Densely Annotated Photorealistic Virtual Dataset Generation for Abnormal Event Detection

Densely Annotated Photorealistic Virtual Dataset Generation for Abnormal Event Detection

Rico Montulet and Alexia Briassouli(✉)

Department of Data Science and Knowledge Engineering, Maastricht University, Maastricht, The Netherlands
rico@montulet.nl, alexia.briassouli@maastrichtuniversity.nl

Abstract. Many timely computer vision problems, such as crowd event detection, individual or crowd activity recognition, person detection and re-identification, tracking, pose estimation, segmentation, require pixel-level annotations. This involves significant manual effort, and is likely to face challenges related to the privacy of individuals, due to the intrinsic nature of these problems, requiring in-depth identifying information. To cover the gap in the field and address these issues, we introduce and make publicly available a photorealistic, synthetically generated dataset, with detailed dense annotations. We also publish the tool we developed to generate it, that will allow users to not only use our dataset, but expand upon it by building their own densely annotated videos for many other computer vision problems. We demonstrate the usefulness of the dataset with experiments on unsupervised crowd anomaly detection in various scenarios, environments, lighting, weather conditions. Our dataset and the annotations provided with it allow its use in numerous other computer vision problems, such as pose estimation, person detection, segmentation, re-identification and tracking, individual and crowd activity recognition, and abnormal event detection. We present the dataset as is, along with the source code and tool to generate it, so any modification can be made and new data can be created. To our knowledge, there is currently no other photorealistic, densely annotated, realistic, synthetically generated dataset for abnormal crowd event detection, nor one that allows for flexibility of use by allowing the creation of new data with annotations for many other computer vision problems. **Dataset and source code available:** https://github.com/RicoMontulet/GTA5Event.

1 Introduction

The State of the Art (SoA) deep learning methods in computer vision achieve high accuracy by leveraging large, diverse and correctly annotated datasets, with the most detailed annotations desired being at a pixel level. For problems like crowd event detection, pose estimation, person detection, recognition, segmentation, re-identification, tracking, activity recognition, the production of detailed

Funded under the H2020 project MindSpaces, Grant number # 825079.

A. Del Bimbo et al. (Eds.): ICPR 2020 Workshops, LNCS 12664, pp. 5–19, 2021.
https://doi.org/10.1007/978-3-030-68799-1_1

ground truth requires great manual effort, is very time-consuming, error-prone and labor intensive. This is even more so the case in tasks requiring pixel-level accuracy, such as fine-grained activity recognition, pose estimation [9], person re-identification and tracking, as well as activity/event recognition. Moreover, the advent of privacy regulations such as GDPR (https://gdpr-info.eu/) has led to the removal of datasets of individuals that have not given explicit consent, and makes the creation of new annotated datasets challenging. The current Covid-19 related restrictions on large gatherings and crowds of people are posing additional obstacles to the creation of benchmark datasets. However, the need for data with high quality annotations is continuously increasing, for training data, or augmentation of existing training data. A solution to this problem that is gaining increasing attention is the creation of realistic synthetic datasets using commercial video game engines, for the creation of highly realistic data with dense, high quality annotations in varying lighting and environmental conditions.

In this work we create photorealistic videos using the Rockstar Advanced Game Engine (RAGE) in the video game GTA V [38], as it allows for the creation of densely annotated and very realistic datasets. Its license allows for this, and specifically states: "*The publisher of Grand Theft Auto V allows non-commercial use of footage from the game as long as certain conditions are met, such as non-commercial use and not distributing spoilers*" [2]. The engine provides great flexibility, allowing for the generation of videos with wide ranging, detailed and realistic activities of individuals and groups of people in different indoors and outdoors environments, lighting and weather conditions. Our dataset comprises of 54 videos with resolution of 2560×1440, from 54 unique locations. Each video has 450 frames recorded from a static camera at varying heights. Detailed ground-truth data is provided for every frame, comprising of weather conditions, time of day, person segmentation, bone coordinates, depth maps and the type of group of people. The videos are rendered at different frames per second to simulate different frame rates on common security cameras. In this work, we choose to apply the generated datasets to the problem of unsupervised, abnormal crowd event detection. The motivation for this is the long-standing lack of high quality, densely annotated data for this problem, as explained in Sect. 2. We demonstrate the usefulness of our dataset in experiments on unsupervised event detection, with annotations that can also be used for a number of other computer vision problems. To our knowledge, there is no other dataset providing annotations for such a wide range of vision problems.

This paper is structured as follows: Sect. 2 describes the related work on synthetic dataset generation, and the datasets available. Section 3 describes the process for generating our synthetic dataset, and the resulting annotations and Sect. 4 shows how it can be used for the successful, unsupervised detection of abnormal events in a variety of environments, while conclusions are drawn in Sect. 5.

2 Related Work

The role of synthetic datasets as supplements to existing training data, or as data in and of themselves, is receiving increased attention [36,37,43,44], as deep learning requires extensive high quality annotated data to perform well, which is not always easy to obtain in the real world. To this end, various synthetic datasets have been recently generated to solve different computer vision problems, with the works in [6,25,37,38,40,49] all using synthetic data. Several of them, namely [25,37,38,49] use the GTA V Rockstar Advanced Game Engine (RAGE), similarly to our work, but focusing on different computer vision problems. The reason for choosing RAGE is the high quality of the resulting graphics, as well as the policy of the game engine, which allows for non-commercial use of its footage [2]. Data generation tools and datasets that use virtual worlds to generate annotated image datasets are described below.

2.1 Related Tools for Synthetic Data Generation

CARLA. CARLA is an open-source platform from Intel for developing and testing autonomous driving systems [16] with various environments, sensors, and full control over data. Scene segmentation has been achieved using CARLA in [41], and LiDAR object detection uses CARLA in [17]. A challenge was also setup in 2019 with realistic driving scenarios, for autonomous driving benchmarks (https://carlachallenge.org/).

Unity ML. The Unity game engine allows the development of realistic virtual game environments for applications like Deep Learning, with Google's DeepMind recently having used it to train its deep learning models [1]. Unity ML [3,26] use machine learning agents to create realistic and varied environments.

Unreal. Unreal [34] is a game engine for virtual environments that also generates realistic images to train deep learning methods. It has been used for AirSim [42], a simulator for drones and autonomous vehicles, and other annotated datasets [33], to train deep learning methods for autonomous vehicles. It has also been used to generate a synthetic dataset for 3D object detection and pose estimation [45].

Blender. Blender, a tool generating 3D scenes for video games, has also been used for training data for computer vision. In [13], an open-source modular pipeline, presented for photorealistic 3D scenes and images, is tested on object segmentation. Medical imaging, and specifically robotic surgery has recently benefited from data created with Blender [10], with the paper receiving a best paper award in 2020.

Europilot. Europilot, an environment based on Euro Truck Simulator 2, simulates all aspects of a driving vehicle: acceleration, breaking, steering and collision detection etc. It also offers visual rendering of the scene for computer vision purposes and is used for training autonomous vehicles [20].

Grand Theft Auto 5. Grand Theft Auto 5 (GTA V) poses a different paradigm, as it generates very photorealistic video game data. One of the main reasons it is selected specifically for computer vision tasks is the excellent quality of its graphics. Another work that uses GTA V for generating photorealistic data is Richter et al. [37,38], who injected their own software inbetween the game and the graphics card, so as to collect information about geometry and textures. In contrast to [37,38], our tool uses native RAGE functions, which allows us to get the annotations directly from the game environment, to change the scene, set weather conditions, and customize the behavior of individuals and groups.

Table 1. Abnormal event detection datasets.

Dataset	# of frames	Resolution	Events
UCSD Ped1 [27], 2014	14000	238×158	Abnormal object in one frame
UCSD Ped2 [27], 2014	4560	360×240	Abnormal object in one frame
UMN [46], 2014	3855	320×240	Staged crowd events
Subway entrance [4], 2008	86535	512×382	Few abnormalities
Subway exit [4], 2008	38940	512×382	Few abnormalities
CUHK avenue [29], 2013	30652	640×360	Few abnormalities
Street scene [35], 2020	203257	1280×720	Few abnormalities in street behavior
Ours: GTAV event, 2020	24000	2560×1440	Limitless crowd events, abnormal crowd behaviors

2.2 Related Synthetic Data

A wide range of synthetically generated data has been produced for computer vision problems https://github.com/unrealcv/synthetic-computer-vision, but not for the application that we are examining, namely crowd event detection. Moreover, unlike our dataset, existing ones do not offer the flexibility to be used for several other applications, from recognition of individual or group activities/interactions in a variety of scenarios and environments, to person tracking, segmentation re-identification and others, as detailed below.

Synthetic Datasets for Optical Flow. Synthetic datasets have been used to develop accurate optical flow algorithms since 1987 [21], with Yosemite [7] (1994) being one of the most widely used. In [30], what makes a good synthetic dataset is described, with an extensive overview of existing synthetic optical flow benchmarking datasets, including recent ones like Flying Chairs [15], and SYNTHIA [39]. It should be noted that the SoA optical flow Flownet2 [24] used

in this work has also been developed using synthetically generated data, which allowed it to outperform the SoA.

Synthetic Human, Crowd Datasets. Previous efforts on synthetic data generation for crowd simulation [5,31] focus on crowd group dynamics, but not on the quality of the graphics, making them less appropriate for deep learning, which requires large amounts of high quality annotated data. Recently, a dataset and tool for crowd counting was published [48], which only generates crowd images, and not crowd videos, nor crowd event scenarios.

In [12], a dataset for human activity recognition has been generated, but with Unity, contains one person per activity, and no abnormal crowd events. Motion tracking and activity recognition can take place using the synthetic data in [19], however it features only one person per frame, and has a blank background.

Human segmentation and depth estimation datasets have also been synthetically generated recently [47], based on motion capture data. However, they use Blender [11], and comprise of single person images rather than continuous video. A large scale synthetic image dataset of images of street scenes with dense semantic segmentation maps, generated by the Unity game engine, is SYNTHIA. It has been used for semantic image segmentation, image-to-image translation [23,28] and adversarial domain adaptation [22], among others.

Our tool generates densely annotated crowds and events, but can also be used for the generation of individual or small group activities, tracking, pose, segmentation, providing solutions for a wider range of computer vision problems than existing datasets. At the same time, it provides densely annotated benchmarking data for abnormal crowd event detection, for which existing real-world datasets have been limited in size, quality and amount of events (see Sect. 2.3).

2.3 Real-World Datasets for Abnormal Crowd Event Detection

In this work we use our dataset for the problem of abnormal crowd event detection, although it can also be used for other problems, like person detection, segmentation, re-identification, tracking, individual or crowd activity recognition. We focus on crowd event detection due to the long-standing and well-documented lack of datasets for this problem. Existing datasets are small, of poor resolution, with few abnormal events and often with inconsistent annotations [35].

In Table 1 we present real-world datasets on crowd event detection, most often used for benchmark comparisons. The frequently encountered, UCSD pedestrian [27], shows pedestrians walking outdoors, with a few anomalies like a bike passing through them etc. It is small in size, contains a few abnormalities, and events are based on changes in a frame, rather than changes in behavior and motion. The University of Minnessotta (UMN) dataset [46] is even smaller, with 11 videos and 3 scenes, with simple, staged events. Only two long real-world videos, Subway and Mall, are presented in [4], with few, specific events, making them inadequate for robust testing. CUHK Avenue [29] contains data from a surveillance camera, with few anomalies, caused mostly by individual actions rather than crowd behaviors. Recently, StreetScene [35] was made available, containing a far larger number of frames at higher resolution, with the corresponding

annotations. However, this dataset does not contain crowd events or abnormal individual/crowd behaviors, as it focuses on abnormalities related to street scenes and rules, such as pedestrians crossing illegally or bicycles on sidewalks.

Our dataset features significantly more frames than most of the above datasets, with the exception of StreetScene, however the number of frames in our dataset can be directly increased by using our tool. Our dataset also features a wide range of weather conditions and environments, as well as events related to abnormalities in crowd behavior and motion, rather than anomalies related to appearance changes in one frame (as in UCSD, UMN). Thus it is more appropriate for detecting abnormalities in videos, in the behavior of crowds but also individuals. It does not contain annotations only for events, but also for person segmentation, tracking, re-identification, pose classification and more, detailed in Sect. 3.2. Its graphics are of high quality, as they have been generated with the GTA V engine [14], which can produce a wide range of high quality, photo-realistic scenes.

3 Dataset Creation and Description

3.1 Dataset Creation: Interacting with the Virtual World

Our method uses the plugin Scripthook, developed by Alexander Blade, that allows developers to interact with the Rockstar Advanced Game Engine (RAGE). The scene generation and data collection is done by a plugin written in C# that controls the environment using Scripthook. Virtual scenes, weather conditions, character models, lighting conditions, movement and behaviours can all be changed using a config file. In order to create a diverse dataset, different locations and camera placements needed to be explored. Conveniently, GTA V has a massive 252 square kilometer map with a vast amount of different locations. Every location is unique with a high degree of similarity to real world locations. From beaches to shopping malls to mountain ranges, the possibilities are vast.

In this work, a subset of 54 locations were used, where stationary cameras have been placed at various heights and angles, and a region of interest was selected in them. There are 704 unique person models available in the game, with different skin color, body shape (height, weight) and hair styles, with varied types of clothing for each person model. The people can move in several ways, such as walking, standing, crouching, standing having a conversation etc.

Groups of people of various sizes are spawned in the designated Region of Interest (ROI) for each location. These groups range in size from 5 to 25 people and there are 1 to 5 groups. These parameters can be changed in a configuration file to suit any needs. Every person within a group gets a specific task, whether to talk to one of their group members, wander around the scene, make a phone-call or just stand there. Once groups have been spawned the weather is randomly changed, the camera is set to render at a random FPS, and finally, the time of day is randomized. All of this happens while the game is paused. This is achieved by setting the timescale parameter in the game engine to zero. The locations along with group sizes, group locations, person locations, weather,

time and FPS information is available. Figure 1 provides an overview of the distribution of the generated actions, weather conditions, fps for each video, and times of day, showing that our solution can cover numerous scenarios. There are 24,301 frames in total, with 1,825,493 annotated bounding boxes and a total of 177,372,250 bones. There are 5719 people in total having an average trajectory length of 319 frames. All locations cover a ROI of 27,635 square units, with an average ROI of 512 square units per location, where one 3D unit was found to be approximately 0.85 m [14].

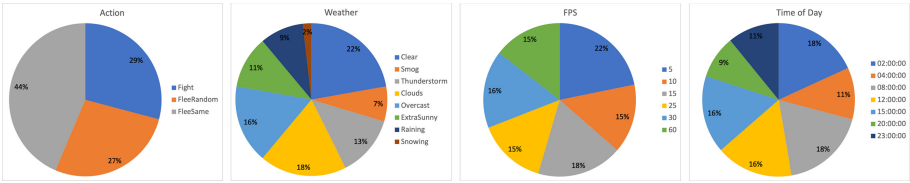

Fig. 1. The distribution of group behaviours, weathers, frames per second used to render the videos and of the times during which the videos were recorded.

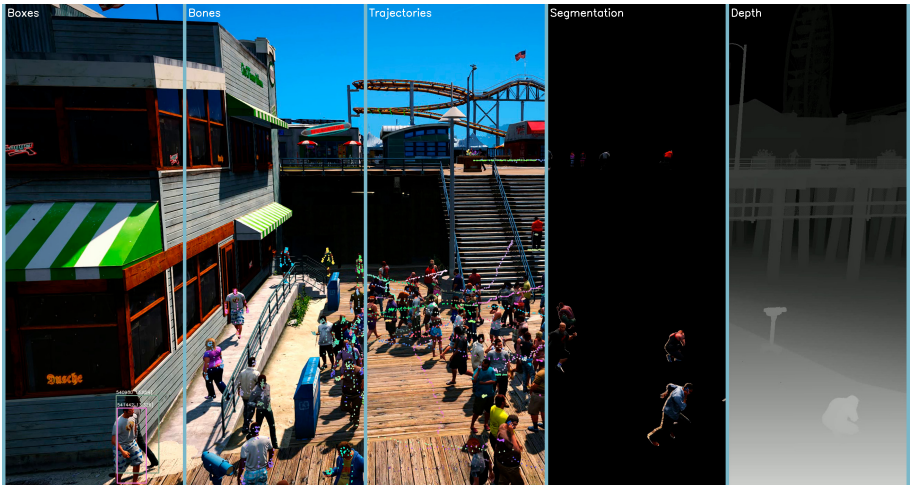

Fig. 2. Our frame annotations: from left to right, the bounding boxes of people along with their ID and distance to the camera, the bone coordinates of the people in the scene, the trajectories people have walked, the pixel-wise segmentation of people, and lastly pixel-wise depth information relative to the camera plane.

Now that the scene is set up, the timescale is set to 1.0, and the game starts to render the scene. Meanwhile, the script saves images and annotations, which include RGB, depth, segmentation information, person pose information (bone

locations), bounding boxes, and person IDs. Thus, the post-processing script can find trajectories, tighten bounding boxes, segmentations, and depth estimates. Figure 2 shows a sample synthetically generated frame using our method, with annotations corresponding to person detection (bounding boxes), bones, trajectories, person segmentation and depth maps.

3.2 Dataset Description

The resulting dataset contains detailed scene information and annotations, described below:

1. a bitmap stencil image
   - 0: Environment objects like floor, stairs, buildings
   - 1: Persons
   - 2: Cars and trucks
   - 3: Waving flags, plants, trees
   - 4: Beach sand, grass
   - 7: sky
2. a depth map, as a single channel image with float values in range [0, 1], where 1 is close to the camera and 0 is far (http://www.adriancourreges.com/blog/2015/11/02/gta-v-graphics-study/)
3. location information in a json file, containing:
   - location name
   - camera position
   - camera rotation
   - player position
   - player rotation
   - ROI of the location
   - PedGroups: contains the initial positions of all the people that belong to this group
   - PedCenters: contains the original centers around which the people were spawned
   - PedIdGroup: contains the handles of all the spawned peds, and the cluster center they belong to
   - fps: frames per second the video was rendered at
   - Action: the action that happens at a specific frame
   - Current time: the time the video was recorded in game time
   - Currentweather: the weather of that scene [0 = ExtraSunny, 1 = Clear, 2 = Clouds, 3 = Smog, 5 = Overcast, 6 = Raining, 7 = Thunderstorm, 13=Snow]
   - CamFOV: the field of view of the recording camera (always 50)
4. Per frame annotations that contain:
   - handle which is unique ID for every person
   - their distance from the camera in meters
   - normalized onscreen [0,1] bone coordinates

4 Unsupervised Abnormal Event Detection Using GTA V

The datasets created using our method can be very useful in the problem of abnormal crowd event detection, where there is a lack of densely annotated high resolution benchmark data, as explained in Sect. 2.3. We consider realistic crowd event scenarios in various indoors/outdoors environments and weather conditions, and specifically events such as crowd dispersion or scattering, crowd fleeing and a fight breaking out in a crowd, with some examples shown in Fig. 3.

Fig. 3. Examples from our photorealistic crowd event datasets. Left to right: School yard before students merge towards the exit and leave, villa garden before a crowd leaves, beach before people run away, mall after a crowd disperses.

4.1 Cumulative Sum Method for Abnormal Event Detection

The event taking place, as well as the time it takes place, is unknown beforehand, and is characterized mostly by a change in the crowd motion. For this reason, we choose to detect possible events by analyzing the statistics of the crowd optical flow over time. The basic assumption is that current crowd behavior is "normal", and a significant deviation from it would be abnormal, which is often the case in several realistic scenarios like the ones considered in our work.

We use sequential statistical change detection, namely the Cumulative Sum (CUSUM) approach [8,18], to detect a change between normal and abnormal crowd motion, as it is can effectively and quickly detect changes between statistical distributions. The first w_0 frames ($w_0 = 15$ here) are considered to characterize baseline (normal) crowd behavior/motion and the most recent $w_0$characterize "current" crowd behavior/motion. Their motion is found by estimating dense optical flow using SoA FlowNet2 [24] and its empirical distribution through the histogram of the optical flow. It should be noted that FlowNet2 also uses synthetically generated data to improve its accuracy, and has surpassed previous SoA optical flow estimation methods because of this.

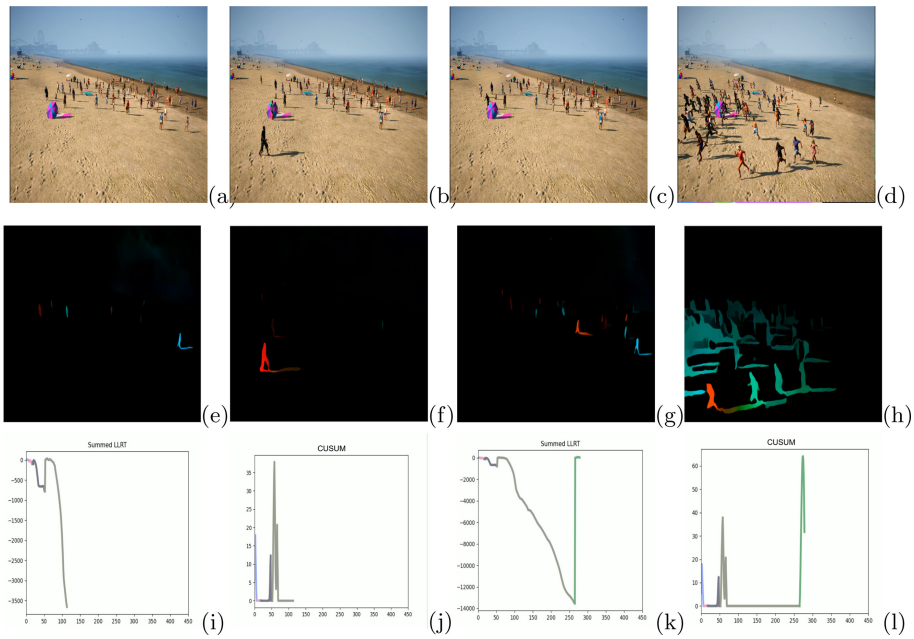

Fig. 4. Beach crowd events. Frames: (a) before 1st event (b) after 1st event (person entering, walking in a specific direction) (c) before 2nd event, (d) after 2nd event (crowd running, dispersing). Optical flow: (e) before 1st event, (f) after 1st event (g) before 2nd event, (h) after 2nd event. Change in the values of: (i) Summed LLRT for 1st event, (j) CUSUM for 1st event, (k) Summed LLRT for 2nd event, (l) CUSUM for 2nd event.

The histogram of the optical flow approximates its statistical distribution, and is denoted at frame k with $f_k(\bar{r})$ at each pixel $\bar{r} = (x, y)$, while $f_0(\bar{r})$ represents the distribution of the baseline motion. These two distributions are used to estimate the log-likelihood ratio L_k, a commonly used test statistic for detecting changes between statistical distributions, given by:

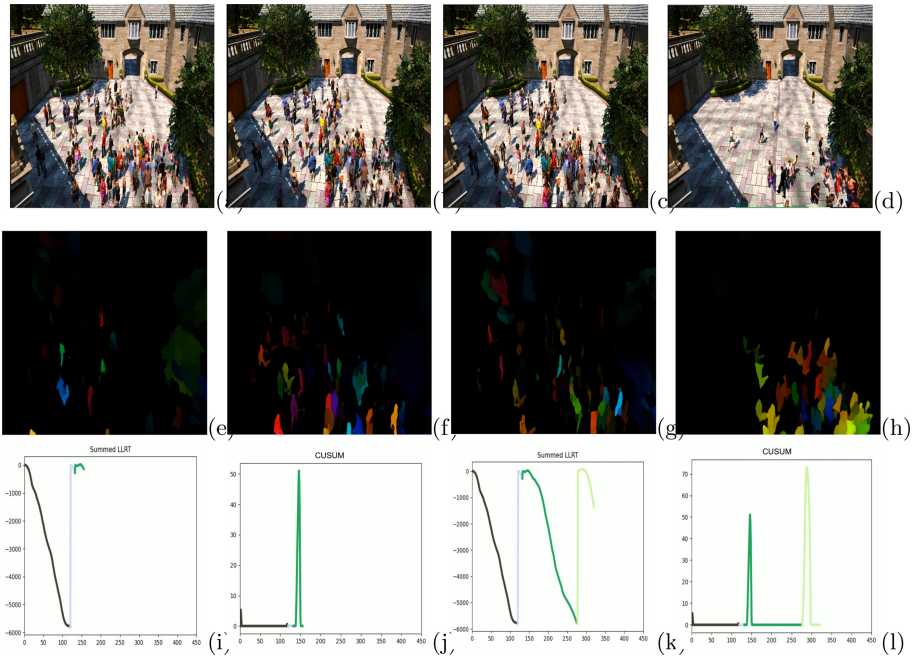

Fig. 5. Villa crowd events. Frames: (a) before 1st event (b) after 1st event (crowd moving more in a stochastic way) (c) before 2nd event, (d) after 2nd event (crowd running towards exit). Optical flow: (e) before 1st event, (f) after 1st event (g) before 2nd event, (h) after 2nd event. Change in the values of: (i) Summed LLRT for 1st event, (j) CUSUM for 1st event, (k) Summed LLRT for 2nd event, (l) CUSUM for 2nd event.

$$L_k = \ln\left(\frac{f_k(\bar{r})}{f_0(\bar{r})}\right) \tag{1}$$

The CUSUM test uses the log-likelihood ratio (LLRT) as test statistic L_k for the test at frame k, expressed in the computationally efficient iterative form [32]:

$$T_k = \max(0, T_{k-1} + L_k), \tag{2}$$

where we set $S_k = T_{k-1} + L_k$ as the Summed LLRT, and initialize $T_0 = 0$. When the data deviates from f_0, T_k increases significantly, and a change can be detected at that point. There is no theoretically founded method for setting the T_k threshold, so in this work, a value equal to 100 was empirically found to provide accurate results. In the case of multiple events, when an event is detected, T_k is re-set to 0 and the entire process restarts. This can be seen in the Summed LLRT and CUSUM plots in Figs. 4–5 where several events take place. These statistics, the videos and their optical flow, can be seen evolving in real time, for the videos examined here, as well as other videos generated by our tool, on our video demos on our GitHub.

4.2 Experimental Results: Abnormal Event Detection

Figures 4–5 show two different crowd event scenarios, on a sunny day at the beach, with the sea water moving, and in a villa yard surrounded by moving tree leaves. As detailed in the captions of Figs. 4–5, they display characteristic frames before events, the optical flow between them, the Summed LLRT and CUSUM values until those frames. In Fig. 4 we see the results of a crowd on a beach where a man suddenly enters, walking in one direction (first event), and the crowd later suddenly runs away (second event). The optical flow before and after the events is shown, and is clearly different, which is also reflected in the Summed LLRT and CUSUM for each event, whose values change sharply when the person enters the beach area, and when the people start to run. It should be noted that the motion caused by people slowly walking around, and small background motions from the sea, did not affect the detection of the actual events. Figure 5 contains a crowd of people standing/walking in the yard of a villa. Two main crowd events take place: in the first event, the crowd that is standing in the yard moves more quickly. In the second event, the crowd runs towards the exit. We display frames before/after both events, the optical flow images and the corresponding Summed LLRT, CUSUM in Fig. 5. It is clear that the changes in the crowd motion are reflected in the Summed LLRT and CUSUM.

It is interesting to note that the first event, of the crowd's motion becoming more stochastic/random, is not easy to detect visually, however our method detects it. We provide videos of the sequential detection of events in these videos, and additional examples with different scenarios and environments generated by our tool, in our GitHub. These results show CUSUM can robustly detect abnormal crowd behaviors, originating from changes in behavior that correspond to changes in motion, in a variety of environments and scenarios.

5 Conclusions

In this work, we have presented a method for generating high-resolution photorealistic synthetic datasets using the game engine from GTA V. The method used allows for the control of the scenes in great detail, and results in detailed annotations for a wide variety of scenes, events, activities, and crowds. Apart from data, we provide the tool itself for generating additional datasets on our GitHub[1], which can be used to solve a wide range of computer vision problems such as activity recognition, person detection and tracking, event detection and others. We demonstrate the usefulness of our dataset for abnormal crowd event detection, as there is a significant lack of annotated datasets for this problem. A sequential statistical change detection method for finding changes in the statistical distribution of datasets is applied to the optical flow of our synthetic scenes. The optical flow is estimated using SoA deep learning Flownet2, which provides dense accurate flow estimates. The results show accurate and quick detection of

[1] https://github.com/RicoMontulet/GTA5Event.

changes in different scenarios, indoors and outdoors, at different times of day and in different environmental conditions. Future work will expand upon the detection of abnormal events on more extensive synthetic and real datasets, but also on the use of our data for problems like person tracking, re-identification, activity and interaction recognition.

References

1. How Google's DeepMind will train its AI inside Unity's video game worlds (2018). https://web.archive.org/web/20180927024638/www.fastcompany.com/90240010/deepminds-ai-will-learn-inside-unitys-video-game-worlds
2. Policy on posting copyrighted rockstar games material (Oct 2020). https://tinyurl.com/RockstarPrivacyPolicy. Accessed 15 Sept 2020
3. Unity Machine Learning Agents (2020). https://unity.com/products/machine-learning-agents
4. Adam, A., Rivlin, E., Shimshoni, I., Reinitz, D.: Robust real-time unusual event detection using multiple fixed-location monitors. IEEE Trans. Pattern Anal. Mach. Intell. **30**(3), 555–560 (2008)
5. Andrade, E., Fisher, B.: Simulation of crowd problems for computer vision. In: 1st International Workshop on Crowd Simulation (V-CROWDS '05), pp. 71–80 (2005)
6. Bąk, S., Carr, P., Lalonde, J.-F.: Domain adaptation through synthesis for unsupervised person re-identification. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018, Part XIII. LNCS, vol. 11217, pp. 193–209. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01261-8_12
7. Barron, J.L., Fleet, D.J., Beauchemin, S.S.: Performance of optical flow techniques. Int. J. Comput. Vis. **12**, 43–77 (1994)
8. Basseville, M., Nikiforov, I.: Detection of Abrupt Changes: Theory and Application. Prentice-Hall Inc., Englewood Cliffs (1993)
9. Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., Sheikh, Y.A.: OpenPose: real-time multi-person 2D pose estimation using part affinity fields. IEEE Trans. Pattern Anal. Mach. Intell. **43**(1), 172–186 (2019)
10. Cartucho, J., Tukra, S., Li, Y., Elson, D.S., Giannarou, S.: VisionBlender: a tool to efficiently generate computer vision datasets for robotic surgery. In: Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization (2020)
11. Community, B.O.: Blender - a 3D modelling and rendering package. Blender Foundation, Stichting Blender Foundation, Amsterdam (2018). http://www.blender.org
12. De Souza, C.R., Gaidon, A., Cabon, Y., Lpez, A.M.: Procedural generation of videos to train deep action recognition networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2594–2604 (2017)
13. Denninger, M., et al.: BlenderProc: reducing the reality gap with photorealistic rendering. In: Robotics: Science and Systems (RSS) Workshops (2020)
14. Doan, A.D., Jawaid, A.M., Do, T.T., Chin, T.J.: G2D: from GTA to Data (2018)
15. Dosovitskiy, A., et al.: FlowNet: learning optical flow with convolutional networks. In: IEEE International Conference on Computer Vision (ICCV), pp. 2758–2766 (2015)
16. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: an open urban driving simulator. In: 1st Annual Conference on Robot Learning, pp. 1–16 (2017)

17. Dworak, D., Ciepiela, F., Derbisz, J., Izzat, I., Komorkiewicz, M., Wjcik, M.: Performance of LiDAR object detection deep learning architectures based on artificially generated point cloud data from CARLA simulator. In: 2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR), pp. 600–605 (2019)
18. Einmahl, J., McKeague, I.: Empirical likelihood based hypothesis testing. Bernoulli **9**, 267–290 (2003)
19. Elanattil, S., Moghadam, P.: Synthetic human model dataset for skeleton driven non-rigid motion tracking and 3D reconstruction (2019)
20. Gyuri, I.: Europilot: A toolkit for controlling Euro Truck Simulator 2 with Python to develop self-driving algorithms (2017). https://github.com/marshq/europilot
21. Heeger, D.J.: Model for the extraction of image flow. J. Opt. Soc. Am. A **4**(8), 1455–1471 (1987)
22. Hoffman, J., et al.: CyCADA: cycle-consistent adversarial domain adaptation. In: International Conference on Machine Learning, pp. 1989–1998 (Jul 2018)
23. Huang, X., Liu, M.-Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018, Part III. LNCS, vol. 11207, pp. 179–196. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01219-9_11
24. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: FlowNet 2.0: evolution of optical flow estimation with deep networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1647–1655 (2017)
25. Johnson-Roberson, M., Barto, C., Mehta, R., Sridhar, S.N., Rosaen, K., Vasudevan, R.: Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? arXiv preprint arXiv:1610.01983 (2016)
26. Juliani, A., et al.: Unity: a general platform for intelligent agents. arXiv preprint arXiv:1809.02627 (2020). https://github.com/Unity-Technologies/ml-agents
27. Li, W., Mahadevan, V., Vasconcelos, N.: Anomaly detection and localization in crowded scenes. IEEE Trans. Pattern Anal. Mach. Intell. **36**(1), 18–32 (2014)
28. Liu, M., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, Red Hook, NY, USA, pp. 700–708 (2017)
29. Lu, C., Shi, J., Jia, J.: Abnormal event detection at 150 fps in matlab. In: Proceedings of the 2013 IEEE International Conference on Computer Vision, ICCV '13, IEEE Computer Society, USA, pp. 2720–2727 (2013)
30. Mayer, N., et al.: What makes good synthetic training data for learning disparity and optical flow estimation? Int. J. Comput. Vis. **126**, 942–960 (2018)
31. Oghaz, M.M., Argyriou, V., Remagnino, P.: Learning how to analyse crowd behaviour using synthetic data. In: Proceedings of the 32nd International Conference on Computer Animation and Social Agents, pp. 11–14 (2019)
32. Page, E.S.: Continuous inspection scheme. Biometrika **41**, 100–115 (1954)
33. Pollok, T., Junglas, L., Ruf, B., Schumann, A.: UnrealGT: using unreal engine to generate ground truth datasets. In: Bebis, G., et al. (eds.) ISVC 2019, Part I. LNCS, vol. 11844, pp. 670–682. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33720-9_52
34. Qiu, W., et al.: UnrealCV: virtual worlds for computer vision. In: ACM Multimedia Open Source Software Competition (2017)
35. Ramachandra, B., Jones, M.J.: Street scene: a new dataset and evaluation protocol for video anomaly detection. In: IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, 1–5 March 2020 (2020)

36. Ranjan, A., Hoffmann, D.T., Tzionas, D., Tang, S., Romero, J., Black, M.J.: Learning multi-human optical flow. Int. J. Comput. Vis. (IJCV) **128**, 873–890 (2020). http://humanflow.is.tue.mpg.de
37. Richter, S.R., Hayder, Z., Koltun, V.: Playing for benchmarks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2213–2222 (2017)
38. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: ground truth from computer games. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016, Part II. LNCS, vol. 9906, pp. 102–118. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_7
39. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The SYNTHIA dataset: a large collection of synthetic images for semantic segmentation of urban scenes. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3234–3243 (2016)
40. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The SYNTHIA dataset: a large collection of synthetic images for semantic segmentation of urban scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3234–3243 (2016)
41. Saleh, F.S., Aliakbarian, M.S., Salzmann, M., Petersson, L., Alvarez, J.M.: Effective use of synthetic data for urban scene semantic segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018, Part II. LNCS, vol. 11206, pp. 86–103. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01216-8_6
42. Shah, S., Dey, D., Lovett, C., Kapoor, A.: Airsim: high-fidelity visual and physical simulation for autonomous vehicles. In: Field and Service Robotics (2017). https://arxiv.org/abs/1705.05065
43. Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.: Semantic scene completion from a single depth image. In: Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition (2017)
44. Tremblay, J., et al.: Training deep networks with synthetic data: bridging the reality gap by domain randomization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2018)
45. Tremblay, J., To, T., Birchfield, S.: Falling things: a synthetic dataset for 3D object detection and pose estimation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2119–21193 (2018)
46. UMN: University of Minnesota dataset. http://mha.cs.umn.edu/proj_events.shtml
47. Varol, G., et al.: Learning from synthetic humans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
48. Wang, Q., Gao, J., Lin, W., Yuan, Y.: Learning from synthetic data for crowd counting in the wild. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8198–8207 (2019)
49. Xiang, S., Fu, Y., You, G., Liu, T.: Attribute analysis with synthetic dataset for person re-identification. arXiv preprint:2006.07139 (2020)