Software Requirements Specification
for
Color Image Compression
using
Two Dimensional Principle Component Analysis
(2D-PCA)

Prepared By:

Himanshu Gupta (0606810037)
Ankur Gupta (0606810020)
Amit Kumar Singh (0606810013)
Anup Kumar (0506810020)

Under the Guidance

of

MR. Ganesh Kumar Yadav
Lecturer (C.S.E. Department)

Department of Computer Science & Engineering
MEERUT INSTITUTE OF ENGINEERING & TECHNOLOGY
Approved by A.I.C.T.E.
Affiliated to U. P. Technical University, Lucknow
Meerut – 250 005, (Batch: 2006-2010)

Table of Contents :-

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

Image compression is the application of data compression on digital images. In effect, the objective is to reduce redundancy of the image data in order to be able to store or transmit data in an efficient form.

Image compression can be lossy or lossless :-

Lossless compression is sometimes preferred for artificial images such as technical drawings, icons or comics. This is because lossy compression methods, especially when used at low bit rates, introduce compression artifacts.

Lossless compression methods may also be preferred for high value content, such as medical imagery or image scans made for archival purposes. Lossy methods are especially suitable for natural images such as photos in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate. The lossy compression that produces imperceptible differences can be called visually lossless.

1.2 Document Conventions

The following terms are used throughout the SRS:-
PCA :- Principle Component Analysis
2DPCA :- Two Dimensional Principle Component Analysis
DiaPCA :- Diagonal Principle Component Analysis
$(2D)^2$ PCA :- 2-Directional 2-Dimensional PCA

1.3 Intended Audience

The intended audience for this project are the head of department, the faculties of Department of Computer Science & Engineering, Meerut Institute of Engineering & Technology and the developers of this project.

1.4 Project Scope

This project can be used during image transmission for compressing the image before transmission & decompressing the image after transmission. This project can also be used for storing the image in the hard disk to reduce the space requirements for storing.

1.5 References

1. Jian Yang; Zhang, D.; Frangi, A.F.; Jing-yu Yang "Two-dimensional PCA: a new approach to appearance-based face representation and recognition" Pattern Analysis and Machine Intelligence, IEEE Transactions on, Volume: 26,Issue: 1, Jan 2004 Pages: 131-137

2. Daoqiang Zhang, Zhi-Hua Zhou and Songcan. Chen, "Diagonal Principal Component Analysis for Face Recognition," Pattern Recognition, 2006, 39(1):140-142

3. Daoqiang Zhang and Zhi-Hua Zhou. "(2D-2D)PCA: 2-directional 2-dimensional PCA for efficient face representation and recognition," Neurocomputing, 2005, 69: 224-231

2. Overall Description

2.1 Product Perspective

Data volume reduction is a common task in image processing. There is a huge amount of algorithms based on various principles leading to the image compression. Algorithms based on the image color reduction are mostly lossy but their results are still acceptable for some applications. The image transformation from color to the gray-level (intensity) image belongs to the most common algorithms.

The above proposed problem overlaps with a common problem in Statical pattern recognition, feature selection or extraction. Feature selection is a process where by data is transformed into feature space having same dimension as original data. However data is transformed in such a way that data is represented by reduced number features effectively represent data retaining most of information. This property of transformation makes its application in pattern recognition, detection an decompression. One of such technique is principal component analysis.

2.2 Product Features

Two dimensional principal component analyses (2DPCA) is recently proposed technique for face representation and recognition. The standard PCA works on 1-dimensional vectors which has inherent problem of dealing with high dimensional vector space data such as images, whereas 2DPCA directly works on matrices i.e. in 2DPCA, PCA technique is applied directly on original image without transforming into 1 dimensional vector. This feature of 2DPCA has advantage over standard PCA in terms of dealing with high dimensional vector space data. In this paper a working principle is proposed for color image compression using 2DPCA. Several other variants of 2DPCA are also applied and the proposed method effectively combines several 2DPCA based techniques.

2.3 User Classes & Characteristics

The users of this project are all the computer users. Since now a days every person deals with images and images take lots of space for storage, everybody need a way to store the image in less space.

2.4 Operating Environment

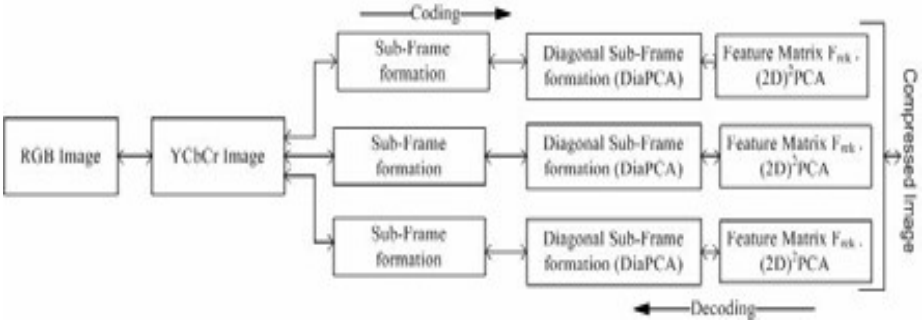
2.4.1 Hardware Environment

C.P.U. : Intel Pentium-4 or more
Hard Disk : 5 GB
RAM : 256 MB

2.4.2 Software Environment

1. GNU/Linux O.S.
2. GNU Compiler Collection
3. Intel OpenCV Library
4. GTK Library

2.5 Application Workflow



2.6 Assumptions & Dependencies

It is assumed that GNU Compiler Collection, Intel's OpenCV Library & the GTK library are pre-installed over the GNU/Linux Operating System.

3. System Features

3.1 Image Compression

This module will be used for compressing the image using 2DPCA technique. It will take an uncompressed image & then with the help of 2DPCA technique, it will convert the image into feature vectores and will store the image in the form of these feature vectors.

3.2 Image Decompression

This module will be used for decompressing the image using 2DPCA technique. This module will take the images stored in the form of feature vectors and then will convert the image back into original image.

4. External Interfae Requirements

4.1 User Interfaces

The user interface will be built using gtk library. GTK is a highly usable, feature rich toolkit for creating graphical user interfaces which boasts cross platform compatibility and an easy to use API.

The interface will be used for two modules of the project:-

1. Image Compression:-

For Image Compression, the user interface will have an address bar for selecting an image for compression and one button to perform image compression. After pressing the compress button, the software will ask for storage location of the image. After telling the url of the location, the image will be stored in compressed form at the given location.

2. Image Decompression:-

For Image Decompression, the user interface will have an address bar for selecting the compressed image for decompression and one button to perform image decompression. After pressing the decompress button, the software will display the decompressed image in a new window.

4.2 Software Interfaces

The following softwares will be used for working on this project:-

1. GCC:-

The GNU Compiler Collection (usually shortened to GCC) is a compiler system produced by the GNU Project supporting various programming languages. GCC is a key component of the GNU toolchain. As well as being the official compiler of the GNU operating system, GCC has been adopted as the standard compiler by most other modern Unix-like computer operating systems, including GNU/Linux, the BSD family and Mac OS X.

2. GTK:-

GTK is a highly usable, feature rich toolkit for creating graphical user interfaces which boasts cross platform compatibility and an easy to use API.

3. OpenCV:-

OpenCV is a computer vision library originally developed by Intel. It is free for use under the open source BSD license. The library is cross-platform. It focuses mainly on real-time image processing, as such, if it finds Intel's Integrated Performance Primitives on the system, it will use these commercial optimized routines to accelerate itself.

5. Performance Requirements

The performance of this software can be calculated by the following formula:-

$$CR=\frac{\alpha}{\beta\cdot\mu+\theta+\varphi}$$

where, CR=Compression Ratio
α=size of frame of an image
β=size of feature matrix F
μ=no. of sub-frames in a frame
θ=size of row projection matrix X
Φ=size of column projection matrix Z

As the compression ratio increases, the size of the image as well as the quality of the image will reduce. The compression ratio should be selected such that there is a tradeoff between the image size & its quality.