Road Segment Re-Identification in Dashcam Videos

Yukihiro Tsukamoto Tatsuya Amano Akihito Hiromori Hirozumi Yamaguchi Teruo Higashino
Graduate School of Information Science and Technology
Osaka University
Suita, Japan

***Index Terms*—Dashcam, Re-Identification, Computer Vision**

***Abstract*—Due to the widespread of dashcams, we will have more videos that capture roads/streets in driving. Consequently, a vast amount of the videos will be available and can be utilized for analyzing road safety and similar purposes. For example, suppose different dashcams can take vehicle/pedestrian traffic at a risky intersection at different timings. In that case, the collection of such videos will effectively recognize the cause of dangerous situations without surveillance camera infrastructure. However, identifying a Road Segment of Interest (RSI) in the video, such as near the intersection region, is challenging as the video frames do not usually include location tags. In this paper, we present a unique approach to attack this challenge. Assume that a video segment, called reference video, captures an RSI. We re-identify the RSI taken in another video (called test video) that captures the roads containing that RSI. By this approach, we can automatically extract the video segment corresponding to RSI from a given test video, using the reference video. We introduce AKAZE features to assess frame-level similarity and develop an algorithm to find frame-by-frame matching between reference and test videos. We have evaluated our method using 10 reference videos that correspond to 10 RSIs, each with 5 test videos. The result has shown that the average frame error distance was only 3.03 in daytime and 4.93 in nighttime, which are sufficiently low to re-identify RSI in the newly obtained test videos.**

I. Introduction

The latest survey in Japan [1] has reported that in FY2020, the number of dashcams sold in the Japanese market was about 4.6 million, among 76 million vehicles in Japan. Moreover, the state-of-the-art onboard units have already integrated dashcams to support safe driving. Consequently, a lot of services have been designed and proposed that make use of dashcam videos. For example, in [2] the authors have proposed a method to detect vacant parking spaces using deep learning approaches. The BBC News has reported that road users can submit footage of dangerous driving to police in England and Wales [3].

However, we may often want to obtain the *video segment* that takes a particular region, *i.e.*, a Road Segment of Interest (RSI), within a long video. For example, suppose that we want to analyze the vehicle/pedestrian traffic at a small intersection of narrow roads, where accidents or risky situations are likely to happen, to identify the cause of the risk. Assuming that not a few vehicles pass the intersection every day, we may rely on crowds to provide their dashcam videos and extract the video segment taken at the RSI, *e.g.*, roads near the intersection. However, this segmentation task is time-consuming for humans, as the collected videos do not usually have location tags. Even though recent dashcams can specify GNSS location tags to the video, it is difficult to provide an exact location tag to each frame due to the non-synchronicity of two different sources (video and GNSS). Besides, GNSS errors may often occur, which make more drift from the ground truth.

In this paper, we present an approach to re-identifying an RSI given as a *reference video*, within a video given as a *test video* that contains that RSI. Figure 1 illustrates a scenario. On day 0, a vehicle drove on a road and obtained a dashcam video. We manually extract the video segment (colored by orange) that captures RSI (near intersection region in this scenario) and uses it as a reference video, as shown in the upper part of the figure. On days 1, 2, and 3, we obtained three videos as test videos. We apply our matching technique using the reference video to find video segments that correspond to the RSI (the orange parts in the test videos, in the lower part of the figure).

The challenge here is that we do not rely on any location information. Since the time, days, speeds, lanes, and behavior of vehicles differ between two videos, we should carefully design the matching algorithm. For example, one vehicle did stop-and-go a lot at the RSI in the evening with congested traffic, while another smoothly passed through RSI with less traffic at noon. Therefore, we adopt a frame-by-frame feature extraction and matching to find the segment-level similarity, pursuing both accuracy and computational time. More specifically, we introduce AKAZE features for similarity calculation, which achieve a good tradeoff between computational overhead and feature quality, and develop an algorithm to find frame-by-frame matching between two videos.

We have evaluated our method using 10 reference videos that correspond to 10 RSIs, each with 5 test videos. The result has shown that the average frame error distance was only 3.03 in daytime and 4.93 in nighttime, which are sufficiently low to re-identify RSI in the newly obtained test videos.

II. Related Work

Videos captured by dashcams are used for many purposes, for example, road surface damage detection [4]. Several approaches have introduced deep learning to detect different kinds of cracks. Ref [2] also utilizes videos recorded by a dashcam to detect vacant parking spaces for parking guidance systems. In order to observe the situation in urban areas over a long period, it is necessary to identify the location where the video was taken. Our method compares still images in different videos to detect the images recorded in close locations based

978-1-6654-2854-5/21/$31.00 ©2021 IEEE

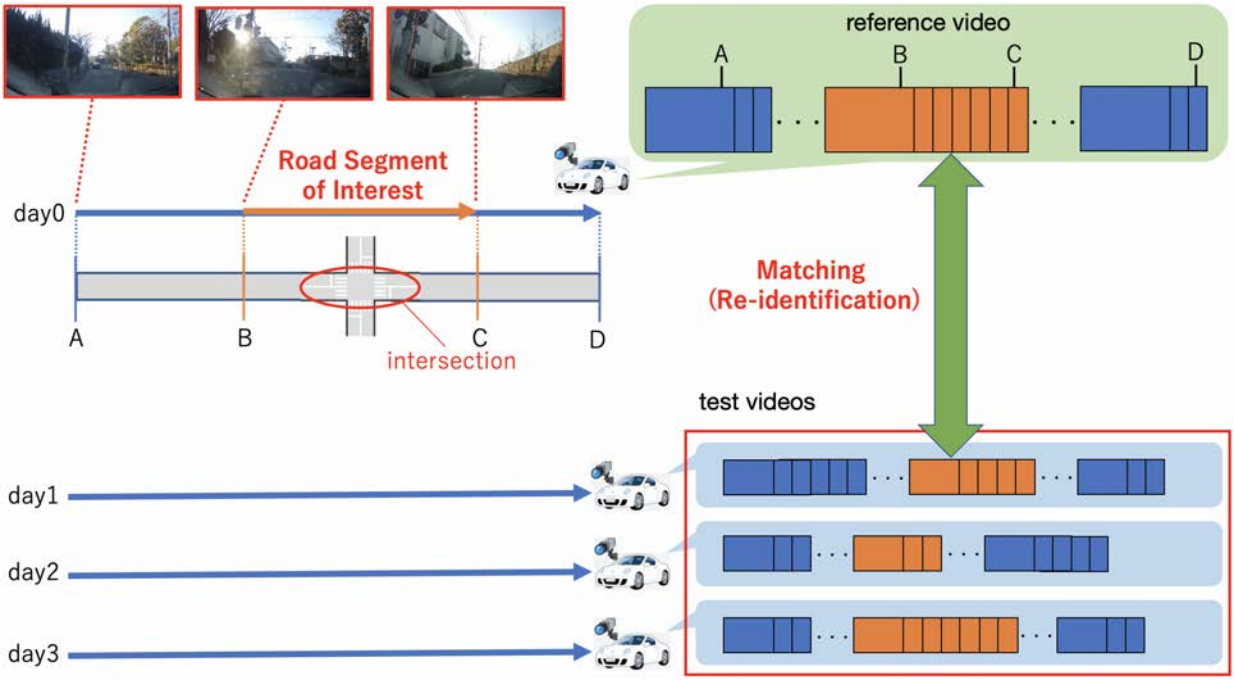

Fig. 1. Road Segment of Interest (RSI) Re-identification in Videos

on image matching techniques. This is called re-identification in this paper.

Several image matching techniques recognize objects or scenes from multiple images using image features derived from the images. Lowe proposes SIFT features that have many properties that make them suitable for matching differing images of a scene [5]. Since the SIFT features are invariant to image scaling and rotation and partially invariant to change in illumination and produce numerical descriptors for each point in each image, these descriptors can identify points or objects in images taken from different perspectives and under different conditions. It means that the SIFT features can accurately determine the position and orientation from which an image was taken. Using the characteristic, Structure from Motion (SfM) creates large-scale 3D models based on the SIFT features from image sequences taken of objects or scenes during the same period [6].

Meanwhile, photo geolocation techniques also utilize not only image features but also additional information attached to images to identify locations of a single image [10]–[13]. Im2GPS [10], [11] derive locations to a given image from millions of geotagged images using global image descriptors. Ref. [13] also proposes a localization method based on the structured dataset of GPS-tagged Google Maps Street View images. PlaNet [12] trains a convolutional neural network (CNN) using millions of geotagged photos to predict the locations of ambiguous photos.

On the other hand, several image localization approaches focus on specializing in specific applications. For example, several approaches are proposed to estimate positions from images from UAVs [14]–[16]. Ref. [15] introduces a deep learning technique for overhead imagery captured from UAVs for large-scale datasets. For intelligent transport systems, Ref. [17] proposes a robust and accurate localization technique that combines visual odometry with map information from OpenStreetMaps. Unlike these methods, our method compares two videos to find pairs of images captured in the same locations and uses only image features to achieve frame-level accuracy with reasonable computation costs.

III. Captured Position Matching Method

A. Preprocessing

At first, the proposed method drops frames of the test video to reduce the processing time for similarity calculation. In this work, the frame rate of the test video is empirically reduced to 6 fps. Next, the method compresses its resolution to 480x270 and converts the image to grayscale in the videos. Both of these processes are intended to reduce the processing time required for the calculation of similarity. Similarly, a given reference video segment is also processed in the same way, *i.e.*, decreasing the frame rate by dropping four of every five frames, compressing its resolution to 480 × 270, and converting the image to grayscale. This process also contributes to reducing the processing time for similarity calculation.

B. Comparing Frames of Reference and Test Videos

In this section, we describe an algorithm that matches test video frames and reference video frames using the AKAZE features.

1) Similarity Calculation: The same buildings and landscapes may appear in the frames of both reference and test videos, if they are taken during the driving on the same road toward the same direction. Therefore, by quantifying the visual features of the buildings and landscapes, it is possible to calculate the similarity of the locations of given two frames. The proposed method uses such visual similarity of frames to find the pair of frames (from test and reference) with the highest similarity. Then we regard them as the frames from the same location. In this work, we use a feature called AKAZE to calculate the similarity. The AKAZE feature is a modification of the SIFT feature [5], which has been traditionally used in the image processing field, and is mainly used for image matching. Since the AKAZE feature has more robustness to changes in lighting conditions, we use this feature to calculate the similarity. To use the AKAZE feature for similarity calculation, we first calculate the AKAZE feature points in the both frames of the test and reference videos, where each feature point is represented as a 1-D feature vector. Then *feature point matching* is performed on a pair of frames from the both videos. In this feature point matching, the Hamming distance between each pair of the feature vectors of the two frames is used to calculate similarity of the two feature points, and the closest pair is regarded as the correspondence. Figure 2 shows an example of feature point matching over two images, where each horizontal line shows a matched pair of feature points.

Then we take the five pairs of feature points in the two frames with the minimal Hamming distances among the other pairs. Then we use the average Hamming distance of these pairs as the similarity value of the given two frames. If there are less than five pairs in the two frames, the frame pair is not suitable for estimating the similarity, and is not considered as similar location images. Figure 3 shows the result of calculating the similarity between a frame of a test video and each frame of the reference video. We can see that, the average Hamming distance is the smallest with the 125th frame of the reference video. This pair of frames is shown in Fig. 4 where the two frames clearly show the same location.

2) Block-wise estimation: Since the similarity calculation described above is done on a frame-by-frame basis, the high similarity may appear with frames taken at different locations but with similar scenery. Also they similarity may be affected by lighting conditions and some other factors. Therefore the same location images do not always show the highest similarity value. Therefore, we use the sequential characteristics of videos, where frame similarity values between two videos are kept high if they capture the same road. For this purpose, we divide the test video into equal-sized blocks and processing them in the following way.

We first consider a block consisting of N consecutive test video frames. Then we let x_i denote the frame number of the test video that appears as i-th frame in the block, and let y_i denote the fraem of the reference video that is best-matched (i.e. most similar) with the x_i-th frame. where $x_i \in \{1, 2, ..., IF\}$ and $y_i \in \{1, 2, ..., HF\}$, IF is the total number of frames of the test video and HF is the total number of frames of the reference video. We consider pairs $P_i = (x_i, y_i)$ where $i \in \{1, 2, ..., N\}$, and plot them on the 2D-space. Then, from the points plotted in the two-dimensional space, we consider N(N-1)/2 straight lines represented by the following equation, and adopt the line with the most points in the vicinity of the straight line as the matching result in that block. In the following, we denote the line passing through the points P_i and P_j as the line $L_{i,j}$.

$$L_{i,j} : y - y_i = \frac{y_i - y_j}{x_i - x_j}(x - x_i) \tag{1}$$

Figure 5 shows example plotting and lines in two-dimensional space. The left figure shows that line $L_{3,4}$ contains several corresponding points near the line, while $L_{4,5}$ of the right figure shows almost no points are near the line. In this case, $L_{3,4}$ is more likely to represent the correct correspondence. After finding the best line, we may identify outliers, that do not follow the tread represented by the line. Allowing outliers to some extent, we decide whether the block is matched or not.

3) Video Matching: Finally, we match the reference video with the test video to identify the correspondence. We note that the number of reference video frames used to calculate the similarity for a given test video frame has a significant impact on the execution time of the entire process. Therefore, it is desirable to limit the number of reference frames when calculating the similarity. The proposed method reduces it by the following procedure.

After the pre-processing, the matching procedure searches the head location of the test video block among the reference video. In finding correspondence, we use frame-by-frame processing using the similarity calculation described above, and block-by-block processing to remove outlier effect. The head position is estimated by calculating the similarity between the first block of the test frame and all the frames of the reference video. After this, in order to reduce the number of reference video frames to be used to calculate the similarity, we only use 18 reference video frames right after the last matched frames of the reference video. This continues until all the blocks are examined. This sequential matching can reduce the number of similarity calculations, which contributes to speedup the procedure. We note that the frame rate of the test video and the reference video is set to 6 fps by the pre-processing. Therefore, 18 frames represent 3 seconds in the reference video. This means that the similarity is calculated with a sufficient range of frames in time.

IV. EVALUATION

A. Dataset

In this section, we describe the dataset used in the experiments to investigate the accuracy of the proposed method.

In this work, we deal with dashcam videos captured on the same road. Therefore, we prepared a vehicle equipped with an onboard camera [18] and collected daily commuting dashcam videos to create a dataset.

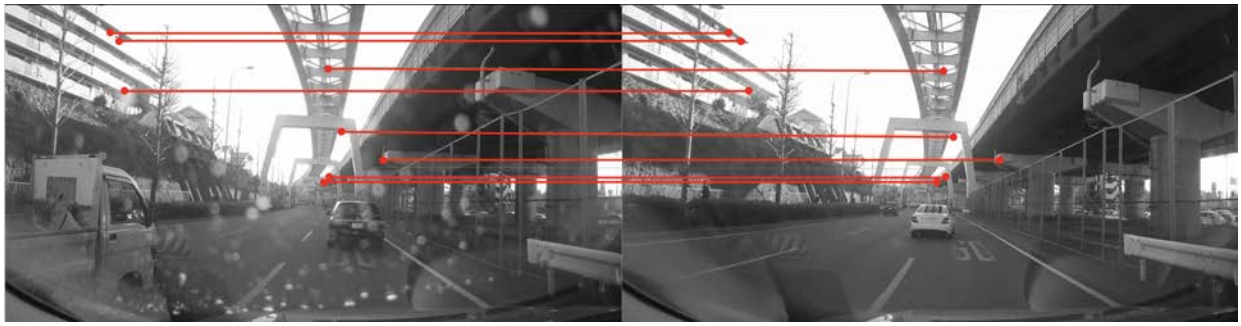

Fig. 2. Example of feature point matching using AKAZE feature

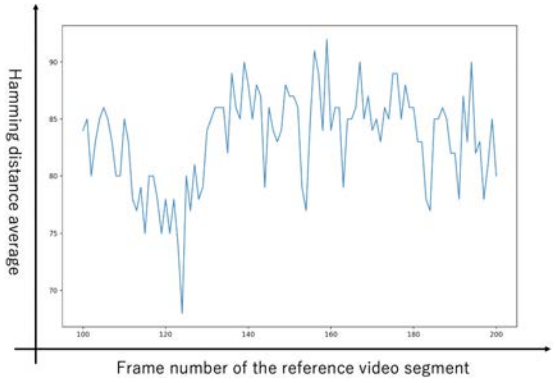

Fig. 3. Transition of similarity in a video

Fig. 4. Similarity calculation results

The commuting route contains both urban areas with many characteristic buildings and rural roads with few buildings. Since all the trips are for commuting, the videos were taken at a similar time every day (6 days totally). Still, the conditions such as weather, driving speed, the number of vehicles in the vicinity, and occlusion are very different in the dataset.

To create reference videos, we picked up six locations (RSIs) on the route traveled during the daytime and four locations (RSIs) on the route traveled during the nighttime. Then for each RSI, we used a 20-second video of day 1 on the route that contains the RSI as a reference video.

Then, for each RSI, 10-second videos containing the RSI from day 2 to day 6, were used as test videos. Each of the test videos is obtained as a portion of the route and has different characteristics in terms of weather, travel speed, the number of surrounding vehicles, and occlusion.

Finally, we manually compared each test video with the corresponding reference video for annotating the ground-truth.

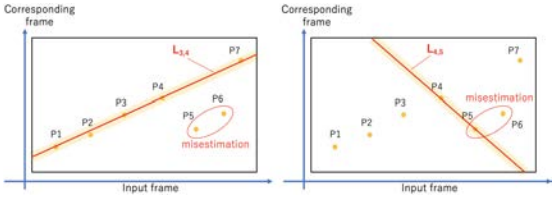

Fig. 5. Two example of matching trend estimations. Left: the most likely trend. Right: less likely trend.

More specifically, for every 10 frames of test video, we found the counterpart in the reference video for ground truth labeling. Figures 6 and 7 show the still images from the dashcam videos in the daytime and nighttime, respectively.

B. Evaluation Result

1) Matching Accuracy: We have applied our method to the dataset above. The results are shown in Table I. We used 45,000 video frames as a total in this experiment, and the evaluation results are summarized for both the daytime nighttime data.

The errors in the table are defined as the distance from the estimated frame and actual frame. For example, if a frame of a test video is matched with the $(i+a)$-th frame of the reference video but if i-th frame is the actual frame, the matching error is $|a|$. As a result, the average error was 3.03 frames in the daytime data, which corresponds to 0.105 seconds at the speed of 50km/h. We note that since both the test and reference videos were 29 fps, one frame corresponds to 1.45m at that speed. On the other hand, the average error in the nighttime data was 4.93 frames, corresponding to 0.170 seconds and 2.36 meters at the same speed. In both cases, the average values are small enough for re-identification purposes. Meanwhile, we have 50 and 58 errors (about 2-second difference) at the maximum. This happens less among several frames, and the small average values show the fact.

To further investigate the error distributions, we have measured the ratio of frames that satisfy a given error threshold. Table II shows the ratio of frames with smaller errors than a given threshold. We prepared two threshold values, 3 and 6. We can see that about 85% of frames can be re-identified in

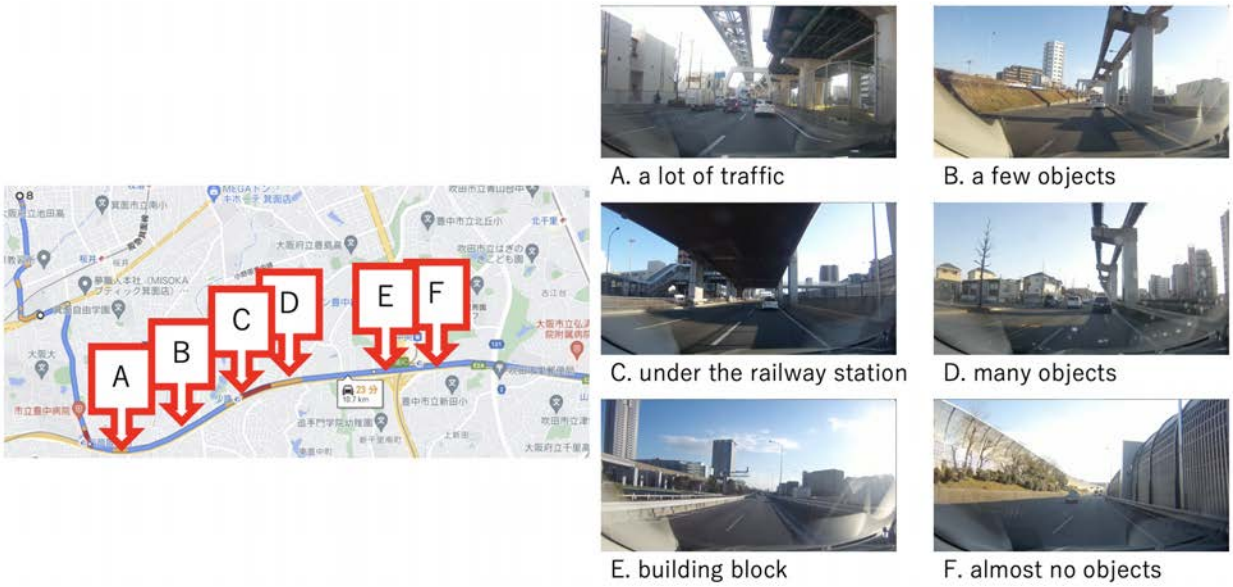

Fig. 6. Daytime Dataset

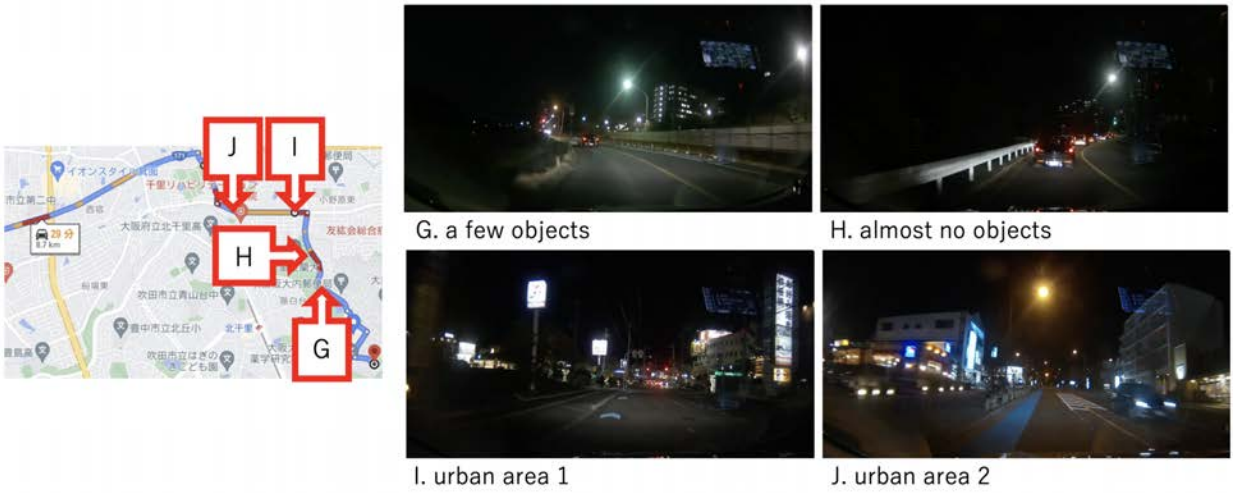

Fig. 7. Nighttime Dataset

the daytime, allowing 0, 1, or 2 frame errors (¡3). Allowing up to 5 frame errors, 93% of the frames could be re-identified. We believe that these results are sufficiently accurate for the re-identification of RSIs. The nighttime results are less accurate than the daytime ones because some roads are illuminated little by streetlights, making it difficult to find effective AKAZE feature points.

TABLE I
MATCHING ERROR (# OF FRAMES FROM TRUTH)

	Daytime	Nighttime
Average	3.0305	4.9310
Variance	39.14	71.52
Maximum	50	58

The AKAZE Feature in the daytime and nighttime scenes are visualized in Figures 8 and 9, respectively. As seen from the figures, many feature points could be found in the background in the daytime. Meanwhile, in the nighttime, the feature points could only be found on moving objects such as vehicles and surrounding environments such as guardrails. The features directly affect the accuracy of both scenes – more feature points result in better matching accuracy.

2) Processing Time: Next, in the same experiment as before, we also measured the processing time required to execute the proposed method. We used a MacBook Pro laptop PC, 2.8GHz quad-core Intel Core i7, with 16GB main memory.

The average time for each process (a pair of 10s test videos and 20s reference videos) was 17.2s. This means that it took

TABLE II
RATIO OF IDENTIFIED FRAMES WITHIN ERROR THRESHOLD

	daytime	night
$Error < 3$ frames	0.8594	0.6694
$Error < 6$ frames	0.9337	0.8195

Fig. 8. Feature points in daytime

0.156s per frame for the start location identification phase and 0.043s per frame for the subsequent processing. Although the processing could not be in real-time, the factor is small enough, and the order of the process is $O(n + m)$ where n and m are the length of test and reference videos, respectively. Besides, our program was implemented using Python and the standard library using a single processor core. We believe that proper optimization and parallelization can be done, which enables real-time processing.

V. CONCLUSION

In this paper, we have proposed re-identifying a road segment of interest (RSI) in a video. The method takes two videos as input, where one is a reference video that captures an RSI, and another contains RSI but is captured at different timing. This method can be used to find the video clips of the same region from many videos that may include video segments of the location. The segmentation accuracy (matching accuracy) was accurate enough. That is, only three frames drift from the correct frames.

Our future work includes enabling real-time processing and building a video repository that collects, via crowdsourcing, risky location videos. Then our method allows clipping particular location videos, which are helpful for assessing safety levels of the locations.

VI. ACKNOWLEDGEMENT

This work was supported by #193 of the Commissioned Research of National Institute of Information and Communications Technology (NICT), JAPAN.

Fig. 9. Feature points at night

REFERENCES

[1] Japan Electronics and Information Technology Industries Association. FY 2020 drive recorder domestic shipment results. https://www.jeita.or.jp/japanese/stat/drive/2020/.

[2] Ming-Che Wu and Mei-Chen Yeh. Early detection of vacant parking spaces using dashcam videos. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9613–9618, Jul. 2019.

[3] BBC News. Dashcam footage submission website goes live. https://www.bbc.com/news/technology-44682669.

[4] Minh-Tu Cao, Quoc-Viet Tran, Ngoc-Mai Nguyen, and Kuan-Tsung Chang. Survey on performance of deep learning models for detecting road damages using multiple dashcam image resources. *Advanced Engineering Informatics*, 46:101182, 2020.

[5] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[6] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016.

[7] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *2011 International Conference on Computer Vision*, pages 667–674, 2011.

[8] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1744–1756, 2017.

[9] Jakob Iglhaut, Carlos Cabo, Stefano Puliti, Livia Piermattei, James O'Connor, and Jacqueline Rosette. Structure from motion photogrammetry in forestry: A review. *Current Forestry Reports*, 5(3):155–168, 2019.

[10] James Hays and Alexei A Efros. Im2gps: estimating geographic information from a single image. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[11] James Hays and Alexei A Efros. Large-scale image geolocalization. In *Multimodal location estimation of videos and images*, pages 41–62. Springer, 2015.

[12] Tobias Weyand, Ilya Kostrikov, and James Philbin. PlaNet - photo geolocation with convolutional neural networks. In *Computer Vision – ECCV 2016*, pages 37–55, 2016.

[13] Amir Roshan Zamir and Mubarak Shah. Accurate image localization based on google maps street view. In *Computer Vision – ECCV 2010*, pages 255–268, 2010.

[14] Aurélien Yol, Bertrand Delabarre, Amaury Dame, Jean-Émile Dartois, and Eric Marchand. Vision-based absolute localization for unmanned aerial vehicles. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3429–3434, 2014.

[15] Yuxin Tian, Xueqing Deng, Yi Zhu, and Shawn Newsam. Cross-time and orientation-invariant overhead image geolocalization using deep local features. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2501–2509, 2020.

[16] Muhammad Hamza Mughal, Muhammad Jawad Khokhar, and Muhammad Shahzad. Assisting uav localization via deep contextual image matching. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:2445–2457, 2021.

[17] Georgios Floros, Benito van der Zander, and Bastian Leibe. Openstreetslam: Global vehicle localization using openstreetmaps. In *2013 IEEE International Conference on Robotics and Automation*, pages 1054–1059, 2013.

[18] TOYOTA MOTOR CORPORATION. Dash cam DRT-H66A. http://www.e-iserv.jp/top/driverecorder/drt-h66a/. [Online; accessed 13-February-2020].