CS 639: Introduction to Computer Vision

Project Proposal: Object State Tracking in 3D Space by Using Two Webcams

Do-Men Su, Bo-Hsun Chen

1. Introduction

6D object pose estimation in real time, including location and orientation, is an important technique in many fields. Such helpful information can facilitate many attractive ongoing researches, such as robotic manipulation tasks and autonomous self-driving cars. However, most of the existing state-of-the-art papers for object pose estimation depend on a lot of labeled training data to conduct supervised learning. Besides, most of the current research utilized RGB-D or point-cloud cameras, which are expensive and hard to access for ordinary people. So in this project, we try to use a 3D deep auto-encoder (DAE) with spatial softmax [1] to get the 6D pose of the target object, which only requires few vision data without annotation in unsupervised learning. Furthermore, the algorithm will depend on the RGB-D images, which will be generated from a custom-made vision system composed of two inexpensive commercial 2D webcams.

2. Related Work

Some research papers proposed state-of-the-art algorithms to detect object 6D pose by using RGB-D data, which are listed in the website [2]. Xiang et al. [3] proposed PoseCNN based on convolutional neural network (CNN) structure. For each RGB image, PoseCNN firstly classifies all the image pixels into classes of objects. Secondly, for each pixel of an object, PoseCNN estimates the x and y directions which point to the center of this object in the 2D image, as well as predicting the depth from the camera to get the 3D translation. Lastly, 3D rotation data can be obtained by regressing the values in the quaternion of the object orientation. Besides, Wang et al [4] proposed a framework called DenseFusion to handle RGB-=D data. Firstly, DenseFusion respectively encodes the RGB image into color embedding through a CNN and the depth map into geometry embedding through the PointNet. And then, a proposed dense fusion network fuses the two types of embedding to extract pixel-wise feature embedding and estimate the pose of the object.

However, using an RGB-D camera like Kinect v2, which has been off-produced, is too expensive or hard for ordinary people to access. Besides, these algorithms mostly depend on large amounts of training data to train the models, and maybe these objects in the training data are not the same as the robot uses or the car recognizes in the tasks. So, these well-behaved results may be different for other usages. In this project, we proposed a cheaper way to build 3D visions and achieve tracking the pose of target objects in 3D space by using two webcams, based on a DAE with unsupervised learning. And we will experimentally verify the proposed method in the realistic world.

3. The Proposed Method

In our project, we will use two webcams to construct an RGB-D image, which is a cheaper way to construct 3D images and is the same way as the human eyes do. This process can be done with OpenCV [5]. In OpenCV, we need to calibrate the images first before extracting the depth map. Different cameras have different extents of distortion, so it is needed to extract the distortion coefficients for each camera model. Then, the calibrated images from the two cameras can be used to generate an RGB-D image. In this step, the algorithm needs to match the two images, which is called Stereo matching. OpenCV has a built-in algorithm for this step, so we can simply use StereoBM in OpenCV [5, 6]. Besides OpenCV's built-in algorithm, we may try implementing other algorithms [7, 8].

Secondly, these RGB-D vision data will be used in the unsupervised training based on the DAE to extract spatial feature points of the object and further estimate the pose of the object. In [1], the RGB image was sent into a DAE composed of three layers of convolutional neural networks. Creatively, it used a layer of softmax as activation functions in the end of the structure to encode the image into feature points with emphasis on object locations. The results showed that the embedded feature points in the image had a high relationship with the object location. And it trained the model by using only 5000 frames without annotation for each task. So, we want to extend this structure to a 3D version. Firstly, the RGB-D data will be sent into the DAE with a layer of 3D spatial softmax in the end to find the 3D feature points for images. Then, the estimated 3D feature points can be used to calculate and obtain the pose of a moving object by recognizing the change of associated feature points of the object.

4. Evaluation

We will evaluate the proposed algorithm by one of two possible ways. For the first way, we can collect the data by ourselves by using the custom-made dual-webcam system. We will place some objects with known poses under the web-cam system, or use the hand to catch-and-move objects along a desired trajectory to collect testing data. And then, we can compare the proposed method with other up-to-date algorithms by using the collected data. For the second way, we can directly use the RGB-D dataset called *YCB-Video* offered by [3], which contains precise 6D object poses of 21 objects in 92 videos with 133,827 frames, or use the *LineMOD* dataset which comprises 13 objects in 13 videos. We can directly utilize the 6D object pose estimation part and skip the dual-webcam system part in our method, and compare with other state-of-the-art approaches based on these well-known datasets.

5. Schedule

- Survey related papers and approaches (1 week)
- Set up dual webcams and calibration (1 week)
- Generate RGB-D images from two images (2 week)
- Extract feature points from 2D images (1 week)
- **Mid-term report (due November 3)**
- Build the DAE and extract spatial feature points from RGB-D images (1 week)
- Calculate the pose of the object from the spatial feature points (2 week)
- Gather Datasets / Evaluation and re-execute current approaches (1 week)
- **Final presentation (between December 1 through December 10)**

6. Reference

[1] C. Finn, Xin Yu Tan, Yan Duan, T. Darrell, S. Levine and P. Abbeel, "Deep spatial autoencoders for visuomotor learning," *IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, 2016, pp. 512-519, doi: 10.1109/ICRA.2016.7487173.

[2] Papers With Code, “6D Pose Estimation using RGBD,” accessed on: Sep. 29, 2020. [Online]. Available: https://paperswithcode.com/task/6d-pose-estimation-using-rgbd

[3] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, “PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes,” arXiv:1711.00199v3, 2018.

[4] C. Wang et al., "DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, CA, USA, 2019, pp. 3338-3347. doi: 10.1109/CVPR.2019.00346.

[5] OpenCV. Camera Calibration and 3D Reconstruction. Retrieved September 30, 2020 from https://docs.opencv.org/master/d9/db7/tutorial_py_table_of_contents_calib3d.html

[6] I. Culjak, D. Abram, T. Pribanic, H. Dzapo and M. Cifrek, "A brief introduction to OpenCV," *2012 Proceedings of the 35th International Convention MIPRO, Opatija, 2012*, pp. 1725-1730.

[7] C. Hernandez Esteban and F. Schmitt, "Multi-stereo 3D object reconstruction," Proceedings. First International Symposium on 3D Data Processing Visualization and Transmission, Padova, Italy, 2002, pp. 159-166, doi: 10.1109/TDPVT.2002.1024055.

[8] L. Zou and Y. Li, "A method of stereo vision matching based on OpenCV," *2010 International Conference on Audio, Language and Image Processing*, Shanghai, 2010, pp. 185-190, doi: 10.1109/ICALIP.2010.5684978.