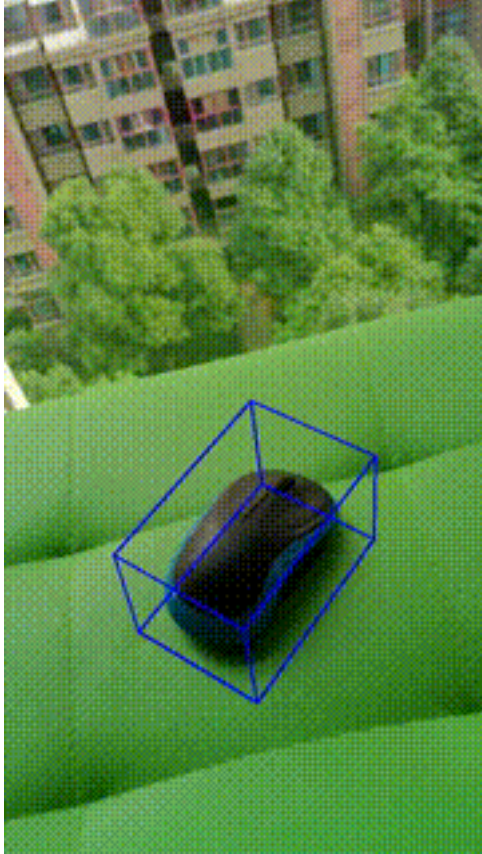

Gen6D

Gen6D is able to estimate 6DoF poses for unseen objects like the following video.



Project page | Paper

Todo List

- ☒ Pretrained models and evaluation codes.
- ☒ Pose estimation on custom objects.
- ☒ Training codes.

Installation

Required packages are list in [requirements.txt](#). To determine how to install PyTorch along with CUDA, please refer to the [pytorch-documentation](#)

Download

1. Download pretrained models, GenMOP dataset and processed LINEMOD dataset at [here](#).
2. Organize files like

```
1 Gen6D
2 |-- data
3     |-- model
4         |-- detector_pretrain
5             |-- model_best.pth
6         |-- selector_pretrain
7             |-- model_best.pth
8         |-- refiner_pretrain
9             |-- model_best.pth
10    |-- GenMOP
11        |-- chair
12        ...
13    |-- LINEMOD
14        |-- cat
15        ...
```

Evaluation

```
1 # Evaluate on the object TFormer from the GenMOP dataset
2 python eval.py --cfg configs/gen6d_pretrain.yaml --object_name genmop/
  tformer
3
4 # Evaluate on the object cat from the LINEMOD dataset
5 python eval.py --cfg configs/gen6d_pretrain.yaml --object_name linemod/
  cat
```

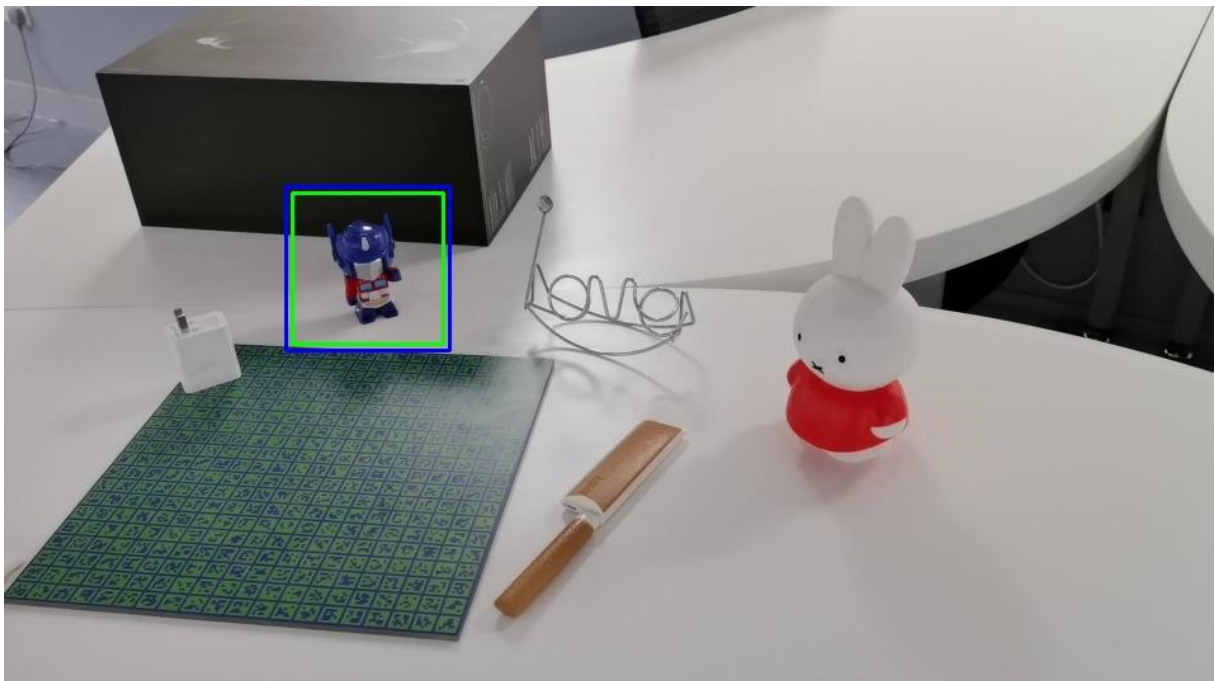
Metrics about ADD-0.1d and Prj-5 will be printed on the screen.

Qualitative results

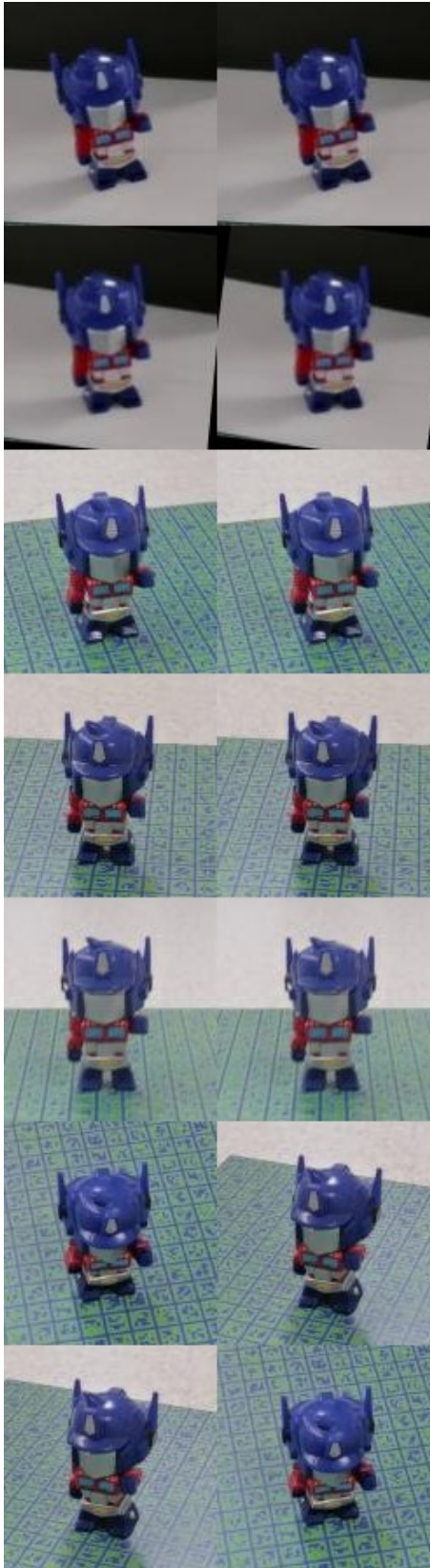
3D bounding boxes of estimated poses will be saved in `data/vis_final/gen6d_pretrain/genmop/tformer`. Ground-truth is drawn in green while prediction is drawn in blue.



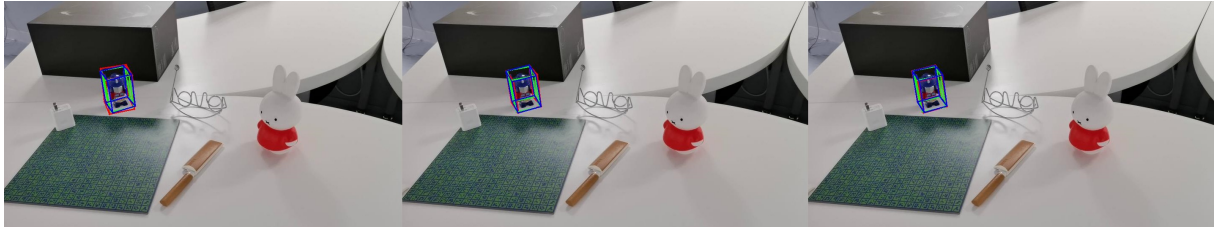
Intermediate results about detection, viewpoint selection and pose refinement will be saved in `data/vis_inter/gen6d_pretrain/genmop/tformer`.



This image shows detection results.



This image shows viewpoint selection results. The first row shows the input image to the selector. The second row shows the input images rotated by the estimated in-plane rotation (left column) or the ground-truth in-plane rotation(right column) Subsequent 5 rows show the predicted (left) or ground-truth (right) 5 reference images with nearest viewpoints to the input image.



This image shows the pose refinement process. The red bbox represents the input pose, the green one represents the ground-truth and the blue one represents the output pose for the current refinement step.

Pose estimation on custom objects

Please refer to custom_object.md

Training

1. Download processed co3d data (co3d.tar.gz), google scanned objects data (google_scanned_objects.tar.gz) and ShapeNet renderings (shapenet.tar.gz) at [here](#).
2. Download COCO 2017 training set.
3. Organize files like

```
1 Gen6D
2 |-- data
3     |-- GenMOP
4         |-- chair
5         ...
6     |-- LINEMOD
7         |-- cat
8         ...
9     |-- shapenet
10        |-- shapenet_cache
11        |-- shapenet_render
12        |-- shapenet_render_v1.pkl
13    |-- co3d_256_512
14        |-- apple
15        ...
16    |-- google_scanned_objects
17        |-- 06K3jXvzqIM
```

```
18         ...
19     |-- coco
20     |-- train2017
```

4. Train the detector

```
1 python train_model.py --cfg configs/detector/detector_train.yaml
```

5. Train the selector

```
1 python train_model.py --cfg configs/selector/selector_train.yaml
```

6. Prepare the validation data for training refiner

```
1 python prepare.py --action gen_val_set \
2     --estimator_cfg configs/gen6d_train.yaml \
3     --que_database linemod/cat \
4     --que_split linemod_val \
5     --ref_database linemod/cat \
6     --ref_split linemod_val
7
8 python prepare.py --action gen_val_set \
9     --estimator_cfg configs/gen6d_train.yaml \
10    --que_database genmop/tformer-test \
11    --que_split all \
12    --ref_database genmop/tformer-ref \
13    --ref_split all
```

This command will generate the information in the `data/val`, which will be used in producing validation data for the refiner. 7. Train the refiner

```
1 python train_model.py --cfg configs/refiner/refiner_train.yaml
```

8. Evaluate all components together.

```
1 # Evaluate on the object TFormer from the GenMOP dataset
2 python eval.py --cfg configs/gen6d_train.yaml --object_name genmop/
   tformer
3
4 # Evaluate on the object cat from the LINEMOD dataset
5 python eval.py --cfg configs/gen6d_train.yaml --object_name linemod/cat
```

How to make a GenMOP object for evaluation

The process of making the GenMOP dataset is described as follows: 1. run SfM on the reference sequence using COLMAP. 2. run SfM on the test sequence using COLMAP. **Note the test sequence**

for evaluation need be captured in a static scene. 3. Manually label at least 4 keypoints on two images from the query sequence and label the same 4 keypoints on two images from the test sequence. For example, we label 4 keypoints on the frame40.jpg and frame620.jpg from the reference sequence of the `TFormer` object, which is stored in `aling-data/tformer-anno/ref-frame40(620)`. And we label 4 keypoints on the frame130.jpg and frame540.jpg from the test sequence of the `TFormer` object, which is stored in `aling-data/tformer-anno/test-frane130(540)`. 4. Compute the alignment poses and scale for two sequences and save the results in `align.pkl` in `tformer-test/`. We provide an example in `compute_align_poses.py`. 5. With `align.pkl`, you may use the `GenMOPDatabase` by `parse_database_name('genmop/tformer-que')` and `parse_database_name('genmop/tformer-que')`.

We use the annotation tools from <https://github.com/luigivieira/Facial-Landmarks-Annotation-Tool> to label keypoints for the GenMOP dataset.

Acknowledgements

In this repository, we have used codes or datasets from the following repositories. We thank all the authors for sharing great codes or datasets.

- PVNet
- hloc
- COLMAP
- ShapeNet
- COCO
- Co3D
- Google Scanned Objects
- MVSNet_pl
- AnnotationTools

We provide a paper list about recent generalizable 6-DoF object pose estimators at <https://github.com/liuyuan-pal/Awsome-generalizable-6D-object-pose>.

Citation

```
1 @inproceedings{liu2022gen6d,  
2   title={Gen6D: Generalizable Model-Free 6-DoF Object Pose Estimation  
   from RGB Images},  
3   author={Liu, Yuan and Wen, Yilin and Peng, Sida and Lin, Cheng and  
   Long, Xiaoxiao and Komura, Taku and Wang, Wenping},  
4   booktitle={ECCV},
```

```
5  year={2022}  
6  }
```