randPick (Calls: 1000, Time: 72.540 s)

Generated 28-дек.-2020 04:20:57 using performance time.
Function in file C:\Users\yarba\Desktop\ДО\course_4\code\randPick.m
Copy to new window for comparing multiple runs

Parents (calling functions)

Function Name	Function Type	Calls
ImageProcessing	Script	1000

Lines that take the most time

Line Number	Code	Calls	Total Time (s)	% Time	Time Plot
120	`if(squeeze(imageGS(j, i, :)) ~= 0)`	22457164	58.316	80.4%	
57	`if(squeeze(imageGS(randY, randX, :)) ~= 0)`	1298044	3.988	5.5%	
45	`if (any(repeat(:) == 0))`	15529	2.097	2.9%	
41	`randX = randi(columns, 1);`	1313573	1.860	2.6%	
42	`randY = randi(rows, 1);`	1313573	1.451	2.0%	
All other lines			4.829	6.7%	
Totals			72.540	100%	

Children (called functions)

Function Name	Function Type	Calls	Total Time (s)	% Time	Time Plot
squeeze	Function	23755208	22.020	30.4%	
deg2rad	Function	1000	0.007	0.0%	
Self time (built-ins, overhead, etc.)			50.514	69.6%	
Totals			72.540	100%	

Code Analyzer results

Line Number	Message
4	`The value assigned to variable 't' might be unused.`
5	`The value assigned to variable 'h' might be unused.`
18	`The value assigned to variable 'yPx' might be unused.`
23	`The value assigned to variable 'x' might be unused.`
24	`The value assigned to variable 'y' might be unused.`
29	`The value assigned to variable 'currentX' might be unused.`
30	`The value assigned to variable 'currentY' might be unused.`
47	`Consider using newline, semicolon, or comma before this statement for readability.`
58	`The variable 'coloredX2' appears to change size on every loop iteration. Consider prealloca…`
59	`The variable 'coloredY2' appears to change size on every loop iteration. Consider prealloca…`
65	`Using ISEMPTY is usually faster than comparing LENGTH to 0.`
65	`Using ISEMPTY is usually faster than comparing LENGTH to 0.`
67	`The value assigned to variable 'ranY' might be unused.`
121	`The variable 'coloredX2' appears to change size on every loop iteration. Consider prealloca…`
122	`The variable 'coloredY2' appears to change size on every loop iteration. Consider prealloca…`
123	`The value assigned to variable 'find' might be unused.`

Coverage results

Show coverage for parent folder

Total lines in function	130
Non-code lines (comments, blank lines)	38
Code lines (lines that can run)	92
Code lines that did run	89
Code lines that did not run	3
Coverage (did run/can run)	96.74 %

Function listing

Time	Calls	Line	
		1	function [randX, randY] = randPick(imageGS)
		2	% параметры устройства
< 0.001	1000	3	d = 0.0003; % диаметр отверстия
< 0.001	1000	4	t = 0.00005; % толщина отверстия
< 0.001	1000	5	h = 0.0007; % высота отверстия
		6	
0.058	1000	7	format long
< 0.001	1000	8	height = 4.51e-3; % Размеры матрицы
< 0.001	1000	9	width = 2.88e-3;
< 0.001	1000	10	pxW = 752;
< 0.001	1000	11	pxH = 480;
< 0.001	1000	12	pxSize = width / pxW;
		13	
0.015	1000	14	angle = deg2rad(0 : 359); % вспомогательный массив углов
0.023	1000	15	defCircleX = (d/2) * cos(angle); % координата x контура пятна
0.009	1000	16	defCircleY = (d/2) * sin(angle); % координата y контура пятна
0.004	1000	17	xPx = ceil((defCircleX + width/2) / pxSize); % x - контур пятна в пикселях
0.003	1000	18	yPx = ceil((defCircleY + height/2) / pxSize); % y - контур пятна в пикселях
0.003	1000	19	defPxRadius = ceil((max(xPx) - min(xPx)) / 2);
		20	
< 0.001	1000	21	columns = 752;
< 0.001	1000	22	rows = 480;
0.003	1000	23	x = 1:columns;
0.002	1000	24	y = 1:rows;
		25	
< 0.001	1000	26	coloredX2 = [];
< 0.001	1000	27	coloredY2 = [];
		28	
< 0.001	1000	29	currentX = 0;
< 0.001	1000	30	currentY = 0;
		31	
< 0.001	1000	32	find = false;
0.030	1000	33	repeat = zeros(pxH, pxW);
		34	
		35	% поиск первой точки пятна
< 0.001	1000	36	iter = 1;
< 0.001	1000	37	while(find == 0)
0.048	1313680	38	if (iter > 10000)
< 0.001	107	39	break;
0.049	1313573	40	end
1.860	1313573	41	randX = randi(columns, 1);
1.451	1313573	42	randY = randi(rows, 1);
		43	
0.303	1313573	44	if (repeat(randY, randX) == 1)
2.097	15529	45	if (any(repeat(:) == 0))
0.003	15529	46	continue;
		47	else break;
		48	end
0.046	1298044	49	end
		50	
0.057	1298044	51	repeat(randY, randX) = 1;
		52	% repeat(end + 1) = [randX, randY];
		53	
		54	% repeatX(end + 1) = randX;
		55	% repeatY(end + 1) = randY;
		56	
3.988	1298044	57	if(squeeze(imageGS(randY, randX, :)) ~= 0)
0.009	893	58	coloredX2(end + 1) = randX;
0.003	893	59	coloredY2(end + 1) = randY;
< 0.001	893	60	find = true;
0.041	1298044	61	end
0.046	1298044	62	iter = iter + 1;
0.064	1298044	63	end
		64	
< 0.001	1000	65	if (length(coloredX2) == 0 || length(coloredY2) == 0)
< 0.001	107	66	randX = 0;
< 0.001	107	67	ranY = 0;
0.025	107	68	return;
< 0.001	893	69	end
		70	% Т.К. ПОЛУЧЕННОЕ ПЯТНО 100% МЕНЬШЕ ЧЕМ ТО, ЧТО ПОЛУЧАЕТСЯ БЕЗ НАКЛОНА

		71	% МАТРИЦЫ МОЖНО ПОИСКАТЬ ПИКСЕЛИ В ПРЕДЕЛАХ ДИАМЕТРА НЕИЗМЕННЁНОГО КРУГА
		72	% В 4 КВАДРАНТАХ
		73	
		74	% левый верхний квадрат граница
< 0.001	893	75	leftTopX = coloredX2(1) - defPxRadius * 2;
< 0.001	893	76	leftTopY = coloredY2(1) - defPxRadius * 2;
		77	% левый нижний квадрат граница
< 0.001	893	78	leftBottomX = coloredX2(1) - defPxRadius * 2;
< 0.001	893	79	leftBottomY = coloredY2(1) + defPxRadius * 2;
		80	% правый верхний квадрат граница
< 0.001	893	81	rightTopX = coloredX2(1) + defPxRadius * 2;
< 0.001	893	82	rightTopY = coloredY2(1) - defPxRadius * 2;
		83	% правый нижний квадрат
< 0.001	893	84	rightBottomX = coloredX2(1) + defPxRadius * 2;
< 0.001	893	85	rightBottomY = coloredY2(1) + defPxRadius * 2;
		86	
0.001	893	87	quadroPositionsX = [leftTopX, rightTopX, rightBottomX, leftBottomX];
< 0.001	893	88	quadroPositionsY = [leftTopY, rightTopY, rightBottomY, leftBottomY];
		89	
		90	% проверка не выходят ли координаты за границы матрицы, если выходят ставим
		91	% = размер матрицы или 0
0.001	893	92	for i = 1 : 4
< 0.001	3572	93	if (quadroPositionsX(i) <= 0)
< 0.001	32	94	quadroPositionsX(i) = 1;
< 0.001	3540	95	else
< 0.001	3540	96	if (quadroPositionsX(i) > pxW)
< 0.001	22	97	quadroPositionsX(i) = pxW - 1;
< 0.001	3540	98	end
< 0.001	3572	99	end
< 0.001	3572	100	if (quadroPositionsY(i) <= 0)
< 0.001	96	101	quadroPositionsY(i) = 1;
< 0.001	3476	102	else
< 0.001	3476	103	if (quadroPositionsY(i) > pxH)
< 0.001	76	104	quadroPositionsY(i) = pxH - 1;
< 0.001	3476	105	end
< 0.001	3572	106	end
0.002	3572	107	end
		108	
		109	% imageProcessed(leftTopY:rightTopY, leftTopX:rightTopX, [2 2 2]) = 5;
		110	% imageProcessed(rightTopY:rightBottomY, rightTopX:rightBottomX, [2 2 2]) = 5;
		111	% imageProcessed(rightBottomY:leftBottomY, leftBottomX:rightBottomX, [2 2 2]) = 5;
		112	% imageProcessed(leftTopY:leftBottomY, leftBottomX:leftTopX, [2 2 2]) = 5;
		113	
		114	
< 0.001	893	115	for i = quadroPositionsX(1) : quadroPositionsX(3)
0.015	142836	116	for j = quadroPositionsY(1) : quadroPositionsY(3)
0.889	22457164	117	if (i <= 0 || j <= 0)
		118	continue;
0.723	22457164	119	end
58.316	22457164	120	if(squeeze(imageGS(j, i, :)) ~= 0)
0.362	1483652	121	coloredX2(end + 1) = i;
0.267	1483652	122	coloredY2(end + 1) = j;
0.054	1483652	123	find = true;
0.700	22457164	124	end
0.832	22457164	125	end
0.010	142836	126	end
		127	
0.003	893	128	randX = ceil(sum(coloredX2) / length(coloredX2));
0.001	893	129	randY = ceil(sum(coloredY2) / length(coloredY2));
0.106	893	130	end