Robust Driver Head Pose Estimation in Naturalistic Conditions from Point-Cloud Data

Tiancheng Hu, Sumit Jha and Carlos Busso
Head pose estimation is an important task

- With applications in areas including
  - advanced driver assistance system [Murphy-Chutorian et al. 2007]
  - visual attention modelling [Ba and Odobez 2009]
  - gaze estimation system [Zhang et al. 2015]

- In the automotive domain, it is a challenging task due to

(a) Occlusion  (b) Extreme head pose  (c) Illumination
Time-of-Flight Depth Camera

- Time-of-flight camera
  - Utilize active infrared lighting
  - Calculate distance between camera and object based on round trip time
  - Immune to illumination change!

- We adopt a pico flexx camera, which provides

  ![Point cloud data](https://pmdtec.com/picofamily/flexx/)
  ![Grayscale image](https://pmdtec.com/picofamily/flexx/)

Image credit: https://pmdtec.com/picofamily/flexx/
Point Cloud Processing
- Wu et al. (2015) represents point clouds as 3D voxel grids and use 3D CNNs to process them
- Su et al. (2015) renders 2D images from point cloud at different angles and process the set of 2D images using CNNs
- PointNet [Qi et al. 2017 ICCV] and PointNet++ [Qi et al. 2017 NIPS] directly process 3D point-cloud data without converting to any other intermediate representation

Driver Head Pose Estimation
- Borghi et al. (2017) utilizes a multi-modal approach, with CNNs trained on RGB, depth map, and optical flow data which are then fused to predict head pose
- Schwarz et. al (2017) proposes a CNN based model which fuses information from infrared images and depth maps and regresses head pose
PointNet++ [Qi et al. 2017 NIPS]

Multi-scale, Multi-layer Feature Extraction

Hierarchical point set feature learning

Segmentation

Classification

Task specific layers
Set Abstraction Layer

- **Sampling**: iterative farthest point sampling, resulting in $N$ “anchor points”
  - Motivation: Point clouds are usually large and contain redundant points
  - Goal: Decrease redundancy while maintaining useful information
- **Grouping**: group points within a radius $R$ of the “anchor points”
  - Motivation: CNN captures the local features of a neighborhood
  - Goal: Capture the relationship between anchor points and the neighborhood

Represent rotation in 6D, according to Zhou et al. 2019
• Set Abstraction Layer
  • **PointNet**: multi-layer perceptron to lift the feature up to a higher dimension
    • Motivation: Just as many deep learning networks, we need to extract high-level feature
    • Goal: find a discriminative feature representation
  • Each set abstraction layer has different $N$ and $R$ values to capture features of different scale
  • Multiple set abstraction layer stacked together to extract high-level feature

Represent rotation in 6D, according to Zhou et al. 2019

![Set Abstraction Layer Diagram](image)
Dataset

- Multimodal Driver Monitoring (MDM) Dataset
  - 4 GoPro RGB cameras, 1 pico flexx depth camera
  - Naturalistic driving with a diverse range of head poses
  - Head pose labels provided by Fi-Cap [Jha and Busso 2018]
  - 59 subjects (27 male 32 female, mostly college students)
    - Used 22 in this study, total duration 17 hours 39 minutes
Setup - Sensors

- GoPro Camera in the back
- Fi-Cap
- Microphones array
- CAN-BUS Information
- Depth Camera
- GoPro Camera recording the road
- GoPro Camera facing the driver
- GoPro Camera facing the Driver close to the rear mirror
Example data from different sensors
Setup - Markers
Protocol — Phase 1

- **Phase 1: Natural Gaze – (Parked Vehicle)**
  - Subject asked to look at target markers on the windshield in random order
  - Subject asked to look at a trackable marker that the researcher moves around in front of the car
Protocol — Phase 2

- **Phase 2: Natural task – Driving**
  - Subject asked to follow navigation on a smart phone
    - Multiple destinations in sequence
  - Subject asked to change radio channels when driving
Protocol — Phase 3

- **Phase 3: Natural Gaze – Driving**
  - Subject asked to look at landmarks on the road and answer questions
  - Subject asked to look at points on the windshield
Training Details

- **Point Cloud Preprocessing**
  - Distance-based filtering -> grid-based sampling -> 5000 points -> normalized (centroid at (0,0,0), all points in unit sphere)

- **Rotation Representation**
  - Represent rotation in 6D, according to Zhou et al. 2019
  - Easily convertible to full rotation matrix

- **Training Detail**
  - Train: 14 subject; development: 4 subject; test: 4 subject
  - Adam optimizer, learning rate = 0.001 with learning rate decay of 0.7 per 2 million steps
  - L2 loss
Baseline Approaches

- **OpenFace 2.0** [Baltrušaitis et al. 2018]
  - State-of-the-art (SOTA) toolkit for face analysis, including head pose estimation

- **Face Alignment Network (FAN)** [Bulat and Tzimiropoulos, 2017]
  - One of the SOTAs for facial landmark estimation
  - Use singular value decomposition to get rotation from landmarks

- **For Both**
  - Full resolution (1920x1080) RGB image captured from GoPro is used as input
  - To avoid difference in angle definition:
    - Subject-wise transformation applied between baseline prediction and ground truth
Result — Test Set Ground Truth Distribution

Most Data in this region

log10 scale
Result — OpenFace 2.0 Detection Failure

Fails in mostly frames with larger rotation

4.8%
Result — FAN Detection Failure

- Better than OpenFace
- Failure also concentrated in larger ground truth rotation

2.1%
Result — Proposed Method

- No head detection needed
- Thus 0% detection failure

0%
<table>
<thead>
<tr>
<th></th>
<th>Errors (Model Set)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Roll(°)</td>
<td>Yaw(°)</td>
<td>Pitch(°)</td>
</tr>
<tr>
<td>Proposed Model</td>
<td>5.91</td>
<td>7.32</td>
<td>6.68</td>
</tr>
<tr>
<td>Errors (OpenFace Set)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Roll(°)</td>
<td>Yaw(°)</td>
<td>Pitch(°)</td>
</tr>
<tr>
<td>Proposed Model</td>
<td>5.48</td>
<td>7.15</td>
<td>6.39</td>
</tr>
<tr>
<td>OpenFace 2.0</td>
<td>9.32</td>
<td>6.21</td>
<td>8.42</td>
</tr>
<tr>
<td>Errors (FAN Set)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Roll(°)</td>
<td>Yaw(°)</td>
<td>Pitch(°)</td>
</tr>
<tr>
<td>Proposed Model</td>
<td>5.66</td>
<td>7.24</td>
<td>6.53</td>
</tr>
<tr>
<td>FAN</td>
<td>11.30</td>
<td>19.28</td>
<td>8.47</td>
</tr>
</tbody>
</table>
Proposed Model

- Proposed model has lower error overall
- Error especially lower in large yaw rotations

\[ \Delta(R_1, R_2) = \| \log(R_1 R_2^T) \| \]
Conclusion & Future Work

- **Ours:** 1st deep learning based head pose estimation algorithm directly from point cloud
- Evaluated on Multimodal Driver Monitoring dataset
- Achieve better performance than the baselines
- Model reliable overall, especially in large rotations
- **Future Work**
  - Multimodal approach (depth, RGB and more) to jointly model head pose
  - Temporal modelling for driver head pose
  - Build a more parameter-efficient model for real-time applications
This work was supported by Semiconductor Research Corporation (SRC) / Texas Analog Center of Excellence (TxACE), under task 2810.014.

Our Research: msp.utdallas.edu