Cairo UniversityA symbol of a god

Description automatically generated with medium confidenceA logo of a company

Description automatically generated

Faculty of Engineering

Systems and Biomedical Engineering Department

Second Semester - 2023/2024

Computer Vision - Task 05

**Team #02**

Nouran Mahmoud

Nada Omran

Mohammed Sami

Kareem Salah

Youssef Ahmed

Table of Contents

[Appearance Matching 3](#_Toc166551266)

[Face recognition 3](#_Toc166551267)

[Extracting the Features. 3](#_Toc166551268)

# Appearance Matching

Appearance matching is one of the tasks that tries to recognize and represent a 3D object using 2D visual appearance. This can be done in many ways, two of them are **Template Matching** and **Parametric Appearance Representation**. Template matching/shape modelling (using explicit shapes) is known to be computationally expensive and tedious, especially in the context of generic object detection. While in visual appearance, we use the intrinsic (i.e. shape, reflectance) and extrinsic properties (i.e. illumination, pose) of the object.

## Face recognition

Face recognition problem is a subset of appearance matching problems. Having a training dataset of faces from multiple angles and with different expressions, this facilitates the recognition of a new angle/expression but creates redundancy in the . We’ll use the technique of Appearance Representation.

### Extracting the Features.

Given a **2D grayscale image** of dimensions **P x Q = N**, it can be flattened to a features vector of dimensions Nx1, representing the intensity levels in the image. This features vector contains N dimensions, each one represents image intensity at a corresponding pixel.

#### Principal Components Analysis

It’s a technique widely used for dimensionality reduction. It can be applied on images to generate a compact representation of the dataset. We shall explore its steps in the coming section.

#### Steps

1. **Creating Features Matrix.**

Given the flattened features vectors for a group of images, concatenate them into a **matrix** **(X)** of dimensions (number\_of\_subjects x number\_of\_faces\_per\_subject x N).

1. **Transform into Mean Images.**

First, calculate the mean of all the vectors in the matrix and for each vector/flattened image in the matrix, subtract the mean from it. This creates what are called mean images.

1. **Create Covariance Matrix.**

A covariance matrix is created through multiplying the mean images matrix with its transpose.

Each entry in the matrix represents the covariance between two variables, which is a measure of how much they change together. A positive covariance indicates that the variables increase or decrease together, while a negative covariance indicates that one variable increase when the other decreases.

1. **Calculating the Eigen Values and Eigen Vectors.**