Mobility21

A USDOT NATIONAL UNIVERSITY TRANSPORTATION CENTER

Carnegie Mellon University

Penn
University of Pennsylvania

The Ohio State University

CCAC

Bus on the Edge: Passengers

PI: Christoph Mertz
https://orcid.org/0000-0001-7540-5211
Co-PI: Patrick Carrington
https://orcid.org/0000-0001-8923-0803

FINAL RESEARCH REPORT

Contract # 69A3551747111

Overview

Before the start of this project we developed one bus-on-the-edge system that uses cameras mounted on a bus to monitor infrastructure and traffic. The hardware components are a standard security system for transit buses: five cameras observing the surrounding of the bus and a computer to record the videos (Figure 1).

Figure 1 The left four images show the commuter bus, an image of the computer, the cabinet that contains the computer and electronics, and one of the cameras. The right diagram shows the field of view of each installed camera, where one faces forward and four side cameras face opposite directions from each other.

We have installed our own software on that computer. The main functions of the software is to analyze and manage the data. The videos from the five cameras are far too much data to upload to the cloud. Instead, it needs to be pre-analyzed on the bus itself and then data of interest is sent to a central location where it is analyzed more thoroughly and used for various applications (Figure 2).

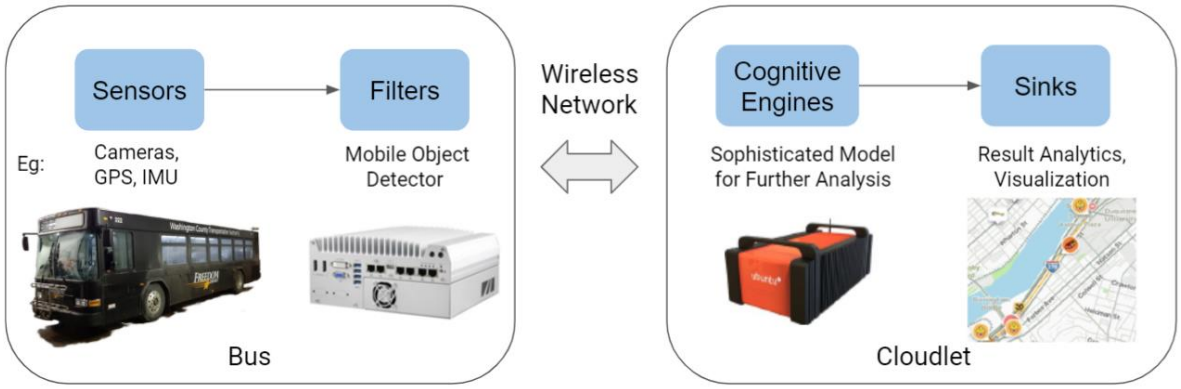

Figure 2 Schematic of data and analysis flow. The raw sensor data are pre-analyzed by a computer on the bus. Interesting data is wirelessly transmitted to a server ("cloudlet"), where it is analyzed in detail and finally sent to the end-user.

The computer has a cellular connection for time critical communication and a WiFi link to exchange large amounts of data. The system was installed on a FreedomTransit bus in February 2021 and we tested a first set of applications at that time.
In the project of the past year we equipped a second bus with a similar system. The added features of the second system are two additional cameras in the inside of the bus and a more powerful computer with a GPU that can run deep learning algorithms. An example of a view from inside a bus with a passenger is shown in Figure 5. The installation in the

second bus took longer than expected because of a defect in the computer that needed to be fixed by the manufacturer. The applications we developed and implemented in this project are described in detail in the master's thesis "A Single View Perspective Prior for Efficient Detection"[1] and the papers "Learned Two-Plane Perspective Prior based Image Resampling for Efficient Object Detection"[2] and "Detection and Tracking of Accessibility Challenges"[3]. These publications have been jointly supported by this project and a NSF-CPS project. Additionally, data and software from this transit bus project was used for the master's thesis "Domain Adaptation by Revisiting Static Objects with a Transit Bus"[4].

Efficient Detection

One issue our bus-on-the-edge system has to deal is that it needs to analyze a lot of images with limited computational power. One simple way to increase throughput is to decrease the size of the images, which of course will decrease the detection quality. To limit the deterioration in detection quality one can warp the image, so that areas of the image which have a high likelihood to contain interesting objects will decrease less in size than other areas (Figure 3).

Figure 3 Original and warped image.

In this work, we developed learnable geometry-guided prior that incorporates rough geometry of the 3D scene (a ground plane and a plane above) to resample images for efficient object detection. This significantly improves small and far-away object detection performance while also being more efficient in terms of both latency and memory. Table 1 shows the results. The AP for a regular image ("Faster R-CNN 1.0x") is 33.3% and simple size reduction ("Faster R-CNN 0.5x") is 24.2% thereby significantly reducing the computation time from 220 ms to 78 ms. With SOTA ("Fovea (L:S_I) 0.5x") warping the corresponding numbers are 28.1% and 85.4 ms. Our method ("Two Plane Prior 0.5x") has a better AP of 30.8% than the SOTA with a somewhat longer computation time of 105 ms.

---

[1] A. Ghosh, "A Single View Perspective Prior for Efficient Detection", Master's Thesis, Tech. Report, CMU-RI-TR-23-42, August, 2023 https://www.ri.cmu.edu/app/uploads/2023/08/Anurag_MSR_Thesis.pdf

[2] Ghosh, Anurag, N. Dinesh Reddy, Christoph Mertz, and Srinivasa G. Narasimhan. "Learned Two-Plane Perspective Prior based Image Resampling for Efficient Object Detection." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13364-13373. 2023. https://openaccess.thecvf.com/content/CVPR2023/html/Ghosh_Learned_Two-Plane_Perspective_Prior_Based_Image_Resampling_for_Efficient_Object_CVPR_2023_paper.html

[3] Camden Cummings, Anurag Ghosh, and Christoph Mertz, "Detection and Tracking of Accessibility Challenges", RISS journal, 2023, to be published. https://riss.ri.cmu.edu/research_showcase/working-papers-journals/

[4] Philip Neugebauer, "Domain Adaptation by Revisiting Static Objects with a Transit Bus", Master's Thesis in Informatics, Technische Universität München, 2023.

Details of our method and results can be found in the master thesis[1] and paper[2].

Method	Scale	AP	AP_S	AP_M	AP_L	Latency (ms)
Faster R-CNN	0.5x	24.2	4.9	29.0	50.9	78.4 ± 1.8
Fovea (S_D) [50]	0.5x	26.7	8.2	29.7	54.1	83 ± 2.5
Fovea (S_I) [50]	0.5x	28.0	10.4	31.0	**54.5**	85 ± 2.7
Fovea (L:S_I) [50]	0.5x	28.1	10.3	30.9	54.1	85.4 ± 2.7
Two-Plane Pr. (Pseudo.)	0.5x	27.1	9.8	28.9	50.2	104.5 ± 8.5
Two-Plane Prior	0.5x	**30.8**	**14.5**	**31.6**	52.9	105 ± 8.5
Baseline at higher scales						
Faster R-CNN	0.75x	29.2	11.6	32.1	53.3	142 ± 2.5
Faster R-CNN	1.0x	33.3	16.8	34.8	53.6	220 ± 1.7

Table 1 Evaluation on Argoverse-HD: Two-Plane Prior outperforms both SOTA's dataset-wide and temporal priors in overall accuracy. Our method improves small object detection by +4.1APS or 39% over SOTA.

CVPR2023 AVA Accessibility Vision and Autonomy Challenge - Segmentation Track

We participated in the "CVPR2023 AVA Accessibility Vision and Autonomy Challenge - Segmentation Track"[5]. The goal of this challenge is to provide vision-based benchmarks and methods relevant to accessibility (e.g., people with disabilities and mobility aids are currently mostly absent from large-scale datasets in pedestrian detection). The benchmark dataset consists of synthetic images with semantic annotation (Figure 4).

Figure 4: An example from the instance segmentation challenge for perceiving people with mobility aids[6].

[5] https://eval.ai/web/challenges/challenge-page/1998/overview
[6] https://accessibility-cv.github.io/index.html#challenge

Rank	Participant team	AP (↑)	AP_50 (↑)	AP_75 (↑)	AP_small (↑)	AP_medium (↑)	AP_large (↑)	AP_person (↑)	AP_person-wc (↑)
1	hsslab_inspur	69.79	89.69	80.30	62.82	74.80	75.70	83.44	75.04
2	DeepblueAI	59.96	84.23	69.66	48.61	67.01	70.12	76.30	60.89
3	ZCX	57.06	81.64	66.42	44.47	65.35	69.48	74.44	56.97
4	STAR	55.46	80.02	65.10	42.87	60.28	67.74	74.52	55.61
5	UD-DREAL (method 9)	52.68	79.46	59.66	38.06	59.29	67.67	71.66	50.78
6	Navlab	41.98	69.96	41.33	14.66	60.07	85.35	59.00	37.74
7	FraunhoferIOSB_Team_Segmentation	40.98	67.30	41.57	21.69	49.73	65.34	63.63	43.76
8	AVA_Team_Segmentation (Test) B V	24.08	49.45	19.93	10.92	26.94	40.97	47.93	11.57
9	andromeda	16.54	37.96	12.05	6.22	18.22	27.60	38.15	8.76
10	SF	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 2 Final leaderboard of the AVA challenge[7]. Our group name is "Navlab" and the top score we achieved in the AP_large category is underlined in red.

We successfully participated in the challenge, overall we ranked in the middle, but in one category (AP_large) we achieved the best result (see Table 2).

Detection and tracking of vulnerable people

Figure 5 Tracking of a person inside the transit bus.

After detecting vulnerable people (see previous section) the next step is to track them. For detection we used the mmDetection[8] implementation of Mask2Former[9]. We trained it with 20928 synthetic images (see previous section) and 961 real-world labeled images from OpenImages version 6. The model detected correctly 41% of the time. We tested tracking with the Hungarian

[7] https://eval.ai/web/challenges/challenge-page/1998/leaderboard
[8] https://github.com/open-mmlab/mmdetection
[9] Cheng, Bowen, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. "Masked-attention mask transformer for universal image segmentation." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 1290-1299. 2022.

method and with DeepSORT[10] and found that DeepSORT was more reliable. Figure 5 shows an example of a person tracked inside the transit bus. Details about the detection and tracking can be found in the paper[3].

Domain Adaptation by Revisiting Static Objects

End of 2022 and early 2023 we had a student visitor in our lab. He used our bus data for his master's thesis[4]. This thesis introduces a novel method that addresses the challenge of sparsity of labeled data by automatically labeling static objects in images under new domains. The approach leverages a sparse set of labels from one domain and unlabeled images from new domains captured in similar locations. The method works by locating a labeled object, revisiting the same object under a different domain, and transferring the label to the new domain. Figure 6 shows how a new image of an object labeled in another image is registered with the labeled image and the annotation is transferred:

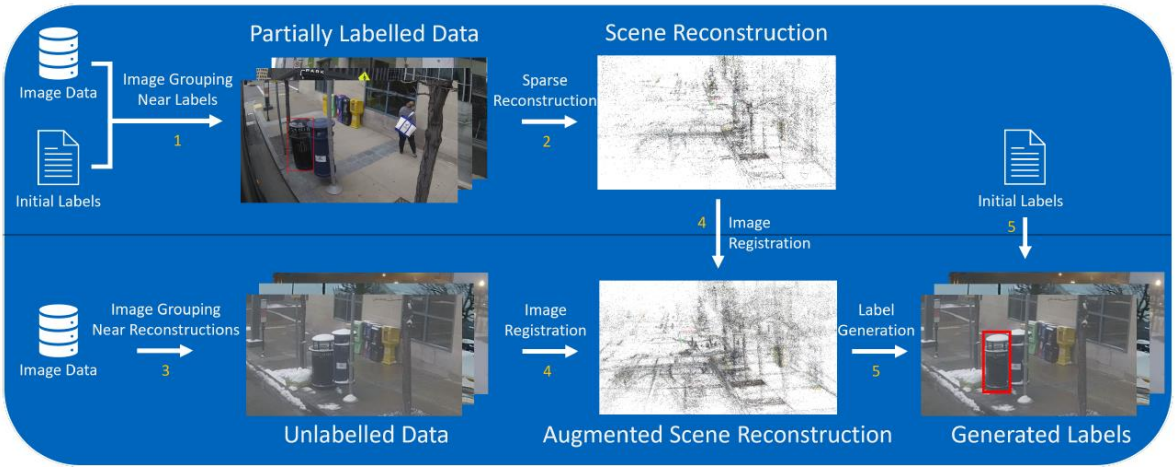

Figure 6: Illustration of the method to automatically generate labels for images taken of the same object at a different time.

1. Partially labeled images are split into groups of images that are nearby a group of labels.
2. A sparse reconstruction is created for each group of labeled objects that helps to relocate unlabeled nearby images.
3. Images from unlabeled data get grouped based on their proximity to a group of labels, using the GPS data.
4. Using the sparse reconstructions created in step 2. and the grouped images corresponding to the sparse reconstructions, the 3D positions of feature points of the unlabeled images are determined.
5. Using the 3D positions of feature points inside the given labels and their corresponding position in the unlabeled images, new labels are generated.

Table 3 shows the result of training only with the original hand-labeled data and training with the additional labels generated with the method above. The results show that adding generated data to the hand-labeled data improves detection performance in most cases.

[10] N. Wojke, A. Bewley and D. Paulus, "Simple online and realtime tracking with a deep association metric," 2017 IEEE International Conference on Image Processing (ICIP), 2017, pp. 3645-3649, doi: 10.1109/ICIP.2017.8296962.

Train Dataset \ Test Dataset	$\text{Cloudy}_{\text{Test}}$	$\text{Snowy}_{\text{Day}}$	$\text{Snowy}_{\text{Night}}$	RoadBotics
Sunny	38.4	22.6	29.4	23.7
$\text{Sunny+Generated}_{\text{snowy}}$	40.5	22.6	**32.5**	30.2
$\text{Sunny+Generated}_{\text{snowy,unique}}$	39.2	22.9	28.3	26.2
$\text{Sunny+Generated}_{\text{snowy,reviewed}}$	40.2	22.5	30.7	28.3
$\text{Sunny+Generated}_{\text{all domains}}$	**42.5**	**23.6**	27.8	**30.4**

Table 3 Average precision after training on hand-labeled + generated data and tested on various datasets.

Outlook

Our aim with this study was to begin to characterize actions and events that occur within the context of both safety equipment and ridership. We began analyzing the captured data to identify people and situations of interest including finding and utilizing available seating areas, interacting with safety equipment (with or without assistance), and entering and exiting the bus.

To date, we have been able to extract events and situations of interest based on the identification of people who may need assistance while traveling on the bus. This has been primarily due to motor impairments, mobility aids, personal transportation devices, or other encumbrances such as baggage. We have begun to use these data to train models that recognize potential access challenges or moments within a journey when accessibility may be impacted. For example, while searching for users with motor disabilities we encountered other situational impairments that may necessitate assistance. The detection algorithms show promise in identifying situations that may warrant further inspection or examination of prevalence across different situations over time. Further analysis is needed to extract full journeys for passengers of interest.