```
function [final_avg_matrix final_indexes] = partition_image_vectorized(I)

img_matrix_R = I(:,:,1);
img_matrix_G = I(:,:,2);
img_matrix_B = I(:,:,3);

%these values will depend upon application, but these should be fine in general
N_min = 5;
N_max = 25;

max_total_diff = 0;

for N = N_min : N_max

  [total_diff_R avg_matrix_R indexes] = test_image_consistency(img_matrix_R,N);

  %calls a more efficient version once we've solved for the indexes
  [total_diff_G avg_matrix_G] = calculate_total_im_diff(indexes, img_matrix_G, N);
  [total_diff_B avg_matrix_B] = calculate_total_im_diff(indexes, img_matrix_B, N);

  current_total_diff = (total_diff_R + total_diff_G + total_diff_B)/N^2;

  if(current_total_diff > max_total_diff)

    max_total_diff = current_total_diff;
```

```
    final_avg_matrix = [avg_matrix_R; avg_matrix_G; avg_matrix_B];
    final_indexes = indexes;

  endif

endfor

endfunction
```