3dcv report

r12922095, r12944032

NTU

Abstract. The majority of the motion sensing game requires additional peripherals like motion sensors to be bought, for example, the Xbox Kinect, or the Switch Joy-con. We plan to eliminate these requirements through the universality of modern RGB cameras, which we use as the controller of games. In this report, we estimate the pose of a object in the real world, via a generalizable 3D pose estimation model called Gen6D, and integrate it into the game we made, as the pose of the in-game object.

Keywords: 6DoF, Pose Estimation, 3D Object Detection

1 Introduction

3D pose estimation refers to the task of determining the six degree-of-freedom (6D) pose of an object in 3D space. It involves estimating the position and orientation of an object in a scene, and is a fundamental problem in computer vision and robotics. The problem is challenging since we rely only on RGB inputs, which have severe limitations under low-light conditions and suffer from motion blur. Also, in order to be integrated into games, the real-timeness should be taken into consideration, which limits the workload of the method we choose.

2 Pose Estimation

2.1 Survey of methods

In the beginning of this project, we decide to survey the currently available pose estimation methods. The first we look into is the EfficientPose released in 2020, which utilizes EfficientNet as its backbone. Since the goal is to integrate the pose estimation result into games, the matter of being able to compute the pose in real-time concerns us. The network is based on EfficientNet which is claimed to be accurate and computational efficient, and inherits the properties of the EfficientNet. Considering only accuracy and efficiency, this method should be suitable for our projects.

However, after some discussions, we found out that methods that require predefined object as input may not suit our needs well, since our original intention is to make motion-sensing accessible and easy to normal households. Using predefined object can be inconvenient and it's hard to ensure that the users of this system have the exact object defined in the dataset. Therefore, we turned to seek generalizable method that can make predictions on novel objects.

After searching through the list [1], we found out that the Gen6D[2], which is a conference paper of 2022 ECCV, suits our needs perfectly. The Gen6D authors present a generalizable model-free 6-DoF object pose estimator. Here, the term "model-free" means that this method doesn't require the high-quality object model, such as CAD, which is not conveniently accessible at all. The Gen6D estimator only requires some posed images of the unseen object and is able to accurately predict poses of the object in arbitrary environments. Also, the images required are normal RGB camera images, and depth informations are not needed.

2.2 Gen6D

Here's an overview of how the Gen6D works. As shown in Fig. 1, the pipeline consists of three components, the detector, the selector and the refiner. The detector first detects the region of interest from the query image. Then, the selector predicts a coarse prediction of the viewpoint of the query image. These two steps gives out a coarse prediction of the translation and the rotation of the input, respectively. Lastly, through the refiner, a more pose is generated.

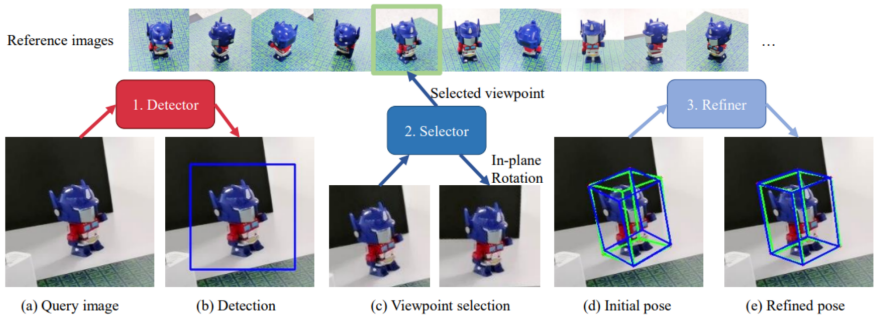


Fig. 1. Gen6D pipeline

Detector As shown in Fig. 2 (b), both the reference images and the query image are fed into in the feature extraction network to extract their feature maps. The query image resized into several predefined scales before being forwarded into the network. Then, the convoluted feature maps of the reference images act as convolution kernels and are convoluted with the feature maps of different scales. A score map of different scales is then generated. Based on the multi-scale score map, a heat map and a scale map are regressed. The location with the max value on the heat map is selected, and on the same location the corresponding scale s is selected. A bounding box of the object is generated via calculating its size

$S_q = S_r * s$, where S_r is the size of the reference images. With the predicted scale, we compute the coarse 3D translation and we crop the region of interest of the input image.

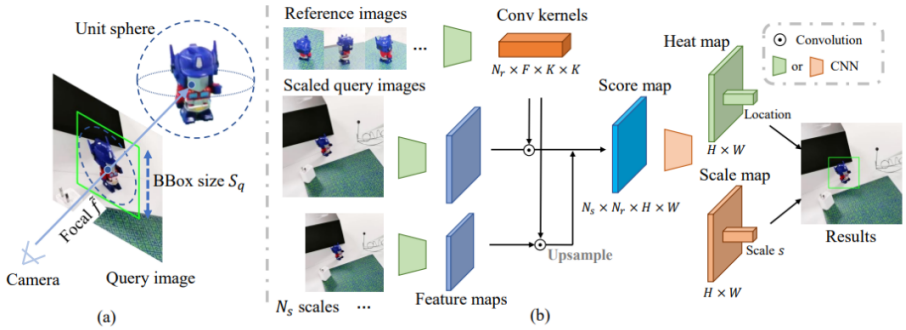


Fig. 2. The Detector

Selector As shown in Fig. 3, both the reference images and the previously cropped query image are fed into the feature extracting network. Before, being forwarded into the network, every reference image is rotated by N_a predefined angles, and every version of the rotated image is used in the element-wise product. Then, for every reference image, a correlation score map is produced via doing element-wise product with the query image. The correlation score map is latter produced by a similarity network, which consists of convolution layers, global normalization layers and attention layers, to produce similarity a score and a relative in-plane score for each referenced image. After this, the result with the highest similarity score is selected and the corresponding in-plane rotation is thus selected.

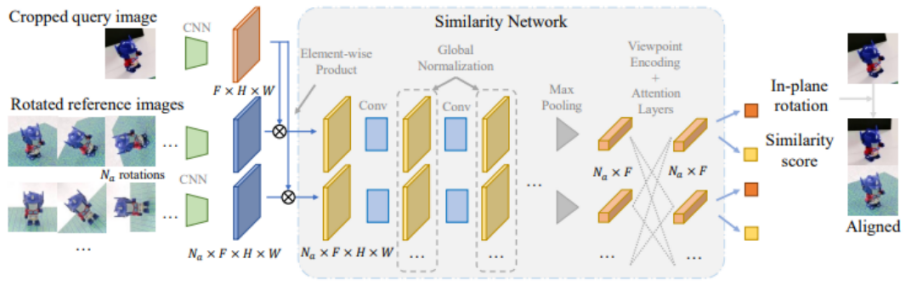
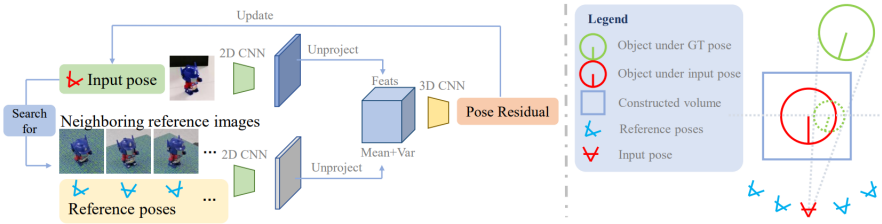


Fig. 3. The Selector

Refiner Combining the scale predicted by the detector and the rotation predicted by the selector, we can have a coarse pose of the query image. Some reference images that are near the input pose are selected. Feature maps are extracted from these reference images via a feature extraction network. Then, these feature maps are unprojected into a predefined 3D volume, and we compute the mean and the variance of the features among all reference images. The query image is also forwarded by the same feature extraction network, and is also unprojected into the 3D volume. The features are further concatenated with the previously calculated mean features and the variance features, and we forward the concatenated features with a 3D CNN to produce a pose residual. Finally, a final pose is calculated through a similarity transformation.



(a) Architecture of the pose refiner (b) Similarity approximation

Fig. 4. The Refiner

2.3 Custom Object

The Gen6D can be used on unknown object. Yet, the reference images and corresponding poses must be prepared beforehand. Here, we use the Pepsi cup as example (Fig. 5).

Fig. 5. The Pepsi cup

We first film a video around the cup. Then, we utilize colmap to generate the image-pose pairs for our cup via SFM(structure from motion). After this, we get

the pointcloud of the film scene besides the image-pose pairs. Like we shown in Fig. 6, we have to manually segment out the points of the cup from the whole pointcloud and determine the $+x$ and $+z$ direction of the cup points in order to determine the orientation and the size of the object.

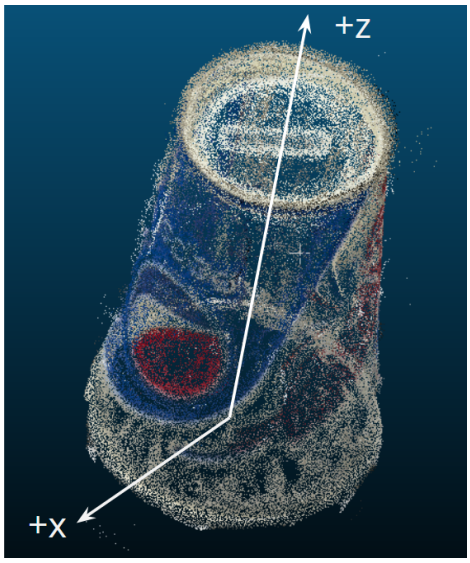


Fig. 6. Pointcloud of Pepsi cup

3 6DoF Game

3.1 Game

The 6DoF estimation is ready to go, now we just need to prepare a stage (Fig. 7) for it to shine. The objective of the game is to move the controller to the corresponding position as the notes fall.

We developed this game using the Panda3D engine due to its convenient simulation and display capabilities for objects in three-dimensional space. Panda3D aligns with our preference for Python as the expected game development language, which is consistent with the programming language used in our pose estimation. We believe that uniformity in programming languages enhances efficiency.

3.2 Integrating 6Dof Controller To Game

Using a cup as a controller, we capture the 6DoF information of the cup as parameters transmitted into the game. Objects within the game then rotate based on these parameters.

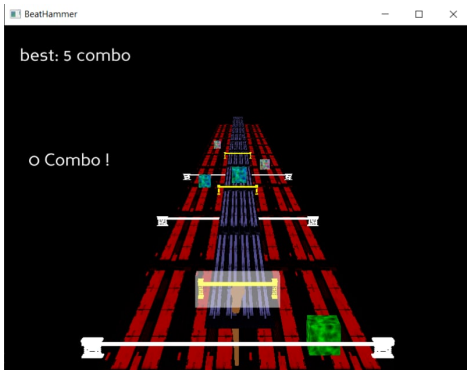


Fig. 7. Our designed rhythm game

However, camera's coordinate system is different from panda3D's coordinate system (shown in Fig. 8). To align the rotation of in-game objects intuitively with the real-world rotation of the cup, it is necessary to process the 6DoF information of the cup.

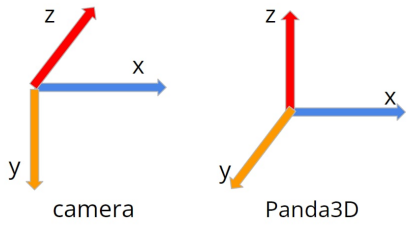


Fig. 8. Difference of coordinate system

4 Conclusion

4.1 Limitaion

The current method we employ can detect only one object at a time and lacks robustness; frequently, the predicted pose of the object does not align with the actual pose. Additionally, the computational speed of the 6DoF estimation method we adopted is considerably slower than the rate at which motion controllers in mainstream motion-sensing games provide information. Consequently, this method may not be suitable for highly complex real-time games.

4.2 Future Works

Exploring enhancements to achieve greater accuracy and faster speed is a potential avenue for improvement. Exploring methods capable of simultaneously detecting multiple objects is also our target. All the endeavor aims to expand the applicability of this technology to a broader range of games.

References

1. awesome-6d-object, https://github.com/ZhongqunZHANG/awesome-6d-object
2. Gen6D, https://github.com/liuyuan-pal/Gen6D