

CMSC 828D: Fundamentals of Computer Vision

Homework 5¹

Instructors: Larry Davis, Ramani Duraiswami,
Daniel DeMenthon, and Yiannis Aloimonos

Solution based on homework submitted by Haiying Liu

Date: October 27, 2000

1. Generate a parameterization along this curve using chord lengths...

Solution: The Matlab script and the three column vectors are listed in appendix. The $x(s)$ and $y(s)$ are plotted in [Figure 1].

2. Using the function `polyfit` and the three vectors ...

Solution: The Matlab script and evaluations are listed in appendix. The $x(s)$ and $y(s)$ fitted by polynomial functions are shown in [Figure 2].

3. Read chapter 3.3 of *Numerical recipes*. ...

Solution: Done.

4. Fit cubic splines through the same data using the function `spline` ...

Solution: The Matlab script and evaluations are listed in appendix. The $x(s)$ and $y(s)$ fitted by polynomial functions are shown in [Figure 3].

5. You can evaluate the derivatives ...

Solution: The Matlab scripts and evaluations are listed in appendix. The figures are shown in [Figure 4]. From both figures and data, we can see the numerical result of $\frac{dy}{dx} = \frac{dy/ds}{dx/ds}$ is almost the same as the analytical expression.

6. Write a function that applies the Roberts operators to a given image ...

Solution: The Matlab scripts are listed in appendix. The figures are shown in [Figure 5], including horizontal and vertical components of the gradient. Please note that the gradient directly from Roberts operators are 135° and 225° counter-clock-wise from the x -axis and needed to be rotated back to get the correct direction. The gaussian filter is normalized so that the sum of all entries is one. The output of function `my_gradient` in `hw5_6.m` includes the gradient magnitude, direction, and a mask whose non-zero entries are those pixels with maximum gradient magnitudes along their gradient directions. The mask is obtained by the function `normMaximasSupress.m`, which uses 3×3 window to compute the local maxima. The linear interpolation is used to get more-accurate result. Please see the Matlab scripts for detail.

¹The e-version of this homework solution is posted at <http://www.cfar.umd.edu/~hyliu/CMSC828D/hw5.zip>.

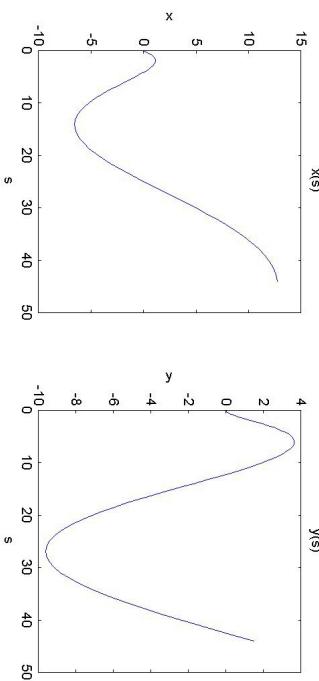


Figure 1: Original function $x(s)$ and $y(s)$.

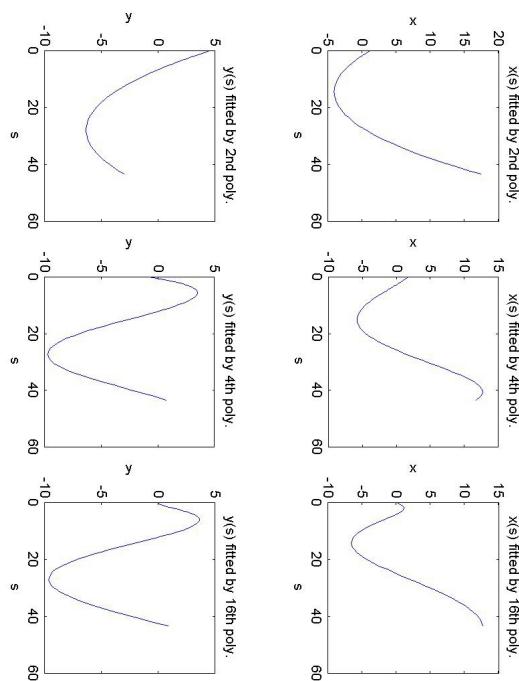
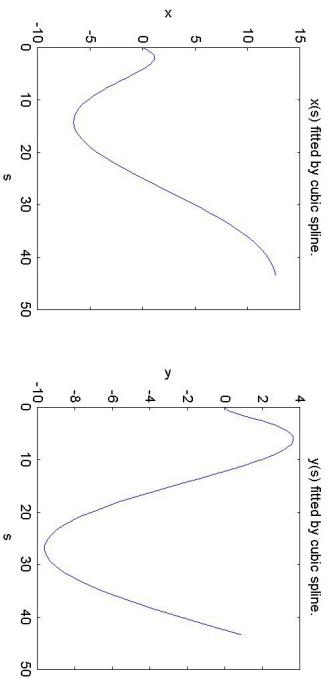
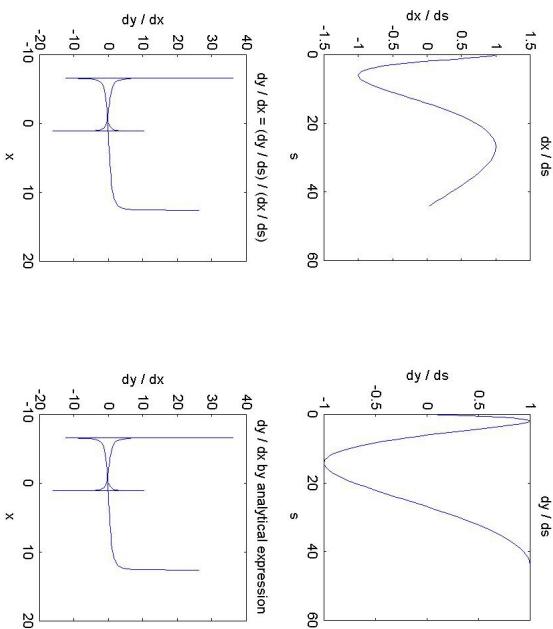
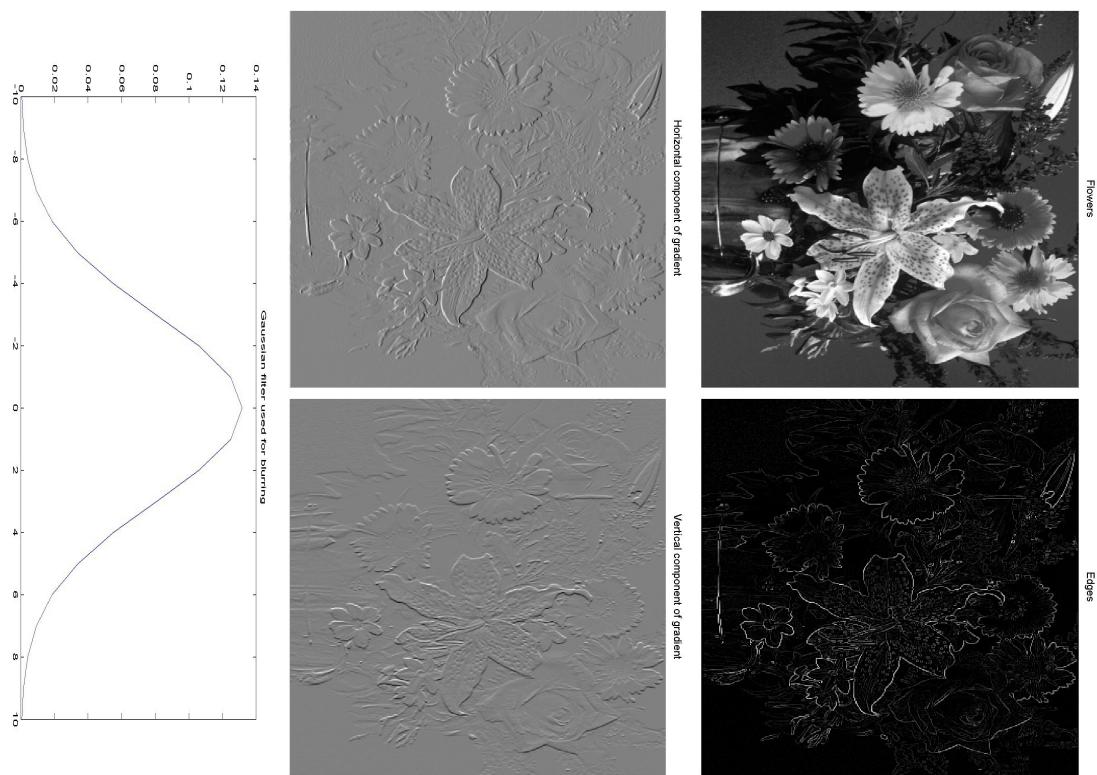
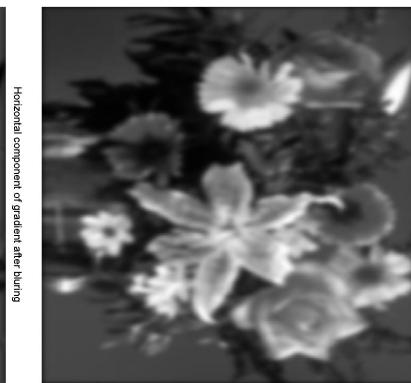
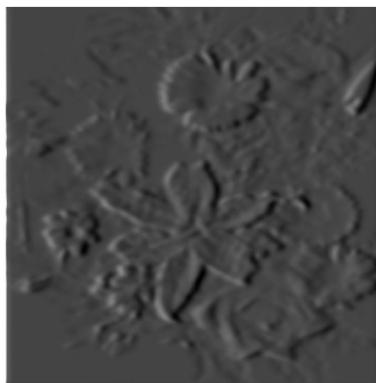


Figure 2: The $x(s)$ and $y(s)$ fitted by polynomial functions.

Figure 3: The $x(s)$ and $y(s)$ fitted by cubic spline.Figure 4: Derivatives $\frac{dy}{dx}$ by $\frac{dy}{ds} = \frac{dy/ds}{dx/ds}$ and by analytical expression.

Flowers after blurring



Edges after blurring



Vertical component of gradient after blurring

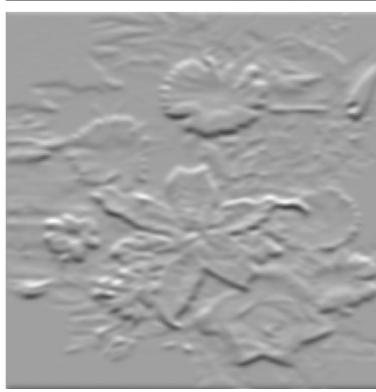


Figure 5: Images and their edges by Roberts operators.

Appendix:

- **hw5_1to5.m:**

```

function hw5_1to5
%
% Syntax: hw5_1to5
%
% Description: CMSC828D HW5_1
%
% Author: Haiying Liu
%
% Date: Oct. 2, 2000
%
%%%%%%%%%%%%%
%
dbstop if error
%
msg = narginchk(0, 0, nargin);
if (~isempty(msg))
    error(['error: ', msg]);
end

clear msg;

%%%%%%%%%%%%%
%
% Turn on the diary to save the result.
%
diary off;
%
filename = 'hw5_1to5.txt';
if (exist(filename, 'file'))
    delete(filename);
end

eval(['diary ', filename]);

%%%%%%%%%%%%%
%
% Generate the Archimedean spiral.
%
from = 0;
to = 6.4;
step = 0.1;
theta = [from:step:to];
r = 2 .* theta;

figure;
plot(x, y);
axis square;
title('Archimedean spiral');

print -djpeg archimedeanSpiral;
%
%%%%%%%%%%%%%
%
% Generate a parameterization along the curve.
%
```

```

CMSC 828D Fundamentals of Computer Vision Haiying Liu

s_mid = midpoints(s);
fit_poly_x = polyval(poly_x, s_mid);
fit_poly_y = polyval(poly_y, s_mid);
disp(' ');
disp(' ');
disp(' ');
disp(' ');
disp(['Fitted by the polynomial function in the order of ', num2str(N)]);
s_mid
fit_poly_x
fit_poly_y

subplot(2, norder, index);
plot(s_mid, fit_poly_x);
axis square;
xlabel('s');
ylabel('x');
title(['x(s) fitted by ', num2str(N), 'nd poly.']);

else
title(['x(s) fitted by ', num2str(N), 'th poly.']);
end

subplot(2, norder, norder + index);
plot(s_mid, fit_poly_y);
axis square;
xlabel('s');
ylabel('y');
title(['y(s) fitted by ', num2str(N), 'th poly.']);

if N == 2
title(['y(s) fitted by ', num2str(N), 'nd poly.']);
else
title(['y(s) fitted by ', num2str(N), 'th poly.']);
end

print -djpeg hw5_2

%%%%%
% Fit the curve by 2nd, 4th, and 16th polynomial functions.
%%%%%
disp(' ');
pp_x = spline(s, x);
pp_y = spline(s, y);
fit_spline_x = ppval(pp_x, s_mid);
fit_spline_y = ppval(pp_y, s_mid);
disp(' ');
disp(' ');
disp(' ');
disp(' ');
disp(' ');
disp(['Fitted by cubic spline']);
s_mid
fit_spline_x

```

```

figure;
% Draw the result.
figure;

subplot(1, 2, 1);
plot(s_mid, fit_spline_x);
axis square;
xlabel('s');
ylabel('x');
title(['(x)s fitted by cubic spline.']);

subplot(1, 2, 2);
plot(s_mid, fit_spline_y);
axis square;
xlabel('s');
ylabel('y');
title(['(y)s fitted by cubic spline.']);

print -djpeg hw5_4
=====

%==== Evaluate dx/ds, dy/ds, dy/dx.
=====
disp('.....');
disp('..... Problem 5 ..');
disp('.....');
disp('');

% Evaluate dy/dx by dy/ds, dx/ds.
dpp_x = diffpp(pp_x);
dpp_y = diffpp(pp_y);

fit_dpp_x = ppval(dpp_x, s);
fit_dpp_y = ppval(dpp_y, s);

dy_dx = fit_dpp_y ./ fit_dpp_x;

% Please note that x(1) = y(1) = 0, it is ignored.
nPoints = length(x);
for index = 2:nPoints
    xi = x(index);
    yi = y(index);
    norm = sqrt(xi * xi + yi * yi);
    tan_half_norm = tan(0.5 * norm);
    dy_dx_analytical(index) = -(2 * tan_half_norm * norm * norm + xi * xi + ...
        -tan_half_norm * tan_half_norm * xi * xi) / ...
        (-2 * norm + xi * yi + xi * tan_half_norm * tan_half_norm * yi);
end

% Display result.

disp('x:');
x
disp('dy/dx by (dy/ds)/(dx/ds): ');
dy_dy_dx
disp('dy/dx by analytical expression: ');
dy_dy_dx_analytical

%==== Stop recording.
diary off;
=====

%==== Stop recording.
diary off;
=====

%==== Stop recording.
diary off;
=====

function s_mid = midpoint(s)
% Syntax: s_mid = midpoint(s)
% Description: Return the midpoints of 's'.
% Author: Haiying Liu
% Date: Oct. 2, 2000
%
nPoints = length(s);
s_mid = zeros(size(s));
s_mid = s_mid(1:nPoints - 1);
for index = 1:nPoints - 1

```

```

s_mid(index) = (s(index) + s(index + 1)) ./ 2;
end

• diffpp.m:

function dpp = diffpp(pp)
% DIFFPP differentiate a pp function
%
dpp = diffpp(pp)

% returns the first derivative of the spline in PP .
%
% 2 oct 95 cb

```

```

[breaks,coefs] = unmkpp(pp);
[i,k] = size(coefs);
if k==1,
    dpp = mkpp(breaks,zeros(1,1));
else
    [k,1:-1:1];
    ans = ones(1,1);.*coefs(:,1:k-1);
    dpp = mkpp(breaks,ans,:);
end

```

```

• Result of hw5_1to5.m

s =
:: Problem 1 ::::: 5.0518
:: Problem 1 ::::: 5.4721
:: Problem 1 ::::: 5.9102
-----
Original function
s =
0
0.2000
0.4020
0.6079
0.8195
1.0386
1.2666
1.5049
1.7547
2.0169
2.2926
2.5823
2.8869
3.2068
3.5425
3.8946
4.2633
4.6489

```

```

y =
0.0200
0.0795
0.1773
0.3115
0.4794
0.6776
0.9019
1.1478
1.4100
1.6229
1.9607
2.2369
2.5053
2.7593
-----
Fitted by the
polynomial function
in the order of 2
s_mid =
0.1000
0.3010
0.5049
0.7137
0.9291
1.1526
1.3858
1.6298
1.8858
2.1548
2.4375
2.7346
3.0469
3.3747
3.7185
4.0790
4.4561
4.8504
5.2620
5.6911
6.1381
6.6031
7.0861
7.5875
8.1072
8.6454
9.2023
9.7779

```

CMSC 828D	Fundamentals of Computer Vision	Haiying Liu
-----------	---------------------------------	-------------

```

10. 3722          -1. 7582          3. 5544          -5. 7766
10. 9854          -1. 9566          3. 3733          -5. 7421
11. 6176          -2. 1543          3. 1850          -5. 6466
12. 2687          -2. 3500          2. 9892          -5. 4851
12. 9389          -2. 5528          2. 7855          -5. 2527
13. 6282          -2. 7312          2. 5740          -4. 9452
14. 3367          -2. 9142          2. 3543          -4. 5585
15. 0643          -3. 0901          2. 1266          -4. 0896
15. 8113          -3. 2578          1. 8910          -3. 5361
16. 5775          -3. 4155          1. 6475          -2. 8965
17. 3630          -3. 5618          1. 3969          -2. 1709
18. 1678          -3. 6550          1. 1390          -1. 3602
18. 9921          -3. 8134          0. 8745          -0. 4673
19. 8357          -3. 9152          0. 6037          0. 5031
20. 6988          -3. 9886          0. 3273          1. 5442
21. 5813          -4. 0617          0. 0455          2. 6467
22. 4834          -4. 1025          -0. 2405          3. 7988
23. 4049          -4. 1189          -0. 5306          4. 9855
24. 3460          -4. 1090          -0. 8239          6. 1883
25. 3066          -4. 0705          -1. 1195          7. 3850
26. 2868          -4. 0012          -1. 4169          8. 5494
27. 2865          -3. 8982          -1. 7151          9. 6502
28. 3059          -3. 7610          -2. 0132          10. 6513
29. 3448          -3. 5854          -2. 3104          11. 5112
30. 4034          -3. 3694          -2. 6057          12. 1818
31. 4816          -3. 1107          -2. 8981          12. 6086
32. 5795          -2. 8065          -3. 1867          12. 7299
33. 6970          -2. 4542          -3. 4703          12. 4759
34. 8343          -2. 0511          -3. 7478          11. 7683
35. 9912          -1. 5944          -4. 0182          5. 2620
37. 1678          -1. 0813          -4. 2803          5. 6911
38. 3641          -0. 5088          -4. 5329          0. 5835
39. 5801          -0. 1260          -4. 7745          0. 4014
40. 8158          -0. 8261          -5. 0044          0. 2076
42. 0713          -1. 5946          -5. 2208          0. 0019
43. 3465          -2. 4348          -5. 4225          0. 0119
43. 3497          -3. 3497          -5. 6082          -0. 7710
44. 3427          -4. 3427          -5. 9257          -0. 4304
45. 4171          -5. 4171          -6. 0546          -0. 4304
46. 5164          -6. 5164          -6. 1615          -0. 1018
47. 8239          -7. 8239          -6. 2451          -0. 2173
48. 1633          -9. 1633          -6. 3037          -0. 5285
49. 5990          -10. 5990          -6. 3355          0. 8326
50. 9768          -12. 1318          -6. 3391          -1. 2093
51. 7683          -13. 7683          -6. 3127          -1. 4359
52. 5113          -15. 5113          -6. 2545          -1. 4192
53. 2572          -17. 3647          -6. 1629          -1. 7000
54. 0950          -17. 3647          -6. 0361          -2. 3686
55. 2434          -18. 2434          -5. 8722          -2. 6754
56. 4198          -19. 4198          -5. 6694          -2. 9852
57. 6008          -20. 6008          -5. 4257          -3. 2957
58. 7861          -21. 7861          -5. 1394          -3. 6041
59. 9752          -22. 9752          -4. 8083          -3. 8975
60. 1677          -23. 1677          -4. 6223          -3. 0763
61. 3629          -24. 3629          -4. 4306          -3. 2275
62. 5813          -25. 5813          -4. 2223          -3. 3477
63. 8290          -26. 8290          -4. 0041          -3. 4337
64. 7291          -27. 7291          -3. 5269          -3. 4909

```

Fitted polynomial function in the order of 4

```

s_mid = 0.1000
0.3010
0.5049
0.7137
0.9291
1.1526
1.3858
1.6298
1.8858
2.1548
2.4375
2.7345
3.0469
3.3747
3.6717
3.9747
4.2717
4.5717
4.8717
5.1717
5.4717
5.7717
6.0717
6.3717
6.6717
6.9717
7.2717
7.5717
7.8717
8.1717
8.4717
8.7717
9.0717
9.3717
9.6717
9.9717
10.2717
10.5717
10.8717
11.1717
11.4717
11.7717
12.0717
12.3717
12.6717
12.9717
13.2717
13.5717
13.8717
14.1717
14.4717
14.7717
15.0717
15.3717
15.6717
15.9717
16.2717
16.5717
16.8717
17.1717
17.4717
17.7717
18.0717
18.3717
18.6717
18.9717
19.2717
19.5717
19.8717
20.1717
20.4717
20.7717
21.0717
21.3717
21.6717
21.9717
22.2717
22.5717
22.8717
23.1717
23.4717
23.7717
24.0717
24.3717
24.6717
24.9717
25.2717
25.5717
25.8717
26.1717
26.4717
26.7717
27.0717
27.3717
27.6717
27.9717
28.2717
28.5717
28.8717
29.1717
29.4717
29.7717
30.0717
31.4916
32.5795
33.6570
34.8243
35.9912
36.1360
37.1678
38.3441
39.5801
40.8158
42.0713
43.3465
43.9855
46.1883
47.3850
48.5494
49.6502
50.7124
51.7593
52.8162
53.8731
54.9300
55.9869
56.1038
57.1507
58.1976
59.2445
59.2914
59.3383
59.3852
59.4321
59.4790
59.5259
59.5728
59.6197
59.6666
59.7135
59.7604
59.8073
59.8542
59.8911
59.9380
59.9849
60.0318
60.0787
60.1256
60.1725
60.2194
60.2663
60.3132
60.3601
60.4070
60.4539
60.4908
60.5377
60.5846
60.6315
60.6784
60.7253
60.7722
60.8191
60.8660
60.9129
60.9598
61.0067
61.0536
61.1005
61.1474
61.1943
61.2412
61.2881
61.3350
61.3819
61.4288
61.4757
61.5226
61.5695
61.6164
61.6633
61.7102
61.7571
61.8040
61.8509
61.8978
61.9447
61.9916
62.0385
62.0854
62.1323
62.1792
62.2261
62.2730
62.3209
62.3678
62.4147
62.4616
62.5085
62.5554
62.6023
62.6492
62.6961
62.7430
62.7899
62.8368
62.8837
62.9306
62.9775
63.0244
63.0713
63.1182
63.1651
63.2120
63.2589
63.3058
63.3527
63.3996
63.4465
63.4934
63.5303
63.5772
63.6241
63.6710
63.7179
63.7648
63.8117
63.8586
63.9055
63.9524
64.0003
64.0472
64.0941
64.1410
64.1879
64.2348
64.2817
64.3286
64.3755
64.4224
64.4693
64.5162
64.5631
64.6099
64.6568
64.7037
64.7506
64.7975
64.8444
64.8913
64.9382
64.9851
65.0320
65.0789
65.1258
65.1727
65.2196
65.2665
65.3134
65.3603
65.4072
65.4541
65.4999
65.5468
65.5937
65.6396
65.6865
65.7334
65.7793
65.8262
65.8731
65.9199
65.9668
66.0137
66.0606
66.1075
66.1544
66.1999
66.2468
66.2937
66.3396
66.3865
66.4334
66.4799
66.5268
66.5737
66.6196
66.6665
66.7134
66.7599
66.8068
66.8537
66.8996
66.9465
66.9934
67.0399
67.0868
67.1337
67.1796
67.2265
67.2734
67.3193
67.3662
67.4131
67.4599
67.5068
67.5537
67.5996
67.6465
67.6934
67.7393
67.7862
67.8331
67.8799
67.9268
67.9737
68.0206
68.0675
68.1144
68.1613
68.2082
68.2551
68.2999
68.3468
68.3937
68.4396
68.4865
68.5334
68.5793
68.6262
68.6731
68.7199
68.7668
68.8137
68.8596
68.9065
68.9534
69.0003
69.0472
69.0941
69.1410
69.1879
69.2348
69.2817
69.3286
69.3755
69.4224
69.4693
69.5162
69.5631
69.6099
69.6568
69.7037
69.7506
69.7975
69.8444
69.8913
69.9382
69.9851
70.0320
70.0789
70.1258
70.1727
70.2196
70.2665
70.3134
70.3603
70.4072
70.4541
70.4999
70.5468
70.5937
70.6396
70.6865
70.7334
70.7793
70.8262
70.8731
70.9199
70.9668
71.0137
71.0606
71.1075
71.1544
71.1999
71.2468
71.2937
71.3396
71.3865
71.4334
71.4799
71.5268
71.5737
71.6196
71.6665
71.7134
71.7599
71.8068
71.8537
71.8996
71.9465
71.9934
72.0399
72.0868
72.1337
72.1796
72.2265
72.2734
72.3193
72.3662
72.4131
72.4599
72.5068
72.5537
72.6036
72.6505
72.6974
72.7443
72.7912
72.8381
72.8850
72.9319
72.9788
73.0257
73.0726
73.1195
73.1664
73.2133
73.2602
73.3071
73.3540
73.3999
73.4468
73.4937
73.5396
73.5865
73.6334
73.6793
73.7262
73.7731
73.8199
73.8668
73.9137
73.9596
74.0065
74.0534
74.0993
74.1462
74.1931
74.2399
74.2868
74.3337
74.3806
74.4275
74.4744
74.5213
74.5682
74.6151
74.6619
74.7088
74.7557
74.8026
74.8495
74.8964
74.9433
74.9892
75.0361
75.0829
75.1298
75.1767
75.2236
75.2695
75.3164
75.3633
75.4092
75.4561
75.5029
75.5498
75.5967
75.6436
75.6895
75.7364
75.7833
75.8292
75.8761
75.9230
75.9699
76.0168
76.0637
76.1096
76.1565
76.2034
76.2493
76.2962
76.3431
76.3890
76.4359
76.4828
76.5297
76.5766
76.6235
76.6694
76.7163
76.7632
76.8091
76.8560
76.9029
76.9498
77.0000
77.0469
77.0938
77.1407
77.1876
77.2345
77.2814
77.3283
77.3752
77.4221
77.4690
77.5159
77.5628
77.6097
77.6566
77.7035
77.7504
77.7973
77.8442
77.8911
77.9380
77.9849
78.0318
78.0787
78.1256
78.1725
78.2194
78.2663
78.3132
78.3599
78.4068
78.4537
78.4996
78.5465
78.5934
78.6393
78.6862
78.7331
78.7799
78.8268
78.8737
78.9206
78.9675
79.0144
79.0613
79.1082
79.1551
79.1999
79.2468
79.2937
79.3396
79.3865
79.4334
79.4799
79.5268
79.5737
79.6196
79.6665
79.7134
79.7599
79.8068
79.8537
79.8996
79.9465
79.9934
80.0393
80.0862
80.1331
80.1799
80.2268
80.2737
80.3206
80.3675
80.4144
80.4613
80.5082
80.5551
80.6020
80.6489
80.6958
80.7427
80.7896
80.8365
80.8834
80.9303
80.9772
81.0241
81.0710
81.1179
81.1648
81.2117
81.2586
81.3055
81.3524
81.3993
81.4462
81.4931
81.5399
81.5868
81.6337
81.6796
81.7265
81.7734
81.8193
81.8662
81.9131
81.9599
82.0068
82.0537
82.0996
82.1465
82.1934
82.2393
82.2862
82.3331
82.3799
82.4268
82.4737
82.5206
82.5675
82.6144
82.6613
82.7082
82.7551
82.8019
82.8488
82.8957
82.9426
82.9895
83.0364
83.0833
83.1299
83.1768
83.2237
83.2706
83.3175
83.3644
83.4113
83.4582
83.5051
83.5520
83.5989
83.6458
83.6927
83.7396
83.7865
83.8334
83.8793
83.9262
83.9731
84.0199
84.0668
84.1137
84.1606
84.2075
84.2544
84.2999
84.3468
84.3937
84.4406
84.4875
84.5344
84.5813
84.6282
84.6751
84.7220
84.7689
84.8158
84.8627
84.9096
84.9565
85.0034
85.0503
85.0972
85.1441
85.1910
85.2379
85.2848
85.3317
85.3786
85.4255
85.4724
85.5193
85.5662
85.6131
85.6599
85.7068
85.7537
85.8006
85.8475
85.8944
85.9413
85.9882
86.0351
86.0820
86.1289
86.1758
86.2227
86.2696
86.3165
86.3634
86.4103
86.4572
86.5041
86.5510
86.5979
86.6448
86.6917
86.7386
86.7855
86.8324
86.8793
86.9262
86.9731
87.0199
87.0668
87.1137
87.1606
87.2075
87.2544
87.2999
87.3468
87.3937
87.4406
87.4875
87.5344
87.5813
87.6282
87.6751
87.7220
87.7689
87.8158
87.8627
87.9096
87.9565
88.0034
88.0503
88.0972
88.1441
88.1910
88.2379
88.2848
88.3317
88.3786
88.4255
88.4724
88.5193
88.5662
88.6131
88.6599
88.7068
88.7537
88.8006
88.8475
88.8944
88.9413
88.9882
89.0351
89.0820
89.1289
89.1758
89.2227
89.2696
89.3165
89.3634
89.4103
89.4572
89.5041
89.5510
89.5979
89.6448
89.6917
89.7386
89.7855
89.8324
89.8793
89.9262
89.9731
90.0199
90.0668
90.1137
90.1606
90.2075
90.2544
90.2999
90.3468
90.3937
90.4406
90.4875
90.5344
90.5813
90.6282
90.6751
90.7220
90.7689
90.8158
90.8627
90.9096
90.9565
91.0034
91.0503
91.0972
91.1441
91.1910
91.2379
91.2848
91.3317
91.3786
91.4255
91.4724
91.5193
91.5662
91.6131
91.6599
91.7068
91.7537
91.8006
91.8475
91.8944
91.9413
91.9882
92.0351
92.0820
92.1289
92.1758
92.2227
92.2696
92.3165
92.3634
92.4103
92.4572
92.5041
92.5510
92.5979
92.6448
92.6917
92.7386
92.7855
92.8324
92.8793
92.9262
92.9731
93.0199
93.0668
93.1137
93.1606
93.2075
93.2544
93.2999
93.3468
93.3937
93.4406
93.4875
93.5344
93.5813
93.6282
93.6751
93.7220
93.7689
93.8158
93.8627
93.9096
93.9565
94.0034
94.0503
94.0972
94.1441
94.1910
94.2379
94.2848
94.3317
94.3786
94.4255
94.4724
94.5193
94.5662
94.6131
94.6599
94.7068
94.7537
94.8006
94.8475
94.8944
94.9413
94.9882
95.0351
95.0820
95.1289
95.1758
95.2227
95.2696
95.3165
95.3634
95.4103
95.4572
95.5041
95.5510
95.5979
95.6448
95.6917
95.7386
95.7855
95.8324
95.8793
95.9262
95.9731
96.0199
96.0668
96.1137
96.1606
96.2075
96.2544
96.2999
96.3468
96.3937
96.4406
96.4875
96.5344
96.5813
96.6282
96.6751
96.7220
96.7689
96.8158
96.8627
96.9096
96.9565
97.0034
97.0503
97.0972
97.1441
97.1910
97.2379
97.2848
97.3317
97.3786
97.4255
97.4724
97.5193
97.5662
97.6131
97.6599
97.7068
97.7537
97.8006
97.8475
97.8944
97.9413
97.9882
98.0351
98.0820
98.1289
98.1758
98.2227
98.2696
98.3165
98.3634
98.4103
98.4572
98.5041
98.5510
98.5979
98.6448
98.6917
98.7386
98.7855
98.8324
98.8793
98.9262
98.9731
99.0199
99.0668
99.1137
99.1606
99.2075
99.2544
99.2999
99.3468
99.3937
99.4406
99.4875
99.5344
99.5813
99.6282
99.6751
99.7220
99.7689
99.8158
99.8627
99.9096
99.9565
100.0034
100.0503
100.0972
100.1441
100.1910
100.2379
100.2848
100.3317
100.3786
100.4255
100.4724
100.5193
100.5662
100.6131
100.6599
100.7068
100.7537
100.8006
100.8475
100.8944
100.9413
100.9882
101.0351
101.0820
101.1289
101.1758
101.2227
101.2696
101.3165
101.3634
101.4103
101.4572
101.5041
101.5510
101.5979
101.6448
101.6917
101.7386
101.7855
101.8324
101.8793
101.9262
101.9731
102.0199
102.0668
102.1137
102.1606
102.2075
102.2544
102.2999
102.3468
102.3937
102.4406
102.4875
102.5344
102.5813
102.6282
102.6751
102.7220
102.7689
102.8158
102.8627
102.9096
102.9565
103.0034
103.0503
103.0972
103.1441
103.1910
103.2379
103.2848
103.3317
103.3786
103.4255
103.4724
103.5193
103.5662
103.6131
103.6599
103.7068
103.7537
103.8006
103.8475
103.8944
103.9413
103.9882
104.0351
104.0820
104.1289
104.1758
104.2227
104.2696
104.3165
104.3634
104.4103
104.4572
104.5041
104.5510
104.5979
104.6448
104.6917
104.7386
104.7855
104.8324
104.8793
104.9262
104.9731
105.0199
105.0668
105.1137
105.1606
105.2075
105.2544
105.2999
105.3468
105.3937
105.4406
105.4875
105.5344
105.5813
105.6282
105.
```

CMSC 828D Fundamentals of Computer Vision **Haiying Liu**

```

2. 8462
2. 5685
2. 2400
1. 8609
1. 4324
0. 9562
-0. 1281
-0. 7292
polyfit.m at line 52
In
= 1.578650e-027.

0. 5543
-5. 4621
-5. 7950
-6. 0769
-6. 3014
-6. 4623
-6. 5543
-6. 5729
-6. 5142
-6. 3751
-6. 1532
-5. 8471
-5. 4562
-4. 9815
-4. 4248
-3. 7898
-3. 0813
-2. 3050
-2. 3050
-1. 4678
-1. 4678
-0. 5771
0. 3590
-3. 5564
0. 0551
-0. 7060
-1. 3894
-1. 3894
-2. 0377
-2. 8228
-4. 4375
-3. 7346
-3. 0469
-0. 7060
-1. 3747
-3. 7346
-4. 8504
-4. 8504
-5. 0114
-5. 7145
-5. 6911
-6. 3893
-6. 6031
-7. 0861
-7. 5875
-8. 1072
-8. 6454
-9. 2826
-9. 7779
-10. 3722
-10. 9854
-11. 6176
-12. 2687
-12. 9389
-13. 6282
-14. 3367
-15. 0643
-15. 8113
-15. 5775
-16. 3630
-17. 3630
18. 1678
18. 9921
19. 8357
20. 6988
21. 5813
22. 4834
23. 4049
24. 3460
25. 3066
26. 2868
27. 2865
28. 3059
29. 3448
30. 4034
31. 4816
32. 5795
33. 6970
34. 8343
35. 9912
37. 1678
38. 3641
39. 5801
40. 8158
42. 0713
43. 3465
45. 3806
5. 3806
6. 3727
6. 3323
7. 3323
8. 2484
9. 1101
9. 9057
10. 6223
11. 2472
11. 7717
11. 9529
12. 1915
12. 4963
12. 6645
1. 0325
1. 0952
5. 2620
5. 6911
5. 1206
6. 1381
6. 6031
6. 0459
7. 0861
0. 9411
0. 7898
0. 5920
0. 3489
0. 0624
-0. 2644
-0. 6280
-1. 0240
-1. 4476
-1. 8934
-2. 3555
-2. 8277
-3. 0984
-3. 4933
-3. 8989
-4. 2406
-2. 7229
-1. 6309
-0. 4135
0. 8536
1. 0253
1. 2786
1. 5457
1. 8212
2. 0989

Warning: Matrix is
close to singular or
badly scaled.
Results may
be inaccurate. RCOND
= 1.578650e-027.

Fitted by cubic
s_mid =
-----
```

```

CMSC 828D          Fundamentals of Computer Vision          Haiying Liu
-----          -----          -----
CMSC 828D          Fundamentals of Computer Vision          Haiying Liu
-----          -----          -----

```

CMSC 828D

Fundamentals of Computer Vision

Haiying Liu

CMSC 828D

Fundamentals of Computer Vision

Haiying Liu

```

2.0974
0.9979
0.5522
0.5955
0.8611
0.0861
0.0588
1.0588
1.3048
1.6271
2.0790
2.7767
4.0324
7.0629
26.3075

dy_dx_by_analytical
expression:
ans =
0.2024
0.4197
0.6717
0.9903
1.4395
2.1783
3.7580
10.3786
-16.1035
-0.7742
0.3304
0.6313
0.7794
0.9784
-1.3311
-1.0114
0
-4.5882
-2.6393
-1.8078
-4.4558
-4.8820
-5.2764
-1.3311
-0.0114
0
-5.6316
-0.5551
-0.4258
-0.4258
-6.1946
-6.3891
-0.2053
-6.5174
-6.5174
-0.1565
-6.5742
-6.5742
-0.0345
-6.5552
-6.4567
-6.2759
-6.2759
-6.0114
-6.0114
-4.6440
-5.6623
-5.6623
-0.6313
-5.2291
-5.2291
-0.7794
-4.7135
-4.7135
-4.1182
-4.1182
-6.0114
-6.0114
-4.5880
-5.6623
-5.6623
-0.6313
-4.258
-4.258
-1.8078
-1.8078
-1.0114
-1.0114
-0.7742
-0.7742
-0.5851
-0.5851
-0.4258
-0.4258
-0.2853
-0.2853
-0.1565
-0.1565
-0.0345
-0.0345
-0.0850
-0.0850
-0.2054
-0.2054
-0.3304
-0.3304
-0.4640
-0.4640
-0.6113
-0.6113
-0.7794
-0.7794
-1.4204
-1.4204
-0.9784
-0.9784
1.2250
1.2250
1.5477
1.5477
2.0015
2.0015
2.7045
2.7045
2.0015
2.0015
-0.2054
-0.2054
-0.3304
-0.3304
-0.4640
-0.4640
0.0015
0.0015
0.6113
0.6113
0.7794
0.7794
1.4204
1.4204
0.9784
0.9784
1.2250
1.2250
1.5477
1.5477
2.0015
2.0015
2.7045
2.7045
3.3048
3.3048
4.0085
4.0085
4.7089
4.7089
5.4085
5.4085
6.1085
6.1085
6.8085
6.8085
7.5085
7.5085
8.2085
8.2085
8.9085
8.9085
9.6085
9.6085
10.3085
10.3085
11.0085
11.0085
11.7085
11.7085
12.4085
12.4085
13.1085
13.1085
13.8085
13.8085
14.5085
14.5085
15.2085
15.2085
15.9085
15.9085
16.6085
16.6085
17.3085
17.3085
18.0085
18.0085
18.7085
18.7085
19.4085
19.4085
20.1085
20.1085
20.8085
20.8085
21.5085
21.5085
22.2085
22.2085
22.9085
22.9085
23.6085
23.6085
24.3085
24.3085
25.0085
25.0085
25.7085
25.7085
26.4085
26.4085
27.1085
27.1085
27.8085
27.8085
28.5085
28.5085
29.2085
29.2085
29.9085
29.9085
30.6085
30.6085
31.3085
31.3085
32.0085
32.0085
32.7085
32.7085
33.4085
33.4085
34.1085
34.1085
34.8085
34.8085
35.5085
35.5085
36.2085
36.2085
36.9085
36.9085
37.6085
37.6085
38.3085
38.3085
39.0085
39.0085
39.7085
39.7085
40.4085
40.4085
41.1085
41.1085
41.8085
41.8085
42.5085
42.5085
43.2085
43.2085
43.9085
43.9085
44.6085
44.6085
45.3085
45.3085
46.0085
46.0085
46.7085
46.7085
47.4085
47.4085
48.1085
48.1085
48.8085
48.8085
49.5085
49.5085
50.2085
50.2085
50.9085
50.9085
51.6085
51.6085
52.3085
52.3085
53.0085
53.0085
53.7085
53.7085
54.4085
54.4085
55.1085
55.1085
55.8085
55.8085
56.5085
56.5085
57.2085
57.2085
57.9085
57.9085
58.6085
58.6085
59.3085
59.3085
60.0085
60.0085
60.7085
60.7085
61.4085
61.4085
62.1085
62.1085
62.8085
62.8085
63.5085
63.5085
64.2085
64.2085
64.9085
64.9085
65.6085
65.6085
66.3085
66.3085
67.0085
67.0085
67.7085
67.7085
68.4085
68.4085
69.1085
69.1085
69.8085
69.8085
70.5085
70.5085
71.2085
71.2085
71.9085
71.9085
72.6085
72.6085
73.3085
73.3085
74.0085
74.0085
74.7085
74.7085
75.4085
75.4085
76.1085
76.1085
76.8085
76.8085
77.5085
77.5085
78.2085
78.2085
78.9085
78.9085
79.6085
79.6085
80.3085
80.3085
81.0085
81.0085
81.7085
81.7085
82.4085
82.4085
83.1085
83.1085
83.8085
83.8085
84.5085
84.5085
85.2085
85.2085
85.9085
85.9085
86.6085
86.6085
87.3085
87.3085
88.0085
88.0085
88.7085
88.7085
89.4085
89.4085
90.1085
90.1085
90.8085
90.8085
91.5085
91.5085
92.2085
92.2085
92.9085
92.9085
93.6085
93.6085
94.3085
94.3085
95.0085
95.0085
95.7085
95.7085
96.4085
96.4085
97.1085
97.1085
97.8085
97.8085
98.5085
98.5085
99.2085
99.2085
100.0085
100.0085
100.7085
100.7085
101.4085
101.4085
102.1085
102.1085
102.8085
102.8085
103.5085
103.5085
104.2085
104.2085
104.9085
104.9085
105.6085
105.6085
106.3085
106.3085
107.0085
107.0085
107.7085
107.7085
108.4085
108.4085
109.1085
109.1085
109.8085
109.8085
110.5085
110.5085
111.2085
111.2085
111.9085
111.9085
112.6085
112.6085
113.3085
113.3085
114.0085
114.0085
114.7085
114.7085
115.4085
115.4085
116.1085
116.1085
116.8085
116.8085
117.5085
117.5085
118.2085
118.2085
118.9085
118.9085
119.6085
119.6085
120.3085
120.3085
121.0085
121.0085
121.7085
121.7085
122.4085
122.4085
123.1085
123.1085
123.8085
123.8085
124.5085
124.5085
125.2085
125.2085
125.9085
125.9085
126.6085
126.6085
127.3085
127.3085
128.0085
128.0085
128.7085
128.7085
129.4085
129.4085
130.1085
130.1085
130.8085
130.8085
131.5085
131.5085
132.2085
132.2085
132.9085
132.9085
133.6085
133.6085
134.3085
134.3085
135.0085
135.0085
135.7085
135.7085
136.4085
136.4085
137.1085
137.1085
137.8085
137.8085
138.5085
138.5085
139.2085
139.2085
140.0085
140.0085
140.7085
140.7085
141.4085
141.4085
142.1085
142.1085
142.8085
142.8085
143.5085
143.5085
144.2085
144.2085
144.9085
144.9085
145.6085
145.6085
146.3085
146.3085
147.0085
147.0085
147.7085
147.7085
148.4085
148.4085
149.1085
149.1085
149.8085
149.8085
150.5085
150.5085
151.2085
151.2085
151.9085
151.9085
152.6085
152.6085
153.3085
153.3085
154.0085
154.0085
154.7085
154.7085
155.4085
155.4085
156.1085
156.1085
156.8085
156.8085
157.5085
157.5085
158.2085
158.2085
158.9085
158.9085
159.6085
159.6085
160.3085
160.3085
161.0085
161.0085
161.7085
161.7085
162.4085
162.4085
163.1085
163.1085
163.8085
163.8085
164.5085
164.5085
165.2085
165.2085
165.9085
165.9085
166.6085
166.6085
167.3085
167.3085
168.0085
168.0085
168.7085
168.7085
169.4085
169.4085
170.1085
170.1085
170.8085
170.8085
171.5085
171.5085
172.2085
172.2085
172.9085
172.9085
173.6085
173.6085
174.3085
174.3085
175.0085
175.0085
175.7085
175.7085
176.4085
176.4085
177.1085
177.1085
177.8085
177.8085
178.5085
178.5085
179.2085
179.2085
179.9085
179.9085
180.6085
180.6085
181.3085
181.3085
182.0085
182.0085
182.7085
182.7085
183.4085
183.4085
184.1085
184.1085
184.8085
184.8085
185.5085
185.5085
186.2085
186.2085
186.9085
186.9085
187.6085
187.6085
188.3085
188.3085
189.0085
189.0085
189.7085
189.7085
190.4085
190.4085
191.1085
191.1085
191.8085
191.8085
192.5085
192.5085
193.2085
193.2085
193.9085
193.9085
194.6085
194.6085
195.3085
195.3085
196.0085
196.0085
196.7085
196.7085
197.4085
197.4085
198.1085
198.1085
198.8085
198.8085
199.5085
199.5085
200.2085
200.2085
200.9085
200.9085
201.6085
201.6085
202.3085
202.3085
203.0085
203.0085
203.7085
203.7085
204.4085
204.4085
205.1085
205.1085
205.8085
205.8085
206.5085
206.5085
207.2085
207.2085
207.9085
207.9085
208.6085
208.6085
209.3085
209.3085
210.0085
210.0085
210.7085
210.7085
211.4085
211.4085
212.1085
212.1085
212.8085
212.8085
213.5085
213.5085
214.2085
214.2085
214.9085
214.9085
215.6085
215.6085
216.3085
216.3085
217.0085
217.0085
217.7085
217.7085
218.4085
218.4085
219.1085
219.1085
219.8085
219.8085
220.5085
220.5085
221.2085
221.2085
221.9085
221.9085
222.6085
222.6085
223.3085
223.3085
224.0085
224.0085
224.7085
224.7085
225.4085
225.4085
226.1085
226.1085
226.8085
226.8085
227.5085
227.5085
228.2085
228.2085
228.9085
228.9085
229.6085
229.6085
230.3085
230.3085
231.0085
231.0085
231.7085
231.7085
232.4085
232.4085
233.1085
233.1085
233.8085
233.8085
234.5085
234.5085
235.2085
235.2085
235.9085
235.9085
236.6085
236.6085
237.3085
237.3085
238.0085
238.0085
238.7085
238.7085
239.4085
239.4085
240.1085
240.1085
240.8085
240.8085
241.5085
241.5085
242.2085
242.2085
242.9085
242.9085
243.6085
243.6085
244.3085
244.3085
245.0085
245.0085
245.7085
245.7085
246.4085
246.4085
247.1085
247.1085
247.8085
247.8085
248.5085
248.5085
249.2085
249.2085
249.9085
249.9085
250.6085
250.6085
251.3085
251.3085
252.0085
252.0085
252.7085
252.7085
253.4085
253.4085
254.1085
254.1085
254.8085
254.8085
255.5085
255.5085
256.2085
256.2085
256.9085
256.9085
257.6085
257.6085
258.3085
258.3085
259.0085
259.0085
259.7085
259.7085
260.4085
260.4085
261.1085
261.1085
261.8085
261.8085
262.5085
262.5085
263.2085
263.2085
263.9085
263.9085
264.6085
264.6085
265.3085
265.3085
266.0085
266.0085
266.7085
266.7085
267.4085
267.4085
268.1085
268.1085
268.8085
268.8085
269.5085
269.5085
270.2085
270.2085
270.9085
270.9085
271.6085
271.6085
272.3085
272.3085
273.0085
273.0085
273.7085
273.7085
274.4085
274.4085
275.1085
275.1085
275.8085
275.8085
276.5085
276.5085
277.2085
277.2085
277.9085
277.9085
278.6085
278.6085
279.3085
279.3085
279.9999
279.9999
280.6085
280.6085
281.2085
281.2085
281.8085
281.8085
282.4085
282.4085
283.0085
283.0085
283.6085
283.6085
284.2085
284.2085
284.8085
284.8085
285.4085
285.4085
286.0085
286.0085
286.6085
286.6085
287.2085
287.2085
287.8085
287.8085
288.4085
288.4085
289.0085
289.0085
289.6085
289.6085
290.2085
290.2085
290.8085
290.8085
291.4085
291.4085
292.0085
292.0085
292.6085
292.6085
293.2085
293.2085
293.8085
293.8085
294.4085
294.4085
295.0085
295.0085
295.6085
295.6085
296.2085
296.2085
296.8085
296.8085
297.4085
297.4085
298.0085
298.0085
298.6085
298.6085
299.2085
299.2085
299.8085
299.8085
300.4085
300.4085
301.0085
301.0085
301.6085
301.6085
302.2085
302.2085
302.8085
302.8085
303.4085
303.4085
304.0085
304.0085
304.6085
304.6085
305.2085
305.2085
305.8085
305.8085
306.4085
306.4085
307.0085
307.0085
307.6085
307.6085
308.2085
308.2085
308.8085
308.8085
309.4085
309.4085
310.0085
310.0085
310.6085
310.6085
311.2085
311.2085
311.8085
311.8085
312.4085
312.4085
313.0085
313.0085
313.6085
313.6085
314.2085
314.2085
314.8085
314.8085
315.4085
315.4085
316.0085
316.0085
316.6085
316.6085
317.2085
317.2085
317.8085
317.8085
318.4085
318.4085
319.0085
319.0085
319.6085
319.6085
320.2085
320.2085
320.8085
320.8085
321.4085
321.4085
322.0085
322.0085
322.6085
322.6085
323.2085
323.2085
323.8085
323.8085
324.4085
324.4085
325.0085
325.0085
325.6085
325.6085
326.2085
326.2085
326.8085
326.8085
327.4085
327.4085
328.0085
328.0085
328.6085
328.6085
329.2085
329.2085
329.8085
329.8085
330.4085
330.4085
331.0085
331.0085
331.6085
331.6085
332.2085
332.2085
332.8085
332.8085
333.4085
333.4085
334.0085
334.0085
334.6085
334.6085
335.2085
335.2085
335.8085
335.8085
336.4085
336.4085
337.0085
337.0085
337.6085
337.6085
338.2085
338.2085
338.8085
338.8085
339.4085
339.4085
340.0085
340.0085
340.6085
340.6085
341.2085
341.2085
341.8085
341.8085
342.4085
342.4085
343.0085
343.0085
343.6085
343.6085
344.2085
344.2085
344.8085
344.8085
345.4085
345.4085
346.0085
346.0085
346.6085
346.6085
347.2085
347.2085
347.8085
347.8085
348.4085
348.4085
349.0085
349.0085
349.6085
349.6085
350.2085
350.2085
350.8085
350.8085
351.4085
351.4085
352.0085
352.0085
352.6085
352.6085
353.2085
353.2085
353.8085
353.8085
354.4085
354.4085
355.0085
355.0085
355.6085
355.6085
356.2085
356.2085
356.8085
356.8085
357.4085
357.4085
358.0085
358.0085
358.6085
358.6085
359.2085
359.2085
359.8085
359.8085
360.4085
360.4085
361.0085
361.0085
361.6085
361.6085
362.2085
362.2085
362.8085
362.8085
363.4085
363.4085
364.0085
364.0085
364.6085
364.6085
365.2085
365.2085
365.8085
365.8085
366.4085
366.4085
367.0085
367.0085
367.6085
367.6085
368.2085
368.2085
368.8085
368.8085
369.4085
369.4085
369.8085
369.8085
370.4085
370.4085
371.0085
371.0085
371.6085
371.6085
372.2085
372.2085
372.8085
372.8085
373.4085
373.4085
374.0085
374.0085
374.6085
374.6085
375.2085
375.2085
375.8085
375.8085
376.4085
376.4085
377.0085
377.0085
377.6085
377.6085
378.2085
378.2085
378.8085
378.8085
379.4085
379.4085
379.8085
379.8085
380.4085
380.4085
381.0085
381.0085
381.6085
381.6085
382.2085
382
```

CMSC 828D Fundamentals of Computer Vision **Haiying Liu**

CMSC 828D Fundamentals of Computer Vision **Haibing Liu**

```

function hw5_6
% Syntax: hw5_6
% Description: CMSC828D HW5_1
% Author: Haiying Liu
% Date: Oct. 3, 2000
%
msg = nargchk(0, 0, nargin);
if (~isempty(msg))
    error(strcat('ERROR:', msg));
end

clear msg;

%=====
% Turn on the diary to save the result.
diary off;

filename = 'hw5_6.txt';
if (exist(filename, 'file'))
    delete(filename);
end

eval(['diary ', filename]);
%
% Read image data.
image = imread('flowers.tif');

% Convert the color image into gray scale image.
image = rgb2gray(image);

% Convert the data type from uint8 to double.
image = double(image);

figure;
plot(t, gau);

```

```
print -djpeg hw5_6_gaus;
```

```
% Blur the image
```

```
% Apply gaussian filter along x and y directions.
```

```
blurredImage = conv2(gau', gau, image, 'same');
```

```
figure;
```

```
axis square;
```

```
title('Flowers after bluring');
```

```
print -djpeg hw5_6_flower_blur;
```

```
% Apply roberts operators.
```

```
gradient.blur = my_gradient(blurredImage);
```

```
% Show result.
```

```
figure;
```

```
imshow(gradient.blur.mag .* gradient.blur.mask, []);
```

```
axis square;
```

```
title('Edges after bluring');
```

```
print -djpeg hw5_6_edge_blur;
```

```
figure;
```

```
imshow(gradient.blur.mag .* cos(gradient.blur.dir), []);
```

```
axis square;
```

```
title('Vertical component of gradient after bluring');
```

```
print -djpeg hw5_6_edge_blur_v;
```

```
figure;
```

```
imshow(gradient.blur.mag .* sin(gradient.blur.dir), []);
```

```
axis square;
```

```
title('Horizontal component of gradient after bluring');
```

```
print -djpeg hw5_6_edge_blur_h;
```

```
%=====
% Stop recording.
```

```
diary off;
```

```
%=====
% Syntax: gradient = my_gradient(data)
```

```
%
% data - data to be computed
```

```
%
% gradient - a structure contains results:
```

```
%
% .mag : magnitude of the gradient
%         larger than 'threshold'
```

```
%
% .dir : direction of the gradient
%         .mask : mask for local maxima
%         1 : is local maxima
%         0 : is not local maxima
```

```
%
% .mag .* .mask yields edges.
```

```
%
% Description: Compute gradient by roberts operators
```

```
%
% Author: Haiying Liu
```

```
% Date: Oct. 3, 2000
```

```
%=====
%
```

```
%=====
%
```

```
dbstop if error
```

```
msg = nargchk(1, 1, nargin);
```

```
if (~isempty(msg))
```

```
error(strcat('ERROR: ', msg));
```

```
end
```

```
clear msg;
```

```
%=====
%
```

```
% Construct roberts operator.
```

```
op_roberts = [1, 0; 0, -1] / sqrt(2);
```

```
% Get gradient.
```

```
bx = filter2(op_roberts, data, 'same');
```

```
by = filter2(rot90(op_roberts), data, 'same');
```

```
% Compute magnitude and direction.
```

```
theta = 135 * pi / 180;
```

```
dx = bx .* cos(theta) - by .* sin(theta);
```

```
dy = bx .* sin(theta) + by .* cos(theta);
```

```
grad = dx + i * dy;
```

```
gradient.mag = abs(grad);
```

```
gradient.dir = angle(grad);
```

```
% Suppress non-maxima
```

```
gradient.Mask = zeros(size(data));
```

```
nRow = size(data, 1);
```

```
nCol = size(data, 2);
```

```
for row = 2:nRow - 1
```

```
    for col = 2:nCol - 1
```

```
wnd = grad(row - 1:row + 1, col - 1:col + 1);
```

```
gradient.mask(row, col) = ...
```

```
nonMaximasuppress(real(wnd), imag(wnd));
```

```
end
```

- **nonMaximasuppress.m:**

```
function r = nonMaximasuppress(gradX, gradY)
```

```
Syntax: r = nonMaximasuppress(gradX, gradY)
```

```
Arg: gradX(Y) - local 3x3 gradient matrix.
```

```
r - boolean result.
```

```
1: is local maxima
```

```
0: is NOT local maxima
```

```
% Description: Determine if the center of "local" is the max gradient
```

```

% Reference : Numerical Recipes
% Author: Haiying
% Date : Oct 3, 2000

% Check arguments.
msg = narginchk(2, 2, nargin);
if (~isempty(msg))
    error(strcat('ERROR: ', msg));
end

clear msg;

%===== For each pixel (except those on the image boundary), check if
% the gradient magnitude is a local (3x3 neighbourhood) maxima in
% the gradient orientation:
r = 0;

y = 2;
x = 2;

dx = gradx(y, x);
dy = grady(y, x);

mag = sqrt(gradx .* gradx + grady .* grady);

if (dx ~= 0 | dy ~= 0)
    if (abs(dy) > abs(dx))
        ux = abs(dx) / abs(dy);
        uy = 1;
    else
        gb = mag(y - 1, x);
        gd = mag(y + 1, x);
        ga = mag(y - 1, x - 1);
        gc = mag(y + 1, x + 1);
        gc = mag(y + 1, x - 1);
    end
    else
        ux = abs(dy) / abs(dx);
        uy = 1;
    end
    gb = mag(y, x + 1);
    gd = mag(y, x - 1);
    if (sign(dx) == sign(dy))
        ga = mag(y + 1, x + 1);
        gc = mag(y - 1, x - 1);
    else
        ga = mag(y + 1, x + 1);
        gc = mag(y - 1, x - 1);
    end
end

g1 = (ux * ga) + (uy - ux) * gb;
g2 = (ux * gc) + (uy - ux) * gd;
g = sqrt(dx * dx + dy * dy);
if (g >= g1 & g >= g2)
    r = 1;
else
    r = 0;
end
end

```