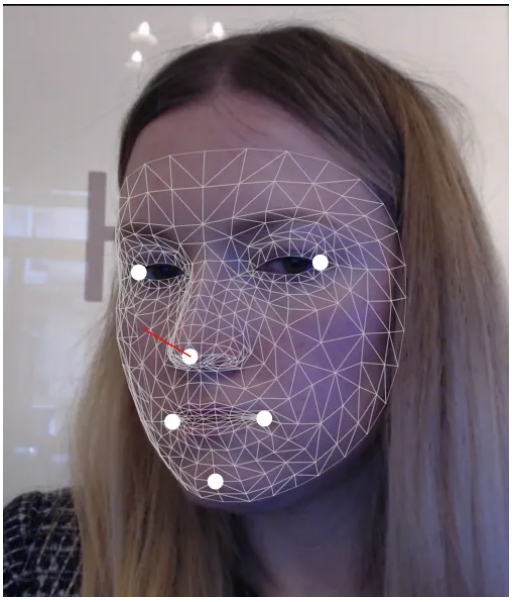
Head Pose Estimation with MediaPipe and OpenCV in Javascript

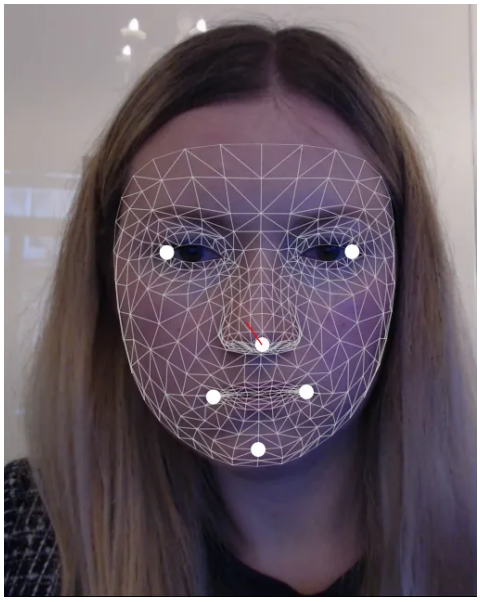
Susanne Thierfelder · Follow

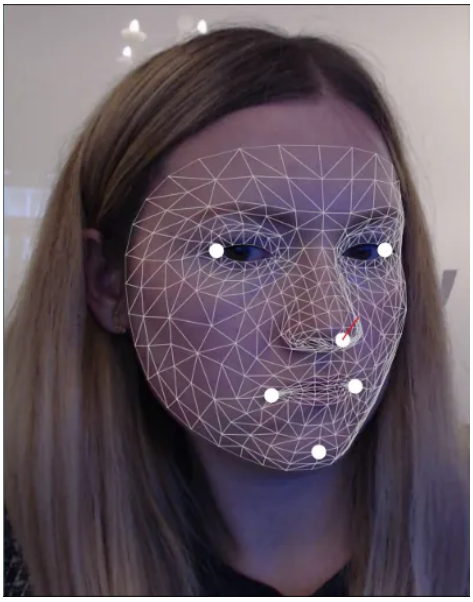
5 min read · Nov 15, 2022

90 1

In this blog post, I demonstrate how to estimate the head pose from a single image using MediaPipe FaceMesh and OpenCV in Javascript. Check out my demo on CodePen!



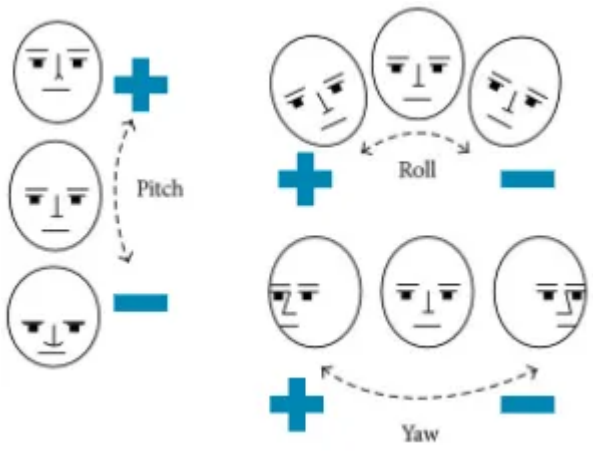




Landmarks for pose estimation

Currently, MediaPipe JavaScript Solution API does not include the option 'enableFaceGeometry' which allows obtaining the pose of each detected face automatically (see issue #2673).

In this article, I will look at the principles of pose computation from 3D-2D point correspondence, explain the relevant OpenCV algorithms, talk about landmarks in MediaPipe, and finally show you how to display the pose of the face.

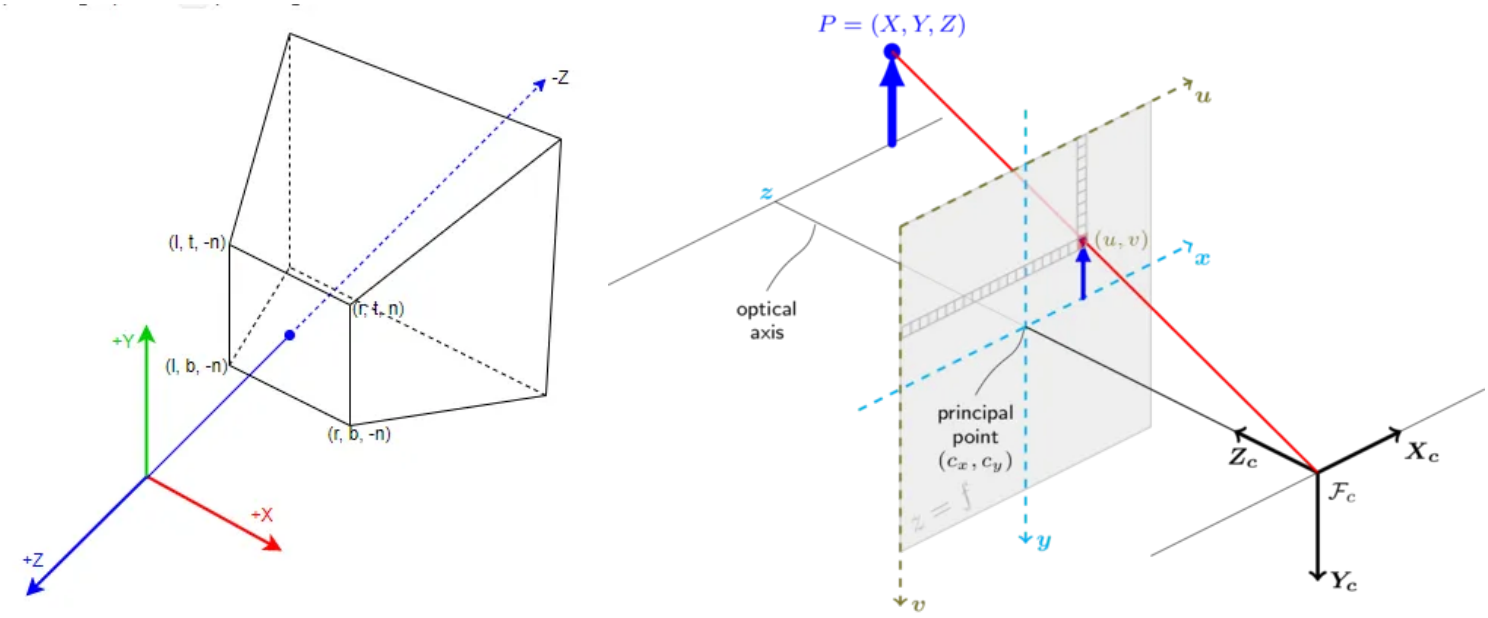


Roll, pitch, and yaw angles for head pose estimation

Steps to estimate the face's yaw, pitch, and roll angles in a given image :

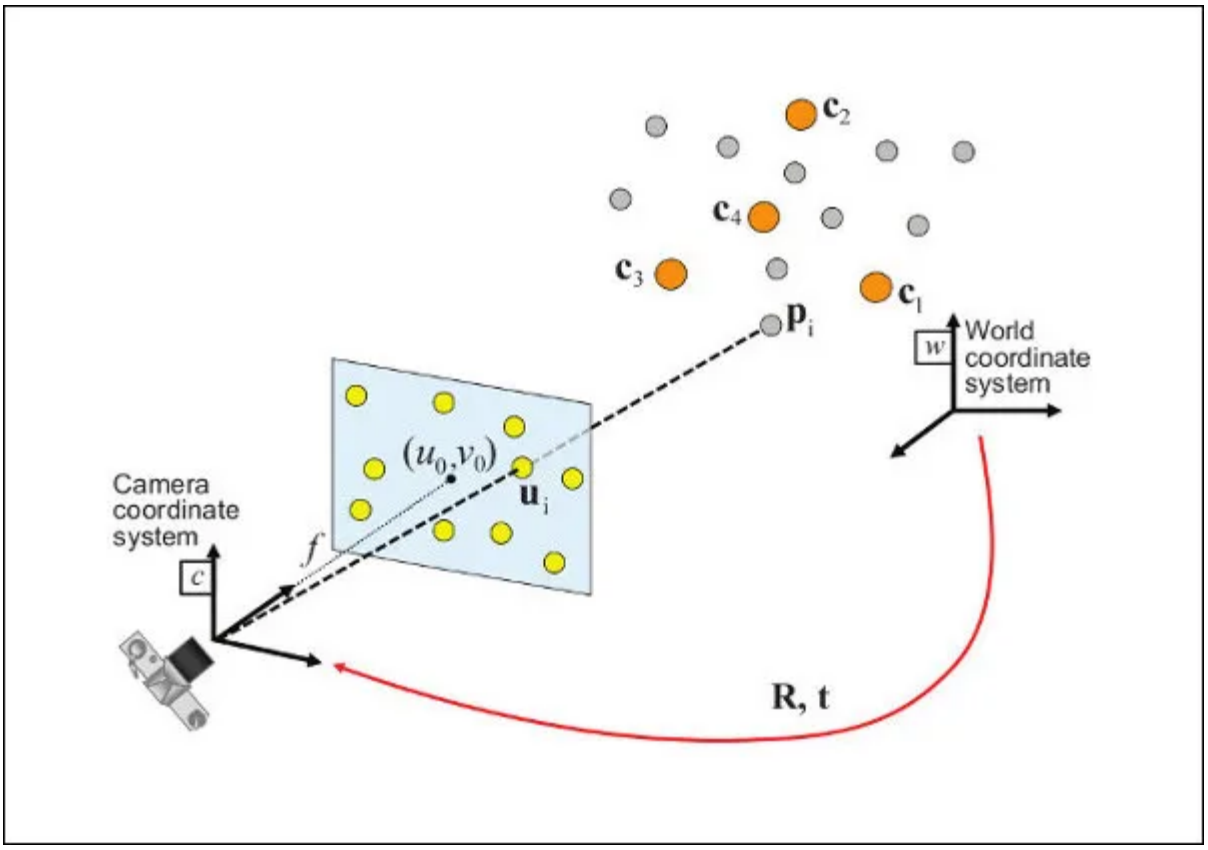
1) Find face landmarks using Mediapipe 'FaceMesh'
2) Produce rotation vector with OpenCV-Javascript ***solvePnP*** function
3) Pass rotation vector to OpenCV ***Rodrigues*** function to get rotation matrix
4) Finally, decompose the rotation matrix to get Euler angles

The **pose computation problem** consists in solving for the rotation and translation that **minimizes the reprojection error from 3D-2D point correspondences** (OpenCV, see also publication). The reprojection error is a geometric error corresponding to the image distance between a projected point and a measured one (see Wikipedia).



MediaPipe vs OpenCV coordinate systems

cv.solvePnP: Finds an object pose from 3D-2D point correspondences. This function returns the rotation **R** and the translation vectors **t** that transform a 3D point expressed in the world frame into the camera frame:



Pose computation overview (OpenCV)

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A}\,\mathbf{\Pi}\,{}^{\mathbf{c}}\mathbf{T}_{\mathbf{w}} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

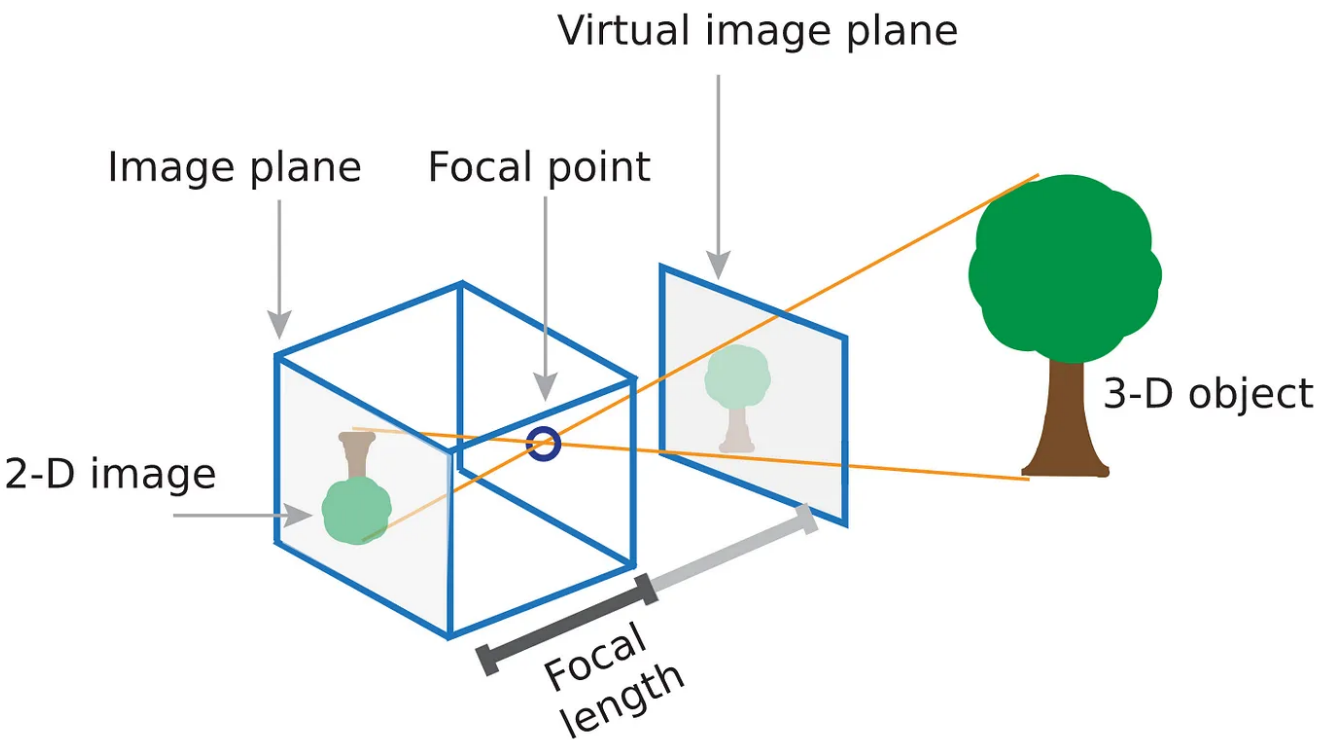
Points expressed in the world frame are projected into the image plane [u,v] using the perspective projection model **Π** and the camera intrinsic parameters matrix **A** (also denoted **K** in the literature).

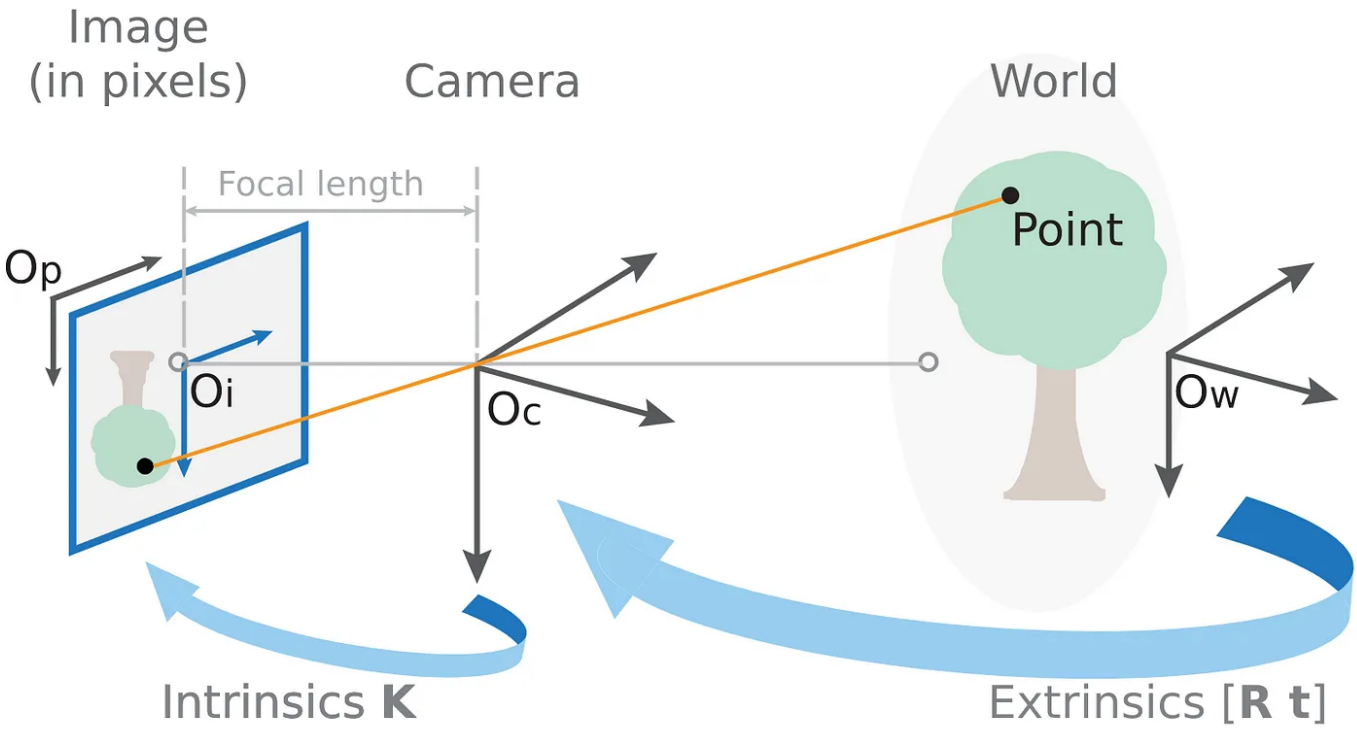
$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = {}^{c}\mathbf{T}_{\mathbf{w}} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Points expressed in the world frame are projected into the camera frame.

The camera intrinsic parameters include the focal length *f*, the optical center *c*, also known as the principal point, and the skew coefficient (read more here).





We need an array of 3D object points in the object coordinate space (landmarks of the reference model in MediaPipe), an array of corresponding image points (2D landmarks of ***multiFaceLandmarks*** in MediaPipe), intrinsic camera matrix, distortion coefficients (for simplicity we assume that there is no distortion).

Here we use 'SOLVEPNP_ITERATIVE' as a pose computation method which uses a non-linear Levenberg-Marquardt minimization scheme. The initial solution for non-planar "objectPoints" needs at least 6 points and uses the Direct linear transformation algorithm.

The output brings points from the world coordinate system to the camera coordinate system :

- Output rotation vector (see **Rodrigues**)
- Output translation vector

Rodrigues' rotation formula is an efficient algorithm for rotating a vector in space, given an **axis and angle of rotation** (Wikipedia).

***cv.Rodrigues*:** Converts a rotation matrix to a rotation vector or vice versa. Here we pass the rotation vector to get a rotation matrix.

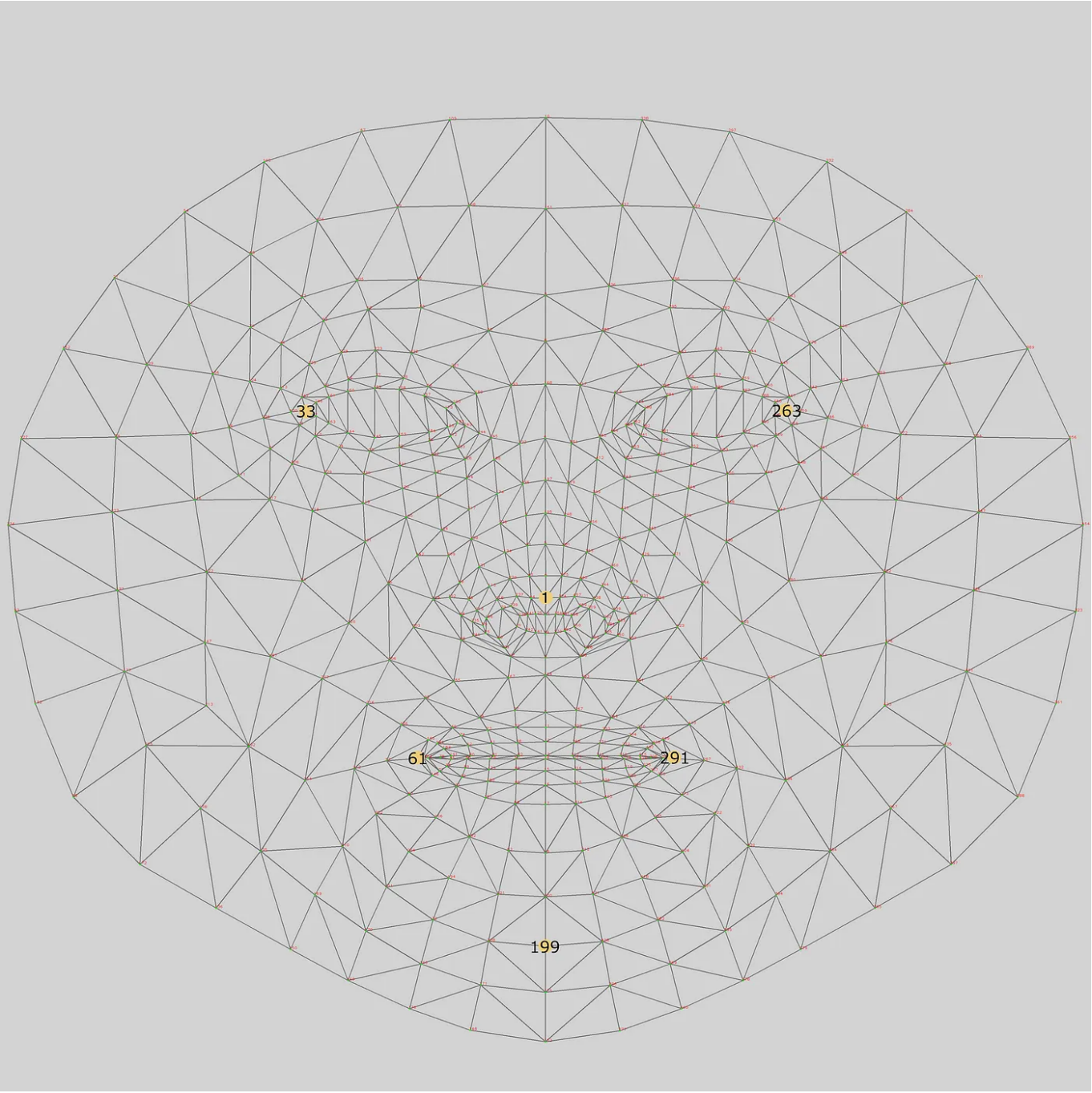
cv.RQDecomp3x3: function not found in this OpenCV version, we calculate Euler angles by hand (see rotationMatrixToEulerAngles).

```
sy = math.sqrt(R[0,0] * R[0,0] +  R[1,0] * R[1,0])

singular = sy < 1e-6

if  not singular :
    x = math.atan2(R[2,1] , R[2,2])
    y = math.atan2(-R[2,0], sy)
    z = math.atan2(R[1,0], R[0,0])
else :
    x = math.atan2(-R[1,2], R[1,1])
    y = math.atan2(-R[2,0], sy)
    z = 0
```

For each face, MediaPipe FaceMesh contains a bounding box of the detected face and an array of 468 keypoints. Each keypoint or facial landmark has 2D (x, y) or 3D (x, y, z) coordinate locations of facial features, such as lips or eyes corners, points on the eyebrows, irises, and face contours, and intermediate points on the cheeks and forehead.



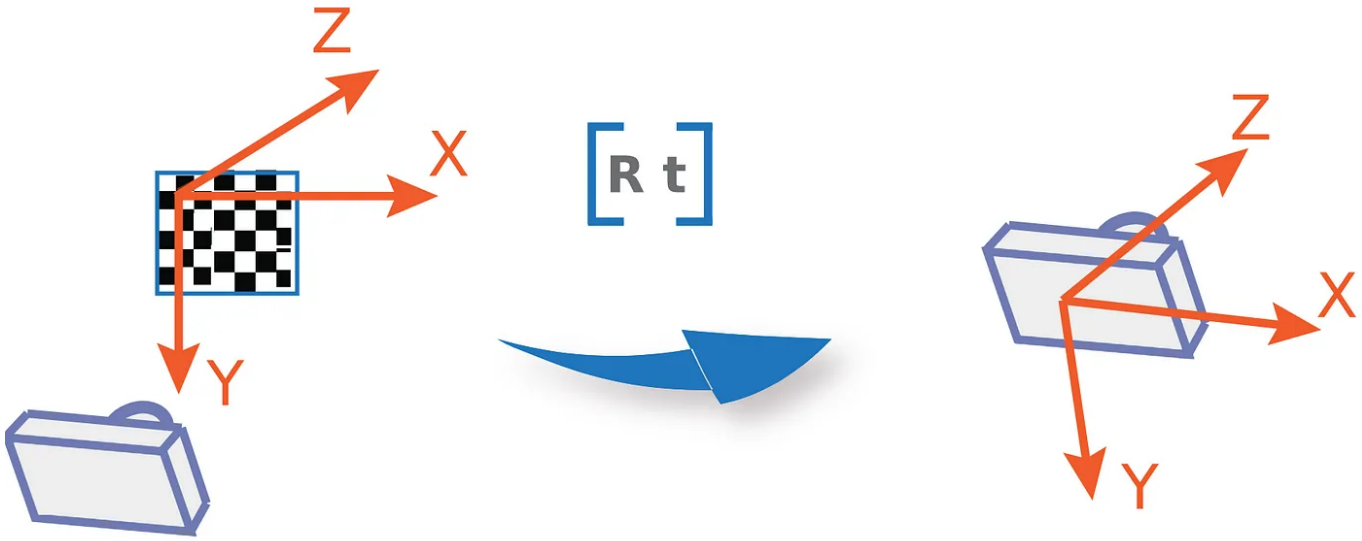
MediaPipe FaceMesh landmarks with yellow landmarks used for pose estimation (see here)

For the keypoints, x and y represent the actual keypoint position in the image pixel space. z represents the depth with the center of the head being the origin, and the smaller the value the closer the keypoint is to the camera. The magnitude of z uses roughly the same scale as x.

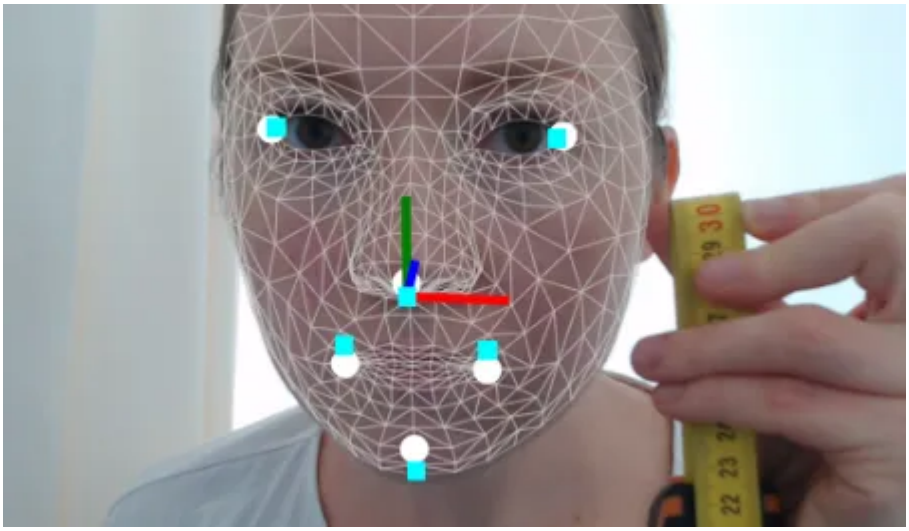
The indices of the landmarks we are interested in are 1, 33, 263, 61, 291, and 199 which are evenly distributed on the face.

To display the pose of the face as a coordinate system on the nose tip we project the nose landmark to an image plane.

and translation vector t, camera matrix K, and distortion coefficients.



Finally, we draw a line from the 2D nose tip to the projected nose tip.



Final pose estimation result with MediaPipe landmarks in white, the estimated pose as a coordinate system on the nose, and landmarks projected using the estimated pose in cyan.

References

1. GitHub repository: https://google.github.io/mediapipe/
2. Solution “MediaPipe FaceMesh” : https://google.github.io/mediapipe/solutions/face_mesh
3. Nicolai Nielson’s Demo in Python : https://github.com/niconielsen32/ComputerVision/blob/master/headPoseEstimation.py
4. Mike C. : A head pose estimation Cycle.js demo app using opencv.js and tensorflow.js’ posenet https://codesandbox.io/s/008olz2wmn?file=/src/index.js:1273-1277
5. Camera Calibration : https://fr.mathworks.com/help/vision/ug/camera-calibration.html
6. Learn OpenCV : https://learnopencv.com/head-pose-estimation-using-opencv-and-dlib/
7. Grab the code from here — https://codepen.io/Susanne-Thierfelder/pen/yLEOQXq

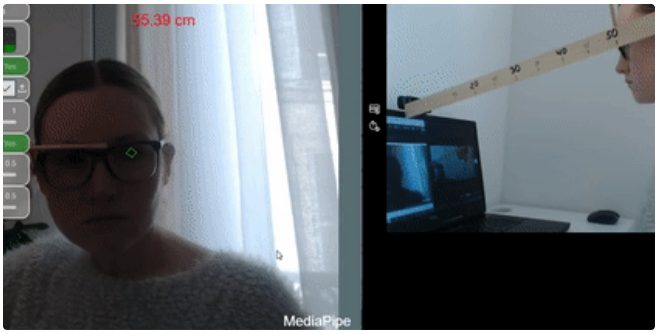
Opencv JavaScript Computer Vision Mediapipe Pose Estimation

Written by Susanne Thierfelder

12 Followers

Follow

More from Susanne Thierfelder



Susanne Thierfelder

Create your own depth measuring tool with MediaPipe FaceMesh in...

Getting started with MediaPipe and measuring depth from a single image

4 min read · Nov 3, 2022

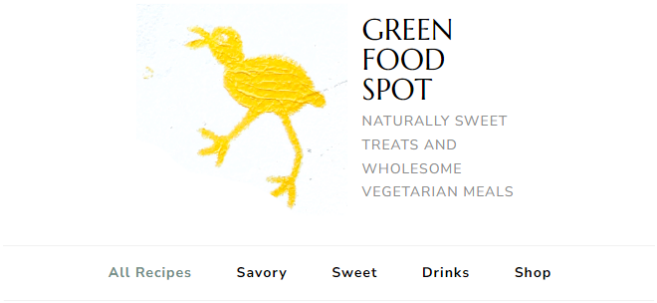
Susanne Thierfelder

Automate Monthly Wordpress.org Posts for a Food Blog without a...

Use Case: Food Blog using the Blossom Recipes Theme hosted by Bluehost.

2 min read · Oct 5

15 1



Susanne Thierfelder

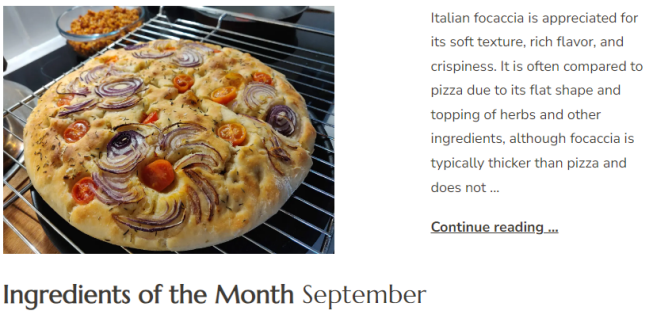
Start a Wordpress Food Blog in 2023

Are you thinking about finally turning your yearlong passion into a blog?

5 min read · Sep 19

6 2

3



Susanne Thierfelder

Automate Monthly Wordpress.org Posts for a Food Blog using a...

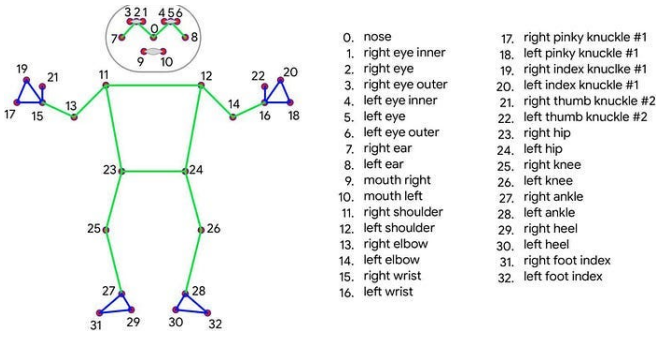
This article shows how to automate a Wordpress Blog depending on the current...

3 min read · Sep 27

3

See all from Susanne Thierfelder

Recommended from Medium



MUHAMMD TAYYAB

Pose Detection Using Mediapipe Solutions (DabMove Detection)...

Introduction

13 min read · May 11

6

Robert John

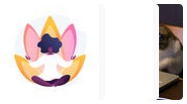
How to build a Text to Image App with free AI model in 30 lines of...

Generate amazing Image with AI, a step by step guide to building a Test-to-Image App

6 min read · May 28

9

Lists



Stories to Help You Grow as a Software Developer

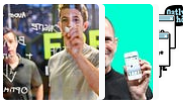
19 stories · 508 saves

It's never too late or early to start something

15 stories · 190 saves

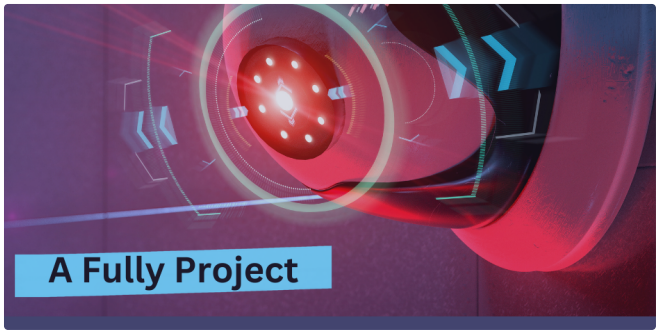
General Coding Knowledge

20 stories · 523 saves



Modern Marketing

38 stories · 226 saves



Adithya SM in AI Mind

Real-Time Hand Gesture Recognition Using OpenCV: A...

Introduction

4 min read · Jun 30

11

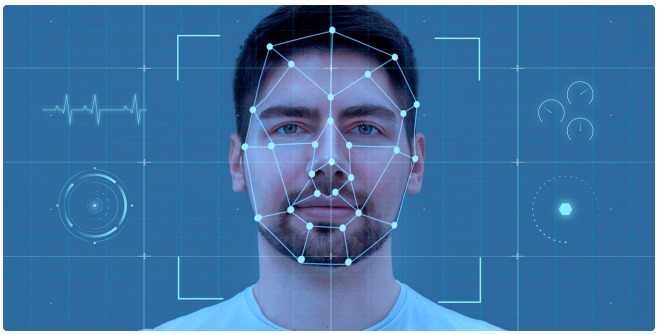
Gal Ben-Evgi

Building a Camera Recorder with Human Detection using Python a...

Build a Camera Recorder with Human Detection using Python and OpenCV:...

6 min read · Aug 1

66



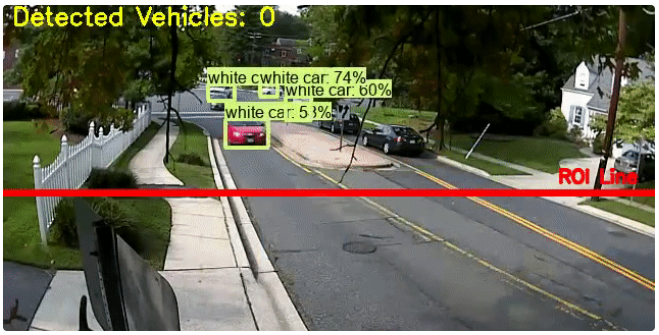
KHWAB KALRA

FACE RECOGNITION

Methods, Implementation, and Practical Examples

5 min read · Jul 11

2



Everton Gomede, PhD

Counting the Number of Objects in an Image: A Machine Learning...

Introduction

5 min read · Sep 1

22

See more recommendations