

```

                                generate_data_tree
function [category_tree delta_tree anchor_tree] = generate_data_tree(data_array)

[x num_items] = size(data_array);

depth = 1;
top_num_categories = 1;
num_leaves = 0;

category_tree{1,depth} = data_array;
temp = [Inf Inf Inf];
anchor_tree{1, depth} = temp;
delta_tree{1,depth} = Inf;

%this main loop repeatedly calls the optimize categories algorithm until each item
in the data array appears as a leaf
while(num_leaves < num_items && top_num_categories > 0)

    tree_index = 1;
    i = 1;

    %this loop runs at a given depth, and generates the next lower level of the tree
    while(i <= top_num_categories && num_leaves < num_items && top_num_categories > 0)

        temp_data = category_tree{i,depth};

        [x items_in_category] = size(temp_data)

        %tests whether the top level category is a leaf
        %if the test fails, then the top level category is a leaf, and we do not run the
optimize categories algorithm
        if(items_in_category > 1 && iscell(temp_data))

            [data_categories_array category_vec anchor_array H_final delta] =
optimize_categories_3D(temp_data,0);
            [x sub_num_categories] = size(data_categories_array)

            %if this is false, then running the algorithm did not create any categories
            if(sub_num_categories < items_in_category && sub_num_categories > 1)

                %this loop loads the categories generated into the next lower level of the
tree
                for j = 1 : sub_num_categories

                    [x items_in_category] = size(data_categories_array{j});

                    %if true, then we're adding a new node to the tree that is not a leaf
                    if(iscell(data_categories_array{j}) && items_in_category > 1)

```

```

                                generate_data_tree
        category_tree{tree_index, depth + 1} = data_categories_array{j};

%if false, then we're adding a leaf, and we want to ensure it is added as
a vector
elseif(iscell(data_categories_array{j}))

        category_tree{tree_index, depth + 1} = data_categories_array{j};
        num_leaves = num_leaves + 1;

else

        category_tree{tree_index, depth + 1} = data_categories_array;
        num_leaves = num_leaves + 1;

endif

        %in each case we update the delta tree and anchor tree, and increase the
tree_index
        delta_tree{tree_index, depth + 1} = delta;
        anchor_tree{tree_index, depth + 1} = anchor_array{j};
        tree_index = tree_index + 1;

    endfor

    endif

endif

clear data_categories_array
i = i + 1;

endwhile

depth = depth + 1
top_num_categories = tree_index - 1

endwhile

endfunction

```