PCA Analysis using Core Image

Daniel Beatty

May 18, 2007

1 Introduction

The core essence of Principle Component Analysis is in finding a set of orthogonal basis vectors for a collection of vectors assembled in a matrix. Such a basis allows the original data to mapped into this orthogonal space. There are well studied methods for obtaining principle components such as Karhunen-Loeve and Singular Value Decomposition. While it is true that they each obtain equivalent results, how they do it determines their effectiveness at achieving these results. Also, some methods have side-products which may be useful.

In the end, the principle components are simply defined in terms of eigenvectors and their eigenvalues for a "scatter" matrix. These vectors happen to belong to the columns of the V matrix of the SVD $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$. Likewise, the eigenvalues of the scatter matrix are part of the diagonal matrix Λ. Since the SVD was part of the LAPACK wrappers of the dcgRenaissance, this logical choice of implementation for acquiring the SVD.

This experiments original instructions was to use Principal Component Analysis (PCA) in combination with the Whitening transform and Linear Discriminant Analysis (LDA) to classify and segment the optic disc from the background. The PCA by itself is not a classifier. Rather it finds orthogonal vectors that a representative of the data. Applying the principle components to the data simply transforms the data into another orthogonal space. Also, Fisher's Linear/Multiple Discriminant Analysis depends on previous samples for each desired class.

For demonstration purposes, the points in question are preselected on a pixel coordinate system. The pixels for each class are therefore used in determining the LDA. Accuracy of the LDA does have some dependency on the quality of the samples chosen. The chosen test image happens to be a classic image of a human optic disc. Fortunately, the disc is in the center of the image. Thus one choice for member of the disc would be all pixels in a radius of 25 pixels from the center. Pixels close to borders of the image are definitely not part of the optic disc. Thus we have a collection samples to call our class samples.

It it were not for this convenient example, a more elaborate interface would be required to identify samples belong to specific classes. This is useful in an application, but not a unit test type experiment. Such region of interest exercises, even with Core Image, are deemed beyond the scope of this exercise.

Many parameters may be considered for PCA. A single point itself is not particularly interesting, as such an eigenvector would be a scalar. However, neighborhoods of both the image itself, and transformations such a gradients, edge transformations, and others tend to be quite interesting for this experiment.

The neighborhood concept comes from digital image processing techniques of segmentation called Local Processing Edge Linking and Boundary Detection. In this technique, boundaries are determined by the image's gradient in row wise, column wise, and in some cases diagonal wise. From any of these, a set of row vectors may be formed consisting the values for the

point itself in both the original image and resulting images from the application of gradient and other operators. In addition the row vector can include neighborhood values, differences, angles, and angle differences. At which point one has a substantial set of parameters to categorize any point. The use of PCA allows these parameters to be mapped to an orthogonal space, which in some cases allows simplified categorization.

In this case, we are using 5 by 5 neighborhoods of the gradient differences, angle differences, the original image neighborhood, and gradient row and column neighborhoods. While this is certainly excessive, it does allow for a demonstration.

$$\mathbf{A} = \begin{Bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{Bmatrix} \quad (1)$$

$$\mathbf{U} = \begin{Bmatrix} -0.13 & 0.82 & -0.54 & 0.03 \\ -0.34 & 0.42 & 0.75 & 0.36 \\ -0.54 & 0.03 & 0.13 & -0.8 \\ -0.75 & -0.36 & -0.341 & 0.42 \end{Bmatrix} \quad (2)$$

$$\mathbf{\Lambda} \begin{Bmatrix} 38 & 0. & 0. & 0. \\ 0. & 2.07 & 0. & 0. \\ 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. \end{Bmatrix} \quad (3)$$

$$\mathbf{V}^T = \begin{Bmatrix} -0.42 & -0.71 & -0.16 & -0.52 \\ -0.47 & -0.27 & 0.60 & 0.57 \\ -0.52 & 0.17 & -0.72 & 0.41 \\ -0.56 & 0.61 & 0.28 & -0.47 \end{Bmatrix} \quad (4)$$

$$\text{SVD}(\mathbf{A}) = U\Lambda V^T \quad (5)$$

$$V = \text{eigenvectos}(\mathbf{A}\mathbf{A}^T) \quad (6)$$

$$\Lambda = \text{eigenvalues}(\mathbf{A}\mathbf{A}^T) \quad (7)$$

The values for $\mathbf{U}$, $\mathbf{\Lambda}$, and $\mathbf{V}$. are computed using Mathematica's pre-canned Singular Value Decomposition method. A comparison between this and the unit test can be valuable.

Similarly, the results from the MDA assignment using Mathematica can be used as a unit test for the LDA/ MDA component. The results are expected to be similar, within a scaling factor.

2 Unit Test of PCA

The experiment requested for this project is significantly more complex than what a unit test should test. Any unit test should have a simple input, simple output, and known comparison for the output. An example of this would be say a matrix

3 Interface design

In order to complete the assignment, a tool is necessary that uses this build framework, identifies the regions of interest for the MDA/LDA preprocessing, and neighborhoods, and use these libraries to produce the principle components. The results from the preprocess operations need to be fed into a mask generating classifier.

numberOfClassificationsSlider

Region of Interest Control:

- List Control

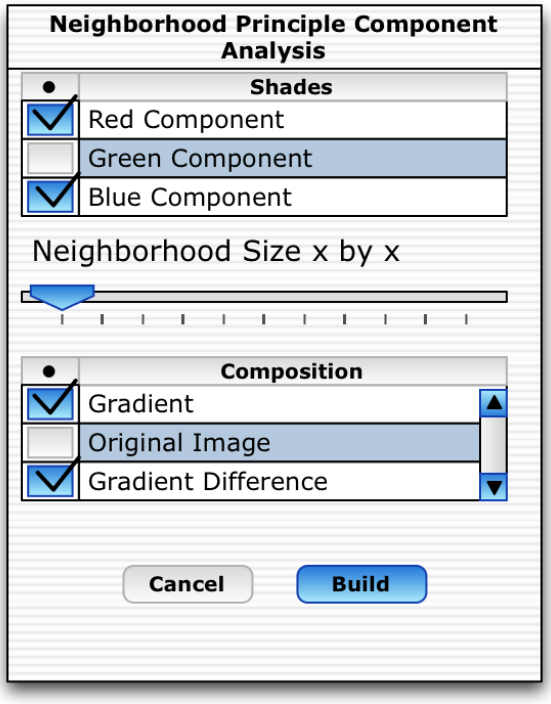

Figure 1: This panel selects the neighborhood, and components to use.

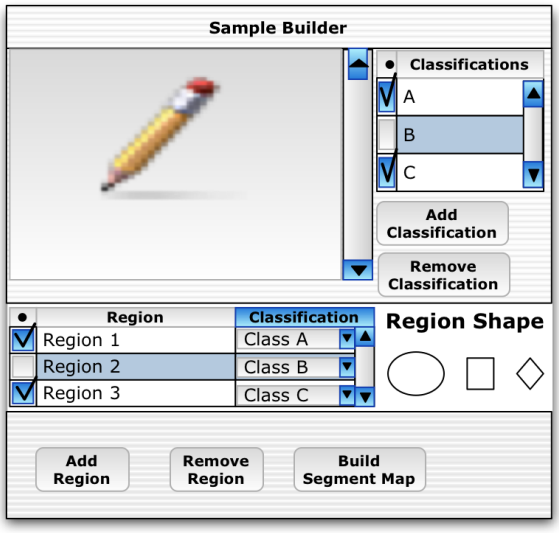

Figure 2: This panel selects the regions of interest, and instructs the building of classes.

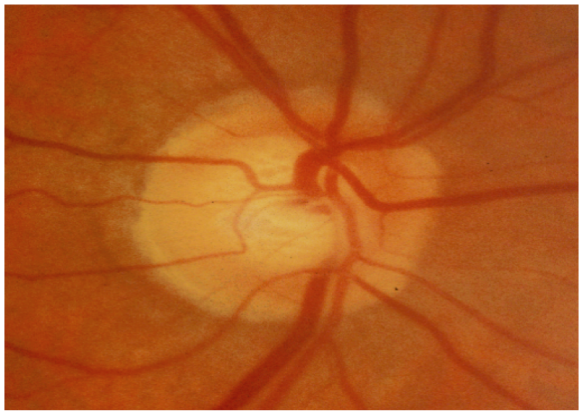

Figure 3: Original image of the optic disc of a human eye.

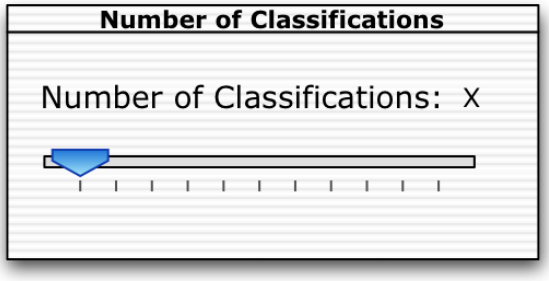

Figure 4: Slider panel for specifying the number of classifications.

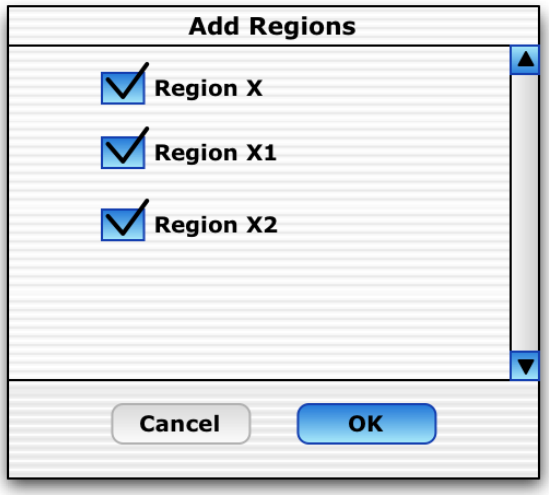

Figure 5: Table for selecting regions to add to a classification.

Non-kernel based Region of Interest View applies to a view that displays both the image and boxes identifying regions of interest. A region of interest contains points known to belong to a classification.

- ROIView → OpenGLView? QuartzView?
- Initialize called by either the opening of a nib file or object initialization.
  - "initWithFrame"
  - "awakeFromNib"
- Drawing methods
  drawRect
- What do we need:
  - Need ability to identify selected ROI.
  - Overrides on mouse and arrow.

Mask Selection:

- Selection of masks (maskControl)

Neighborhood Builder

- Size
- Component
- Composition
  - gradient
  - Original Image
  - Gradient difference
  - Gradient Angle
  - Gradient Angle Difference

Neighborhood Control

- Component List
- (Action) Construct Neighborhoods

Bayes Gaussian Multivariate Control Control

- Multivariate Classes
- Classifier List
  - Name
  - Member regions
  - (Action) add classification
  - (Action) add region to classification.

References