

```

                                gather_regions_delta
function region_matrix = gather_regions_delta(N,S_mat,delta);

% THIS IS ANOTHER VERSION OF A SIMILAR FUNCTION EXCEPT THIS USES A LOCAL TEST
CONDITION, INSTEAD OF GLOBAL CATEGORIES

% This function takes in an array that contains the category of each region of an
image, based upon the S-score of the region.
% It returns a matrix that labels each region as part of some larger region, in
effect, grouping individual regions into larger ones.

region_number = 0; %an arbitrary number used to distinguish between the larger
regions

region_matrix = zeros(N,N); %a matrix populated with region numbers to keep track of
what regions of the image belong together as a single larger region

Is_taken = zeros(N,N);

[i j max_s] = find_max(S_mat);

while(max_s > 0) %this loop iterates until all entries in S_mat have been set to
zero

    if(Is_taken(i,j) == 0)

        region_number = region_number + 1;
        Is_taken(i,j) = 1;
        region_matrix(i,j) = region_number;

        clear temp_region_array;
        temp_region_array{1} = [i,j];
        pointer = 0;
        index = 1;
        anchor = S_mat(i,j);

        while(pointer < index)

            pointer = pointer + 1;
            x = temp_region_array{pointer};
            k = x(1);
            m = x(2);

            [up_neighbor down_neighbor left_neighbor right_neighbor] =
local_test(N,k,m,Is_taken,S_mat,anchor,delta);

            if(up_neighbor == 1)

                index = index + 1;

```

```

                                gather_regions_delta
    temp_region_array{index} = [k-1,m];
    Is_taken(k-1,m) = 1;
    region_matrix(k-1,m) = region_number;

endif

if(down_neighbor == 1)

    index = index + 1;
    temp_region_array{index} = [k+1,m];
    Is_taken(k+1,m) = 1;
    region_matrix(k+1,m) = region_number;

endif

if(left_neighbor == 1)

    index = index + 1;
    temp_region_array{index} = [k,m-1];
    Is_taken(k,m-1) = 1;
    region_matrix(k,m-1) = region_number;

endif

if(right_neighbor == 1)

    index = index + 1;
    temp_region_array{index} = [k,m+1];
    Is_taken(k,m+1) = 1;
    region_matrix(k,m+1) = region_number;

endif

endwhile

endif

S_mat(i,j) = 0;
[i j max_s] = find_max(S_mat);

endwhile

endfunction

```