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.

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 \( q \in \mathbb{R}^4, \|q\|_2 = 1 \) has the antipodal problem which results in \( q \) and \(-q\) 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

![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.](image)

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.

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 \( q \in \mathbb{R}^4, \|q\|_2 = 1 \) has the antipodal problem which results in \( q \) and \(-q\) 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

![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.](image)
poses an algorithm called Cascaded Pose Regression (CPR) for finding correspondences between 2D and 3D feature points. This approach first detects landmarks in the image and then estimates the pose by solving the constraints in the deep network pipeline.

2.1. Landmark-Based Approaches

These approaches typically detect key landmarks from the image first and then estimate the poses by solving the correspondences between 2D and 3D feature points. For example, [3] proposes a cascaded regression function from training data that it uses to obtain a set of facial landmarks from the image with this function.

With the advent of CNN, numerous CNN-based methods have been designed and achieve superior performances than their predecessors. [28] presents an approach which draws on a three-level convolutional network to estimate the positions of facial landmarks. [37] introduces a new method using a novel 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] proposes 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] presents a method 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.

With the advent of CNN, numerous CNN-based methods have been designed and achieve superior performances than their predecessors. [28] presents a new method which draws on a three-level convolutional network to estimate the positions of facial landmarks. [37] introduces a new method using a novel 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] proposes 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. Related Work

2.1. Landmark-Based Approaches

These approaches typically detect key landmarks from the image 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)
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 \( v_1 = [1, 0, 0]^T \), \( v_2 = [0, 1, 0]^T \) and \( v_3 = [0, 0, 1]^T \) respectively. After applying a rotation matrix \( R_{3 \times 3} = [r_1, r_2, r_3] \) where \( r_i \) denotes the \( i^{th} \) column vector in \( R \), the three vectors then become \( v'_1 = Rv_1 = r_1 \), \( v'_2 = Rv_2 = r_2 \) and \( v'_3 = Rv_3 = r_3 \). 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 us human beings to perceive the difference of rotation angles through the distance between endpoints 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 \( X = \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\} \), find a mapping function \( F \) such that estimates \( \hat{R}^{(i)} = F(x^{(i)}) \) where \( \hat{R}^{(i)} = [\hat{r}_1^{(i)}, \hat{r}_2^{(i)}, \hat{r}_3^{(i)}] \) that matches the ground truth rotation matrix \( R \) as close as possible. We try to find an optimal \( F \) for all \( X \) by minimizing the sum of squared \( L_2 \) norm between the predicted and ground truth vectors.

\[
\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left( \| \hat{r}_1^{(i)} - \hat{r}_1^{(i)} \|_2 + \| \hat{r}_2^{(i)} - \hat{r}_2^{(i)} \|_2 + \| \hat{r}_3^{(i)} - \hat{r}_3^{(i)} \|_2 \right)^2 (1)
\]

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 \( 1 \times 1 \) 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.
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:

\[
\hat{y} = \frac{1}{S} \sum_{s=1}^{S} \sum_{i=1}^{n^{(s)}} \left( p_i^{(s)} \cdot q_i^{(s)} \right)
\]

(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 \(L_{\text{reg}}\) and the orthogonal loss \(L_{\text{ortho}}\) which measures the orthogonality between each pair of the predicted vectors. The overall objective loss is the weighted sum of two losses:

\[
\mathcal{L} = L_{\text{reg}}(\mathbf{v}_i, \hat{\mathbf{v}}_i) + \alpha L_{\text{ortho}}(\hat{\mathbf{v}}_i, \hat{\mathbf{v}}_j)
\]

(3)

where \(\hat{\mathbf{v}}_i\) and \(\mathbf{v}_i\) are the \(i^{th}\) predicted and ground truth vectors respectively. The weighted term \(\alpha\) 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:

\[
L_{\text{reg}} = \sum_{i=1}^{3} \text{mse}(\mathbf{v}_i, \hat{\mathbf{v}}_i)
\]

(4)

\[
L_{\text{ortho}} = \sum_{i \neq j} \text{mse}(\hat{\mathbf{v}}_i, \hat{\mathbf{v}}_j, 0) \text{ where } i, j = 1, 2, 3
\]

(5)
the difference between $R$ is given, $R$ depends on which two rows or columns of $R$.

Its simple geometric interpretation makes it very popular, we adopt the Gram–Schmidt process to either its rows or columns.

A way to find a rotation matrix from a noisy matrix is applying

sure of closeness needs to have physical meaning. A naive

This paper adopts the measure of Euclidean or Frobenius norm of $R$

subject to $\hat{R}^T \hat{R} = I$ and $\det \hat{R} = +1$

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

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

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 $-99^\circ$ and $99^\circ$ 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
### 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Euler angles errors</th>
<th>Vector errors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>roll</td>
<td>pitch</td>
</tr>
<tr>
<td>Hopenet[23]</td>
<td>6.132</td>
<td>7.120</td>
</tr>
<tr>
<td>Quatnet[10]*</td>
<td>3.920</td>
<td>5.615</td>
</tr>
<tr>
<td>HPE[11]*</td>
<td>4.800</td>
<td>6.180</td>
</tr>
<tr>
<td>TriNet</td>
<td>(4.042</td>
<td>5.767</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Euler angles errors</th>
<th>Vector errors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>roll</td>
<td>pitch</td>
</tr>
<tr>
<td>3DDFA[12]</td>
<td>13.224</td>
<td>41.899</td>
</tr>
<tr>
<td>Hopenet[23]</td>
<td>3.719</td>
<td>5.885</td>
</tr>
<tr>
<td>Quatnet[10]*</td>
<td>2.936</td>
<td>5.492</td>
</tr>
<tr>
<td>HPE[11]*</td>
<td>3.120</td>
<td>5.180</td>
</tr>
<tr>
<td>TriNet</td>
<td>(4.112</td>
<td>4.758</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Euler angles</th>
<th>Vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>roll</td>
<td>pitch</td>
</tr>
<tr>
<td>TriNet</td>
<td>(2.928</td>
<td>3.035</td>
</tr>
</tbody>
</table>

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.

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.

<table>
<thead>
<tr>
<th>Component 1</th>
<th>Component 2</th>
<th>Component 3</th>
<th>MAE</th>
<th>MAEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attention mapping</td>
<td>orthogonality</td>
<td>Capsule</td>
<td>5.120</td>
<td>6.487</td>
</tr>
<tr>
<td>orthogonality</td>
<td>Capsule</td>
<td>Capsule</td>
<td>4.977</td>
<td>6.181</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>Capsule</td>
<td>4.951</td>
<td>6.306</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Component 1</th>
<th>Component 2</th>
<th>Component 3</th>
<th>MAE</th>
<th>MAEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIWI (train)</td>
<td>-</td>
<td>Attention mapping</td>
<td>3.422</td>
<td>4.920</td>
</tr>
<tr>
<td>BIWI (test)</td>
<td>Capsule</td>
<td>-</td>
<td>2.978</td>
<td>4.185</td>
</tr>
<tr>
<td>orthogonality</td>
<td>Capsule</td>
<td>-</td>
<td>3.333</td>
<td>4.791</td>
</tr>
<tr>
<td>MAE</td>
<td>3.422</td>
<td>2.978</td>
<td>3.333</td>
<td>4.920</td>
</tr>
<tr>
<td>MAEV</td>
<td>4.920</td>
<td>4.185</td>
<td>4.791</td>
<td>4.061</td>
</tr>
</tbody>
</table>

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.
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 $33^\circ$. 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 $60^\circ$ for large pitch and roll angles while MAEV has only around $30^\circ$. 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 ($\gamma$), yaw ($\beta$) and roll ($\alpha$) and denoted by $(\alpha, \beta, \gamma)$. 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 $3^\circ$, this causes no observ-
References


