

NOVA: Rendering Virtual Worlds with Humans for Computer Vision Tasks

A. Kerim , C. Aslan, U. Celikcan , E. Erdem  and A. Erdem 

Hacettepe University, Department of Computer Engineering, Turkey



Figure 1: A sample panorama displaying procedurally generated humans by the NOVA framework in a controllable, configurable environment along with their annotations. The first half is photorealistic renderings transitioning between different times of day and the latter half is demonstrating some of the pixel-level annotations NOVA generates for use in various computer vision tasks: (from left to right) instance segmentation, semantic segmentation, optical flow, surface normals and the depth data.

Abstract

Today the cutting edge of computer vision research greatly depends on the availability of large datasets, which are critical for effectively training and testing new methods. Manually annotating visual data, however, is not only a labor-intensive process but also prone to errors. In this study, we present NOVA, a versatile framework to create realistic-looking 3D rendered worlds containing procedurally generated humans with rich pixel-level ground truth annotations. NOVA can simulate various environmental factors such as weather conditions or different times of day, and bring diverse set of human agents to life, each having a distinct body shape, gender and age. To demonstrate NOVA’s capabilities, we generate two synthetic datasets for person tracking. The first one consists of 108 sequences, each with different levels of difficulty like tracking in crowded scenes or at nighttime and aims for testing the limits of current state-of-the-art trackers. A second dataset of 97 sequences with normal weather conditions is used to show how our synthetically generated sequences can be utilized to train and boost the performance of deep-learning based trackers. Our results indicate that the synthetic data generated by NOVA represents a good proxy of the real-world and can be exploited for computer vision tasks.

CCS Concepts

• *Computing methodologies* → *Rendering; Tracking;*

1. Introduction

The rapid progress in the field of computer vision and other AI related disciplines has been significantly driven by learning based methods, most notably those based on deep learning. Getting the best out of these approaches, however, broadly depends on the availability of large training data, and hence a major bottleneck on the way towards solving many computer vision tasks is the lack of diverse, accurate and large scale datasets. Manually curating such large datasets is labor-intensive and often error-prone. Although Amazon’s Mechanical Turk or similar services can alleviate those

issues, these tools are very expensive, especially for small research groups, if one wishes to capture the real-world in its full glory. But maybe more importantly, such crowdsourcing platforms become impractical for collecting ground truth data for some computer vision tasks (e.g. optical flow estimation). A neat idea to overcome these difficulties is to utilize synthetic data for machine learning, which has gained momentum over the past few years.

Recent improvements in game technologies have made the creation of photorealistic and physically accurate games possible. Since designing virtual worlds from scratch can be very expen-

1 sive and requires highly skilled artists, it is possible to make use of
2 the games that are already available. Making modifications on an
3 open-sourced game or capturing the information sent by the game
4 to graphics card can help to generate large synthetic datasets. How-
5 ever, the fact that commercial games do not represent a proxy of
6 many real-world scenarios poses an essential problem with this ap-
7 proach, limiting its benefits.

8 Another way to create large synthetic datasets is to design the
9 virtual world based on the needs. While it usually requires more
10 effort to create and configure, this approach makes it possible to
11 produce a high-fidelity proxy of the targeted scenarios. With the
12 advances in graphics engine capabilities within the past decade, the
13 photorealistic and physically-based simulations realized by using
14 these engines allowed to minimize the gap between real and virtual
15 world data.

16 Procedural generation has been proposed as a solution for creat-
17 ing realistic looking environments in relatively short amounts of
18 time, making it easier and cheaper for users to generate virtual
19 worlds from scratch. In its simplest form, a procedural generation
20 framework follows some systematic recipes and generates scenes,
21 populations and actions, based on the given set of instructions. Our
22 work contributes to this line of research, in which we pay special
23 attention to the human generation aspect - in addition to offering
24 a comprehensive variety of automatic ground truth annotation fea-
25 tures that are partially available in other synthetic data generation
26 frameworks.

27 The large-scale benchmark datasets that were collected in the
28 past few years [DDS*09; LMB*14; KH09; GZW*] has lead to the
29 unprecedented progress in deep learning based computer vision ap-
30 proaches. Although the exponential increase in the amount of digi-
31 tal data today can make data collection easier than before, manual
32 labeling of large volumes of examples with high quality and accu-
33 rate labels still requires too much effort and comes with a tremen-
34 dous cost. Our proposed NOVA framework, with its procedural and
35 automated generation capabilities, provides a solution to this daunt-
36 ing data collection/annotation challenge by letting the users create
37 and render 3D virtual worlds containing human agents with dif-
38 ferent characteristics in real-time. Since users have full control of
39 the scenes, scene elements and humans, along with the illumination
40 and weather conditions, NOVA allows them to study many factors
41 affecting the success of their algorithms during development time.

42 2. Related Work

43 Creating realistic scenery, humans, actions and materials that
44 mimic their actual world counterparts has been a major aim since
45 the early days of video games. However, such a goal was not possi-
46 ble until recently. The ability to create photorealistic and physically
47 accurate games motivated many researchers to investigate the possi-
48 bility of utilizing them for the task of synthetic data generation.
49 The works in this scope fall under either of the two main method-
50 ologies. The first is to adapt a specific game for the task of generat-
51 ing the synthetic dataset as in the works by Richter et al. [RVRK16;
52 RHK17] where Grand Theft Auto V game was adapted to generate
53 synthetic datasets. Essentially, they exploited the communication
54 between the game and graphics hardware via injection of a mid-
55 dleware between the two to pull the necessary information for the

56 desired annotations. Another work [TCB07] modified Half-Life 2
57 game to evaluate a surveillance camera system. Using their pro-
58 posed Object Video Virtual Video (OVVV) framework, they were
59 able to generate bounding boxes and accurate segmentation labels
60 for arbitrary number of frames automatically. In addition to that,
61 they discussed how it is possible to integrate some noise and de-
62 formation techniques to produce more natural and realistic scenes.
63 Similarly, [SLS16] deployed a photorealistic video game to gener-
64 ate a large set of synthetic images, which were used to train a con-
65 volutional neural network for depth estimation and image segmen-
66 tation. They concluded with many experiments that pre-training on
67 synthetic data or training on both synthetic and real data achieve
68 similar or better results compared to using only organic data for
69 the training process. Nevertheless, using existing video games has
70 the significant disadvantage of lacking diversity, as it does limit the
71 number of scenarios, environments, actions, objects, and humans
72 that can be included in a synthetic dataset.

73 The second methodology adopts using a graphics engine for data
74 generation rather than individual video games. [QY16] used this
75 concept by providing a plugin for Unreal Engine to generate ground
76 truth for certain computer vision tasks by making some modifica-
77 tions on the internal data structures of a game and controlling a vir-
78 tual camera to explore the scenes. Similarly, [HUI13] used an open
79 source driving simulator framework, VDrift, to generate a synthetic
80 dataset, which incorporates high resolution images with their cor-
81 responding ground truth labels for semantic segmentation, depth
82 and optical maps, specifically for multiclass image segmentation.
83 A conditional random field model was trained with the synthetic
84 data and used to analyze how various combinations of features af-
85 fect the segmentation performance.

86 As an alternative, it is possible to refer to the open source anima-
87 tion movies to modify the rendering process to generate certain an-
88 notations along with the movie frames. One work [BWSB12] used
89 this method for generating a synthetic optical flow dataset. They
90 showed that optical flow statistics of their synthetic sequences and
91 real video sequences are in agreement. Moreover, the dataset pro-
92 vided was larger than Middlebury [BSL*11] and KITTI [GLU12]
93 which allowed further studies on optical flow research. However,
94 the inability to modify the scene structure of the animation consti-
95 tutes the main drawback with this approach, making it even more
96 limited for the purpose of synthetic data generation than using
97 available photorealistic games.

98 Perhaps the most unrestricted way of creating arbitrarily large
99 datasets together with their automated ground truth labels is tak-
100 ing the approach of using a graphics engine further by making
101 use of procedural generation techniques in virtual world creation.
102 De Souza et al. [DGCP17] investigated the possibility of adapt-
103 ing this concept with ragdoll physics, random perturbations and
104 muscle weakening to generate a wide range of human actions sys-
105 tematically with their corresponding labels. They have defined 17
106 actions and showed that integrating the real-world data with their
107 generated synthetic data can enhance the recognition performance.
108 Another work [CWB*16] applied the concept of procedural gener-
109 ation to generate labeled crowd videos. As a proof of concept,
110 it was shown that integrating their generated synthetic data with
111 real-world data can improve the crowd behavior classifier's accu-

1 racy and the overall performance of pedestrian detection notice-
2 ably. Wrenninge et al. [WU18] demonstrated a photorealistic and
3 diverse synthetic dataset that can be generated entirely procedu-
4 rally. The ability to parameterize the scene generation process and
5 the fact that these parameters are not correlated are the main contri-
6 butions of this work. They showed that training on their synthetic
7 dataset and fine-tuning on organic dataset gives better performance
8 compared to training only on the latter one only.

9 Due to the advancements in real-time rendering, the number of
10 synthetic datasets that can be used for a wide spectrum of com-
11 puter vision tasks has seen a considerable boost in the recent years.
12 PHAV (Procedural Human Action Videos) [DGCP17] dataset is an
13 example of a large scale synthetic dataset that was generated pro-
14 cedurally. It is mainly proposed for action recognition, and con-
15 tains around 6 million frames in total. Another example, LCCrowdV
16 (Labeled Crowd Video) [CWB*16] dataset, which was produced
17 by applying procedural modeling and rendering techniques, can be
18 used for tasks such as pedestrian count, flow estimation and ob-
19 ject detection and has more than 20 millions frames. On the other
20 hand, there is VKITTI (Virtual KITTI) [GWCV16] dataset of ap-
21 proximately 21 thousand frames which can be used for multi-object
22 tracking, scene level and instance level semantic segmentation and
23 depth estimation in addition to object detection and optical flow es-
24 timation. SYNTHIA (Synthetic Collection of Imagery and Anno-
25 tations) dataset [RSM*16], with more than 200 thousand images,
26 is purposed for semantic segmentation and scene understanding of
27 outdoor scenes for autonomous driving tasks. However, being spe-
28 cially designed for driving scenarios makes it inapplicable for many
29 other computer vision tasks. Another similar and recent dataset is
30 ParallelEye [LWT*18] which was generated by taking images from
31 a synthetic car moving in a virtual city and contains around 40 thou-
32 sand frames. It can be used for several tasks such as object detec-
33 tion, semantic and instance segmentation, and optical flow.

34 With our proposed NOVA framework, our main aim is to fur-
35 ther advance the efforts in computer vision by facilitating the au-
36 tomated creation of new arbitrarily large synthetic datasets with an
37 extensive variety of ground truth annotations. NOVA lets users eas-
38 ily create photorealistic 3D virtual worlds containing procedurally
39 generated humans, and allows to obtain frame and pixel-level an-
40 notations about a scene and its elements in real-time, making it
41 a versatile framework for automatic data collection and labeling
42 pipeline for a wide range of tasks including but not limited to vi-
43 sual tracking, crowd counting, semantic segmentation, optical flow
44 estimation, and depth estimation. It can simulate several illumina-
45 tion and weather conditions such as fog, rain, snow, daytime, night-
46 time, which help to test both favorable and adverse settings for
47 these tasks. Moreover, procedural generation capabilities of NOVA
48 allows to generate unique synthetic humans with very diverse char-
49 acteristics regarding body shape, gender, age and clothing, making
50 NOVA a perfect tool for generating realistic-looking synthetic data
51 for problems involving persons.

52 3. NOVA: Framework of Rendering Virtual Worlds with 53 People for Computer Vision Tasks

54 Our framework NOVA is built on the widely used Unity graph-
55 ics engine. The framework, when all annotations are enabled (ex-

56 cept bounding boxes, which are computed offline) and the num-
57 ber of synthetic humans to be generated is set to vary between
58 5 and 15, runs at real-time speeds (rendering between 42 and 60
59 frames per second on average) using current generation hardware
60 (Intel Core i7-7700HQ, GeForce GTX 1070, with SSD and 32GB
61 RAM). Readers are referred to visit the project website <https://graphics.cs.hacettepe.edu.tr/NOVA> for an online
62 demo of the framework that allows to observe all procedural gener-
63 ation and visual ground-truth annotation features of NOVA at real-
64 time by adjusting various scene-level attributes.

65 NOVA consists of the following data generation and annotation
66 features to facilitate the creation of arbitrarily large datasets for a
67 diverse array of computer vision tasks from pedestrian detection to
68 scene understanding.

70 3.1. Humans

71 NOVA populates an environment with synthetic humans on a ran-
72 dom selection of predefined spawning points that are within the
73 view volume of the generated camera. A sparsity parameter is used
74 to control the distribution of the spawning points, which determines
75 the level of human crowdedness in the view.

76 The synthetic humans are procedurally generated at run-time by
77 making use of several content creation layers which consist of pre-
78 defined set of categorizable, annotatable randomizations as well as
79 procedural, low-level randomizations in order to preserve unique-
80 ness in generated humans even in arbitrarily large sets (Fig. 2). This
81 population process is built upon the publicly available UMA sys-
82 tem [Sys].

83 To procedurally generate a synthetic human, a unique body and
84 face shape for the human are first created from either male or fe-
85 male base meshes. The blend shape set to be used for morphing
86 the body mesh to build different body types is calculated from a
87 base set of pre-determined body attributes. For each gender, there
88 are three sets of height types (*short, average, tall*), three sets of
89 weight types (*thin, athletic, overweight*), and two sets of age types
90 (*child, adult*) available. One from every attribute type is randomly
91 selected and the values are blended together considering their ef-
92 fects on different morph points. For instance, a tall child, while be-
93 ing taller than the average of the children generated, would still be
94 shorter than an average adult. After calculating a distinguishable
95 and annotatable body type (e.g., '*short athletic adult male*'), the
96 calculated blend shapes are further randomized by applying a rather
97 small white noise with uniform distribution to each of the morph
98 sections in order to ensure uniqueness while still resembling the
99 tagged body type for the human. This process theoretically allows
100 to create infinitely many unique bodies which can be categorized
101 into 36 major body types.

102 After calculating the bones, a set of clothes and facial attributes
103 are generated for the synthetic human from a set of recipes, which
104 create a content instance by mixing and recoloring several recipe
105 items in unique ways (Table 1). For example, a recipe for creating
106 a beard texture contains three options for beard masks which are
107 randomly selected in varying numbers, blended together (if more
108 than one mask is selected) and used for applying a beard matched to
109 the human's hair color, potentially generating eight different beard



Figure 2: A sample of 21 synthetic humans (in focus) from a set containing 9112 unique humans generated by NOVA.

Table 1: Statistics about unique item variations in the procedural generation of synthetic humans. Possible variations in color are additionally provided inside parentheses.

Facial Items			Clothing and Accessory Items		
Item	Male	Female	Item	Male	Female
Hair	4 (48)	3 (32)	Upper-Body Clothing	7 (28)	7 (28)
Eyebrows	2 (24)	2 (24)	Lower-Body Clothing	6 (240)	13 (520)
Beard	8 (96)	- / -	Outerwear	2 (80)	3 (120)
			Shoes	5 (40)	10 (80)
			Bags	3 (12)	3 (12)
			Other	2 (4)	3 (18)

1 shapes. On the other hand, a recipe for choosing a shoe is relatively
 2 simple and selects one of the shoe meshes provided for the corre-
 3 sponding gender.

4 A shared color system is used for applying colors, such that, each
 5 recipe chooses a color from a set of different palettes for skin, hair
 6 and clothing types. These colors are then multiplied with one of the
 7 alternative mask textures in order to yield variety in hair and skin
 8 textures and clothing patterns. The resulting colored and patterned
 9 textures are then used as the diffuse channel of the material while
 10 others (specular channel, gloss channel, etc.) are kept unchanged
 11 in order to retain correct physically-based material properties. This
 12 recoloring scheme allows us to further diversify the created humans
 13 while still keeping an easily categorizable generation system.

14 The resulting meshes from the recipe-based generation process
 15 are skinned onto the skeleton with the body mesh and the additional
 16 texture masks which are used to cull the body parts that will be cov-
 17 ered by these meshes are added onto the base mesh textures during
 18 sampling. Fig. 2 shows an arbitrarily chosen subset of a sample of
 19 9112 unique humans generated by NOVA. Although the instances
 20 in the figure are arranged with respect to perceptual similarity, it
 21 can be seen that even the humans in the small subset are still easily
 22 distinguishable from one another.

23 The animations for the humans are procedurally generated by
 24 blending between several motion captured animation sets including
 25 standing idle, walking, running and arguing. In order to create a
 26 unique motion instance at each time, two of these sets are randomly
 27 chosen and blended together using random weighting with uniform
 28 distribution.

29 3.2. Environments

30 Currently, NOVA can create sequences in three outdoor environ-
 31 ments (a town square, a suburban street and a metropolitan ur-
 32 ban district) and one indoor environment (a subway station) (Fig.
 33 3a). Each environment is equipped with at least 20 different spawn
 34 points, which are selected at random during population process.
 35 Lighting in the 3D environments is parametrically generated to sim-
 36 ulate different hours of a day (Fig. 3b) and weather types based on
 37 sun direction and altitude (Fig. 3c). The skybox, which provides
 38 ambient lighting for the 3D environments, and the weather effects
 39 are procedurally generated using the Enviro system [Wor].

40 Moreover, NOVA also makes use of HDR cubemaps that are cap-
 41 tured from real-life (Fig. 3d). In this case, the synthetic human re-
 42 ceives directional lighting from the virtual sun and ambient light-
 43 ing from the cubemap by using the image-based lighting method
 44 [Deb02]. In order to blend the generated human with the environ-
 45 ment further, the shadow that would be cast by the human on the
 46 ground is simulated by using a transparent plane, which receives
 47 shadow from the human’s mesh. Although the background seems
 48 more realistic compared to the 3D environments, the drawback to
 49 using cubemaps is that illumination and weather changes can not
 50 be applied to them procedurally without ending up looking non-
 51 realistic in general.

52 3.3. Cameras

53 NOVA simulates different camera types as follows.



(a) Sample images of the 3D environments. First row: a subway station. Second row from left: a metropolitan urban district, a town square, and a suburban street.



(b) Different times of day.



(c) Various weather conditions.



(d) Samples using HDR cubemaps [Zaa] captured from real-world

Figure 3: Illustrating the diversity in NOVA’s computer-rendered synthetic environments.

Algorithm 1: Algorithm for Non-Surveillance Camera Operation

```

Activate Camera Paths for the Specified Camera Type;
Set Camera Parameters;
 $ID_{tracked} \leftarrow$  ID of the Synthetic Human Being Tracked;
 $ID_{tracked}.Collider.Radius \leftarrow$  Higher Collider Radius Value
than Others;
foreach  $CameraPathCollider \in Active\ Camera\ Path$ 
  Colliders do
    if  $CameraPathCollider$  is triggered by
       $ID_{tracked}.Collider$  then
        Set the Camera Attached to  $CameraPathCollider$  as
        the Active Camera;
         $ID_{tracked}.Collider.Radius \leftarrow$  Regular Collider
        Radius Value;
        Set the Active Camera to Follow and Look at the
        Object Rotating about  $ID_{tracked}.Joints.Hip$ ;
    while  $ID_{tracked}$  is occluded do
      Wait;
  Start Recording;

```

17 cally rendered for each frame. All annotations, except the textual
18 metadata, are at the pixel-level.

19 For each screen-space annotation, a separate camera is created
20 and each camera uses different shaders, shader-specific parameters
21 and culling parameters in order to create that annotation’s frame.
22 An effects shader containing sub-shaders for the annotations is set
23 to each of these cameras as replacement shader which then uses the
24 sub-shader with the matching render type of the specified annotation.
25 That is, the camera renders the scene as it normally would, i.e.,
26 the objects still use their own materials, but the actual shader that
27 ends up being used for annotation is changed, overriding shaders
28 for regular rendering, and, instead, outputting the annotation.

29 **Optical Flow.** For the optical flow pass, the pixel motions are en-
30 coded in screen UV space to a screen-sized RG16 (16-bit float per
31 channel) texture. Color encoding is done according to per-pixel mo-
32 tion vectors with respect to the camera. This information comes
33 from an extra render pass into which moving objects are rendered
34 and their motion is constructed with respect to inter-frame differ-
35 ences. Different optical flow annotation schemes can be applied by
36 changing mappings for the encoding in order to make it compatible
37 with existing datasets. Fig. 4b exemplifies two such alternative en-
38 coding schemes. Optical flow sensitivity can be adjusted as desired
39 so that the amount of movement that is to be observed is encoded
40 in a normalized manner.

41 **Surface Normals.** During the surface normals pass, surfaces are
42 color encoded according to their orientation with respect to the
43 camera (Fig. 4c). Encoding is done using stereographic projection
44 into a 16 bit value which is packed into two 8 bit channels of a
45 screen-sized texture. This information comes directly from the G-
46 buffer.

47 **Depth Map.** For the depth map creation, pixels are gray-level in-
48 dexed based on per-pixel distance to the camera (Fig. 4d). The in-

1 *Surveillance Cameras:* include both static and PTZ type surveil-
2 lance cameras. The PTZ camera performs panning, tilting and
3 zooming to keep the human being tracked in its field-of-view.

4 *Non-Surveillance Cameras:* include UAV and ground-level camera
5 types. The first one simulates a camera attached to a UAV while
6 the second one imitates a pedestrian carrying a camera and record-
7 ing others. For each type, there is a predefined set of camera paths,
8 which has a separate camera assigned per path, in each environ-
9 ment. The non-surveillance camera operation is outlined in Algo-
10 rithm 1. To avoid having the tracked human always right in the
11 middle of the view, the camera follows a virtual object rotating in
12 an orbit around the human’s hip instead of tracking the human di-
13 rectly.

14 3.4. Ground Truth Annotations

15 NOVA automatically generates ground-truth annotations on-the-fly
16 as the simulated scene is procedurally created and photorealistic

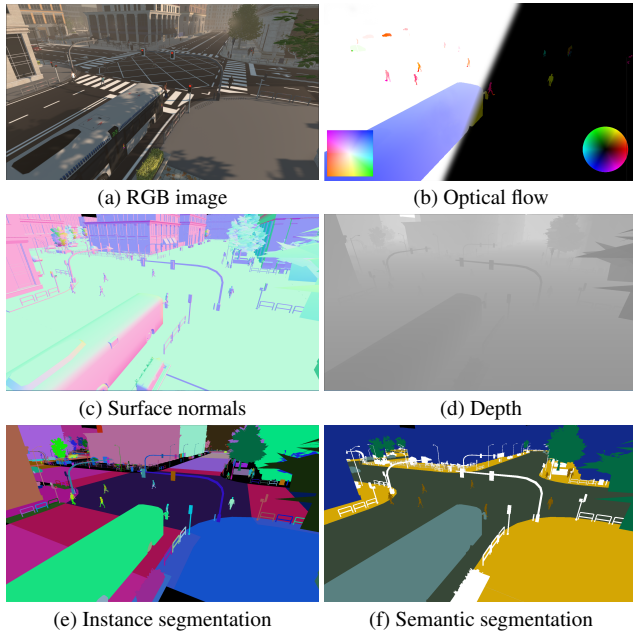


Figure 4: Sample of scene level annotations automatically generated by NOVA.

1 formation for depth map textures comes directly from the actual
2 depth buffer which is also a product of the G-buffer rendering.

3 **Instance Segmentation.** For every frame, each distinct entity
4 within the camera view is assigned a unique identifier color
5 (Fig. 4e).

6 **Semantic Segmentation.** Entities within the camera view are also
7 assigned colors based on layers representing their category, e.g.,
8 human, vehicle, road (Fig. 4f). The variety of categories can be
9 expanded as desired by defining additional layers.

10 While creating instance segmentation and semantic segmentation
11 frames, unique object identifiers and layers are encoded into
12 RGB color values, set into a block of material values and passed

13 into the shaders to be used. This process is repeated every time a
14 change occurs in the scene, e.g. when a new human is generated.

15 NOVA can also provide the class and instance -level segmentation
16 maps for which only a set of chosen objects are culled, e.g., to
17 generate ground truth data for person tracking, everything except
18 the synthetic humans in the frame are culled. These masked ver-
19 sions work in the same fashion as their non-masked counterparts
20 but are rendered using a separate camera instance that only uses
21 that set's layer for culling.

22 **Bounding Box.** For the bounding boxes, NOVA provides a seg-
23 mentation that masks each human in view with a different color.
24 This segmentation is used for min-max calculations to compute the
25 per-frame bounding box for each human. Since this process takes
26 considerably more time than the other annotations NOVA gener-
27 ates, especially for crowded simulations, the second step is carried
28 out offline once all the other data is generated at real-time.

29 **Body Part Segmentation.** Body part segmentation of a synthetic
30 human (Fig. 5c) is generated by assigning separate vertex colors to
31 each vertex for torso, head, arms and legs. For this, NOVA checks
32 the bone weights of every vertex of a human mesh when it is first
33 generated. Each vertex is assigned to one of the six colors for the
34 respective body part depending on the weights of the bones that
35 the vertex is connected to. The colors are then linearly interpolated
36 during the fragment stage to achieve the final result. This process
37 allows scalability as it can be carried only once when a synthetic
38 human is first generated, allowing to keep using GPU for skinning
39 with a higher frame rate during rendering.

40 **Body Pose.** To create the body pose information of a synthetic hu-
41 man in a frame, the positions of the skeletal joints are transferred
42 into the screen-space and output as values normalized with respect
43 to image size. In addition to the screen-space positions of the joints,
44 NOVA also outputs a depth value per joint which can be used to re-
45 solve conflicts such as overlapping or occlusion. The output is in
46 textual metadata form to allow flexibility in visualization. For in-
47 stance, the body pose visualization in Fig. 5d is compatible with
48 the keypoint detection format of COCO dataset [LMB*14].

49 **Other Textual Annotations.** Some other attributes (see Fig. 5e)



Figure 5: Sample of human-level annotations automatically generated for a synthetic human.

1 of a generated human that are not suitable to be output as im-
 2 age modalities are output as textual metadata. Most of these at-
 3 tributes were chosen to reflect the ones which are present in existing
 4 datasets of real images purposed for person re-identification. Fur-
 5 thermore, a set of frame level annotations most of which identify
 6 miscellaneous environment parameters that were used to generate
 7 the frame are also included in the textual annotations of that frame.
 8 The frame level annotations include the environment type, weather
 9 and time of day markers, and applied post-fx presets (if any).

10 4. Experimental Analysis

11 In this section, using visual tracking as a test bed, we demon-
 12 strate how the proposed framework can be used to create realistic-
 13 looking and diverse synthetic datasets with auto-generated ground
 14 truth annotations. In our analysis, we specifically carry out two dif-
 15 ferent sets of experiments. First, we demonstrate how our frame-
 16 work can be used to generate synthetic sequences with various chal-
 17 lenging scenarios to evaluate the limits of state-of-the-art trackers
 18 (Sec. 4.3). Second, we show how our synthetically generated se-
 19 quences can be utilized for training to boost the performance of
 20 deep-learning based visual trackers (Sec. 4.4). Before the analysis,
 21 we first briefly review the existing datasets proposed for tracking
 22 (Sec. 4.1) and present the evaluation measures used in our experi-
 23 ments (Sec. 4.2).

24 4.1. Existing Tracking Datasets

25 Tracking humans in videos is one of the most important topics
 26 in computer vision, with applications ranging from video surveil-
 27 lance to activity analysis. However, the widely-used benchmark
 28 datasets such as OTB100 [WLY15], VOT [KML*16; KML*19]
 29 and TC128 [LBL15], which are indeed proposed for evaluat-
 30 ing generic object trackers, have relatively small number of
 31 instances containing humans as objects of interest. Some datasets
 32 provide tracking sequences under very specific conditions, e.g.
 33 UAV123 [MSG16] that presents sequences for low altitude UAV
 34 cameras and NUS-PRO [LLW*16] that contains videos that are
 35 mostly recorded by moving cameras. There exists some datasets
 36 that are specifically built for evaluating human trackers, such as
 37 DUKEMTMC [RSZ*16], CamNeT [ZSFR15], MOT [MLR*16]
 38 and NLPR-MCT [CCC*15], but these are mainly limited in both
 39 size and variability since obtaining annotated data for this task is
 40 difficult and time consuming. Either the sequences are captured
 41 with fixed cameras so the backgrounds are in general static or
 42 the lightning conditions do not vary much. To alleviate such
 43 shortcomings, in our experiments, we specifically focus on the
 44 task of tracking humans and use NOVA to generate two different
 45 datasets containing sequences with different levels of difficulty.
 46 Fig. 6 shows some sample sequences from our synthetic datasets,
 47 together with real-world sequences from NUS-PRO [LLW*16],
 48 TC128 [LBL15], UAV123 [MSG16], OTB100 [WLY15],
 49 VOT [KML*16; KML*19], and MOT [MLR*16] datasets. It is
 50 seen that NOVA is able to generate sequences that are compatible
 51 with the real-world sequences.

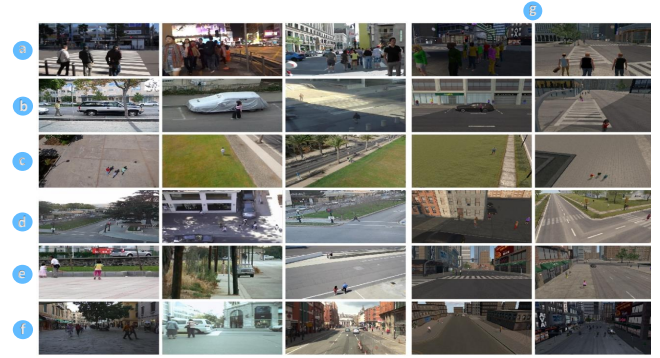


Figure 6: Real vs. synthetic sequences. In terms of appearance, the sequences in (a) NUS-PRO, (b) TC128, (c) UAV123, (d) OTB100, (e) VOT, and (f) MOT datasets (first three frames in each row) are compatible with the synthetic ones produced by (g) NOVA (last two frames in each row).

52 4.2. Evaluation Measures

53 In our experiments, we consider *precision* and *expected average*
 54 *overlap* (EAO), two commonly used metrics in evaluating visual
 55 trackers. Precision calculates the distance between the center of
 56 tracker bounding box and ground truth bounding box and checks
 57 whether this center error is within specified limits. We employ the
 58 conventional threshold of 20 pixels and consider the tracking as ac-
 59 curate for a frame if the center error is smaller than this value. We
 60 then extract the percentage of accurately predicted bounding boxes
 61 for each sequence in our dataset. EAO, on the other hand, is used to
 62 express accuracy and robustness of the tracker performance with a
 63 single score. At the beginning, the tracker is initialized and allowed
 64 to track the target until the end of the sequence or failure. When
 65 the tracker fails, it is reinitialized again and this process is repeated
 66 a number of times (3 times in our case). The mean of the average
 67 overlaps between the predicted and the ground truth bounding
 68 boxes gives EAO.

69 4.3. Using Synthetic Data to Evaluate Visual Trackers

70 **Data Generation.** To assess the limits of current state-of-the-art
 71 trackers, we use NOVA to generate a new synthetic dataset called
 72 VirtualPTB1 (Virtual Person Tracking Benchmark #1), unique in
 73 terms of its characteristics. As can be seen in Table 2, it includes
 74 sequences with different adverse weather conditions, crowdedness
 75 levels, and challenging factors due to different times of day and
 76 camera altitudes. VirtualPTB1 consists of 108 sequences, which are
 77 on average 5 secs long and have more than 13K frames altogether,
 78 along with per-frame bounding boxes for the persons of interest.
 79 The sequences are annotated with a total of 17 attributes from 6
 80 different classes. Fig. 7 presents sample frames from VirtualPTB1
 81 exhibiting the diversity and the photorealism of the generated se-
 82 quences.

83 **Visual Trackers.** To analyze how the state-of-the-art generic ob-
 84 ject trackers perform on VirtualPTB1, we have selected six dif-
 85 ferent correlation filter based tracking approaches, which perform

Table 2: Distributions of attributes across the sequences in our synthetic person tracking dataset generated by using NOVA.

Attribute	Crowdedness			Camera Altitude			Times of the Day				Weather Condition				Occlusion		Scale Variation	
Sub-Attributes	1 Person	3 People	10 People	Low	Medium	High	Sunset/Sunrise	Midday	Night	Normal	Snow	Fog	Lightstorm	Low	High	No	Yes	
# of Sequences	36	36	36	36	36	36	36	36	36	27	27	27	27	80	28	58	50	

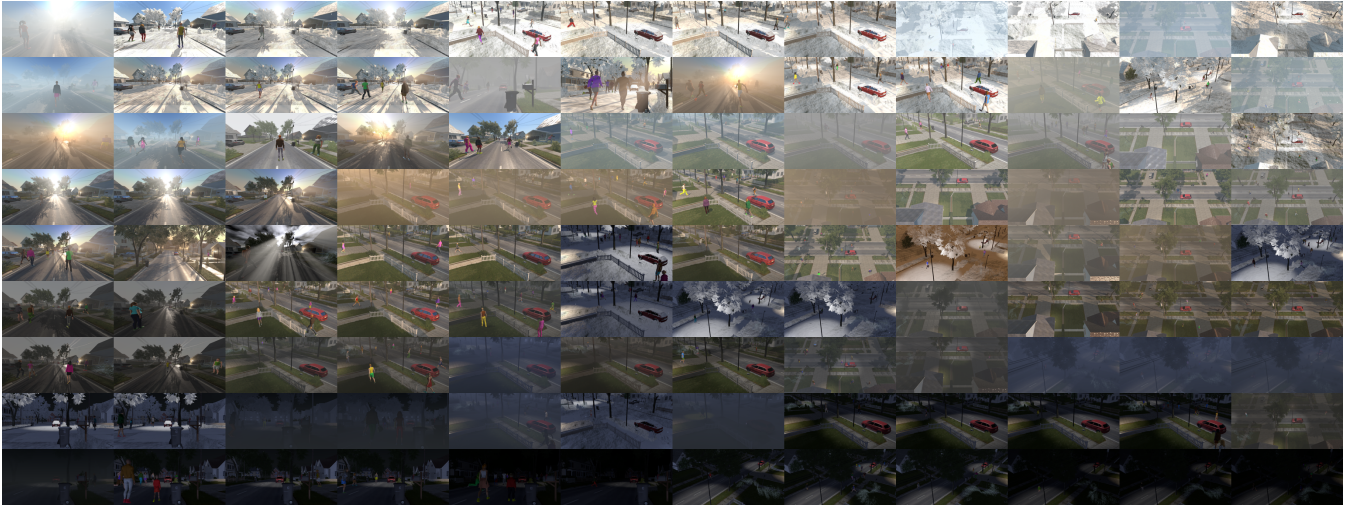


Figure 7: VirtualPTB1, our proposed synthetic tracking dataset, consists of 108 sequences, each with a unique set of attributes. The first frames of each sequence are shown here, illustrating the variations in crowdedness, camera altitude, weather conditions and times of day.

1 well on the existing tracking benchmark datasets. These are *ECO* 30
 2 [DBSF17], *BACF* [KFL17], and context aware (CA) [MSG17] 31
 3 versions of *MOSSE*, [BBDL10], *DCF* [HCMB15], *SAMF* [LZ14] and 32
 4 *STAPLE* [BVG*16]. 33

5 **Results.** In Fig. 8 and Fig. 9, we demonstrate the overall per- 34
 6 formances of the trackers on VirtualPTB1. As can be seen from 35
 7 Fig. 8, there are only a few sequences where the trackers give 36
 8 highly accurate results. In the remaining ones, they fail to precisely 37
 9 track the persons of interest, demonstrating how challenging Vir- 38
 10 tualPTB1 is. According to the precision rates, *ECO* tracker out- 39
 11 performs the others. *BACF* tracker and context aware versions of 40
 12 *STAPLE* and *SAMF* have nearly the same average precision scores 41
 13 although the sequences they show good performances are differ- 42
 14 ent. The examined trackers make use of different approaches and, 43
 15 hence, exhibit nonidentical performances on VirtualPTB1. Another 44
 16 key observation is that these scores are relatively low as compared 45
 17 to those reported in benchmark datasets containing real-world se- 46
 18 quences [DBSF17; KFL17; MSG17]. This is in line with our de- 47
 19 sign objectives for VirtualPTB1 as it introduces certain challenges 48
 20 which are mostly not present in the available benchmark sets. Sam- 49
 21 ple qualitative tracking results can be found in the supplementary 50
 22 video. 51

23 Our detailed analysis reveals that tracking people in highly 52
 24 crowded scenes causes the trackers to lose the target very frequently 53
 25 as the persons of interest are highly likely to be occluded by the 54
 26 other persons. Moreover, it is noticed that the trackers perform 55
 27 poorly at night time and in foggy weather conditions. Under these 56
 28 circumstances, the trackers mostly cannot distinguish the tracked 57
 29 person from the background. Similarly, high camera altitude poses

certain challenges as well since such altitudes cause the target to 30
 appear very small and, consequently, very hard to track. In Fig. 10, 31
 the corresponding precision plots for these challenging attributes 32
 are shown. Please refer to the supplementary material for an ex- 33
 tended presentation and discussion of the results. 34

4.4. Using Synthetic Data to Train Visual Trackers

Data Generation and Collection. For our second set of experi- 35
 ments, we employ NOVA to generate a large volume of synthetic 36
 sequences that can be used to train deep learning based trackers. We 37
 consider different training scenarios including synthetic and real se- 38
 quences, and also a hybrid of those. In contrast to the former part, 39
 we carry our analysis on real test sequences for this set of experi- 40
 ments. In particular, NOVA is used generate 97 synthetic sequences 41
 and their ground truths annotations with pixel-level accuracy. How- 42
 ever, to match the characteristics of the available real datasets, we 43
 limit the weather attribute to normal weather conditions, namely, 44
 clear-sky and three different variations of cloudy weather condi- 45
 tions. At the same time, we vary all other procedural generation 46
 parameters such as time of day, camera type, scene crowdedness 47
 and environment. In creating this set, it was aimed to mimic the 48
 general pattern of the existing real-world datasets, maintaining both 49
 the photorealism and the diversity at compatible levels. 50
 51

In addition to the created synthetic dataset, we collect 125 real- 52
 world sequences from OTB100 [WLY15], VOT [KML*16; 53
 KML*19], TC128 [LBL15], UAV123 [MSG16], NUS- 54
 PRO [LLW*16] and MOT [MLR*16] datasets. We especially pick 55
 the sequences containing humans in outdoor environments and 56
 under normal weather conditions. Finally, we randomly divide 57

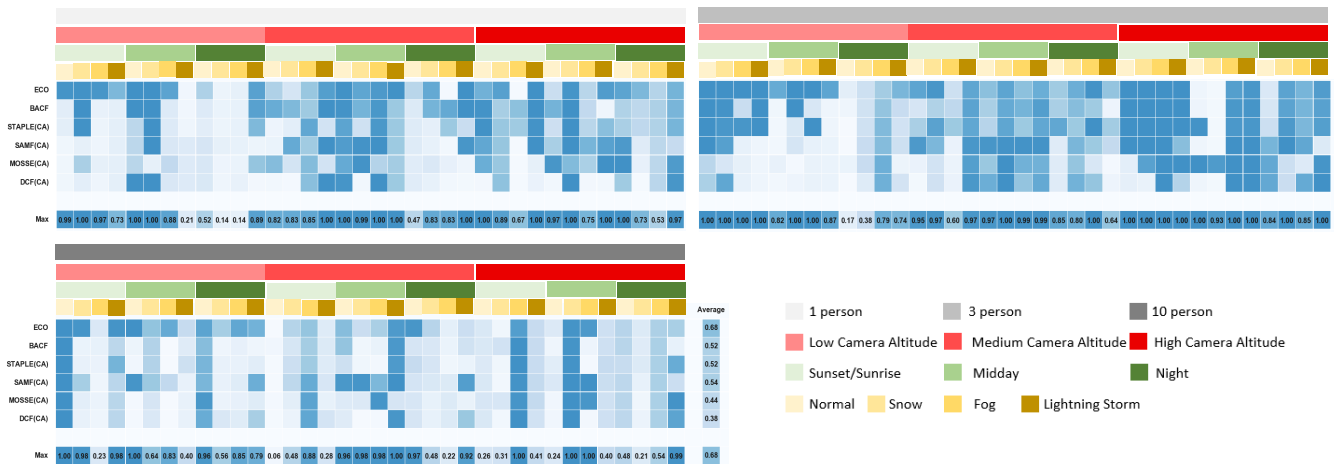


Figure 8: Heatmap showing the precision of each tracker on each sequence of VirtualPTB1. The last row (Max) indicates the maximum performance achieved by the set of trackers on each sequence. The last column (Average) shows the average precision of a specific tracker over all sequences. Each color indicates different scene attribute. Gray, red, green and orange bars demonstrate scene crowdedness, camera altitude, time of day and weather condition, respectively, for a specific sequence below them by color variations that indicate their sub-attributes as given in the legend.

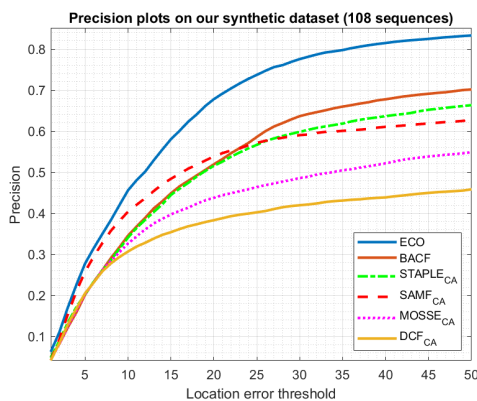


Figure 9: Precision plot of the evaluated trackers on our dataset.

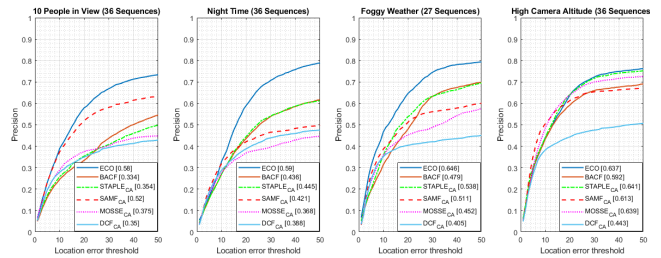


Figure 10: Precision plots for the four challenging cases. Crowded scenes, night time, foggy weather and high camera altitude all cause a clear performance degradation.

1 these 125 real sequences into training and testing parts, where 97
 2 sequences were selected for training and 28 for testing. Please
 3 refer to the supplementary material for some sample frames from
 4 the synthetic and real-world sequences used.

5 **Visual Trackers.** We employ two state-of-the-art deep trackers in our
 6 experiments, namely CFNet [VBH*17] and DiMP [BDGT19].
 7 Correlation filter based tracking (CFNet) is a deterministic, end-
 8 to-end representation learning tracker which considers correlation
 9 filter (CF) as a differentiable layer in a CNN architecture. This al-
 10 lows the error gradients to pass through the CF layer and tune the
 11 CNN features. DiMP, on the other hand, is a deep-learning based
 12 tracker that depends on Siamese architecture which accounts for
 13 the target and the background information while predicting the tar-

14 get object’s location. The parameters of the tracker is learned in an
 15 end-to-end manner using a discriminative loss function.

16 **Training Protocol.** We consider training scenarios for the two deep
 17 trackers in two different schemes, as follows.

18 *Training from Scratch.* In the first scheme, we train each tracker
 19 from scratch by randomly initializing the model parameters using a
 20 different training set in each training scenario. The first scenario in-
 21 volves training the trackers using only the synthetic sequences gen-
 22 erated by NOVA (E1). For the second one, the trackers are trained
 23 by employing only the real sequences from the training split of the
 24 dataset we collected (E2). Finally, in the last scenario, we consider
 25 a hybrid approach and explore the advantages of expanding the set
 26 of real sequences with the synthetic ones and training the trackers
 27 using this combined set (E3).

28 *Fine-Tuning.* For this scheme, instead of training the trackers from
 29 scratch, we perform fine-tuning considering their pre-trained ver-
 30 sions again in three different scenarios. In the first and the second

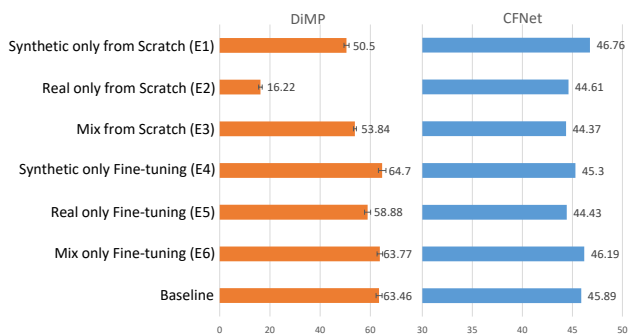


Figure 11: EAO scores obtained with the six different training scenarios as compared to those of the baselines. Error bars on the DiMP results give the standard deviation of the EAO score. Fine-tuning the baselines on a mixture of synthetic and real sequences improves the performance. At the same time, training on synthetic sequences alone achieves better results compared to training solely on real sequences.

scenarios, the trackers are fine-tuned considering only the synthetic sequences (E4) and only the real training sequences (E5), respectively. The third scenario involves fine-tuning using the hybrid set containing both the synthetic and real sequences (E6).

Results. In Fig. 11, the results of our quantitative analysis are presented with the average overlap scores for DiMP and CFNet trackers obtained with each training scenario and compared to the baseline scores. Given the stochastic nature of DiMP tracker, we report the average and the standard deviation of its results for five repetitions. While training the trackers from scratch, using the synthetic sequences achieves better results as compared to using real sequences. Basically, this advantage can be attributed to the diverse and realistic nature of our synthetically generated sequences, which cover different environments, including indoor and outdoor ones, diverse weather conditions, multiple time-of-the days, various camera types and distinctive humans. These factors enrich the generalization capability of the trained trackers, allowing them to learn better features and lead to more accurate results even on the real testing sequences. Moreover, comparable performances with the baseline models are achieved using only 97 synthetic sequences. Note that, in their original setting, the baseline CFNet model was trained using 3862 sequences with more than 1 million frames while the baseline DiMP model was trained by four different datasets, namely, LaSOT [FLY*19], GOT10k [HZH19], TrackingNet [MBG*18], and COCO [LMB*14], which amount to a much larger set than the number of our training sequences. As for our fine-tuning experiments, we found out that fine-tuning the baseline models of DiMP and CFNet trackers on the mixture of synthetic and real sequences further improves their performances as expected. The gain is especially significant for CFNet, whose baseline model was pre-trained on ILSVRC Video dataset that does not contain humans as objects of interest. Another important observation is that fine-tuning the baselines only on our synthetic sequences seems more advantageous than fine-tuning on real-world sequences alone. This further demonstrates the advantage of using our synthetic data.

5. Conclusion

In this work, we have presented a novel engine called NOVA for creating photorealistic 3D rendered worlds containing synthetic humans, along with ground truth annotations at scene, object and pixel-levels. The proposed framework automates data collection and labeling pipeline for a wide range of low and high-level computer vision tasks. In particular, the engine emphasizes procedural generation of humans, which makes NOVA unique compared to existing systems. It allows to produce diverse arrays of human agents, in terms of body shape, clothing, gender and age characteristics, accessories and action variety. Moreover, NOVA lets users play with weather and illumination conditions within the created 3D virtual worlds, establishing it as a test bed for evaluating adverse cases such as low light, nighttime, rain, snow, or fog. These capabilities make NOVA a distinct and versatile framework to quickly generate arbitrarily large amounts of synthetic data for a multitude of computer vision tasks. These large synthetic datasets can be used in model training to boost the performance of state-of-the-art learning based computer vision models.

As a case study, we considered visual tracking and employed our proposed NOVA framework to create two different datasets for different purposes. The first dataset, VirtualPTB1, includes 108 sequences with automatically generated ground truths and a total of 17 scene level attributes. Under short-term tracking scenarios, the sequences demonstrate a wide variety of factors including weather conditions, times of day, overall crowdedness of the scene, camera altitude, occlusion and scale variation. Our thorough analysis of various state-of-the-art trackers on VirtualPTB1 sheds light on trackers' weaknesses in adverse conditions such as high crowdedness, high camera altitude, night time, and foggy weather. Our second synthetic dataset, on the other hand, consists of 97 sequences with normal weather conditions. We have used this dataset to train two deep trackers, CFNet and DiMP. Our results reveal that using our synthetic sequences during training leads to a performance boost in several aspects for both of these trackers. Thus, it is shown that the variety and the level of realism of the scene attributes in our dataset make it a good proxy of the real-world for evaluating and training visual trackers.

An online demo of NOVA along with videos illustrating its capabilities and some tracking results and the VirtualPTB1 dataset are available at the project website <https://graphics.cs.hacettepe.edu.tr/NOVA>.

As a future work, we plan to increase the procedural generation capability of our NOVA framework even further, especially regarding generation of dynamic scene elements other than humans. The feasibility of using physically based rendering will be explored for enhancing the level of provided photorealism. Moreover, using NOVA, we are planning to generate a special benchmark for evaluating the performance of general purpose trackers under adverse weather conditions. At the same time, extending this work to help solving other computer vision tasks like semantic segmentation and instance segmentation can be another research direction.

References

[BBDL10] BOLME, DAVID S., BEVERIDGE, J. ROSS, DRAPER, BRUCE A., and LUI, YUI MAN. "Visual object tracking using adaptive corre-

- 1 lation filters". *The IEEE Conference on Computer Vision and Pattern*
2 *Recognition (CVPR)*. 2010 **8**.
- 3 [BDGT19] BHAT, GOUTAM, DANELLJAN, MARTIN, GOOL, LUC VAN,
4 and TIMOFTE, RADU. "Learning discriminative model prediction for
5 tracking". *Proceedings of the IEEE International Conference on Com-*
6 *puter Vision*. 2019, 6182–6191 **9**.
- 7 [BSL*11] BAKER, SIMON, SCHARSTEIN, DANIEL, LEWIS, JP, et al. "A
8 database and evaluation methodology for optical flow". *International*
9 *Journal of Computer Vision* 92.1 (2011), 1–31 **2**.
- 10 [BVG*16] BERTINETTO, LUCA, VALMADRE, JACK, GOLODETZ, STU-
11 ART, et al. "Staple: Complementary learners for real-time tracking". *Pro-*
12 *ceedings of the IEEE conference on computer vision and pattern recog-*
13 *nition*. 2016, 1401–1409 **8**.
- 14 [BWSB12] BUTLER, DANIEL J, WULFF, JONAS, STANLEY, GARRETT
15 B, and BLACK, MICHAEL J. "A naturalistic open source movie for
16 optical flow evaluation". *European Conference on Computer Vision*.
17 Springer. 2012, 611–625 **2**.
- 18 [CCC*15] CAO, LIJUN, CHEN, WEIHUA, CHEN, XIAOTANG, et al. "An
19 equalised global graphical model-based approach for multi-camera ob-
20 ject tracking". *arXiv preprint arXiv:1502.03532* (2015) **7**.
- 21 [CWB*16] CHEUNG, ERNEST, WONG, TSAN KWONG, BERA, ANIKET,
22 et al. "Lcrowdv: Generating labeled videos for simulation-based crowd
23 behavior learning". *European Conference on Computer Vision*. Springer.
24 2016, 709–727 **2, 3**.
- 25 [DBSF17] DANELLJAN, MARTIN, BHAT, GOUTAM, SHAHBAZ KHAN,
26 FAHAD, and FELSBURG, MICHAEL. "Eco: Efficient convolution oper-
27 ators for tracking". *Proceedings of the IEEE Conference on Computer*
28 *Vision and Pattern Recognition*. 2017, 6638–6646 **8**.
- 29 [DDS*09] DENG, J., DONG, W., SOCHER, R., et al. "ImageNet: A Large-
30 Scale Hierarchical Image Database". *CVPR09*. 2009 **2**.
- 31 [Deb02] DEBEVEC, PAUL. "Image-based lighting". *IEEE Computer*
32 *Graphics and Applications* 22.2 (2002), 26–34 **4**.
- 33 [DGCP17] DE SOUZA, CÉSAR ROBERTO, GAIDON, ADRIEN, CABON,
34 YOHANN, and PEÑA, ANTONIO MANUEL LÓPEZ. "Procedural Gener-
35 ation of Videos to Train Deep Action Recognition Networks." *CVPR*.
36 2017, 2594–2604 **2, 3**.
- 37 [FLY*19] FAN, HENG, LIN, LITING, YANG, FAN, et al. "Lasot: A high-
38 quality benchmark for large-scale single object tracking". *Proceedings*
39 *of the IEEE Conference on Computer Vision and Pattern Recognition*.
40 2019, 5374–5383 **10**.
- 41 [GLU12] GEIGER, ANDREAS, LENZ, PHILIP, and URTASUN, RAQUEL.
42 "Are we ready for autonomous driving? the kitti vision benchmark
43 suite". *Computer Vision and Pattern Recognition (CVPR)*, 2012 *IEEE*
44 *Conference on*. IEEE. 2012, 3354–3361 **2**.
- 45 [GWCV16] GAIDON, ADRIEN, WANG, QIAO, CABON, YOHANN, and
46 VIG, ELEONORA. "Virtual worlds as proxy for multi-object tracking
47 analysis". *Proceedings of the IEEE conference on computer vision and*
48 *pattern recognition*. 2016, 4340–4349 **3**.
- 49 [GZW*] GE, YUYING, ZHANG, RUIMAO, WU, LINGYUN, et al. "A Ver-
50 satile Benchmark for Detection, Pose Estimation, Segmentation and Re-
51 Identification of Clothing Images". () **2**.
- 52 [HCMB15] HENRIQUES, JOÃO F, CASEIRO, RUI, MARTINS, PEDRO,
53 and BATISTA, JORGE. "High-speed tracking with kernelized correlation
54 filters". *IEEE transactions on pattern analysis and machine intelligence*
55 37.3 (2015), 583–596 **8**.
- 56 [HUI13] HALTAKOV, VLADIMIR, UNGER, CHRISTIAN, and ILIC, SLO-
57 BODAN. "Framework for generation of synthetic ground truth data for
58 driver assistance applications". *German Conference on Pattern Recogni-*
59 *tion*. Springer. 2013, 323–332 **2**.
- 60 [HZH19] HUANG, LIANGHUA, ZHAO, XIN, and HUANG, KAIQI. "Got-
61 10k: A large high-diversity benchmark for generic object tracking in the
62 wild". *IEEE Transactions on Pattern Analysis and Machine Intelligence*
63 (2019) **10**.
- 64 [KFL17] KIANI GALOOGAHI, HAMED, FAGG, ASHTON, and LUCEY, SI-
65 MON. "Learning background-aware correlation filters for visual track-
66 ing". *Proceedings of the IEEE International Conference on Computer*
67 *Vision*. 2017, 1135–1143 **8**.
- 68 [KH09] KRIZHEVSKY, ALEX and HINTON, GEOFFREY. *Learning multi-*
69 *ple layers of features from tiny images*. Tech. rep. Citeseer, 2009 **2**.
- 70 [KML*16] KRISTAN, MATEJ, MATAS, JIRI, LEONARDIS, ALEŠ, et al. "A
71 Novel Performance Evaluation Methodology for Single-Target Track-
72 ers". *IEEE Transactions on Pattern Analysis and Machine Intelligence*
73 38.11 (Nov. 2016), 2137–2155. ISSN: 0162-8828. DOI: [10.1109/](https://doi.org/10.1109/TPAMI.2016.2516982)
74 [TPAMI.2016.2516982](https://doi.org/10.1109/TPAMI.2016.2516982) **7, 8**.
- 75 [KML*19] KRISTAN, MATEJ, MATAS, JIRI, LEONARDIS, ALES, et al.
76 "The seventh visual object tracking vot2019 challenge results". *Proce-*
77 *edings of the IEEE International Conference on Computer Vision Work-*
78 *shops*. 2019, 0–0 **7, 8**.
- 79 [LBL15] LIANG, PENG PENG, BLASCH, ERIK, and LING, HAIBIN. "En-
80 coding color information for visual tracking: Algorithms and bench-
81 mark". *IEEE Transactions on Image Processing* 24.12 (2015), 5630–
82 5644 **7, 8**.
- 83 [LLW*16] LI, A, LIN, M, WU, Y, et al. "NUS-PRO: A New Visual Track-
84 ing Challenge". *IEEE Transactions on Pattern Analysis and Machine In-*
85 *telligence* 38.2 (2016), 335–349 **7, 8**.
- 86 [LMB*14] LIN, TSUNG-YI, MAIRE, MICHAEL, BELONGIE, SERGE, et al.
87 "Microsoft coco: Common objects in context". *European conference*
88 *on computer vision*. Springer. 2014, 740–755 **2, 6, 10**.
- 89 [LWT*18] LI, XUAN, WANG, KUNFENG, TIAN, YONGLIN, et al. "The
90 ParallelEye Dataset: A Large Collection of Virtual Images for Traffic
91 Vision Research". *IEEE Transactions on Intelligent Transportation Sys-*
92 *tems* 99 (2018), 1–13 **3**.
- 93 [LZ14] LI, YANG and ZHU, JIANKE. "A scale adaptive kernel correlation
94 filter tracker with feature integration". *European conference on computer*
95 *vision*. Springer. 2014, 254–265 **8**.
- 96 [MBG*18] MULLER, MATTHIAS, BIBI, ADEL, GIANCOLA, SILVIO, et al.
97 "Trackingnet: A large-scale dataset and benchmark for object tracking in
98 the wild". *Proceedings of the European Conference on Computer Vision*
99 *(ECCV)*. 2018, 300–317 **10**.
- 100 [MLR*16] MILAN, ANTON, LEAL-TAIXÉ, LAURA, REID, IAN, et al.
101 "MOT16: A benchmark for multi-object tracking". *arXiv preprint*
102 *arXiv:1603.00831* (2016) **7, 8**.
- 103 [MSG16] MUELLER, MATTHIAS, SMITH, NEIL, and GHANEM,
104 BERNARD. "A benchmark and simulator for uav tracking". *European*
105 *conference on computer vision*. Springer. 2016, 445–461 **7, 8**.
- 106 [MSG17] MUELLER, MATTHIAS, SMITH, NEIL, and GHANEM,
107 BERNARD. "Context-Aware Correlation Filter Tracking". *The IEEE*
108 *Conference on Computer Vision and Pattern Recognition (CVPR)*. July
109 2017 **8**.
- 110 [QY16] QIU, WEICHAO and YUILLE, ALAN. "Unrealcv: Connecting
111 computer vision to unreal engine". *European Conference on Computer*
112 *Vision*. Springer. 2016, 909–916 **2**.
- 113 [RHK17] RICHTER, STEPHAN R, HAYDER, ZEESHAN, and KOLTUN,
114 VLADLEN. "Playing for benchmarks". *Proceedings of the IEEE Inter-*
115 *national Conference on Computer Vision*. 2017, 2213–2222 **2**.
- 116 [RSM*16] ROS, GERMAN, SELLART, LAURA, MATERZYNSKA,
117 JOANNA, et al. "The synthia dataset: A large collection of synthetic
118 images for semantic segmentation of urban scenes". *Proceedings of*
119 *the IEEE conference on computer vision and pattern recognition*.
120 2016, 3234–3243 **3**.
- 121 [RSZ*16] RISTANI, ERGYS, SOLERA, FRANCESCO, ZOU, ROGER, et
122 al. "Performance measures and a data set for multi-target, multi-
123 camera tracking". *European Conference on Computer Vision*. Springer.
124 2016, 17–35 **7**.
- 125 [RVRK16] RICHTER, STEPHAN R, VINEET, VIBHAV, ROTH, STEFAN,
126 and KOLTUN, VLADLEN. "Playing for data: Ground truth from com-
127 puter games". *European Conference on Computer Vision*. Springer.
128 2016, 102–118 **2**.

- 1 [SLS16] SHAFAEI, ALIREZA, LITTLE, JAMES J, and SCHMIDT, MARK.
2 “Play and learn: Using video games to train computer vision models”.
3 *arXiv preprint arXiv:1608.01745* (2016) 2.
- 4 [Sys] SYSTEM, UNITY MULTIPURPOSE AVATAR. *UMA git repo*. <https://github.com/umasteeringgroup/UMA>. Online; accessed:
5 // github.com/umasteeringgroup/UMA. Online; accessed:
6 2019-02-20 3.
- 7 [TCB07] TAYLOR, GEOFFREY R, CHOSAK, ANDREW J, and BREWER,
8 PAUL C. “Ovvv: Using virtual worlds to design and evaluate surveillance
9 systems”. *Computer Vision and Pattern Recognition, 2007. CVPR’07.*
10 *IEEE Conference on.* IEEE. 2007, 1–8 2.
- 11 [VBH*17] VALMADRE, JACK, BERTINETTO, LUCA, HENRIQUES, JOAO,
12 et al. “End-to-end representation learning for correlation filter based
13 tracking”. *Proceedings of the IEEE Conference on Computer Vision and*
14 *Pattern Recognition.* 2017, 2805–2813 9.
- 15 [WLY15] WU, Y., LIM, J., and YANG, M. “Object Tracking Benchmark”.
16 *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.9
17 (Sept. 2015), 1834–1848. ISSN: 0162-8828. DOI: [10.1109/TPAMI.](https://doi.org/10.1109/TPAMI.2014.2388226)
18 [2014.2388226](https://doi.org/10.1109/TPAMI.2014.2388226) 7, 8.
- 19 [Wor] WORLDS, PROCEDURAL. *Enviro webpage*. Online; accessed: 2019-
20 02-20. URL: [http://www.procedural-worlds.com/gaia/](http://www.procedural-worlds.com/gaia/gaia-extensions/enviro/4)
21 [gaia-extensions/enviro/4](http://www.procedural-worlds.com/gaia/gaia-extensions/enviro/4).
- 22 [WU18] WRENNINGE, MAGNUS and UNGER, JONAS. “Synscapes: A
23 photorealistic synthetic dataset for street scene parsing”. *arXiv preprint*
24 *arXiv:1810.08705* (2018) 3.
- 25 [Zaa] ZAAL, G. *HDRI Haven*. <https://hdrihaven.com/hdriis>.
26 Online; accessed: 2019-02-20 5.
- 27 [ZSFR15] ZHANG, SHU, STAUDT, ELLIOT, FALTEMIER, TIM, and ROY-
28 CHOWDHURY, AMIT K. “A camera network tracking (CamNeT) dataset
29 and performance baseline”. *2015 IEEE Winter Conference on Applica-*
30 *tions of Computer Vision.* IEEE. 2015, 365–372 7.