Volume 0 (1981), Number 0 pp. 1–13 COMPUTER GRAPHICS *forum*

NOVA: Rendering Virtual Worlds with Humans for Computer Vision Tasks

Abdulrahman Kerim[1,3†], Cem Aslan[1,†], Ufuk Celikcan[1], Erkut Erdem[1] and Aykut Erdem[2]

[1]Hacettepe University, Department of Computer Engineering, Ankara, Turkey
cemaslan96@outlook.com, ufuk.celikcan@gmail.com, erkut@cs.hacettepe.edu.tr
[2]Koç University, Department of Computer Engineering, Istanbul, Turkey
aerdem@ku.edu.tr
[3] Lancaster University, School of Computing and Communications, UK
a.kerim@lancaster.ac.uk

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 an exceptionally diverse set of humans 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 includes 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 synthetic 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.

Keywords: procedural content generation, synthetic-data for learning, visual tracking

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

† These authors share the first authorship of this work.

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

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

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

The large-scale benchmark datasets that were collected in the past few years [DDS*09; LMB*14; KH09; GZW*] has lead to the unprecedented progress in deep learning based computer vision approaches. Although the exponential increase in the amount of digital data today can make data collection easier than before, manual labeling of large volumes of examples with high quality and accurate labels still requires too much effort and comes with a tremendous cost. Our proposed NOVA framework, with its procedural and automated generation capabilities, provides a solution to this daunting data collection/annotation challenge by letting the users create and render 3D virtual worlds containing human agents with different characteristics in real-time. The authors in [DGCP17] previously proposed a similar framework but their focus is mainly on human action recognition and thus their framework has limited functionalities. On the other hand, in our proposed NOVA framework, the users have full control of the scenes, scene elements and humans, along with the illumination and weather conditions, allowing to study various factors affecting the success of their algorithms during development time and opening up a possibility being used in a wider range of computer vision tasks.

The main contributions of this work can be summarized as follows:

- We present a novel procedural content generation engine called NOVA. It is capable of generating large-scale and photo-realistic videos of human agents performing various actions on many different scenes along with the annotations for various computer vision tasks.
- Using our NOVA rendering engine, we generate two synthetic datasets specifically designed for person tracking. While we use the first dataset to assess the performance of existing visual trackers on various conditions, we employ the second one to train deep visual trackers to boost their performances on real sequences.
- Our experiments demonstrate that the existing trackers perform poorly in highly crowded scenes, or in scenes captured at night and in foggy weather conditions. Moreover, our generated synthetic sequences present a good proxy of the real sequences in that when used as training data, it improves the performances of deep visual trackers.

2. Related Work

Creating realistic scenery, humans, actions and materials that mimic their actual world counterparts has been a major aim since the early days of video games. However, such a goal was not possible until recently. The ability to create photorealistic and physically accurate games motivated many researchers to investigate the possibility of utilizing them for the task of synthetic data generation. The works in this scope fall under either of the two main methodologies. The first is to adapt a specific game for the task of generating the synthetic dataset as in the works by Richter et al. [RVRK16; RHK17] where Grand Theft Auto V game was adapted to generate synthetic datasets. Essentially, they exploited the communication between the game and graphics hardware via injection of a middleware between the two to pull the necessary information for the desired annotations. Another work [TCB07] modified Half-Life 2 game to evaluate a surveillance camera system. Using their proposed Object Video Virtual Video (OVVV) framework, they were able to generate bounding boxes and accurate segmentation labels for arbitrary number of frames automatically. In addition to that, they discussed how it is possible to integrate some noise and deformation techniques to produce more natural and realistic scenes. Similarly, [SLS16] deployed a photorealistic video game to generate a large set of synthetic images, which were used to train a convolutional neural network for depth estimation and image segmentation. They concluded with many experiments that pre-training on synthetic data or training on both synthetic and real data achieve similar or better results compared to using only organic data for the training process. Nevertheless, using existing video games has the significant disadvantage of lacking diversity, as it does limit the

number of scenarios, environments, actions, objects, and humans that can be included in a synthetic dataset.

The second methodology adopts using a graphics engine for data generation rather than individual video games. [QY16] used this concept by providing a plugin for Unreal Engine to generate ground truth for certain computer vision tasks by making some modifications on the internal data structures of a game and controlling a virtual camera to explore the scenes. Similarly, [HUI13] used an open source driving simulator framework, VDrift, to generate a synthetic dataset, which incorporates high resolution images with their corresponding ground truth labels for semantic segmentation, depth and optical maps, specifically for multiclass image segmentation. A conditional random field model was trained with the synthetic data and used to analyze how various combinations of features affect the segmentation performance.

As an alternative, it is possible to refer to the open source animation movies to modify the rendering process to generate certain annotations along with the movie frames. One work [BWSB12] used this method for generating a synthetic optical flow dataset. They showed that optical flow statistics of their synthetic sequences and real video sequences are in agreement. Moreover, the dataset provided was larger than Middlebury [BSL*11] and KITTI [GLU12] which allowed further studies on optical flow research. However, the inability to modify the scene structure of the animation constitutes the main drawback with this approach, making it even more limited for the purpose of synthetic data generation than using available photorealistic games.

Perhaps the most unrestricted way of creating arbitrarily large datasets together with their automated ground truth labels is taking the approach of using a graphics engine further by making use of procedural generation techniques in virtual world creation. De Souza et al. [DGCP17] investigated the possibility of adapting this concept with ragdoll physics, random perturbations and muscle weakening to generate a wide range of human actions systematically with their corresponding labels. They have defined 17 actions and showed that integrating the real-world data with their generated synthetic data can enhance the recognition performance. Another work [CWB*16] applied the concept of procedural generation to generate labeled crowd videos. As a proof of concept, it was shown that integrating their generated synthetic data with real-world data can improve the crowd behavior classifier's accuracy and the overall performance of pedestrian detection noticeably. Wrenninge et al. [WU18] demonstrated a photorealistic and diverse synthetic dataset that can be generated entirely procedurally. The ability to parameterize the scene generation process and the fact that these parameters are not correlated are the main contributions of this work. They showed that training on their synthetic dataset and fine-tuning on organic dataset gives better performance compared to training only on the latter one only.

Due to the advancements in real-time rendering, the number of synthetic datasets that can be used for a wide spectrum of computer vision tasks has seen a considerable boost in the recent years. PHAV (Procedural Human Action Videos) [DGCP17] dataset is an example of a large scale synthetic dataset that was generated procedurally. It is mainly proposed for action recognition, and contains around 6 million frames in total. Another example, LCrowdV (Labeled Crowd Video) [CWB*16] dataset, which was produced by applying procedural modeling and rendering techniques, can be used for tasks such as pedestrian count, flow estimation and object detection and has more than 20 millions frames. On the other hand, there is VKITTI (Virtual KITTI) [GWCV16] dataset of approximately 21 thousand frames which can be used for multi-object tracking, scene level and instance level semantic segmentation and depth estimation in addition to object detection and optical flow estimation. SYNTHIA (Synthetic Collection of Imagery and Annotations) dataset [RSM*16], with more than 200 thousand images, is purposed for semantic segmentation and scene understanding of outdoor scenes for autonomous driving tasks. However, being specially designed for driving scenarios makes it inapplicable for many other computer vision tasks. Another similar and recent dataset is ParallelEye [LWT*18] which was generated by taking images from a synthetic car moving in a virtual city and contains around 40 thousand frames. It can be used for several tasks such as object detection, semantic and instance segmentation, and optical flow.

As discussed above, using computed generated imagery has become an important research direction especially for data-hungry deep learning approaches. That being said, the existing frameworks have some drawbacks. For instance, the main limitation of the frameworks proposed in [RVRK16; RHK17; BWSB12] is that they do not allow to configure the virtual environments as they use existing computer games or computer generated movies while generating annotations for synthetic data. NOVA framework, on the other hand, lets the user to play with the environment along with the environment conditions such as weather, time of day, crowdedness and camera types. Moreover, including new features like new environments, new objects, or new character animations can be easily done due to its flexible design that supports procedural generation as opposed to the tools such as UnrealCV [QY16] which is just a plug-in for the Unreal game engine or the frameworks such as VDrift [HUI13] that only supports driving based scenarios.

Another advantage of NOVA lies in the annotations it supports. As compared to the frameworks suggested in [TCB07; SLS16], NOVA allows to extract a richer set of annotations for a user generated scene. These include accurate annotations for some low-level vision tasks such as scene depth, optical flow and surface maps, and annotations for some high-level tasks such as object detection, visual tracking, semantic segmentation and instance segmentation. Besides, from the human agents perspective, our main focus is not the human action recognition as in [DGCP17] or crowd behavior learning and counting in [CWB*16]. With the capability of procedurally generating a large and diverse set of synthetic humans and their character animations, it suggests a more generic solution which opens many possible applications.

With the proposed NOVA framework, our main aim is to further advance the efforts in computer vision by facilitating the automated creation of new arbitrarily large synthetic datasets with an extensive variety of ground truth annotations. NOVA lets users easily create photorealistic 3D virtual worlds containing procedurally generated humans, and allows to obtain frame and pixel-level annotations about a scene and its elements in real-time, making it a versatile framework for automatic data collection and labeling pipeline for a wide range of tasks including but not limited to vi-

sual tracking, crowd counting, semantic segmentation, optical flow estimation, and depth estimation. It can simulate several illumination and weather conditions such as fog, rain, snow, daytime, nighttime, which help to test both favorable and adverse settings for these tasks. Furthermore, procedural generation capabilities of NOVA allows to generate unique synthetic humans with very diverse characteristics regarding body shape, gender, age and clothing, making NOVA a perfect tool for generating realistic-looking synthetic data for problems involving persons.

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

Our framework NOVA is built on the widely used Unity graphics engine. The framework, when all annotations are enabled (except bounding boxes, which are computed offline) and the number of synthetic humans to be generated is set to vary between 5 and 15, runs at real-time speeds (rendering between 42 and 60 frames per second on average) using current generation hardware (Intel Core i7-7700HQ, GeForce GTX 1070, with SSD and 32GB RAM). Readers are referred to visit the project website https://graphics.cs.hacettepe.edu.tr/NOVA for an online demo of the framework that allows to observe all procedural generation and visual ground-truth annotation features of NOVA at real-time by adjusting various scene-level attributes.

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

3.1. Humans

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

The synthetic humans are procedurally generated at run-time by making use of several content creation layers which consist of a predefined set of categorizable, annotatable features as well as procedural, low-level randomizations to these features. The low-level randomizations further enhance the variations realized by the hand-tailored annotatable features in order to substantiate uniqueness in generated humans in arbitrarily large sets (Fig. 2). This population process is built upon the publicly available UMA system [Sys].

To procedurally generate a synthetic human, a unique body and face shape are first created from either male or female base meshes. The attribute set to morph the body mesh is calculated from a base set of pre-determined body attributes. For each gender, there are three sets of height types (*short, average, tall*), three sets of weight types (*thin, athletic, overweight*), and two sets of age types (*child, adult*) available. One from every attribute type is randomly selected and the values are blended together considering their effects on different morph points. For instance, a tall child, while being taller than the average of the children generated, would still be shorter than an average adult. Once a distinguishable and annotatable body type (e.g., '*short athletic adult male*') is realized from the blended attribute set, it is further randomized by applying a rather small white noise with uniform distribution to each morph point on the body in order to ensure uniqueness while still resembling the tagged body type. This process theoretically allows to create infinitely many unique bodies which can be categorized into 36 major body types.

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

Item	Male	Female
Hair	4 (48)	3 (32)
Eyebrows	2(24)	2 (24)
Beard	8 (96)	- / -

Clothing and Accessory Items

Item	Male	Female
Upper-Body Clothing	7 (28)	7 (28)
Lower-Body Clothing	6 (240)	13 (520)
Outerwear	2 (80)	3 (120)
Shoes	5 (40)	10 (80)
Bags	3 (12)	3 (12)
Other	2 (4)	3 (18)

Then, a set of clothes and facial attributes are generated for the synthetic human from a set of recipes, which create a content instance by mixing and recoloring several recipe items in unique ways (Table 1). For example, a recipe for creating a beard texture contains three options for beard masks which are randomly selected in varying numbers, blended together (if more than one mask is selected) and used for applying a beard matched to the human's hair color, potentially generating eight different beard shapes. On the other hand, a recipe for choosing a shoe is relatively simple and selects one of the shoe meshes provided for the corresponding gender.

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

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

The animations for the humans are procedurally generated by blending between several motion captured animation sets including standing idle, walking, running and arguing. In order to create a unique motion instance at each time, two of these sets are randomly chosen and blended together. The blending is handled using linear interpolation, such that a blended animation is an average of the separate animations weighted randomly by uniformly distributed

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

blending parameters. As the humans are created using a common rig structure that adapts automatically, each can be assigned a randomly blended animation with seamless instant mapping.

The employed motion sets are limited to the ones that are most commonly encountered within the compatible real-world datasets. Additional sets of motions can be easily incorporated into the framework to advance variety. It should be noted that the duration of the generated video sequences is not limited by the duration of the motion clips and NOVA can generate video sequences of arbitrary duration by looping the blended animations as needed.

The blended animations involving locomotion are kept consistent with the environment geometry by using Unity's navigation mesh system which facilitates path planning and obstacle avoidance along a path. The destination of a path is assigned randomly by NOVA and if the destination is reached before the sequence ends, a new one is assigned.

3.2. Environments

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

Moreover, NOVA also makes use of HDR cubemaps that are captured from real-life (Fig. 3d). In this case, the synthetic human receives directional lighting from the virtual sun and ambient lighting from the cubemap by using the image-based lighting method [Deb02]. In order to blend the generated human with the environment further, the shadow that would be cast by the human on the ground is simulated by using a transparent plane, which receives shadow from the human's mesh. Although the background seems more realistic compared to the 3D environments, the drawback to using cubemaps is that illumination and weather changes can not be applied to them procedurally without ending up looking non-realistic in general.

3.3. Cameras

NOVA simulates different camera types as follows.

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

Non-Surveillance Cameras: include UAV and ground-level camera types. The first one simulates a camera attached to a UAV while the second one imitates a pedestrian carrying a camera and recording others. For each type, there is a predefined set of camera paths, which has a separate camera assigned per path, in each environment. The non-surveillance camera operation is outlined in Algorithm 1. To avoid having the tracked human always right in the middle of the view, the camera follows a virtual object rotating in an orbit around the human's hip instead of tracking the human directly.

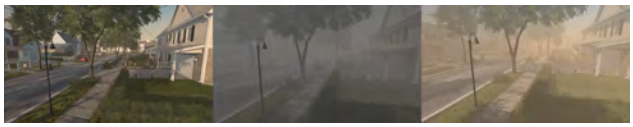
3.4. Ground Truth Annotations

NOVA automatically generates ground-truth annotations on-the-fly as the simulated scene is procedurally created and photorealistically rendered for each frame. All annotations, except the textual metadata, are at the pixel-level.

For each screen-space annotation, a separate camera is created and each camera uses different shaders, shader-specific parameters and culling parameters in order to create that annotation's frame. An effects shader containing sub-shaders for the annotations is set to each of these cameras as replacement shader which then uses the

(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.

sub-shader with the matching render type of the specified annotation. That is, the camera renders the scene as it normally would, i.e., the objects still use their own materials, but the actual shader that ends up being used for annotation is changed, overriding shaders for regular rendering, and, instead, outputting the annotation.

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

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

Algorithm 1: Algorithm for Non-Surveillance Camera Operation

```
Activate Camera Paths for the Specified Camera Type;
Set Camera Parameters;
ID_tracked ← ID of the Synthetic Human Being Tracked;
ID_tracked.Collider.Radius ← Higher Collider Radius Value
  than Others;
foreach CameraPathCollider ∈ 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 ← 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;
```

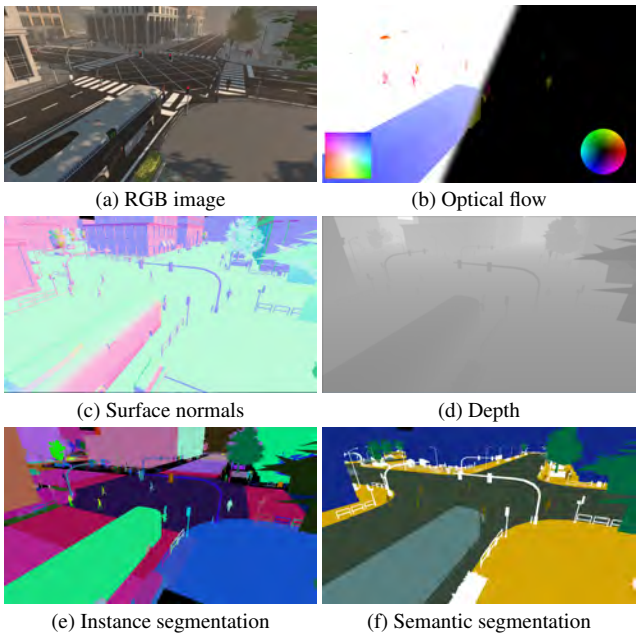
Depth Map. For the depth map creation, pixels are gray-level indexed based on per-pixel distance to the camera (Fig. 4d). The information for depth map textures comes directly from the actual depth buffer which is also a product of the G-buffer rendering.

Instance Segmentation. For every frame, each distinct entity within the camera view is assigned a unique identifier color representing its object ID (Fig. 4e). The view is then rendered by outputting the respective color without additional shading to obtain the instance segmentation pass.

Semantic Segmentation. Entities within the camera view are also assigned colors based on layers representing their category, e.g., human, vehicle, road (Fig. 4f). The assigned colors are then rendered without additional shading to obtain the semantic segmentation pass. The variety of categories can be expanded as desired by defining additional layers. The layers should be assigned to the respective objects or their prefabs during the content creation process.

While creating instance segmentation and semantic segmentation frames, unique object identifiers and layers are encoded into RGB color values, set into a block of material values and passed into the replacement shaders [Uni] to be used. This process is repeated every time a change occurs in the scene, e.g. when a new human is generated.

NOVA can also provide the class and instance -level segmentation maps for which only a set of chosen objects are culled, e.g., to generate ground truth data for person tracking, everything except the synthetic humans in the frame are culled. These masked versions work in the same fashion as their non-masked counterparts



(a) RGB image (b) Optical flow

(c) Surface normals (d) Depth

(e) Instance segmentation (f) Semantic segmentation

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

but are rendered using a separate camera instance that only uses that set's layer for culling.

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

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

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

Other Textual Annotations. Some other attributes (see Fig. 5e) of a generated human that are not suitable to be output as image modalities are output as textual metadata. Most of these attributes were chosen to reflect the ones which are present in existing datasets of real images purposed for person re-identification. Furthermore, a set of frame level annotations most of which identify miscellaneous environment parameters that were used to generate the frame are also included in the textual annotations of that frame. The frame level annotations include the environment type, weather and time of day markers, and applied post-fx presets (if any).

4. Experimental Analysis

In this section, using visual tracking as a test bed, we demonstrate how the proposed framework can be used to create realistic-looking and diverse synthetic datasets with auto-generated ground truth annotations. In our analysis, we specifically carry out two different sets of experiments. First, we demonstrate how our framework can be used to generate synthetic sequences with various challenging scenarios to evaluate the limits of state-of-the-art trackers (Sec. 4.3). Second, we show how our synthetically generated sequences can be utilized for training to boost the performance of

(a) RGB image (b) Bounding box (c) Body part segmentation (d) Body pose (e) Textual annotations

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

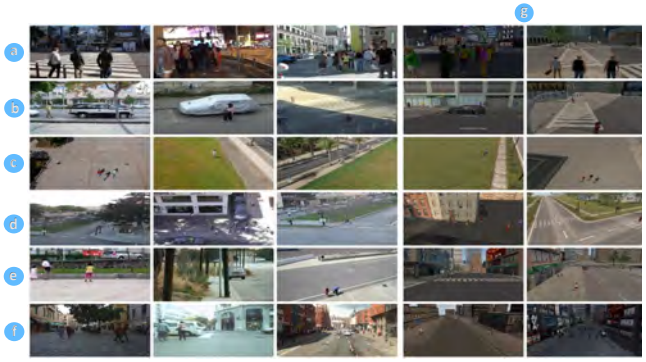


Figure 6: Real vs. synthetic sequences. In terms of appearance, the sequences in (a) NUS-PRO, (b) TC128, (c) UAV123, (d) OTB100, (e) VOT, and (f) MOT datasets (first three frames in each row) are compatible with the synthetic ones produced by (g) NOVA (last two frames in each row).

deep-learning based visual trackers (Sec. 4.4). Before the analysis, we first briefly review the existing datasets proposed for tracking (Sec. 4.1) and present the evaluation measures used in our experiments (Sec. 4.2).

4.1. Existing Tracking Datasets

Tracking humans in videos is one of the most important topics in computer vision, with applications ranging from video surveillance to activity analysis. However, the widely-used benchmark datasets such as OTB100 [WLY15], VOT [KML*16; KML*19] and TC128 [LBL15], which are indeed proposed for evaluating generic object trackers, have relatively small number of instances containing humans as objects of interest. Some datasets provide tracking sequences under very specific conditions, e.g. UAV123 [MSG16] that presents sequences for low altitude UAV cameras and NUS-PRO [LLW*16] that contains videos that are mostly recorded by moving cameras. There exists some datasets that are specifically built for evaluating human trackers, such as DUKEMTMC [RSZ*16], CamNeT [ZSFR15], MOT [MLR*16] and NLPR-MCT [CCC*15], but these are mainly limited in both size and variability since obtaining annotated data for this task is difficult and time consuming. Either the sequences are captured with fixed cameras so the backgrounds are in general static or the lightning conditions do not vary much. To alleviate such shortcomings, in our experiments, we specifically focus on the task of tracking humans and use NOVA to generate two different datasets containing sequences with different levels of difficulty. Fig. 6 shows some sample sequences from our synthetic datasets, together with real-world sequences from NUS-PRO [LLW*16], TC128 [LBL15], UAV123 [MSG16], OTB100 [WLY15], VOT [KML*16; KML*19], and MOT [MLR*16] datasets. It is seen that NOVA is able to generate sequences that are compatible with the real-world sequences. We provide a more detailed comparison between our synthetic sequences and the curated real sequences used in the experiments in the supplementary material.

4.2. Evaluation Measures

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

4.3. Using Synthetic Data to Evaluate Visual Trackers

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

Visual Trackers. To analyze how the state-of-the-art generic object trackers perform on VirtualPTB1, we have selected six different correlation filter based tracking approaches, which perform well on the existing tracking benchmark datasets. These are *ECO* [DBSF17], *BACF* [KFL17], and context aware (CA) [MSG17] versions of *MOSSE*, [BBDL10], *DCF* [HCMB15], *SAMF* [LZ14] and *STAPLE* [BVG*16].

Results. In Fig. 8 and Fig. 9, we demonstrate the overall performances of the trackers on VirtualPTB1. As can be seen from Fig. 8, there are only a few sequences where the trackers give highly accurate results. In the remaining ones, they fail to precisely track the persons of interest, demonstrating how challenging VirtualPTB1 is. According to the precision rates, ECO tracker outperforms the others. BACF tracker and context aware versions of STAPLE and SAMF have nearly the same average precision scores although the sequences they show good performances are different. The examined trackers make use of different approaches and, hence, exhibit nonidentical performances on VirtualPTB1. Another key observation is that these scores are relatively low as compared to those reported in benchmark datasets containing real-world sequences [DBSF17; KFL17; MSG17]. This is in line with our design objectives for VirtualPTB1 as it introduces certain challenges which are mostly not present in the available benchmark sets. Sam-

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

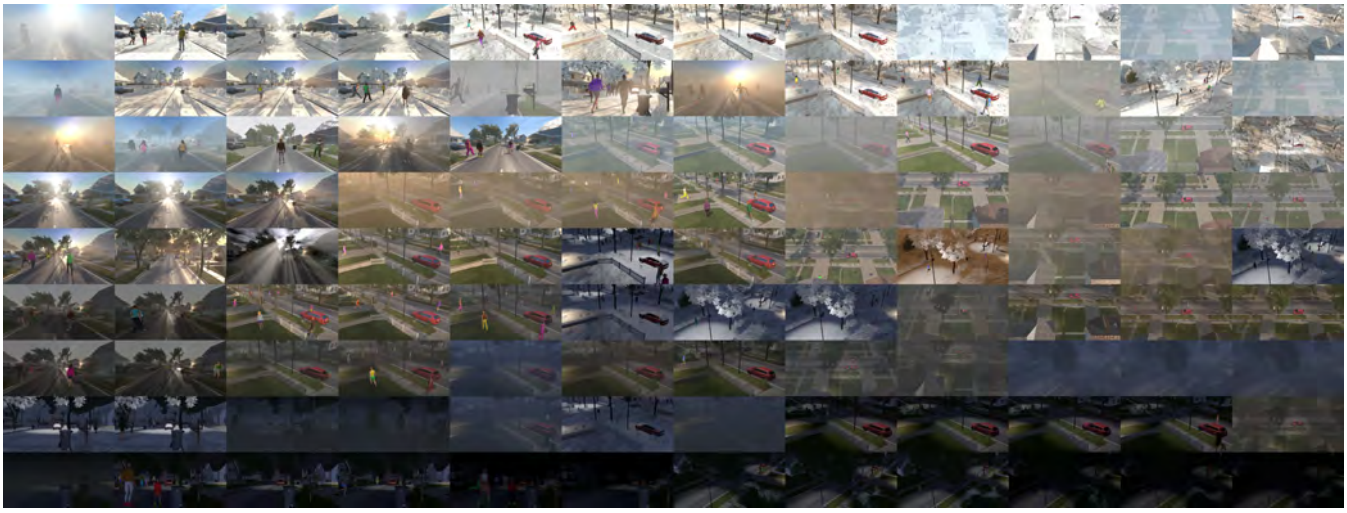


Figure 7: VirtualPTB1, our proposed synthetic tracking dataset, consists of 108 sequences, each with a unique set of attributes. The first frames of each sequence are shown here, illustrating the variations in crowdedness, camera altitude, weather conditions and times of day.

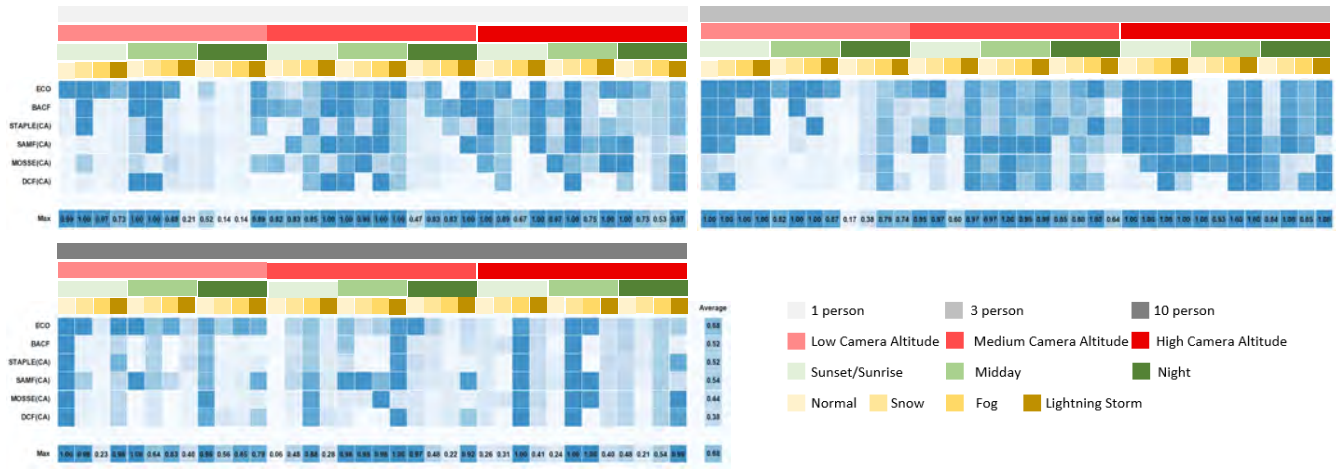


Figure 8: Heatmap showing the precision of each tracker on each sequence of VirtualPTB1. The last row (Max) indicates the maximum performance achieved by the set of trackers on each sequence. The last column (Average) shows the average precision of a specific tracker over all sequences. Each color indicates 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.

ple qualitative tracking results can be found in the supplementary video.

Our detailed analysis reveals that tracking people in highly crowded scenes causes the trackers to lose the target very frequently as the persons of interest are highly likely to be occluded by the other persons. Moreover, it is noticed that the trackers perform poorly at night time and in foggy weather conditions. Under these circumstances, the trackers mostly cannot distinguish the tracked person from the background. Similarly, high camera altitude poses certain challenges as well since such altitudes cause the target to appear very small and, consequently, very hard to track. In Fig. 10,

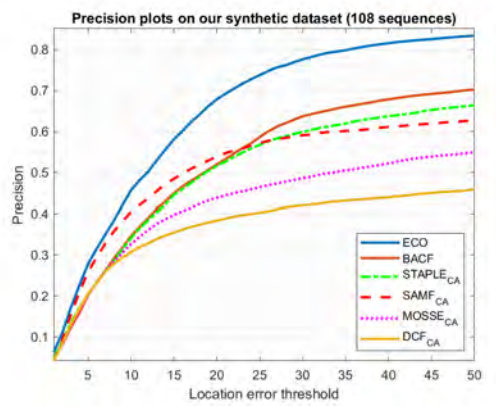


Figure 9: Precision plot of the evaluated trackers on our dataset.

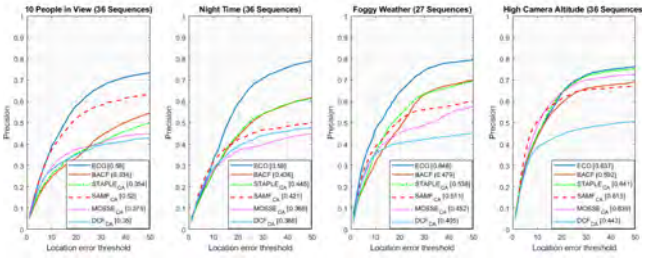


Figure 10: Precision plots for the four challenging cases. Crowded scenes, night time, foggy weather and high camera altitude all cause a clear performance degradation.

the corresponding precision plots for these challenging attributes are shown. Please refer to the supplementary material for an extended presentation and discussion of the results.

4.4. Using Synthetic Data to Train Visual Trackers

Data Generation and Collection. For our second set of experiments, we employed NOVA to generate a set of synthetic sequences that can be used to train deep learning based trackers. Here, we consider different training scenarios including synthetic and real sequences, and also a hybrid of those. In contrast to the former part, we carry our analysis on real test sequences for this set of experiments. In particular, NOVA is used generate 97 synthetic sequences and their ground truths annotations with pixel-level accuracy. However, to match the characteristics of the available real datasets, we limit the weather attribute to normal weather conditions, namely, clear-sky and three different variations of cloudy weather conditions. At the same time, we vary all other procedural generation parameters such as time of day, camera type, scene crowdedness and environment. In creating this set, it was aimed to mimic the general pattern of the existing real-world datasets, maintaining both the photorealism and the diversity at compatible levels.

In addition to the created synthetic dataset, we collect 125 real-world sequences from OTB100 [WLY15], VOT [KML*16; KML*19], TC128 [LBL15], UAV123 [MSG16], NUS-PRO [LLW*16] and MOT [MLR*16] datasets. We especially pick the sequences containing humans in outdoor environments and under normal weather conditions. Finally, we randomly divide these 125 real sequences into training and testing parts, where 97 sequences were selected for training and 28 for testing.

Please refer to the supplementary material for some sample frames from the synthetic and real-world sequences used. The synthetic sequences along with a file containing the links to the real-world sequences are provided at our project website under the name HybridPTB (Hybrid Person Tracking Benchmark).

Visual Trackers. We employ two state-of-the-art deep trackers in our experiments, namely CFNet [VBH*17] and DiMP [BDGT19]. Correlation filter based tracking (CFNet) is a deterministic, end-to-end representation learning tracker which considers correlation filter (CF) as a differentiable layer in a CNN architecture. This allows the error gradients to pass through the CF layer and tune the CNN features. DiMP, on the other hand, is a deep-learning based tracker that depends on Siamese architecture which accounts for the target and the background information while predicting the target object's location. The parameters of the tracker is learned in an end-to-end manner using a discriminative loss function.

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

Training from Scratch. In the first scheme, we train each tracker from scratch by randomly initializing the model parameters using a different training set in each training scenario. The first scenario involves training the trackers using only the synthetic sequences generated by NOVA (E1). For the second one, the trackers are trained by employing only the real sequences from the training split of the dataset we collected (E2). Finally, in the last scenario, we consider a hybrid approach and explore the advantages of expanding the set of real sequences with the synthetic ones and training the trackers using this combined set (E3).

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

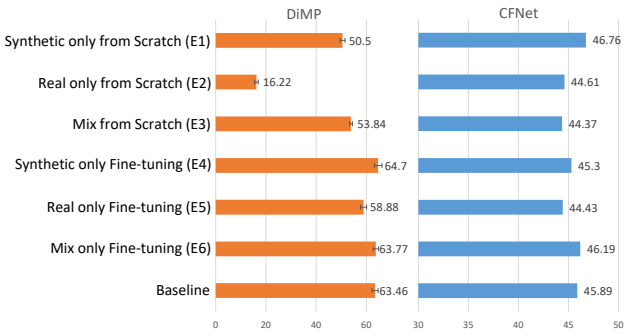


Figure 11: EAO scores obtained with the six different training scenarios as compared to those of the baselines. Error bars on the DiMP results give the standard deviation of the EAO score. Fine-tuning the baselines on a mixture of synthetic and real sequences improves the performance. At the same time,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

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 improves their performances to a greater extent 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. It is worth noting that, these results are also taken to indicate that the domain gap due to the differences between the synthetic and the real-world sequences seems to be minimal. Although the trackers were trained on NOVA's synthetic sequences and testing was carried out on real-world sequences, *i.e.*, our training and test sequences do not share the same level of photorealism, it is seen that using synthetic person sequences during training let the trackers learn more fine-grained features for person tracking, and, in return, leads to better performances.

5. Discussion

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 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.

5.1. Limitations and Future Work

Investigating the usability of synthetic data generated by the NOVA rendering engine is an important aspect of this study. Here, we demonstrated that using synthetic data generated by NOVA can both boost the performance of the state-of-the-art trackers and provide a better medium for testing tracking algorithms under a number of challenging attributes. However, one concern is the generalizability of these findings to other computer vision tasks such as semantic segmentation, depth estimation and so on. Considering the procedural generation capabilities of NOVA, including rich variety of annotations it can produce, there are various other directions that can be explored to thoroughly address the matter. Accordingly, it is our plan to extend this study toward exploring other computer vision tasks in future works.

The lack of performance of the trackers on NOVA's synthetic test sequences could be partially attributed to the domain gap problem, as the trackers were trained on real-world data. However, the photorealism of the generated sequences is expected to have mitigated this gap. In parallel to that, the improvement in performance of the deep trackers on tests with real-world data upon having been trained on the synthetic data sheds light on the cohesion of the synthetic data with real-world data. In addition, the fact that not just the deep trackers but also the correlation-filter -based trackers, which rely solely on online learning, showed poor performance on NOVA's synthetic test sequences further signifies that the main factor at play is the challenging nature of these sequences as the domain gap is not thought to cause such a clear degradation in performance across the board.

As a future work, we plan to increase the procedural generation capabilities of NOVA, especially regarding the 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. Additionally, we are planning to implement other camera types such as body-worn cameras and third-person-view cameras along with camera artifacts such as motion blur and chromatic aberration to simulate a wider range of real-world video captures. Moreover, using NOVA, we are planning to generate a special benchmark for evaluating the performance of general purpose trackers under adverse weather conditions.

6. 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 allows to 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. Our results show that the scenes that are either highly crowded, or taking place at night or at foggy weather conditions pose certain challenges for the state-of-the-art trackers. It is also seen that using synthetic data generated by NOVA for training can boost the performance of learning-based trackers on real videos.

An online demo of NOVA and videos illustrating NOVA's capabilities are available at the project website https://graphics.cs.hacettepe.edu.tr/NOVA along with VirtualPTB1 and HybridPTB, featuring the synthetic sequences generated by NOVA for the first and the second set of experiments, respectively.

Acknowledgements

We thank the anonymous reviewers for their constructive comments that helped improve this work. This work was supported in part by TUBITAK-1001 Program (Grant No. 217E029), GEBIP 2018 fellowship of Turkish Academy of Sciences awarded to E. Erdem, and BAGEP 2021 Award of the Science Academy awarded to A. Erdem.

References

[BBDL10] BOLME, DAVID S., BEVERIDGE, J. ROSS, DRAPER, BRUCE A., and LUI, YUI MAN. "Visual object tracking using adaptive correlation filters". *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010 8.

[BDGT19] BHAT, GOUTAM, DANELLJAN, MARTIN, GOOL, LUC VAN, and TIMOFTE, RADU. "Learning discriminative model prediction for tracking". *Proceedings of the IEEE International Conference on Computer Vision*. 2019, 6182–6191 10.

[BSL*11] BAKER, SIMON, SCHARSTEIN, DANIEL, LEWIS, JP, et al. "A database and evaluation methodology for optical flow". *International Journal of Computer Vision* 92.1 (2011), 1–31 3.

[BVG*16] BERTINETTO, LUCA, VALMADRE, JACK, GOLODETZ, STUART, et al. "Staple: Complementary learners for real-time tracking". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 1401–1409 8.

[BWSB12] BUTLER, DANIEL J, WULFF, JONAS, STANLEY, GARRETT B, and BLACK, MICHAEL J. "A naturalistic open source movie for optical flow evaluation". *European Conference on Computer Vision*. Springer. 2012, 611–625 3.

[CCC*15] CAO, LIJUN, CHEN, WEIHUA, CHEN, XIAOTANG, et al. "An equalised global graphical model-based approach for multi-camera object tracking". *arXiv preprint arXiv:1502.03532* (2015) 8.

[CWB*16] CHEUNG, ERNEST, WONG, TSAN KWONG, BERA, ANIKET, et al. "Lcrowdv: Generating labeled videos for simulation-based crowd behavior learning". *European Conference on Computer Vision*. Springer. 2016, 709–727 3.

[DBSF17] DANELLJAN, MARTIN, BHAT, GOUTAM, SHAHBAZ KHAN, FAHAD, and FELSBERG, MICHAEL. "Eco: Efficient convolution operators for tracking". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, 6638–6646 8.

[DDS*09] DENG, J., DONG, W., SOCHER, R., et al. "ImageNet: A Large-Scale Hierarchical Image Database". *CVPR09*. 2009 2.

[Deb02] DEBEVEC, PAUL. "Image-based lighting". *IEEE Computer Graphics and Applications* 22.2 (2002), 26–34 5.

[DGCP17] DE SOUZA, CÉSAR ROBERTO, GAIDON, ADRIEN, CABON, YOHANN, and PEÑA, ANTONIO MANUEL LÓPEZ. "Procedural Generation of Videos to Train Deep Action Recognition Networks." *CVPR*. 2017, 2594–2604 2, 3.

[FLY*19] FAN, HENG, LIN, LITING, YANG, FAN, et al. "Lasot: A high-quality benchmark for large-scale single object tracking". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, 5374–5383 11.

[GLU12] GEIGER, ANDREAS, LENZ, PHILIP, and URTASUN, RAQUEL. "Are we ready for autonomous driving? the kitti vision benchmark suite". *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, 3354–3361 3.

[GWCV16] GAIDON, ADRIEN, WANG, QIAO, CABON, YOHANN, and VIG, ELEONORA. "Virtual worlds as proxy for multi-object tracking analysis". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 4340–4349 3.

[GZW*] GE, YUYING, ZHANG, RUIMAO, WU, LINGYUN, et al. "A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-Identification of Clothing Images". () 2.

[HCMB15] HENRIQUES, JOÃO F, CASEIRO, RUI, MARTINS, PEDRO, and BATISTA, JORGE. "High-speed tracking with kernelized correlation filters". *IEEE transactions on pattern analysis and machine intelligence* 37.3 (2015), 583–596 8.

[HUI13] HALTAKOV, VLADIMIR, UNGER, CHRISTIAN, and ILIC, SLOBODAN. "Framework for generation of synthetic ground truth data for driver assistance applications". *German Conference on Pattern Recognition*. Springer. 2013, 323–332 3.

[HZH19] HUANG, LIANGHUA, ZHAO, XIN, and HUANG, KAIQI. "Got-10k: A large high-diversity benchmark for generic object tracking in the wild". *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019) 11.

[KFL17] KIANI GALOOGAHI, HAMED, FAGG, ASHTON, and LUCEY, SIMON. "Learning background-aware correlation filters for visual tracking". *Proceedings of the IEEE International Conference on Computer Vision*. 2017, 1135–1143 8.

[KH09] KRIZHEVSKY, ALEX and HINTON, GEOFFREY. *Learning multiple layers of features from tiny images*. Tech. rep. Citeseer, 2009 2.

[KML*16] KRISTAN, MATEJ, MATAS, JIRI, LEONARDIS, ALEŠ, et al. "A Novel Performance Evaluation Methodology for Single-Target Trackers". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.11 (Nov. 2016), 2137–2155. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2016.2516982 8, 10.

[KML*19] KRISTAN, MATEJ, MATAS, JIRI, LEONARDIS, ALES, et al. "The seventh visual object tracking vot2019 challenge results". *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2019, 0–0 8, 10.

[LBL15] LIANG, PENGPENG, BLASCH, ERIK, and LING, HAIBIN. "Encoding color information for visual tracking: Algorithms and benchmark". *IEEE Transactions on Image Processing* 24.12 (2015), 5630–5644 8, 10.

[LLW*16] LI, A, LIN, M, WU, Y, et al. "NUS-PRO: A New Visual Tracking Challenge". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.2 (2016), 335–349 8, 10.

[LMB*14] LIN, TSUNG-YI, MAIRE, MICHAEL, BELONGIE, SERGE, et al. "Microsoft coco: Common objects in context". *European conference on computer vision*. Springer. 2014, 740–755 2, 7, 11.

[LWT*18] LI, XUAN, WANG, KUNFENG, TIAN, YONGLIN, et al. "The ParallelEye Dataset: A Large Collection of Virtual Images for Traffic Vision Research". *IEEE Transactions on Intelligent Transportation Systems* 99 (2018), 1–13 3.

[LZ14] LI, YANG and ZHU, JIANKE. "A scale adaptive kernel correlation filter tracker with feature integration". *European conference on computer vision*. Springer. 2014, 254–265 8.

[MBG*18] MULLER, MATTHIAS, BIBI, ADEL, GIANCOLA, SILVIO, et al. "Trackingnet: A large-scale dataset and benchmark for object tracking in the wild". *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, 300–317 11.

[MLR*16] MILAN, ANTON, LEAL-TAIXÉ, LAURA, REID, IAN, et al. "MOT16: A benchmark for multi-object tracking". *arXiv preprint arXiv:1603.00831* (2016) 8, 10.

[MSG16] MUELLER, MATTHIAS, SMITH, NEIL, and GHANEM, BERNARD. "A benchmark and simulator for uav tracking". *European conference on computer vision*. Springer. 2016, 445–461 8, 10.

[MSG17] MUELLER, MATTHIAS, SMITH, NEIL, and GHANEM, BERNARD. "Context-Aware Correlation Filter Tracking". *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017 8.

[QY16] QIU, WEICHAO and YUILLE, ALAN. "Unrealcv: Connecting computer vision to unreal engine". *European Conference on Computer Vision*. Springer. 2016, 909–916 3.

[RHK17] RICHTER, STEPHAN R, HAYDER, ZEESHAN, and KOLTUN, VLADLEN. "Playing for benchmarks". *Proceedings of the IEEE International Conference on Computer Vision*. 2017, 2213–2222 2, 3.

[RSM*16] ROS, GERMAN, SELLART, LAURA, MATERZYNSKA, JOANNA, et al. "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 3234–3243 3.

[RSZ*16] RISTANI, ERGYS, SOLERA, FRANCESCO, ZOU, ROGER, et al. "Performance measures and a data set for multi-target, multi-camera tracking". *European Conference on Computer Vision*. Springer. 2016, 17–35 8.

[RVRK16] RICHTER, STEPHAN R, VINEET, VIBHAV, ROTH, STEFAN, and KOLTUN, VLADLEN. "Playing for data: Ground truth from computer games". *European Conference on Computer Vision*. Springer. 2016, 102–118 2, 3.

[SLS16] SHAFAEI, ALIREZA, LITTLE, JAMES J, and SCHMIDT, MARK. "Play and learn: Using video games to train computer vision models". *arXiv preprint arXiv:1608.01745* (2016) 2, 3.

[Sys] SYSTEM, UNITY MULTIPURPOSE AVATAR. *UMA git repo*. https://github.com/umasteeringgroup/UMA. Online; accessed: 2019-02-20 4.

[TCB07] TAYLOR, GEOFFREY R, CHOSAK, ANDREW J, and BREWER, PAUL C. "Ovvv: Using virtual worlds to design and evaluate surveillance systems". *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE. 2007, 1–8 2, 3.

[Uni] UNITY. *Rendering with Replaced Shaders*. https://docs.unity3d.com/Manual/SL-ShaderReplacement.html. Online; accessed: 2019-02-20 6.

[VBH*17] VALMADRE, JACK, BERTINETTO, LUCA, HENRIQUES, JOAO, et al. "End-to-end representation learning for correlation filter based tracking". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, 2805–2813 10.

[WLY15] WU, Y., LIM, J., and YANG, M. "Object Tracking Benchmark". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.9 (Sept. 2015), 1834–1848. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2014.2388226 8, 10.

[Wor] WORLDS, PROCEDURAL. *Enviro webpage*. Online; accessed: 2019-02-20. URL: http://www.procedural-worlds.com/gaia/gaia-extensions/enviro/ 5.

[WU18] WRENNINGE, MAGNUS and UNGER, JONAS. "Synscapes: A photorealistic synthetic dataset for street scene parsing". *arXiv preprint arXiv:1810.08705* (2018) 3.

[Zaa] ZAAL, G. *HDRI Haven*. https://hdrihaven.com/hdris. Online; accessed: 2019-02-20 6.

[ZSFR15] ZHANG, SHU, STAUDT, ELLIOT, FALTEMIER, TIM, and ROY-CHOWDHURY, AMIT K. "A camera network tracking (CamNeT) dataset and performance baseline". *2015 IEEE Winter Conference on Applications of Computer Vision*. IEEE. 2015, 365–372 8.