

**VECTOR REPRESENTATION TO ENHANCE POSE
ESTIMATION FROM RGB IMAGES**

by
Zongcheng Chu

A Thesis

*Submitted to the Faculty of Purdue University
In Partial Fulfillment of the Requirements for the Degree of*

Master of Science



Department of Computer Graphics Technology

West Lafayette, Indiana

May 2020

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Yingjie Chen, Chair

Department of Computer and Graphics Technology

Dr. Vetrica Byrd,

Department of Computer and Graphics Technology

Dr. Baijian Yang,

Department of Computer and Information Technology

Approved by:

Prof. Nicoletta Adamo

Head of the Graduate Program

This study is wholeheartedly dedicated to my beloved parents, Wenrong Chu and Mingfang Shen, and my sister Zhenran Chu, who have been my source of inspiration and gave me strength when I thought of giving up, who continuously provide their moral, spiritual, emotional and financial support.

ACKNOWLEDGMENTS

I wish to gratefully acknowledge my thesis committee, Dr. Yingjie Chen, Dr. Vetrice Byrd and Dr. Baijian Yang, for their insightful comments and guidance. In addition, I would like to appreciate the help from Zhiwen Cao for running the experiments with me together.

TABLE OF CONTENTS

| | |
|--|------|
| LIST OF TABLES | vii |
| LIST OF FIGURES | viii |
| ABSTRACT | ix |
| CHAPTER 1. INTRODUCTION | 1 |
| 1.1 Background | 1 |
| 1.2 Significance | 4 |
| 1.3 Research Question | 6 |
| 1.4 Assumptions | 7 |
| 1.5 Limitations | 7 |
| 1.6 Delimitations | 7 |
| 1.7 Definitions | 8 |
| 1.8 Summary | 9 |
| CHAPTER 2. REVIEW OF RELEVANT LITERATURE | 10 |
| 2.1 Limitations For Rotation | 10 |
| 2.2 Classical Methods | 11 |
| 2.3 RGB-D Methods | 13 |
| 2.4 CNN-based Methods | 14 |
| 2.5 State-of-the-Art Approaches | 15 |
| 2.6 Applications of VIVE Trackers | 17 |
| 2.7 Summary | 18 |
| CHAPTER 3. FRAMEWORK AND METHODOLOGY | 19 |
| 3.1 Vector-Based Representation | 19 |
| 3.2 TriNet Architecture | 20 |
| 3.3 Multi-loss approach | 23 |
| 3.4 Vector Refinement | 25 |
| 3.5 Data Collection | 26 |
| 3.6 Summary | 30 |
| CHAPTER 4. EXPERIMENT AND RESULTS | 31 |

| | | |
|-------|--|----|
| 4.1 | Implementation Details | 31 |
| 4.2 | Datasets and Evaluation Metrics | 32 |
| 4.2.1 | VR dataset | 32 |
| 4.2.2 | 300W-LP | 32 |
| 4.2.3 | AFLW | 33 |
| 4.2.4 | BIWI | 33 |
| 4.2.5 | AFW | 34 |
| 4.3 | Comparison to State-of-the-art Methods | 34 |
| 4.4 | Architecture Evaluation | 36 |
| 4.5 | Ablation Study | 36 |
| 4.6 | Evaluation on VR data | 38 |
| | CHAPTER 5. SUMMARY | 41 |
| | REFERENCES | 43 |

LIST OF TABLES

| | | |
|-----|---|----|
| 4.1 | Mean average error of Euler angles using different methods on AFLW2000 dataset. | 35 |
| 4.2 | Mean average error of Euler angles using different methods on BIWI dataset. | 36 |
| 4.3 | MAE across different bin numbers and loss weights on AFLW2000 dataset using ResNet50 as basenet | 37 |
| 4.4 | Ablation study for different training methods. Results are evaluated on both AFLW2000 and BIWI datasets(Model parameters: ResNet50, 60 bins, $\alpha=1$, $\beta=0.07$). | 38 |
| 4.5 | Average angle errors on training and testing set. Model parameters: MobileNetV2, 60 bins, $\alpha = 1, \beta = 0.07$ | 39 |

LIST OF FIGURES

| | | |
|-----|---|----|
| 1.1 | Sample results of head pose estimation using proposed method. | 3 |
| 1.2 | Examples from 300W_LP dataset. | 5 |
| 3.1 | Detailed overview of our proposed TriNet. An input image goes through a backbone network followed by three subnet modules. Each subnet has identical structure and can generate one unit vector. A Post-processing step for vector refinement is proposed to achieve our final results. | 21 |
| 3.2 | Soft label encoding with different expand ratio. | 23 |
| 3.3 | Before distortion removal(left) and after(right). | 29 |
| 3.4 | Proposed pipeline for data collection with VR devices. | 30 |
| 4.1 | First row : ground truth of BIWI dataset, Second row: our results | 33 |
| 4.2 | Comparison with other methods on the AFW dataset | 35 |
| 4.3 | Distribution of Euler angle errors on AFLW2000 (in degrees) | 37 |
| 4.4 | Testing accuracy on VR dataset using MobileNetV2 as backbone. Right vector(left) and front vector(right). | 39 |
| 4.5 | Video clip(left) and visualization animation(right). | 40 |

ABSTRACT

Author: Chu, Zongcheng. M.S.

Institution: Purdue University

Degree Received: May 2020

Title: Vector representation to enhance pose estimation from RGB images

Major Professor: Yingjie Chen

Head pose estimation is an essential task to be solved in computer vision. Existing research for pose estimation based on RGB images mainly uses either Euler angles or quaternions to predict pose. Nevertheless, both Euler angle- and quaternion-based approaches encounter the problem of discontinuity when describing three-dimensional rotations. This issue makes learning visual pattern more difficult for the convolutional neural network(CNN) which, in turn, compromises the estimation performance. To solve this problem, we introduce TriNet, a novel method based on three vectors converted from three Euler angles(roll, pitch, yaw). The orthogonality of the three vectors enables us to implement a complementary multi-loss function, which effectively reduces the prediction error. Our method achieves state-of-the-art performance on the AFLW2000, AFW and BIWI datasets. We also extend our work to general object pose estimation and show results in the experiment part.

CHAPTER 1. INTRODUCTION

1.1 Background

Head pose estimation is an important task in computer vision, which has drawn a lot of research attention in recent years. A large amount of work has also been done related to face poses such as face alignment (Jourabloo & Liu, 2016), face landmark detection (Lv, Shao, Xing, Cheng, & Zhou, 2017; Sun, Wang, & Tang, 2013; Zhu & Ramanan, 2012), eye gaze estimation (Chong et al., 2018; X. Zhang, Sugano, Fritz, & Bulling, 2015) and 3D face modeling (Jackson, Bulat, Argyriou, & Tzimiropoulos, 2017; Jourabloo & Liu, 2016; Yu, Mora, & Odobez, 2017). However, most of the aforementioned studies need to use additional inputs aside from images to conduct the estimation. For instance, some works require depth information as supplementary (Fanelli, Gall, & Van Gool, 2011; Liu, Liang, Wang, Li, & Pei, 2016; Xiang, Schmidt, Narayanan, & Fox, 2017; Zakharov, Shugurov, & Ilic, 2019), which is usually obtained by depth sensors. Since the depth sensors are not always available, the applications of these methods are limited. Other studies (Gu, Yang, De Mello, & Kautz, 2017; R. Li, Danielsen, & Taskiran, 2008; Murphy-Chutorian & Trivedi, 2008) analyze human head movements from frame sequences by recurrent neural network (RNN)-based methods. The limitation of this type of approach is notable because it can only work under video domain.

Single image pose estimation so far mainly relies on facial landmark detection (Fanelli et al., 2011; Kumar, Alavi, & Chellappa, 2017; Murphy-Chutorian, Doshi, & Trivedi, 2007; Valle, Buenaposada, Valdés, & Baumela, 2019). These approaches show great robustness in dealing with scenarios where occlusion may occur by establishing a 2D-3D correspondence matching between images and 3D face models. However, they still have notable limitations when extracting key feature points from large poses such as profile figures. This limitation causes significant errors when predicting actual poses. To solve the issues, a large array of research has been directed to employ CNN-based

methods to predict head pose directly from a single RGB image. Several public benchmark datasets (Gourier, Maisonnasse, Hall, & Crowley, 2006; Martin Koestinger & Bischof, 2011; S. Yang, Luo, Loy, & Tang, 2016; Zhu, Lei, Liu, Shi, & Li, 2016) have been contributed in this area for the purpose of validating the effectiveness of these single image pose estimation methods. Among them, (Hsu, Wu, Wan, Wong, & Lee, 2018; Raytchev, Yoda, & Sakaue, 2004; Ruiz, Chong, & Rehg, 2018; T.-Y. Yang, Chen, Lin, & Chuang, 2019) try to address the problem by direct regression of either three Euler angles or quaternions from images using CNN models. They achieved results with impressive accuracy and lower the error down to a satisfactory level on these public datasets. However, These studies either use Euler angles or quaternions as their 3D rotation representations. Both Euler angles and quaternions have drawbacks when used to represent rotations. When using Euler angles, the rotation order must be decided in advance. Specifically, when two rotating axes become parallel, one degree of freedom will be lost. This causes the ambiguity problem known as gimbal lock (Fua & Lepetit, 2005). A quaternion ($\mathbf{q} \in \mathbb{R}^4, \|\mathbf{q}\|_2 = 1$) has the antipodal problem which results in \mathbf{q} and $-\mathbf{q}$ corresponding to the same rotation (Saxena, Driemeyer, & Ng, 2009). In addition, the results from (Y. Zhou, Barnes, Lu, Yang, & Li, 2019) show that any representation of rotation with four or fewer dimensions is discontinuous. This finding indicates that Euler angles and quaternions are not suitable for training a neural network to estimate pose.

We put forward a novel fine-grained vector-based head pose estimation method in this paper by training an end-to-end CNN model. Instead of using Euler angles or quaternions, we use three vectors to represent human head poses. As shown in Fig. 1.1, every head pose can be represented by a left vector (blue), a down vector (green) and a front vector (red). We convert Euler angles (roll, pitch, yaw) to a rotation matrix and get three vectors from this matrix. Our new vector-based network is designed for regressing three components for each of the three vectors.

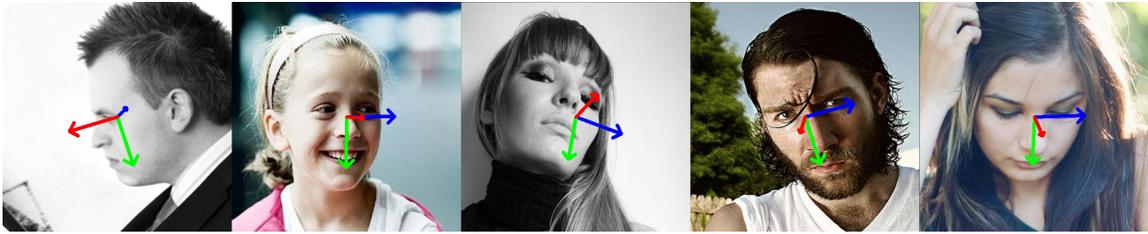


Figure 1.1. Sample results of head pose estimation using proposed method.

As mentioned by (Martin Koestinger & Bischof, 2011), AFLW dataset has coarse pose annotations so the ground-truth labels (roll, pitch and yaw) are not accurately annotated. To handle this problem, we replace the one-hot labeling with soft labels to incorporate constraints into both intraclass and interclass correlations. With this design, our method is robust to tackle the compromised data labelling caused by the inaccurate ground-truth annotations.

In order to have our network effectively learn the data, we formulate a new loss function to train the network. Combining with three regression losses measuring the angle errors between ground-truth vectors and our predicted ones, we refine the network structure by having multiple losses for training. Since the three output vectors from the neural network are still not mutually perpendicular, a post processing step following the network output is applied by solving an optimization function to get three perfectly orthonormal vectors.

We use extensive experiments to assess our method. Evaluations on three different public datasets are conducted. Different from Euler-angle-based methods which need to filter out large pose data samples (Ruiz et al., 2018) to avoid the inherent issues with Euler angle representation itself, our vector-based approach is robust for all kinds of poses and can use all the data for learning and testing. Our experiment results show state-of-the-art performance on AFLW2000, AFW and BIWI datasets.

In order to verify the generalization of our proposed method on other objects, we build our own dataset, which contains a 3D-printed bottle object, by using a set of VR devices.

The contributions of our work include:

1. We put forward a new vector-based method to represent rotations, which avoids the discontinuity problem of Euler angles and quaternions.
2. We propose a new fine-grained CNN model with multi-loss followed by refinements to predict the three vectors.
3. We achieve state-of-the-art performances on the AFLW2000, BIWI and AFW benchmark datasets.
4. We propose a new pose data collection scheme without manual labeling.

1.2 Significance

As discussed in section 1.1, many researchers [sruiz2018fine](#), [hsu2018quatnet](#) have uncovered the potential problem of using Euler angles and quaternions for head pose estimation, in that both of them are discrete in the real Euclidean Space, which leads to situations where two identical orientations may result in the same rotation. Such cases make the neural network training process extremely difficult. Because a single neural network can be considered as a many-to-one mapping, multiple inputs with similar patterns go through the hidden layers in network and give the same output. Typical supervised learning algorithms such as SVM, RandomForest and Neural Network try to learn the relationship between input and output, however, in cases where we adopt either Euler angles or quaternions, two similar input pose images show huge difference on their ground truth labels, which largely compromises such learning process. Fig. 1.2 shows several examples from [300W_LP](#) [sagonas2013300](#) dataset. As we can see, all the three listed images show a right-towards profile face. When checking with its original pose annotations, we notice that the labeling results are quite distinctive. Even the equivalent quaternion representation shows obvious difference. As we discussed in the background section about the reason why such situation may occur, these data samples further validate our previous claims about the potential problems existing in Euler angles and quaternions.

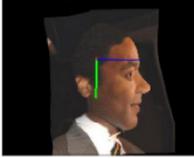
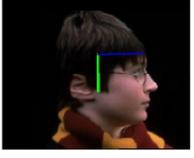
| Head Pose | Euler Angles(ground truth) | Quaternion |
|---|--|--|
|  | $[-80.8^\circ, -88.9^\circ, 78^\circ]$ | $[-0.695174, -0.123281, 0.694842, 0.136852]$ |
|  | $[-19.9^\circ, -87.4^\circ, 8.0^\circ]$ | $[-0.172084, -0.670119, 0.168758, 0.702031]$ |
|  | $[87.7^\circ, -89.3^\circ, -87.9^\circ]$ | $[0.706564, -0.022824, -0.706579, 0.031463]$ |

Figure 1.2. Examples from 300W_LP dataset.

In order to address this problem, we propose a new 9D vector-based rotation representation which can be a good alternative solution for Euler angles and quaternions and cause no ambiguity. More specifically, three orthogonal directional vectors are used for describing the 3D rotation. Our experiments show good performance on public datasets by using the proposed methods for head pose estimation task. We also achieve satisfactory results on our own dataset. The finding of this study reveals the limitation of widely used rotation representations and can also contribute to other pose related work such as robotic vision for grasping and augmented reality applications. In addition, a fine-grained convolutional neural network(CNN) structure combined with multi-loss, TriNet, is presented in this study to predict the three directional vectors from a single RGB image. TriNet outperforms other state-of-the-art methods on head pose estimation task and has brought a new idea for people thinking about rotation.

Another work involved in this study that has great implication is that a new data collection approach is developed relying on VR devices. Existing pose annotation methods can be classified as three main categories: 1. Manual labeling, 2. Key points matching between 2D images and 3D models. 3. Point cloud matching using depth camera such as Kinetic. In this study, we utilize VIVE goggle and VIVE trackers to accurately capture each video frame and the corresponding pose data(roll, pitch and yaw) to build our own dataset. A post processing step is then proposed to fix the image distortion problem due to some hardware issue with the fisheye camera on the goggle. The advantages of our proposed method can be concluded as following:

- VIVE trackers are accurate for computing the orientation(roll, pitch and yaw) in 3D world.
- Huge amount of image data can be generated by extracting each frame from recorded videos.
- A good combination with Unreal technology makes the data collection procedure more reliable and manageable.

In the methodology section, we will present a thorough discussion about the devices we use and the techniques involved.

1.3 Research Question

Does it exist any other representations that could be used for describing the 3D rotation without ambiguity issue and is more suitable for the neural network learning?

What is the performance of new rotation representation based machine learning algorithm?

Can our proposed method can be applied to other object pose estimation task other than human head pose? What is the performance?

1.4 Assumptions

The assumptions for this study include:

- Public datasets we are using have accurate ground truth labels.
- VR trackers can provide us with accurate pose information.

1.5 Limitations

The limitations for this study include:

- Public head pose datasets such as AFLWkoestinger11a have coarse pose annotations which make the neural network training process difficult.
- Public head pose datasets lack large poses, so that we can not verify our proposed method could be applied to poses in large degrees.
- We use a set of VR equipment for collecting our own object pose data, however because of the fisheye camera on VR goggles, images we obtained are distorted.
- Due to the limitation of hardware, We can only record videos for one minute to match the video stream with pose log data.
- We can not ensure the starting orientation of the tracker is always (0,0,0) since it relies on the camera calibration.
- Public datasets with pose annotations lack of elaboration on labeling details such as rotation orders and coordinate system used.

1.6 Delimitations

The delimitations for this study include:

- We only focus on head pose estimation and bottle pose estimation. We do not extend our work to other objects.
- Our data collection process relies on the commercial VR devices-HTC VIVE, and will not focus on other user input device. And we choose to collect the data in our lab environment as background.
- We only compare our results with those deep learning based start-of-the-art methods. We do not compare with key points based approaches or any other traditional methods for head pose estimation.

1.7 Definitions

In the broader context of thesis writing, we define the following terms:

Pose Estimation Determining the object's orientation relative to some coordinate systems

Convolutional Neural Network A computer vision technique for object detection, recognition and classification.

Euler Angle A orientation representation that consists of three angles that describe the rotation process with respect to a fixed 3-D coordinate.

Quaternions Another orientation representation that consists of four components. The first three defines a rotation axis and the last one determines the angle degrees.

Orthogonality In the context of this study, it refers to that two vectors are perpendicular to each other.

Rotation Matrix A matrix that is used to perform a rotation in Euclidean space.

Mean Square Error An estimator measures the average of the squares of the errors.

Kullback–Leibler Divergence A method for measuring the difference between two probability distribution. If we have two exactly same distribution, we can obtain 0 as the result.

Ablation Study A procedure where certain parts of the network are removed in order to gain a better understanding of the network's behaviour.

Multi-loss Approach Training a neural network using multiple individual losses.

State-of-the-art Methods Methods that have the best performance on public benchmark datasets.

1.8 Summary

This chapter provides the background, significance, research questions, assumptions, limitation, delimitation, definitions, and other background information for the research project. In the background section, We identify the research gap by demonstrating the discontinuous property of Euler angles and quaternions and then present our solutions. Next, we further elaborate the problem by showcasing several examples from public benchmark dataset and show our contribution of this work in the significance part. Then, four research questions are proposed and need to be answered by the end of this study. Limitations and delimitations are listed for showing variables we can not control over and the scope of this study. For readers' better understanding, we add definitions of some technical terminology used in this paper at the end of introduction section.

The next chapter provides a review of the literature relevant to this research.

CHAPTER 2. REVIEW OF RELEVANT LITERATURE

Recent work using deep neural networks has achieved great success. Such work deploys CNNs to learn an end-to-end mapping from a single RGB or RGB-D image to the actual object poses. Compared with the traditional methods mentioned above, deep neural networks (DNNs) are more robust against the changes of the environment. A myriad of researches have been conducted in the past few years since advances in deep learning (such as the GPU-support computing and open-source framework) make it possible to easily train complex CNNs on large datasets. This chapter provides a thorough discussion about methods for head pose estimation and object pose estimation. In this section, before we jump into some very popular deep learning based approaches for pose estimation, we first start with the introduction of some current issue with Euler angles and quaternions for rotation representation. Then, we will discuss some classical methods and depth-based methods, which all make great contributions to this area before the arise of deep neural network. We further elaborate the limitation of existing rotation representations based on more recent work. Finally, some research work utilizing VIVE trackers is presented.

2.1 Limitations For Rotation

The widely used rotation representations such Euler angles and quaternion are always adopted as the ground truth annotations for a given dataset that contains pose information (Rennie, Shome, Bekris, & De Souza, 2016; Xiang et al., 2017; Zhu et al., 2016). Methods tested on these datasets have been validated to achieve state-of-the-art results.

However, recent work by Zhou et al (Zhu et al., 2016) argue that for rotation, all the representations are discontinuous in the real Euclidean spaces of four or fewer dimensions. Therefore, the Euler angles and quaternion that we use for almost every dataset would be discontinuous as well. Such discontinuity in 3D space makes it difficult for neural network to learn. Instead, in their work, they show that the 3D rotation has

continuous representation in 5D or 6D which are more suitable for neural network learning and demonstrate how any of the n dimensional rotations $SO(3)$ can be transformed into higher-dimension (5D and 6D) continuous representation. The empirical results found those representations with continuity properties work better for the learning process.

One of the previous works proposed by Grassia et al. (Saxena et al., 2009) also point out that parameterizing three degree-of-freedom (DOF) rotations is difficult. Widely used parameterizations such as Euler angles and quaternions are not able to compute and differentiate positions and orientations of articulated figures. They present the exponential map with three or two DOF rotations as a more robust representation. The exponential map is free from the gimbal lock issue which is a huge challenge for Euler angles representation. Compared with Zhou et al. (Zhu et al., 2016), it requires less parameters to parameterize $SO(3)$. Saxena et al. (Saxena et al., 2009) discuss the ambiguity of representations such as quaternions because the space of orientations is non-Euclidean. But they did not provide a general rotation representation but a symmetry invariant and continuous representation to address these problems since they focus more on the object with specific symmetries.

2.2 Classical Methods

Object pose estimation, especially head pose has always been a big concern in computer vision area. According to Murphy-Chutorian et al. (Murphy-Chutorian & Trivedi, 2008), In the context of computer vision, head pose estimation is most commonly interpreted as the ability to infer the orientation of a person's head relative to the view of a camera. Moreover, they (Murphy-Chutorian & Trivedi, 2008) argue that the human head is limited to three degree of freedom in pose, which can be characterized by pitch, roll, and yaw angles. Traditional methods like Appearance Template Models seeks to compare a new testing image with a set of exemplars (usually annotated with discrete pose labels) in order to find the most matching one (Ng & Gong, 2002; Sherrah, Gong, & Ong, 1999, 2001). Then, the testing image is given the same pose label that is assigned to the most similar within these templates. With the success of frontal face detection (Osuna, Freund,

& Girodit, 1997; Rowley, Baluja, & Kanade, 1998), Detector Arrays Methods are once very popular. The idea behind it is to train multiple face detectors, each specifies different head poses and assign a discrete pose to the detector with the greatest support. Detector Array and Appearance Template have many aspects in common such as they both operate directly on an image patch. Instead of only comparing the testing sample to a large set of templates, Detector Arrays evaluate the image based on the trained detectors which is a supervised learning process. One downside about Detector Arrays is that it requires many positive and negative data samples to train different face detectors for each discrete pose which is a very cumbersome process. One of the traditional pose estimation approaches that is similar to modern deep learning-based method is called nonlinear regression methods (Rowley et al., 1998). It learns a functional mapping from input images to pose measurement in a regression way. The biggest challenge in nonlinear mapping is the high dimensionality of an input image. Dimension reduction approaches such as PCA and localized gradient orientation histograms have been applied to address this problem very successfully. Therefore, regression tools like Support Vector Regressor can be used for learning the nonlinear relationship between the input and output. For human beings, the most straightforward way to identify the face orientation is by observing the facial features like nose, eyes and mouth, and the shape of the head especially when it is tilted. Inspired by this, a geometric method is proposed for pose estimation using local features to determine the pose configurations. According to Gee et al.'s work (Gee & Cipolla, 1994), we can use several facial landmarks to perform face alignment. If the assumption that some facial feature distances have fixed ratio is established, then the facial orientation could be determined from a geometric perspective. Geometric methods for pose estimation is fast and simple and the overall process is extremely explicable. The most challenging part is to detect the facial features accurately even in some hard cases where the images are in low resolution or the important features are highly occluded. Profile view could also be big challenge since some parts of the face will be invisible.

Other related methods include 3D model-based registration. Li et al. (Y. Li, Gu, & Kanade, 2011) improved the previous shape alignment methods by randomly and repeatedly sub-sampling the feature points and select the one with the least prediction error. It is more robust to the occlusion and background disturbance. Work from Lowe (1991) presents a method for fitting 3D model with arbitrary curved surfaces and any number of internal parameters to matching a 2D image features.

2.3 RGB-D Methods

RGB-D based methods become very popular since the emergence of inexpensive and good quality depth cameras as well as the advanced sensing technology. (Michel et al., 2017) is one of the large-scale datasets collected using RGB-D camera, which contains multiple views of 300 different objects. The availability of these datasets enables the fast growth of visual-based object detection and pose estimation.

Depth based methods like (Lowe, 1991) learns an intermediate representation of the 3D object coordinates. It deals with both textured and textured-less object as well as the lighting changing condition. Another method proposed by Choi et al. (Brachmann et al., 2014) specifically target on textured-less objects pose estimation. Possible coarse pose hypotheses first are established by chamfer matching. An annealing refinement process is then applied to improve the correspondence. Choi et al. (Choi & Christensen, 2016) also proposed another object pose estimation approach which fully exploits the depth and color information from RGB-D images and is independent of object segmentation. It is a voting-based approach and shows good performance even in unstructured environment. Lai et al. (Lai, Bo, Ren, & Fox, 2011) designed a new scalable approach called Object-Pose Trees. Three sub-tasks: category, instance and pose recognition can be solved simultaneously in near real-time by searching the whole database in a top down way because of the semantical tree-structure. Nearly 28,000 classifiers are used for training in the leaf level and stochastic gradient descent helps to speed up the training process when new objects are added to the database. According to (Michel et al., 2017), pose estimation task usually follows three steps: 1. find local

features of the target. 2. a set of pose hypotheses are established. 3. refinement is applied to the hypotheses and select the best one. Their work (Michel et al., 2017) focuses on the second step by using global reasoning to generate the hypotheses pool. In particular, an efficient two-step process based on Conditional Random Field (CRF) is proposed to generate small number of hypotheses. Their results outperform the state-of-art approach on Occluded Object Dataset. Pose estimation only based on range images is a big challenge. Zach et al. (Zach, Penate-Sanchez, & Pham, 2015) improved the previous random sampling based approach by keeping good samples of inliers for pose estimation in a dynamic programming way to efficiently address this problem.

2.4 CNN-based Methods

Posenet (Kendall, Grimes, & Cipolla, 2015) is one of the first attempts to use CNNs for pose estimation task. It trains an end-to-end convolutional neural network to directly regress the 6-DOF for a single RGB image. Both indoor and outdoor scenarios could be handled in real-time by the algorithm. Their success demonstrates that the deep neural networks have the capacity to learn pose information. (Xiang et al., 2017) proposed PoseCNN, a new convolutional neural network for 6D object pose estimation. The 3D rotation can be estimated by regressing to the quaternion representation. PoseCNN consists of a backbone network and three different branches for different tasks: semantic labeling, translation estimation and rotation estimation. A novel loss function is also introduced in their work for solving the pose estimation problem of symmetric objects. (C. Li, Bai, & Hager, 2018) presents a unified architecture for inferring the 6-DoF pose from both single-view and multi-view network. However, these methods highly rely on some refinement steps to fully utilize the depth information which could significantly slow down the computing. Therefore, alternative solutions are proposed again in some real-time needed applications.

In the work from (Patacchiola & Cangelosi, 2017), an end-to-end SSD based network is proposed to estimate 6-DoF space. Though it largely speeds up the computing process, the approximate 6D pose results need to be refined as a post-processing step. Inspired by the work (Kehl, Manhardt, Tombari, Ilic, & Navab, 2017), a new single-shot approach (Tekin, Sinha, & Fua, 2018) is proposed for simultaneously detecting an object from a RGB image and inferring the 6-DoF pose. The output is accurate enough so that there is no need for a post-processing step. In the cases where objects are partially occluded, a new method is proposed called BB8 (Rad & Lepetit, 2017). Object segmentation is first applied to mask around the object of interest. Then a holistic strategy is implemented using CNNs to predict the 3D pose in the form of 2D projections of the detected 3D bounding boxes. Furthermore, when an object shows the property of rotational symmetry which makes the neural network hard to train, a new classifier is introduced to filter out poses that are not in specified ranges before estimating it in a neural network. Their work improves the state-of-art result on LINEMOD dataset from 73.7% to 89.3% of correctly registered RGB frames.

Recent works involving head pose estimation such as (Hsu et al., 2018) and (Ruiz et al., 2018) predict the head pose from RGB images without depth information. A base network is used for feature extraction and then several branches are extended out for regressing each component of the Euler angle or quaternion. Their joint binned pose classification and regression achieved state-of-art performance on several benchmark datasets.

2.5 State-of-the-Art Approaches

Latest state-of-the-art approaches explore the research boundary and improve the topic by a significant margin. In order to avoid using Euler angles to address the problem of gimbal lock, (Hsu et al., 2018) further presents their quaternion based approach to head pose estimation task. (Hsu et al., 2018) designs a quaternions based method with multi-regression loss which achieves state-of-the-art performance on the AFW and AFLW2000 test sets. It outperforms other methods that utilize depth data and still

achieves high precision. Keypoints-based facial analysis enables us to accurately recover the 3D head pose. To answer the question of whether keypoints-based approach is the best way forward in applications where all we need to be estimated is the pose, (Ruiz et al., 2018) presents a fine-grained structure with a multi-loss to determine head poses on public datasets such as AFLW2000 and 300W-LP, by training a neural network to predict three Euler angles directly from an image through binned classification and expectation regression. (Ruiz et al., 2018) conduct extensive experiments on common pose benchmark datasets to show their state-of-the-art results. More recently, (Chen, Wu, Richter, Konrad, & Ishwar, 2016; T.-Y. Yang et al., 2019) propose a fine-grained structure for learning the importance of spatial features. Model ensemble is then performed by using aggregated features. Experiment results show that their approach (T.-Y. Yang et al., 2019) outperforms other state-of-the-art methods including depth based approaches and keypoints based approach. Results on several public datasets also show that the yaw angle prediction is extremely accurate compared with methods that utilize more multimodality information such as depth and time sequence. Latest research which is similar to our method is from (Y. Zhou et al., 2019). (Y. Zhou et al., 2019) suggests that achieving a continuous representation of rotations in 3D space requires the use of at least 5 dimensions of information to complete the pose estimation task. However, their method cannot have precise prediction on the first column of rotation matrix which introduces bias in the following computation.

Different from the discussed methods, we use three vectors obtained by converting three Euler angles(roll, pitch, yaw) to a rotation matrix. Our new vector-based network learns a three-component regression for each of the three vectors. Our experiments show state-of-the-art results on AFLW2000, BIWI and AFW datasets.

2.6 Applications of VIVE Trackers

Recent virtual reality technology has developed many applications for entertainment and education purposes. All of the efforts are contributing to create an immersive environment to enhance people's feeling of being in the virtual world. HTC Vive Tracker is a fist-sized and self-contained unit that allows a wide range of objects to be tracked.

VR and human body movement have built strong connection since the best way to make people feel present in the virtual reality is to synchronize the avator's action with body motion. Work from (Caserman, Garcia-Agundez, Konrad, Göbel, & Steinmetz, 2019) addresses the problem of real-time body tracking in virtual reality, they adopt HTC Vive tracker and headset to animate the moving trajectory of the avatar as smoothly as possible. With two base stations and multiple infrared sensors, their approach suffers less from the occlusion situation and still acquires precision rotation data. The alignment between virtual and real spaces is an important set-up work. Peer et al.(Peer, Ullich, & Ponto, 2018) propose a new alignment approach for the Vive tracking system by using three different trackers to keep track of the origin in the real space. This helps to reproduce the movement in virtual environment more accurately. Another method for calibrating the HoloLens's front color camera is proposed(Bai, Gao, & Billinghurst, 2017) by tracking Vive tracker with lighthouse sensors, which enables efficient user input for controlling augmented objects compared with gesture based interface. Similar to (Spitzley & Karduna, 2019), they believe that Virtual Reality system has great potential to be an effective tools to collect kinematic data. Therefore, they adopt HTC VIVE VR system to validate its accuracy by measuring transnational and orientation signals. (Luckett, 2018) presents a quantitative study of HTC Vive tracking system and its latency. Their investigation shows that the system precision is high while the latency is relatively low. However, they also notice that large change in offset between virtual and physical space due to the lost of tracking may lead to the problem of unsynchronization, which is not suitable for scientific research that requires fast and accurate visual manipulation.

2.7 Summary

This chapter provides a review of the literature relevant to head pose estimation and current deep learning based state-of-the-art approaches. The next chapter provides the framework and methodology to be used in the research project.

CHAPTER 3. FRAMEWORK AND METHODOLOGY

This chapter provides the framework and methodology to be used in the research study. Here, we present a thorough discussion about our proposed TriNet architecture for head pose estimation from a single image. TriNet is composed of three different subnets for predicting three vectors. Each subnet is trained independently and then jointly (See Fig. 3.1). Here, we first formulate the head pose estimation problem by using vector-based representation (Sec 3.1) and explain the advantages of using vectors for rotation representation. Then, we give an overview of the proposed network (Sec 3.2). Implementation details are introduced (Sec 3.3). We propose an optimization strategy for vector refinement to attain three orthogonal vectors (Sec 3.4). Finally, we demonstrate the process of collecting our own data with VR tracking system (Sec 3.5).

3.1 Vector-Based Representation

As mentioned in section 1, we use three vectors (left, down, and front vectors) to describe human head poses and also use them as the output of our neural network. Since the datasets (Zhu et al., 2016; Zhu & Ramanan, 2012) we use in the experiments are annotated by Euler angles, we need to convert them to vectors. We first get the rotation matrix $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$ from Euler angles in accordance with the order stipulated by the author of the dataset. Then from rotation matrix \mathbf{R} we can get the left vector (\mathbf{v}_l), down vector (\mathbf{v}_d), and front vector (\mathbf{v}_f) respectively by the following equations:

$$\mathbf{v}_l = \mathbf{R} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \mathbf{r}_1, \mathbf{v}_d = \mathbf{R} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \mathbf{r}_2, \mathbf{v}_f = \mathbf{R} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{r}_3 \quad (3.1)$$

The equations above show that predicting three vectors of head pose is in essence equivalent to predicting the three column vectors of rotation matrices. Since rotation matrices form a continuous special orthogonal group $SO(3)$, and have a non-ambiguous representation of rotations (Saxena et al., 2009), using three vectors to describe rotations shares the same advantages and doesn't have the issue of discontinuity or ambiguity of Euler angles and quaternions.

Even though the third vector seems redundant as it can be obtained by the cross product of the first two vectors, our experiments show that the neural network can not predict two vectors with the same accuracy. Since it's impossible to know in advance on which two vectors the neural network would have better performance, the results would be highly biased if the output only contains two vectors. We predict all three vectors and put constraints between each pairing of them to punish when they are not perpendicular to each other.

3.2 TriNet Architecture

The main contribution in this study is that we propose a novel deep neural network structure that can be utilized to predict three directional vectors directly from a RGB image. Fig. 3.1 describes the overall architecture of our proposed TriNet. Unlike many other previous works, we do not supervise the neural network to regress Euler angles or quaternions, but rather a more robust 9-D rotation representation composed by vectors is presented for the neural network learning. We describe how we can obtain three vectors in Sec 3.1.

As shown in Fig.3.1, the whole architecture can be well divided into four different components: input, subnets, output and refinement. Our vector-based approach is built on a convolutional network architecture that produces three unit-vectors for determining the 3-D orientation. The basenet which is well pretrained on large-scale image datasets for classification can be replaced by any backbone networks which map input images to feature maps. Three auxiliary subnets sharing one backbone network are designed for predicting three vectors separately. Each subnet has the same structure with three

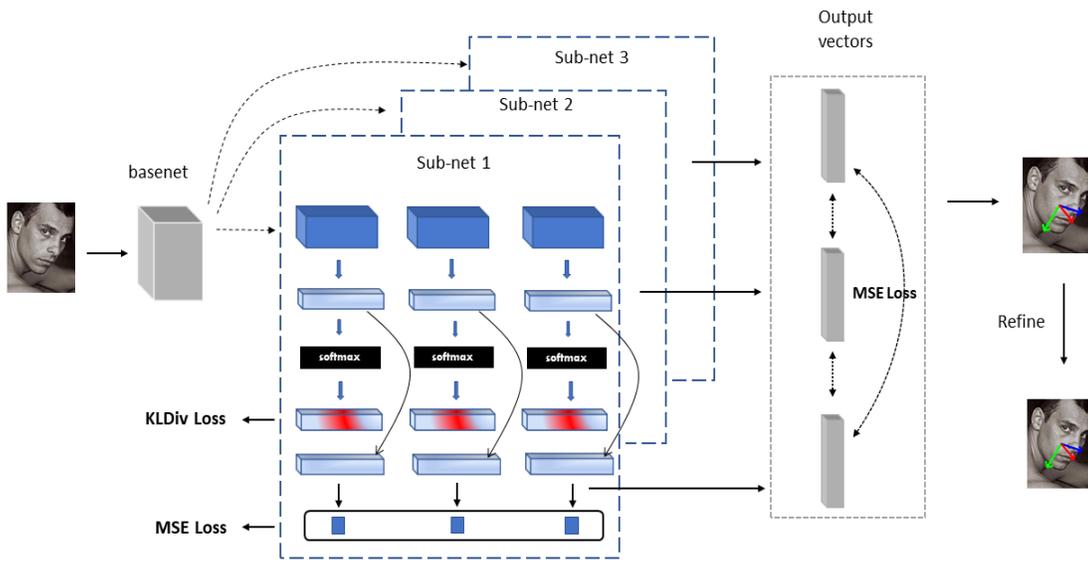


Figure 3.1. Detailed overview of our proposed TriNet. An input image goes through a backbone network followed by three subnet modules. Each subnet has identical structure and can generate one unit vector. A Post-processing step for vector refinement is proposed to achieve our final results.

fully-connected layers of fixed size. A fine-grained structure here addresses a regression problem by converting it to a binned classification task. By mapping the network outputs into a probability distribution using softmax layer, we can predict each vector component in a more robust way by computing the aggregated expectation of all the bin categories as our final output. Three subnets then can be trained jointly by adding orthogonal constraints onto each pair of vectors. The purpose of this step is to largely minimize the dot-product results of two vectors. We add an additional post processing step followed by the neural network outputs to fix three unit vectors to be orthogonal. The refinement process can be accomplished by converting it to an optimization problem. More details on how we solve the optimization problem will be discussed in section 3.4. For the rest of this section, we will mainly introduce some of the key implementations in our proposed method, including soft-labeling, expectation regression and vectors orthogonality.

Soft labeling One-hot is a very widely-used technique and is always been in the dominant position in image classification tasks. It encodes the object categories into binary representation, where 1 denotes that the image/object falls into that specific category while 0 stands for the rest all. In this work, we replace the one-hot encoded labeling with soft-labels for the binned classification task since soft-labels better capture the interclass relationship. We also notice that the public benchmark dataset such as AFLW has coarse manually-annotated pose data. By smoothing the one-hot target, the neural network attempts to predict the neighbors of the ground-truth target. An weighted average output helps to reduce the prediction error caused by the inaccurate data labeling. Here, let $C = \{c_1, c_2, \dots, c_m\}$ denote m bin categories and the target falls into the k^{th} bin. We compute each element instance s_i in our encoded soft-label vector as:

$$s_i = \frac{e^{-(\mu c_k - \mu c_i)^2}}{\sum_{n=1}^m e^{-(\mu c_k - \mu c_n)^2}} \quad (3.2)$$

where $(\mu c_k - \mu c_i)^2$ is a distance measurement between i^{th} bin and the target bin and μ is an expand ratio parameter that determines how smooth the probability distribution will be. The instance value decreases as the distance to the k^{th} bin becomes larger as shown in Fig 3.2. Too large expand ratio will make the probability distribution close to one-hot distribution and if the expand ratio is set to a extremely small one, the values will be generated evenly with only slight difference which does not help train the neural network. For all the experiments discussed in section 4, we set μ value to be 0.5, in that the neighbors of the target bin that has the largest probability span nearly 10 degrees, which we believe is within the range of label errors.

See Fig. 3.2

Expectation Regression Each subnet regresses to obtain three components of a vector. By performing softmax to the fully-connected layer outputs, a probability distribution is then generated for each bin category, suggesting that the final regression value is highly related to bins that are assigned with high probability values. We take advantages of the probabilities produced by the soft-label and compute the weighted average value for the binned outputs to obtain a more robust prediction.

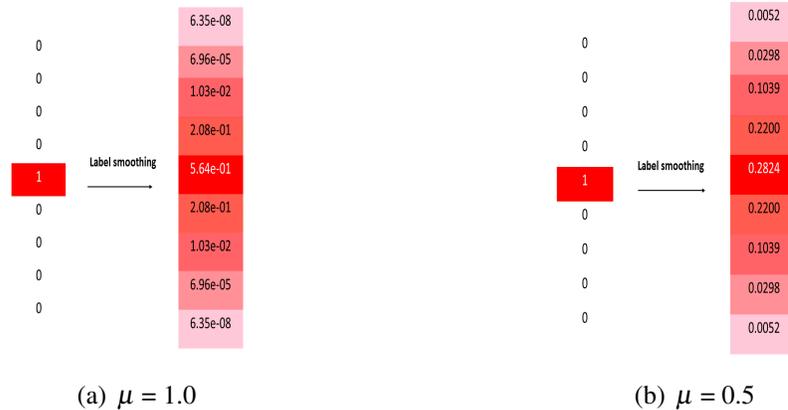


Figure 3.2. Soft label encoding with different expand ratio.

$$\begin{aligned}
 \mathbb{E}([b_1, b_2, \dots, b_m]) &= \sum_{i=1}^m b_i * p_i \\
 &= \sum_{i=1}^m b_i * \frac{e^{x_i}}{\sum_{j=1}^m e^{x_j}}
 \end{aligned} \tag{3.3}$$

Vectors orthogonality Three subnets are trained independently to produce three vectors which will pose a problem of loss of orthogonality between any two vectors. However, our vector-based representation requires pairwise orthogonal. We address this issue by two steps. In the training stage, we first add an orthogonality loss to each pair of predicted vectors. Then, as a post-processing step, we propose an optimization strategy to refine three output vectors.

3.3 Multi-loss approach

Multiple losses are proposed in our network, namely, KullbackLeibler divergence loss (\mathcal{L}_1) for binned classification, MSE Loss (\mathcal{L}_2) for vector regression and MSE loss (\mathcal{L}_3) for measuring vector orthogonality.

KullbackLeibler divergence is used as our classification loss. It measures the similarity between two probability distributions. In cases where we have an exact same match between network’s outputs ($\hat{\mathbf{y}}$) and soft labels (\mathbf{y}), the classification loss will reduce to 0. This also helps prevent the situation of overfitting on training dataset, since the network learns to generate relatively high probability values for bins that are close to ground truth target. We present two regression components in our network, the first one \mathcal{L}_2 is to minimize the distance between our predicted vectors ($\hat{\mathbf{v}}$) and ground truth vectors (\mathbf{v}). The second one \mathcal{L}_3 is to regress vectors to be orthogonal. In particular, different metrics can be used for evaluating the distance between two vector, here, we compute the angle difference instead of the Euclidean distance. They are equivalent and can achieve the same effect. Multi-loss approach is a necessary step for training the neural network since our target prediction is a 9-D representation and involves multiple interrelationship. Our ablation experiments on several datasets in section 4 show that the removal of any of these losses would cause the low accuracy performance on testing set.

$$\mathcal{L}_1 = \text{KLDivLoss}(\text{softmax}(\hat{\mathbf{y}}), \mathbf{y}) \quad (3.4)$$

$$\mathcal{L}_2 = \text{MSE}(\arccos(\mathbf{v}^T \hat{\mathbf{v}}), 0) \quad (3.5)$$

$$\mathcal{L}_3^{(i,j)} = \text{MSE}(\hat{\mathbf{v}}_i^T \hat{\mathbf{v}}_j, 0) \quad (3.6)$$

Additionally, we have three orthogonality losses $\mathcal{L}_3^{(1,2)}$, $\mathcal{L}_3^{(1,3)}$ and $\mathcal{L}_3^{(2,3)}$ for each pairing of predicted vectors. We add each $\mathcal{L}_3^{(i,j)}$ to its associated subnets(i and j). Each subnet then has a total loss \mathcal{L}_{sub}^i which is the linear combination of \mathcal{L}_1 , \mathcal{L}_2 and \mathcal{L}_3 with loss weights α and β . The best testing result can be obtained by fine-tuning the weight coefficients. By having orthogonality losses incorporated in the training process, we have three signals that can be backpropagated into previous layers in each subset simultaneously which improves the learning.

$$\mathcal{L}_{sub}^i = \mathcal{L}_1 + \alpha \mathcal{L}_2 + \beta \sum_{i \neq j} \mathcal{L}_3^{(i,j)} \quad (3.7)$$

Techniques and implementation details inside the architecture are all mentioned in the above sections. Next, we will introduce the method we use for vectors refinement.

3.4 Vector Refinement

Even though we impose penalty terms $\mathcal{L}_3^{(i,j)}$ in the loss function as orthogonality constraints between each pair of vectors, the three vectors that TriNet predicts may still not be perpendicular to each other. Therefore, it's necessary to select three orthogonal vectors to match the estimated vectors as closely as possible.

We use \mathbf{v}_i and \mathbf{v}'_i to denote the real location of each vector and what the neural network predicts respectively. Suppose \mathbf{v}'_i are the results of \mathbf{v}_i affected an independent and identically distributed Gaussian noise and hence perturbed by angle $\Delta\theta_i$. Through maximum likelihood inference, the problem of finding three best-match orthogonal vectors can be transformed into solving the following optimization problem with 6 constraints:

$$\begin{aligned} \mathbf{v}_i^* &= \operatorname{argmax}_{\mathbf{v}_i} \prod_{i=1}^3 P(\mathbf{v}_i | \mathbf{v}'_i) \\ &= \operatorname{argmax}_{\mathbf{v}_i} \prod_{i=1}^3 \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\Delta\theta_i)^2}{2\sigma^2}\right) \\ &= \operatorname{argmin}_{\mathbf{v}_i} \sum_{i=1}^3 (\Delta\theta_i)^2 \\ &\text{subject to } \|\mathbf{v}_i\|_2^2 = 1, \\ &\quad \mathbf{v}_i^T \mathbf{v}_j = 0, \text{ where } i \neq j, \quad i, j = 1, 2, 3. \end{aligned} \quad (3.8)$$

As we show in the section 3.1 that the three vectors are in nature equivalent to columns of rotation matrices. Suppose a rotation matrix \mathbf{R} is formed by Euler angles in the order of roll (α), pitch (β), yaw (γ):

$$\mathbf{R}(\alpha, \beta, \gamma) = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix} \quad (3.9)$$

The ambiguity problem, i.e. the phenomenon of gimbal lock can be eliminated by limiting the angles in specific ranges. Then we can turn the constrained optimization problem to an unconstrained problem by solving the Euler angles as following:

$$\begin{aligned} \alpha^*, \beta^*, \gamma^* &= \operatorname{argmin}_{\alpha, \beta, \gamma} \sum_{i=1}^3 (\Delta\theta_i)^2 \\ &= \operatorname{argmin}_{\alpha, \beta, \gamma} \sum_{i=1}^3 \|\arccos(\mathbf{r}_i^T \mathbf{v}'_i)\|^2 \\ &\text{where } \alpha, \gamma \in (-\pi, \pi], \beta \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right). \end{aligned} \quad (3.10)$$

\mathbf{v}'_1 , \mathbf{v}'_2 and \mathbf{v}'_3 are the left, down and front vectors that our network predicts respectively. With Euler angles obtained, we then build the rotation matrix and take its three column vectors as our final results.

3.5 Data Collection

In this study, we mainly focus on comparing the performance of our proposed method with other state-of-the-art methods on several public head pose datasets. As a supplement to existing datasets for head poses, we also like to test the applicability of proposed method for estimating poses from other object to demonstrate its generalization

and replicability beyond the scope of head pose. Therefore, we utilize VR devices including HTC VIVE Goggle and VIVE Trackers to build our own dataset. The target object is an 3D-printed bottle in white color. In order to make it more distinguishable, we manually colorize it and add more marks on it.

We propose a new pipeline for pose data collection. The devices include: 1. HTC VIVE tracker, 2. HTC VIVE pro headset. We use Unreal Engine 4 to build a virtual environment which sets a global coordinate system for both tracker and headset. Fig.3.4 describes the overall procedure for data collection. The tracker is a small sensor device that can produce translational and rotation signals relative to two base stations. It can be easily attached to an object and move around along with the object. The whole process can be demonstrated as following:

1. The tracker will be attached to a known object and the initial tracker position will be adjusted to ensure that it begins with the same position.
2. Participants will be asked to wear the VIVE Goggle and hold the object in their hands.
3. Participants will be given instructions of how and when they should start moving the object. They will be notified if the bottle object is out of the camera view.
4. Once the start signal is sent out, OBS starts to record the video and meanwhile, Unreal Engine starts to save log data. By checking the timestamp, we can match each video frame with its corresponding data record.
5. Two processes will be applied separately on log data and image data: one is to remove image distortion and the other is to convert Euler angles to vector representation.

Particularly, in step 5, we define the front vector of the object as $x^f = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ and

the left vector as $x^l = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$.

When the participants put on the VIVE headset, they need to follow instructions to move and rotate the objects for 60s. The RGB camera on the headset can be used for capturing the video stream. Video frame extraction is implemented by setting the extract frequency to 30 HZ because tracker device receives the signal and transmits to Unreal Engine with the time interval of $\frac{1}{30}$ seconds. Therefore, each participant can generate about 1800 images during this time period. We then further process the video and pose data to convert them to the format that we can later use.

These images suffer from severe distortion due to the fact that VIVE headset use wide-angle lenses for its front-facing camera. We remove the distortion by using Opencv toolkit (Bradski, 2000). The results can be shown as Fig. 3.3.

We then compute the corresponding front and left vectors in the headset camera coordinate system. Since both tracker and headset only provide the pose information in the form of Euler angles (roll, pitch, yaw), we first need to convert these three angles to three rotation matrices M_{roll} , M_{pitch} and M_{yaw} . We then compute front and left vectors using the following formulas:

$$M = M_{\text{roll}} * M_{\text{pitch}} * M_{\text{yaw}}$$

$$\mathbf{x}^f = M \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}; \quad \mathbf{x}^l = M \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

(3.11)



Figure 3.3. Before distortion removal(left) and after(right).

As the steps mentioned above, our dataset consists of two major components: images and labeled pose data. The total amount of images in our dataset is about 25000 with 15 different IDs. We split the data into training set and testing set with the ratio of 4:1. Therefore, 80% of the data will be used for the neural network training and the rest 20% is used for evaluating the performance of the trained model. We separate the data in the way that one individual will either appear in training set or testing set in order to ensure the validity of the testing process.

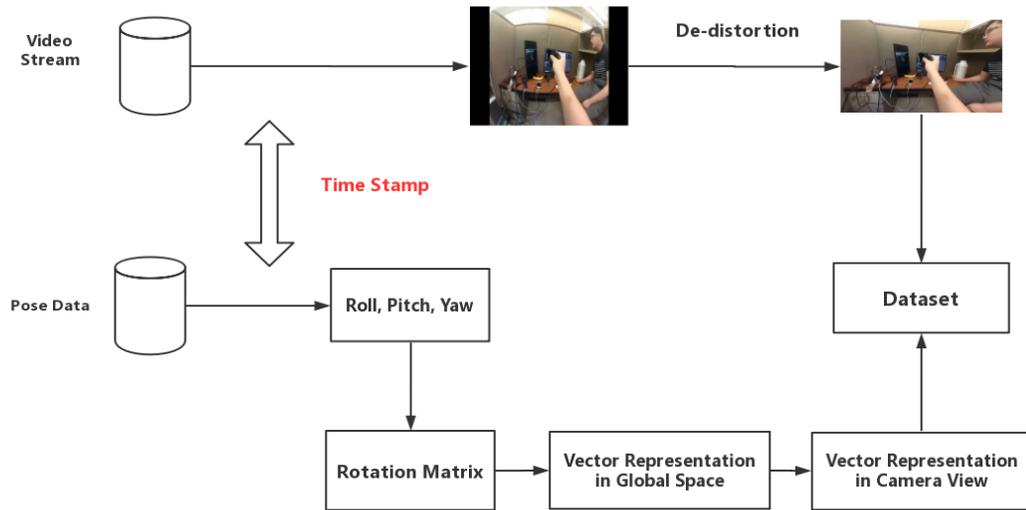


Figure 3.4. Proposed pipeline for data collection with VR devices.

3.6 Summary

This chapter provides the framework and methodology to be used in the research study. We start with the problem formulation in this section and present our approach for addressing the pose estimation problem by converting angle representation to vector representation. Then, a novel deep neural network structure named TriNet is proposed in this study to predict three directional vectors mentioned above. Structure details are introduced in section 3.2. A multi-loss strategy is proposed for training the neural network which improves the learning. We give a detailed description of multiple losses that we use. Another key component followed by the network is the refinement stage. The purpose of this step is to fix the orthogonality of two vectors. Finally, we present a new method for pose data collection using Vive devices in order to test our proposed network on different objects. For the next chapter, we will report our experiment results on several datasets, including our own data.

CHAPTER 4. EXPERIMENT AND RESULTS

In this chapter, we articulate the details of our experiments and results. We use extensive experiments to evaluate the proposed method.

4.1 Implementation Details

We implement our proposed network using Pytorch. Several approaches are adopted for augmenting the data including: image blurring, image grayscale, random crop and random flip. We recompute three vectors for the cases that the images are flipped. We train the network using Adam optimizer with an initial learning rate of 0.001 over 30 epochs. For the first 8 epochs, we set the learning rate for trainable parameters in both convolutional layers and fully connected layers to be 10 times of the initial learning rate to speed up the training process. After the 8th epoch, the learning rate starts from 0.001 with a decay rate of 0.9 for every epoch. We present our multi-loss network by using two different backbone networks, namely, ResNet50 (He, Zhang, Ren, & Sun, 2016) and MobileNetV2 (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018) to investigate the influence of different backbones on our network. Experiments show that ResNet50 can achieve better precision results while MobileNetV2 is more lightweight (but still achieves satisfactory results). We experiment with three different hyperparameters: number of bins (n), regression loss weight (α) and orthogonality loss weight (β) and show the results in Table 4.3. The experiments are conducted on a lab PC with two RTX 2080 Ti GPU support.

4.2 Datasets and Evaluation Metrics

We use three public datasets to carry out our experiments: AFW (S. Yang et al., 2016), AFLW (Martin Koestinger & Bischof, 2011), and BIWI (Fanelli, Dantone, Gall, Fossati, & Van Gool, 2013) datasets, along with our own VR dataset. We measure the performances of different methods by calculating the mean absolute error (MAE) of Euler angles.

4.2.1 VR dataset

As discussed in the methodology section, since Vive tracking system has very high accuracy and produces the feedback in real-time, it then can be used for pose data collection. The raw video data captured by Vive headset has the resolution of 1280 * 720. After the removal of distortion, the new processed video frame is now of the new size of 960 * 720. The orientation signal we obtain from the tracker is in the form of Euler angle with the rotation order of roll, pitch and yaw in a left-handed coordinate system. We collect 25000 images containing 15 different participants.

4.2.2 300W-LP

The 300W-LP dataset is expanded from 300W dataset (Sagonas, Tzimiropoulos, Zafeiriou, & Pantic, 2013) which is constituted of several standardized datasets, including AFW (Zhu & Ramanan, 2012), HELEN (E. Zhou, Fan, Cao, Jiang, & Yin, 2013), IBUG (Sagonas et al., 2013) and LFPW (Belhumeur, Jacobs, Kriegman, & Kumar, 2013). By means of face profiling, this dataset generates 61,225 synthesized images based on around 4,000 pictures from the 300W dataset.

4.2.3 AFLW

The AFLW dataset (Martin Koestinger & Bischof, 2011) contains 24,384 annotated face images obtained from the web. The first 2,000 images are also known as AFLW2000 dataset (Zhu, Lei, Yan, Yi, & Li, 2015). The pose information is obtained by fitting a mean 3D model (Storer, Urschler, & Bischof, 2009) to the annotated landmarks on the images. Even though the estimated poses are not perfectly accurate, this dataset possesses a wide range of varieties in facial appearances and background settings which make it a good dataset to train and test our network.

4.2.4 BIWI

The BIWI dataset (Fanelli et al., 2013) contains 15,678 pictures of 20 participants who try to span all possible Euler angles by turning their heads around freely in an indoor environment. Since the dataset does not provide bounding box of human heads, we use MTCNN (K. Zhang, Zhang, Li, & Qiao, 2016) to detect human faces and loosely crop the area around its results as the bounding boxes of the human heads. Samples of ground truth of BIWI dataset and the estimation results of our method are shown in Fig. 4.1.

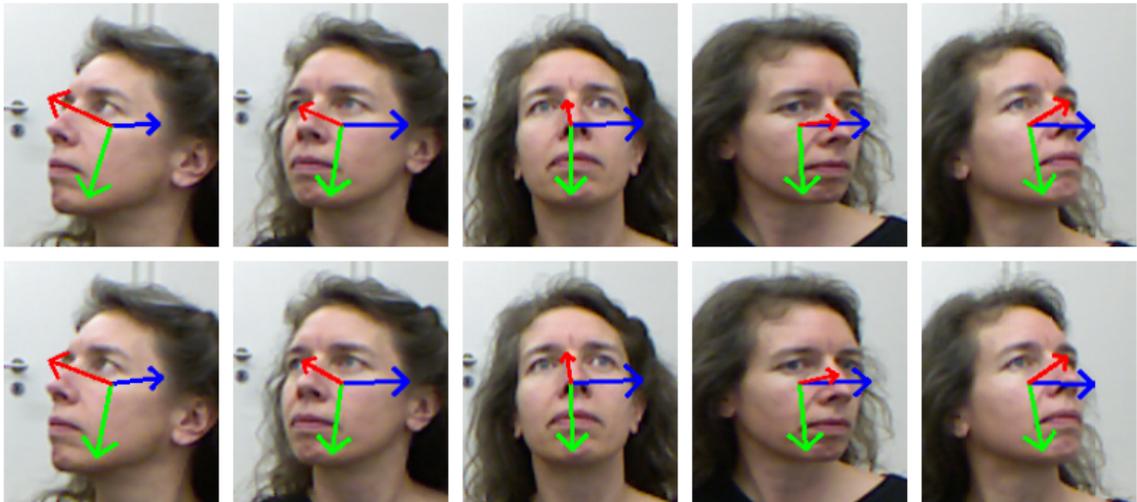


Figure 4.1. First row : ground truth of BIWI dataset, Second row: our results

4.2.5 AFW

The AFW dataset (S. Yang et al., 2016) is comprised of 205 images with 468 human faces and their poses are coarsely annotated with the accuracy of 15° . Similar to AFLW, the images also have a large difference in their head poses and environments which makes it a widely used benchmark to evaluate performances.

4.3 Comparison to State-of-the-art Methods

We evaluate TriNet on ALFW2000 (Martin Koestinger & Bischof, 2011), BIWI (Fanelli et al., 2013) and AFW (S. Yang et al., 2016) datasets and compare our results with other state-of-the-art methods. Traditionally, facial landmark based approach 3DDFA (Zhu et al., 2016) tries to fit a dense 3D model to an RGB image through a Cascaded CNN architecture. The alignment framework applies to large poses up to 90 degrees. KELPLER (Kumar et al., 2017) presents H-CNN for face keypoints detection as well as 3D poses as a by-product. Recently, some deep learning based methods estimate head poses from a single image without depth information achieve state-of-the-art results. Hopenet (Ruiz et al., 2018) proposes a fine-grained structure by combining classification loss and regression loss to predict the head pose in a more robust way. Quatnet (Hsu et al., 2018) uses quaternions labeling data for training the model to avoid the ambiguity of Euler angle representation. FSA-Net (T.-Y. Yang et al., 2019) proposes a fine-grained structure for learning the importance of spatial features. Model ensemble is then performed by using aggregated features.

We follow the train and test set split convention of (Ruiz et al., 2018). We train our TriNet on the AFLW dataset without the first 2,000 images and test it on AFLW2000 and AFW dataset. AFW dataset labels the data with 15° intervals, so we round the testing results to the nearest 15° multiple. For BIWI, we split the whole dataset into training and testing sets by person's IDs with the ratio of 7:3.

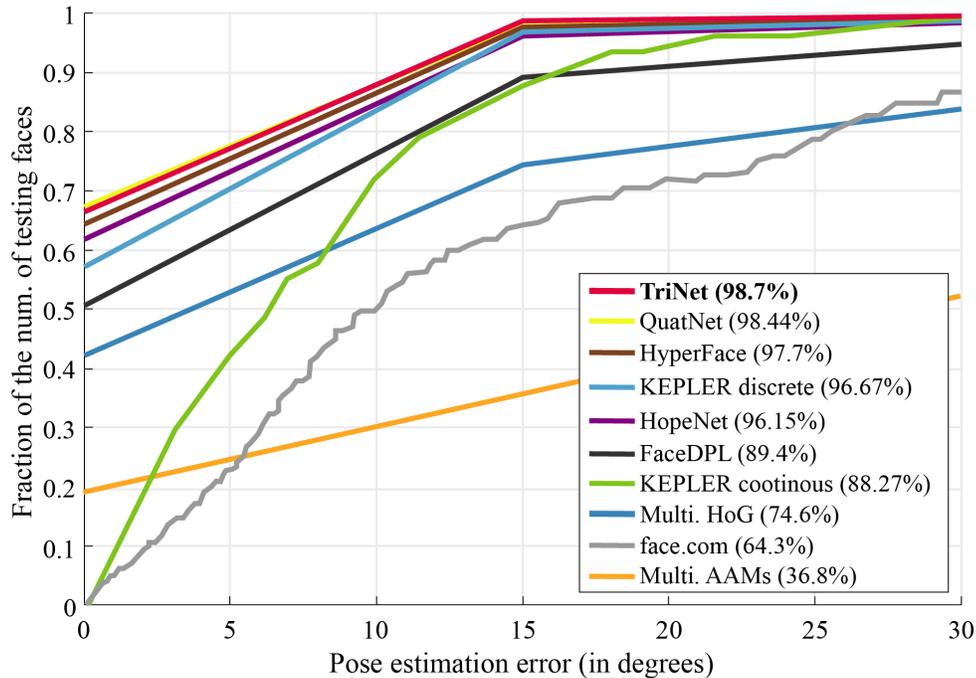


Figure 4.2. Comparison with other methods on the AFW dataset

We achieve mean absolute error (MAE) of 3.86 degrees on AFLW2000 dataset and 3.97 degrees on BIWI dataset respectively. The comparison between our TriNet with other state-of-the-art methods on AFLW2000 and BIWI datasets is shown in Table 4.1 and 4.2. For the AFW dataset, with the error margin of 15° , our method achieves the accuracy of 98.7% on yaw angles prediction. The results are shown in Fig. 4.2.

Table 4.1. Mean average error of Euler angles using different methods on AFLW2000 dataset.

| Method | roll | pitch | yaw | MAE |
|--|-------------|-------------|-------------|-------------|
| Dlib(Kazemi & Sullivan, 2014) | 10.545 | 13.633 | 23.153 | 15.777 |
| 3DDFA(Zhu et al., 2016) | 8.250 | 8.530 | 8.540 | 7.393 |
| Hopenet(Ruiz et al., 2018) | 5.674 | 6.637 | 6.920 | 6.410 |
| FSA-Caps-Fusion(T.-Y. Yang et al., 2019) | 4.64 | 6.08 | 4.50 | 5.07 |
| Quatnet(Hsu et al., 2018) | 3.920 | 5.615 | 3.973 | 4.503 |
| TriNet(MobileNetV2) | 2.86 | 5.25 | 5.34 | 4.48 |
| TriNet(ResNet50) | 2.36 | 4.00 | 3.94 | 3.43 |

Table 4.2. Mean average error of Euler angles using different methods on BIWI dataset.

| Method | roll | pitch | yaw | MAE |
|--|-------------|-------------|-------------|-------------|
| Dlib(Kazemi & Sullivan, 2014) | 9.324 | 15.505 | 20.581 | 15.137 |
| KEPLER(Kumar et al., 2017) | 16.196 | 17.277 | 8.084 | 13.852 |
| 3DDFA(Zhu et al., 2016) | 11.770 | 11.180 | 8.691 | 10.547 |
| Trinet(MobileNetV2) | 6.94 | 6.75 | 4.02 | 5.90 |
| Hopenet(Ruiz et al., 2018) | 3.269 | 6.606 | 4.810 | 4.895 |
| Quatnet(Hsu et al., 2018) | 2.936 | 5.492 | 4.010 | 4.146 |
| FSA-Caps-Fusion(T.-Y. Yang et al., 2019) | 2.76 | 4.96 | 4.27 | 4.00 |
| TriNet(ResNet50) | 3.67 | 4.99 | 3.27 | 3.97 |

4.4 Architecture Evaluation

In this section, we evaluate the influence of different settings of network hyperparameters have on the performances of evaluations. Table 4.3 shows the results on AFLW2000 test dataset across different bins numbers and weight coefficients. We observe the best performance on this dataset when we have 90 bins and set regression weight(α) and orthogonality weight(β) to be 1.0 and 0.75, respectively. In our experiment, we show that increased accuracy can be achieved when having the orthogonality loss for training. However, this is such a strong constraint that only by keeping it below a certain value can it help the learning process and not compromise the vector prediction. The neural network fails to converge on the training set when we have β values larger than 2.0. TriNet shows satisfactory results on AFLW2000 dataset in Fig. 4.3. By allowing 10 degrees of prediction error from the ground truth, we achieve 98% accuracy on roll angles, 91% accuracy on pitch angles and 90% accuracy on yaw angles.

4.5 Ablation Study

In this section, we conduct an ablation study to analyze how each network component will affect the performance on AFLW2000 and BIWI datasets. We present our results in Table 4.4. For method A, we only use the regression module to directly predict three vectors without orthogonality constraints to achieve a baseline result. Then, we test

Table 4.3. MAE across different bin numbers and loss weights on AFLW2000 dataset using ResNet50 as basenet

| | | | Vector errors | | | | Equivalent Euler angle errors | | | |
|-----|----------|---------|---------------|-------------|-------------|-------------|-------------------------------|-------------|-------------|-------------|
| bin | α | β | v_1 | v_2 | v_3 | MAE | roll | pitch | yaw | MAE |
| 40 | 1.0 | 0.25 | 5.06 | 4.94 | 6.21 | 5.40 | 2.44 | 4.03 | 4.27 | 3.59 |
| | | 0.50 | 4.85 | 5.08 | 6.04 | 5.32 | 2.50 | 4.01 | 3.96 | 3.49 |
| | | 0.75 | 4.97 | 5.03 | 6.21 | 4.40 | 2.64 | 4.10 | 4.34 | 3.69 |
| | | 1.0 | 5.01 | 5.15 | 6.22 | 5.46 | 2.62 | 4.15 | 4.30 | 3.69 |
| | 2.0 | 0.25 | 4.93 | 5.30 | 6.34 | 5.52 | 2.50 | 4.31 | 4.16 | 3.66 |
| | | 0.50 | 5.04 | 5.15 | 6.29 | 5.50 | 2.46 | 4.08 | 4.16 | 3.57 |
| | | 0.75 | 5.05 | 5.07 | 6.33 | 5.48 | 2.60 | 4.22 | 4.44 | 3.76 |
| | | 1.0 | 5.02 | 5.16 | 6.32 | 5.50 | 2.83 | 4.48 | 4.62 | 3.98 |
| 60 | 1.0 | 0.25 | 5.11 | 5.38 | 6.53 | 5.67 | 2.54 | 4.46 | 4.28 | 3.76 |
| | | 0.50 | 4.90 | 5.07 | 6.23 | 5.40 | 2.39 | 4.10 | 4.10 | 3.53 |
| | | 0.75 | 5.36 | 5.46 | 6.71 | 5.85 | 2.57 | 4.37 | 4.43 | 3.80 |
| | | 1.0 | 5.24 | 5.18 | 6.52 | 5.65 | 2.38 | 4.12 | 4.44 | 3.65 |
| | 2.0 | 0.25 | 4.91 | 5.09 | 6.21 | 5.40 | 2.59 | 4.19 | 4.31 | 3.70 |
| | | 0.50 | 5.20 | 5.08 | 6.25 | 5.51 | 2.54 | 3.98 | 4.28 | 3.60 |
| | | 0.75 | 4.92 | 5.12 | 6.26 | 5.43 | 2.50 | 4.17 | 4.21 | 3.63 |
| | | 1.0 | 4.98 | 5.09 | 6.22 | 5.43 | 2.71 | 4.22 | 4.40 | 3.78 |
| 90 | 1.0 | 0.25 | 4.91 | 5.31 | 6.38 | 5.53 | 2.33 | 4.22 | 4.02 | 3.52 |
| | | 0.50 | 5.15 | 5.21 | 6.47 | 5.61 | 2.47 | 4.24 | 4.35 | 3.69 |
| | | 0.75 | 4.90 | 5.10 | 6.15 | 5.39 | 2.36 | 4.00 | 3.94 | 3.43 |
| | | 1.0 | 5.13 | 5.22 | 6.34 | 5.57 | 2.44 | 4.05 | 4.17 | 3.55 |
| | 2.0 | 0.25 | 4.99 | 4.95 | 6.12 | 5.35 | 2.37 | 3.88 | 4.22 | 3.49 |
| | | 0.50 | 4.97 | 5.10 | 6.17 | 5.41 | 2.60 | 4.10 | 4.26 | 3.65 |
| | | 0.75 | 5.29 | 5.26 | 6.52 | 5.69 | 2.63 | 4.27 | 4.43 | 3.78 |
| | | 1.0 | 5.06 | 5.10 | 6.29 | 5.48 | 2.39 | 4.05 | 4.22 | 3.55 |

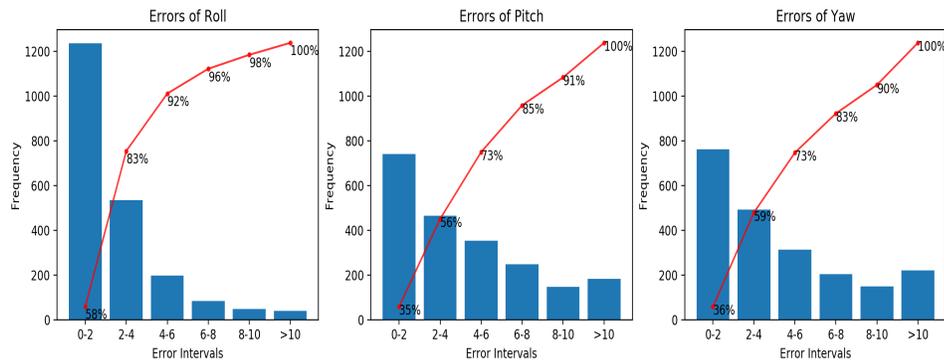


Figure 4.3. Distribution of Euler angle errors on AFLW2000 (in degrees)

method B by adding orthogonality constraint to the regression module to improve the result by 7% on AFLW2000. Method C uses both classification module and regression module and obtain even better results of 4.11 and 4.14 mean average errors on two datasets. Our model shows the best performance on both AFLW2000 and BIWI datasets when having all the modules and constraints in a single network.

Table 4.4. Ablation study for different training methods. Results are evaluated on both AFLW2000 and BIWI datasets (Model parameters: ResNet50, 60 bins, $\alpha=1$, $\beta=0.07$).

| method | | | AFLW2000 | | | | BIWI | | | |
|----------------|---------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| classification | orthogonality | regression | roll | pitch | yaw | MAE | roll | pitch | yaw | MAE |
| | | ✓ | 3.30 | 5.06 | 5.51 | 4.62 | 4.88 | 6.48 | 3.97 | 5.11 |
| | ✓ | ✓ | 2.89 | 4.60 | 5.29 | 4.26 | 5.16 | 6.69 | 3.37 | 5.07 |
| ✓ | | ✓ | 2.85 | 4.42 | 5.04 | 4.11 | 3.95 | 4.74 | 3.772 | 4.14 |
| ✓ | ✓ | ✓ | 2.61 | 4.26 | 4.70 | 3.86 | 3.67 | 4.99 | 3.27 | 3.97 |

4.6 Evaluation on VR data

We study the effect of our proposed method on VR dataset. Different from previous datasets mentioned above where they provide us with accurate face bounding boxes so that we can crop the face area and remove the background noise. Here, existing tools can not generate the object bounding boxes for us. The neural network takes the entire image as input and resize it to certain width and height. This is one of the potential limitations in our dataset. In this section, we only show the experiment results of our proposed TriNet on VR dataset. We adopt MobileNetv2 as the basenet prior to three subnets. Since we don't compare with other methods, we measure the angle difference between our predicted vectors and ground-truth vectors. We follow the data splitting convention mentioned in section 4.1.

Table 4.5 shows the angle errors on both training and testing set. As we can see, our method has better performance on predicting front vector on both training and testing stage. Furthermore, we display the accuracy results in Figure 4.4, which shows the accuracy at different thresholds. A pose is correctly predicted if the absolute angle error between the predicted vector and the ground-truth is lower or equal than the threshold

presented. At the threshold of 5, about 95% of the testing data are correctly predicted. The accuracy rate achieves 100% when we set the restriction to 10 degrees. For front vector, the testing results show that only by setting the threshold to 5 degrees can we get the prediction of all the testing samples correct.

Table 4.5. Average angle errors on training and testing set. Model parameters: MobileNetV2, 60 bins, $\alpha = 1$, $\beta = 0.07$

| | train(degrees) | test(degrees) |
|--------------|----------------|---------------|
| front vector | 1.779 | 2.360 |
| right vector | 2.561 | 2.770 |

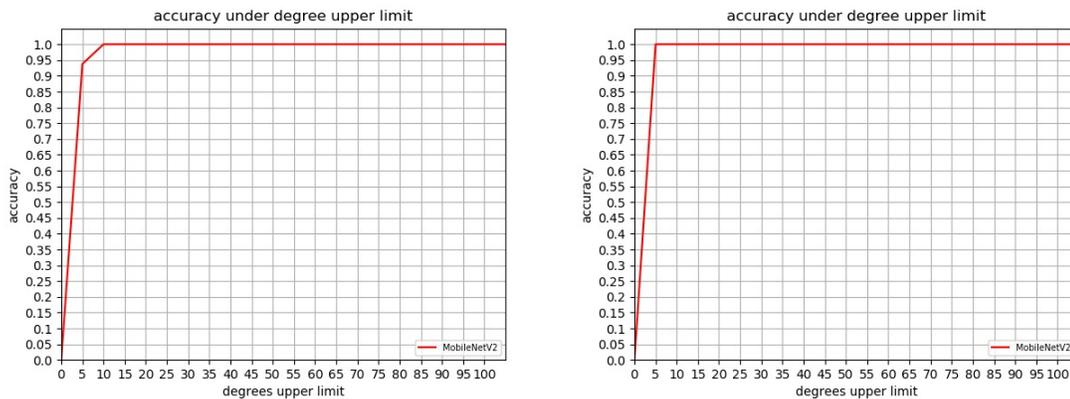


Figure 4.4. Testing accuracy on VR dataset using MobileNetV2 as backbone. Right vector(left) and front vector(right).

We build a visualization tool utilizing OpenGL(Shreiner, Group, et al., 2009) as shown in Fig 4.5. On the left are three frames we extract from a video clip and on the right are the corresponding pose animations. The overall workflow can be described as:

1. A bottle mesh is imported into the OpenGL environment for visualization.
2. We adjust the position of the bottle mesh to match the initial position of the physical bottle object.
3. Based on the output from neural network and refinement, we set the mesh to certain rotation position.



Figure 4.5. Video clip(left) and visualization animation(right).

CHAPTER 5. SUMMARY

The main focus of this study is to present an alternative solution for addressing the problem of pose estimation from a single RGB image. Such alternative solution works better in terms of the accuracy against widely-used Euler angles and quaternions. The entire paper can be divided into four parts: introduction, related work, methodology and experiments.

In the introduction, we start with the background of our research work, which involves the limitation of Euler angle and quaternions rotation representation for training a neural network. We also propose our solution in the background part. Next, we demonstrate the significance of our study and list four important research questions we need to answer by the end of this work. As a standard part of research process, several limitations and delimitations are mentioned in the introduction. Finally, we give the definition of some key words that appear frequently in this paper for readers' better comprehension.

In the literature review, we first investigate works that introduce the limitations for rotation representation, especially Euler angles and quaternions to support our research idea presented in this paper. Then, we mainly focus on reviewing those previous works that use different methods for addressing the pose estimation task. We conduct the literature review work in the order of timeline of when they were being proposed. We first investigate some classical methods such as key points based and geometry based methods. Then, with the emergence of depth camera, we introduce some methods relying on depth information of objects. After seeing the success of deep neural networks on image based tasks, people start to use CNNs to predict poses from images which achieves better precision results compared with traditional methods. These research works have great implication for us, in that our work is built on deep neural network structure.

In the methodology part, we put forward a new vector-based method to represent rotation, which avoids the discontinuity problem of Euler angles and quaternions. A new fine-grained CNN model with multi-loss followed by a refinement step is proposed for head pose estimation. We also devise a new pose data collection pipeline to expand our dataset to have multiple objects.

In the experiment, We test the performance of our proposed method on several public head pose datasets including AFLW, BIWI and AFW and achieve state-of-the-art results compared to other methods. We reduce the mean average errors to 3.86 and 3.97 on AFLW2000 and BIWI dataset. On AFW, our method achieves 98.7% accuracy which is the best result among all the previous approaches. We apply TriNet to our own dataset and also achieve satisfactory results.

Experiment evidence shows that our proposed method outperforms most of the previous methods, in that our vector based representation is a 9D representation. Compared with Euler angles and quaternions, it is continuous in Euclidean space. The one-to-many correspondence problem then can be solved. We show a deep neural network combined with a multi-loss approach can accurately and robustly estimate the object pose from RGB images. We introduce an orthogonality check for each paring of the predicted vectors as one of the training losses which further improves the prediction performance. However, for the multi-loss approach, how to balance the weight of each loss still remains unclear. In this study, we vary the weight of each loss components to seek for the optimal solution. Another limitation of our proposed method is the final outputs depend on the vector refinement stage. An end-to-end system then can not be established.

REFERENCES

- Bai, H., Gao, L., & Billinghamurst, M. (2017). 6dof input for hololens using vive controller. In *Siggraph asia 2017 mobile graphics & interactive applications* (pp. 1–1).
- Belhumeur, P. N., Jacobs, D. W., Kriegman, D. J., & Kumar, N. (2013). Localizing parts of faces using a consensus of exemplars. *IEEE transactions on pattern analysis and machine intelligence*, 35(12), 2930–2940.
- Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., & Rother, C. (2014). Learning 6d object pose estimation using 3d object coordinates. In *European conference on computer vision* (pp. 536–551).
- Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- Caserman, P., Garcia-Agundez, A., Konrad, R., Göbel, S., & Steinmetz, R. (2019). Real-time body tracking in virtual reality using a vive tracker. *Virtual Reality*, 23(2), 155–168.
- Chen, J., Wu, J., Richter, K., Konrad, J., & Ishwar, P. (2016). Estimating head pose orientation using extremely low resolution images. In *2016 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)* (pp. 65–68).
- Choi, C., & Christensen, H. I. (2016). Rgb-d object pose estimation in unstructured environments. *Robotics and Autonomous Systems*, 75, 595–613.
- Chong, E., Ruiz, N., Wang, Y., Zhang, Y., Rozga, A., & Rehg, J. M. (2018). Connecting gaze, scene, and attention: Generalized attention estimation via joint modeling of gaze and scene saliency. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 383–398).

- Fanelli, G., Dantone, M., Gall, J., Fossati, A., & Van Gool, L. (2013). Random forests for real time 3d face analysis. *International journal of computer vision*, *101*(3), 437–458.
- Fanelli, G., Gall, J., & Van Gool, L. (2011). Real time head pose estimation with random regression forests. In *Cvpr 2011* (pp. 617–624).
- Fua, P., & Lepetit, V. (2005). Monocular model-based 3d tracking of rigid objects. *Comput. Graph. Vis*, *1*(1), 1–89.
- Gee, A., & Cipolla, R. (1994). Determining the gaze of faces in images. *Image and Vision Computing*, *12*(10), 639–647.
- Gourier, N., Maisonnasse, J., Hall, D., & Crowley, J. L. (2006). Head pose estimation on low resolution images. In *International evaluation workshop on classification of events, activities and relationships* (pp. 270–280).
- Gu, J., Yang, X., De Mello, S., & Kautz, J. (2017). Dynamic facial analysis: From bayesian filtering to recurrent neural network. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1548–1557).
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).
- Hsu, H.-W., Wu, T.-Y., Wan, S., Wong, W. H., & Lee, C.-Y. (2018). Quatnet: Quaternion-based head pose estimation with multiregression loss. *IEEE Transactions on Multimedia*, *21*(4), 1035–1046.

- Jackson, A. S., Bulat, A., Argyriou, V., & Tzimiropoulos, G. (2017). Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. In *Proceedings of the ieee international conference on computer vision* (pp. 1031–1039).
- Jourabloo, A., & Liu, X. (2016). Large-pose face alignment via cnn-based dense 3d model fitting. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 4188–4196).
- Kazemi, V., & Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1867–1874).
- Kehl, W., Manhardt, F., Tombari, F., Ilic, S., & Navab, N. (2017). Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *Proceedings of the ieee international conference on computer vision* (pp. 1521–1529).
- Kendall, A., Grimes, M., & Cipolla, R. (2015). Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the ieee international conference on computer vision* (pp. 2938–2946).
- Kumar, A., Alavi, A., & Chellappa, R. (2017). Kepler: Keypoint and pose estimation of unconstrained faces by learning efficient h-cnn regressors. In *2017 12th ieee international conference on automatic face & gesture recognition (fg 2017)* (pp. 258–265).
- Lai, K., Bo, L., Ren, X., & Fox, D. (2011). A scalable tree-based approach for joint object and pose recognition. In *Twenty-fifth aai conference on artificial intelligence*.
- Li, C., Bai, J., & Hager, G. D. (2018). A unified framework for multi-view multi-class object pose estimation. In *Proceedings of the european conference on computer vision (eccv)* (pp. 254–269).

- Li, R., Danielsen, C. M., & Taskiran, C. M. (2008, August 12). *Apparatus and methods for head pose estimation and head gesture detection*. Google Patents. (US Patent 7,412,077)
- Li, Y., Gu, L., & Kanade, T. (2011). Robustly aligning a shape model and its application to car alignment of unknown pose. *IEEE transactions on pattern analysis and machine intelligence*, 33(9), 1860–1876.
- Liu, X., Liang, W., Wang, Y., Li, S., & Pei, M. (2016). 3d head pose estimation with convolutional neural network trained on synthetic images. In *2016 IEEE International Conference on Image Processing (ICIP)* (pp. 1289–1293).
- Lowe, D. G. (1991). Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis & Machine Intelligence*(5), 441–450.
- Luckett, E. (2018). *A quantitative evaluation of the htc vive for virtual reality research*. Unpublished doctoral dissertation, The University of Mississippi.
- Lv, J., Shao, X., Xing, J., Cheng, C., & Zhou, X. (2017). A deep regression architecture with two-stage re-initialization for high performance facial landmark detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3317–3326).
- Martin Koestinger, P. M. R., Paul Wohlhart, & Bischof, H. (2011). Annotated Facial Landmarks in the Wild: A Large-scale, Real-world Database for Facial Landmark Localization. In *Proc. First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*.
- Michel, F., Kirillov, A., Brachmann, E., Krull, A., Gumhold, S., Savchynskyy, B., & Rother, C. (2017). Global hypothesis generation for 6d object pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 462–471).

- Murphy-Chutorian, E., Doshi, A., & Trivedi, M. M. (2007). Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation. In *2007 IEEE Intelligent Transportation Systems Conference* (pp. 709–714).
- Murphy-Chutorian, E., & Trivedi, M. M. (2008). Head pose estimation in computer vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, *31*(4), 607–626.
- Ng, J., & Gong, S. (2002). Composite support vector machines for detection of faces across views and pose estimation. *Image and Vision Computing*, *20*(5-6), 359–368.
- Osuna, E., Freund, R., & Girosit, F. (1997). Training support vector machines: an application to face detection. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 130–136).
- Patacchiola, M., & Cangelosi, A. (2017). Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods. *Pattern Recognition*, *71*, 132–143.
- Peer, A., Ullrich, P., & Ponto, K. (2018). Vive tracking alignment and correction made easy. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (pp. 653–654).
- Rad, M., & Lepetit, V. (2017). Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 3828–3836).
- Raytchev, B., Yoda, I., & Sakaue, K. (2004). Head pose estimation by nonlinear manifold learning. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* (Vol. 4, pp. 462–466).

- Rennie, C., Shome, R., Bekris, K. E., & De Souza, A. F. (2016). A dataset for improved rgb-d-based object detection and pose estimation for warehouse pick-and-place. *IEEE Robotics and Automation Letters*, *1*(2), 1179–1185.
- Rowley, H. A., Baluja, S., & Kanade, T. (1998). Neural network-based face detection. *IEEE Transactions on pattern analysis and machine intelligence*, *20*(1), 23–38.
- Ruiz, N., Chong, E., & Rehg, J. M. (2018). Fine-grained head pose estimation without keypoints. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 2074–2083).
- Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., & Pantic, M. (2013). 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Proceedings of the IEEE international conference on computer vision workshops* (pp. 397–403).
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510–4520).
- Saxena, A., Driemeyer, J., & Ng, A. Y. (2009). Learning 3-d object orientation from images. In *2009 IEEE international conference on robotics and automation* (pp. 794–800).
- Sherrah, J., Gong, S., & Ong, E.-J. (1999). Understanding pose discrimination in similarity space. In *Bmvc* (pp. 1–10).
- Sherrah, J., Gong, S., & Ong, E.-J. (2001). Face distributions in similarity space under varying head pose. *Image and Vision Computing*, *19*(12), 807–819.
- Shreiner, D., Group, B. T. K. O. A. W., et al. (2009). *OpenGL programming guide: the official guide to learning OpenGL, versions 3.0 and 3.1*. Pearson Education.

- Spitzley, K. A., & Karduna, A. R. (2019). Feasibility of using a fully immersive virtual reality system for kinematic data collection. *Journal of biomechanics*, *87*, 172–176.
- Storer, M., Urschler, M., & Bischof, H. (2009). 3d-mam: 3d morphable appearance model for efficient fine head pose estimation from still images. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops* (pp. 192–199).
- Sun, Y., Wang, X., & Tang, X. (2013). Deep convolutional network cascade for facial point detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3476–3483).
- Tekin, B., Sinha, S. N., & Fua, P. (2018). Real-time seamless single shot 6d object pose prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 292–301).
- Valle, R., Buenaposada, J. M., Valdés, A., & Baumela, L. (2019). Face alignment using a 3d deeply-initialized ensemble of regression trees. *Computer Vision and Image Understanding*, *189*, 102846.
- Xiang, Y., Schmidt, T., Narayanan, V., & Fox, D. (2017). Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*.
- Yang, S., Luo, P., Loy, C. C., & Tang, X. (2016). Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yang, T.-Y., Chen, Y.-T., Lin, Y.-Y., & Chuang, Y.-Y. (2019). Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1087–1096).

- Yu, Y., Mora, K. A. F., & Odobez, J.-M. (2017). Robust and accurate 3d head pose estimation through 3dmm and online head model reconstruction. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)* (pp. 711–718).
- Zach, C., Penate-Sanchez, A., & Pham, M.-T. (2015). A dynamic programming approach for fast and robust object pose recognition from range images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 196–203).
- Zakharov, S., Shugurov, I., & Ilic, S. (2019). Dpod: 6d pose object detector and refiner. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1941–1950).
- Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10), 1499–1503.
- Zhang, X., Sugano, Y., Fritz, M., & Bulling, A. (2015). Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4511–4520).
- Zhou, E., Fan, H., Cao, Z., Jiang, Y., & Yin, Q. (2013). Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *Proceedings of the IEEE International Conference on Computer Vision Workshops* (pp. 386–391).
- Zhou, Y., Barnes, C., Lu, J., Yang, J., & Li, H. (2019). On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5745–5753).
- Zhu, X., Lei, Z., Liu, X., Shi, H., & Li, S. Z. (2016). Face alignment across large poses: A 3d solution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 146–155).

Zhu, X., Lei, Z., Yan, J., Yi, D., & Li, S. Z. (2015). High-fidelity pose and expression normalization for face recognition in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 787–796).

Zhu, X., & Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 2879–2886).