FACE POSE ESTIMATION

A PREPRINT

Denys Maletskyi
Faculty of Applied Sciences
of Ukrainian Catholic University
Kozelnytska St 2a, L'viv, Ukraine
maletskyi@ucu.edu.ua

ABSTRACT

Head Pose Estimation is the problem of estimating the pose that consists of 6 degrees-of-freedom (DOF) which are made up of the rotation (roll, pitch, and yaw) and 3D translation of the camera with respect to the world from a single RGB camera. For solving this problem the classical computer vision approaches are used, such as HaarCascade for detecting face features, Perspective-n-Point algorithm for estimating head pose and further usage of filters for smoothing the estimated pose.

Code will be available at: https://github.com/maletsden/face-pose-estimation

Keywords Pose Estimation · HaarCascade · Perspective-n-Point · Kalman Filter

1 Introduction

Head Pose Estimation (HPE) is the task of estimating the pose that consists of 6 degrees-of-freedom (DOF) from images or video. Head pose estimation is often used in virtual and augmented reality solution, marker-less motion capture for 3D facial alignment, face orientation estimation, and 3D face modeling from the single camera.

2 Background & Related Work

WithTeeth: Denture Preview in Augmented Reality (Amirkhanov et al. 2018). WithTeeth article propose the algorithms for estimating upper and lower jaws pose that would be discussed in this report. The face features extraction is implemented using Haar Cascades (Viola and Jones 2001) classifier. Further extracted face features (face landmarks) are used together with previously prepared canonical face 3D model for solving Perspective-n-Point for estimating face pose (estimating 3D face model points to 2D image landmarks transformation), authors uses 2 different sets of points from that canonical face 3D model for tracking upper and lower jaws pose independently. However, due to poor performance of Haar Cascades combined with estimation errors aggregated in after solving Perspective-n-Point additional smoothing must be applied for the estimated pose to improve visual user experience.

BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs (Bazarevsky et al. 2019). BlazeFace is a lightweight feature extraction network inspired by, but distinct from MobileNetV1/V2, a GPU-friendly anchor scheme modified from Single Shot MultiBox Detector (SSD), and an improved tie resolution strategy alternative to non-maximum suppression. This model is perfectly optimized for using on the edge devices, both the size and the computational speed are optimized that allows usage on any mobile devices even with very limited memory accessibility.

3 Method

This report will review the method for head pose estimation described in the WithTeeth article and analyze advantages and disadvantages of the approach proposed bu the authors.

3.1 Face Feature Extraction

Face Feature Extraction is one of the main problem that need to be solved for the head pose estimation. One of the possible approaches for extracting face features from the input image is to use Haar Cascade Classifier. Haar feature-based cascade classifiers is an object detection method for finding features in the image by applying Haar-features to the analyzing image, but previously grouping Haar-features in the cascade to reduce expensive computational costs.

The main disadvantages of this method is that it is working with limited scene conditions (head should be directed towards the camera with only limited rotation radius) with still high computation cost that is not suitable for less powerful devices.

The results of applying this method to the input image is a set of 2D image points (UV coordinates) that corresponds to major face features.

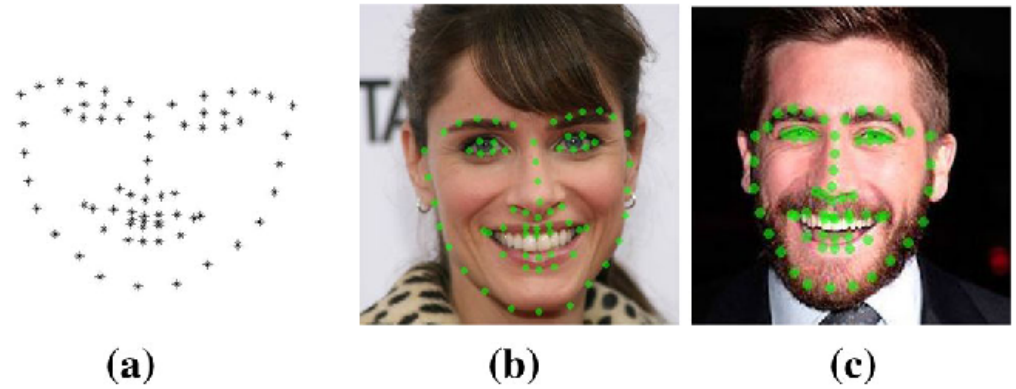

Figure 1: Face Features Extraction

3.2 Solving Perspective-n-Point Problem

Perspective-n-Point Problem is the problem of estimating pose of some 3D object of interest. The solution to this problems is the algorithm that estimates the transformation $\beta \in \mathbb{R}^6$ from 3D world object (set of world points $a_i \in \mathbb{R}^3$, in the terms of this report - the 3D canonical face model) to 2D image landmarks (set of image points $b_i \in \mathbb{R}^2$, in the terms of this report - 2D face features extracted on the previous stage).

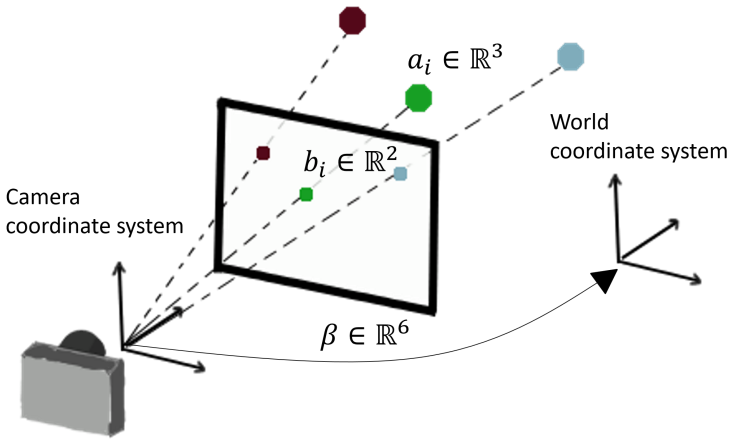

Figure 2: Perspective-n-Point Problem

The purpose estimated $\beta \in \mathbb{R}^6$ transformation can be used together with camera intrinsic (C) and perspective projection (P) models to project 3D world coordinates on the image plain, such as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = C \cdot P \cdot \beta \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Estimated transformation β has 6 degrees-of-freedom (DOF) - 3 rotation angles and 3 translations:

$$\begin{bmatrix} X^c \\ Y^c \\ Z^c \\ 1 \end{bmatrix} = \beta \cdot \begin{bmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{bmatrix}$$

The solution for this problem used in this project is the iterative method that is based on a Levenberg-Marquardt optimization (Levenberg 1944, Marquardt 1963). In this case the function finds such a pose that minimizes reprojection error, that is the sum of squared distances between the image points $b_i \in \mathbb{R}^3$ in the homogeneous coordinate system and the projected world points $a_i \in \mathbb{R}^3$ into image plane:

$$\min \sum_{i=0}^{n} \|\hat{b}_i - b_i\|^2,$$

where

$$\hat{b}_i = C \cdot P \cdot \beta \cdot a_i$$

The major disadvantage of this algorithm for face pose estimation is physiological differences in human faces, such that pose estimation accuracy using canonical 3D face model will dependent on physiological features of user's face.

3.3 Smoothing Estimated Pose

The achieved estimated head pose from previous stage should be smoothed (filtered) before usage in real-world application due to disadvantages of previously used methods and therefore their errors. In this work two methods for smoothing estimated head pose will be used: standard linear filtering and the Kalman filter.

3.4 Linear Filter

Linear Filter process time-varying input signals $x[n]$ to produce the output signal $y[n]$, subject to the constraint of linearity. The linear filter can be defined be standard equation:

$$\sum_{i=0}^{M} a_i \cdot y[n-i] = \sum_{j=0}^{N} b_j \cdot x[n-j]$$

or equivalently:

$$a_0 \cdot y[n] = \sum_{j=0}^{N} b_j \cdot x[n-j] - \sum_{i=1}^{M} a_i \cdot y[n-i]$$

The sets of $\{a_i \in \mathbb{R} \,|\, i \in [0;M]\}$ and $\{b_i \in \mathbb{R} \,|\, i \in [0;N]\}$ filter coefficients must be manually configured for designing the desirable features of the output signal.

3.5 Kalman filter

Kalman filter (Kalman 1960), also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed over time, including statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone, by estimating a joint probability distribution over the variables for each timeframe.

Kalman filter algorithm consists of two stages: prediction and update.

Prediction:

Predicted state estimate	$\hat{x}_k^- = F\hat{x}_{k-1}^+ + Bu_{k-1}$
Predicted error covariance	$P_k^- = FP_{k-1}^+F^T + Q$

Update:

Measurement residual	$\tilde{y}_k = z_k - H\hat{x}_k^-$
Kalman gain	$K_k = P_k^- H^T(R + HP_k^- H^T)^-1$
Updated state estimate	$\hat{x}_k^+ = \hat{x}_k^- + K\tilde{y}$
Updated error covariance	$P_k^+ = (I - K_k H)P_k^-$

4 Results

For testing the performance of the final application were conducted experiments with real RGB webcamera. Firstly, camera was calibrated to find all camera intrinsic parameters. Were conducted 6 experiments: 3 experiments of rotating head around different axis and 3 experiments of moving head in space in different directions. The next results were achieved:

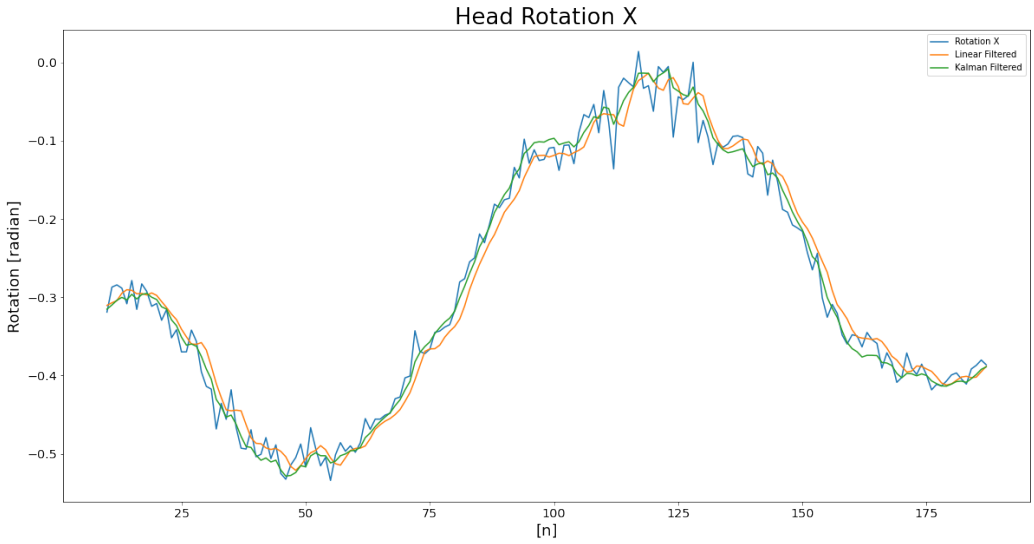

Figure 3: Head Rotation X

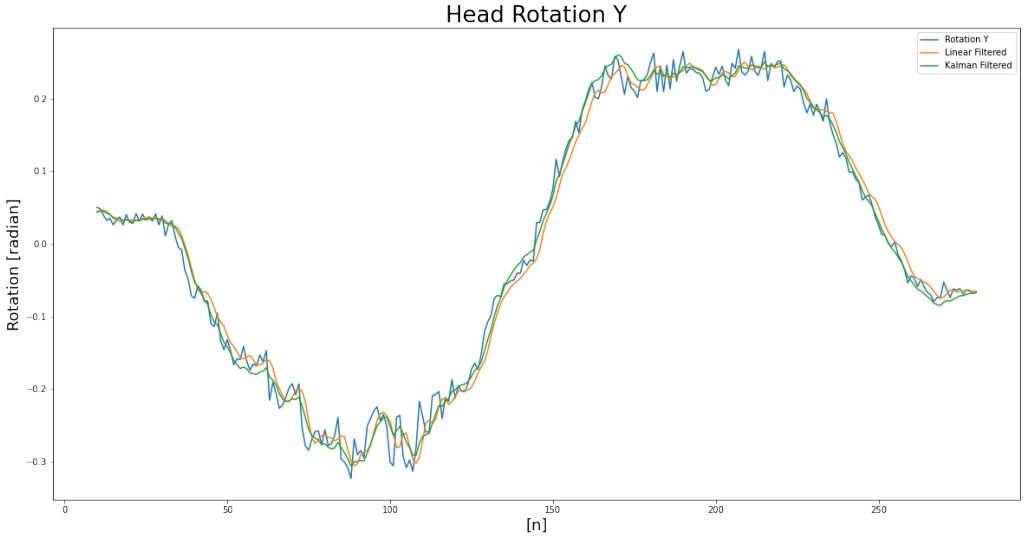

Figure 4: Head Rotation Y

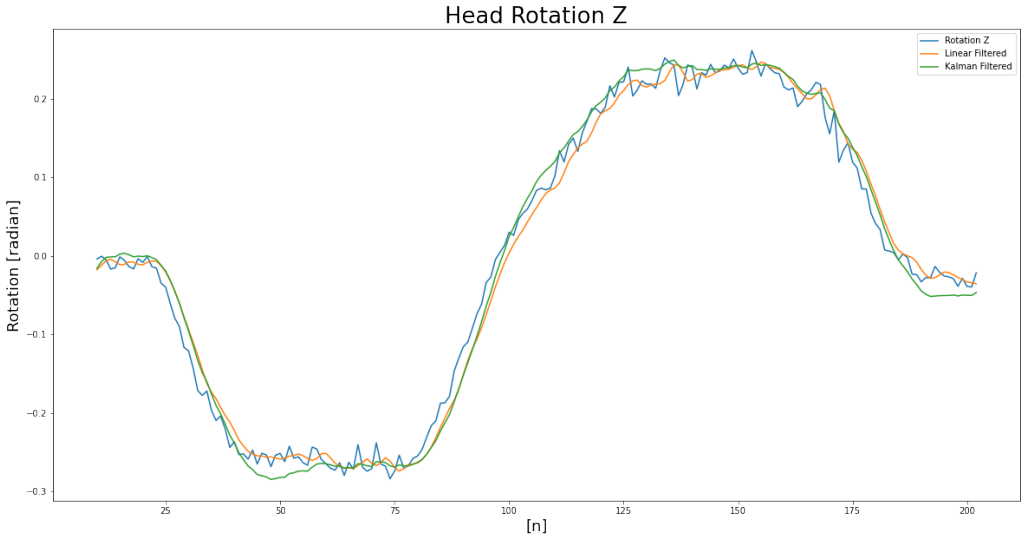

Figure 5: Head Rotation Z

The results of first 3 experiments shows that head rotations tracking is full of noises due to the limitation of the used methods described above. Fortunately, smoothing of estimated head pose shows better results.

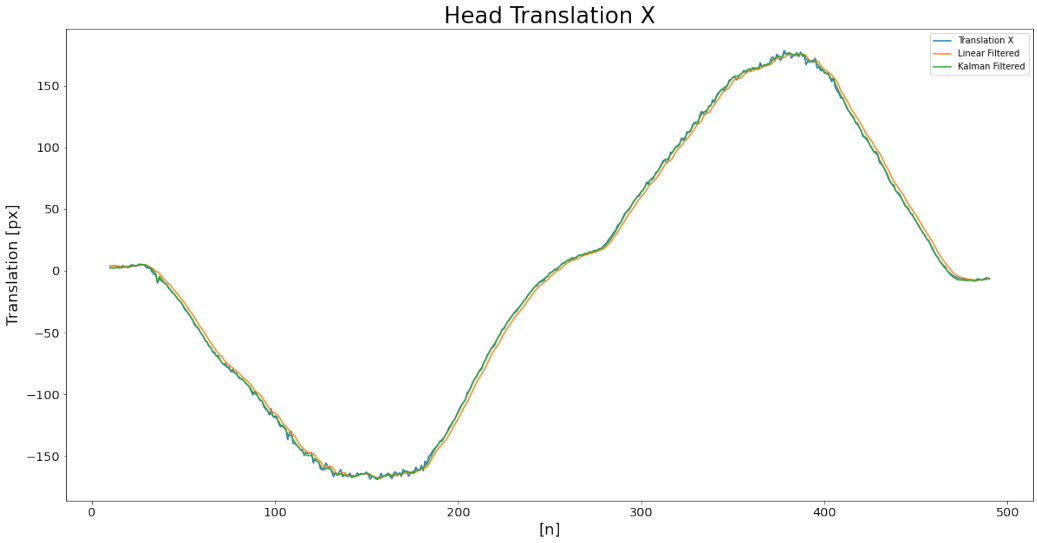

Figure 6: Head Translation X

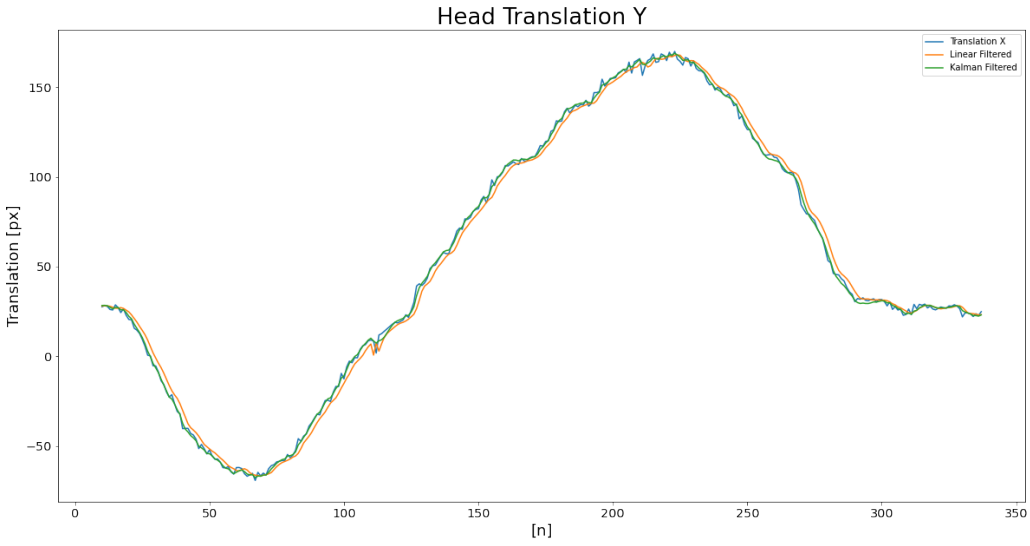

Figure 7: Head Translation Y

The results of conducting 2 experiments of translating the head in 2 directions (x and y axes) shows much better and more stable results almost without any noises. Those results are expected since this 2 DOF are the only one that actually are independent from the 3D face model.

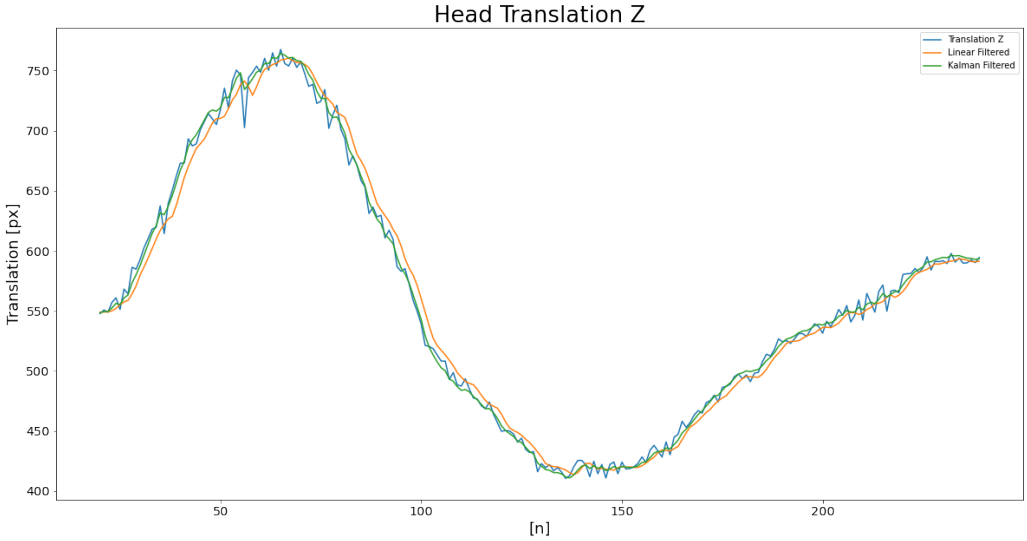

Figure 8: Head Translation Z

The last experiment is shown a bit more nosier results comparing with the previous 2 translation, because the estimating of this head translation depends on canonical 3D head model that used in the previous method.

In general, results of conducted experiments shown that the selected methods for estimating head pose shows good results only under limited conditions. The main pitfall of the solution is the face features extraction that is done using Haar Cascade Classifier, which works properly only with limited range of head rotations, especially around the Y and Z axis.

5 Applications

Proposed head pose estimation application can be used in virtual and augmented reality solution, marker-less motion capture for 3D facial alignment and other astonishing user experiments.

6 Future Perspective

In the next version of the application the plan is to improve the performance of the face features extraction, a good solution for that can be switch from using Haar Cascade Classifier to, for example, BlazeFace discussed in the previous sections of this report shows promising perspectives.

The another step for improving the performance of the pose estimation can be the creation of user calibration to illuminate the the pitfalls of Perspective-n-Point Problem due to physiological differences in human faces and canonical 3D face model.

References

[Ami+18] Aleksandr Amirkhanov et al. “WithTeeth: Denture Preview in Augmented Reality”. In: *VMV*. 2018.

[Baz+19] Valentin Bazarevsky et al. *BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs*. 2019. arXiv: `1907.05047 [cs.CV]`.

[Kal60] Rudolph Emil Kalman. “A New Approach to Linear Filtering and Prediction Problems”. In: *Transactions of the ASME–Journal of Basic Engineering* 82.Series D (1960), pp. 35–45.

[Lev44] Kenneth Levenberg. "A METHOD FOR THE SOLUTION OF CERTAIN NON-LINEAR PROBLEMS IN LEAST SQUARES". In: *Quarterly of Applied Mathematics* 2.2 (1944), pp. 164–168. ISSN: 0033569X, 15524485. DOI: 10.2307/43633451. URL: http://www.jstor.org/stable/43633451.

[Mar63] Donald W. Marquardt. *An Algorithm for Least-Squares Estimation of Nonlinear Parameters*. 1963. DOI: 10.1137/0111030.

[VJ01] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features". In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1. 2001, pp. I–I. DOI: 10.1109/CVPR.2001.990517.