Instituto Superior de Engenharia de Lisboa

**Computer Vision and Mixed Reality**

**Report**

*Feature Based*

*Face Detection and Recognition for Augmented Reality*

Mestrado em Engenharia Informática e Multimédia

Rodrigo Dias, 45881

Semestre de Verão, 2021/2022

# Introduction

# Face detection

For the face detection, it was used the Cascade Classifier to detect faces and eyes. It was created a class to handle the face detection, the eye detection and the alignment needed (rotation, scale, and translation). It was needed to calibrate the behavior of the classifier to correctly detect the eyes. Otherwise, there were times where the nose would be considered as an eye for example.

# Chart, bar chart Description automatically generatedDatabase

It was created a dataset with 57 images total, with 5 different faces.

# Normalization

To normalize the images, it was used this classifier to find the faces in each image, which returns the part of the image containing the face and then apply the eyes detection. This is to better avoid detecting eyes were there’s no face. From that we obtain the coordinates of the eyes in relation to the entire image.

Afterwards, each image is downscaled to 56x46 pixels,converted to grayscale and, having the location of both eyes, we can calculate the angle of rotation and the scale factor and align the face accordingly using *warpAffine*, putting the eyes in line 26, columns 16 and 31, therefore normalizing the entire image.

Doing this to a set of images (57 in total), the dataset was prepared to apply the *eigenfaces* and *fisherfaces* algorithms.

# Train/Test split

Before applying these algorithms, the dataset was split into training and test sets, having 2 images from each class in the test set. The training set now has 47 images while the test has only 10. Then the training set was randomized to better apply the algorithms and the test set was ordered.

# Eigenfaces

# Fisherfaces

# Classification

# Combine real and virtual objects

In this step it was used the same concepts of the normalization. The idea is to have some objects with a green screen to create a mask around them, so that we can attach them to a real image.

The first step is to create the mask, placing the pixels that have a range of green to 1 and the rest to 0. The idea is to apply this mask to the original image and the inversion of this mask to the object so that we can add them together.

Once the masks are created, it was applied the concept of the normalization. For example, for the glasses, we define the location of the eyes and with that, its possible to get the scale factor and the rotation of the glasses and we just need to know the location and orientation of the eyes on the original image to make the transformation and translation.

Once the masks are in place with the original image, we multiply the masks as referred above and add them to create the final product.

A person in a suit

Description automatically generated with medium confidenceA picture containing ax

Description automatically generated

Figure 1 Original image

Figure 2 Hat

A picture containing text, clipart, vector graphics

Description automatically generated

Figure 3 Glasses

In this example, there were used two objects, glasses and a hat, and the original image, shown above, to create the final one, shown right.

Figure 4 Result