

Paintings at an Exhibition

EE368 Group 17 Project Report

Mithun Kamat

Stanford University

mkamat at stanford dot edu

Abstract — An algorithm is developed and implemented to recognize paintings on display at the Cantor Arts Center based on snapshots taken with a camera-phone. The training images were used to create a database of low dimensional fisher images. The image processing was carried out in three steps. First, the training images were gray scaled and down sampled from the high resolution color patterns while preserving the key features. The images were also adjusted for brightness by normalization. Second, the training images were segmented to identify edges corresponding to the frames of the paintings. The location of the frames of the paintings was used to crop and resize the training images to a consistent scale. Finally, the training images were transformed to a lower order database using Fisher Linear Discriminant (FLD) transformation. The test image is compared against the database using Euclidean distance nearest neighbor similarity matching. The algorithm demonstrated reasonably good computational efficiency and accuracy.

This algorithm was developed as part of the project requirements for EE368 class at Stanford University. The algorithm was selected by trading off the computational and accuracy requirements for the project. The code was implemented in Matlab™ version 6.5 using the Image Processing Toolbox.

This work could be useful as part of an electronic museum guide wherein the users would point their camera-phone at a painting of interest and would hear commentary based on the recognition result.

Keywords — augmented reality, eigen images, FLD, fisher images, image processing, painting recognition

I. INTRODUCTION

IMAGE RECOGNITION is a complex problem due to the difficulty in recognizing and matching image features in the presence of variable orientation, illumination, and scale among other things. Additional complexity is involved in the case of

recognition of paintings because no two paintings are fundamentally alike unlike faces where certain facial features (e.g. nose, eyes) tend to be similar between different individuals. The applications for image recognition are numerous and growing. For example, an electronic museum guide could be envisioned where the user would point their camera-phone at a painting of interest and would hear commentary based on the recognition result. Such an application would involve processing of the image and comparing with a known database of library images. The images could vary from paintings, art displays to historical displays such as World War II artifacts. The matched image could provide the user the relevant audio-visual information about the image at hand. Applications of this kind are usually referred to as "augmented reality" applications. Implemented on hand-held mobile devices, they are called "mobile augmented reality."

The task of the project [1] is to focus on the image recognition part of the above applications. Photographs were taken in the European Gallery of the Cantor Arts Center in California. Three photos were taken of each of 33 paintings. The photos were taken with a Nokia N93 with a resolution of 3 mega pixels. The paintings did not represent any certain type and included portraits as well as landscapes. An image recognition algorithm will perform the necessary processing on the "training" images such that a new image from the same set of paintings could be identified accurately in a computationally efficient manner.

II. ALGORITHM DESCRIPTION

A. ALGORITHM SELECTION

A literature search was performed to identify the most relevant image processing algorithm. A large body of literature exists for image processing related to face recognition [2-5]. The most cited algorithms include the eigen-faces or fisher-faces method that adopt Principal

Components Analysis (PCA) and Fisher Linear Discriminant Analysis (FLD). PCA techniques, also known as Karhunen-Loeve methods [2], choose a dimensionality reducing linear projection that maximizes the scatter of all training images. A drawback of this approach is that the scatter being maximized is due not only to the between-class scatter that is useful for classification, but also to the within-class scatter that, for classification purposes, is unwanted information [3]. Fisher's Linear Discriminant (FLD) [6] is an example of a class specific method, in the sense that it tries to shape the scatter in order to make it more reliable for classification. This method selects a transformation of the raw images such a way that the ratio of the between-class scatter and the within-class scatter is maximized. Both FLD and PCA methods are not invariant to image alignment and scaling [7]. Hence, preprocessing would be involved before applying these methods. More recently, a more robust approach has been developed for feature identification. The Scale Invariant Feature Transform (SIFT) algorithm [8] detects key features that are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination.

The simplest algorithm that can be chosen for this particular problem is to do a nearest neighbor search of pair wise correlation of the image with every image in the training database. Obviously such an algorithm will require very high computational space and may be too sensitive to disturbances. Selection of the image processing algorithm involved consideration to the following elements.

- i. The images presented challenges due to low-light conditions. The algorithm had to adjust for illumination and brightness.
- b. The images were taken with a hand held camera and were quite noisy and motion blurs and/or defocus.
- c. The algorithm had to be invariant to rotation or alignment, and scaling as the images were taken from different vantage points and orientation views.
- d. The algorithm had to be computationally inexpensive as the turn around on matching had to be on the order of less than a minute.

A SIFT based image recognition method [8] would automatically cover a – c. However, the algorithm is patented (not readily usable in a commercial setting), and is not computationally efficient in Matlab without the usage of external interfaces such as MEX files, compared with other available methods. On the other hand, FLD approach seemed to be computationally very efficient and academically free to

use. Based upon the discussion provided in [1], FLD seemed more relevant than PCA. However, both those algorithms are not invariant to scaling and alignment and hence preprocessing would be required before applying the FLD transformation.

B. PROCEDURE

The proposed algorithm is made up of three steps. The step wise procedure is described in Figure 1.

0100090000037800000002001c00000000004000000030108
00050000000b0200000000050000000c02f9021a0404000000
2e0118001c000000fb021000070000000000bc020000000010
2022253797374656d00021a0400008d5c0000a835110070838
23928258b030c020000040000002d010000040000000201010
01c000000fb02ceff0000000000009001000000000440001254
696d6573204e657720526f6d616c00000000000000000000
00000000000040000002d010100050000000902000000020d
000000320a2d00000001000400000000001a04f70220d41600
040000002d010000030000000000

Figure 1: Step-wise Image Processing Algorithm for detection of paintings

In the first step, the raw training images are pre-processed to account for the variations in lighting, noise, blur and scaling. Also, images are down sampled in order to reduce the computational time. In the second step, the images are decomposed initially based upon the PCA method (Karhunen Loeve Transform – KLT) to a reduced order dimension. This step avoids singularity of the within class scatter matrix in FLD. Subsequently, the reduced order images are used to compute the transformation matrix using the FLD transformation approach. The transformation is applied on the training images and the reduced order training images are stored in the database for similarity matching. In the final step, the test images are processed using the similar approach identified in the first step, transformed using the transformation matrix identified in step two, and finally compared with the reduced order image database using the nearest neighbor Euclidean distance similarity matching method. The algorithm is coded in Matlab™ version 6.5. Commands from the Image Processing Toolbox within the Matlab library are used heavily in this algorithm.

C. IMAGE PRE-PROCESSING

1) Convert to Gray Scale

The training images obtained from the library were in the in the form of RGB array. The image is converted into gray scale with pixel values depicting intensities between 0 to 255. Figure 2 shows one of the original paintings in gray scale. The gray scale image retained the key features of the color images

without compromising the efficiency of the algorithm. The gray scale image has significant computational advantage due to lower storage requirement.



Figure 2: Gray-scale image of "The Resurrection".

2)Down Sample

The gray scale image is then down sampled to 15% of the original image size. This step was performed to avoid the computational burden of a high resolution image on a desktop PC. As will be described below, the KLT method involves the calculation of covariance matrix. Such matrix multiplication operations were found to be computationally extremely expensive especially on a desktop PC. With regard to the objective of this work, it was found that a down sampled image retained the features to a sufficient level provided the correct down sampling scheme is adopted. In this work, the image was down sampled using a bilinear interpolation. Figure 3 shows one of the paintings down sized by a factor of 0.15 using the bilinear interpolation.



Figure 3: Down sampled version of "The Resurrection"

A visual inspection of Figure 3 shows that the down sampled image has retained the key features of the original image.

3)Normalization

In order to eliminate differences with regard to contrast and brightness, the images are normalized using a dynamic range expansion. For example, if the intensity range of a gray scale image is 30 to 190, then the pixel values are first subtracted by 30 and then scaled by a factor of 255/190 in order to achieve normalization from 0 to 255. Figure 4 shows image "The Resurrection" after normalization.



Figure 4: Normalized "The Resurrection" image.

4)Isolation of the painting

This is one of the most important steps in the pre-processing algorithm. The original images were taken at different vantage points. See Figure 5a and 5b for two images of the same painting taken from a different distance and angle from each other. Clearly, the images could possibly contain the paintings as well as other background information such as corners in the building, clips of other paintings or objects. First, the key edges in the images are detected using an edge detection algorithm. In this work, the canny edge detection algorithm is used as it optimizes the ability of several types of edge detectors and also demonstrates invariance to noise. The masked image with edges is then dilated using a structuring element of a 3x3 matrix of ones in order to clearly segment the edges of the painting. Note that the edges of the painting are basically the frames of the paintings. Second, the edges corresponding to the frames of the paintings are used to segment the painting itself from the rest of the background. This is performed using a simple segmentation operation using labeling. Figure 5c and 5d show the segmented images after edge detection and labeling.