RayTracer>RayTracer.rayTrace (Calls: 1, Time: 2450.109 s)

Generated 27-Mar-2023 12:52:54 using performance time.
Class method in file /Users/lhess/ffr-photon-simulator-matlab/mask-simulator/RayTracer.m
Copy to new window for comparing multiple runs

Parents (calling functions)

Lines that take the most time

Line Number	Code	Calls	Total Time (s)	% Time	Time Plot
309	`currentQuadrant = obj.findCurrentQuadrant(photon);`	23374568	661.725	27.0%	
280	`photonPaths(pathsIdx,:) = previousPhotonCoords;`	233765	494.888	20.2%	
298	`obj.currFFRLayer = obj.findCurrFFRLayer(ffr, photon);`	23374568	488.035	19.9%	
290	`[hasCrossedFFRBound, crossedFFRBound] = obj.checkIfAtF…`	23376568	365.806	14.9%	
310	`[hasReflected, reflectedFiberCoords] = obj.checkIfRefl…`	23374568	203.748	8.3%	
All other lines			235.907	9.6%	
Totals			2450.109	100%	

Children (called functions)

Function Name	Function Type	Calls	Total Time (s)	% Time	Time Plot
RayTracer>RayTracer.findCurrentQuadrant	Class method	23374568	625.637	25.5%	
RayTracer>RayTracer.findCurrFFRLayer	Class method	23374568	454.754	18.6%	
RayTracer>RayTracer.checkIfAtFFRBound	Class method	23376568	349.589	14.3%	
RayTracer>RayTracer.checkIfReflected	Class method	23374568	183.052	7.5%	
RayTracer>RayTracer.isAtInteriorBound	Class method	23374568	76.789	3.1%	
RayTracer>RayTracer.movePhoton	Class method	23376568	54.576	2.2%	
Photon>Photon.getCoords	Class method	23376568	22.973	0.9%	
Debug>Debug.msg	Class method	4000	0.368	0.0%	
RayTracer>RayTracer.findCrossedBound	Class method	18542	0.188	0.0%	
InteriorBoundary>InteriorBoundary.addCrossing	Class method	18542	0.151	0.0%	
RayTracer>RayTracer.calculateNewSteps	Class method	24209	0.077	0.0%	
Photon>Photon.setSteps	Class method	24209	0.057	0.0%	
FFRBoundary>FFRBoundary.addCrossing	Class method	2000	0.020	0.0%	
RayTracer>RayTracer.resetCurrFFRLayer	Class method	2000	0.011	0.0%	
Self time (built-ins, overhead, etc.)			681.866	27.8%	
Totals			2450.109	100%	

Code Analyzer results

No Code Analyzer messages.

Coverage results

Show coverage for parent folder

Total lines in function	71
Non-code lines (comments, blank lines)	32
Code lines (lines that can run)	39
Code lines that did run	39
Code lines that did not run	0
Coverage (did run/can run)	100.00 %

Function listing

```
Time      Calls    Line
                    251          function [photonPaths, boundInfo] = rayTrace(obj, ffr, incomingPhotons)
                    252            % Ray traces photons starting from initialCoords through an entire FFR.
< 0.001         1   253            boundInfo = [];
                    254            % Preallocate a massive photonPaths array.
```

```
                     % Increase size to 10.000.000.
< 0.001         1    photonPaths = zeros(10000000,2);
                     % We need to keep track of the position within the photonPaths array
                     % so we can overwrite the preallocated nan values. Increment this
                     % each time coordinates are added to photonPaths.
< 0.001         1    pathsIdx = 1;

                     % Get number of rows in first column.
< 0.001         1    nPhotons = size(incomingPhotons, 1);

                     % Iterate through each incoming photon.
< 0.001         1    for photonNum = 1:nPhotons
  0.224      2000      Debug.msg("Incident photon number " + photonNum, 0);
  0.020      2000      photon = incomingPhotons(photonNum);
                       % Initialize values:
< 0.001      2000      hasCrossedFFRBound = false;
  0.017      2000      obj.resetCurrFFRLayer();
                       % Reflect the photon until it reaches a boundary.
< 0.001      2000      while hasCrossedFFRBound == false
                         % Update the previous FFR layer.
 33.472  23376568        obj.prevFFRLayer = obj.currFFRLayer;
                         % We need to track the previous photon's coordinates to determine the reflected path.
 37.139  23376568        previousPhotonCoords = photon.getCoords(); % [x y]
                         % Only store the photon coordinates every 1000 moves.
  1.765  23376568        if rem(pathsIdx, 100) == 0
494.888    233765          photonPaths(pathsIdx,:) = previousPhotonCoords;
  0.526  23376568        end
  0.566  23376568        pathsIdx = pathsIdx + 1;
                         % Move the photon and check if it has reflected or has crossed a boundary
 65.881  23376568        obj.movePhoton(photon);
                         % We want to record any boundary crossings. The photon can either cross an FFR bound or
                         % an interior bound.
                         %  - If it crosses an FFR bound, we move to the next photon, and do not check for reflection.
                         %  - If it crosses an interior bound, it could also potentially have  reflected off a fiber
                         %    lying immediately past that bound.
365.806  23376568        [hasCrossedFFRBound, crossedFFRBound] = obj.checkIfAtFFRBound(photon, ffr);
  0.536  23376568        if hasCrossedFFRBound == true
                           % Move to the next incident photon if the current one has left the FFR.
  0.027      2000          crossedFFRBound.addCrossing(photon);
  0.195      2000          Debug.msg('Photon ' + string(photonNum) + ' reached ffr bound: ' + crossedFFRBound.type, 0);
  0.569  23374568        else
                           %Debug.msg('Not at FFR bound. Check if at interior bound.', 1);
                           % Update the current FFR Layer.
488.035  23374568          obj.currFFRLayer = obj.findCurrFFRLayer(ffr, photon);
                           % Check for interior bound crossings.
 90.799  23374568          if obj.isAtInteriorBound()
                             %Debug.msg("At interior bound.", 1);
  0.288     18542            [crossedInteriorBound, direction] = obj.findCrossedBound(photon);
  0.191     18542            crossedInteriorBound.addCrossing(photon, direction);
  0.508  23356026          else
                             %Debug.msg('Not at interior bound.', 1);
  0.514  23374568          end
                           % Check for reflection off a fiber.
                           %Debug.msg('Check if reflected.', 1);
661.725  23374568          currentQuadrant = obj.findCurrentQuadrant(photon);
203.748  23374568          [hasReflected, reflectedFiberCoords] = obj.checkIfReflected(photon, currentQuadrant);
  0.597  23374568          if hasReflected == true
                             % Calculate the new steps and make a new Photon with those steps.
  0.023     24209            reflectionPoint = [photon.x, photon.y];
  0.143     24209            [newXStep, newYStep] = obj.calculateNewSteps(reflectionPoint, previousPhotonCoords, reflectedF
  0.093     24209            photon.setSteps(newXStep, newYStep);
                             %Debug.msg('Photon ' + string(photonNum) + ' reflected at fiber: ' + obj.coordToString(reflect
  0.537  23374568          end
  0.527  23376568        end
  0.746  23376568      end
< 0.001      2000    end
```

< 0.001	1	**321**	end

Local functions in this file are not included in this listing.