

COC202 Computer Vision

Lab 6 – PCA/MDS for image database visualisation – Solutions

1.

```
function [pcs, evals, projdata] = mypca(data)

c = cov(data); % covariance matrix
[v d] = eig(c); % get eigenvectors
d = diag(d);
[~, ind] = sort(d, 'descend'); % sort eigenvalues
pcs = v(:,ind);
evals = d(ind);
projdata = data * pcs; % project onto PCA space
```

2.

```
% PCA image database visualisation based on colour histograms

% create image datastore
imds = imageDatastore('.', 'FileExtensions', '.bmp');
imgs = readall(imds); % read in all images

% create colour histograms
for i=1:length(imgs)
    ch = colourhist(imgs{i});
    allhists(i,:) = ch(:); % reshape into vector and add to data matrix
end

% do PCA on data
[pcs evals projdata] = mypca(allhists);
projimgs = projdata(:,1:2);

% normalise co-ordinates into [0;1] range
coords = normalize(projimgs, 'range');

% reduce range to [0;0.9] so that we have space to plot in [0;1] range
coords = 0.9 * coords;

figure
for i=1:length(imgs)
    axes('pos', [coords(i,1) coords(i,2) 0.1 0.1]);
    imshow(imgs{i});
end
```

3. (code that is different from that for Exercise 2 is bolded)

```
% MDS image database visualisation based on colour histograms

% create image datastore
imds = imageDatastore('.', 'FileExtensions', '.bmp');
imgs = readall(imds); % read in all images

% create colour histograms
for i=1:length(imgs)
    ch = colourhist(imgs{i});
    allhists(i,:) = ch(:); % reshape into vector and add to data matrix
end

% calculate distance matrix
dists = zeros(length(imgs)); % square dist matrix
for i=1:length(imgs)
    for j=i+1:length(imgs)
        dists(i,j) = 1-histint(allhists(i,:),allhists(j,:));
        dists(j,i) = dists(i,j); % matrix is symmetric
    end
end

% do MDS on data (use toolbox function)
[projimgs stress] = mdscale(dists, 2); % same as mdscale(dists, 2,
'criterion', 'stress');
stress % print stress

% normalise co-ordinates into [0;1] range
coords = normalize(projimgs, 'range');

% reduce range to [0;0.9] so that we have space to plot in [0;1] range
coords = 0.9 * coords;

figure
for i=1:length(imgs)
    axes('pos', [coords(i,1) coords(i,2) 0.1 0.1]);
    imshow(imgs{i});
end
```