

```

1  % FiltScript2.m  v2
2  % Call TestFilt2 multiple times
3  % Iain Richardson 1 August 24
4
5  clear; % Clear previous variable contents
6  randomseedOn = 1; % 1 = repeatable sequence of random numbers, 0 = random each time this
   is run.
7  numruns = 1000;
8  UseAltThreshold=0; % Set to 0: 841/659 method uses thresholds Thr1 and Thr2 from '841
   16:59-60. Set to 1: 841/659 method uses alternative threshold.
9  Uselcodec4Diff = 0; % v2 Set to 0: use Diff equation from Q15J28 at 4.5.6.1 and 4.5.6.2,
   based on original d or e. Set to 1: use Diff calculation from lcodec4.0 image.c at lines
   491 and 499. v2
10
11 % Open files for writing
12 if ((UseAltThreshold==0) && (Uselcodec4Diff==0)) % Use 841/659 Thr1 and Thr2, Q15J28
   Diff equation
13     FIDstats = fopen('FiltStatsThr1Thr2_2a.txt', 'w+');
14     FIDblocks = fopen('FiltBlocksThr1Thr2_2a.txt', 'w+');
15     fprintf(FIDstats, ''841/'659 method: using Thr1 and Thr2\n');
16     fprintf(FIDblocks, ''841/'659 method: using Thr1 and Thr2\n');
17     fprintf(FIDstats, 'Pixels c, f: use Diff equation from Q15J28 at 4.5.6.1 and
   4.5.6.2\n');
18     fprintf(FIDblocks, 'Pixels c, f: use Diff equation from Q15J28 at 4.5.6.1 and
   4.5.6.2\n');
19 elseif ((UseAltThreshold==1) && (Uselcodec4Diff==0)) % Use alternative threshold, Q15J28
   Diff equation
20     FIDstats = fopen('FiltStatsAltThr_2a.txt', 'w+');
21     FIDblocks = fopen('FiltBlocksAltThr_2a.txt', 'w+');
22     fprintf(FIDstats, ''841/'659 method: using S01 and S02\n');
23     fprintf(FIDblocks, ''841/'659 method: using S01 and S02\n');
24     fprintf(FIDstats, 'Pixels c, f: use Diff equation from Q15J28 at 4.5.6.1 and
   4.5.6.2\n');
25     fprintf(FIDblocks, 'Pixels c, f: use Diff equation from Q15J28 at 4.5.6.1 and
   4.5.6.2\n');
26 elseif ((UseAltThreshold==0) && (Uselcodec4Diff==1)) % Use 841/659 Thr1 and Thr2,
   lcodec4.0 Diff equation v2
27     FIDstats = fopen('FiltStatsThr1Thr2_2b.txt', 'w+');
28     FIDblocks = fopen('FiltBlocksThr1Thr2_2b.txt', 'w+');
29     fprintf(FIDstats, ''841/'659 method: using Thr1 and Thr2\n');
30     fprintf(FIDblocks, ''841/'659 method: using Thr1 and Thr2\n');
31     fprintf(FIDstats, 'Pixels c, f: use Diff equation from lcodec4.0 image.c at lines
   491 and 499\n');
32     fprintf(FIDblocks, 'Pixels c, f: use Diff equation from lcodec4.0 image.c at lines
   491 and 499\n');
33 elseif ((UseAltThreshold==1) && (Uselcodec4Diff==1)) % Use alternative threshold,
   lcodec4.0 Diff equation v2
34     FIDstats = fopen('FiltStatsAltThr_2b.txt', 'w+');
35     FIDblocks = fopen('FiltBlocksAltThr_2b.txt', 'w+');
36     fprintf(FIDstats, ''841/'659 method: using S01 and S02\n');
37     fprintf(FIDblocks, ''841/'659 method: using S01 and S02\n');
38     fprintf(FIDstats, 'Pixels c, f: use Diff equation from lcodec4.0 image.c at lines
   491 and 499\n');
39     fprintf(FIDblocks, 'Pixels c, f: use Diff equation from lcodec4.0 image.c at lines
   491 and 499\n');
40 else
41     fprintf(FIDstats, 'Error: unknown setting of UseAltThreshold and UseQ15J28Diff\n\n');
42     return;
43 end
44 fprintf(FIDstats, 'Generated using FiltScript2.m and TestFilt2.m, %s\n', datetime("today"
   ));
45 fprintf(FIDblocks, 'Generated using FiltScript2.m and TestFilt2.m, %s\n', datetime(
   "today"));
46
47 DiffDE = zeros(numruns, 8);

```

```

48 DiffC = zeros(numruns, 8);
49 DiffF = zeros(numruns, 8);
50 ModPixArray1 = zeros(numruns, 8);
51 FirstOff = zeros(numruns, 8);
52 SecondOff = zeros(numruns, 8);
53 ThirdOff = zeros(numruns, 8);
54 FirstOffClip = zeros(numruns, 8);
55 SecondOffClip = zeros(numruns, 8);
56 ThirdOffClip = zeros(numruns, 8);
57 ModPixArray2 = zeros(numruns, 8);
58 NumPixFilt2 = zeros(numruns, 8);
59 BlockInTotal = zeros(numruns, 8,8);
60 j28outTotal = zeros(numruns, 8,8);
61 PatentOutTotal = zeros(numruns, 8,8);
62 Act11Total = zeros(numruns, 8); % Record Act11 for all runs, v2
63 Act22Total = zeros(numruns, 8); % Record Act22 for all runs, v2
64 ActEdgeTotal = zeros(numruns, 8); % Record ActEdge for all runs, v2
65
66 for run = 1:numruns % Number of repeated runs, each time generate and process 1 pair of
    blocks
67     if (randomseedOn==1)
68         rng(run); % Seed the random number generator
69     end
70     qp = round(rand*16)+4; % QP in range 4-20 (Q15J28)
71     fprintf(FIDstats, 'Block %d:\n', run);
72     fprintf(FIDblocks, '\nBlock %d:\n', run);
73     TestFilt2; % Changed to v2
74     DiffDE(run,:) = Diff_de;
75     DiffC(run, :) = Diff_c;
76     DiffF(run, :) = Diff_f;
77     ModPixArray1(run,:) = ModPixels1;
78     FirstOff(run, :) = FirstOffset;
79     SecondOff(run, :) = SecondOffset;
80     ThirdOff(run, :) = ThirdOffset;
81     FirstOffClip(run,:) = FirstOffsetClip;
82     SecondOffClip(run,:) = SecondOffsetClip;
83     ThirdOffClip(run,:) = ThirdOffsetClip;
84     ModPixArray2(run,:) = ModPixels2;
85     NumPixFilt2(run,:) = FiltPixels2;
86     BlockInTotal(run, :, :) = pixin;
87     j28outTotal(run, :, :) = j28out;
88     PatentOutTotal(run, :, :) = BlockOutNoClip;
89     Act11Total(run,:) = Act11block; % v2
90     Act22Total(run,:) = Act22block; % v2
91     ActEdgeTotal(run,:) = ActEdgeblock; % v2
92 end
93
94 fprintf(FIDstats, 'Number of pixels filtered per row, Q15J28: 4\n'); % Q15J28 always
    filters c, d, e and f pixels
95 fprintf(FIDstats, 'Average number of pixels filtered per row, ''841/''659: %f\n', mean(
    mean(NumPixFilt2)));
96 correlationNumPix=corrcoef(ModPixArray1, ModPixArray2);
97 fprintf(FIDstats, 'Correlation of number of pixels with non-zero offsets, Q15J28 vs
    ''841/''659: %f\n', correlationNumPix(1,2));
98
99 j28diff = j28outTotal-BlockInTotal; % Sample-by-sample difference between filtered block
    outputs and block inputs, Q15J28 method
100 patentdiff = PatentOutTotal-BlockInTotal; % Sample-by-sample difference between filtered
    block outputs and block inputs, '841/'659 method
101
102 correlationp1 = corrcoef(j28diff(:, :, 3), patentdiff(:, :, 3));
103 correlationp0 = corrcoef(j28diff(:, :, 4), patentdiff(:, :, 4));
104 correlationq0 = corrcoef(j28diff(:, :, 5), patentdiff(:, :, 5));
105 correlationq1 = corrcoef(j28diff(:, :, 6), patentdiff(:, :, 6));
106 fprintf(FIDstats, 'Correlation of filter adjustment to p1 pixels, Q15J28 method vs

```

```

107 fprintf(FIDstats, 'Correlation of filter adjustment to p0 pixels, Q15J28 method vs
''841/'659 method: %f\n', correlationp0(1,2));
108 fprintf(FIDstats, 'Correlation of filter adjustment to q0 pixels, Q15J28 method vs
''841/'659 method: %f\n', correlationq0(1,2));
109 fprintf(FIDstats, 'Correlation of filter adjustment to q1 pixels, Q15J28 method vs
''841/'659 method: %f\n\n', correlationq1(1,2));
110
111 correlationDE=corrcoef(DiffDE, FirstOff);
112 fprintf(FIDstats, 'Correlation of Diff_de and First Offset: %f\n', correlationDE(1,2));
113 correlationC=corrcoef(DiffC, SecondOff);
114 fprintf(FIDstats, 'Correlation of Diff_c and Second Offset: %f\n', correlationC(1,2));
115 correlationF=corrcoef(DiffF, ThirdOff);
116 fprintf(FIDstats, 'Correlation of Diff_F and Third Offset: %f\n\n', correlationF(1,2));
117
118 % Calculate the frequency of occurrence of Act11, Act22 and ActEdge values, v2
119 Act11Hist = zeros(1,4); % Histogram of Act11 values, first entry counts 0s, fourth entry
counts 3s
120 Act22Hist = zeros(1,4); % Histogram of Act22 values, first entry counts 0s, fourth entry
counts 3s
121 ActEdgeHist = zeros(1,4); % Histogram of ActEdge values, first entry counts 0s, fourth
entry counts 3s
122
123 for i = 1:numruns % For each pair of blocks
124     for j = 1:8 % For each row in the pair of blocks
125         Act11Hist(Act11Total(i,j)+1) = Act11Hist(Act11Total(i,j)+1) + 1;
126         Act22Hist(Act22Total(i,j)+1) = Act22Hist(Act22Total(i,j)+1) + 1;
127         ActEdgeHist(ActEdgeTotal(i,j)+1) = ActEdgeHist(ActEdgeTotal(i,j)+1) + 1;
128     end
129 end
130 fprintf(FIDstats, 'Act11 occurrences : 0: %d, 1: %d, 2: %d, 3:%d\n', Act11Hist);
131 fprintf(FIDstats, 'Act22 occurrences : 0: %d, 1: %d, 2: %d, 3:%d\n', Act22Hist);
132 fprintf(FIDstats, 'ActEdge occurrences: 0: %d, 1: %d, 2: %d, 3:%d\n\n', ActEdgeHist);
133
134 % Calculate the frequency of occurrence of pairs of filter adjustments,
135 % e.g. (-2, -2), (-2, -1) etc., v2
136
137 % Extract each of the sets of pixel difference values into a 1-dimensional
138 % array. 'cdiff' lists the difference values for pixel c in each of the
139 % 8000 rows, etc.
140 cdiff=reshape(j28diff(:,:,3), [], 1);
141 ddiff=reshape(j28diff(:,:,4), [], 1);
142 ediff=reshape(j28diff(:,:,5), [], 1);
143 fdiff=reshape(j28diff(:,:,6), [], 1);
144 pldiff=reshape(patentdiff(:,:,3), [], 1);
145 p0diff=reshape(patentdiff(:,:,4), [], 1);
146 q0diff=reshape(patentdiff(:,:,5), [], 1);
147 qldiff=reshape(patentdiff(:,:,6), [], 1);
148
149 % Count the number of occurrences of each pair of differences within the range (-2, +2)
150 histplc = zeros(5,5);
151 for i=1:(numruns*8)
152     histplc(cdiff(i)+3, pldiff(i)+3) = histplc(cdiff(i)+3, pldiff(i)+3)+1;
153 end
154 histp0d = zeros(5,5);
155 for i=1:(numruns*8)
156     histp0d(ddiff(i)+3, p0diff(i)+3) = histp0d(ddiff(i)+3, p0diff(i)+3)+1;
157 end
158 histq0e = zeros(5,5);
159 for i=1:(numruns*8)
160     histq0e(ediff(i)+3, q0diff(i)+3) = histq0e(ediff(i)+3, q0diff(i)+3)+1;
161 end
162 histqlf = zeros(5,5);
163 for i=1:(numruns*8)
164     histqlf(fdiff(i)+3, qldiff(i)+3) = histqlf(fdiff(i)+3, qldiff(i)+3)+1;

```

```

165 end
166
167 % Print the number of occurrences of each pair of differences
168 fprintf(FIDstats, 'Number of occurrences of each pair of differences, in the range (-2,
+2):\n');
169 fprintf(FIDstats, ' pldiff:');
170 fprintf(FIDstats, ' %4d', -2:2);
171 for i = -2:2
172     fprintf(FIDstats, '\ncdiff %2d:', i);
173     for j = -2:2
174         fprintf(FIDstats, ' %4d', histplc(i+3,j+3));
175     end
176 end
177 fprintf(FIDstats, '\n\n');
178 fprintf(FIDstats, ' p0diff:');
179 fprintf(FIDstats, ' %4d', -2:2);
180 for i = -2:2
181     fprintf(FIDstats, '\nndiff %2d:', i);
182     for j = -2:2
183         fprintf(FIDstats, ' %4d', histp0d(i+3,j+3));
184     end
185 end
186 fprintf(FIDstats, '\n\n');
187 fprintf(FIDstats, ' q0diff:');
188 fprintf(FIDstats, ' %4d', -2:2);
189 for i = -2:2
190     fprintf(FIDstats, '\nediff %2d:', i);
191     for j = -2:2
192         fprintf(FIDstats, ' %4d', histq0e(i+3,j+3));
193     end
194 end
195 fprintf(FIDstats, '\n\n');
196 fprintf(FIDstats, ' q1diff:');
197 fprintf(FIDstats, ' %4d', -2:2);
198 for i = -2:2
199     fprintf(FIDstats, '\nfdiff %2d:', i);
200     for j = -2:2
201         fprintf(FIDstats, ' %4d', histqlf(i+3,j+3));
202     end
203 end
204 fprintf(FIDstats, '\n\n');
205
206 plc_agree = histplc(1,1)+histplc(2,2)+histplc(4,4)+histplc(5,5);
207 plc_disagree = (numruns*8)-histplc(3,3)-plc_agree;
208 fprintf(FIDstats, 'Excluding (0,0), pl and c were filtered the same way %d times and
different ways %d times\n', plc_agree, plc_disagree);
209
210 p0d_agree = histp0d(1,1)+histp0d(2,2)+histp0d(4,4)+histp0d(5,5);
211 p0d_disagree = (numruns*8)-histp0d(3,3)-p0d_agree;
212 fprintf(FIDstats, 'Excluding (0,0), p0 and d were filtered the same way %d times and
different ways %d times\n', p0d_agree, p0d_disagree);
213
214 q0e_agree = histq0e(1,1)+histq0e(2,2)+histq0e(4,4)+histq0e(5,5);
215 q0e_disagree = (numruns*8)-histq0e(3,3)-q0e_agree;
216 fprintf(FIDstats, 'Excluding (0,0), q0 and e were filtered the same way %d times and
different ways %d times\n', q0e_agree, q0e_disagree);
217
218 qlf_agree = histqlf(1,1)+histqlf(2,2)+histqlf(4,4)+histqlf(5,5);
219 qlf_disagree = (numruns*8)-histqlf(3,3)-qlf_agree;
220 fprintf(FIDstats, 'Excluding (0,0), ql and f were filtered the same way %d times and
different ways %d times\n', qlf_agree, qlf_disagree);
221
222
223

```