randPick (Calls: 500, Time: 1.944 s)

Generated 01-июн.-2021 13:43:50 using performance time.
Function in file E:\course_4\code\randPick.m
Copy to new window for comparing multiple runs

Parents (calling functions)

Function Name	Function Type	Calls
ImageProcessing	Script	500

Lines that take the most time

Line Number	Code	Calls	Total Time (s)	% Time	Time Plot
74	`if(imageGS(y, x) ~= 0)`	5482486	0.210	10.8%	▪
89	`end`	5462005	0.184	9.5%	▪
73	`end`	5482486	0.181	9.3%	▪
71	`if (y <= 0 || x <= 0)`	5482486	0.180	9.3%	▪
88	`end`	5462005	0.177	9.1%	▪
All other lines			1.012	52.1%	▬
Totals			1.944	100%	

Children (called functions)

Function Name	Function Type	Calls	Total Time (s)	% Time	Time Plot
deg2rad	Function	500	0.005	0.3%	
Self time (built-ins, overhead, etc.)			1.939	99.7%	▬
Totals			1.944	100%	

Code Analyzer results

Line Number	Message
18	`The value assigned to variable 'yPx' might be unused.`
21	`If you intend to specify expression precedence, use parentheses () instead of brackets [].`
22	`If you intend to specify expression precedence, use parentheses () instead of brackets [].`
75	`The variable 'coloredX' appears to change size on every loop iteration. Consider preallocat…`
76	`The variable 'coloredY' appears to change size on every loop iteration. Consider preallocat…`

Coverage results

Show coverage for parent folder

Total lines in function	99
Non-code lines (comments, blank lines)	25
Code lines (lines that can run)	74
Code lines that did run	72
Code lines that did not run	2
Coverage (did run/can run)	97.30 %

Function listing

```
   Time        Calls      Line
                            1  function [randX, randY] = randPick(imageGS, spotX, spotY)
                            2
                            3      % параметры устройства
 < 0.001         500        4      d = 0.0003;     % диаметр отверстия
                            5
   0.062         500        6      format long
 < 0.001         500        7      height = 4e-3;
 < 0.001         500        8      width = 4e-3;
                            9
 < 0.001         500       10      pxH = length(imageGS);
   0.005         500       11      pxW = length(imageGS(1, :));
 < 0.001         500       12      pxSize = width / pxW;
                           13
```

```
0.012        500    14   angle = deg2rad(0 : 359);          % вспомогательный массив углов
0.018        500    15   defCircleX = (d/2) * cos(angle);            % координата x контура пятна
0.007        500    16   defCircleY = (d/2) * sin(angle);            % координата y контура пятна
0.003        500    17   xPx = ceil((defCircleX + width/2) / pxSize);  % x - контур пятна в пикселях
0.003        500    18   yPx = ceil((defCircleY + height/2) / pxSize);  % y - контур пятна в пикселях
0.002        500    19   defPxRadius = ceil((max(xPx) - min(xPx)) / 2);
                    20
< 0.001      500    21   coloredX = [spotX];
< 0.001      500    22   coloredY = [spotY];
                    23
                    24   % Т.К. ПОЛУЧЕННОЕ ПЯТНО 100% МЕНЬШЕ ЧЕМ ТО, ЧТО ПОЛУЧАЕТСЯ БЕЗ НАКЛОНА
                    25   % МАТРИЦЫ МОЖНО ПОИСКАТЬ ПИКСЕЛИ В ПРЕДЕЛАХ ДИАМЕТРА НЕИЗМЕНЁНОГО КРУГА
                    26   % В 4 КВАДРАНТАХ
                    27
                    28   % левый верхний квадрат граница
< 0.001      500    29   leftTopX = coloredX(1) - defPxRadius *  2;
< 0.001      500    30   leftTopY = coloredY(1) - defPxRadius * 2;
                    31   % левый нижний квадрат граница
< 0.001      500    32   leftBottomX = coloredX(1) - defPxRadius * 2;
< 0.001      500    33   leftBottomY = coloredY(1) + defPxRadius * 2;
                    34   % правый верхний квадрат граница
< 0.001      500    35   rightTopX = coloredX(1) + defPxRadius * 2;
< 0.001      500    36   rightTopY = coloredY(1) - defPxRadius * 2;
                    37   % правый нижний квадрат
< 0.001      500    38   rightBottomX = coloredX(1) + defPxRadius * 2;
< 0.001      500    39   rightBottomY = coloredY(1) + defPxRadius * 2;
                    40
< 0.001      500    41   quadroPositionsX = [leftTopX, rightTopX, rightBottomX, leftBottomX];
< 0.001      500    42   quadroPositionsY = [leftTopY, rightTopY, rightBottomY, leftBottomY];
                    43
                    44   % проверка не выходят ли координаты за границы матрицы, если выходят ставим
                    45   % = размер матрицы или 0
< 0.001      500    46   for i = 1 : 4
< 0.001     2000    47      if (quadroPositionsX(i) <= 0)
  0.001      108    48         quadroPositionsX(i) = 1;
< 0.001     1892    49      else
< 0.001     1892    50          if (quadroPositionsX(i) > pxW)
< 0.001        8    51              quadroPositionsX(i) = pxW;
< 0.001     1892    52          end
< 0.001     2000    53      end
< 0.001     2000    54      if (quadroPositionsY(i) <= 0)
< 0.001      112    55         quadroPositionsY(i) = 1;
< 0.001     1888    56      else
< 0.001     1888    57          if (quadroPositionsY(i) > pxH)
< 0.001       10    58              quadroPositionsY(i) = pxH;
< 0.001     1888    59          end
< 0.001     2000    60      end
  0.002     2000    61   end
                    62
< 0.001      500    63   isFlag = false;
< 0.001      500    64   isFlag2 = false;
< 0.001      500    65   coloredX = [];
< 0.001      500    66   coloredY = [];
                    67
< 0.001      500    68   for y = quadroPositionsY(1) : quadroPositionsY(3)
  0.002    55206    69       columnFlag = false;
  0.004    55206    70      for x = quadroPositionsX(1) : quadroPositionsX(3)
  0.180  5482486    71          if (y <= 0 || x <= 0)
                    72             continue;
  0.181  5482486    73          end
  0.210  5482486    74           if(imageGS(y, x) ~= 0)
  0.169   746802    75               coloredX(end + 1) = x;
  0.125   746802    76               coloredY(end + 1) = y;
                    77
  0.023   746802    78               columnFlag = true;
  0.023   746802    79               isFlag2 = true;
                    80
  0.024   746802    81               if (length(coloredX) == 2)
< 0.001      450    82                  isFlag = true;
  0.022   746802    83               end
  0.158      4735684     84           else
  0.156      4735684     85             if (columnFlag)
  0.001        20481     86               break;
  0.154      4715203     87             end
  0.177      5462005     88           end
  0.184      5462005     89         end
  0.002        55206     90         if (isFlag)
  0.001        35978     91           if (~isFlag2)
                         92             break;
  0.001        35978     93           end
  0.002        55206     94         end
  0.004        55206     95       end
                         96
  0.002          500     97       randX = ceil(sum(coloredX) / length(coloredX));
< 0.001          500     98       randY = ceil(sum(coloredY) / length(coloredY));
  0.010          500     99   end
```