Impacts of Synthetically Generated Data on Trackformer-based Multi-Object Tracking

Matthew Lee, Clayton Harper, William Flinchbaugh,
Eric C. Larson, and Mitchell A. Thornton
Darwin Deason Institute for Cyber Security
Southern Methodist University
Dallas, Texas, USA
{leemh, caharper, wflinchbaugh, eclarson, mitch}@smu.edu

***Abstract*—As the scale of deep learning tasks continues to expand, the generation of sufficiently large datasets has become increasingly costly and time-consuming. In particular, resource demands for manual annotations for computer vision tasks such as multi-object tracking have contributed to the growing popularity of synthetic computer vision datasets created through simulation engines. Simulations facilitate the creation of automatically annotated datasets with complete control over environmental variables that are typically uncontrollable in real-world scenarios. Leveraging this control, we generate multi-object tracking datasets isolating specific environmental variables including subject scale, camera movement, and lighting changes. Our evaluation focuses on the TrackFormer architecture, an end-to-end, transformer-based solution designed for multi-object tracking. The resulting insights into how each environmental variable affects multi-object tracking performance can guide future architectural improvements. Furthermore, our data generation process can serve as a template for evaluating deep learning architectures in simulated environments.**

***Index Terms*—synthetic data, simulation, multi object tracking, machine learning, computer vision**

I. Introduction

One major challenge in computer vision, and deep learning in general, is generating datasets of sufficient size, diversity, and annotation quality to train complex deep learning models. Real-world computer vision datasets can be expensive and time consuming to create since they often require a human annotator to manually annotate each image, sometimes on the pixel level [41]. This challenge has given rise to synthetic computer vision datasets which leverage powerful modern graphics engines [1]–[3], [5], [9]. Graphics simulations provide several advantages. Low cost pixel-perfect annotations can be produced ensuring precise ground truth labeling for training and evaluation. Moreover, simulation environments can be tailored to specific subject domains that may be otherwise inaccessible in the real world. Simulations can additionally provide complete control over environmental factors such as lighting and weather conditions that can be difficult to maintain or reproduce.

Much of the existing and current research in synthetic data for computer vision seeks to narrow the domain gap between simulated worlds and the real world in terms of fidelity and diversity [10]–[14]. However, there is an opportunity to leverage the extra control offered by a simulated environment to isolate and investigate the effects of certain environmental variables on a computer vision model.

In a simulated environment, we are able to parameterize certain environmental variables to create multiple datasets across which only a select set of variables are changed, minimizing the impact of any confounding factors. This methodology enables the ability to gain deeper insights on the impact of specific environmental factors on model performance—a challenging task for limited real-world data.

We evaluate multi-object tracking performance across a variety of synthetically produced datasets using the *TrackFormer* architecture [6]. TrackFormer is a trainable, end-to-end multi-object tracking architecture that utilizes self-attention when tracking objects from frame to frame. Although TrackFormer has seen recent success in the MOTs challenge [15], [16], it is unclear how well the architecture works under varying environmental conditions. We seek to understand the resilience and shortcomings of the TrackFormer architecture under previously unexplored conditions. These conditions are created through the manipulation of several environmental variables including:

- **Subject scale**: Proportional size of objects relative to the screen.
- **Camera movement**: The transportation of the camera from one position to another while observing a scene.
- **Lighting Changes**: How much the illumination of a subject changes over time.

In existing datasets, it can be difficult to isolate these variables or find sufficient examples of each at various levels. In a simulated environment, each can be controlled independently. For data generation, we use common road vehicles simulated using a custom annotation tool built in the *Presagis* software suite [56], [57]. We evaluate TrackFormer performance using standard multi-object tracking metrics such as multiple object tracking accuracy (MOTA) and IDF1 score. [32], [53].

979-8-3503-5952-7/23/$31.00 ©2023 IEEE

Fig. 1. Simulated video frame in Vega Prime with automatic annotation

II. Related Work

A. Synthetic Data in Computer Vision

Advances in 3D graphics rendering and the development of software such as Unreal Engine [55] and Unity [54] have enabled projects such as [2], [3], [17] to provide the means to create synthetic datasets for a variety of computer vision tasks. The flexibility and customization of modern graphics engines allows for the creation of datasets from a wide range of domains and tasks such as object detection/tracking [9], [18], [20], pose estimation [21], [22], lighting estimation [23], [24], and robotics [25], [26].

Despite the success of synthetic data for computer vision tasks, a domain gap between real and synthetic data persists. Many studies (such as [1]) evaluate the effects of fidelity and lighting on computer vision models to elucidate which aspects of a simulation contribute to this disparity. One method for bridging this gap is domain randomization [10], [11]. Domain randomization aims to relax contextual and photo realism in favor of enhanced randomness and sample diversity. Consider an object detection task on identifying cars. Rather than creating a simulation in which a car is as photorealistic as possible with accurate lighting, texture, movement, and position, a domain randomization approach would involve placing the car in random positions, orientations, and textures. This approach exposes the network to more diverse samples of cars and helps guide the network away from learning spurious data artifacts that may be present in a real-world dataset in which the diversity of subjects is more limited. While our approach also involves adjusting simulation variables potentially beyond realism, our goal is to evaluate the impact on multi-object tracking performance rather than solely enhancing overall model performance.

B. Multi-Object Tracking

The primary objective of multi-object tracking is to associate object detections across video frames while maintaining identities for objects as they move and interact with the scene. This differs from object detection systems like the *YOLO* architecture where objects are detected and identified in a single, still frame [33]. Historically, approaches have utilized *tracking-by-detection* where a two-stage architecture is employed [34]–[37]. First, objects are detected in individual frames. Subsequently, associations between the detected objects are established between frames in order to track over time. While these methods have been effective, the disconnected nature of the two stages does not allow for joint optimization of detection and tracking.

C. TrackFormer

TrackFormer provides an end-to-end architecture that jointly trains detection and tracking through the use of self-attention. The architecture is based on the encoder-decoder Transformer [38] architecture implementing a *tracking-by-attention* paradigm. The TrackFormer architecture is auto-regressive where frames are processed by a common CNN backbone producing frame-level features. The features are encoded with self-attention with the Transformer encoder and queries are decoded with the Transformer decoder. Queries are mapped to box and class predictions using a multi-layer perceptron (MLP).

While recent work has investigated synthetic data in multi-object tracking [39], [40], a comprehensive analysis of the impacts of environmental variables on TrackFormer has not been systematically conducted. This work aims to address this gap by leveraging synthetic data to thoroughly examine the impacts of different variables.

III. Experiments

The goal of this work is to evaluate TrackFormer's performance in contexts different from those in which it was originally trained and evaluated. The results of these evaluations provide insights into the strengths and weaknesses of the TrackFormer architecture and guide future improvements and modifications. We systematically generate datasets with various settings to evaluate model performance under a diverse range of environmental conditions. In order to best isolate these environmental variables and have the most control over our datasets, we used 3D simulation software to create synthetic datasets for our evaluations. The following section will describe the dataset generation and training methodology we used for these experiments.

A. Synthetic Dataset Generation

In order to generate our synthetic datasets, we used the Presagis software suite, specifically Vega Prime [56] and STAGE [57]. STAGE is a tool for simulating the behavior of a variety of entities including ground vehicles, aircraft, and personnel in both civilian and defense scenarios. STAGE allowed us to create repeatable scenarios in which we could collect data. We used two locations that are readily available in the Presagis common database for our scenarios: (1)Camp Pendleton and (2)Yemen environments. Vega Prime is a visualization tool which allowed us to render a 3D visualization of our scenarios created in STAGE. We created a custom application in Vega Prime allowing the extraction of annotations in the form of semantic segmentations and bounding boxes from a virtual observer in our simulated environment (Fig. 1). To ensure compatibility with the evaluation framework of TrackFormer, the annotations were processed to closely align with the MOTChallenge format [15], [16] used for TrackFormer's initial training and evaluation.

Fig. 2. MOTChallenge [16] frame vs. a frame from our datasets. Despite the domain differences, we achieve strong performance on our synthetic datasets.

We created twelve scenarios to serve as the base for our experiments. These scenations consisted of two classes of common street vehicles in various settings: a black sedan and a white van. We chose these vehicles due to their distinctive color, size, and shape, as we wanted any variation in performance to come from the environmental variables rather than from a particularly difficult or inconsistent tracking subject. The scenarios featured simple traffic movement with our vehicles moving $\sim$10 meters per second, making turns and avoiding collisions. The base scenarios from which we would vary our environmental variables were designed to be somewhat similar to those seen in the MOTChallenge datasets in an attempt to maximize TrackFormer performance in our virtual domain. All base scenarios are about 40 seconds long, captured at 25 frames per second from a stationary virtual camera, and each scenario features clearly visible and illuminated objects. We captured frames at 1280x736 pixel resolution to match model input dimensionality. From these base scenarios, we generated a dataset of $\sim 11,000$ frames which serves as our baseline dataset.

After creating our base scenarios, we developed variants in which we carefully altered certain environmental variables. All scenarios were configured with the same amount of visual fidelity and detail in order to isolate environmental variables. For this work, we chose the following environmental variables to investigate: subject scale, camera movement, and lighting variation.

1) Subject Scale: Subject scale refers to the size of an object in a given image. As objects get smaller, they are represented by fewer pixels, which means there is less information to detect them. Related work has demonstrated the importance of subject scale in various computer vision tasks, which motivates our choice to use it as an environmental variable in this work [42]–[44]. We varied the subject scale in our scenarios by progressively repositioning the camera at increasing distances from the scene objects. We collected data at 10 roughly exponentially increasing increments from 25 meters to 1000 meters with bounding boxes ranging from as small as 2 px to 20,000 px, which results in a dataset of $\sim 100,000$ frames. An alternative approach is to directly scale the sizes of the 3D models of the scene subjects, holding the virtual camera position constant. However, we chose to move the virtual camera directly to create a more realistic background, as surrounding objects will be similarly scaled.

2) Camera Movement: Camera movement refers to the movement of the camera relative to the rest of the scene. While many cameras have pan-tilt-zoom capabilities which present their own challenges for object detection and tracking tasks [48], [49], we instead address the case in which the camera changes locations [45]–[47]. We do not introduce any camera jitter or instability. In our experiments, we evaluate the effects of camera movements at 5 different speed subsets from 0 to 40 meters per second resulting in a dataset of $\sim 50,000$ frames.

3) Lighting Changes: Lighting changes refers to how much the illumination of a subject changes throughout a scenario. Lighting changes which significantly change the appearance of an object could lead to missed or incorrect detections reducing tracking performance. Since lighting conditions can vary widely in a single location, especially outdoors, resilience to lighting changes is important for many object tracking systems [50]–[52]. In order to create datasets which include these lighting changes, we alternated the global illumination setting from full daytime lighting to low light conditions mimicking dusk. We used 5 different time interval subsets from 1 to 20 seconds between lighting changes, resulting in a dataset of $\sim 50,000$ frames. To create the most drastic and challenging effect, we made the lighting changes instant rather than a gradual shift.

We collected data for each of these environmental variables across our 12 base scenarios to create our three environmental variable datasets which we used to perform our main evaluation.

B. Evaluation Procedure

In our general approach to evaluating TrackFormer's resilience to different environmental variables, we split our experiments into three phases: a baseline evaluation, environmental variable evaluations with no tuning, and environmental variable evaluations with tuning. Since TrackFormer was originally trained and evaluated on real-world MOTChallenge data, our first phase trains TrackFormer on our base scenario to

establish baseline performance on synthetic data. This initial phase evaluates whether or not we are able to achieve strong performance in our simulated domain. Strong baseline performance ensures that any performance variation in the following phases is due to the environmental variables and not poor or inconsistent overall performance in the simulated domain. In our second phase, we evaluate the baseline model trained in the first phase using our environmental variable datasets. This phase serves to evaluate the resilience of TrackFormer to unfamiliar conditions. Finally, we tune our baseline model with each of our environmental variable datasets and perform the same evaluations to see whether TrackFormer is able to adapt to those same conditions.

C. Training Procedure

We follow a similar training procedure to prior work [6]. The learning rates for the encoder-decoder were set to 0.0002 and 0.00002, respectively. We trained the baseline model for 20 epochs on the base scenarios. For each environmental variable dataset, we tune the baseline model for an additional 20 epochs. Thus two models are created, a "baseline" and a "tuned" model to support the investigations described in our evaluation procedure. All training procedures were conducted on an NVIDIA SuperPOD at the SMU high performance computing center. The nodes in the center consist of 8 NVIDIA A100 GPUs, each equipped with 80 GB of memory. In running our experiments, we tune our models for 20 epochs spanning approximately 36 hours.

IV. Results

A. Evaluation Metrics

We evaluate our models with standard performance metrics for multi-object tracking. In addition to conventional precision and recall, we use the multi-object tracking specific variants: ID precision, ID recall, and IDF1 score. These measures take the nuances of multi-object tracking into account, such as periods of frequent ID switching, which can cause inconsistent values for traditional precision and recall scores [32]. We also use multiple object tracking accuracy (MOTA) [53], which summarizes object identification errors including mismatched IDs, false positives, and misses, while ignoring bounding box placement errors. Overall performance for each model is reported using these six metrics. To investigate performance trends with respect to environmental variables, we focus on presentation of IDF1 and MOTA for visual clarity and because these measures tend to capture the overall performance trends.

B. Baseline

Table I shows the evaluation results for our baseline experiment. Our baseline model shows that despite working in a completely different domain (see Fig. 2 virtual vehicles vs. real pedestrians) we can achieve strong performance. The capability to perform well in the synthetic domain is crucial as it establishes a foundation to compare TrackFormer under various synthetic conditions. We observe that most scores are greater than 0.75, with an overall MOTA score of about 0.81. Thus we seek to understand how environmental variations shift these scores through our experiments.

TABLE I
Baseline Results

Model	IDF1	IDP	IDR	Rcll	Prcn	MOTA
Baseline	0.784	0.760	0.810	0.941	0.882	0.812

C. Subject Scale

For our first variation, our datasets included a wide range of subject scales. Table II shows performance across the entire subject scale dataset for the baseline and tuned models, where we see a notable improvement across all metrics for the tuned model. Fig. 4 shows performance of both models in terms of average bounding box size for each of our 10 subject scale subsets. We see a steady drop in performance for the baseline model, whereas the tuned model seems to have a slight knee around 110 pixels^2 in area. The tuned model actually achieves a worse MOTA score than the tuned model for the very small objects. It may be that tuning the model with these small objects caused it to over-detect, leading to more false positives and thus a negative MOTA, while the baseline model under-detected. Given the range of subject scales present in the dataset (Fig. 3), this performance trend is roughly what we expected to see from this experiment. This result indicates that TrackFormer may not generalize to datasets containing objects of different sizes than the training set without additional tuning. Moreover, even a tuned model may not provide sufficient performance detecting smaller objects. Thus an aim for future research should be to investigate methods to help increase performance when larger and smaller objects may be present in a scene together.

TABLE II
Subject Scale Results

Model	IDF1	IDP	IDR	Rcll	Prcn	MOTA
Baseline	0.372	0.389	0.361	0.483	0.527	0.228
Tuned	0.624	0.620	0.627	0.684	0.676	0.368

D. Camera Movement

In our camera movement experiment, we also see a significant improvement in performance for the tuned model vs. the baseline model (Table III). In Fig. 5, we see a slight downward trend in performance as the camera speed increases for both models, however there are notable differences in the 0 to 10 m/s range. We see the baseline model experience a drop in performance from 0 to 5 m/s, but then plateau at 10, 20, and 40 m/s. This initial dip in performance is expected as the base data on which the baseline model was trained was all from a stationary camera, making the difference between the 0 m/s subset and the 5 m/s more drastic (*i.e.*, stationary to non-stationary camera). For the rest of the subsets, the differences likely stem from how much the subjects change position between captured frames, shown in Fig. 6, as the

Fig. 3. Subset of different scales present in our subject scales dataset.

camera followed the same path in each subset, but at different speeds. For the tuned model, we see that the tuning appears to rectify the initial drop in performance; however, we still see the same drop in performance after 10 m/s. This presents itself as a potential area for improvement for the TrackFormer architecture, as we see a similar trend for both baseline and tuned models.

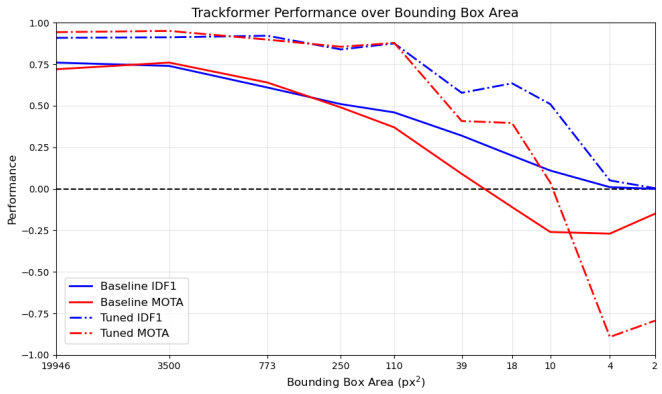

Fig. 4. MOTA and IDF1 scores vs. mean bounding box size in pixels for baseline and tuned TrackFormer, note that the x axis is a log scale.

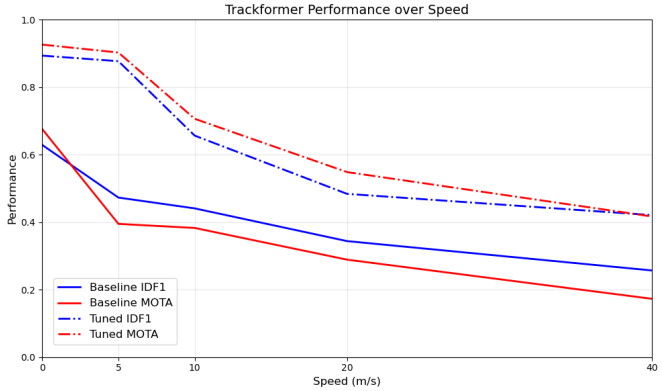

Fig. 5. MOTA and IDF1 scores vs. camera movement speed in m/s for baseline and tuned TrackFormer.

TABLE III
CAMERA MOVEMENT RESULTS

Model	IDF1	IDP	IDR	Rcll	Prcn	MOTA
Baseline	0.429	0.467	0.401	0.612	0.726	0.383
Tuned	0.667	0.670	0.664	0.845	0.856	0.700

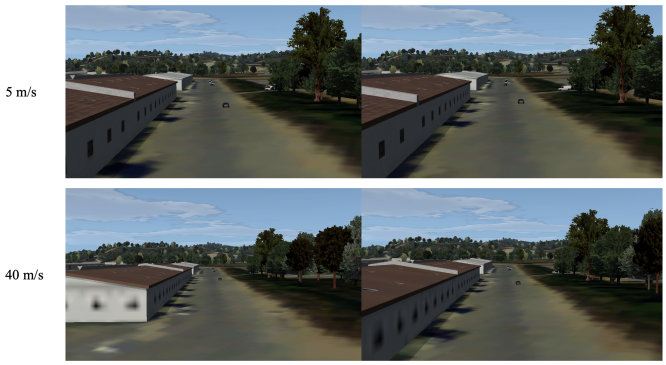

Fig. 6. Difference in frames over 1 second for 5 m/s (top) and 40 m/s (bottom).

E. Lighting Changes

For our lighting changes experiment, we expected to see a noticeable drop in performance as the interval between changes decreased—that is, the subjects switched between being well- and poorly-illuminated more often. However, we see in Fig. 7 that performance for both the baseline and tuned models remains fairly high. We expected detections in poor lighting conditions to be more difficult to identify. Therefore, we anticipated a reduction in tracking performance during poor illumination periods. However, we do not see the clear trends we see in the other experiments with illumination variances. This is most likely due to the backbone CNN of the TrackFormer architecture being more resilient to lighting conditions than we expected. We also see that the baseline model performance is almost constant across all subsets, whereas the tuned model experiences a dip from 1s to 5s, then improves from 5s to 20s. This indicates that the model has more difficulty learning these intermediate intervals, although performance is still improved slightly over the baseline model. It may be that the model learned to handle quick lighting

changes and long periods of different lighting conditions, but was unable to learn to bridge the gap over the tuning period. Despite the anomaly in the tuned model's performance, we see that overall lighting changes have little effect on tracking performance.

TABLE IV
LIGHTING CHANGES RESULTS

Model	IDF1	IDP	IDR	Rcll	Prcn	MOTA
Baseline	0.789	0.778	0.800	0.898	0.874	0.764
Tuned	0.879	0.877	0.882	0.937	0.932	0.896

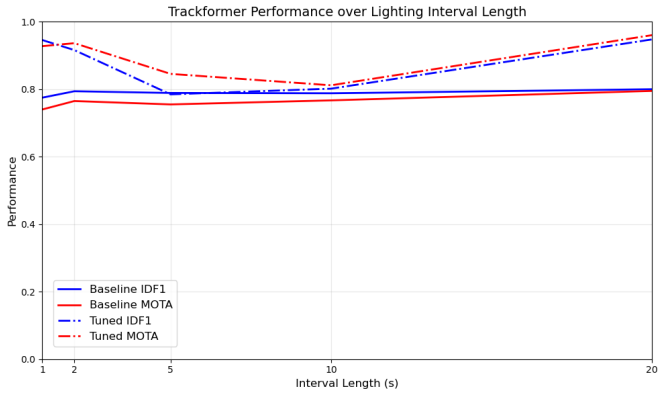

Fig. 7. MOTA and IDF1 scores vs. lighing change interval in seconds for untuned and tuned TrackFormer.

Fig. 8. Difference in lighting conditions for lighting changes datasets.

V. CONCLUSION

Synthetic data can be a valuable tool for evaluating a neural network architectures. The ability to generate automatic annotations and simulate various conditions allows the creation of specific datasets to test different aspects of an architecture. In this work we demonstrate how to apply this synthetic-data-driven evaluation approach to the TrackFormer architecture—testing its resilience to changes in three environmental variables: subject scale, camera movement, and lighting changes. We found negative performance trends when decreasing subject scale and increasing camera movement, which could motivate modifications and improvements to the TrackFormer architecture.

REFERENCES

[1] M. Mousavi, A. Khanal, and R. Estrada, "AI Playground: Unreal Engine-based Data Ablation Tool for Deep Learning," 2020, doi: 10.48550/ARXIV.2007.06153.

[2] P. Martinez-Gonzalez et al., "UnrealROX+: An Improved Tool for Acquiring Synthetic Data from Virtual 3D Environments," 2021, doi: 10.48550/ARXIV.2104.11776.

[3] S. Borkman et al., "Unity Perception: Generate Synthetic Data for Computer Vision," 2021, doi: 10.48550/ARXIV.2107.04259.

[4] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison, "SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation?," in 2017 IEEE International Conference on Computer Vision (ICCV), Venice: IEEE, Oct. 2017, pp. 2697–2706. doi: 10.1109/ICCV.2017.292.

[5] G. Paulin and M. Ivasic-Kos, "Review and analysis of synthetic dataset generation methods and techniques for application in computer vision," Artif Intell Rev, Jan. 2023, doi: 10.1007/s10462-022-10358-3.

[6] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "TrackFormer: Multi-Object Tracking with Transformers," 2021, doi: 10.48550/ARXIV.2101.02702.

[7] P. Zhou, B. Ni, C. Geng, J. Hu, and Y. Xu, "Scale-Transferrable Object Detection," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT: IEEE, Jun. 2018, pp. 528–537. doi: 10.1109/CVPR.2018.00062.

[8] Y. Li, Y. Chen, N. Wang, and Z.-X. Zhang, "Scale-Aware Trident Networks for Object Detection," in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South): IEEE, Oct. 2019, pp. 6053–6062. doi: 10.1109/ICCV.2019.00615.

[9] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual Worlds as Proxy for Multi-Object Tracking Analysis," 2016, doi: 10.48550/ARXIV.1605.06457.

[10] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC: IEEE, Sep. 2017, pp. 23–30. doi: 10.1109/IROS.2017.8202133.

[11] J. Tremblay et al., "Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT: IEEE, Jun. 2018, pp. 1082–10828. doi: 10.1109/CVPRW.2018.00143.

[12] R. Liu, C. Yang, W. Sun, X. Wang, and H. Li, "StereoGAN: Bridging Synthetic-to-Real Domain Gap by Joint Optimization of Domain Translation and Stereo Matching," 2020, doi: 10.48550/ARXIV.2005.01927.

[13] A. Prakash et al., "Structured Domain Randomization: Bridging the Reality Gap by Context-Aware Synthetic Data," in 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada: IEEE, May 2019, pp. 7249–7255. doi: 10.1109/ICRA.2019.8794443.

[14] M. Maximov, K. Galim, and L. Leal-Taixe, "Focus on Defocus: Bridging the Synthetic to Real Domain Gap for Depth Estimation," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA: IEEE, Jun. 2020, pp. 1068–1077. doi: 10.1109/CVPR42600.2020.00115.

[15] P. Dendorfer et al., "MOT20: A benchmark for multi object tracking in crowded scenes," 2020, doi: 10.48550/ARXIV.2003.09003.

[16] P. Dendorfer et al., "MOTChallenge: A Benchmark for Single-Camera Multiple Target Tracking," 2020, doi: 10.48550/ARXIV.2010.07548.

[17] M. Müller, V. Casser, J. Lahoud, N. Smith, and B. Ghanem, "Sim4CV: A Photo-Realistic Simulator for Computer Vision Applications," Int J Comput Vis, vol. 126, no. 9, pp. 902–919, Sep. 2018, doi: 10.1007/s11263-018-1073-7.

[18] J. Shermeyer, T. Hossler, A. Van Etten, D. Hogan, R. Lewis, and D. Kim, "RarePlanes: Synthetic Data Takes Flight," 2020, doi: 10.48550/ARXIV.2006.02963.

[19] T. Sun et al., "SHIFT: A Synthetic Driving Dataset for Continuous Multi-Task Domain Adaptation," in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA: IEEE, Jun. 2022, pp. 21339–21350. doi: 10.1109/CVPR52688.2022.02068.

[20] M. Fabbri et al., "MOTSynth: How Can Synthetic Data Help Pedestrian Detection and Tracking?," 2021, doi: 10.48550/ARXIV.2108.09518.

[21] G. Varol et al., "Learning from Synthetic Humans," 2017, doi: 10.48550/ARXIV.1701.01370.

[22] A. Sengupta, I. Budvytis, and R. Cipolla, "Synthetic Training for Accurate 3D Human Pose and Shape Estimation in the Wild," 2020, doi: 10.48550/ARXIV.2009.10013.

[23] P. Kán and H. Kafumann, "DeepLight: light source estimation for augmented reality using deep learning," Vis Comput, vol. 35, no. 6–8, pp. 873–883, Jun. 2019, doi: 10.1007/s00371-019-01666-x.

[24] F. Einabadi, J. Guillemaut, and A. Hilton, "Deep Neural Models for Illumination Estimation and Relighting: A Survey," Computer Graphics Forum, vol. 40, no. 6, pp. 315–331, Sep. 2021, doi: 10.1111/cgf.14283.

[25] Y. Lin, C. Tang, F.-J. Chu, and P. A. Vela, "Using Synthetic Data and Deep Networks to Recognize Primitive Shapes for Object Grasping," in 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France: IEEE, May 2020, pp. 10494–10501. doi: 10.1109/ICRA40945.2020.9197256.

[26] T. Lips, V.-L. De Gusseme, and F. wyffels, "Learning Keypoints from Synthetic Data for Robotic Cloth Folding," 2022, doi: 10.48550/ARXIV.2205.06714.

[27] F. Baldassarre and H. Azizpour, "Explainability Techniques for Graph Convolutional Networks," 2019, doi: 10.48550/ARXIV.1905.13686.

[28] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization," 2016, doi: 10.48550/ARXIV.1610.02391.

[29] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning Important Features Through Propagating Activation Differences," 2017, doi: 10.48550/ARXIV.1704.02685.

[30] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," 2017, doi: 10.48550/ARXIV.1705.07874.

[31] C. Molnar, T. Freiesleben, G. König, G. Casalicchio, M. N. Wright, and B. Bischl, "Relating the Partial Dependence Plot and Permutation Feature Importance to the Data Generating Process," 2021, doi: 10.48550/ARXIV.2109.01433.

[32] Ergys Ristani, Francesco Solera, Roger S. Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In Eur. Conf. Comput. Vis. Workshops, 2016.

[33] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[34] Roberto Henschel, Laura Leal-Taixe, Daniel Cremers, and ´ Bodo Rosenhahn. Improvements to frank-wolfe optimization for multi-detector multi-object tracking. In IEEE Conf. Comput. Vis. Pattern Recog., 2017.

[35] Margret Keuper, Siyu Tang, Bjoern Andres, Thomas Brox, and Bernt Schiele. Motion segmentation & multiple object tracking by correlation co-clustering. In IEEE Trans. Pattern Anal. Mach. Intell., 2018.

[36] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M. Rehg. Multiple hypothesis tracking revisited. In Int. Conf. Comput. Vis., 2015.

[37] Laura Leal-Taixe, Gerard Pons-Moll, and Bodo Rosenhahn. ´ Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. Int. Conf. Comput. Vis. Workshops, 2011.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Adv. Neural Inform. Process. Syst., 2017.

[39] L. Li, C. Dai, Y. Xia and L. Svensson, "Deep Fusion of Multi-Object Densities Using Transformer," ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-5, doi: 10.1109/ICASSP49357.2023.10096214.

[40] Specker, Andreas, et al. "Improving multi-target multi-camera tracking by track refinement and completion." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

[41] C. Vondrick, D. Patterson, and D. Ramanan, "Efficiently Scaling up Crowdsourced Video Annotation: A Set of Best Practices for High Quality, Economical Video Labeling," Int J Comput Vis, vol. 101, no. 1, pp. 184–204, Jan. 2013, doi: 10.1007/s11263-012-0564-1.

[42] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI: IEEE, Jul. 2017, pp. 936–944. doi: 10.1109/CVPR.2017.106.

[43] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection," in Computer Vision – ECCV 2016, vol. 9908, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., in Lecture Notes in Computer Science, vol. 9908. , Cham: Springer International Publishing, 2016, pp. 354–370. doi: 10.1007/978-3-319-46493-0_22.

[44] F. Feng, B. Shen, and H. Liu, "Visual object tracking: in the simultaneous presence of scale variation and occlusion," Systems Science & Control Engineering, vol. 6, no. 1, pp. 456–466, Jan. 2018, doi: 10.1080/21642583.2018.1536899.

[45] M. Yazdi and T. Bouwmans, "New trends on moving object detection in video images captured by a moving camera: A survey," Computer Science Review, vol. 28, pp. 157–177, May 2018, doi: 10.1016/j.cosrev.2018.03.001.

[46] T. Chen and S. Lu, "Object-Level Motion Detection From Moving Cameras," IEEE Trans. Circuits Syst. Video Technol., vol. 27, no. 11, pp. 2333–2343, Nov. 2017, doi: 10.1109/TCSVT.2016.2587387.

[47] D. Zhou, V. Frémont, B. Quost, Y. Dai, and H. Li, "Moving object detection and segmentation in urban environments from a moving platform," Image and Vision Computing, vol. 68, pp. 76–87, Dec. 2017, doi: 10.1016/j.imavis.2017.07.006.

[48] J. Park, D. H. Kim, Y. S. Shin, and S. Lee, "A comparison of convolutional object detectors for real-time drone tracking using a PTZ camera," in 2017 17th International Conference on Control, Automation and Systems (ICCAS), Jeju: IEEE, Oct. 2017, pp. 696–699. doi: 10.23919/ICCAS.2017.8204318.

[49] D. Avola, L. Cinque, G. L. Foresti, C. Massaroni, and D. Pannone, "A keypoint-based method for background modeling and foreground detection using a PTZ camera," Pattern Recognition Letters, vol. 96, pp. 96–105, Sep. 2017, doi: 10.1016/j.patrec.2016.10.015.

[50] Kalpesh R Ranipa and Dr. Kiritkumar Bhatt, "Illumination Condition Effect on Object Tracking: A Review", GJCST, vol. 14, no. F5, pp. 9–13, Oct. 2014.

[51] G. Liu, S. Liu, K. Muhammad, A. K. Sangaiah, and F. Doctor, "Object Tracking in Vary Lighting Conditions for Fog Based Intelligent Surveillance of Public Spaces," IEEE Access, vol. 6, pp. 29283–29296, 2018, doi: 10.1109/ACCESS.2018.2834916.

[52] T. Caselitz, M. Krawez, J. Sundram, M. Van Loock, and W. Burgard, "Camera Tracking in Lighting Adaptable Maps of Indoor Environments," in 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France: IEEE, May 2020, pp. 3334–3340. doi: 10.1109/ICRA40945.2020.9197471.

[53] K. Bernardin and R. Stiefelhagen, "Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics," EURASIP Journal on Image and Video Processing, vol. 2008, pp. 1–10, 2008, doi: 10.1155/2008/246309.

[54] Unity, "Unity Technologies," 2023.

[55] Unreal engine, "Epic Games," 2023.

[56] Vega Prime, "Presagis," 2019.

[57] STAGE, "Presagis," 2022.