COC202 Computer Vision

Lab 9 – Colour constancy and colour invariance – Solutions

1.

```
% QBE with colour histograms on images taken under various
lights/devices

imds = imageDatastore('*.tiff'); % create image datastore
imgs = readall(imds); % read in all images

for i=1:length(imgs)
    disp(sprintf('%2d - %s', i, imds.Files{i}));
    allhists(i,:,:,:) = colourhist(imgs{i});
end
sel = input('Select query image by number: ');

qhist = allhists(sel,:,:,:);
for i=1:length(imgs)
    mhist = allhists(i,:,:,:);
    sim(i) = histint(qhist, mhist);
end

[d, ind] = sort(sim, 'descend');
figure
for i=1:length(ind)
    subplot(10,10,i);
    imshow(imgs{ind(i)});
end
```

2.

```
function [image_GW] = rgb2gw(image_RGB)

% Greyworld colour constancy & white balancing

image_RGB = im2double(image_RGB);

illum = squeeze(mean(image_RGB, [1 2])); % estimate illuminant as mean
of channels
illum = illum / norm(illum, 2); % normalise to unit length

for i=1:size(image_RGB, 3)
       image_GW(:,:,i) = min(1, (1/sqrt(3))/illum(i) *
image_RGB(:,:,i));
end

image_GW = im2uint8(image_GW);
```

3.

```
% QBE with Greyworld + colour histograms

imds = imageDatastore('*.tiff'); % create image datastore
imgs = readall(imds); % read in all images

for i=1:length(imgs)
    disp(sprintf('%2d - %s', i, imds.Files{i}));
    allhists(i,:,:,:) = colourhist(rgb2gw(imgs{i}));
end
sel = input('Select query image by number: ');

qhist = allhists(sel,:,:,:);
for i=1:length(imgs)
    mhist = allhists(i,:,:,:);
    sim(i) = histint(qhist, mhist);
end

[d, ind] = sort(sim, 'descend');
fo = figure;
fn = figure;
for i=1:length(ind)
    figure(fo)
    subplot(10,10,i);
    imshow(imgs{ind(i)}); % show original image
    figure(fn)
    subplot(10,10,i);
    imshow(rgb2gw(imgs{ind(i)})); % show image after Greyworld
end
```

4.

```
function [image_maxrgb] = rgb2maxrgb(image_RGB)

% MaxRGB colour constancy & white balancing

image_RGB = im2double(image_RGB);

illum = squeeze(max(image_RGB, [], [1 2])); % estimate illuminant as
max of channels
illum = illum / norm(illum, 2); % normalise to unit length

for i=1:size(image_RGB, 3)
       image_maxrgb(:,:,i) = min(1, (1/sqrt(3))/illum(i) *
image_RGB(:,:,i));
end

image_maxrgb = im2uint8(image_maxrgb);
```

5.

```
% QBE with MaxRGB + colour histograms

imds = imageDatastore('*.tiff'); % create image datastore
imgs = readall(imds); % read in all images

for i=1:length(imgs)
    disp(sprintf('%2d - %s', i, imds.Files{i}));
    allhists(i,:,:,:) = colourhist(rgb2maxrgb(imgs{i}));
end
sel = input('Select query image by number: ');

qhist = allhists(sel,:,:,:);
for i=1:length(imgs)
    mhist = allhists(i,:,:,:);
    sim(i) = histint(qhist, mhist);
end

[d, ind] = sort(sim, 'descend');
fo = figure;
fn = figure;
for i=1:length(ind)
    figure(fo)
    subplot(10,10,i);
    imshow(imgs{ind(i)}); % show original image
    figure(fn)
    subplot(10,10,i);
    % imshow(imgs{ind(i)}); % show original image
    imshow(rgb2maxrgb(imgs{ind(i)})); % show image after MaxRGB
end
```

6.

```
function [image_eq] = rgb2histeqrgb(image_RGB)

% Colour normalisation through histogram equalisation

nLevels = 255;
for i=1:size(image_RGB, 3)
       image_eq(:,:,i) = histeq(image_RGB(:,:,i), nLevels);
end
```

7.

```
% QBE with histogram equalisation + colour histograms

imds = imageDatastore('*.tiff'); % create image datastore
imgs = readall(imds); % read in all images

for i=1:length(imgs)
    disp(sprintf('%2d - %s', i, imds.Files{i}));
    allhists(i,:,:,:) = colourhist(rgb2histeqrgb(imgs{i}));
end
sel = input('Select query image by number: ');

qhist = allhists(sel,:,:,:);
for i=1:length(imgs)
    mhist = allhists(i,:,:,:);
    sim(i) = histint(qhist, mhist);
end

[d, ind] = sort(sim, 'descend');
fo = figure;
fn = figure;
for i=1:length(ind)
    figure(fo)
    subplot(10,10,i);
    imshow(imgs{ind(i)}); % show original image
    figure(fn)
    subplot(10,10,i);
    % imshow(imgs{ind(i)}); % show original image
    imshow(rgb2histeqrgb(imgs{ind(i)})); % show histogram equalised
image
end
```