A Vector-based Representation to Enhance Head Pose Estimation

Zhiwen Cao * Zongcheng Chu * Dongfang Liu
Yingjie Chen
Department of Computer Graphics Technology, Purdue University
West Lafayette, 47907, USA
{cao270, chu153, liu2538, victorchen}@purdue.edu

Abstract

This paper proposes to use the three vectors in a rotation matrix as the representation in head pose estimation and develops a new neural network based on the characteristic of such representation. We address two potential issues existed in current head pose estimation works: 1. Public datasets for head pose estimation use either Euler angles or quaternions to annotate data samples. However, both of these annotations have the issue of discontinuity and thus could result in some performance issues in neural network training. 2. Most research works report Mean Absolute Error (MAE) of Euler angles as the measurement of performance. We show that MAE may not reflect the actual behavior especially for the cases of profile views. To solve these two problems, we propose a new annotation method which uses three vectors to describe head poses and a new measurement Mean Absolute Error of Vectors (MAEV) to assess the performance. We also train a new neural network to predict the three vectors with the constraints of orthogonality. Our proposed method achieves state-of-the-art results on both AFLW2000 and BIWI datasets. Experiments show our vector-based annotation method can effectively reduce prediction errors for large pose angles.

1. Introduction

Single image head pose estimation is an important task in computer vision which has drawn a lot of research attention in recent years. So far it mainly relies on facial landmark detection [19, 13, 29, 5]. These approaches show 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 it is difficult to extract key feature points from large poses such as profile views. To solve this issue, a large array of research has been directed to employ Convolutional Neural Network (CNN) based methods to predict head pose directly from a single image. Several public benchmark datasets [18, 31, 37, 8] have been contributed in this area for the purpose of validating the effectiveness of these approaches. Among these approaches, [23, 10, 32, 22] try to address the problem by direct regression of either three Euler angles or quaternions from images using CNN models.

Euler angles	[−80.8° −88.9° 78.0°]	[−19.9° −87.4° 8.0°]	[87.7° −89.3° −87.9°]
Quaternions	[−0.70 −0.12 0.69 0.14]	[−0.17 −0.67 0.17 0.70]	[0.70 −0.02 −0.70 0.03]
Vectors (unit)	[0.004 −0.222 −0.975] [−0.019 0.975 −0.225] [0.999 0.019 −0.001]	[0.004 −0.206 −0.978] [−0.063 0.978 −0.206] [0.999 0.015 0.042]	[0.004 −0.003 −0.999] [0.012 0.999 −0.003] [0.999 −0.012 0.001]

Figure 1: Data samples from 300W-LP dataset and their Euler angles, converted quaternions and three-vector annotations. From top to bottom, three vectors are left (red), down (green) and front (blue) vectors respectively.

However, these studies use either Euler angles or quaternions as their 3D rotation representations. Both Euler angles and quaternions have limitations when they are used to represent rotations. For example, when using Euler angles, the rotation order must be defined 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 [6]. 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 [26]. In addition, the results from [36] show that any representation of rotation with four or fewer dimensions is discontinuous. These findings indicate that it is inappropriate to use Euler angles or quaternions to annotate head poses.

This issue can be illustrated by several samples. Fig. 1 shows three images with similar large pose angles from the

*The two first authors made equal contributions.

300W-LP dataset. However, neither the Euler angle nor quaternion annotations between any two of these show similarity. This leads to two major problems:

(1) It makes training a neural network difficult since the network learns to regress different outputs from the same visual patterns.

(2) It makes mean absolute error (MAE) of Euler angles a problematic measurement of performance. If the second image is included in the training samples while the first image is the testing case, the network's Euler angle prediction will be close to $[-19.9°, -87.4°, 8.0°]$ since it learns an image-to-pose relationship from the second one. However, if we compare this prediction with the ground truth $[-80.8°, -88.9°, 78.0°]$, the MAE will give the result of $44.2°$. This is a large error which cannot reflect the actual model performance.

These issues can be solved by the introduction of three pose vectors. As shown in Fig. 2, head pose can be depicted by a left vector (red), a down vector (green) and a front vector (blue). Using vectors to represent head pose has the following advantages:

(1) It makes the annotations consistent, as we show in the third row in Fig. 1. The vector representations of three images are close to each other.

(2) Instead of using MAE of Euler angles, we put forward a new measurement which calculates MAE of angles between the vectors (MAEV) that our model predicts and corresponding ground truth. Continue the example above, if the network learns to predict vectors, the MAEV is $(2.663° + 1.504° + 2.579°)/3 = 2.249°$. This is an accurate measure of performance.

Specifically, our work contributes at:

(1) We illustrate that Euler angle annotation has issues of discontinuity and that Euler angle based MAE cannot fully measure the actual performance especially for face profile image.

(2) We instead present MAEV metric that measures angles between vectors, which is a more reliable indicator for the evaluation of pose estimation results.

(3) Based on the vector representation and MAEV, we proposed a deep network pipeline with vectors' orthogonal constraints. To our knowledge, this is the first attempt to formulate head pose estimation problem with vector representation meanwhile consider the vectors' orthogonal constraints in the deep network pipeline.

2. Related Work

2.1. Landmark-Based Approaches

These approaches typically detect key landmarks from images first and then estimate the poses by solving the correspondences between 2d and 3d feature points. [3] proposes an algorithm called Cascaded Pose Regression (CPR) which progressively refines a rough initial guess by different regressors in each refinement. [2] learns a vectorial regression function from training data that it uses to obtain a set of facial landmarks from the image with this function.

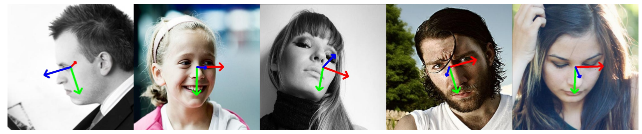

Figure 2: Sample results of head pose estimation by using proposed method.

With the advent of CNN, numerous CNN-based methods have been designed and achieve superior performances than their predecessors. [28] puts forward an approach which draws on a three-level convolutional network to estimate the positions of facial landmarks. [37] proposes a new cascaded neural network called 3D Dense Face Alignment (3DDFA) which fits a dense morphable 3D face model to the image. They also propose a method to synthesize large-scale training samples in profile views for data labeling. Based on 300W dataset [24], they create the synthesized 300W-LP dataset which includes 122,450 samples. This has become a widely accepted benchmark dataset. [9] makes a step further by proposing a new optimization strategy to regress 3DMM parameters. Their network model simply predicts 9 elements and constructs a rotation matrix from them. As a result, this can never guarantee it to be a rotation matrix.

Some methods treat head pose estimation as an auxiliary task. They perform various facial related tasks jointly with CNN. [21] proposes Hyperface which uses a single CNN model to perform face detection, pose estimation, feature localization and gender recognition simultaneously. [13] proposes a H-CNN (Heatmap-CNN) which refines the locations of the facial keypoints iteratively and provides the pose information in Euler angles as a by-product.

These methods rely heavily on the quality of landmark detection. If it fails to detect the landmark accurately, a large error will be introduced.

2.2. Landmark-Free Approaches

The latest state-of-the-art landmark-free approaches explore the research boundary and improve the results by a significant margin. [23] puts forward a CNN combined with multi-losses. It predicts three Euler angles directly from a single image and outperforms all the prior landmark based methods. [10] further presents their quaternion based approach which avoids the gimbal lock issue lying in Euler angles. [32] proposes a CNN model using a stage-wise regression mechanism. They also adopt an attention mechanism combined with a feature aggregation module to group global spacial features. [15] treats pose estimation as a label distribution learning problem. They associate a Gaussian label distribution instead of a single label with each image and train a network which is similar to Hopenet [23].

2.3. 6D Object Pose Estimation

6D object pose estimation from RGB images includes estimation of 3D orientation and 3D location. The task of orientation estimation resembles our head pose estimation one. The approaches can be divided into two categories: [20, 33, 27] first estimate the object mask to determine its location in the image, then build the correspondence between the image pixels and the available 3D models. After that, The 6d pose can be solved through PnP algorithm [14]. The other type of methods such as [30, 16, 7] use network to predict orientation directly. However, they use either axis-angle or quaternion as their representations of rotation and none of them notice the problem of discontinuity.

3. Method

In this section, we first present a thorough discussion on our vector-based representation and how we formulate the problem (Sec. 3.1). Then, we give an overview of the our network structure (Sec. 3.2). Prediction module implementation is described in Section 3.3. A multi-loss training strategy is then introduced in Section 3.4. Finally, by means of Singular Value Decomposition (SVD), we obtain three orthonormal vectors (Sec. 3.5).

3.1. Representation of Rotation

There are various ways to represent a rotation in a 3D world. Euler angle, quaternion, axis-angle and lie algebra. They describe the rotation in a compact form with at most 4 dimensions. However, [36] shows that it needs at least 5 dimensions of information to achieve a continuous representation of rotations in 3D space which means all the above representation methods will have the same issue of ambiguity as demonstrated in section 1. This makes rotation matrix a good alternative. A 3d rotation matrix has 9 elements and can be described as orthogonal matrices with determinant equals to $+1$. The set of all the rotation matrices forms a continuous special orthogonal group $SO(3)$. When it is used to describe rotation, it doe not have problem of discontinuity or ambiguity.

The question left is what metric we should adopt to measure the closeness of two rotation matrices. A straightforward way is to measure the Frobenius norm of two rotation matrices, i.e. the square root of the sum of squares of differences of all 9 elements. If we define the left, down and front vectors at the reference starting point to be $\boldsymbol{v}_1 = \left[1, 0, 0\right]^T$, $\boldsymbol{v_2} = \left[0, 1, 0\right]^T$ and $\boldsymbol{v_3} = \left[0, 0, 1\right]^T$ respectively. After applying a rotation matrix $\boldsymbol{R}_{3\times 3} = \left[\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3\right]$ where $\boldsymbol{r}_i$ denotes the i^{th} column vector in $\boldsymbol{R}$, the three vectors then become $\boldsymbol{v}'_1 = \boldsymbol{R}\boldsymbol{v_1} = \boldsymbol{r}_1$, $\boldsymbol{v}'_2 = \boldsymbol{R}\boldsymbol{v_2} = \boldsymbol{r}_2$ and $\boldsymbol{v}'_3 = \boldsymbol{R}\boldsymbol{v_3} = \boldsymbol{r}_3$. The equations show that three vectors of head pose is in essence equivalent to the three columns of rotation matrices. As a result, Frobenius norm is equivalent to $\sqrt{d_1^2 + d_2^2 + d_3^2}$ in Fig. 5.

Even though Frobenius norm is an accurate measurement, it is hard for we human beings to perceive the difference of rotation angles through the distance between end-points of pose vectors. Therefore, we put forward a new metric which is more intuitive: the mean absolute error of vectors (MAEV). For each vector, we compute absolute error between the ground truth and predicted one, then we obtain MAEV by calculating the mean value of three errors.

The problem of head pose estimation thus can be defined as: given a set of N training images $\boldsymbol{X} = \{x^{(1)}, x^{(2)}, \cdots, x^{(N)}\}$, find a mapping function F such that estimates $\hat{\boldsymbol{R}}^{(i)} = F(x^{(i)})$ where $\hat{\boldsymbol{R}}^{(i)} = \left[\hat{\boldsymbol{r}}_{\boldsymbol{1}}{}^{(i)}, \hat{\boldsymbol{r}}_{\boldsymbol{2}}{}^{(i)}, \hat{\boldsymbol{r}}_{\boldsymbol{3}}{}^{(i)}\right]$ that matches the ground truth rotation matrix $\boldsymbol{R}$ as close as possible. We try to find an optimal F for all $\boldsymbol{X}$ by minimizing the sum of squared L_2 norm between the predicted and ground truth vectors.

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left\|\boldsymbol{r}_{\boldsymbol{1}}^{(\boldsymbol{i})} - \hat{\boldsymbol{r}}_{\boldsymbol{1}}^{(\boldsymbol{i})}\right\|_2^2 + \left\|\boldsymbol{r}_{\boldsymbol{2}}^{(\boldsymbol{i})} - \hat{\boldsymbol{r}}_{\boldsymbol{2}}^{(\boldsymbol{i})}\right\|_2^2 + \left\|\boldsymbol{r}_{\boldsymbol{3}}^{(\boldsymbol{i})} - \hat{\boldsymbol{r}}_{\boldsymbol{3}}^{(\boldsymbol{i})}\right\|_2^2 \quad (1)$$

3.2. TriNet Overview

Rotation matrix is 9-D dimensional representation which requires the network to predict 9 elements. There is no off-the-shelf network model that we can adopt to perform this task, so we design our TriNet shown as Fig. 3. TriNet is composed of one backbone and three head branches. Each head follows the coarse-to-fine strategy, constitutes a feature mapping and prediction module and is responsible for predicting one vector alone. Ideally, three vectors should be perpendicular to each other, so we further introduce an orthogonal loss function which punishes the model if the predicted ones are not orthogonal.

An input image with fixed size goes through a backbone network (ResNet50 in Fig. 3). We define S stages and at each stage s, a feature map is extracted from the output of an intermediate layer of the backbone network. These are considered as candidate features and fed into the feature grouping module. For feature grouping component, we follow the same implementation as FSA-Net [32]. Since the grouping module requires uniform shape ($w \times h \times c$) of input features, we apply average pooling to reduce the feature map size to $w \times h$ and use c 1×1 convolution operations to transform the feature channels into c. The feature grouping module outputs 3 c'-dimensional vectors.

We then feed them to the prediction module to regress one pose vector. Since three head branches share the identical structure, the other two pose vectors can be obtained in the same way by going through different head branches.

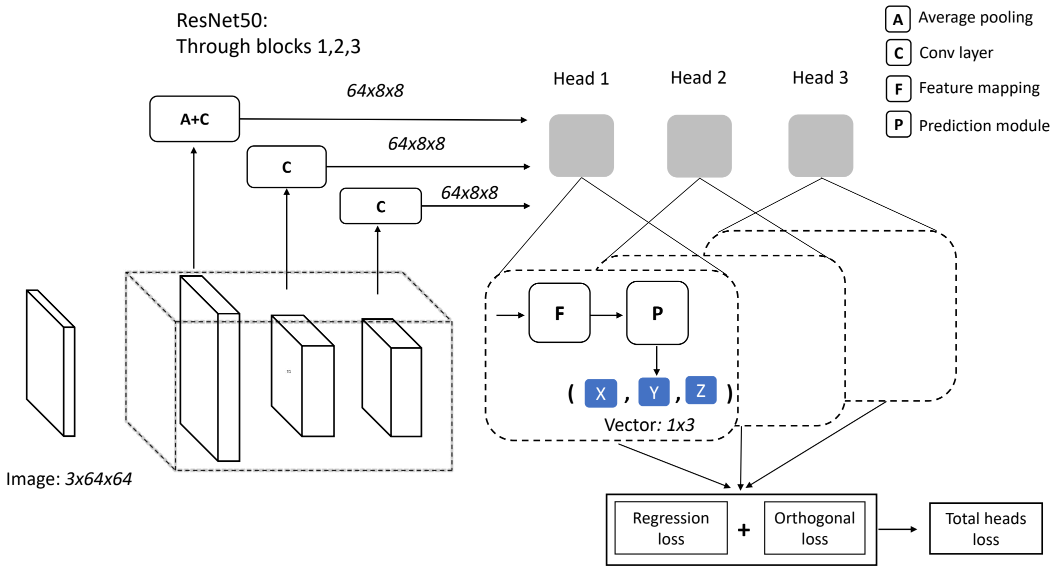

Figure 3: Overview of the network.

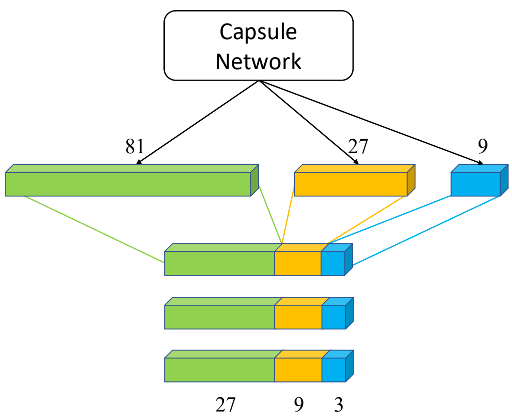

Figure 4: Details of the prediction module.

3.3. Prediction Module

The prediction module follows the strategy of coarse-to-fine multi-stage regression. Features extracted from shallow layers are responsible for performing coarse predictions. As the network goes deeper, the high level features become more informative and can be used for fine-grained and more accurate predictions. Since each component of a unit vector is within the range of $[-1, 1]$, for each stage, we divide the range into different numbers of intervals. The deeper the layer is, the more intervals the range $[-1, 1]$ will be divided into. The prediction module performs the estimation by taking the average of the expectation values from all S stages together:

$$\hat{y} = \frac{1}{S} \sum_{s=1}^{S} \sum_{i=1}^{n^{(s)}} p_i^{(s)} \cdot q_i^{(s)} \tag{2}$$

where $n^{(s)}$ is the number of intervals at stage s, $p_i^{(s)}$ is the probability that the element is in the i^{th} interval and $q_i^{(s)}$ is the mean value of the i^{th} interval.

3.4. Training Objective

The training objective involves multiple losses: regression loss $\mathcal{L}_{reg}$ and the orthogonal loss $\mathcal{L}_{ortho}$ which measures the orthogonality between each pair of the predicted vectors. The overall objective loss is the weighted sum of two losses:

$$\mathcal{L} = \mathcal{L}_{reg}(\boldsymbol{v}_i, \hat{\boldsymbol{v}}_i) + \alpha \mathcal{L}_{ortho}(\hat{\boldsymbol{v}}_i, \hat{\boldsymbol{v}}_j) \tag{3}$$

where $\hat{\boldsymbol{v}}_i$ and $\boldsymbol{v}_i$ are the i^{th} predicted and ground truth vectors respectively. The weighted term α is set to a small number whose range is between $[0.1, 0.5]$. It best setting is found through experiments. Each loss term is shown as follows:

$$\mathcal{L}_{reg} = \sum_{i=1}^{3} \text{mse}(\boldsymbol{v}_i, \hat{\boldsymbol{v}}_i) \tag{4}$$

$$\mathcal{L}_{ortho} = \sum_{i \neq j} \text{mse}(\hat{\boldsymbol{v}}_i \hat{\boldsymbol{v}}_j, 0) \text{ where } i, j = 1, 2, 3 \tag{5}$$

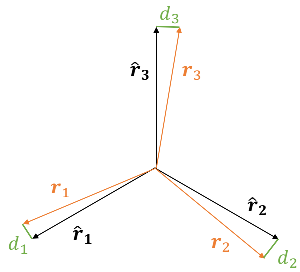

Figure 5: Given a matrix $\boldsymbol{R} = \big(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3\big)$, find the closest rotation matrix $\hat{\boldsymbol{R}} = \big(\hat{\boldsymbol{r}}_1, \hat{\boldsymbol{r}}_2, \hat{\boldsymbol{r}}_3\big)$. The Frobenius norm of the difference between $\boldsymbol{R}$ and $\hat{\boldsymbol{R}}$ is equal to $\sqrt{d_1^2 + d_2^2 + d_3^2}$

We adopt mean square error loss function for both regression loss and orthogonal loss.

3.5. Vector Refinement

Even though we impose orthogonal constraints $\mathcal{L}_{ortho}$ in the loss function, the three vectors that TriNet predicts are still not perpendicular to each other. Therefore, it is necessary to select three orthogonal vectors to match the predicted vectors as close as possible.

This problem can be stated as: Given a noisy predicted matrix $\boldsymbol{R}$, find the closest rotation matrix $\hat{\boldsymbol{R}}$ and the measure of closeness needs to have physical meaning. A naive way to find a rotation matrix from a noisy matrix is applying the Gram–Schmidt process to either its rows or columns. Its simple geometric interpretation makes it very popular, however, the result is rather arbitrary because this method depends on which two rows or columns of $\boldsymbol{R}$ are selected.

This paper adopts the measure of Euclidean or Frobenius norm of $\boldsymbol{R} - \hat{\boldsymbol{R}}$. It can be expressed by the following formula:

$$\begin{gathered} \min ||\hat{\boldsymbol{R}} - \boldsymbol{R}||_F \\ \text{subject to } \hat{\boldsymbol{R}}^T \hat{\boldsymbol{R}} = \boldsymbol{I} \text{ and } \det\hat{\boldsymbol{R}} = +1 \end{gathered} \tag{6}$$

The reasons for choosing Frobenius norm are as follows:

(1) It has a simple geometric interpretation (see Fig. 5).

(2) The solution is unique and can be obtained by a closed-form formula [25].

[17] shows that given a matrix $\boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, the optimal solution can be achieved by $\hat{\boldsymbol{R}} = \boldsymbol{U}\boldsymbol{V}^T$. This method does not guarantee that $\det(\hat{\boldsymbol{R}}) = +1$. If a highly noisy matrix $\boldsymbol{R}$ is given, $\det(\hat{\boldsymbol{R}}) = -1$ may happen. If this is the case, the closest rotation matrix can be obtained by

$$\hat{\boldsymbol{R}} = \boldsymbol{U}\text{diag}(1, 1, -1)\boldsymbol{V}^T \tag{7}$$

4. Experiments

4.1. Implementation Details

We implement our proposed network using Pytorch. We follow the data augmentation strategies from [32] and apply uniformly on the competing methods. We train the network using Adam optimizer with an initial learning rate of 0.0001 over 90 epochs. The learning rate decay parameter is set to be 0.1 for every 30 epochs.

4.2. Datasets and Evaluation

Our experiments are based on three popular public benchmark datasets: 300W-LP [37], AFLW2000 [38], and BIWI [4] datasets.

300W-LP The 300W-LP dataset [37] is expanded from 300W dataset [24] which is composed of several standardized datasets, including AFW [39], HELEN [35], IBUG [24] and LFPW [1]. By means of face profiling, this dataset generates 122,450 synthesized images based on around 4,000 pictures from the 300W dataset.

AFLW2000 The AFLW2000 [38] dataset contains 2,000 images which are the first 2000 images the AFLW dataset [18]. This dataset possesses a wide range of varieties in facial appearances and background settings

BIWI The BIWI dataset [4] contains 15,678 pictures of 20 participants in an indoor environment. Since the dataset does not provide bounding boxes of human heads, we use MTCNN [34] to detect human faces and loosely crop the area around the face to obtain face bounding boxes results.

In order to compare to the state-of-the-art methods, we follow the same training and testing setting as mentioned in Hopenet [23] and FSA-Net [32]. Notice that we also filter out test samples with Euler angles that are not in the range between $-99°$ and $99°$ to keep consistent with the strategies used by Hopenet and FSA-Net. We implement our experiments in two scenarios:

(1) We train the models on 300W-LP and test on two other datasets: AFLW2000 and BIWI.

(2) We apply a 3-fold cross validation on BIWI dataset and report the mean validation errors. We split the dataset into 3 groups and ensure that the images of one person should appear in the same group. Since there are 24 videos in the BIWI dataset, each group contains 8 videos and in a round we have 16 videos for training and 8 for testing.

For all the experiments above, we report both the MAE of Euler angles and MAEV as results.

4.3. Comparison to State-of-the-art Methods

We compare our proposed TriNet with other state-of-the-art methods on public benchmark datasets. To make a fair comparison, we rerun the open-sourced models and ours under the same experiment environment and measure the results by both MAE and MAEV. For those which are not

Method	Euler angles errors				Vector errors			
	roll	pitch	yaw	MAE	left	down	front	MAEV
3DDFA[37]	28.432	27.085	4.710	20.076	(30.570	39.054	18.517	29.380)
Dlib[12]	22.829	11.250	8.494	14.191	(26.559	28.511	14.311	23.127)
Hopenet[23]	6.132	7.120	5.312	6.188	(7.073	5.978	7.502	6.851)
FSA-Net[32]	4.776	6.341	4.963	5.360	(6.753	6.215	7.345	6.771)
Quatnet[10]*	**3.920**	**5.615**	**3.973**	**4.503**	-	-	-	-
HPE[11] *	4.800	6.180	4.870	5.280	-	-	-	-
TriNet	(4.042	5.767	4.198	4.669)	**5.782**	**5.666**	**6.519**	**5.989**

Table 1: Mean absolute errors of Euler angles and vectors on AFLW2000. All trained on 300W-LP. Values in () are converted from the other side. Methods with $*$ are not open source. Their results are claims from authors.

Method	Euler angles errors				Vector errors			
	roll	pitch	yaw	MAE	left	down	front	MAEV
3DDFA[12]	13.224	41.899	5.497	20.207	(23.306	45.001	35.117	34.475)
Dlib[12]	19.564	12.996	11.864	14.808	(24.842	21.702	14.301	20.282)
Hopenet[23]	3.719	5.885	6.007	5.204	(7.650	6.728	8.681	7.687)
FSA-Net[32]	3.069	5.209	4.560	4.280	(6.033	5.959	7.218	6.403)
Quatnet[10]*	**2.936**	5.492	4.010	4.146	-	-	-	-
HPE[11] *	3.120	5.180	4.570	4.290	-	-	-	-
TriNet	(4.112	**4.758**	**3.046**	**3.972**)	**5.565**	**5.457**	**6.571**	**5.864**

Table 2: Mean absolute errors of Euler angles and vectors on BIWI. All trained on 300W-LP. Values in () are converted from the other side. Methods with $*$ are not open source. Their results are claims from authors.

Method	Euler angles				Vectors			
	roll	pitch	yaw	MAE	left	down	front	MAEV
Hopenet[23]	4.334	4.420	4.094	4.283	(6.465	6.272	6.268	6.335)
FSA-Net[32]	4.056	4.558	3.155	3.922	(5.854	6.189	5.440	5.828)
TriNet	(**2.928**	**3.035**	**2.440**	**2.801**)	**4.067**	**4.140**	**3.976**	**4.061**

Table 3: Mean absolute errors of Euler angles and vectors on BIWI. 70% of the data is used for training and the remaining 30% is for testing. Values in () are converted from the other side.

open sourced, we cite their MAE results claimed by the authors in our tables for reference.

Facial landmark based approach 3DDFA [37] 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. Hopenet [23] proposes a fine-grained structure by combining classification loss and regression loss to predict the head pose in a more robust way. Quatnet [10] uses quaternions labeling data for training the model to avoid the ambiguity of Euler angle representation. FSA-Net [32] proposes a network which combines a stage-wise regression scheme and a feature grouping module for learning aggregated the spatial features. [11] proposes to use two stage method which treats classification and regression separately and averages top-k outputs as pose regression subtask.

4.4. Experiment Results

Table 1 and 2 show the results of our proposed TriNet and other methods tested on AFLW2000 and BIWI datasets respectively. All of them are trained on the 300W-LP dataset. Since TriNet predicts three orthonormal vectors, its Euler angle results are obtained through the conversion from the rotation matrix constructed of these three vectors.

As the tables demonstrate, deep learning based landmark-free approaches (FSA-Net, Hopenet and TriNet) outperform landmark based methods (3DDFA and Dlib) on both AFLW2000 and BIWI datasets. In Table 1, we can find that if measured by MAE, FSA-Net surpasses the Hopenet by a large margin. However, their MAEV results are close. Even though Quatnet achieves the best MAE results, we are unable to replicate the MAEV results since it is not open-source. Meanwhile, as shown in Table 2, Our proposed method achieves the best result under both MAE and

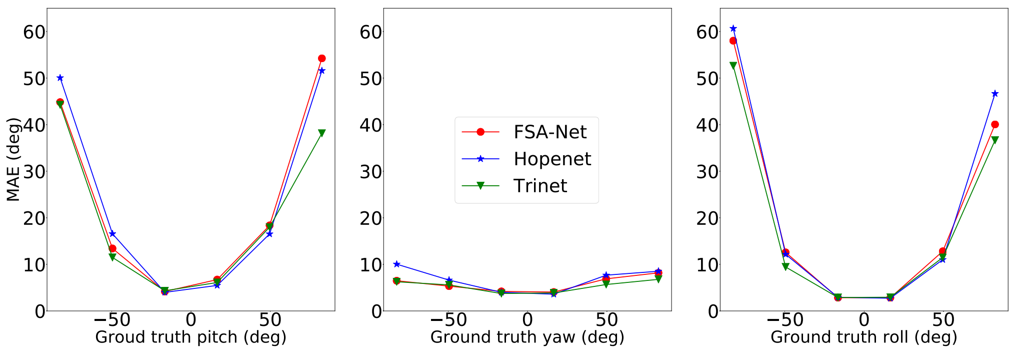

Figure 6: MAE on AFLW2000 using landmark-free methods. All trained on 300W-LP.

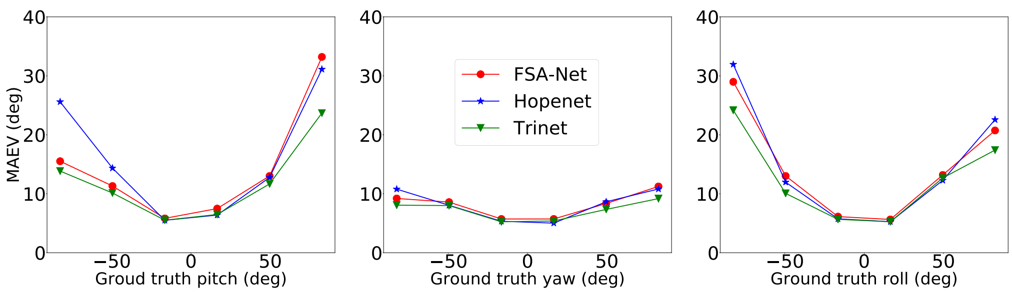

Figure 7: MAEV on AFLW2000 using landmark-free methods. All trained on 300W-LP.

training set	300W-LP															
testing set	AFLW2000								BIWI							
component 1	-				Attention mapping				-				Attention mapping			
component 2	-		orthogonality		-		orthogonality		-		orthogonality		-		orthogonality	
component 3	-	Capsule	-	Capsule	-	Capsule	-	Capsule	-	Capsule	-	Capsule	-	Capsule	-	Capsule
MAE	5.120	4.977	4.979	4.951	4.883	4.740	4.866	**4.669**	4.390	4.280	4.298	4.165	4.175	4.022	4.204	**3.972**
MAEV	6.487	6.181	6.306	6.274	6.167	6.058	6.157	**5.989**	6.450	6.360	6.255	6.186	6.256	5.871	6.280	**5.864**

Table 4: Ablation study for different feature mapping methods (with/ without attention mapping) and loss items (with/ without orthogonality loss) and capsule network (with/without capsule network). Trained on 300W-LP.

training set	BIWI (train)							
testing set	BIWI (test)							
component 1	-				Attention mapping			
component 2	-		orthogonality		-		orthogonality	
component 3	-	Capsule	-	Capsule	-	Capsule	-	Capsule
MAE	3.422	2.978	3.333	2.835	2.840	2.826	2.843	**2.801**
MAEV	4.920	4.185	4.791	4.069	4.162	4.142	4.118	**4.061**

Table 5: Ablation study for different feature mapping methods (with/ without attention mapping) and loss items (with/ without orthogonality loss) and capsule network (with/without capsule network). Trained on BIWI.

MAEV when tested on BIWI dataset.

Table 3 shows the experiment results of 3-fold cross validation on BIWI dataset using different methods. In this scenario, we only compare our proposed method with other RGB-based ones. We compare both the MAE and MAEV results and our proposed TriNet achieves the best performance.

training set	300W-LP				BIWI (train)	
testing set	AFLW2000		BIWI		BIWI (test)	
Uniform prediction	-	✓	-	✓	-	✓
MAE	**4.669**	4.953	**3.972**	4.500	**2.801**	3.302
MAEV	**5.989**	6.245	**5.864**	6.764	**4.061**	4.862

Table 6: Ablation study for using a uniform prediction module.

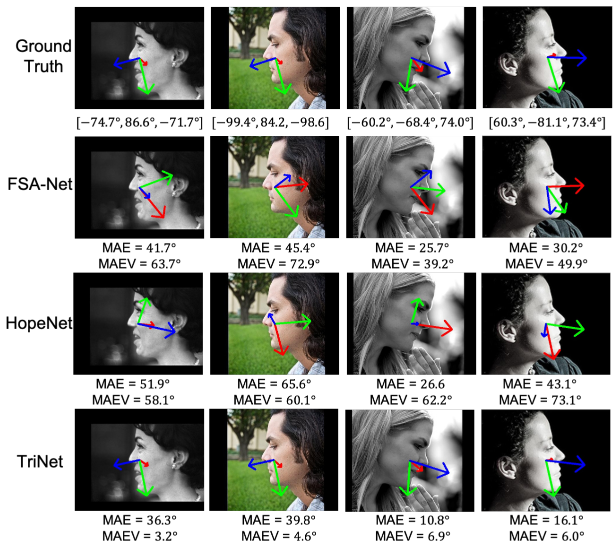

Figure 8: Comparison of pose estimation results on AFLW2000 profile images. All trained on 300W-LP.

4.5. Error Analysis

We conduct the error analysis of three landmark-free methods (FSA-Net, Hopenet and TriNet) on AFLW2000 dataset. The Euler angles' range of $[-99^\circ, 99^\circ]$ are equally divided to intervals that span 33°. The results are shown in Fig. 6 and Fig. 7.

The first thing worth noting is that prediction error of MAEV increases much more slowly than MAE as absolute values of pose angles increase. MAE can achieve about 60° for large pitch and roll angles while MAEV has only around 30°. This conforms to our findings in section 1 that MAE fails to measure performance at large pose angles.

We use Fig. 8 to further illustrate the reason. Since gimbal lock causes ambiguity issue to Euler angles, many researchers limit the yaw angle in the range of $(-90^\circ, 90^\circ)$ to ensure the representation of rotation is unique. However, this brings in a new issue. Assume the rotation is in the order of pitch (γ), yaw (β) and roll (α) and denoted by (α, β, γ). As yaw exceeds the boundary $\pm 90^\circ$, it will cause significant change in pitch and roll angles. For example, assume a person with head pose of $(10^\circ, 89^\circ, 15^\circ)$. If he increases the yaw angle by 3°, this causes no observable difference in image. However, the annotation becomes $(-170^\circ, 88^\circ, -165^\circ)$ since yaw angle is not allowed to surpass 90°. This explains why similar profile images have very different Euler angle labels. As a result, Euler angle based network models hardly learn anything from such profile images. Fig. 8 verifies its validity. It shows the prediction results of different methods on profile images from AFLW2000. FSA-Net and Hopenet have very arbitrary results whereas our TriNet has accurate predictions. This figure also shows the problem of MAE. By comparing the MAE and MAEV results of TriNet (4th row), we can conclude that MAE cannot measure performance of networks on profile images.

The second noticeable thing is that the yaw angle error distribution is different from those of pitch and roll. As the absolute values of ground truth angles grow, the MAEV of yaw grows much more slowly compared with pitch and roll. We attribute this to their different distributions in the 300W-LP dataset.Data samples have their yaw angles evenly distributed across $[-99^\circ, +99^\circ]$ whereas 93% pitch and 90% roll angles concentrate in the range of $[-33^\circ, +33^\circ]$. Shortage of training data for large pitch and roll angles makes their performances worse than the yaw angle.

4.6. Ablation Study

In this section, we conduct ablation studies to analyze how each network component will affect the model performance on different testing sets. We include feature mapping methods, loss item, and capsule module as three testing components. Table 4 and 5 report both the MAE and MAEV results. We observe the best results on all the testing sets when combining all these three modules. In addition, in prediction module, we experiment the influence of uniform sampling. In other words, we use $(9, 9, 9)$ as the dimensions of three output vectors from capsule network instead of $(81, 27, 9)$. Table 6 shows non-uniform sampling can achieve better results.

5. Conclusion

In this paper, we put forward a new vector-based annotation and a new metric MAEV. They can solve the discontinuity issues caused by Euler angles. By the combination of new vector representation and our TriNet, we achieve state-of-the-art performance on the task of head pose estimation.

References

[1] Peter N Belhumeur, David W Jacobs, David J Kriegman, and Neeraj Kumar. Localizing parts of faces using a consensus of exemplars. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2930–2940, 2013.

[2] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. *International Journal of Computer Vision*, 107(2):177–190, 2014.

[3] Piotr Dollár, Peter Welinder, and Pietro Perona. Cascaded pose regression. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1078–1085. IEEE, 2010.

[4] Gabriele Fanelli, Matthias Dantone, Juergen Gall, Andrea Fossati, and Luc Van Gool. Random forests for real time 3d face analysis. *International journal of computer vision*, 101(3):437–458, 2013.

[5] Gabriele Fanelli, Juergen Gall, and Luc Van Gool. Real time head pose estimation with random regression forests. In *CVPR 2011*, pages 617–624. IEEE, 2011.

[6] P Fua and V Lepetit. Monocular model-based 3d tracking of rigid objects. *Comput. Graph. Vis*, 1(1):1–89, 2005.

[7] Ge Gao, Mikko Lauri, Jianwei Zhang, and Simone Frintrop. Occlusion resistant object rotation regression from point cloud segments. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[8] Nicolas Gourier, Jérôme Maisonnasse, Daniela Hall, and James L Crowley. Head pose estimation on low resolution images. In *International Evaluation Workshop on Classification of Events, Activities and Relationships*, pages 270–280. Springer, 2006.

[9] Jianzhu Guo, Xiangyu Zhu, Yang Yang, Fan Yang, Zhen Lei, and Stan Z Li. Towards fast, accurate and stable 3d dense face alignment. *arXiv preprint arXiv:2009.09960*, 2020.

[10] Heng-Wei Hsu, Tung-Yu Wu, Sheng Wan, Wing Hung Wong, and Chen-Yi Lee. Quatnet: Quaternion-based head pose estimation with multiregression loss. *IEEE Transactions on Multimedia*, 21(4):1035–1046, 2018.

[11] Bin Huang, Renwen Chen, Wang Xu, and Qinbang Zhou. Improving head pose estimation using two-stage ensembles with top-k regression. *Image and Vision Computing*, 93:103827, 2020.

[12] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1867–1874, 2014.

[13] Amit Kumar, Azadeh Alavi, and Rama Chellappa. 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)*, pages 258–265. IEEE, 2017.

[14] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009.

[15] Zhaoxiang Liu, Zezhou Chen, Jinqiang Bai, Shaohua Li, and Shiguo Lian. Facial pose estimation by deep learning from label distributions. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[16] Siddharth Mahendran, Haider Ali, and René Vidal. 3d pose regression using convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2174–2182, 2017.

[17] Jianqin Mao. Optimal orthonormalization of the strapdown matrix by using singular value decomposition. *Computers & mathematics with applications*, 12(3):353–362, 1986.

[18] Peter M. Roth Martin Koestinger, Paul Wohlhart and Horst Bischof. 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*, 2011.

[19] Erik Murphy-Chutorian, Anup Doshi, and Mohan Manubhai Trivedi. Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation. In *2007 IEEE Intelligent Transportation Systems Conference*, pages 709–714. IEEE, 2007.

[20] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4561–4570, 2019.

[21] Rajeev Ranjan, Vishal M Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):121–135, 2017.

[22] Bisser Raytchev, Ikushi Yoda, and Katsuhiko Sakaue. Head pose estimation by nonlinear manifold learning. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 4, pages 462–466. IEEE, 2004.

[23] Nataniel Ruiz, Eunji Chong, and James M Rehg. Fine-grained head pose estimation without keypoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2074–2083, 2018.

[24] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 397–403, 2013.

[25] Soheil Sarabandi, Arya Shabani, Josep M Porta, and Federico Thomas. On closed-form formulas for the 3-d nearest rotation matrix problem. *IEEE Transactions on Robotics*, 2020.

[26] Ashutosh Saxena, Justin Driemeyer, and Andrew Y Ng. Learning 3-d object orientation from images. In *2009 IEEE International Conference on Robotics and Automation*, pages 794–800. IEEE, 2009.

[27] Chen Song, Jiaru Song, and Qixing Huang. Hybridpose: 6d object pose estimation under hybrid representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 431–440, 2020.

[28] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3476–3483, 2013.

[29] Roberto Valle, José M Buenaposada, Antonio Valdés, and Luis Baumela. Face alignment using a 3d deeply-initialized ensemble of regression trees. *Computer Vision and Image Understanding*, 189:102846, 2019.

[30] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[31] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[32] Tsun-Yi Yang, Yi-Ting Chen, Yen-Yu Lin, and Yung-Yu Chuang. 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*, pages 1087–1096, 2019.

[33] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1941–1950, 2019.

[34] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

[35] Erjin Zhou, Haoqiang Fan, Zhimin Cao, Yuning Jiang, and Qi Yin. Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 386–391, 2013.

[36] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019.

[37] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z Li. Face alignment across large poses: A 3d solution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 146–155, 2016.

[38] Xiangyu Zhu, Zhen Lei, Junjie Yan, Dong Yi, and Stan Z Li. High-fidelity pose and expression normalization for face recognition in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 787–796, 2015.

[39] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2879–2886. IEEE, 2012.