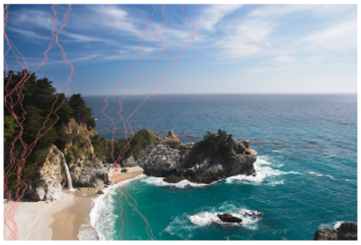
Image carving

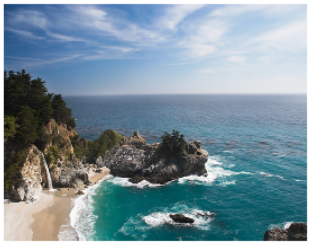
This lab will focus on the concepts of image resizing utilizing the principals supporting minimum-energy seam carving and dynamic programming. The goal of this part of the project is to help you understand an important approach towards resizing images while attempting to preserve the integrity of important image information.

For this lab, we will be testing you strictly on shrinking an image (making it smaller), by sequentially removing vertical seams of minimum energy.

Original Image



Vertical seams with lowest energy



Reduced image

We begin by computing the energy map by

$$E = |\partial I / \partial x| + |\partial I / \partial y|$$

Note the this part has been already coded. Your task is to fill in the code that follows. For this project, we will be testing you strictly on shrinking an image (making it smaller), though the ideas applied towards enlarging an image are very similar to those of shrinking one.

The image shrinking will be achieved by sequentially removing vertical seams of minimum cumulative energy E. The optimal vertical seam (in the sense of minimum cumulative energy) can be found using dynamic programming. The first step is to traverse the image from the second row to the last row and compute the cumulative minimum energy M_x for all possible connected seams for each entry (i, j):

$$M_x(i,j) = E(i,j) + \min\{M_x(i-1,j-1), M_x(i-1,j), M_x(i-1,j+1)\}$$

At the end of this process, the minimum value of the last row in M_x will indicate the end of the minimal connected vertical seam. Note that you have to record a backtrack table along the way. Hence, in the second step we backtrack from this minimum entry on M_x to find the path of the optimal seam. Also, in order to maintain consistency, if the minimum value is found at multiple indices, choose the smaller index. Finally, this process is iterated several times, as many as indicated by the input argument N .

The original image can be attached below. Simply right click on the image and click on "Save As.." option from the menu:

Your Function

Save Reset MATLAB Documentation (https://www.mathworks.com/help/)

```
function Ic = ImageCarving(N)

% N: number of vertical seams you have to remove


% read image
I = im2double(imread('waterfall.png'));

% get grayscale image
Ig0 = rgb2gray(im2double(I));

% colored image
Ic = I;


for iIter = 1:N

    Ig = rgb2gray(Ic);
    Gx = imfilter(Ig,.5*[-1 0 1],'replicate');
    Gy = imfilter(Ig,.5*[-1 0 1]','replicate');
    E = abs(Gx) +  abs(Gy); % energy


    M = zeros(size(Ig)); % cumulative energy

    % your CODE starts here

      val_mat = zeros(size(Ig)); % also cumulative energy; i dont like 'M'
      pth_mat = zeros(size(Ig));

      % initialize
      pth_mat(1,:) = 0;
      val_mat(1,:) = E(1,:);

      [r, c] = size(Ig);



      for i = 2:r
        % setup up the nbhd_mat for each rows
        % in each col are energies to left, direct above, to right
        % in the respective rows
        nbhd_mat = NaN(3,c);
        nbhd_mat(1,2:end) = val_mat(i-1,1:end-1);
        nbhd_mat(2,1:end) = val_mat(i-1,1:end);
        nbhd_mat(3,1:end-1) = val_mat(i-1,2:end);

        % get the min along with index of each column
        [v, idx] = min(nbhd_mat,[],1);

        % update val_mat of cumulative energies with min energies in v:
        val_mat(i,:) = E(i,:) + v;

        % update pth_mat with element giving min energy
        % note NaN does not figure in minima
        pth_mat(i,:) = idx - 2; % note shift by -2 for -1 0 1 convention in class

      end
      M = val_mat;

      % now get min path
      last_row = val_mat(r,:);
      [~, cmin_idx] = min(last_row);

      for color = 1:3
        % reset column index
        c_idx = cmin_idx;
        % go back upward along rows to delete pixels
        for i = r:-1:1
          v_i = Ic(i,:,color);
          v_i(c_idx) = [];
          Ic(i,1:end-1,color) = v_i;

          % now update c_idx with the column index in row above
          idx_shift = pth_mat(i,c_idx);
          c_idx = c_idx + idx_shift;
        end
      end

      Ic = Ic(:,1:end-1,:);

    % your CODE ends here


end


figure(1),imshow(I);
figure(2),imshow(Ic);


end
```

Code to call your function

↻ Reset

```
N = 20;
Ic = ImageCarving(N);
```

▶ Run Function ?

Previous Assessment: All Tests Passed

Submit ?

✓ **Test 1**