Intersection Prediction from Single 360° Image via Deep Detection of Possible Direction of Travel

Naoki Sugimoto[1]
naoki48916@gmail.com
Satoshi Ikehata[2]
sikehata@nii.ac.jp
Kiyoharu Aizawa[1]
aizawa@hal.t.u-tokyo.ac.jp

[1] The University of Tokyo
Tokyo, Japan

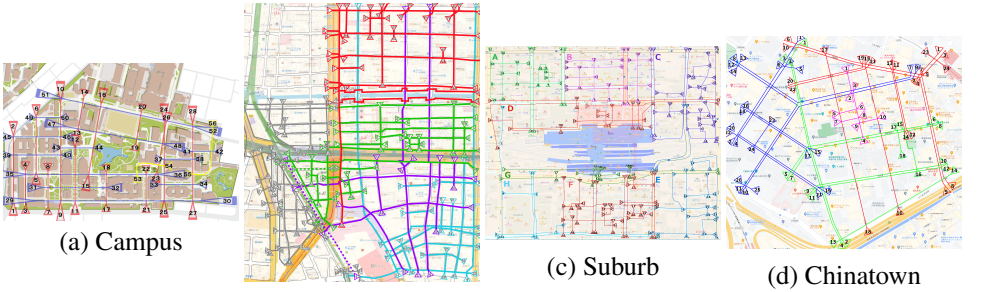
[2] National Institute of Informatics
Tokyo, Japan

Abstract

Movie-Map, an interactive first-person-view map that engages the user in a simulated walking experience, comprises short 360° video segments separated by traffic intersections that are seamlessly connected according to the viewer's direction of travel. However, in wide urban-scale areas with numerous intersecting roads, manual intersection segmentation requires significant human effort. Therefore, automatic identification of intersections from 360° videos is an important problem for scaling up Movie-Map. In this paper, we propose a novel method that identifies an intersection from individual frames in 360° videos. Instead of formulating the intersection identification as a standard binary classification task with a 360° image as input, we identify an intersection based on the number of the possible directions of travel (PDoT) in perspective images projected in eight directions from a single 360°image detected by the neural network for handling various types of intersections. We constructed a large-scale 360° Image Intersection Identification (iii360) dataset for training and evaluation where 360° videos were collected from various areas such as school campus, downtown, suburb, and china town and demonstrate that our PDoT-based method achieves 88% accuracy, which is significantly better than that achieved by the direct binary classification based method. The source codes and a partial dataset will be shared in the community after the paper is published.

1 Introduction

Movie-Map [9, 14] is a digital map application that presents first-person images of a specific location on a map, giving the viewer the immersive experience of actually being there [1, 5, 16]. Internally, Movie-Map comprises short 360° video segments separated by traffic intersections. When passing through an intersection, the video segments are seamlessly connected according to the direction of travel, and the user remains unaware of the connection. However, in reality, shooting a large amount of short videos between intersections is inefficient; therefore street-level videos are typically captured and then split by intersections. In the existing works, this intersection segmentation has been performed manually [9] or by deciding the intersections after SLAM of each street video [14]. However, as shown in Fig. 1,



(a) Campus (b) Downtown (c) Suburb (d) Chinatown

Figure 1: Routes and intersections in the collected 360°walk-around videos. The arrows and triangles on each map indicate the video paths and intersections, respectively.

because of the presence of several intersections in the areas targeted by Movie-Map, manual intersection segmentation requires significant effort. Furthermore, dynamic objects such as pedestrians or cars and less-textured landscapes frequently hamper the accuracy of image-based SLAM [10]. In addition, a comparison between camera poses and visual information from intersecting video sequences of complex streets is not always accurate. Although GPS can be used to obtain the coordinates of the camera locations, the error is sufficiently large to prevent correct localization. Moreover, GPS is not applicable to indoor situations, either.

In this paper, we propose a learning-based algorithm to identify the intersection automatically using a single 360° video frame. The intersection frame was used to split walk-around videos into short video segments. The most challenging condition in this task is the use of a single image; if accomplished, it can be used most generally; that is, we can determine the intersections without using information from other video sequence frames. We formulated this problem as the detection of possible directions of travel (PDoT) for handling various types of intersections. Provided a 360° frame sampled from a video, perspective projections for different views were applied to generate multiple perspective images (*i.e.*, 8 views in our experiments). Then, we classified the possibility of an observer to walk in the forward direction of the field-of-view (FoV) without being blocked by obstacles such as buildings. If PDoTs were observed in three or more views, the frame was identified as a traffic intersection. As a baseline for comparison, we also prepared a direct binary classification of the provided image without using PDoTs. The experimental results showed that our PDoT-based method can achieve 88% accuracy in identifying a single image intersection and generally outperformed the direct classification-based approach for various real scenes.

We trained our PDoT detection network and evaluated the intersection identification method on our new dataset, 360°Image Intersection Identification (iii360), which was generated from 360° walk-around videos in four areas and 360° images from the Google Street View (GSV) panoramas [6]. The change in accuracy of intersection identification was investigated with different combinations of training and testing datasets. The results demonstrated that the proposed method achieved 88% accuracy even for an area not included in any of the three datasets used for training, whereas the näive binary classification network achieved 65% detection accuracy.

2 Related Work

2.1 Movie-Map

Movie-Map [9] was proposed in 1980 as the first interactive map to engage a user in a simulated driving experience. The original Movie-Map system was built using an optical videodisc and four stop-frame film cameras – the cameras, mounted on the top of a car, were triggered approximately every 10 ft. Movie-Map simulates travel by displaying controlled rate sequences of individual frames captured at periodic intervals along a particular street in a town. Despite being an innovative concept, the system was impractical because of the large human efforts involved. For instance, to allow the route to deviate from straight paths down each street, separate sequences were captured to display all the possible turns at every intersection. In addition, the captured videos had to be split manually by intersection to connect the different driving videos through those turn sequences. Owing to the lack of scalability caused by the large human effort as well as that of computational resources and data capacity at that time, Movie-Map was relegated to less importance until recently, and a system based on static 360° images (such as GSV [6]) became the mainstream approach to digital map navigation.

Recently, Sugimoto *et al*. redesigned Movie-Map with modern imaging and information processing technologies and demonstrated its superiority to GSV in exploring unfamiliar scenes [13, 15]. In addition to replacing large-camera and expensive disk systems in [9] with consumer 360° cameras and personal computers, they proposed a method to automate the time-consuming intersection segmentation. Specifically, they applied Visual SLAM [10] to recover and track the 3-D camera trajectories of the walk-around videos, aligned the trajectories onto the map, identified the intersections using the aligned trajectories, and refined them using visual features. However, their method relied heavily on the results of Visual SLAM, which is not robust to dynamic objects such as cars or people and texture-less landscapes. On the other hand, as our method does not rely on either 3-D reconstruction or multiple frames from the entire frame sequence, it is less sensitive to dynamic objects and low-textured scenes.

2.2 Traffic intersection detection

In addition to Movie-Map, intersection identification using a pedestrian viewpoint or vehicle-mounted cameras has recently been actively studied in automated driving and robot navigation fields. Owing to the highly unpredictable behavior of traffic intersections, correct recognition and safe behavior in their proximity is essential for an autonomous agent.

Depending on the input information, intersection identification methods are broadly categorized into two types: those that use non-visual information such as LIDAR or GPS data and those that use images or videos. Without using visual information, Fathi and Krumm [4] identified intersections in an urban-scale traffic network using GPS data from several vehicles. On the other hand, Zhue *et al*. [19] proposed a method for intersection detection using sparse 3-D point clouds obtained from in-vehicle 3-D LIDAR data. Unfortunately, GPS data are frequently inaccurate in urban areas with tall buildings, and 3-D LIDAR is expensive and basically less portable than consumer cameras, which are inappropriate for capturing walk-around videos for some applications such as Movie-Map.

With the development of deep learning technologies, image-based methods for intersection identification are becoming more attractive owing to their cheap installation cost and compatibility with in-vehicle or robot-mounted cameras. Bhatt *et al*. [3] formulated this task as a binary classification problem to identify intersections in short-time on-board videos and

Figure 2: Example of an ambiguous intersection image. In this school campus scene, the road is omnidirectional, and the number of PDoTs is difficult to accurately identify.

developed a variant of long-term recurrent convolutional network as the classifier. Although the classifier has been shown to be effective for in-vehicle videos, its reliability in other domains such as walk-around pedestrian-view videos, is still unclear. In addition, as this model accepts multiple video frames as input, its robustness in dynamic scenes with several walking people, such as in urban shopping malls, is questionable. On the other hand, Astrid *et al*. [2] recently proposed a single-image intersection classification system using ResNet-based architecture trained on a pedestrian-view-level image dataset containing 345 and 498 intersection and non-intersection images, respectively. Although the system achieved a test accuracy of 80%, the training and test data comprised pedestrian-view images of sidewalks with little or no pedestrian traffic, and the generalization of this method to disparate test scenes from the training dataset is still not confirmed.

Unlike all these studies, the input of our method is a single frame from 360° walk-around videos recorded for Movie-Map. To the best of our knowledge, this is the first attempt to identify an intersection from a single 360° image. In addition, our targets include not only in-vehicle images or places with no pedestrians but also various cities, towns and villages with active people and car traffic, including places with different building designs. In such cases, conventional approaches such as direct binary classification of images are difficult to implement. Therefore, we propose a new intersection identification method based on the detection of PDoTs.

3 Proposed Method

The purpose of this study was the automatic identification of intersection frames in walk-around videos shot for Movie-Map and the division of the video into intersection-wise segments. To achieve this goal, we propose a novel single 360° image intersection detection algorithm that is applicable to challenging pedestrian views in various types of scenes.

The task is to classify whether a selected the image was captured at an intersection. The most straightforward method would be a data-driven approach wherein a large number of intersection and non-intersection labeled images are prepared to train deep neural networks, as performed in [2].

Unfortunately, the definition of an intersection in reality is quite vague, except for a clearly sectioned roadway. For instance, a place with people moving in various directions, such as a university campus shown in Fig. 2, should be recognized as an intersection in practical applications such as Movie-Map because of the intersecting people and vehicles. However, even a human would hesitate to label this location as an intersection because of the absence of clearly sectioned road. Conventional intersection detection studies [2, 3, 4, 19] did not encounter this ambiguity as they only focused on on-board cameras or pedestrian

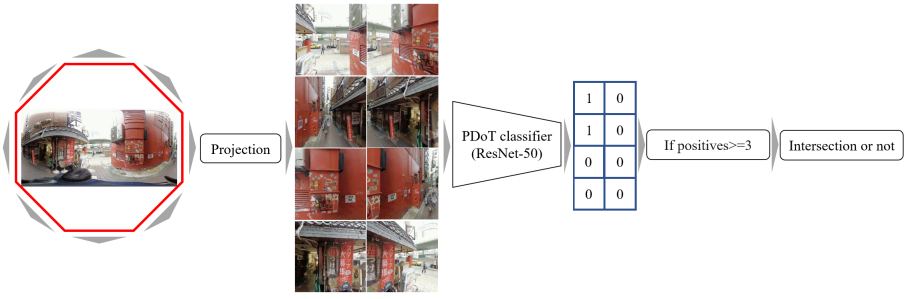


Figure 3: Illustration of the proposed method. Multiple non-overlapping perspective images of the $45^\circ \times 45^\circ$ FoV are cropped in eight view directions and our PDoT classifier is applied to individual images. If three or more PDoTs are detected, the 360° image is classified as an intersection frame.

view on clearly segmented sidewalks.

To address this problem, we redefined the intersection itself more rigorously and defined the problem accordingly. Specifically, an intersection in a 360° image was defined as *a location where there are multiple PDoTs*. Specifically, if a traveler can proceed in three and more directions, including the direction they have already been traveling, the location should be considered an intersection. The overview of our method is shown in Fig. 3 If the direction of travel contains an obstacle such as a building, it is defined as not be travelable. Under this definition, both conventional (*e.g.*, T,Y,X-intersections) as well as omnidirectional traffic intersections (*e.g.*, those in a university campus square) are labeled as intersections. Notable, identification of omnidirectional intersections is crucial in several practical applications. Movie-Map needs to merge walk-around videos in two directions in a park square. Advance detection of large spaces that are expected to contain heavy traffic is also important for robot navigation.

We propose a single 360° image intersection classification network based on the redefined intersection. Deep neural networks for various tasks using 360° images have been studied extensively in recent years and can be mainly divided into three approaches: apply neural networks directly to equirectangular projection (ERP) images [20], define kernels for convolution on a sphere [18], or divide a 360° image into multiple perspective projection images and apply neural networks to them individually [17]. While the ERP-based method is attractive owing to its simplicity, our experiment validated that this direct approach is not always effective for a wide variety of real-world intersections (including omnidirectional intersections) because of large distortions in a 360° image caused by sphere-to-plane projection. Therefore, we formulate this problem as the detection of multiple PDoTs in sampled perspective views, rather than directly identifying the intersections in 360° images.

Specifically, we converted a single 360° image provided in the ERP format (*e.g.*, a single frame of a 360° walk-around video) into multiple non-overlapping normal perspective images. The FoV of each perspective image should not be quite small or quite large because it is insufficient or contains multiple PDoTs, respectively. To balance the trade-off between redundant and insufficient information, we cropped eight perspective images of the $45^\circ \times 45^\circ$ FoV from a target 360° image. For each perspective image, we classified whether the center of the FoV contained a PDoT by applying a variant of the ResNet-50 [7] network that will be explained in the implementation details. The input 360° image was identified as an intersection if a minimum of three PDoTs were observed in the 360° field-of-view. By dividing

(a) Campus (b) Downtown (c) Suburb (d) Chinatown

Figure 4: Image samples from each area. Campus has large wide roads and squares and ambiguous intersections. In Downtown, most of intersections have traffic lights and pedestrian crossings. Suburb has many intersections without such elements, but the roads are clearly separated by walls or buildings. Chinatown consist of all those features where many people come and go. We used the first three areas as training data and used Chinatown data only as test data.

the complex problem of direct intersection identification from the entire 360° view into a set of problems that determine the presence of PDoT in the narrow FoV, the system displays greater robustness to scene divergence, which was validated by the experimental results. Furthermore, by converting a 360° image in the ERP format to perspective images, our method is theoretically less sensitive to sphere-to-plane projective distortions. Notable, although our method requires multiple model inference to determine whether a single 360° image is an intersection, they are independent and can be executed in parallel. Moreover, as perspective local images are much smaller than the original 360° image, the computational cost is quite manageable.

4 iii360 Dataset

The identification of an intersection in a single 360° image is a new problem, with consequent nonavailability of training and test datasets. Therefore, we constructed a large-scale 360° Image Intersection Identification (iii360) dataset for training our PDoT detection network and evaluating our method.

4.1 Training data generation

Training data for PDoT detection: First, we extracted 360° images from the 360° videos in Campus and Downtown areas, obtained for Movie-Map. For each area, we chose 50 videos and selected 10 frames at equal time intervals from each video; 500 frames were obtained per area. The two areas have different characteristics as observed in Fig. 4. For example, on Campus, an intersection on Campus is defined ambiguously, as mentioned previously. In Downtown, streets are demarcated precisely by buildings, with maintained crosswalks or traffic lights, along with dynamic changes such as moving cars in a scene.

The maximum number of Movie-Map videos have been shot by a photographer walking on the sidewalk in a wide street. Therefore, the intersection detector should be trained in the same domain, rather than on videos shot by car-mounted cameras such as in GSV [6]. However, as the diversity of the intersections in the two areas mentioned previously was insufficient for generalizing the intersection identification, we supplemented it by adding another training dataset, Suburb-GSV images. The added area contains well-maintained streets but scarce traffic lights and signs; moreover, the dataset contains a few scenes without any walls or buildings and low textures. We collected 500 suburb intersection 360° image from GSV and used them for training in addition to the images from Campus and Downtown.

Next, for each key interaction frame included in this frame set, PDoTs were manually annotated on a user interface. Specifically, a worker was asked to indicate the PDoT in a

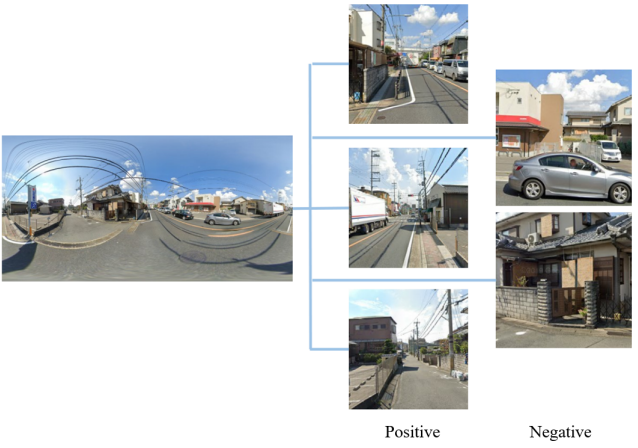


Figure 5: Illustrated strategy for generating positive and negative examples from a 360° image for training the PDoT classifier. Positive examples are sampled as normal-field-of-view (NFoV) perspective images roughly centered at every annotated PDoT. Negative examples are sampled as NFoV perspective images centered at two adjacent PDoTs so that it contains no PDoT in its FoV.

frame with icons specifying a single or "omnidirectional" PDoT, where all the directions are assumed to be PDoT, for example, in a plaza on the university campus. After completing all the annotations, we cropped the NFoV (45°in horizontal and vertical views) images approximately centered at PDoT from the 360° image as positive examples. It should be noted that, to inject noises in the training examples, we did not crop the NFoV image strictly centered at the PDoT and randomly shifted it up to 5 degrees. After the extraction of the positive examples from a single frame, the negative examples were similarly extracted from the same frame as NFoV images centered between two adjacent PDoTs. Notably, this sampling procedure can be applied to both key intersection and non-intersection frames (*i.e.*, non-intersection frames have their PDoTs in the forward and backward directions). As shown in Fig. 5, this procedure basically generates slightly more positive than negative examples because a few negative candidates are discarded when the inner angle between two adjacent PDoTs is less than 45°(*e.g.*, Y-junction; otherwise, the negative example must contain PDoT). To equalize the number of positive and negative examples, we added additional negative examples from other random regions. Consequently, we obtained 3414 positive examples (*i.e.*, 907 in Campus, 971 in Downtown, and 1536 in Suburb) and 3274 negative examples (859 in Campus, 882 in Downtown, and 1510 in Suburb). These samples were precisely labeled and diverse in terms of landscape.

Training data for direct intersection classification: While our proposed method identifies an intersection based on the number of PDoTs in a single 360° frame, the näive network architecture accepts 360° image as input and directly predicts whether the frame is an intersection. We also created training data to train the näive method. Because we required a larger number of samples of images for training in the direct method, more frames were sampled from the video set. Rather than using Suburb-GSV for the PDoT method, we used videos shot in almost the same area. Thus, except Suburb and Suburb-GSV, the same sources (Campus and Downtown) were used for training the PDoT and direct methods. Both the methods used the same testing data.

Specifically, equipped with the annotation of key intersections, we re-assigned the *soft*

intersection label to every 10 frames in the videos based on the frame positions from the key intersection frame. We shot walk-around videos in Suburb similar to that in Campus and Downtown areas. Frames within 0.5 s walking distance (*e.g.*, 15 frames in the 30-fps video) from the key intersection frame were labeled as one (positive) and frames with more than two second walking distance as zero (negative); the frames decreased linearly from one to zero between 0.5 s and two second walking distances. Because a majority of the frames were negative examples (*i.e.*, far more than two second walking distance from the key intersection frame therefore labeled as "False"), we balanced the portions of positive and negative examples by only extracting p percentage of negative ones. We empirically found that $p = 20\%$ is the optimal percentage, which means that 80 percentage of negative samples were discarded. Finally, we obtained 544 positive examples, 1225 soft-labeled examples (*i.e.*, labeled between zero and one), and 8159 negative examples. The total distribution of (positive/soft-labeled/negative) examples for each area was (198/420/2188) for Campus, (185/457/3656) for Downtown, and (161/348/2315) for Suburb. Notable, The labels were treated as a continuous probability distribution.

4.2 Test data generation

The same test task was employed for both the PDoT-based and näive direct approaches: intersection identification from a single 360°image. We prepared 50 intersection and non-intersection frames, respectively, for each of the three areas and added a completely new scene called Chinatown to validate the ability of the network to generalize to unknown scenes. The test data resulted in a total of 400 frames. Notably, all the test images were not included in our training data.

5 Experimental Results

We evaluated our PDoT-based 360°image intersection identification method (our method) on our iii360 test dataset. We compared its performance against that of the näive direct method (Baseline), which we implemented as a binary classification. It should be noted that the qualitative result examples of intersection identification have been demonstrated in the supplemental paper.

5.1 Implementation details

Our method and Baseline were implemented using the PyTorch framework [11]. The backbone architecture for both the methods was Resnet-50 [7], pre-trained on ImageNet [12] except for the final classification layers and fine-tuned on the iii360 dataset. This implies that the only difference between these two networks was the input and output information. The input for our method was an NFoV image cropped from a target 360° image, whereas that for the Baseline was the 360° target image itself. The output was the binary labels w.r.t PDoT or intersection. Both networks accepted images resized to 224×224 as input and were trained and tested on a machine with single NVIDIA TITAN Xp with 12GB of GPU memory. For optimization, we used Adam [8] with a learning rate of 0.001 and batch size of 8. The number of epochs was set to 300 for all network and training set combinations. For data augmentation, horizontal flipping, color jitter, and random erasing were applied.

To identify the intersection, our method merges the multiple PDoT predictions as follows. First, in a provided target 360° image, horizontally non-overlapping eight perspective images of $45° \times 45°$ FoV are cropped (in a random start direction). Then, each NFoV image is

Method	Dataset	Test set domain			
		Campus	Downtown	Suburb	Chinatown
PDoT	Campus	0.78	0.82	0.70	0.65
	Downtown	0.67	0.88	0.75	0.87
	Suburb-GSV	0.68	0.78	0.81	0.57
	Three datasets	0.74	0.86	0.81	0.88
Direct	Campus	0.65	0.54	0.51	0.57
	Downtown	0.52	0.81	0.68	0.67
	Suburb	0.49	0.72	0.78	0.66
	Three datasets	0.57	0.68	0.69	0.65

Table 1: Intersection prediction results.

Views	Accuracy
8	0.65
16	0.58
32	0.55

Table 2: Result of proposed method with different views of cropped images. Training data was the Campus area data and Test data was the Chinatown area data in iii360.

Resolution	Accuracy
224×224	0.57
448×448	0.55
896×896	0.58

Table 3: Result of Direct method with different resolutions. Training data was the Campus area data and Test data was the Chinatown area data in iii360.

individually fed to the PDoT prediction network, and the target 360° image is identified as an intersection if when more than three PDoT predictions are positive.

5.2 The effect of hyperparameters

We investigated the effect of the number of perspective images cropped from 360° images. We varied it within 8, 16 and 32 (*i.e.*, 45° , 22.5° and 11.25° FoV, respectively) and compared the prediction performance by training the network on Campus and testing on Chinatown datasets. The result is shown in Table 2, and we found the result of 8 views is the best.

In addition, in order to evaluate the effect of the resolution of input images in the direct method, we compared the prediction accuracy by training the network on Campus and testing on Chinatown datasets by varying the image size among 224×224, 448×448 and 896×896. Table 3 demonstrated that the difference of input image size is not influential to the performance.

5.3 Quantitative results

We compared our method with Baseline having different combinations of training/test data. In addition to training the networks on examples from individual areas, we attempted training on all the examples from all areas. Notably, we did not evaluate PDoT prediction accuracy because this was not our main task; instead, we assigned more importance to the intersection identification performance.

The comparative intersection identification accuracy results are shown in Table 5.3. It is evident that the prediction accuracy of our method is better than that of Baseline for all the combinations of training and test areas, indicating that our PDoT-based algorithm consistently outperforms the näive direct approach of the Baseline algorithm.

As expected, the performance was better when the training and test examples were drawn from the same area, compared to when drawn from different areas. Interestingly, the network trained on Suburb-GSV still performed satisfactorily on the test Suburb dataset despite the

completely different device setups for obtaining both types of data, indicating that a difference in the image acquisition setup exerts a lesser effect than that in the area where the image was captured. The 0.87 prediction accuracy demonstrated by the network trained on Downtown and tested on Chinatown also supports this observation as both these areas have several common characteristics, such as movement of people through narrow and intricate shopping streets. Conversely, the network trained on Campus did not perform well on Chinatown where both areas have less shared characteristics. Although the excellent performance of the network on Downtown even after training on Campus appears counter-intuitive, the average prediction accuracy on Downtown is consistently high and simply indicates that intersection identification in this area is relatively easier than in other areas with more diversity in the appearance of intersections. It is not surprising that the network trained on all the training examples performed slightly worse than when the training and test examples were drawn from the same area. Instead, this result indicates that the network is capable of covering a variety of areas by drawing training images of intersections from them.

6 Conclusion

In this paper, we presented a new algorithm to identify an intersection from a single 360° image. We propose a PDoT-based method that identifies intersections by the number of possible directions of travel, rather than directly identifying the 360° image with a binary classifier. For training our PDoT detection network and evaluating the method, we constructed a new large-scale 360° Image Intersection Identification (iii360) dataset. Although the original motivation of our work was the automatic intersection segmentation of Movie-Map videos, we believe our method is also applicable in other fields, such as autonomous driving and robot navigation.

As our network architecture was quite basic (ResNet-50) and the training examples were not that large in number, improving the network architecture and increasing the training dataset size is an important future direction. Another important future work is to demonstrate the effectiveness of our method in actual Movie-Map application.

7 Acknowledgement

This work is partially supported by JSPS KAKENHI 21H03460, JST-Mirai Program JPMJMI21H1 and VTEC Lab.

References

[1] Mapillary AB. Mapillary. https://www.mapillary.com/, 2014.

[2] M. Astrid, J. H. Lee, M. Zaigham Zaheer, J. Y. Lee, and S. I. Lee. For safer navigation: Pedestrian-view intersection classification. In *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 7–10, 2020. doi: 10.1109/ICTC49870.2020.9289182.

[3] D. Bhatt, D. Sodhi, A. Pal, V. Balasubramanian, and M. Krishna. Have i reached the intersection: A deep learning-based approach for intersection detection from monocular cameras. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4495–4500, 2017. doi: 10.1109/IROS.2017.8206317.

[4] Alireza Fathi and John Krumm. Detecting road intersections from gps traces. In *Proceedings of the 6th International Conference on Geographic Information Science*, GIScience'10, page 56–69, 2010.

[5] Google. Google map. https://www.google.com/maps/, 2005.

[6] Google. Google street view. https://www.google.co.jp/intl/ja/streetview/, 2005.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. URL http://arxiv.org/abs/1412.6980.

[9] Andrew Lippman. Movie-maps: An application of the optical videodisc to computer graphics. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '80, pages 32–42, New York, NY, USA, 1980. ACM. ISBN 0-89791-021-4. doi: 10.1145/800250.807465. URL http://doi.acm.org/10.1145/800250.807465.

[10] P. Lothe, S. Bourgeois, F. Dekeyser, E. Royer, and M. Dhome. Towards geographical referencing of monocular slam reconstruction using 3d city models: Application to real-time accurate vision-based localization. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2882–2889, June 2009. doi: 10.1109/CVPR.2009.5206662.

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learn pdf.

[12] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[13] Naoki Sugimoto, Yuko Iinuma, and Kiyoharu Aizawa. Walker's movie map: Route vies synthesis using omni-directional videos. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 1050–1052, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450368896. doi: 10.1145/3343031.3350587. URL https://doi.org/10.1145/3343031.3350587.

[14] Naoki Sugimoto, Yoshihito Ebine, and Kiyoharu Aizawa. *Building Movie Map - A Tool for Exploring Areas in a City - and Its Evaluations*, page 3330–3338. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450379885. URL https://doi.org/10.1145/3394171.3413881.

[15] Naoki Sugimoto, Toru Okubo, and Kiyoharu Aizawa. Urban movie map for walkers : Route view synthesis using 360 degree videos. In *Proceedings of the 2020 International Conference on Multimedia Retrieval (ICMR '20), June 8–11, 2020, Dublin, Ireland*. ACM, 2020. ISBN 978-1-4503-7087-5/20/06. doi: 10.1145/3372278.3390707.

[16] M. Tanaka and T. Ichikawa. A visual user interface for map information retrieval based on semantic significance. *IEEE Transactions on Software Engineering*, 14(5):666–670, May 1988. ISSN 2326-3881. doi: 10.1109/32.6144.

[17] Fu-En Wang, Hou-Ning Hu, Hsien-Tzu Cheng, Juan-Ting Lin, Shang-Ta Yang, Meng-Li Shih, Hung-Kuo Chu, and Min Sun. Self-supervised learning of depth and camera motion from 360 degree videos. In *Asian Conference on Computer Vision*, pages 53–68. Springer, 2018.

[18] Chao Zhang, Stephan Liwicki, William Smith, and Roberto Cipolla. Orientation-aware semantic segmentation on icosahedron spheres. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3533–3541, 2019.

[19] Q. Zhu, L. Chen, Q. Li, M. Li, A. Nüchter, and J. Wang. 3d lidar point cloud based intersection recognition for autonomous driving. In *2012 IEEE Intelligent Vehicles Symposium*, pages 456–461, 2012. doi: 10.1109/IVS.2012.6232219.

[20] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2051–2059, 2018.