UnrealCV: Connecting Computer Vision to Unreal Engine

Weichao Qiu, Alan Yuille
(qiuwch, alan.l.yuille)@gmail.com

Johns Hopkins University

Introduction

UnrealCV is a project[1] to help computer vision researchers build virtual worlds using Unreal Engine 4 (UE4). It extends UE4 with a plugin by providing:

- A set of UnrealCV commands to interact with the virtual world.
- Communication between UE4 and an external program, such as Caffe.

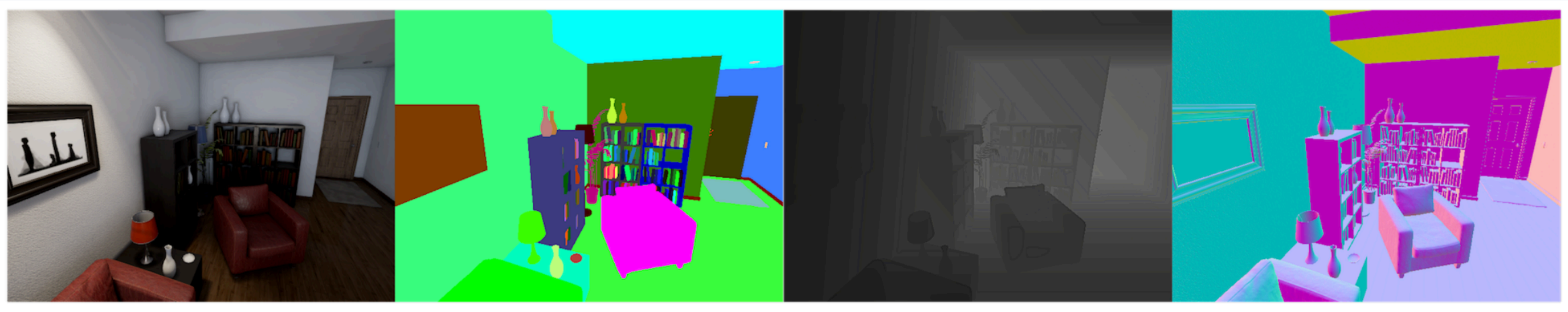


Fig. 1. A synthetic image and its ground truth generated using UnrealCV. The virtual room is from technical demo RealisticRendering, built by Epic Games. From left to right are the synthetic image, object instance mask, depth, surface normal

Motivation

Researchers have created realistic virtual worlds and have demonstrated that synthetic data is very helpful for many applications [1-5]. A virtual world can produce images with ground truth and in which: (i) an agent can perceive, navigate, and take actions, (ii) properties of the worlds can be modified and (iii) physical simulations can be performed. But creating realistic virtual worlds is not easy.

However, the game industry has spent a lot of effort on creating 3D worlds, which a player can interact with. Researchers can use these resources (examples are shown in Fig.2) to create virtual worlds, provided we can access and modify the internal data structures of the video game.

We present UnrealCV, a plugin of UE4, which enables researchers to access and modify the data structures of games, if we have the access to the game project. The goal of UnrealCV is to use the power of UE4 to make virtual worlds easier to **create, use and share.**

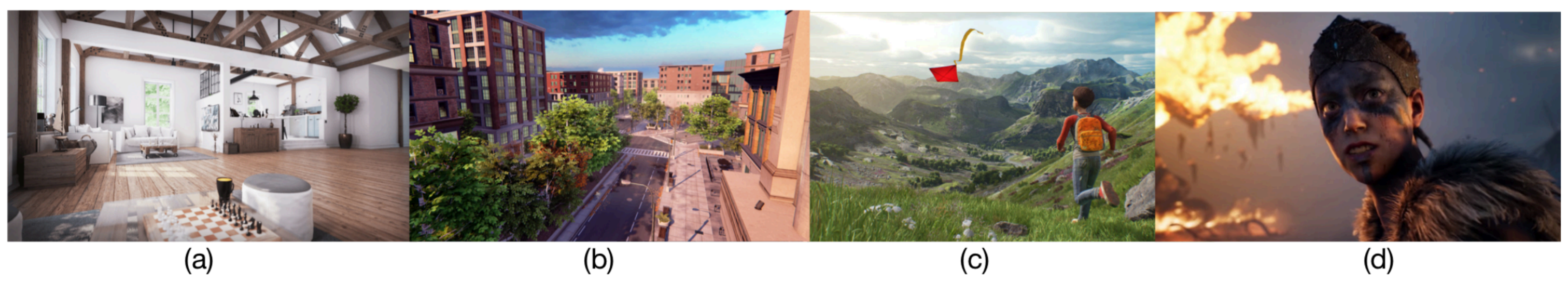


Fig. 2. Images produced by UE4, (a)(b) An architectural visualization and an urban city scene from Unreal Engine marketplace. (c) An open-source outdoor scene KiteRunner. (d) A digital human from the game Hellblade, shown in the conference GDC2016

Features

- Extensibility

Functions of UnrealCV are provided by a set of commands. The command system is designed in a modular way and can be easily extended. Details can be found in the architecture section.

- Ease of use

UnrealCV can be used in two ways. The first is using a compiled game binary with UnrealCV embedded[2]. This is as simple as running a game and no knowledge of Unreal Engine is required. The second is installing UnrealCV plugin to Unreal Engine 4 (UE4) and use the editor of UE4 to build a new virtual world. Tutorials are provided to get started.

- Rich resources

The plugin is compatible with recent versions of UE4 and no need to modify the engine code. Many UE4 projects and resources can be found online[3].

Architecture

UnrealCV consists of two parts. 1. UnrealCV server is a plugin embedded into the UE4 Editor or into a compiled game binary. 2. UnrealCV client is a library written in Python and MATLAB and can be integrated with programs such as Caffe. Client and server communicate with a plain text protocol using socket. The functions are provided by a set of commands. The relationship of components is shown in Fig.3.

- **UnrealCV commands**

We summarize what researchers need for applications, e.g. dataset generation, training with reinforcement learning, and design a set of extensible commands. The first part of a command is an action which can be either `vset` or `vget`. The second part is an URI representing the resource in a virtual world.

Camera control

```
vset /camera/[id]/location [x] [y] [z]
vset /camera/[id]/rotation [pitch] [yaw] [roll]
```

Ground truth generation

```
vget /camera/[id]/depth
vget /camera/[id]/object_mask
```

Control object

```
vset /object/[obj_name]/location [x] [y] [z]
```

Interaction

```
vset /action/presskey [keyname]
```

A complete list of commands is provided in project page[1].

1. https://unrealcv.github.io
2. https://unrealcv.github.io/reference/model_zoo.html
3. https://github.com/unrealcv/unrealcv/wiki/how-to-get-UE4-resource
4. https://github.com/unrealcv/unrealcv/tree/master/client/examples
5. https://github.com/qiuwch/synthetic-computer-vision

Scan QR code or visit:
bit.ly/unrealcv_links
to get these links

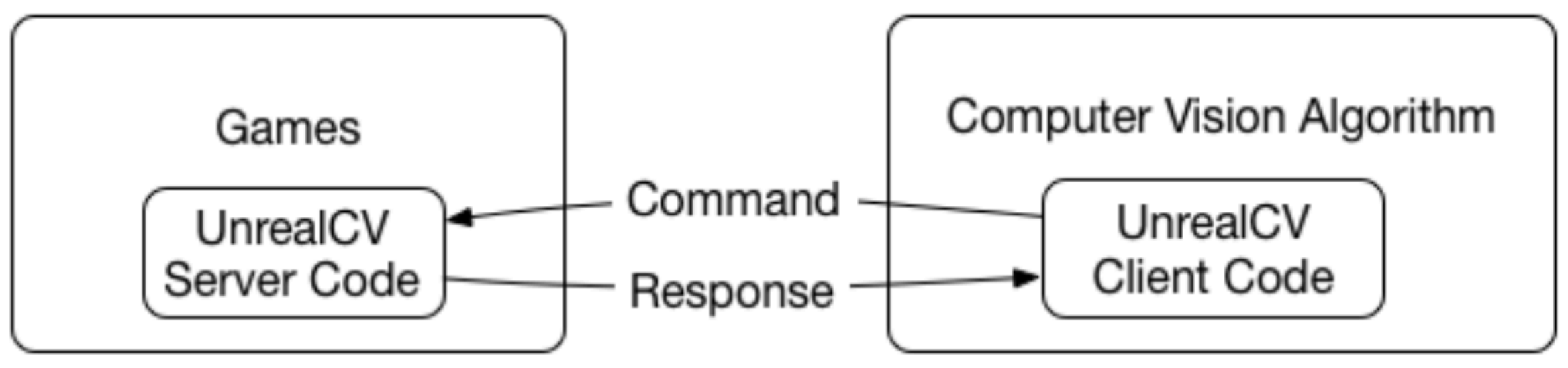


Fig. 3. The UnrealCV server is an UE4 plugin embedded into a game during compilation. An external program uses the UnrealCV client to communicate with the game.

Application

We use two applications to demonstrate the usefulness of UnrealCV

- **Generating a synthetic image dataset with detailed annotations**

Use a virtual camera to take photos, get depth and object masks with 10 lines of python code[4]

```python
import json; camera_trajectory = json.load(open('camera_traj.json'))
from unrealcv import client
client.connect()
# Get object information
obj_info = client.request('vget /objects')
for [loc, rot] in camera_trajectory:
    # Set position of the first camera
    client.request('vset /camera/0/location {x} {y} {z}'.format(**loc))
    client.request('vset /camera/0/rotation {pitch} {yaw} {roll}'.format(**rot))
    # Get image and ground truth
    modes = ['lit', 'depth', 'object_mask']
    [im, dep, obj] = [client.request('vget /camera/0/%s' % m) for m in modes]
    print ['%s is saved to %s' % (k, v) for (k,v) in zip(modes, [im, dep, obj])]
```

- **Diagnosing a deep network algorithm**

We use Faster-RCNN model trained on PASCAL and test it in the virtual world by varying rendering configurations, e.g. viewpoint.

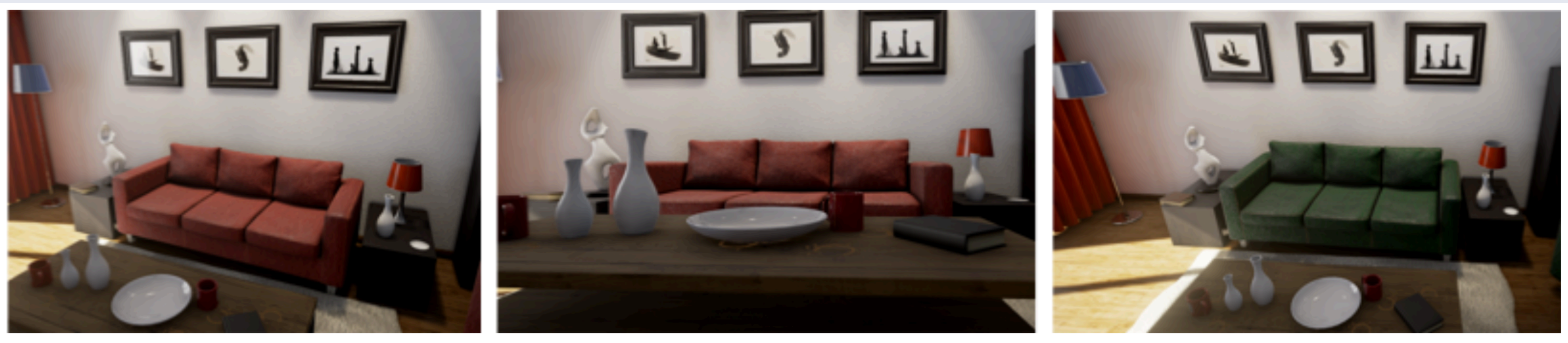


Fig. 4. Images with different camera height and different sofa color.

The Faster-RCNN code tries to detect the sofa from different views. For each az/el combination, the distance from the camera to the sofa was varied from 200cm to 290 cm. The results are in Tab.1.

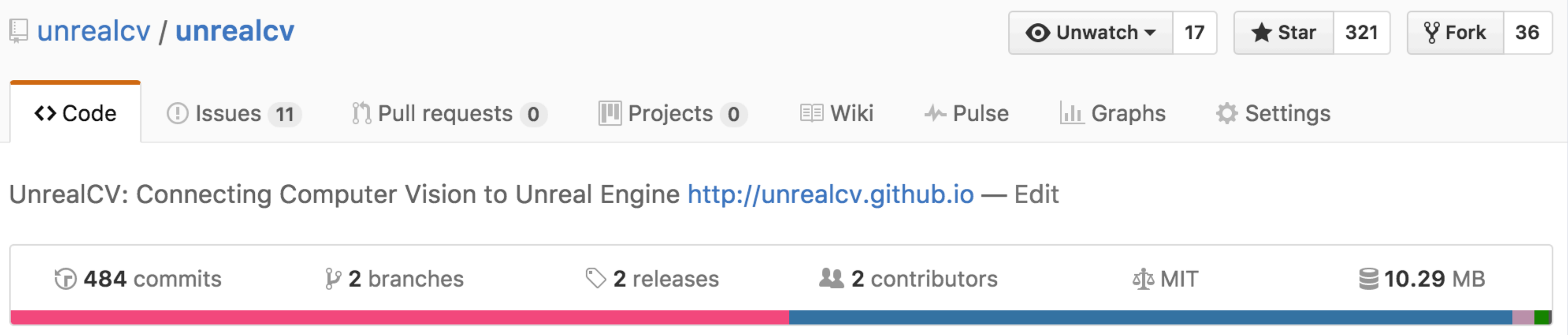
Azimuth Elevation	90	135	180	225	270
0	-	0.713	0.769	0.930	0.319
30	0.900	1.000	0.588	1.000	0.710
60	0.255	0.100	0.148	0.296	0.649

Table 1. The Average Precision (AP) when viewing the sofa from different viewpoints. Observe the AP varies from 0.1 to 1.0 showing the sensitivity to viewpoint. This is perhaps because the biases in the training cause Faster-RCNN to favor specific viewpoints.

Conclusion

UnrealCV provides an Unreal Engine plugin to help researchers create virtual worlds. The project is open-source and hosted in Github, where it got more than 300 stars in the first month since release. We are working on: providing more features, integrating with OpenAI gym and improving stability.

We hope UnrealCV can be useful for your research. We encourage people to modify and extend its functions. If you are interested in using UnrealCV, please send us an email or join online discussion.



UnrealCV: Connecting Computer Vision to Unreal Engine http://unrealcv.github.io — Edit

484 commits · 2 branches · 2 releases · 2 contributors · MIT · 10.29 MB

Reference

[1]. Su, Hao, et al. "Render for CNN: Viewpoint estimation in images using cnns trained with rendered 3d model views." ICCV 2015

[2]. Gaidon, Adrien, et al. "Virtual Worlds as Proxy for Multi-Object Tracking Analysis." CVPR 2016

[3]. Richter, Stephan R., et al. "Playing for data: Ground truth from computer games." ECCV 2016

[4]. Ros, German, et al. "The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes." CVPR 2016

[5]. Butler, Daniel J., et al. "A naturalistic open source movie for optical flow evaluation." ECCV 2012

The more complete and update-to-date related works can be found in our website[5].

Acknowledgement

We would like to thank Yi Zhang, Austin Reiter, Vittal Premachandran, Lingxi Xie and Siyuan Qiao for discussion and feedback. This project is supported by the Intelligence Advanced Research Projects Activity (IARPA) with contract D16PC00007.