EECE 5639- Homework 4

Sreejith Sreekumar: 001277209

October 19, 2018

Question 1

(a) Planar projective transformation in homogeneous coordinates:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} p \\ q \\ 1 \end{bmatrix} \tag{1}$$

(b) There are 8 degrees of freedom for this transformation.

(c) 4 point correspondences are required for this transformation.

(d) Yes. If more points are available, the caliberation necessary for improving the accuracy can be done more precisely (eg. Using linear least squares).

(e) Collinearity, order of contact, Tangent discontinuities and cusps, cross ratio of 4 collinear points etc. are some invarients in planar projective transformation.

(f) Ratio of lengths on collinear or parallel lines, ratio of areas etc. are some invarients.

(g) Both sets have their vanishing point at infinity. In this case the building will be fronto-parallel.

Question 2

```
% create the image
image = zeros(8,8);
image(6,1) = 1;
image(7,2) = 1;
image(8,2) = 1;
image(6,3) = 1;
image(2,2) = 2;
image(6,2) = 2;
image(2,4) = 2;
image(6,6) = 2;
image(4,7) = 2;
image(5,7) = 2;
image(6,8) = 2;
image(3,2) = 2;
image(2,3) = 4;
image(4,3) = 2;
image(6,7) = 4;


%create the patch
template = zeros(3,3);
template(1,1) = 1;
```

```
template(1,3) = 1;
template(2,2) = 1;
template(3,2) = 1;
template(1,2) = 2;
```

(a) Calculation of SSD

```
% calculation of SSD
f = image;
g = template;
ssd = zeros(8,8);

for i = 2:7
    for j = 2: 7
           var  = (f(i-1, j-1) - g(1,1)).^2 + ...
                + (f(i-1, j) - g(1,2)).^2+ ...
                + (f(i-1, j+1) - g(1,3)).^2 + ...
                + (f(i, j-1) - g(2,1)).^2 + ...
                + (f(i,j) - g(2,2)).^2 + ...
                + (f(i,j+1) - g(2,3)).^2 + ...
                + (f(i+1,j-1) - g(3,1)).^2 + ...
                + (f(i+1,j) - g(3,2)).^2 + ...
                + (f(i+1,j+1) - g(3,3)).^2;
       ssd(i,j) = sum(sum(var));
    end
end

ssd(1,:) = 0;
ssd(:,1) = 0;
ssd(8,:) = 0;
ssd(:,8) = 0;

disp("SSD is:");
disp(ssd)
```

```
SSD is:
     0     0     0     0     0     0     0     0
     0    24    28    24    12     8     8     0
     0    16    12    16     8    12     8     0
     0     8     8    12     8    16     8     0
     0    10     7     9    12    28    20     0
     0     9    12     9    12    24    20     0
     0     0     7     7     8    12     8     0
     0     0     0     0     0     0     0     0
```

(b) Calculation of correlation

```
cross_correlation_score = imfilter(image,template);
cross_correlation_score(1,:) = 0;
cross_correlation_score(:,1) = 0;
cross_correlation_score(8,:) = 0;
cross_correlation_score(:,8) = 0;
disp("Cross Correlation score");
disp(cross_correlation_score);
```

```
Cross Correlation score
```

```
     0     0     0     0     0     0     0     0
     0     4     4     2     0     0     0     0
     0    10    14     8     2     0     2     0
     0     4     4     0     0     0     4     0
     0     4     5     2     0     4    10     0
     0     3     1     0     0     4     8     0
     0     8     4     1     2     8    12     0
     0     0     0     0     0     0     0     0
```

(c) Calculation of Normalized Cross Correlation

```
normalized_image = zeros(8,8);
normalized_template = zeros(3,3);

% normalizing the f matrix - f/||f||
for i=2:7
    for j=2:7
      denominator = f(i-1, j-1) .^2 + ...
                  + f(i-1, j) .^2 + ...
                  + f(i-1, j+1) .^2 + ...
                  + f(i, j-1) .^2 ...
                  + f(i,j) .^2 ...
                  + f(i,j+1) .^2 ...
                  + f(i+1,j-1) .^2 ...
                  + f(i+1,j) .^2 ...
                  + f(i+1,j+1) .^2;
       denominator = sqrt(denominator);
       if denominator == 0
        normalized_image(i,j) = 0;
       else
        normalized_image(i,j) = (f(i,j) / denominator);
       end
    end
end

% normalizing the g matrix - g/||g||
normalized_template = g./sqrt(sum(sum(g.^2)));
normalized_cross_correlation = imfilter(normalized_image, normalized_template);

normalized_cross_correlation(1,:) = 0;
normalized_cross_correlation(:,1) = 0;
normalized_cross_correlation(8,:) = 0;
normalized_cross_correlation(:,8) = 0;

disp("Normalized Cross Correlation");
disp(normalized_cross_correlation);
```

```
Normalized Cross Correlation
         0         0         0         0         0         0         0         0
         0    0.2887    0.2673    0.1581         0         0         0         0
         0    0.6682    0.8750    0.5774    0.3536         0    0.3536         0
         0    0.5000    0.5000         0         0         0    0.5000         0
         0    0.4472    0.5893    0.3162         0    0.2673    0.6250         0
         0    0.4009    0.1443         0         0    0.2887    0.5345         0
         0    1.0000    0.5345    0.3536    0.3536    0.6325    0.8660         0
         0         0         0         0         0         0         0         0
```