COC202 Computer Vision
Lab 6 – PCA/MDS for image database visualisation

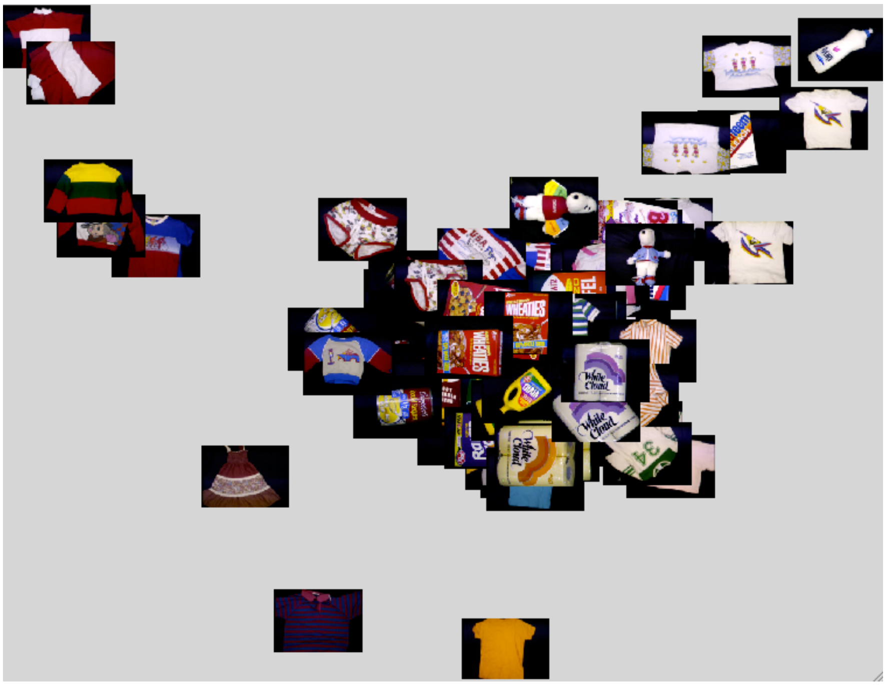
In this lab, you will first implement some code to perform principal component analysis (PCA). You will then use this, as well as a Matlab implementation of MDS, for content-based image database visualisation.

If you have not yet finished the exercises from the previous lab, do them first.

1. Write a function that is passed a data matrix (samples as row vectors) and performs PCA (i.e., calculates the covariance matrix, computes the (sorted) eigenvectors of the matrix, and projects the original data onto the space spanned by the principal components). The function should return the principal components (in matrix form with each PC as a column vector), the corresponding eigenvalues, and the original data projected onto the new space.

   Matlab functions that should be useful here: `cov`, `eig`.

2. Use your PCA function to perform image database visualisation (of the images from the previous lab) based on colour histograms (i.e. extract the colour histograms for all database images, build a data matrix from that where each row vector holds a (reshaped) colour histogram, run PCA on this matrix, and project all images onto the space spanned by the first two principal components). Your results should look similar to the following:

   Hint: `A(:)` returns the data in matrix A as a vector. You can also use Matlab's `reshape()` function which is more versatile.

   Matlab functions that should be useful here: `axes` (for creating a new figure axis to display an image).

Once you have finished all exercises you may leave the lab.

Additional exercises for further study:

3. Next, we want to perform image database visualisation using multi-dimensional scaling (MDS). This is somewhat more involved to implement, but luckily Matlab's Statistics Toolbox provides an implementation of MDS in the form of its `mdscale()` function. Using this function (the default settings/parameters will be fine), visualise the image database.

   Note, that for the MDS function to work correctly it needs to be passed a distance matrix, while your histogram intersection function will return a similarity measure rather than a distance.

   Your result should look similar to the following: