```
function [vor, vorticity, rec, thres] = detectVortex(filename, pathname, data)

    global BLOCK_SIZE;
        global THRESOLD_TOLERANCE;

    [x, y, u, v, cu] = analyseField(filename, pathname);

        pas = x(2)-x(1);

        if(isempty(BLOCK_SIZE))
        BLOCK_SIZE = 2*pas;
    end

        if(isempty(THRESOLD_TOLERANCE))
        THRESOLD_TOLERANCE = 0.7;
    end

    if(nargin < 3)
        fprintf('detectVortex : Veuillez sélectionner le seuil de vorticité \n');
        [xIn, yIn] = ginput(1);

        [mX, iX] = min(abs(x-xIn(1)));
        [mY, iY] = min(abs(y-yIn(1)));

        thres = mean(mean(abs(cu(iX-1:iX+1, iY-1:iY+1)),1),2);

        fprintf('detectVortex : Veuillez sélectionner la zone à étudier \n');
        rec = getrect;
    else
        thres = data{1};
        rec = data{2};
    end

    fprintf('detectVortex : Lancement de la création des groupes à partir du seuillage
        en vorticité \n');
    tic;

        sqX = [rec(1), rec(1)+rec(3)];
        sqY = [rec(2), rec(2)+rec(4)];

        [mrecXL, irecXT] = min(abs(x-sqX(2)));
        [mrecYL, irecYT] = min(abs(y-sqY(2)));
        [mrecXB, irecXB] = min(abs(x-sqX(1)));
        [mrecYB, irecYB] = min(abs(y-sqY(1)));

    [rowF, colF] = find(abs(cu(irecXB:irecXT, irecYB:irecYT)) >= THRESOLD_TOLERANCE*
        thres);
    row = irecXB - 1 +rowF;
    col = irecYB - 1 +colF;

    group = cell(1,length(row));

    for i = 1:length(col)
        indX = find(abs(x(row) - x(row(i))) <= BLOCK_SIZE);
        indY = find(abs(y(col) - y(col(i))) <= BLOCK_SIZE);

        ind = intersect(indX,indY);
        group{i} = [x(row(ind))'; y(col(ind))'];
    end
```

```
58      g = group;
59      cleanGroup = group(~cellfun(@isempty, group));
60
61      elapsed = toc;
62      fprintf('detectVortex : Création des groupes à partir du seuillage en vorticité
            achevée en %d secondes \n',...
63              elapsed);
64
65      fprintf('detectVortex : Lancement de la concaténation des groupes (concatCellsUnique
            ) \n');
66      tic;
67
68      vor = concatCellsUnique(cleanGroup);
69      vor = concatCellsUnique(vor);
70
71      elapsed = toc;
72      fprintf('detectVortex : Concaténation des groupes achevée en %d secondes \n',...
73              elapsed);
74
75      vorticity = cell(1,size(vor,2));
76
77      excl = 6;
78
79      fprintf('detectVortex : Lancement de la suppresion des groupes trop petits (<= 6
            points) \n');
80      tic;
81
82      for i = 1:size(vor,2)
83          if(size(vor{i},2) >= excl)
84              xRep = repmat(x, 1, length(vor{i}(1,:)));
85              xVorRep = repmat(vor{i}(1,:)', 1, length(x));
86
87              dX = abs(xRep-xVorRep');
88              [M, IX] = min(dX);
89
90              yRep = repmat(y, 1, length(vor{i}(2,:)));
91              yVorRep = repmat(vor{i}(2,:)', 1, length(y));
92
93              dY = abs(yRep-yVorRep');
94              [M, IY] = min(dY);
95
96              linInd = sub2ind(size(cu), IX, IY);
97              vorticity{i} = cu(linInd);
98          else
99              vor{i} = {};
100         end
101     end
102
103     vor = vor(~cellfun(@isempty, vor));
104     vorticity = vorticity(~cellfun(@isempty, vorticity));
105
106     elapsed = toc;
107     fprintf('detectVortex : Suppression des groupes trop petits (<= 6 points) achevée en
            %d secondes \n',...
108             elapsed);
109 end
```