orientation of a dashcam camera is always parallel to the road direction. Hence, we simply crop two images from Street View panorama where their orientations are aligned with the road direction (i.e., one forward and one backward). For the field-of-view of the crop, we simply crop a images with 36% vertical field of view and 72% horizontal field of view (i.e., one fifth of the field of view of a panoramic image). We treat these

[1] It is very common that the rough GPS location of a dashcam video is described in the video description for the purpose of reporting accident.



(a) Baseline Image matches (b) Inliers of Alignment (c) Our Full Method

Fig. 5: Illustration of the baseline image matching (Panel (a)), and our sequence alignment (Panel (b,c)) approach. Note that our matched images are well aligned to the ground truth trajectory (Panel (c)), but the baseline matched images are spreading on the map with many outliers (Panel (a)).

cropped images as "reference images" which will be matched to the frames in the dashcam video (Fig. 4). The matching results will let us retrieve a subset of reference images along the path of the dashcam video.

3.4 Dashcam Global Localization

Given the reference images and extracted frames in a dashcam video, we first use a baseline image retrieval approach with geometric verification (similar to [6]) to establish matches between them. Then, we propose a new sequence alignment method to align the reconstructed dashcam path to the locations of the reference images in longitude and latitude coordinate.

Image Matching In the following, we describe our image retrieval pipeline to establish baseline image matches.

Holistic Features Representation. For both the reference images and frames, we detect sparse SIFT keypoints and represent each keypoint using SIFT descriptor [25]. The number of keypoints in each images can be very different depending on the texture in the scene and the image quality. Typically, there are more keypoints in the reference images than in the video frames. In order to represent all images with a fixed feature dimension, we use fisher vector encoding [26] to generate our holistic feature representation for all images. It has been shown in [26] that the fisher vector representation can be used to efficiently and accurately retrieve visually similar pairs of reference images and frames.

Geometric Verification. The similarity between a pair of reference image and frame can be more reliably confirmed by applying geometric verification method consisting of (1) raw SIFT keypoint matches with ratio test (i.e., the ratio between the distance of the top match and the distance of the second top match), and (2) RANSAC matching with Epipolar geometric verification. However, geometric verification is more computational expensive than calculating similarity using holistic representation. Therefore, we limit the number of geometric verification in our pipeline as described below.

Candidate Reference Images along the Path. In our application, we assume the reference images densely cover a square region on the map containing the path of the dashcam. Therefore, only a small set of reference images is along the path. We aim to find a set of "candidate reference images" along the path.

For each frame, we first use its holistic feature to retrieve the top K similar reference images by considering cosine similarity. Then, we apply geometric verification only on the retrieved images to re-order them according to the number of inlier matches which is referred to as the "confidence score". The reference image with the largest confidence score is considered as the candidate reference image. At this point, we obtain many pairs of frames and their corresponding candidate reference images (green dots in Fig. 5a).

The image matching approach has two drawbacks:

- Candidate reference images might miss some truth reference images along the path (i.e., low recall).
- The relative location of candidate reference images can be inconsistent to the relative location of dashcam "sequence" (i.e., low precision).

Hence, we propose to utilize sequence information to increase (i) the precision by removing inconsistent pairs, and (ii) the recall by introducing more candidate reference images.

Sequence Alignment Our sequence alignment method utilizes the following information:

- The relative locations of frames in the model 3D coordinate (as described in Sec. 3.2).
- The relative locations of reference images in the world 2D coordinate (i.e., longitude and latitude).

We aim at estimating the transformation between these two coordinates, given the noisy pairs of candidate reference image and frame. We reduce the model 3D coordinate to a 2D coordinate using Principle Component Analysis (PCA), since dashcam trajectory can be well approximated in a 2D coordinate system. Hence, we estimate a 2D rigid transformation (i.e., rotation, translation, and scale) between reduced model 2D coordinate and world 2D coordinate.

We propose a modified RANSAC to estimate the rigid transformation as follow.

- Guided sampling. We sample a pair with probability proportional to $\sqrt[n]{score}$, where $score$ is the confidence score of a pair of a reference image and frame, and n is the order of the root. In this way, pairs with higher confidence will be sampled more often.
- Transformation Selection. Instead of using the number of inliers to select the Transformation, we use the sum of confidence scores of the inliers. In this way, transformation that preserves pairs with high confidence as inliers will be selected with a higher chance.

The estimated inliers (green dots in Fig. 5b) typically exclude incorrect pairs which are not consistent with the best alignment result. Hence, the precision typically increase while recall is still low.

Increasing Recall. Given the rigid transformation between two coordinate systems, we can transform all frames into the world coordinate. For any unmatched reference image which contains at least one aligned frame within Q meters, we

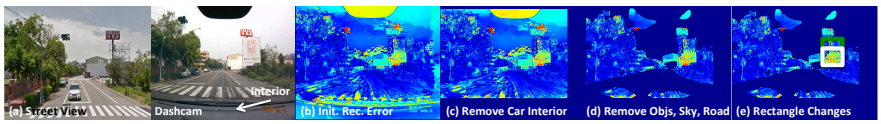


Fig. 6: Pipeline of change detection. Panel (a) shows a pair of Street View image (Left) and dashcam frame (Right). Panel (b) shows the Initial reconstruction error. Panel (c) shows the error after removing car interior. Panel (d) shows the error after removing moving objects, sky, and road segments. Panel (e) shows the detected rectangle changed region (white-box) and ground truth changed region (green-box).

propose it as a new candidate reference image. In this way, we can recall much more candidate reference images. For each new candidate reference image, we retrieve the top K closest aligned frames and apply "geometric verification" to find the best matched frame. At the end, we retrieve much more pairs, while maintaining the precision (green dots in Fig. 5c).

Quality Prediction. In a few rare cases, our method slightly decrease the performance. We propose the following criteria to automatically decide whether to use our results or not. We use our results only if (i) the ratio between number of inlier over the total number of pairs is above a threshold λ; or (ii) the ratio between the average of the confidence scores of our method over the average of the confidence scores of the baseline method is higher than a threshold γ. When our method is not used, we have the same results as the baseline method (see dots along the diagonal line in Fig. 7).

3.5 Change Detection

Many dashcam videos are captured at different months or years with respect to the reference images. This implies that the street scene might have significant changes. Change detection gives us a potential way to automatically update Street View images by processing many dashcam videos. The global localization method described in Sec. 3.4 has simplified the task of detecting changes between a set of images (Street View) and a video (dashcam) into detecting changes in pairs of matched images (i.e., cropped Street View images and dashcam frames). However, there are more challenges as follows,

- Street View images and dashcam videos are typically captured from notably different viewpoints (Fig. 6(a)).
- Dashcam videos typically capture the interior of the car (Fig. 6(a)-Right).
- Changes often are introduced by many moving objects (e.g., cars) and different weather conditions (Fig. 6(a)).

Our method aims to overcome these challenges and detect changed rectangle regions by applying the following pipeline.

Robust Pixel-wise Matches. To overcome the differences in viewpoints, we apply an efficient dense SIFT matching method [3] to find the best match of every dashcam pixel in the Street View image. By using the matched Street View pixels to reconstruct a dashcam frame, we can compute the pixel-wise reconstruction error (Fig. 6(b)). A high reconstruction error implies a potential change between the dashcam frame and the Street View image.

Identify Car Interior. Since the car interior typically differs from the Street View scene, we apply the same method to find the reverse match (i.e., best match of every Street View pixel in the dashcam frame.). Then, we consider every unmatched dashcam pixel as car interior and assign zero reconstruction error for these dashcam pixels (Fig. 6(c)).
Moving Objects, Sky, and Road. We further use state-of-the-art semantic segmentation methods [27,28] to estimate foreground (e.g., car, motorbike, people, etc.), sky and road segments, respectively. We assign zero reconstruction error for these segments in the dashcam frame (Fig. 6(d)). Note that we give special treatment for sky and road, since the appearance of sky and road changes significantly in different weather conditions.
Changed Rectangle Detection. Since structural changes in street scene typically corresponds to a new billboard, a new building, etc. We use candidate object hypotheses [29] as change candidates. In order to rank each candidate, we first convert reconstruction error into binary change map using the $T\%$ percentile of the whole reconstruction error as threshold (i.e., -1 for pixels with error $<$ threshold(T) and $+1$ for pixels with error $\geq$ threshold(T)). Then, we rank the candidates according to the accumulated values of the binary change map within a candidate object hypothesis (white-box in Fig. 6(e)).

3.6 Implementation Detail

Localization. We set K to 15, since the performance gain for $K > 15$ is not significant. We set Q to 10 meters which is twice the 5 meters error metric (Sec. 4) in order to recall most of the missing candidate reference images. We set $n = 3, \lambda = 0.2, \gamma = 0.4$ and we show in Sec. 4 that the performance is not sensitive to these three parameters.
Change Detection. For our method, we set $T = 90$ and obtain reasonable results as in Fig. 6(e).

4 Experiment Results

We evaluate our proposed dashcam localization and change detection methods on a subset of our harvest data in the wild with manually labeled ground truth locations and changes.

4.1 Dashcam Localization

In order to quantitatively evaluate our method's performance, we select 45 high quality videos captured as early as on Jun, 2012 and as late as on May 2015. For each video, we label the best matched frame corresponding to each reference image on the path of the dashcam.
Evaluation Metric. Given a few manually labeled pairs of reference images and matched frames, we align all frames to the longitude and latitude coordinate using alignment method described in Sec. 3.4. In this way, each frame is assigned with a longitude and latitude coordinate, which is treated as its ground truth longitude and latitude. We also obtain the ground truth reference images

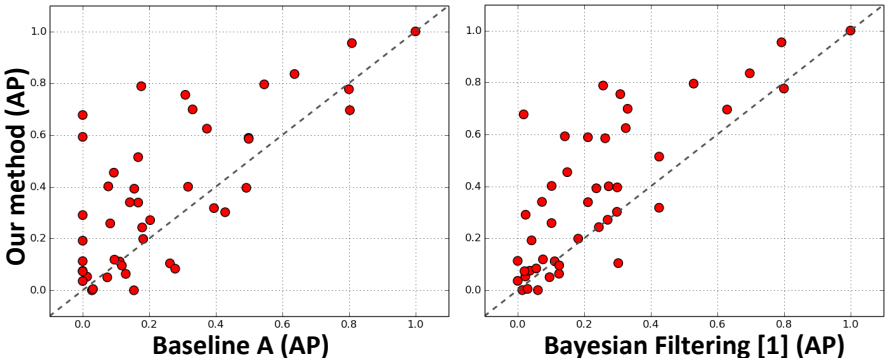


Fig. 7: Average Precision (AP) comparison between our method and baseline methods: a baseline image matching (Baseline A) and a baseline Bayesian filtering [1].

along the path of dashcam. Note that longitude and latitude coordinate can be converted to metric units such as meters. Now, given a predicted pair of reference image and frame, we can calculate the distance in meters between the predicted reference image and the ground truth location of the predicted frame. We further evaluate the precision and recall of the predicted pairs as follows. We consider a predicted pair is correct if the following two conditions are both true: (1) the predicted reference image is in the set of ground truth reference images; (2) the predicted reference location is within 5 meters from the ground truth frame location. The precision is the percentage of correctly predicted pairs among all the predicted pairs. The recall is the percentage of ground true reference images which are retrieved by the correct prediction. Note that we only allow each reference image to exist once in all predicted pairs so that there won't be multiple predicted reference images corresponding to the same frame. Moreover, since each predicted pair is associated to a confidence score, we can remove pairs with scores below a threshold to control the number of predicted pairs. In this way, we can calculate a precision v.s. recall curve for each video sequence. The mean Average Precision (mAP) over all videos is used for comparison.

Sequence Alignment v.s. Baseline Methods. Fig. 7 shows two scatter plots of AP comparison between our sequence alignment method (y-axis) and two baseline methods (x-axis): baseline image matching (Baseline A) and baseline Bayesian filtering, respectively. Our method achieves better AP for most of the videos (i.e., dots scatter in upper-left triangle). Note that the baseline Bayesian filtering is our implementation of [1]. Our proposed alignment method (37% mAP) significantly outperforms the baseline A (23.4% mAP) and Bayesian filtering (24.7% mAP). Qualitative results of our method in Fig. 8 shows that we can reliably align dashcam locations onto longitude and latitude coordinate. The 2nd row example in Fig. 8 even shows that our method can handle Street View images captured in bad weather condition.

Sensitivity Analysis. There are three main parameters involved in our system: the n^{th} root, and thresholds λ and γ. For each parameter, we test 10 different

Fig. 8: Our qualitative results. For each row, we first show alignment result, where the cyan, blue, red, and yellow dots represent all reference images, matched reference images, ground truth aligned frames, and predicted dashcam frames, respectively. Then, we show pairs of reference images (left) and predicted frames (right). Note that our method obtains impressive results even when Street View images are captured in bad weather condition (2nd row).

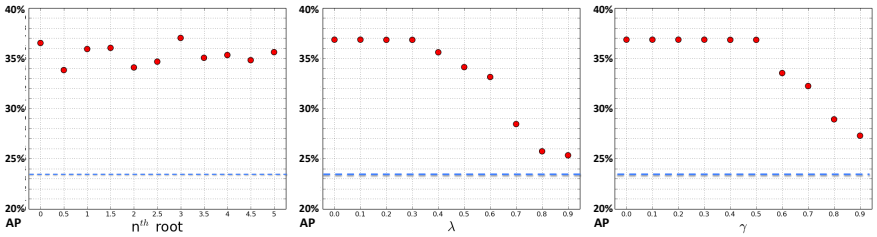


Fig. 9: Sensitivity Analysis: we evaluate 10 different values for parameters n^{th} root, λ, and γ as shown in the left, middle, and right panels, respectively. The blue dash-line indicates the "baseline A" performance (23.4% mAP). When $\lambda < 0.4$ and $\gamma < 0.5$, the performance of our method is stable and consistently better than the baseline.

values while fixing all other parameters. In Fig. 9, we show that our performance is not sensitive to the value changes when $\lambda < 0.4$ and $\gamma < 0.5$.

4.2 Change Detection

Among the 45 videos, we manually confirm that there are 40 predicted pairs of Street View images and frames containing "long-term" changes. Then, we manually annotate ground truth changed rectangle regions in 40 predicted frames. We use the ground truth to evaluate the change detection performance. We consider a predicted rectangle region as a truth change if it overlaps[2] with a ground truth region more than 30%[3]. Our full method (13.54%) outperforms a baseline method without handling car interior and moving objects (11.70%) by 1.84% (relatively 13.6%) in mean AP. Typical examples are shown in Fig. 10. By inspecting failure examples, we found that failures are typically due to severe viewpoint difference between Street View images and dashcam frames, or less ideal dashcam frame quality. Hence, we further conduct a controlled experiment using manually aligned high quality Street View images.

[2] We calculate the intersection over union area.

[3] 30% is used, since ground truth changes are typical irregular and possibly consists of more than one objects. It is very challenging to precisely detect the ground truth change rectangle.

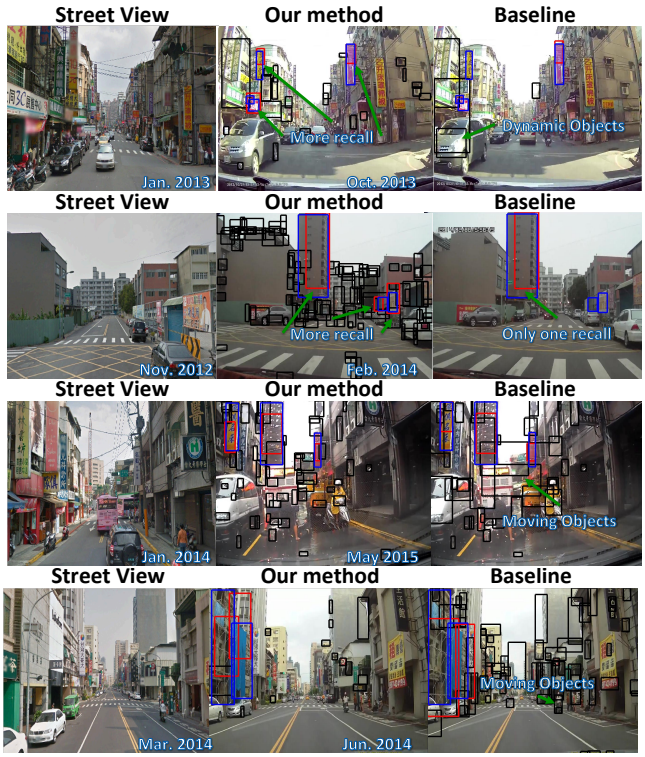


Fig. 10: Change Detection Results om dashcam videos. For each example (three images in a row), we show the Street View image, the detected changes of our method, and the detected changes of the baseline, from left to right, respectively. The month and year of the scene is overlaid on each image. The ground truth changes are marked in blue. True positive predicted changes are marked in red. False positive predicted changes are marked in black. We show the predicted changes from high confidence to low confidence up the the one which gives the maximum recall. Hence, the more recalled true positive, the likely more false positive shown.

Controlled experiment. In order to directly evaluate the performance of change detection (not effected by the quality of alignment), we collected 192 aligned pairs of StreetView images from StreetView Time-machine. Images at the same geo-location in StreetView Time-machine are typically captured at different years. Hence, they are ideal to benchmark "long-term" change detection performance. The ground truth change rectangles are manually labeled for evaluation. Our proposed method (45.57%) outperforms the baseline method (43.67%) in mean AP. Typical examples are shown in Fig. 11. We believe that the overall performance is increased due to the ideal image alignment and much better image quality from Street View than dashcam frames.

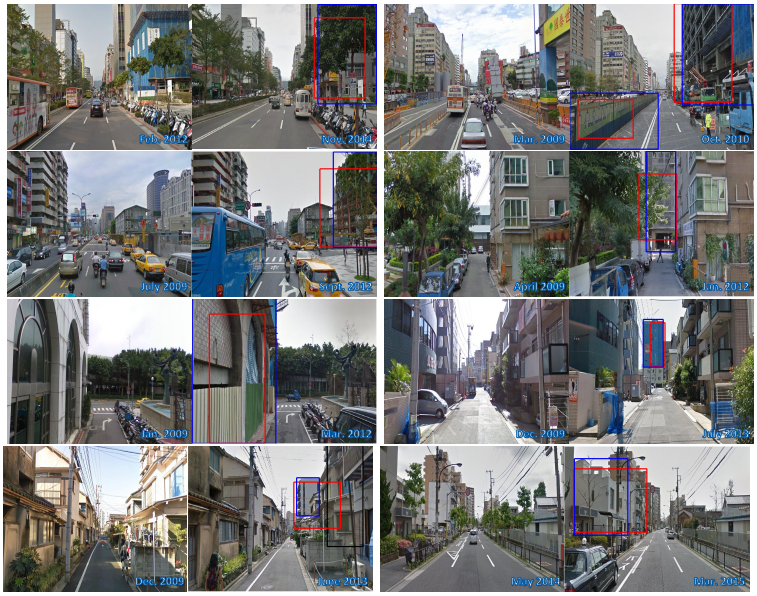


Fig. 11: Change Detection Results on Time-machine images. For each example (two images in a row), we show the Street View image, the detected changes of our method, from left to right, respectively. The month and year of the scene is overlaid on each image. The predicted change and ground truth changes are visualized the same way as in Fig. 10.

5 Conclusion

We demonstrate that consumer-level dashcam videos in the wild can be used to identify changes in high quality Google Street View images. We propose a robust sequence alignment method which overcomes the challenges caused by significant structural changes in the scene, and outperforms the baseline method [1] by 12% mean AP. We also propose a novel change detection method especially designed to detect long-term changes and handle properties in dashcam videos. Our method (13.54%) outperforms a baseline method without handling car interior and moving objects (11.70%) by 1.84% (relatively 13.6%) in mean AP. Moreover, given manually aligned high quality Street View images, our method achieves a significant higher mean AP 45.75%. This suggest that good alignment and image quality are important to reliably detect "long-term" changes. In the future, we will focus on improving alignment quality and extending the change detection method to jointly consider all frames in a video sequence.

Acknowledgement. We thank Industrial Technology Research Institute (ITRI) project grants and MOST 103-2218-E-007-025 and MOST 104-3115-E-007-005 in Taiwan for their support.

References

1. Vaca-Castano, G., Zamir, A., Shah, M.: City scale geo-spatial trajectory estimation of a moving camera. In: CVPR. (2012)
2. Badino, H., Huber, D., Kanade, T.: Real-time topometric localization. In: ICRA. (2012)
3. Kim, J., Liu, C., Sha, F., Grauman, K.: Deformable spatial pyramid matching for fast dense correspondences. In: CVPR. (2013)
4. Robertson, D., Cipolla, R.: An image-based system for urban navigation. In: BMVC. (2004)
5. Zhang, W., Kosecka, J.: Image based localization in urban environments. In: 3DPVT. (2006)
6. Schindler, G., Brown, M., Szeliski, R.: City-scale location recognition. In: CVPR. (2007)
7. Hays, J., Efros, A.A.: Im2gps: estimating geographic information from a single image. In: CVPR. (2008)
8. Zamir, A.R., Shah, M.: Accurate image localization based on google maps street view. In: ECCV. (2010)
9. Cao, S., Snavely, N.: Graph-based discriminative learning for location recognition. In: CVPR. (2013)
10. Bettadapura, V., Essa, I., Pantofaru, C.: Egocentric field-of-view localization using first-person point-of-view devices. In: WACV. (2015)
11. Irschara, A., Zach, C., Frahm, J., Bischof, H.: From structure-from-motion point clouds to fast location recognition. In: CVPR. (2009)
12. Li, Y., Snavely, N., Huttenlocher, D.: Location recognition using prioritized feature matching. In: ECCV. (2010)
13. Sattler, T., Leibe, B., Kobbelt, L.: Fast image-based localization using direct 2d-to-3d matching. In: ICCV. (2011)
14. Li, Y., Snavely, N., Huttenlocher, D., Fua, P.: Worldwide pose estimation using 3d point clouds. In: ECCV. (2012)
15. Taneja, A., Ballan, L., Pollefeys, M.: Never get lost again: Vision based navigation using streetview images. In: ACCV. (2014)
16. Lategahn, H., Schreiber, M., Ziegler, J., Stiller, C.: Urban localization with camera and inertial measurement unit. In: Intelligent Vehicles Symposium (IV). (2013)
17. Floros, G., van der Zander, B., Leibe, B.: OpenStreetSLAM: Global vehicle localization using openstreetmaps. In: ICRA. (2013)
18. Brubaker, M., Geiger, A., Urtasun, R.: Lost! leveraging the crowd for probabilistic visual self-localization. In: CVPR. (2013)
19. Radke, R., Andra, S., Al-Kofahi, O., Roysam, B.: Image change detection algorithms: a systematic survey. TIP **14** (2005) 294–307
20. Pollard, T., Mundy, J.: Change detection in a 3-d world. In: CVPR. (2007)
21. Taneja, A., Ballan, L., Pollefeys, M.: City-scale change detection in cadastral 3d models using images. In: CVPR. (2013)
22. Matzen, K., Snavely, N.: Scene chronology. In: ECCV. (2014)
23. Wu, C.: Towards linear-time incremental structure from motion. In: 3DV. (2013)
24. Wu, C.: Visualsfm: A visual structure from motion system. (http://ccwu.me/vsfm/)
25. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60** (2004) 91–110

26. Jegou, H., Perronnin, F., Douze, M., Snchez, J., Perez, P., Schmid, C.: Aggregating local image descriptors into compact codes. TPAMI (2011)
27. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.: Conditional random fields as recurrent neural networks. In: ICCV. (2015)
28. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. (2015)
29. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: ECCV. (2014)