breadthSearch (Calls: 500, Time: 16.202 s)

Generated 01-июн.-2021 13:42:44 using performance time.
Function in file E:\course_4\code\breadthSearch.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
67	`frontier.push(neighboors(i, :));`	746348	6.713	41.4%	
26	`current = frontier.pop();`	746798	2.902	17.9%	
25	`isOkay = boolean([1 1 1 1]);`	746798	2.129	13.1%	
71	`end`	746798	1.781	11.0%	
33	`neighboors = [neighboor_r; neighboor_l; neighboo…`	746798	0.315	1.9%	
All other lines			2.364	14.6%	
Totals			16.202	100%	

Children (called functions)

Function Name	Function Type	Calls	Total Time (s)	% Time	Time Plot
CQueue>CQueue.push	Class method	746798	5.658	34.9%	
CQueue>CQueue.pop	Class method	746798	2.143	13.2%	
boolean	Function	746849	1.310	8.1%	
CQueue>CQueue.size	Class method	747248	1.033	6.4%	
CQueue>CQueue.CQueue	Class method	450	0.010	0.1%	
Self time (built-ins, overhead, etc.)			6.048	37.3%	
Totals			16.202	100%	

Code Analyzer results

Line Number	Message
21	`If you intend to specify expression precedence, use parentheses () instead of brackets [].`

Coverage results

Show coverage for parent folder

Total lines in function	76
Non-code lines (comments, blank lines)	21
Code lines (lines that can run)	55
Code lines that did run	54
Code lines that did not run	1
Coverage (did run/can run)	98.18 %

Function listing

Time	Calls	Line	
		1	`function [breadthX, breadthY] = breadthSearch(imageGS, spotX, spotY)`
0.001	500	2	`    if (spotX == 0 || spotY == 0)`
< 0.001	50	3	`        breadthX = 0;`
< 0.001	50	4	`        breadthY = 0;`
< 0.001	50	5	`        return;`
< 0.001	450	6	`    end`
		7	
< 0.001	450	8	`    pxH = length(imageGS);`
0.005	450	9	`    pxW = length(imageGS(1, :));`
		10	

< 0.001	450	11	breadthX = spotX;
< 0.001	450	12	breadthY = spotY;
		13	
< 0.001	450	14	sumY = breadthX;
< 0.001	450	15	sumX = breadthY;
< 0.001	450	16	sizeX = 1;
< 0.001	450	17	sizeY = 1;
		18	
0.038	450	19	frontier = CQueue();
0.018	450	20	frontier.push([breadthX, breadthY]);
< 0.001	450	21	maxReached = [imageGS];
0.098	450	22	maxReached(breadthY, breadthX) = 0;
		23	
0.005	450	24	while (frontier.size() ~= 0)
2.129	746798	25	isOkay = boolean([1 1 1 1]);
2.902	746798	26	current = frontier.pop();
		27	
		28	% определение соседей текущей клетки
0.223	746798	29	neighboor_r = [current(1) + 1, current(2)];
0.143	746798	30	neighboor_l = [current(1) - 1, current(2)];
0.139	746798	31	neighboor_t = [current(1), current(2) + 1];
0.142	746798	32	neighboor_b = [current(1), current(2) - 1];
0.315	746798	33	neighboors = [neighboor_r; neighboor_l; neighboor_t; neighboor_b];
		34	
		35	% проверка, чтобы соседи не выходили за матрицу
0.036	746798	36	for i = 1 : length(neighboors)
		37	% neighboors(i, :)
0.112	2987192	38	if (neighboors(i, 1) > pxW || neighboors(i, 1) <= 0)
< 0.001	25	39	isOkay(i) = boolean(0);
< 0.001	25	40	neighboors(i, 1) = -1;
0.105	2987192	41	end
		42	
0.106	2987192	43	if (neighboors(i, 2) > pxH || neighboors(i, 2) <= 0)
< 0.001	26	44	isOkay(i) = boolean(0);
< 0.001	26	45	neighboors(i, 1) = -1;
0.107	2987192	46	end
		47	% neighboors(i, :)
		48	% isOkay(i)
0.120	2987192	49	end
		50	
0.036	746798	51	for i = 1 : length(neighboors)
0.122	2987192	52	if (~isOkay(i))
< 0.001	51	53	continue;
0.102	2987141	54	end
		55	
0.179	2987141	56	if (maxReached(neighboors(i, 2), neighboors(i, 1)) ~= 0)
0.064	746348	57	if (imageGS(neighboors(i, 2), neighboors(i, 1)) == 0)
		58	continue;
0.030	746348	59	end
		60	
0.026	746348	61	sumY = sumY + neighboors(i, 2);
0.029	746348	62	sumX = sumX + neighboors(i, 1);
		63	
0.025	746348	64	sizeX = sizeX + 1;
0.025	746348	65	sizeY = sizeY + 1;
		66	
6.713	746348	67	frontier.push(neighboors(i, :));
0.046	746348	68	maxReached(neighboors(i, 2), neighboors(i, 1)) = 0;
0.112	2987141	69	end
0.117	2987141	70	end
1.781	746798	71	end
		72	
< 0.001	450	73	breadthX = ceil(sumX / sizeX);
< 0.001	450	74	breadthY = ceil(sumY / sizeY);
		75	
0.046	450	76	end