LCrowdV: Generating Labeled Videos for Simulation-based Crowd Behavior Learning

Ernest Cheung[1], Tsan Kwong Wong[1], Aniket Bera[1], Xiaogang Wang[2], and Dinesh Manocha[1]

The University of North Carolina at Chapel Hill[1],
The Chinese University of Hong Kong[2]
http://gamma.cs.unc.edu/LCrowdV

Abstract. We present a novel procedural framework to generate an arbitrary number of labeled crowd videos (LCrowdV). The resulting crowd video datasets are used to design accurate algorithms or training models for crowded scene understanding. Our overall approach is composed of two components: a procedural simulation framework for generating crowd movements and behaviors, and a procedural rendering framework to generate different videos or images. Each video or image is automatically labeled based on the environment, number of pedestrians, density, behavior (agent personality), flow, lighting conditions, viewpoint, noise, etc. Furthermore, we can increase the realism by combining synthetically-generated behaviors with real-world background videos. We demonstrate the benefits of LCrowdV over prior lableled crowd datasets, by augmenting real dataset with it and improving the accuracy in pedestrian detection. LCrowdV has been made available as an online resource.

Keywords: crowd analysis, pedestrian detection, crowd behaviors, crowd datasets, crowd simulation, crowd rendering

1 Introduction

The accessibility of commodity cameras has lead to wide availability of crowd videos. In particular, videos of crowds consisting of tens or hundreds (or more) of human agents or pedestrians are increasingly becoming available on the internet, e.g. YouTube. One of the main challenges in computer vision and related areas is crowded scene understanding or crowd video analysis. There are a range of sub-problems in crowd understanding and analysis, including crowd behavior analysis, crowd tracking, crowd segmentation, crowd counting, abnormal behavior detection, crowd prediction, etc.

The problems related to crowded scene understanding have been extensively studied. Many solutions for each of these sub-problems have been developed by using crowd video datasets [1–5] along with different techniques for computing robust features or learning the models. However, most of these datasets are

[1] {ernestc, ahtsans, ab, dm}@cs.unc.edu
[2] xgwang@ee.cuhk.edu.hk

limited, either in terms of different crowd behavior or scenarios, or the accuracy of the labels.

Machine learning methods, including deep learning, usually require a large set of labeled data to avoid overfitting and to compute accurate results. A large fraction of crowd videos available on the Internet are not labeled or do not have ground truth or accurate information about the features. There are many challenges that arise in terms of using these Internet crowd videos for scene understanding:

- The process of labeling the videos is manual and can be very time consuming.
- There may not be a sufficient number of videos available for certain crowd behaviors (e.g., panic evaluation from a large building or stadium) or for certain cultures (e.g., crowd gatherings in remote villages in the developing world). Most Internet-based videos are limited to popular locations or events.
- The classification process is subject to the socio-cultural background of the human observers and their intrinsic biases. This can result in inconsistent labels for similar behaviors.
- In videos corresponding to medium and high density crowds, it is rather difficult to count the number of pedestrians exactly or classify their behaviors or tracks. This complexity is highlighted in one of the sample images in the UCF Crowd counting dataset [6], shown in Figure 3. Similar problems can arise in noisy videos or the ones recorded in poor lighting conditions.

In this paper, we present a new approach to procedurally generate a very large number of labeled, synthetic crowd videos for crowded scene understanding. Our approach is motivated by prior use of synthetic datasets in computer vision for different applications, including pedestrian detection [3, 7], recognizing articulated objects from a single image [8], multi-view car detection [9, 10], 3D indoor scene understanding [11], etc. In some cases, models trained using synthetic datasets can outperform models trained on real scene-specific data, when labeled real-world data is limited.

Main Results: We present a novel procedural framework to generate labeled crowd videos (LCrowdV). Our approach consists of two main components: procedural simulation and procedural rendering. Given a set of parameters or labels, our procedural framework can automatically generate an arbitrary number of crowd videos. These labels correspond to different indoor or outdoor environments, number of pedestrians, pedestrian density, crowd behaviors, lighting conditions, noise, camera location, abnormal behaviors, etc.

Our approach can be used to generate an arbitrary number of crowd videos or images (e.g. millions) by varying these classification parameters. Furthermore, we automatically generate a large number of labels for each image or video. The quality of each video, in terms of noise and resolution, can also be controlled using our procedural renderer. The generation of each video frame only takes a few milliseconds on a single CPU core. And the entire generation process can be easily parallelized on a large number of machines or servers, by running different instance of the framework on different machine.

We demonstrate the benefits of LCrowdV over prior labeled crowd video datasets on the following applications:

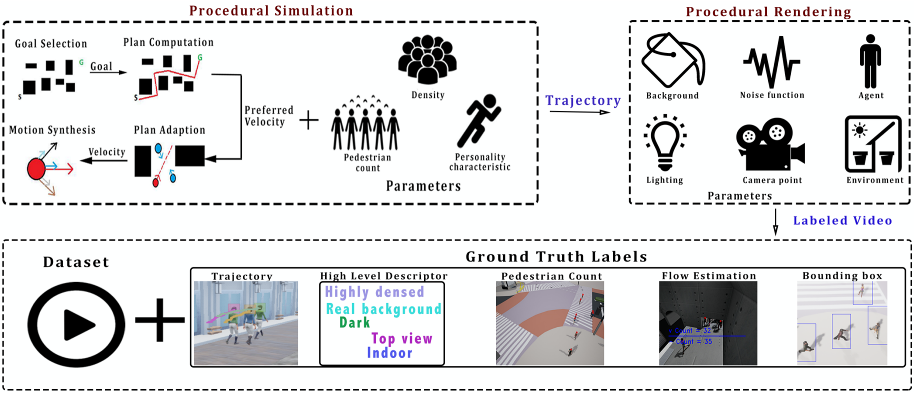


Fig. 1: LCrowdV framework consists of two components: procedural simulation (top left) and procedural renderer (top right). There are several parameters as show in the figure can be adjusted during the Procedural Simulation and Procedural Rendering to produce videos with a range of variety. Each final video/image consists of a number of ground truth labels (bottom). Our approach can automatically generate different videos with accurate labels.

- **Improved Pedestrian Detection using HOG+SVM:** We demonstrate that combining LCrowdV with a few real world annotated videos can improve the average precision by 3%. In particular, the variations in the camera angle in the LCrowdV dataset have the maximal impact on the accuracy. Instead of 70K labeled real-world videos, we only use 1K annotated real-world videos along with LCrowdV to get improved accuracy.
- **Improved Pedestrian Detection using Faster R-CNN:** We demonstrate that combining LCrowdV with a few real world annotated videos can improve the average precision by 7.3%. Furthermore, we only use 50 labeled samples from the real-world dataset and combine them with LCrowdV.

The rest of the paper is organized as follows. We give a brief overview of prior work on crowd analysis and crowd datasets in Section 2. We describe our procedural framework in Section 3 and highlight its benefits for pedestrian detection in Section 4.

2 Related Work

The simulation, observation and analysis of crowd behaviors have been extensively studied in computer graphics and animation, social sciences, robotics, computer vision, and pedestrian dynamics [12]. In this section, we give a brief overview of recent work on crowd video analysis, classification, labeling, and prior crowd datasets.

2.1 Crowd Video Analysis

Different models have been proposed to model the crowd behaviors [13–19]. Other methods focus on extracting specific features in crowds, including head counting [2, 20–24]. Online trackers tend to use the current, previous or recent frames for tracking each pedestrian in the video at interactive rates [6, 25–27]. Tracking the pedestrians to obtain the full trajectories has been extensively studied [1, 13, 28, 29]. There is considerable research on analyzing various crowd behaviors and movements from videos [30]. Most of these methods are designed for offline applications and tend to use a large number of training videos to learn the patterns [1, 14, 15, 31, 32]. Different methods have also been proposed to estimate the crowd flow in videos [4, 33–35] , model activities and interactions in crowded and complicated scenes [16, 17, 36]. There are also work that compare the performance of simulated data and real videos [37].

Many crowd analysis methods [13–17, 33, 36, 38] tend to be scene specific, which implies that they are trained from a specific scene and haven't been tested in terms of generalizing the results across other scenes. One of the challenges is to find complete crowd datasets that include data samples covering enough scenes and behaviors, and provide labeled ground truth data for learning. Some methods [20, 31, 34] don't require real data for training, but they are limited by the size of the crowds or specific conditions, including crowd behaviors and color information.

2.2 Crowd Labeling

Crowd behaviors are diverse, and it is a major challenge to model different behaviors. The work described in [31] classified crowd behaviors in five categories in accordance with their dynamical behavior: bottlenecks, fountainheads, lane formation, ring/arch formation and blocking. Besdies, the work described in [39] classified crowd behaviors into another five categories based on its dynamics: running, loitering, dispersal(center to edge), dispersal(edge to center) and formation. Interestingly, these two methods cannot cover all behaviors that are observed in crowd videos. Other methods focus on labeling the crowd data that can be described by a predefined set of labels [40–42]. Recently, the work described in [5] uses a model along with manually entered labels to classify crowds based on the location, the subject and the action. Our approach is motivated by these prior works on crowd labeling.

2.3 Crowd Video Datasets

Many crowd video datasets that are available can provide ground truth or estimated labels for analysis or training. the work [1] provides trajectories of the pedestrians,the work [2] describes a database with the number of pedestrians for crowd counting, [3] includes bounding boxes of the detected pedestrians, and the work [4] provides ground truth labels for crowd flow estimation. The work of Shao et al. [5] consists of high-level labels to describe crowd characteristics. Most of these datasets are generated by labeling the data manually, and this process is

Dataset	CUHK [43]	Collectiveness [44]	Violence [45]	Data-Driven [1]	UCF [6]	WWW [5]	CVC07 [46]	**LCrowdV**
Videos	474	413	246	212	46	10000	N/A	**>1 M**
Frames	60,384	40,796	22,074	121,626	18,196	>8 M	2,534	**>20 M**
Resolution	Varying	670x1000	320x240	720x480	Varying	640x360	Varying	**Any**
Trajectory	x	x	x	✓	x	x	x	✓
Pedestrian count	x	x	x	x	x	x	x	✓
Flow estimation	x	x	x	x	✓	x	x	✓
Attributes	3	1	0	2	0	3	0	7
Bounding box	x	x	x	x	x	x	✓	✓
Generation Method	Manually						**Automatically**	

Table 1: **Benefits of LCrowdV:** Not only can we generate a significantly higher (or arbitrary) number of labeled videos, but we can also provide a large set of crowd labels and characteristics for each image and video. We can also control the behavior characteristics, environments, resolution and rendering quality of each video to develop a good training set. Unlike prior methods, we can easily generate accurate labels.

time-consuming and error-prone. For example, to allow the deep learning model in [5] to understand the scenes and put appropriate labels, 19 human annotators were involved. Several human operators were needed for determining the number of objects in [2]. On the other hand, LCrowdV compute these labels automatically from the simulation, and therefore is not error-prone and much efficient. Table 1 shows the comparison of existing crowd datasets, as compared to LCrowdV.

2.4 Learning Crowd Behaviors with Simulated Data

The work described in [7] trained a pedestrian detector for a driver assistant system using simulated data. This work focused on the training and testing data from one particular camera angle, which corresponds to the driver's view. The work [3] generated simulated agents on a specific real-world background to enable learning for pedestrian detectors. In contrast with these methods, our work aims at providing a diversified, generic and comprehensive approach for generating different crowd video data for analyzing different crowd behaviors.

3 Synthetic Label Crowd Video Generation

Crowds are observed in different situations in the real-world, including indoor and outdoor scenarios. One of our goals is to develop a procedural framework that is capable of providing all types of crowd videos with appropriate ground truth labels. In this section, we give an overview of our framework and the various parameters used to generate the videos.

3.1 Crowd Generator

Modeling the behavior of large, heterogeneous crowds is important in various domains including psychology, robotics, pedestrian dynamics, and computer graphics. There is more than a century of work in social sciences and psychology on observing and classifying crowds, starting from the pioneering work of Lebon [47].

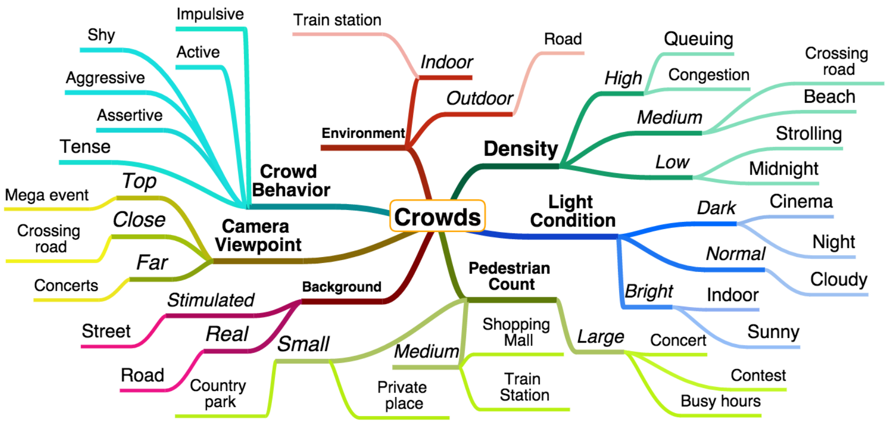


Fig. 2: Hierarchical and parametric classification of crowd behaviors and renderings. Attribute Labels of LCrowdV includes: Background, Crowd Behaviour (Personality), Camera Viewpoint, Density, Environment, Light Condition and Pedestrian Count. We use these labels to classify different characteristics of crowds and use them in our procedural framework. Note that the list above is not exhaustive, as we can further extends attribute like Density to medium-high, medium-low, etc.

Other social scientists have classified the crowds in terms of behaviors, size, and distributions [48]. According to *Convergence Theory*, crowd behavior is not a product of the crowd itself; rather it is carried into the crowd by the individuals [49]. These observations have been used to simulate different crowd behaviors and flows [50–52].

Our procedural crowd simulation framework builds on these prior observations in social sciences [53] and on simulation methods [54]. The overall hierarchical classification is shown in Fig. 2. Each of these labels is used to describe some attributes of the crowds or the pedestrians. In addition to the labels that govern the movements or trajectories of each agent or pedestrian, we also use a few labels that control the final rendering (e.g., lighting, camera position, brightness, field of view) of the crowd video by using appropriate techniques from computer graphics. Finally, we can also add some noise (e.g. Gaussian noise) to degrade the final rendered images, so that these images are similar to those captured using video cameras which actually have such type of noise.

Framework Design: Our framework has two major components: procedural simulation and procedural rendering. The procedural simulator takes as input the number of agents or pedestrians, densities, behavior (personality of agent groups) and flow characteristics and computes the appropriate trajectories and movements of each agent corresponding to different frames. Given the position of each agent during each frame, the procedural renderer generates the image frames based on the different parameters that control the lighting conditions. Each of these input parameters corresponds to a label of the final video.

(a) (b) (c)

Fig. 3: (a) Generated using LCrowdV and consists of 858 agents. (c) From UCF dataset [6], it is very hard to accurately count the number of pedestrians or classify other characteristics such as density or behavior in this real-world image (i.e. generate accurate labels). In contrast, our method can automatically generate accurate labels as demonstrated in (b).

Procedural Crowd Simulation: In this section, we give an overview of our procedural crowd simulator. We use the Menge crowd simulation engine [55], which makes it possible to combine state-of-the-art crowd modeling techniques addressed in the previous section. In particular, we have picked ORCA [54] as our navigation algorithm. Given the labels or high-level descriptors, our method can generate crowd movements or behaviors that fit those descriptions. These include the total number of agents or pedestrians in the scene, their density over different parts of the environment or the scene, their global and local movements, and the behavior characteristics.

Global Crowd Characteristics: In the simulation stage, we vary the global parameters, including the personality settings of different agents, density, and the number of agents used to generate different types of trajectories. The number of agents will control how many pedestrians are in the scene, and the density factor decides whether or not the pedestrians would be located very close to each other. It is essential to include different levels of overlapping in the training data set to avoid overfitting. The personality parameters allow the crowd behavior to be more natural-looking.

Crowd Movement and Trajectory Generation: A key component is simulating the crowd or pedestrian movement. We build on prior research in crowd movement simulation [56, 57] and use the property that movement specification can be decomposed into three major subproblems: agent goal selection, global plan computation, and local plan adaptation (see Fig. 1). We further elaborate on each of these components and give various possible solutions for each of these subproblems, based on the different crowd behavior classifier.

Goal Selection: In the goal selection module, we specify the high-level goal of each pedestrian. For example, the agent may want to go to a particular location in the scene or just visit a few areas, etc. It is expected that the goal can change across time and is affected by the intrinsic personalities or characteristics of the agents or the environmental factors. There is extensive literature on goal selection methods and we can use these methods in our procedural simulation framework [57, 58].

Global Path Planning: Given the goal specification of each agent, we compute a collision-free trajectory to achieve that goal. The path-planning module is a high-level module that computes a preferred velocity or direction for each agent for a given time-step. We use techniques based on potential field methods, roadmaps, navigation meshes, and corridor methods [59–62].

Local Plan Adaptation: Since the path computed in the previous stage usually considers only the static obstacles, we need to adapt the plan to tackle the dynamic obstacles or other agents or pedestrians in the scene. We transform the preferred velocity computed into a feasible velocity that can avoid collisions in real time with multiple agents. Some of the commonly used motion models for local plan adaptation are based on social forces, rule-based methods, and reciprocal velocity obstacles [63–65].

Full Human Motion Computation: The main goal of high degree of freedom human motion computation or motion synthesis is to compute the locomotion or position of each agent in terms of the joint positions, corresponding to the walk cycle as well as to the motion of the head and upper body. We use standard techniques from computer animation based on kinematic, dynamics and control-based methods to generate the exact position of each pedestrian in the scene [66–68].

3.2 Procedural Rendering:

After computing the trajectory or movement specification characterized by the global parameters for each pedestrian in the video, we generate an animation and render it using different parameters. We can control the lighting conditions, resolution, viewpoint, and the noise function to lower the image quality, as needed.

Animated Agent Models: We use a set of animated agent models, that include the gender, skin color, clothes and outlook. We randomly assign these models to every agent. Furthermore, we may associate some objects in the scene with each agent or pedestrian. For example, in the case of a shopping mall, a customer may carry items he or she bought in bags; and in a theme park, there may be parents walking along with the children. These attached items could potentially obstruct the agent and change its appearance.

3D Environments and Backgrounds: Our background scenes include both indoor and outdoor environments. Ideally, we can import any scene with a polygonal model representation and/or textures and use that to represent the environment. We can also vary the lighting conditions to model different weather conditions: a sunny day could have a huge difference in appearance compared to a gloomy day. On top of that, we can also add static and dynamic obstacles to model real world situations. For instance, we could add moving vehicles into a city map and animated trees into a countryside map.

Image-space Projection and and Noise Functions: In order to render the 3D virtual world and the animated agent model, we render the image using a camera projection model: perspective projection or orthogonal projection. Typically, we render the videos with perspective projections to simulate real world videos. At this stage, we use different parameters to the projection model to obtain the videos captured from different viewpoints. In practice, video and images collected from different camera views could result in significant differences in the appearance. We also add a Gaussian noise filter with varying levels of standard deviation to emulate the digital noise in a video frame.

In our current implementation, we use the Unreal game engine [69] for rendering and generating each video frame. It is an open source engine and we can easily specify the geometric representation, lighting and rendering information, and generate images/videos at any resolution or add noise functions. We can easily adjust the different rendering parameters available in Unreal Engine to control the final crowd rendering.

3.3 Ground Truth Labels

The two main labels related to such datasets are the pedestrian count and the trajectory of each pedestrian. And a single video can provide both kind of labels or even more when the framework is extended in the future. This can be rather challenging for dense crowds, where generating such a labeled dataset is a major challenge. In order to accurately generate such labels, we consider each head of an agent in the video that is not obstructed by other scene objects. We compute the screen-space coordinates during each frame for every agent using the given camera parameters. We can also compute the position of lower body or full body contours. Given these head and lower body information, we can accurately compute the count and the trajectories.

Apart from the trajectories of the head, we also use the bounding boxes for pedestrian detection. Using the same technique mentioned above, we compute the bounding box for each pedestrian, which is centered at the centroid of the model used for each agent. This is more accurate than annotating the bounding boxes manually, especially for high density scenes.

Another major problem in crowd scene analysis is computing the crowd flows. The goal is to estimate the number of pedestrians crossing a particular region in the video. For real videos with dense pedestrian flows, it is difficult and labor intensive to compute such flow measures. This is due to the fact that there may be partial occlusion and a human operator needs to review each frame multiple times to obtain this information accurately. On the other hand, we can easily count how many agents are crossing a line or a location in the scene. However, in some of the pedestrian videos, agents could walk around or over the counting line because of collision avoidance. If we can count every agent that crosses the line, this count could increase when an agent is close to the counting line or when an agent repeatedly crosses the line. Therefore, we define an agent as crossing the line only if it has passed a particular tolerance zone or region in the scene.

In addition to these labels corresponding to the tracks, bounding boxes, flows, etc. we also keep track of all the parameters used by the procedural framework, i.e. the seven different high level parameters. As mentioned in the previous section, we have generated the videos using seven different parameters. These parameters can be used to describe the video in a high level manner, as shown in Fig. 2 and Fig. 4.

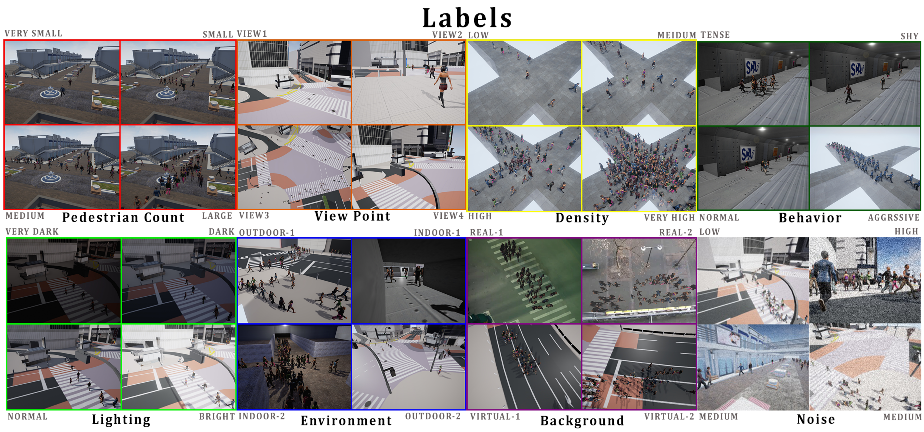


Fig. 4: **Parameters used in LCrowdV**: Samples of images that illustrate the effect of changing different high level parameters in the scene. These parameters are also used as the labels.

4 Applications and Evaluation

In this section, we highlight the benefits of LCrowdV dataset for improved pedestrian detection.

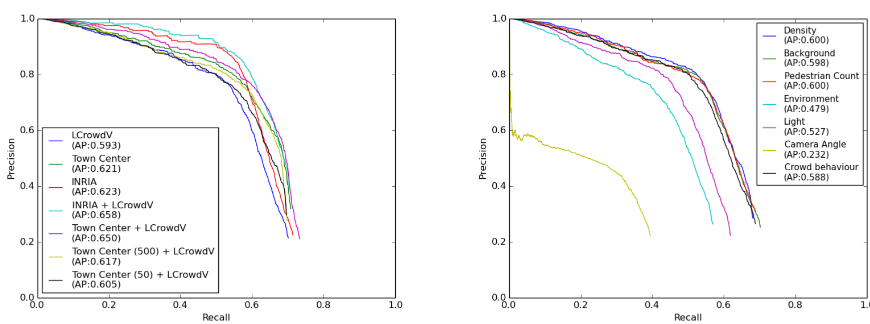
Crowd Datasets: For our evaluations, we used many real-world datasets: INRIA [70], KITTI [71], ETHZ [72], and Town Center [73]. The INRIA dataset contains 1832 training and 741 testing sample images. We used the object detection dataset in KITTI Vision Benchmark Suite. As annotations are not provided in the test set, we divided the train set into two subsets: 1279 images for training and 500 images for verification. In the ETHZ dataset, trained with BAHNHOF (999 image frames) and JELMOLI (936 image frames) and tested on SUNNY DAY (354 frames). The Town Center dataset is a 5-minute video with 7500 frames annotated, which is divided into 6500 for training and 1000 for testing data for pedestrian detection. We have created a new dataset called Person Search Database(PSDB). This database consists of $18,184$ images taken at different angles. Unlike the Town Center dataset, which consists of images at the same viewpoint and scene, the scenes in PSDB are more diverse, including shopping mall, roadside, University, park, etc. For behavior analysis, we evaluated the crowd motion trajectories of the pedestrians, as opposed to the actual appearance. For pedestrian detection, we use selected frames from the dataset.

4.1 Pedestrian detector evaluation

In this section, we highlight the benefit of using LCrowdV to train a learning algorithm and apply the results to pedestrian detection in real videos.

Pedestrian Detection using HOG+SVM: We compute the histogram of oriented gradients [70] on both positive and negative pedestrian samples in the training dataset as feature descriptor. We use a support vector machine to learn from these descriptors in order to determine whether or not a new image patch from the training dataset is a pedestrian or not. We refer to this method as HOG+SVM. In particular, our SVM detector is trained with OpenCV GPU HOG module and SVM light [70].

We trained numerous detectors by combining the real world datasets from INRIA [70], Town Center [73] and our synthetic data in LCrowdV. We used $10,000$ images from our dataset in this experiment. We observe that the detectors that are trained by combining LCrowdV and the INRIA or Town Center datasets have higher accuracy as compared to only using the real world datasets (i.e. INRIA only or Town Center only). In these cases, LCrowdV improves the average precision by 3%, though we observe higher accuracy for certain cases. We also evaluated two detectors which are trained using only limited samples: 50/500 positive + 50/500 negative, from the Town Center datasets and combined with LCrowdV. The results of the detectors trained by $500 + 500$ samples are shown to be comparable to the results of the detector trained by the entire original dataset. These benchmarks and results demonstrate that one does not have to spend extensive effort in annotating 70K image samples to train a detector, merely $1,000$ annotations are sufficient and can be combined with our synthetic LCrowdV dataset. The results are shown in Fig. 5(a). In this case, the use of LCrowdV labeled data can significant improve detectors' accuracy over prior datasets shown in Table 1.



(a) (b)

Fig. 5: Results of trained HOG+SVM detectors: (a) Trained using the realworld and augmented synthetic datasets (INRIA+LCrowdV, Town Center(Full)+LCrowdV, Town Center(500)+LCrowdV and Town Center(50)+LCrowdV)). The use of LCrowdV along with the real-world datasets can result in 3% average precision improvement, as compared to prior results based on only real-world datasets. (b) Different LCrowdV training video datasets obtained by changing each parameter individually. We observe that the variation in the camera angle parameter has maximal impact on improving the accuracy.

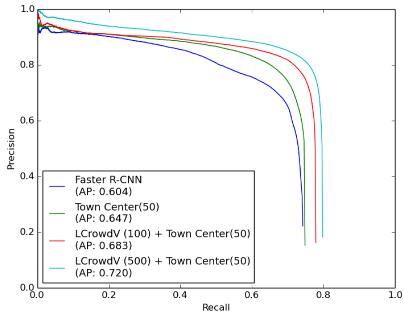
Varying LCrowdV Parameters: Our LCrowdV framework uses seven main parameters, as described in Fig. 2 and Fig. 4. We highlight the effect of using different parameters on the accuracy of our detector. We first train HOG+SVM using a set of synthetic dataset that is generated with variations all seven parameters. Next, we remove the variations in one parameter at a time and repeat the evaluation. The results are shown in Fig. 5(b).

Among the seven parameters, we observe that the variations in the camera angle parameter can affect the average precision by 36%, as compared to the other parameters used in LCrowdV. While it is difficult to capture videos from multiple camera angles in real-world scenarios, it is rather simple to vary these parameters in LCrowdV. These results highlight the benefits of LCrowdV.

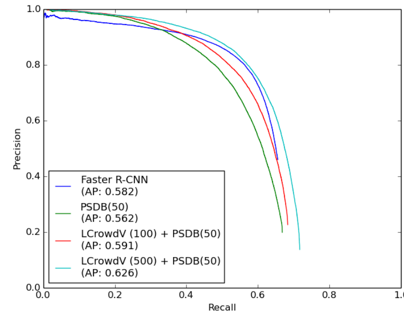
Pedestrian Detection using Faster R-CNN: Apart from HOG+SVM, we have also used LCrowdV to train the Faster Region-based Convolutional Network method (Faster R-CNN) [74], one of the state-of-the-art algorithms for object detection based on deep learning. R-CNN [75] is a convolutional neural network that makes use of region classification, and it has strong performance in terms of object detection. A variant, Fast R-CNN [76], combines several ideas to improve the training and testing speed while also increasing detection accuracy. We use a version of the Fast R-CNN algorithm that makes use of Region Proposal Network (RPN) to improve the performance, namely Faster R-CNN. The RPN makes use of a shared set of convolution layers with the Faster R-CNN network to save computation effort. In particular, we use the Simonyan and Zisserman model [77] (VGG-16) that is a very deep detection network and has 13 shareable convolutional layers. We adopt the Approximate joint training solution that makes it possible to merge RPN and Fast R-CNN network efficiently. In our implementation, we make use of the Caffe deep learning network [78], and we iteratively train the model until the performance converges at roughly 10k to 30k iterations.

We trained the model with an augmented dataset which combines both a small sample of Town Center dataset and LCrowdV, and then we use the model to detect pedestrians on the Town Center dataset. The results are shown in Fig. 6(a). With merely 50 samples annotated in the original Town Center dataset, adding LCrowdV into the training set results in an average precision of 72%, which is 7.3% better than the model trained with Town Center dataset only. In addition, we also verify our results by combining LCrowdV with PSDB, KITTI, ETHZ. For PSDB, the results are shown in Fig. 6(b) where the average precision improvement of 6.4% is observed in our combined training set, when comparing to training with samples from PSDB only. In both experiments, we can observe that as the sample size of LCrowdV increases, the performance of the model becomes better.

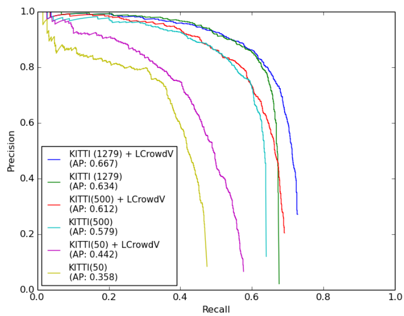
When we evaluate our results on KITTI, we vary the sample size of real annotations to find out also its impact on the performance. When the number of images with real annotations is {50, 125, 250, 500, 750, 1000, 1279}, the AP of KITTI and KITTI+LCrowdV is {35.8, 48.1, 54.9, 57.9, 61.6, 62.0, 63.4}% and {36.3, 48.9, 55.7, 58.6, 62.7, 64.9, 66.7}%, respectively. The summary of this result is also shown in Fig. 6(c). We can see the complementary effect of



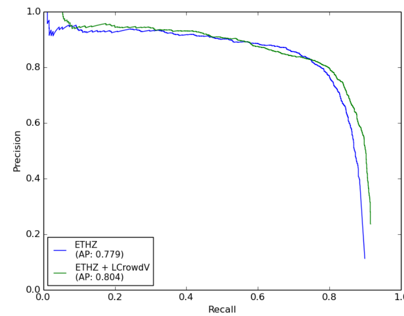
(a)



(b)



(c)



(d)

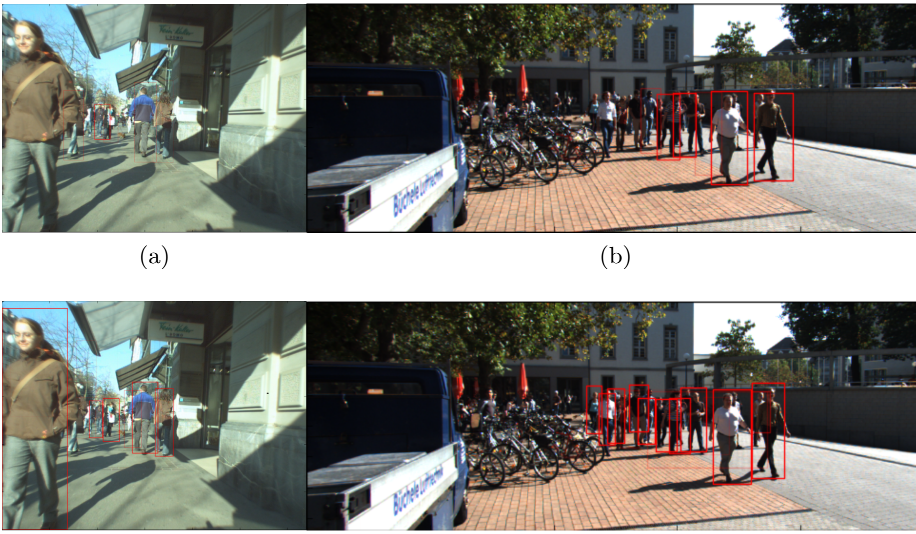
Fig. 6: Results of trained Faster R-CNN model with (a) Town Center dataset, (b) PSDB, (c) KITTI, and (d) ETHZ, and the augmented version of the aforementioned datasets using LCrowdV. The model trained with augmented dataset has an improvement in average precision up to 7.3%, 6.4%, 3.3% and 2.5% comparing to the model trained with the original dataset for (a), (b), (c), and (d) respectively.

LCrowdV on the training is consistently beneficial as the sample size of real annotation varies. We further evaluate the results on a cross-scene scenario using the ETHZ dataset, the improvement of the model trained with combined data is 2.5% as shown in Fig 6(d). Significant improvements are observed in the detection results for KITI and ETHZ are also shown in Fig. 7.

The results from both techniques for pedestrian detection mentioned above demonstrate that by combining a small set of samples from the same scene as the test data with LCrowdV, we can improve the detector/deep model results significantly.

5 Limitations, Conclusion and Future Work

We have presented a novel approach to generate labeled crowd videos (LCrowdV) using procedural modeling and rendering techniques. The main benefit of our



(a) (b)

(c) (d)

Fig. 7: Detection results of CNN trained using (a) ETHZ only, (b) KITTI only, (c) ETHZ+LCrowdV, (d) KITTI+LCrowdV. The alpha channel of the bounding box represents the confidence of the detection.

parametric approach of procedural crowd simulation is that: our formulation is general and can be used to include arbitrary numbers of pedestrians, density, behaviors, flows, rendering conditions, and vary the resolution of the images or video. Compared to prior crowd datasets, our synthetic methods can generate a significantly larger collection of crowd videos with accurate labels or ground truth data in a much easier way. We have demonstrated the benefits of LCrowdV in augmenting real world dataset for pedestrian detections.

Our approach has a few limitations. The current simulation methods may not be able to capture all the details or subtle aspects of human behaviors or movements in certain situations. Our current rendering framework uses the capabilities of Unreal game engine, which may not be able to accurately render many outdoor effects. In particular, we want to improve the trajectory by learning from real videos [19]; and rendering quality using global illumination, data-driven, and full-body animation methods [79]. In terms of future work, we would like to overcome these limitations. We would like to continue investigating how to improve machine learning algorithms in related to crowds, including crowd counting, tracking, abnormal behavior detection, crowd behaviour classification, crowd segmentation and etc. We would also like to include traffic in these videos and generate datasets corresponding to human-vehicle interactions. We also have made the LCrowdV dataset available as an online resource.

6 Acknowledgment

This work is supported in part by ARO grant W911NF-16-1-0085 and Boeing. We are grateful to Sean Curtis and Andrew Best for their help with Menge and UnReal rendering engine.

References

1. Rodriguez, M., Sivic, J., Laptev, I., Audibert, J.Y.: Data-driven crowd analysis in videos. In: 2011 International Conference on Computer Vision. (Nov 2011) 1235–1242
2. Rabaud, V., Belongie, S.: Counting crowded moving objects. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. Volume 1., IEEE (2006) 705–711
3. Hattori, H., Boddeti, V.N., Kitani, K., Kanade, T.: Learning scene-specific pedestrian detectors without real data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3819–3827
4. Ozturk, O., Yamasaki, T., Aizawa, K.: Detecting dominant motion flows in unstructured/structured crowd scenes. In: Pattern Recognition (ICPR), 2010 20th International Conference on. (Aug 2010) 3533–3536
5. Shao, J., Kang, K., Loy, C.C., Wang, X.: Deeply learned attributes for crowded scene understanding. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2015) 4657–4666
6. Ali, S., Shah, M.: A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In: Computer Vision and Pattern Recognition, 2007. CVPR 2007. IEEE Conference on. (June 2007) 1–6
7. Marn, J., Vzquez, D., Gernimo, D., Lpez, A.M.: Learning appearance in virtual scenarios for pedestrian detection. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. (June 2010) 137–144
8. Dhome, M., Yassine, A., Lavest, J.M.: Determination of the pose of an articulated object from a single perspective view. In: BMVC. (1993) 1–10
9. Movshovitz-Attias, Y., Boddeti, V.N., Wei, Z., Sheikh, Y.: 3d pose-by-detection of vehicles via discriminatively reduced ensembles of correlation filters. In: British Machine Vision Conference. (2014)
10. Pepik, B., Stark, M., Gehler, P., Schiele, B.: Teaching 3d geometry to deformable part models. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. (June 2012) 3362–3369
11. Satkin, S., Lin, J., Hebert, M.: Data-driven scene understanding from 3d models. (2012)
12. Ali, S., Nishino, K., Manocha, D., Shah, M.: Modeling, Simulation and Visual Analysis of Crowds: A Multidisciplinary Perspective. Springer (2013)
13. Zhou, B., Wang, X., Tang, X.: Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. (June 2012) 2871–2878
14. Hospedales, T., Gong, S., Xiang, T.: A markov clustering topic model for mining behaviour in video. In: Computer Vision, 2009 IEEE 12th International Conference on, IEEE (2009) 1165–1172
15. Mehran, R., Oyama, A., Shah, M.: Abnormal crowd behavior detection using social force model. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. (June 2009) 935–942

16. Chan, A., Vasconcelos, N.: Modeling, clustering, and segmenting video with mixtures of dynamic textures. Pattern Analysis and Machine Intelligence, IEEE Transactions on **30**(5) (May 2008) 909–926
17. Wang, X., Ma, X., Grimson, W.: Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. Pattern Analysis and Machine Intelligence, IEEE Transactions on **31**(3) (March 2009) 539–555
18. Bera, A., Kim, S., Manocha, D.: Realtime anomaly detection using trajectory-level crowd behavior learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2016) 50–57
19. Kim, S., Bera, A., Manocha, D.: Interactive crowd content generation and analysis using trajectory-level behavior learning. In: 2015 IEEE International Symposium on Multimedia (ISM), IEEE (2015) 21–26
20. Xu, H., Lv, P., Meng, L.: A people counting system based on head-shoulder detection and tracking in surveillance video. In: Computer Design and Applications (ICCDA), 2010 International Conference on. Volume 1. (June 2010) V1–394–V1–398
21. Antonini, G., Thiran, J.P.: Counting pedestrians in video sequences using trajectory clustering. Circuits and Systems for Video Technology, IEEE Transactions on **16**(8) (Aug 2006) 1008–1020
22. Garcia-Bunster, G., Torres-Torriti, M.: A density-based approach for effective pedestrian counting at bus stops. In: Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on, IEEE (2009) 3434–3439
23. Idrees, H., Saleemi, I., Seibert, C., Shah, M.: Multi-source multi-scale counting in extremely dense crowd images. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. (June 2013) 2547–2554
24. Zhang, C., Li, H., Wang, X., Yang, X.: Cross-scene crowd counting via deep convolutional neural networks. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2015) 833–841
25. Zhang, K., Zhang, L., Yang, M.H.: Real-time compressive tracking. In: ECCV 2012. (2012) 864–877
26. Fu, W., Wang, J., Li, Z., Lu, H., Ma, S.: Learning semantic motion patterns for dynamic scenes by improved sparse topical coding. In: Multimedia and Expo (ICME), IEEE International Conference on. (2012) 296–301
27. Song, X., Shao, X., Zhang, Q., Shibasaki, R., Zhao, H., Cui, J., Zha, H.: A fully online and unsupervised system for large and high-density area surveillance: Tracking, semantic scene learning and abnormality detection. TIST (2013)
28. Ali, S., Shah, M.: Floor fields for tracking in high density crowd scenes. In: ECCV 2008. Springer (2008) 1–14
29. Zhu, F., Wang, X., Yu, N.: Crowd tracking with dynamic evolution of group structures. In: ECCV 2014. Springer (2014) 139–154
30. Li, T., Chang, H., Wang, M., Ni, B., Hong, R., Yan, S.: Crowded scene analysis: A survey. Circuits and Systems for Video Technology, IEEE Transactions on **25**(3) (March 2015) 367–386
31. Solmaz, B., Moore, B.E., Shah, M.: Identifying behaviors in crowd scenes using stability analysis for dynamical systems. Pattern Analysis and Machine Intelligence, IEEE Transactions on **34**(10) (2012) 2064–2070
32. Kratz, L., Nishino, K.: Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In: Computer Vision and Pattern Recognition. IEEE Conference on, IEEE (2009) 1446–1453
33. Srivastava, S., Ng, K., Delp, E.: Crowd flow estimation using multiple visual features for scenes with changing crowd densities. In: Advanced Video and Signal-

Based Surveillance (AVSS), 2011 8th IEEE International Conference on. (Aug 2011) 60–65
34. Chen, T.H., Chen, T.Y., Chen, Z.X.: An intelligent people-flow counting method for passing through a gate. In: Robotics, Automation and Mechatronics, 2006 IEEE Conference on, IEEE (2006) 1–6
35. Tsuduki, Y., Fujiyoshi, H.: A method for visualizing pedestrian traffic flow using sift feature point tracking. In: Advances in Image and Video Technology. Springer (2009) 25–36
36. Loy, C.C., Xiang, T., Gong, S.: Multi-camera activity correlation analysis. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. (June 2009) 1988–1995
37. Guy, S.J., Chhugani, J., Curtis, S., Dubey, P., Lin, M., Manocha, D.: Pledestrians: a least-effort approach to crowd simulation. In: Proceedings of the 2010 ACM SIGGRAPH/Eurographics symposium on computer animation, Eurographics Association (2010) 119–128
38. Andrade, E., Blunsden, S., Fisher, R.: Modelling crowd scenes for event detection. In: Pattern Recognition, 2006. ICPR 2006. 18th International Conference on. Volume 1. (2006) 175–178
39. Dee, H., Caplier, A.: Crowd behaviour analysis using histograms of motion direction. In: Image Processing (ICIP), 2010 17th IEEE International Conference on. (Sept 2010) 1545–1548
40. Rodriguez, M.D., Ahmed, J., Shah, M.: Action mach a spatio-temporal maximum average correlation height filter for action recognition. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. (June 2008) 1–8
41. Reddy, K.K., Shah, M.: Recognizing 50 human action categories of web videos. Machine Vision and Applications **24**(5) (2013) 971–981
42. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. CoRR **abs/1212.0402** (2012)
43. Shao, J., Loy, C., Wang, X.: Scene-independent group profiling in crowd. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. (June 2014) 2227–2234
44. Zhou, B., Tang, X., Zhang, H., Wang, X.: Measuring crowd collectiveness. Pattern Analysis and Machine Intelligence, IEEE Transactions on **36**(8) (Aug 2014) 1586–1599
45. Hassner, T., Itcher, Y., Kliper-Gross, O.: Violent flows: Real-time detection of violent crowd behavior. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on. (June 2012) 1–6
46. J. Xu, D. Vazquez, A.M.L.J.M.D.P.: Learning a part-based pedestrian detector in virtual world. IEEE Transactions on Intelligent Transportation Systems (2014)
47. Le Bon, G.: The crowd: A study of the popular mind. Macmillian (1897)
48. James, J.: The distribution of free-forming small group size. American Sociological Review (1953)
49. Turner, R.H., Killian, L.M.: Collective behavior (pp. 1–14, 16) (1987)
50. Pervin, L.: The Science of Personality. Oxford University Press (2003)
51. Moussaïd, M., Perozo, N., Garnier, S., Helbing, D., Theraulaz, G.: The walking behaviour of pedestrian social groups and its impact on crowd dynamics. (2010)
52. Eysenck, H., Eysenck, M.: Personality and individual differences: A natural science perspective (1985)
53. Guy, S.J., Kim, S., Lin, M.C., Manocha, D.: Simulating heterogeneous crowd behaviors using personality trait theory. In: Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, ACM (2011) 43–52

54. Van Den Berg, J., Guy, S.J., Lin, M., Manocha, D.: Reciprocal n-body collision avoidance. In: Robotics research. Springer (2011) 3–19
55. Curtis, S., Best, A., Manocha, D.: Menge: A modular framework for simulating crowd movement. Collective Dynamics **1** (2016) 1–40
56. Funge, J., Tu, X., Terzopoulos, D.: Cognitive modeling: knowledge, reasoning and planning for intelligent characters. In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co. (1999) 29–38
57. Ulicny, B., Thalmann, D.: Towards interactive real-time crowd behavior simulation. In: Computer Graphics Forum. Volume 21., Wiley Online Library (2002) 767–775
58. Shao, W., Terzopoulos, D.: Autonomous pedestrians. In: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation, ACM (2005) 19–28
59. Barraquand, J., Latombe, J.C.: Robot motion planning: A distributed representation approach. The International Journal of Robotics Research **10**(6) (1991) 628–649
60. Snook, G.: Simplified 3d movement and pathfinding using navigation meshes. In DeLoura, M., ed.: Game Programming Gems. Charles River Media (2000) 288–304
61. Lamarche, F., Donikian, S.: Crowd of virtual humans: a new approach for real time navigation in complex and structured environments. In: Computer Graphics Forum. Volume 23., Wiley Online Library (2004) 509–518
62. Geraerts, R., Kamphuis, A., Karamouzas, I., Overmars, M.: Using the corridor map method for path planning for a large number of characters. In: Motion in Games. Springer (2008) 11–22
63. Reynolds, C.: Flocks, herds and schools: A distributed behavioral model. In: Proc. of SIGGRAPH. (1987)
64. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Physical review E (1995)
65. van den Berg, J., Guy, S., Lin, M., Manocha, D.: Reciprocal n-body collision avoidance. In Pradalier, C., Siegwart, R., Hirzinger, G., eds.: Robotics Research. Volume 70 of Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg (2011) 3–19
66. Bruderlin, A., Calvert, T.W.: Goal-directed, dynamic animation of human walking. In: Proc. of SIGGRAPH '89. (1989) 233–242
67. Lee, K.H., Choi, M.G., Hong, Q., Lee, J.: Group behavior from video: a data-driven approach to crowd simulation. In: Symposium on Computer Animation. (2007) 109–118
68. van Basten, B.J.H., Stuvel, S.A., Egges, A.: A hybrid interpolation scheme for footprint-driven walking synthesis. Graphics Interface (2011) 9–16
69. Oliver, P.: Unreal engine 4 elemental. In: ACM SIGGRAPH 2012 Computer Animation Festival. SIGGRAPH '12, New York, NY, USA, ACM (2012) 86–86
70. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Volume 1. (June 2005) 886–893 vol. 1
71. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. (June 2012) 3354–3361
72. Ess, A., Leibe, B., Schindler, K., Gool, L.V.: A mobile vision system for robust multi-person tracking. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. (June 2008) 1–8

73. Benfold, B., Reid, I.: Stable multi-target tracking in real-time surveillance video. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. (June 2011) 3457–3464
74. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NIPS). (2015)
75. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2014) 580–587
76. Girshick, R.: Fast r-cnn. In: 2015 IEEE International Conference on Computer Vision (ICCV). (Dec 2015) 1440–1448
77. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)
78. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R.B., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. CoRR **abs/1408.5093** (2014)
79. S. Narang, A. Best, T.R.A.S., Manocha, D.: Fbcrowd: Interactive multi-agent simulation with coupled collision avoidance and human motion synthesis. In: Technical Report, Department of Computer Science, UNC Chapel Hill, 2016