

```

1 function [vortex, vorticity, cX, cY, excluded] = followVortex(filename, pathname, num, s
   )
2
3     if(nargin < 3)
4         num = 2;
5     end
6
7     if(nargin < 4)
8         s = 1;
9     end
10
11     sp = strsplit(filename, {'_', '.'});
12     n = str2num(sp{2});
13
14     fprintf('followVortex : Traitement de l''image no1 pour les %d vortex\n', s);
15     fprintf('followVortex : Appel à detectVortex \n');
16
17     [tempIni, curIni, rec, thres] = detectVortex(filename, pathname);
18     data = {thres, rec};
19
20     fDir = dir(pathname);
21     spDir = {};
22
23     spDir{1} = strsplit(fDir(end).name, {'_', '.'});
24     spDir{2} = strsplit(fDir(end-1).name, {'_', '.'});
25
26     step = abs(str2num(spDir{1}{2})-str2num(spDir{2}{2}));
27     step
28
29     vortex = {};
30     vorticity = {};
31     cX = [];
32     cY = [];
33     excl = 6;
34
35     copyobj(1,0);
36     figure(2);
37
38     hold on
39     for i = 1:size(tempIni,2)
40         if(size(tempIni{i},2) >= excl)
41             plot(tempIni{i}(1,:), tempIni{i}(2,:), '.');
42         end
43     end
44     hold off
45
46     [xIn, yIn] = ginput(s);
47     close(gcf);
48
49     copyobj(1,0);
50
51     for j = 1:s
52
53         fprintf('followVortex : Suivi du vortex no%d \n', j);
54         tic;
55
56         figure(1);
57
58         [cX{1,j}(1), cY{1,j}(1), ind] = getCloserCell(tempIni, [xIn(j);yIn(j)]);

```

```

59     vortex{1,j} = tempIni{ind};
60     vorticity{1,j} = curIni{ind};
61
62     hold on
63     plot(vortex{1,j}(1,:), vortex{1,j}(2,:), '.', 'Color', 'r');
64     hold off
65     pause(0.3);
66
67     for i = 2:num
68         fprintf('followVortex : Traitement de l\'image no%d \n', i);
69         name = sprintf('%s_%s.%s', ...
70             sp{1}, ...
71             sprintf('%04d', n+step*(i-1)), ...
72             'txt');
73
74         fprintf('followVortex : Appel de detectVortex \n');
75         [temp, cur] = detectVortex(name, pathname, data);
76         [cX{1,j}(i), cY{1,j}(i), ind] = getCloserCell(temp, [cX{1,j}(i-1); cY{1,j}(i-1)]);
77         vortex{i,j} = temp{ind};
78         vorticity{i,j} = cur{ind};
79
80         hold on
81         plot(vortex{i,j}(1,:), vortex{i,j}(2,:), '.', 'Color', 'r');
82         plot(cX{1,j}(:), cY{1,j}(:), '.', 'Color', 'k');
83         hold off
84
85         percent = i*j*100/(s*num);
86         fprintf('followVortex : Traitement à %2.2f %% \n \n', percent);
87         pause(0.3);
88     end
89
90     figure(2);
91
92     hold on
93     plot(cX{1,j}(:), cY{1,j}(:), '.', 'Color', 'k');
94     hold off
95
96     elapsed = toc;
97     fprintf('followVortex : Suivi du vortex no%d achevé en %d secondes \n', ...
98         j, ...
99         elapsed);
100 end
101
102 s = input('followVortex : Voulez vous supprimer des points aberrants ? (o/n) \n', 's');
103 excluded = [];
104
105 if(strcmp(s, 'o'))
106     fprintf('followVortex : Clic droit pour quitter \n');
107
108     while(strcmp(get(gcf, 'SelectionType'), 'normal'))
109         figure(2);
110         recSuppr = getrect;
111
112         fprintf('followVortex : Suppression des points sélectionnés \n');
113
114         sqX = [recSuppr(1), recSuppr(1)+recSuppr(3)];
115         sqY = [recSuppr(2), recSuppr(2)+recSuppr(4)];

```

```

116
117     for i = 1:size(cX,2)
118         iX = find(cX{1,i} >= sqX(1) & cX{1,i} <= sqX(2));
119         iY = find(cY{1,i} >= sqY(1) & cY{1,i} <= sqY(2));
120
121         iFinal = intersect(iX, iY);
122         sFinal = sort(iFinal, 'descend');
123
124         excluded = [excluded; [cX{1,i}(sFinal)' cY{1,i}(sFinal)']];
125         cX{1,i}(sFinal) = [];
126         cY{1,i}(sFinal) = [];
127
128         vorticity(sFinal,i) = {[]};
129         vortex(sFinal,i) = {[]};
130     end
131     hold on
132     plot(excluded(:,1), excluded(:,2), 'x', 'Color', 'r');
133     hold off
134 end
135 end
136
137 end

```