1. 3D 얼굴 시선 추정 라이브러리

1.1. 3D 얼굴 시선 추정 라이브러리 구조

3D 얼굴 시선 추정 라이브러리의 전체 구조는 그림 1 과 같이 입력된 영상에서 얼굴영역을 검출하는 얼굴검출 블록과 검출된 얼굴영역의 얼굴영상으로부터 카메라 좌표계를 기준으로 한 얼굴의 3D 시선방향을 추정하는 3D 얼굴 시선 추정 블록으로 구성되어 있다.

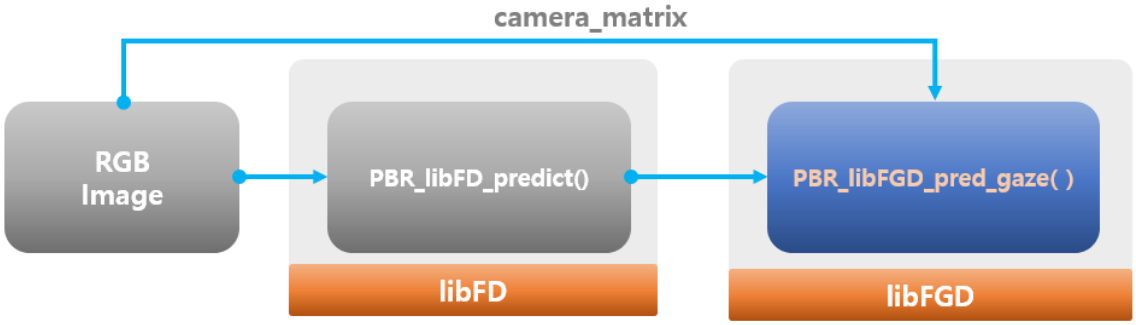

그림 1. 3D 얼굴 시선 추정 라이브러리 구성도

2. 기능별 API

2.1. 고수준 API

3D 얼굴 시선 추정 고수준 API 는 입력영상의 카메라 내부인자(Intrinsic 3x3 matrix) 정보를 기반으로 다음의 고수준 API 를 제공한다.

Function Name	PBR_FGD_predict_gaze		
Prerequisites	Pytorch 로 학습된 모델파일		
Prototype	PBR_FGD_predict_gaze(cv_img, cam_mtx)		
Input	Type	Name	Meaning
	cv2::image	cv_img	3D 얼굴 시선을 검출할 입력 영상
	numpy::array	cam_mtx	Computer Vision 에서 사용되는 3x3 카메라 내부행렬
Return	Type	Name	Meaning
	numpy::array	rmtx	입력영상의 카메라좌표계 기준의 3x3 회전행렬 정보
	numpy::array	tvec	입력영상의 카메라좌표계 기준의 3x1 이동벡터 정보
	numpy::array	lm2D	영상 좌표계 기준의 nx2 2D 랜드마크 정보
	numpy::array	lm3D	얼굴 좌표계 기준의 nx3 3D 랜드마크 정보
	numpy::array	gaze	얼굴 좌표계 기준의 3x1 3D 시선방향 정보
Remark	입력영상의 내부인자를 알고 있고, 카메라 좌표계를 기준으로 한 얼굴의 3D 시선정보를 검출하고자 할 때 사용		

2.2. 3D 얼굴 시선 검출 관련 저수준 API

입력 영상 내 얼굴 검출을 위한 저수준 API 는 다음과 같다

Function Name	PBR_libFD_predict		
Prerequisites	None		
Prototype	PBR_libFD_predict(cv_img)		
Input	Type	Name	Meaning
	cv2::image	cv_imng	얼굴 검출을 수행할 입력영상
Return	Type	Name	Meaning
	numpy::array	roi_boxes	입력영상의 얼굴 ROI 정보 [[sx, sy, ex, ey]]
Remark	입력영상의 얼굴영역을 검출하고 3D 얼굴 포즈 추정 CNN 입력에 적합한 얼굴의 사각영역을 출력		

검출된 얼굴영역 정보를 이용하여 영상 내 각 얼굴영역에 대한 3D 얼굴 시선을 검출하는 저수준 API 는 아래와 같다.

Function Name	PBR_libFGD_predict_gaze		
Prerequisites	Pytorch 로 학습된 모델파일		
Prototype	PBR_libFGD_predict_gaze(cv_img, roi_box, cam_mtx)		
Input	Type	Name	Meaning
	cv2::image	cv_imng	3D 얼굴포즈를 검출할 입력 영상
	numpy::array	roi_box	입력영상 내 얼굴영역 정보 [sx, sy, ex, ey]
	numpy::array	cam_mtx	Computer Vision 에서 사용되는 3x3 카메라 내부행렬
Return	Type	Name	Meaning
	numpy::array	rmtx	입력영상의 카메라좌표계 기준의 3x3 회전행렬 정보
	numpy::array	tvec	입력영상의 카메라좌표계 기준의 3x1 이동벡터 정보
	numpy::array	lm2D	영상 좌표계 기준의 nx2 2D 랜드마크 정보
	numpy::array	lm3D	얼굴 좌표계 기준의 nx3 3D 랜드마크 정보
	numpy::array	gaze	얼굴 좌표계 기준의 3x1 3D 시선방향 정보
Remark	입력영상의 내부인자를 알고 있고, 카메라 좌표계를 기준으로 한 얼굴의 3D 시선방향을 검출하고자 할 때 사용하며 Perspective projection model 가정하에서의 카메라 좌표계를 기준으로 한 얼굴의 3D 포즈 정보와 얼굴 좌표계를 기준으로 한 얼굴의 3D 랜드마크 정보, 3D 시선방향 정보, 영상 좌표계를 기준으로 한 얼굴의 2D 랜드마크 정보를 제공		

2.3. API 활용 예시

입력영상의 카메라 내부행렬 정보를 이용하여 카메라 좌표계를 중심으로 한 실제 공간상의 얼굴 3D 시선방향을 검출하기 위한 예제코드는 다음과 같다.

```
import cv2
import numpy as np
from FGD_api import PBR_FGD_predict_gaze
from libFGD.utils import plot_kpts, plot_axis, get_projected_points, get_projected_axis, vector_to_pitchyaw, plot_gaze


def main():

    """main function

    Note: main function for facial 3D gaze with perspective projection model

    """

    cv_img = cv2.imread('./libFGD/test_image2.png')  #ref. from BIWI dataset
    cv2.imshow('input', cv_img)
    cam_mtx = np.array([[517.679, 0, 320],
                        [0, 517.679, 240.5],
                        [0, 0, 1]])


    # test PBR_FPD_predict_persp
    success, ret = PBR_FGD_predict_gaze(cv_img, cam_mtx)
    pred_lm2D = ret['lm2D']
    pred_lm3D = ret['lm3D']
    pred_rmtx = ret['rmtx']
    pred_tvec = ret['tvec']
    pred_gaze = ret['gaze']


    out_img = cv_img.copy()
    out_img = plot_kpts(out_img, pred_lm2D, (0, 255, 0))

    axis_2D_pred = get_projected_axis(cam_mtx, pred_rmtx, pred_tvec)
    out_img = plot_axis(out_img, axis_2D_pred, 'pred')

    lm2Ds_cam = get_projected_points(cam_mtx, pred_rmtx, pred_tvec, pred_lm3D)[:, 0:2]
    out_img = plot_kpts(out_img, lm2Ds_cam, (255, 0, 0))

    pos2D_eyecenter = ((pred_lm2D[1]+pred_lm2D[2])*0.5).astype(np.int32)
    pred_gaze_py = vector_to_pitchyaw(pred_gaze*-1)
    out_img = plot_gaze(out_img, pos2D_eyecenter, pred_gaze_py, 3, (255, 255, 255))


    cv2.imshow('FGD_gaze', out_img)
    cv2.waitKey(-1)


if __name__=='__main__':
    main()
```