Computer Vision I

Tim Luchterhand, Paul Nykiel (Group 17) 4. Juli 2018

1 Moravec Operator

```
I = double(imread('figures1.png'))/255.0;
_{3} M1 = \begin{bmatrix} 0 & -1 & 0 \end{bmatrix};
            0 \ 1 \ 0;
            [0 \ 0 \ 0];
6 M2 = [0 0 -1;
          0 1
                 0;
          0 0
                 0];
  M3 = [0 \ 0 \ 0;
          0 \ 1 \ -1;
          0 0 0];
  M4 = [0 \ 0 \ 0;
          0 1 0;
15
          0 \ 0 \ -1];
16
   D1 = imfilter(I, M1, 'replicate');
   D2 = imfilter(I, M2, 'replicate');
   D3 = imfilter(I, M3, 'replicate');
   D4 = imfilter(I, M4, 'replicate');
21
   D1 = D1 \cdot * D1;
   D2 = D2 \cdot * D2;
   D3 = D3 \cdot * D3;
   D4 = D4 \cdot * D4;
_{28} B = ones(5,5) * 1/25;
```

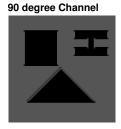
```
A1 = imfilter (D1, B, 'replicate');
  A2 = imfilter (D2, B, 'replicate');
  A3 = imfilter (D3, B, 'replicate');
  A4 = imfilter (D4, B, 'replicate');
34
  S = A1 + A2 + A3 + A4;
35
36
  A1 = A1 . / S;
  A2 = A2 ./ S;
  A3 = A3 \cdot / S;
  A4 = A4 . / S;
40
41
  threshold = 0.15;
  M1 = (A1 + A2 - abs(A1 - A2))/2; % Stolen from: https://
      stackoverflow.com/questions/26634232/element-wise-matrix-min
     -in-matlab
M2 = (A3 + A4 - abs(A3 - A4))/2;
  M = (M1 + M2 - abs(M1- M2))/2;
  underThreshold = find (M < threshold);
  overThreshold = find (M >= threshold);
  M(underThreshold) = 0;
  M(overThreshold) = 1;
51
  figure();
52
  subplot (3,2,1);
54
  imshow(A1);
  title ("0 degree Channel");
  subplot(3,2,2);
  imshow(A2);
  title ("45 degree Channel");
  subplot(3,2,3);
  imshow(A3);
  title ("90 degree Channel");
62
  subplot(3,2,4);
  imshow(A4);
  title ("135 degree Channel");
  subplot(3,2,5);
  imshow(M);
67
  title ("Detected Corners");
  subplot(3,2,6);
70 imshow(I);
  title ("Original Image");
```

```
73 print("sh04ex01.eps", "-depsc");
```

0 degree Channel



... . . .



Detected Corners



45 degree Channel



135 degree Channel



Original Image



Uncorrelated noise has a high frequency and is therefor strongly amplified by the derivation. As a result there are a lot more false positives in the image. To achieve better results one could use a low-pass-filter to surpress high frequencys (in this case the noise). This can be done by convolving the image with a gaussian kernel.

2 Structure Tensor

```
 \begin{array}{ll} {}_{1} & I = double(imread('figures1.png'))/255.0; \\ {}_{2} & Sx = \begin{bmatrix} -1 & 0 & 1; \\ & & -2 & 0 & 2; \\ & & & -1 & 0 & 1 \end{bmatrix}; \\ {}_{5} & Sy = \begin{bmatrix} -1 & -2 & -1; \\ & & & 0 & 0; \\ & & & 1 & 2 & 1 \end{bmatrix}; \\ {}_{9} & \end{array}
```

```
Ix = imfilter(I, Sx, 'replicate');
  Iy = imfilter(I, Sy, 'replicate');
12
  M = sqrt(Ix .* Ix + Iy .* Iy);
13
14
  G = fspecial('gaussian', 13, 3);
15
   S11 = imfilter(Ix \cdot * Ix, G, 'conv');
   S21 = imfilter(Iy .* Ix, G,
                                 'conv');
   S12 = imfilter(Ix \cdot *Iy, G, 'conv');
   S22 = imfilter(Iy .* Iy, G,
                                'conv');
20
  x = zeros(size(Ix,1) * size(Ix,2), 1);
^{21}
  y = zeros(size(Ix,1) * size(Ix,2), 1);
  u = zeros(size(Ix,1) * size(Ix,2), 1);
  v = zeros(size(Ix,1) * size(Ix,2), 1);
   eig1 = zeros(size(Ix,1) * size(Ix,2), 1);
   eig2 = zeros(size(Ix,1) * size(Ix,2), 1);
27
  H = zeros(size(I));
  E = zeros(size(I));
  C = zeros(size(I));
^{30}
31
   for yPos = 1: size(Ix, 1)
32
      for xPos = 1: size(Ix, 2)
33
          [V,D] = eig([S11(yPos,xPos)] S12(yPos,xPos); S21(yPos,xPos)
34
              ) S22(yPos,xPos);
          x((yPos-1)*size(Ix,2) + xPos) = xPos;
35
          y((yPos-1)*size(Ix,2) + xPos) = yPos;
36
37
          e1 = \max(D(1,1),D(2,2));
38
          e2 = \min(D(1,1),D(2,2));
40
          eig1((yPos-1)*size(Ix,2) + xPos) = e1;
41
          eig2((yPos-1)*size(Ix,2) + xPos) = e2;
42
43
          if e1 > 0.03
44
              u((yPos-1)*size(Ix,2) + xPos) = V(1,1);
45
              v((yPos-1)*size(Ix,2) + xPos) = V(2,1);
          else
47
              u((yPos-1)*size(Ix,1) + xPos) = 0;
48
              v((yPos-1)*size(Ix,1) + xPos) = 0;
49
          end
50
          if e1 <= 0.03 && e2 <= 0.03
52
```

```
H(yPos, xPos) = 1;
53
         end
54
55
         56
              E(yPos, xPos) = 1;
57
         end
58
59
         C(yPos, xPos) = 1;
61
         end
62
     end
63
  end
64
65
  figure();
67
  subplot(1,3,1);
68
  imshow(M);
69
  title ("|\nabla I|")
70
  subplot(1,3,2);
  imshow(I);
73
  hold on;
74
  quiver(x, y, u, v);
75
  title ("Quiver plot and the original image");
76
77
  subplot(1,3,3);
  imshow(cat(3,H,E,C));
79
  title ("Homogeneous in red, edges in green, corners in blue");
81
  print("sh04ex02.eps", "-depsc");
```

