Seam Carving - Project Report
Group 16

Problem Statement:

Digital images are often viewed in many different display devices with a variety of resolutions. Variation of resolution makes viewing images difficult because they usually are resized to accommodate limited space. Simple attempts at resizing include scaling and cropping. Scaling reduces perceivable detail and cropping can't be done automatically. Also, cropping alters the image composition and is not always desirable. This project is an implementation of a different image resizing approach called **seam carving**.

Seam carving allows a change in size of the image by modifying the least noticeable pixels in an image. A typical application for seam carving is to reduce the size of an image along one dimension. This can be done by finding one pixel wide paths from the top to the bottom of the image and removing those paths. If the pixels in those paths are similar to surrounding pixels, then their removal may be unnoticed. Other seam carving applications include increasing the size of an image, changing the size of an image in two dimensions and even object removal.

Fig 1: Resizing 334 X 500 image to 334 X 400
(a)Original Image (b) Seam Carved Image (c) Scaled Image (d) Cropped Image

What is Seam Carving?

Seam Carving is a newly developed technique targeting image compression and resizing based on detection of seams from the energy function of the image. The method aims at finding seams(threads) of minimum energy and manipulating the image using them.

- **What is a seam ?**
  Seams can be either vertical or horizontal. A vertical seam is a path of 8 connected pixels from top to bottom in an image with one pixel in each row. A horizontal seam is similar with the exception of the connection being from left to right. The importance/energy function values a pixel by measuring its contrast with its neighbor pixels.

Fig 2: Seams (Red Colored Threads)

Why Seam Carving ?

<u>Effective resizing of images should not only use geometric constraints, but consider the image content as well</u>. Conventional image resizing consists of cropping or evenly down sampling that lead to loss of important features or distortion. This method enables us to remove pixel from uninteresting parts of the image while preserving important content.

- A seen in Fig 1: We can clearly notice that Seam carving method preserves the interesting features of the image(here Mountain and boat) when resized the image.
- On the contrary scaling introduces geometric distortion.
- And cropping resulted in loss of important features of the image (Right end of the boat).

Seam carving can be used for distortion free image expansion by inserting least energy seams in the image. This can also be extended to object removal and object protection in an image.

Seam Carving! How?

- Generating an Energy Map :
  <u>**Method I (Gradient Magnitude)-**</u>
  There are various methods to extract the unnoticeable pixel from an image. First and most basic method is to assign energy to each pixel by using a gradient operator.(Sobel, Prewiit, Robert or Laplacian) to compute the gradient in both x and Y direction.
  The energy function is defined as follows:

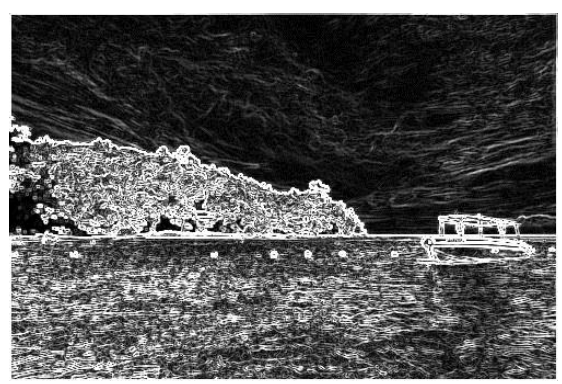


Fig 3: Gradient energy map

$$e(x,y) = \left|\frac{\partial I}{\partial x}\right| + \left|\frac{\partial I}{\partial y}\right|$$

Method II (Entropy)–

The basic gradient function doesn't give defined outputs. A local entropy filter can be applied on the image to improve the basic gradient energy map. We find entropy of 9X9 blocks of image and add it on to the gradient energy map.

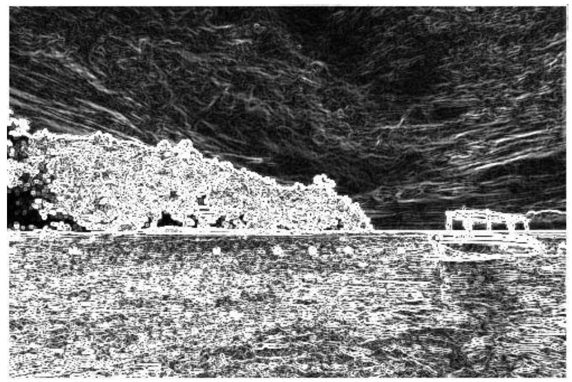


Fig 4: Entropy Map

Method III (HoG)–

Another method to compute the gradient energy map is Histogram of Gradients (Hog) method.

$$e_{HoG}(\mathbf{I}) = \frac{|\frac{\partial}{\partial x}\mathbf{I}| + |\frac{\partial}{\partial y}\mathbf{I}|}{\max\left(HoG(\mathbf{I}(x,y))\right)},$$

Where Hog(I(x, y)) is taken to be a histogram of oriented gradients at every pixel. We use an 8 bin histogram computed over 11X11 window around the pixel.

Fig 5: HoG Map

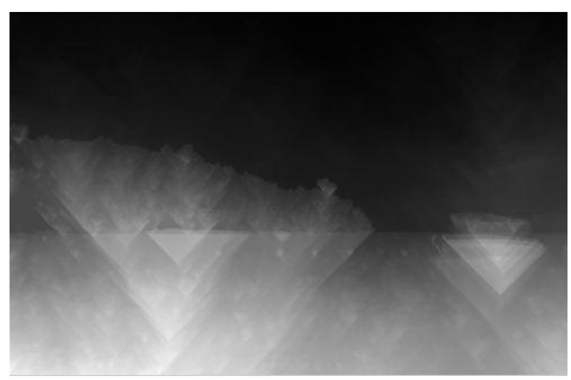


Fig 6: Cumulative Energy Map

- Determining the **Cumulative Energy Map:**
  We now use Dynamic Programming to find the cumulative energy of a path taken from the top (in case of a vertical seam carving) and left(in case of horizontal seam carving) till the pixel(I,j).
  We can define the cumulative energy function as:

  M(I,j) = e(I,j) + min(M(i-1, j-1), M(i-1, j), M(i-1, j+1))

- Detecting the **optimal seam:**

  In order to detect the optimal seam we consider the cumulative energy map such that the Pixel with the lowest value of M(I,j) in the last row (in case of vertical seam carving) or last column (in case of horizontal seam carving) is picked and is backtraced to obtain the optimal seam.

Backtracing in x direction:

$$S^x = \{s_i^x\}_{i=1}^n = \{(x(i), i)\}_{i=1}^n, where\ \forall i, |x(i) - x(i-1)| \le 1$$

Similarly we can extend this to Y direction in case of horizontal seam carving.

Fig 7: (a) One optimal seam detected (b) 20 optimal seams detected in both horizontal and vertical direction

- Manipulating the Optimal Seam:
  We can now use the optimal seam to perform the desired task. For reducing the image size we can remove the optimal seams, to increase the image size we can insert the copy of optimal seams in the image at the appropriate positions.

  In the above method the detection of the Optimal seam is automated. But it can also be controlled by the user by providing certain input which will define a region of interest. In such a case the optimal seam detection would be achieved keeping the interest of user in mind by providing the maximum energy to the selected pixels by the user. And similarly for removal we can give minimum energy to the selected pixels. (Experiments and results are given later in the report.)

Algorithms:

Image size reduction:

- ✓ Generate the energy map using one of the methods.
- ✓ Find the cumulative energy map of the image
- ✓ For each iteration:
  - Detect the optimal seam(in the desired direction-horizontal, vertical)
  - Remove the Seam from the image thereby reducing the size of the image.

Image Size Expansion:

- ✓ Generate the energy map using one of the methods.
- ✓ Find the cumulative energy map of the image
- ✓ For each iteration:
  - Detect the optimal seam(in the desired direction-horizontal, vertical)
  - Copy the optimal seam and extend the image at the point of the optimal seam.

Image size reduction with a protective region provided:

- ✓ Generate the energy map using one of the methods.
- ✓ Find the cumulative energy map of the image
- ✓ Provide the maximum energy values to the M(I,j) if (I,j) is a selected pixel by the user
- ✓ For each iteration:
  - Detect the optimal seam(in the desired direction-horizontal, vertical)
  - Remove the Seam from the image thereby reducing the size of the image.

Object Removal:

- ✓ Generate the energy map using one of the methods.
- ✓ Find the cumulative energy map of the image
- ✓ Provide the minimum energy values to the M(I,j) if (I,j) is a selected pixel by the user
- ✓ For each Iteration:
  - Detect the optimal seam(in the desired direction-horizontal, vertical)
  - Remove the Seam from the image thereby reducing the size of the image.

Experiments, Observations and Results:

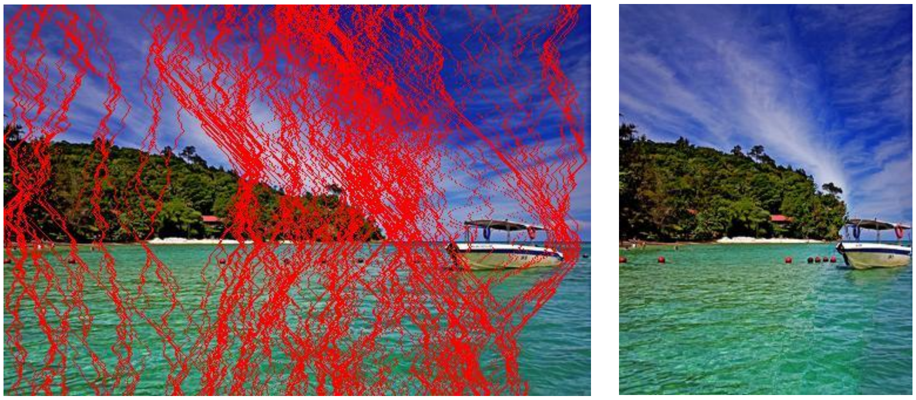
We already showed the difference in scaling, cropping and seam carving on the image Boat.jpg in Fig 1. We will be now experimenting on various images to observe the effects of horizontal seam carving only, vertical seam carving only and horizontal seam carving followed by vertical as well as vertical seam carving followed by horizontal. We will also see the effect of user forced protection on the images. We will then compare the seam removal through various methods in different images. [Experiments on dataset are shown in the end]

Vertical seam carving (Removing 10 seams by Entropy Method):

Fig 8: 10 seams detected and removed

Vertical seam carving(Removing 100 seams by Entropy Method):

Vertical seam carving(removing 250 seams): Fig 10:



Observations:

The above image used in Fig 1 showed comparison of seam carving with other resizing operations. Here we note the difference in seam detection related to the extent of resizing. We can observe the perfect output for reducing 20 image columns. The boat and the mountains are completely preserved. Whereas when we reduce it by 100 columns we see that a few seams pass through the boat, which cannot be observed unless seen the detected seams. And in the last image we can see that the uninterested regions in the mountains are removed.

Comparison Of the Three Methods of Energy Map Generation used, on the final result:

Vertical seam detection using Entropy, Gradient and HoG methods respectively(100 seams):

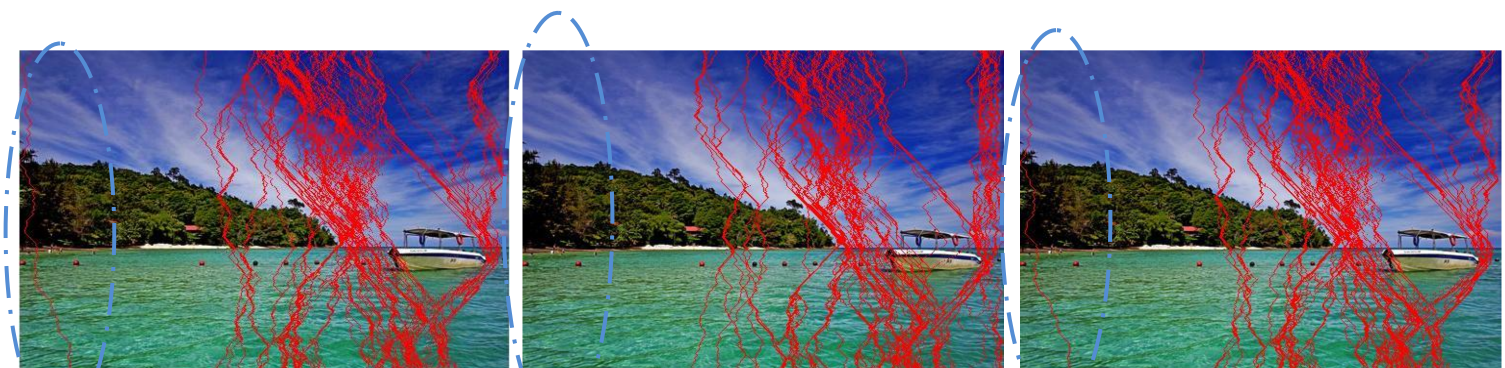
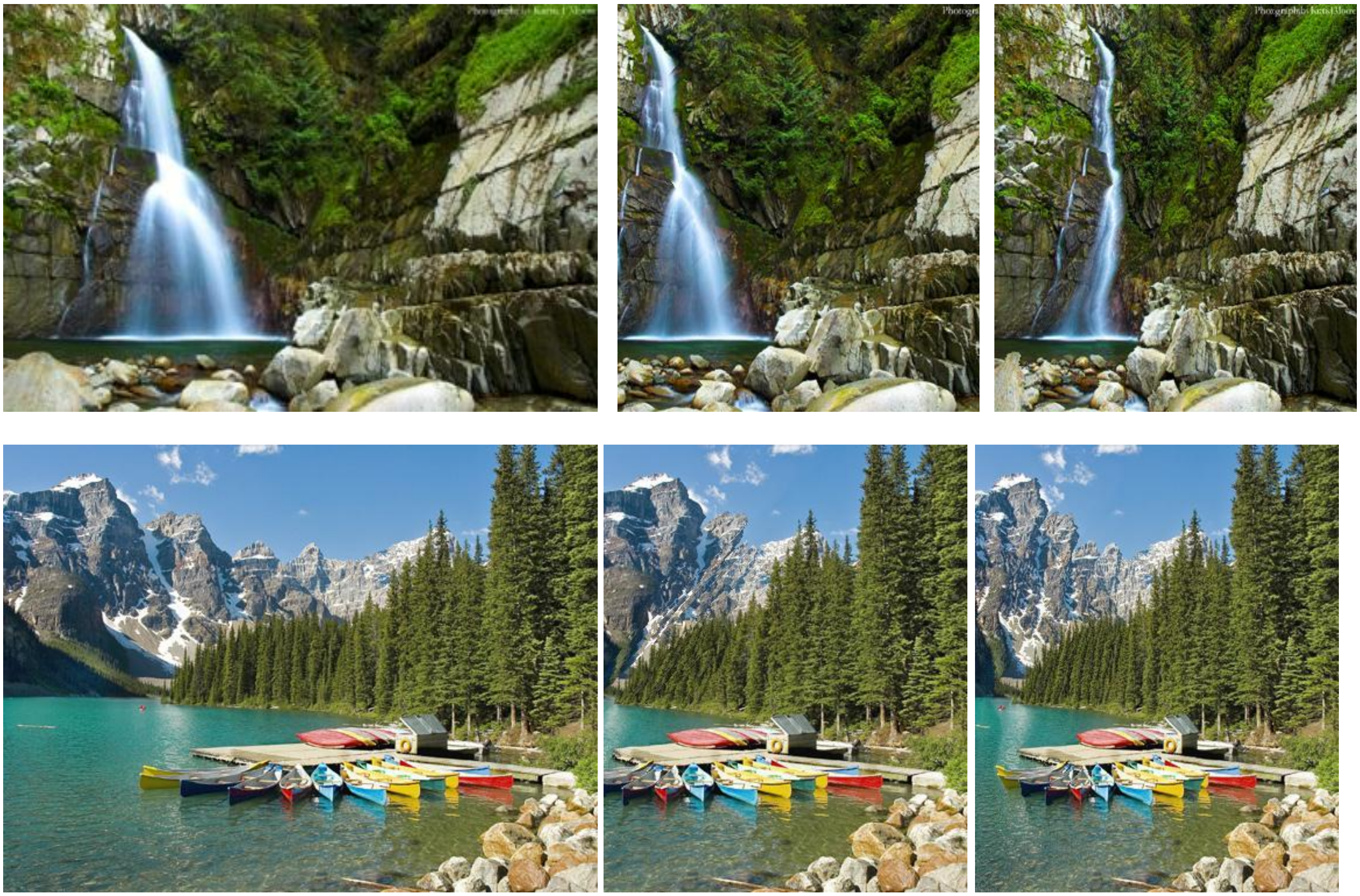


Fig 11: (a) Entropy method (b) Gradient Method (c) HoG Method

Fig 12:

Some more examples to compare the results from different methods of seam carving.

Fig 13: (a),(d),(g) Original Images (b),(e),(h) Entropy Method (c),(f),(i) Image Gradient Method



- **In the first set of images Entropy Method gives better result**

- **Similarly in the 2nd set the Image Gradient Method is better**
- **Whereas in the last set both the methods are equally good.**

Observations:

- We can note that the image seams detected in the Fig 11 are slightly different. In Fig 11(a) and (c) we can see a vertical seam in the left, which is absent in the gradient method (Fig 11(b)).
- There are some other seams too that are totally different in the three images, which can be judged by looking at the number of seams passing through the boat and the mountains.
- In order to find the best method for an image we can count the number of seams that pass through the "regions of interest" in the image.
- For the above image we can see that the number of seams that passing through the "regions of interest" are 20, 15 and 10 approximately. So, we can conclude that for the above image HoG method is the best method.
- Much couldn't be concluded from the lessened images directly as on subjective perception they look almost the same unless examined closely.
- In Fig 13, we can observe that the Entropy method is better for the 1st set, Image gradient method is better for the 2nd set and Both are equally good for the 3rd set of images.

Horizontal Seam Carving:

Similar to above we can implement seam carving in the horizontal direction. Here are the examples:

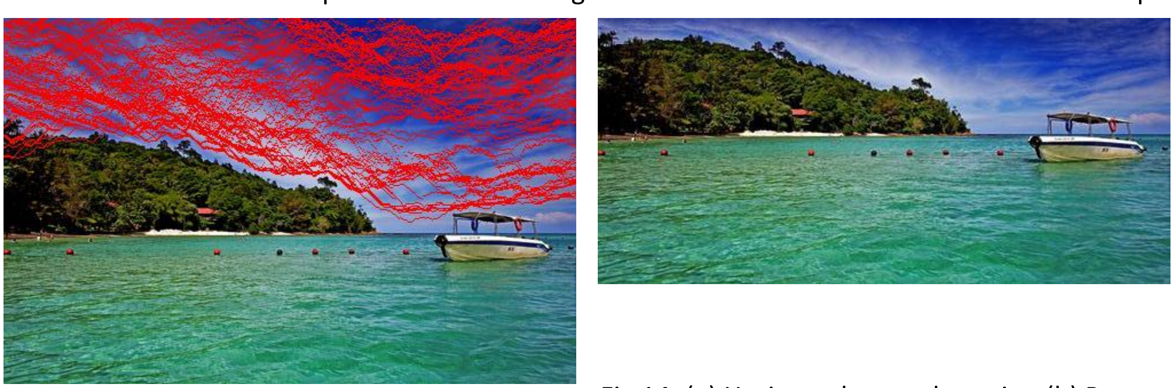


Fig 14: (a) Horizontal seam detection (b) Removal

Comparison Of the Three Methods of Energy Map Generation used, on the final result through horizontal seam carving:

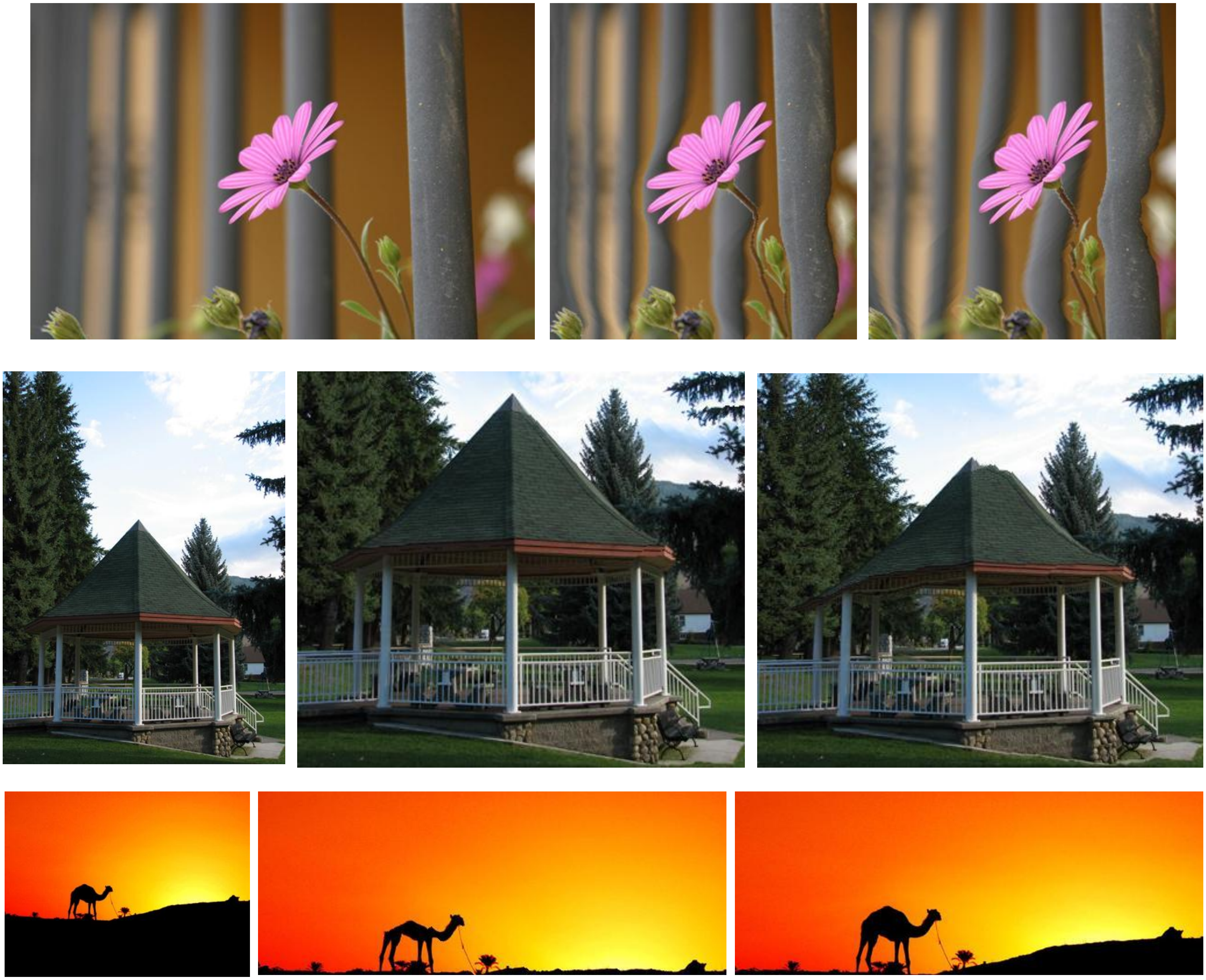


Fig 15: Horizontal seam carving in three sets of images using Entropy and Image gradient method respectively.

- **In the first set of images Entropy Method gives better result**
- **Similarly in the 2nd set the Image Gradient Method is better**
- **Whereas in the last set both the methods are equally good.**

Object Protection:

We can note in few images the “regions of interest” gets distorted and hence the function of Content Aware Image Resizing fails. In such case we can select the regions of interest in an image and then perform seam carving such that the selected region is always protected i.e. given the highest preference.

Example Image where Auto Seam carving Fails:

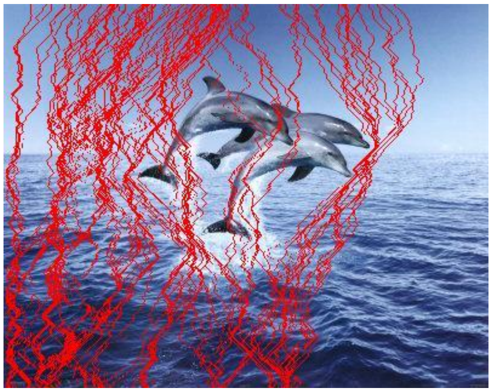
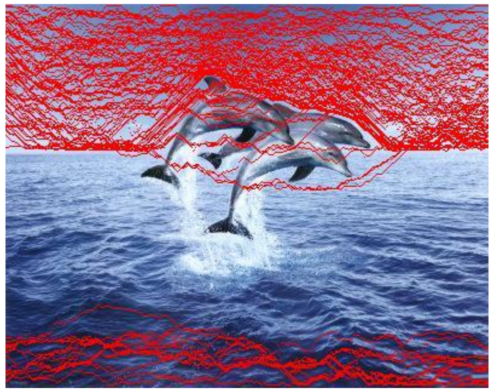



Fig 15: (a) Vertical Seams(100) (b) Horizontal Seams(100)

Observations:

We can see that the interesting region (Dolphins) of the image will get distorted when removed the selected seams. So in order to protect the dolphins we can manually select the dolphin region and then protect the dolphins.

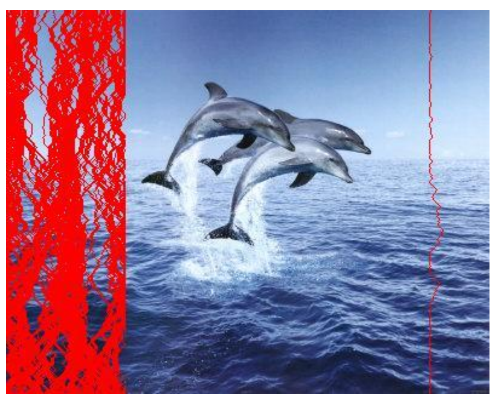



Fig 16: (a) Horizontal Seam carving after applying protection to the dolphins(b) Vertical seam carving after applying protection

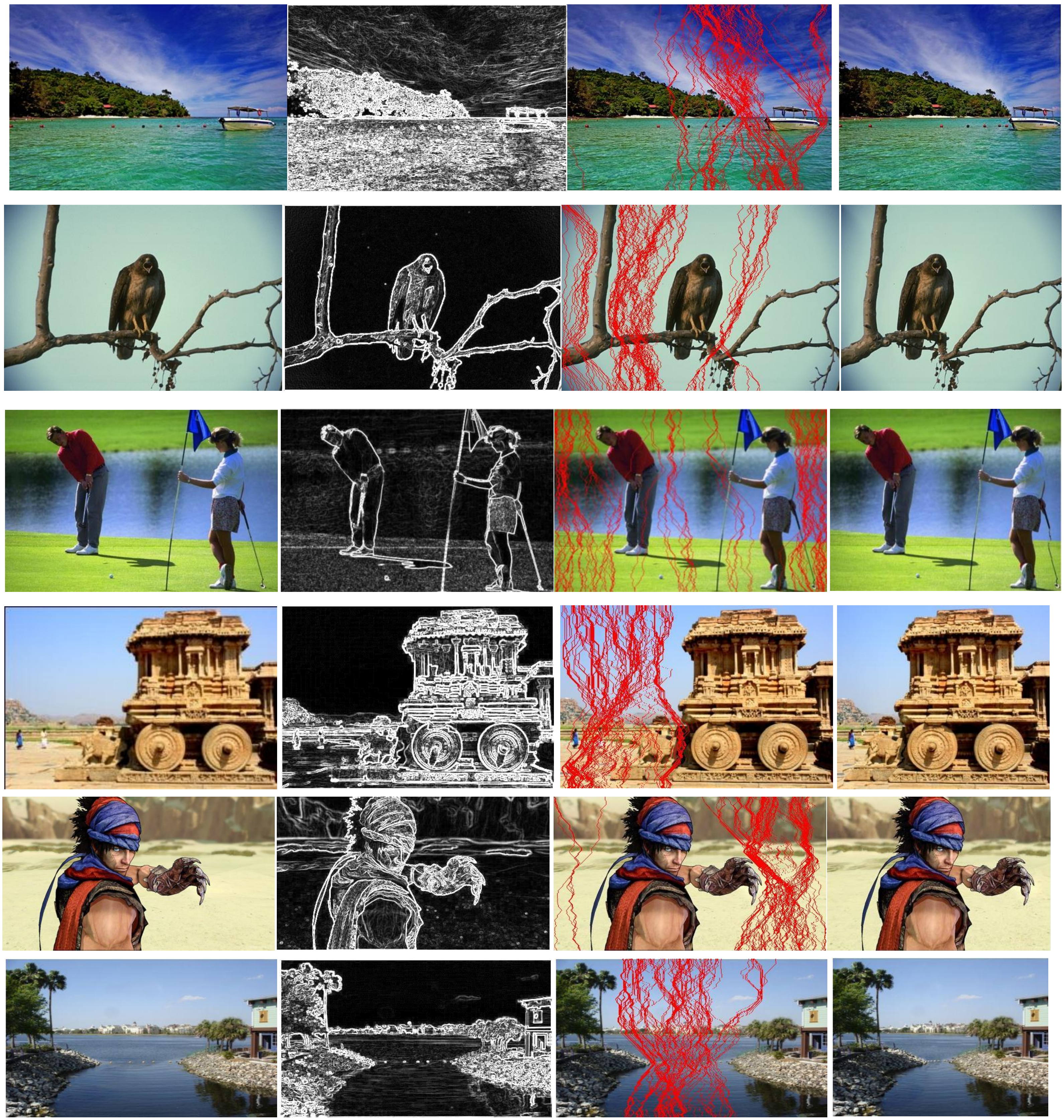
Observations:

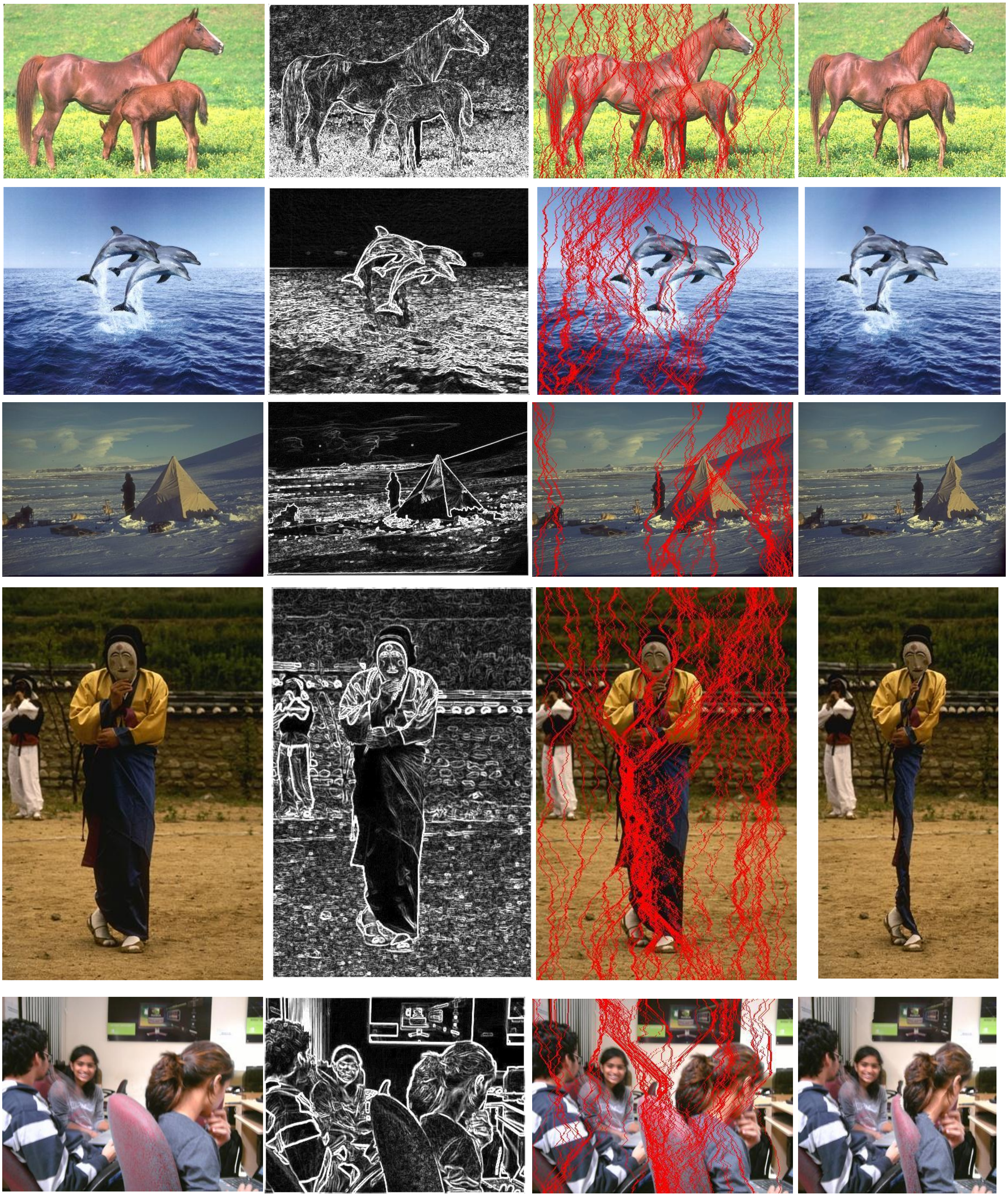
- We can see that the dolphins are totally protected and the seams now pass through the surrounding regions.
- We can see almost a straight line seam at the boundary of the start of the protection layer. This can be explained as the protection layer has the highest energy which earlier has neighbors of less energy.

Similarly, **Object removal** can be implemented which will be included in our next report.

Experiments On different Images (Implementing Vertical Seam Carving)

Original Image Entropy Map Vertical Seams Final





Following are a set of images on which protection and removal has been performed. The areas marked in GREEN have been selected for protection while those marked in RED have been marked for removal.





In this image we can see that traditional method of seam carving distorted the image completely. After applying a protective layer to the image we can see that region was completely protected and better seam carving result was achieved. If we can obtain such a region through any of the image processing algorithms automatically we can directly protect such regions. (Segmentation methods.)

In this set of Images we have removed the man in the background first by keeping the image size same by inserting seams and in the second image the man was removed reducing the image size. In the last image we can see that after the removal of the man, when applied seam carving the body of the mime gets slightly distorted as some seams still pass through the mime. To improve that:

In this set of image we are performing the protection and removal at the same time. Here we have protected the mime and removed the man as well. Best Case.

On the above set of images we can see that an object removed by seam carving which are overlapping will distort the other. This could be a valid area of reseaerch in inpainting.

Here the horses are protected.

Man is removed simultaneousl y protecting the tent.

Some more examples:

In this case we can see that he object was directly removed, the problem in this case is we can note the removal(edges in the second image). We can conclude that a very large object when removed form the image , all the seams pass through that region its almost equivalent to cropping, which results in visual artifacts.

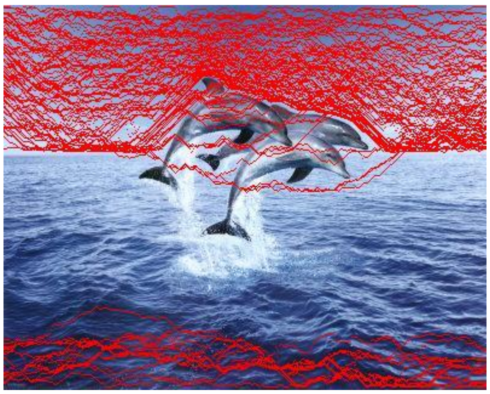
As here we have manually protected the regions of interest. We can improve seam carving algorithm by modifying the energy function used to incorporate methods such as Line detection, segmentation, saliency detection etc. This will automatically take care of the protection regions. The goal will be to obtain the region of interests.

Improvements Methods Applied:

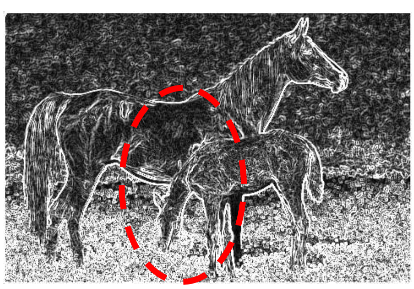
Need For Improvements:

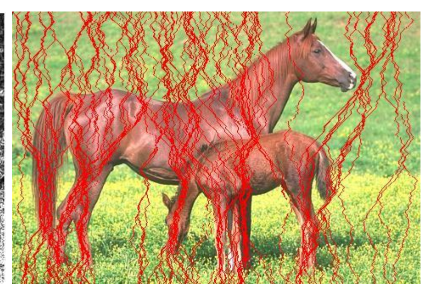
As mentioned earlier that traditional methods of seam carving works only for a very small dataset. It works perfectly only for images in which large portions are occupied by the background without much change in the nature. With uniformity in the background the energy of the region decreases and the less interesting seams are removed. But if the background is smaller and has lots of edges within, In such a case the traditional method of seam carving fails. It also fails if the object of interest has large hollow regions In it as it decreases the overall energy and the region is eliminated from the image. In order to rectify these failures we need improvements in the traditional methods of Seam Carving.

Cases Of Failures Of Traditional Seam Carving:



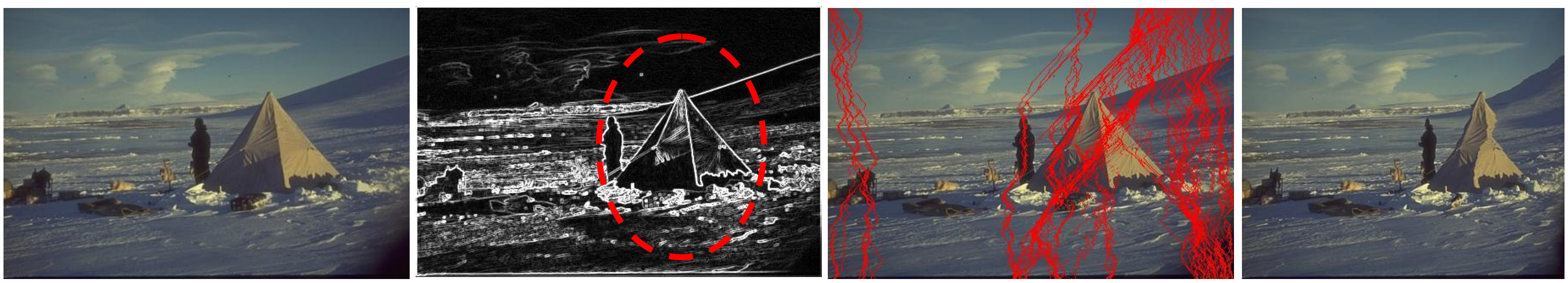
We can see in the above image the water below is of less interest than the dolphins and we would like to remove the water below the dolphins before removing any part of the dolphin. But based on Energy of the image the dolphins have lesser energy because of the uniformity in their skin color. **The traditional method only encounters the edges and totally ignores the color composition of the image.**





In the above images also it can be observed that the horse head is distorted because of the color uniformity which decreases the energy of the head region of the horse. It can also be seen from the energy image (2nd from left) that most of the region is black near the head(marked with red circle).

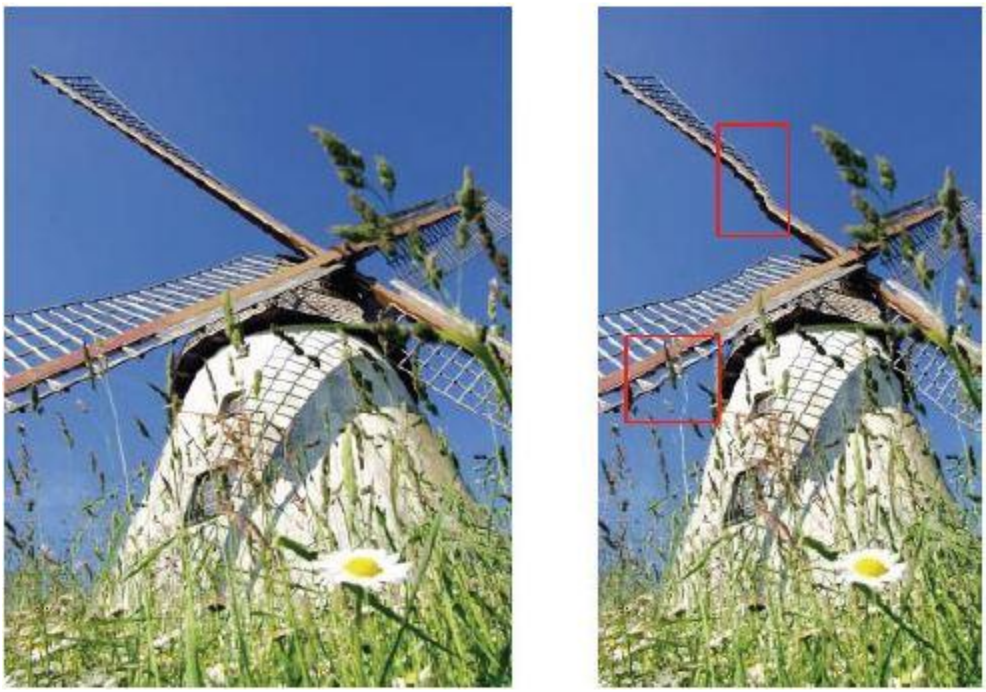
Similar observation and analysis can be done for the above image too. The region near the legs of the man has blank black space and hence least energy. So most of the seams pass through this region only.



In the above image too the tent is completely distorted. And it was expected looking at the energy map of the image that most of the seams would pass through it only.

In the above image too we can note distortions in the marked regions. These distortions occur due to the seam removal from straight line regions. And removing too many seams doesn't leave it a straight line anymore and hence the distortion is clearly visible.

In the above image the region of the pillar can be seen as distorted.



These examples clearly show the need of improvements in the Traditional Seam Carving Algorithm.

General Idea Behind Improving Seam Carving:

The general idea is to combine the traditional method of seam carving with different algorithms. We can do some pre-processing on the image before applying seam carving or we can use different **decision making** algorithms which will decide the energy mapping of the image after or during removal of seams(Line detection algorithm combined with seam carving).

The problems seen can be categorized mainly into 2 categories:

- **Problem1:** The traditional method discards all the color information of the image(uses the gray scale image) and only focuses on the edges and uses them to determine the energy map.

- **Problem2:** The distortions in the Images are mainly visible because of the straight lines in the image being smashed. A regular line when deformed gives the impression of distortion. So protecting the straight lines in the images will also help restricting visual distortions in the image.
- **Problem3:** No clear distinction between the background and foreground. That results in some part of the fore ground to be removed.

Approach to Solving the above problems:

Visual Saliency(Problem 1): A method that can be used to solve problem 1 is combining the seam carving method with saliency detection. As the problem is that Normal seam carving ignores the color content. We can use Visual Saliency detection to take the color content into consideration and then apply seam carving on the new energy map created. We will see this method in detail.

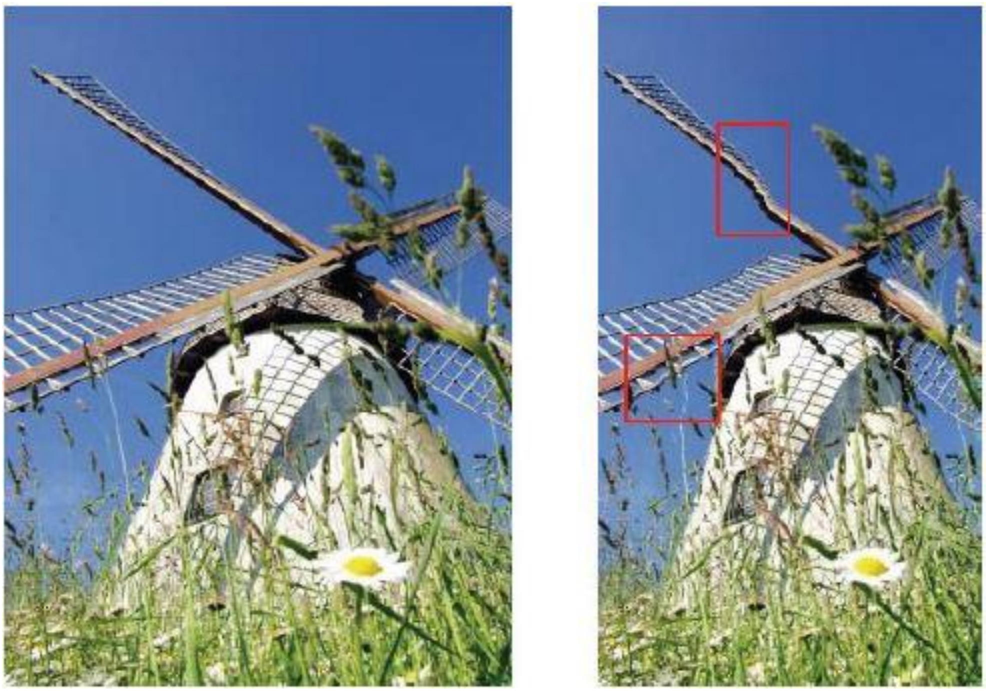
Line Detection(Problem2): A method that can be used to solve problem 2 is combining the seam carving method with Line detection Algorithm. We can detect the straight lines in an image and make sure that too many seams doesn't intersect the line nearby, such that the line doesn't get deformed. One way of doing this is whenever a seam passes through a line, we can assure that no other seams passes that line in the nearby area of some radius.

Image Segmentation(Problem3): A method to solve problem 3 is combining image segmentation algorithms to segment out different regions and increase the energy of the interested regions to protect them. This will result in the detection of the foreground and only the background will be reduced using seam carving. Moreover the regions that are marked similar will rarely have too many edges or boundaries.

Now let's look into each of the methods in detail.

Improved Seam Carving With Line Detection:

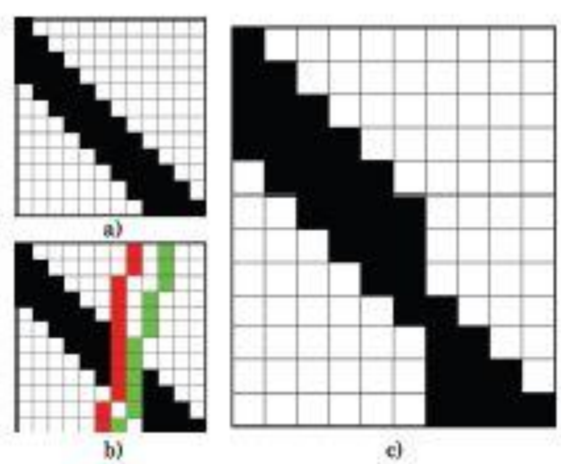
Errors caused in the seam carving are especially disturbing when the original image includes regular patterns or straight lines. These lines may become curved or disconnected in the final image. Example is shown below:



The curved lines on the windmill fan are clearly visible. These distortions may occur if a seam crosses a non-vertical or non-horizontal line.

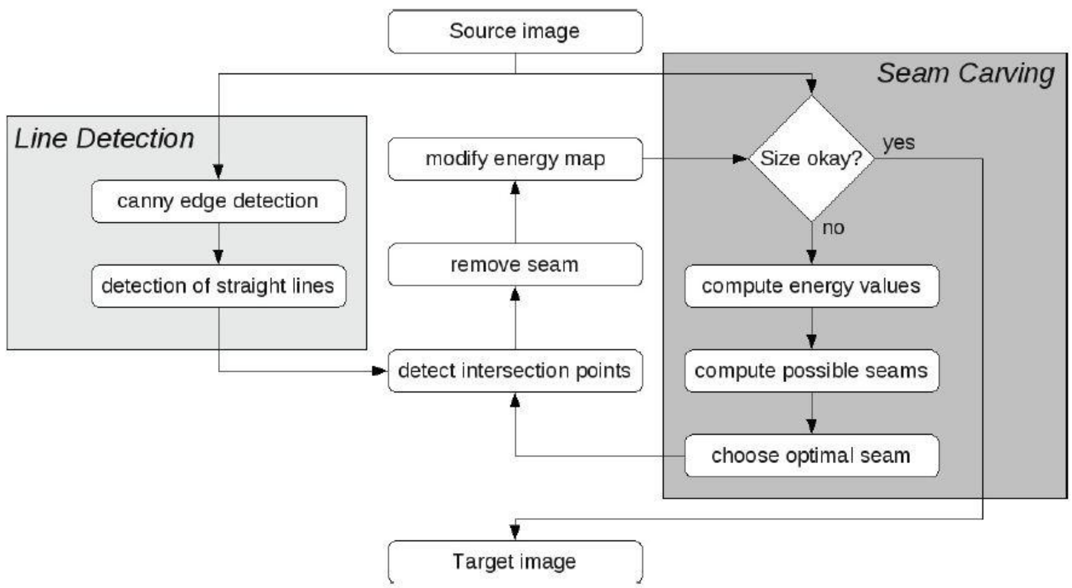
<u>Cause of Distortion:</u>

In the image on the right hand side we can see how a straight line can get deformed when too many adjacent or close by seams are removed from the image. If we are able to detect a line and preserve this then we can reduce significant amount of distortions in an image. As mostly the visible distortions occur due to the deformation in the regular patterns.



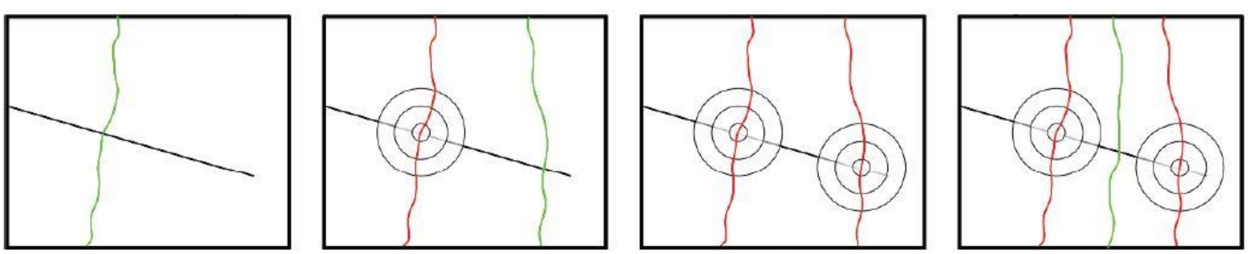
__The Step By Step Approach:__

- **Line Detection**
    - **Detect edges(canny, sobel, prewitt, laplacian, etc)**
    - **Transform into Hough Space**
    - **Find Hough Peaks**
    - **Detect Hough Lines with required parameters**
- **Chose the lines that you want to keep**
- **Find the point of intersection of the seam and the lines selected among the detected lines.**
- **Increase the energy of the region surrounding the points(say upto a distance 'r') of the point of intersection by some amount.**



The flowchart of the process.

After the lines are detected we can modify the energy map accordingly. This can be better explained by the diagram :

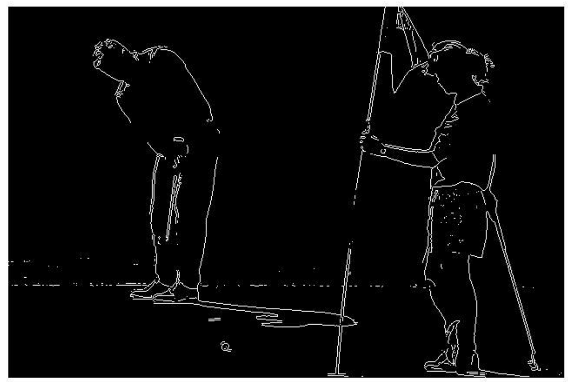


Once a seam intersects the line, we increase the energy of the surrounding seams at some radius ‘r’, so that the next seam is most likely not to pass or intersect the line close to the 1st seam and hence we won’t see noticeable distortions. The same method can be repeated after every seam removal.

Line Detection:

In order to detect lines we need to detect the edges in the image first. This is the most important part of line detection as based on the edges detected the lines will be marked. We can use several methods of edge detection like Sobel, Prewitt, Laplacian, Canny etc. In general for line detections Canny Edge detection method is used as it give refined and sharp edges too. But in our case we will observe that other methods are also preferable a few times.

Let’s take an sample image to explain the line detection algorithm:



The images here are generated through:

1 – sobel edge detection

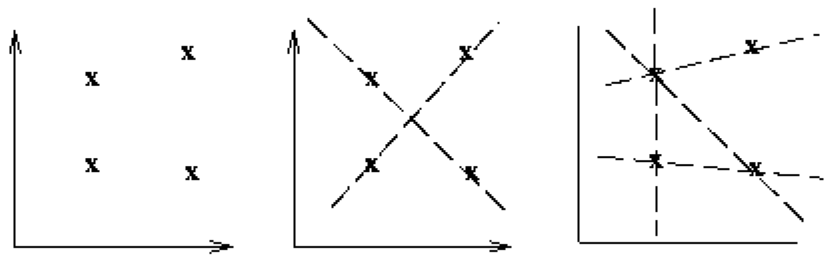
2 – Canny edge detection

As it can be clearly seen that canny edge detection gives large number of edges as compared to sobel edge detection.

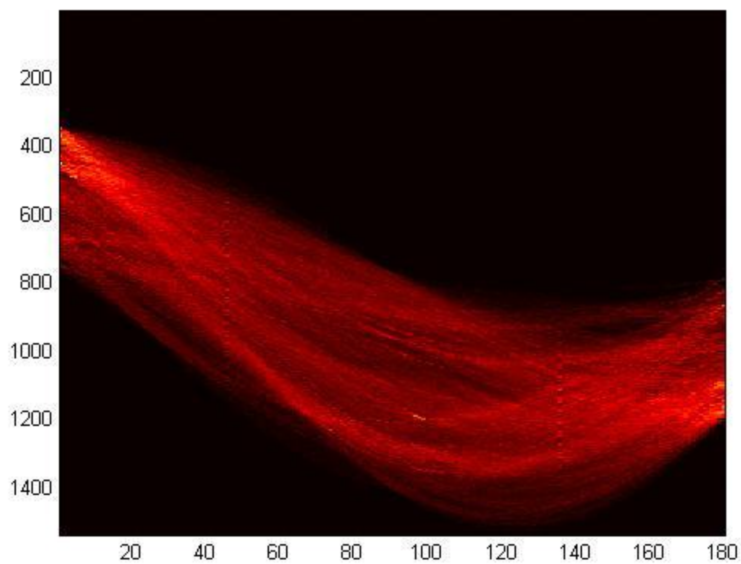
Next we need to transform the obtained edges into Hough Space, which detects the lines in the following manner:

The Hough technique is particularly useful for computing a global description of a feature(s) (where the number of solution classes need not be known *a priori*), given (possibly noisy) local measurements. The motivating idea behind the Hough technique for line detection is that each input measurement (*e.g.* coordinate point) indicates its contribution to a globally consistent solution (*e.g.* the physical line which gave rise to that image point).

As a simple example, consider the common problem of fitting a set of line segments to a set of discrete image points (*e.g.* pixel locations output from an edge detector). Figure 1 shows some possible solutions to this problem. Here the lack of *a priori* knowledge about the number of desired line segments (and the ambiguity about what constitutes a line segment) render this problem under-constrained.



The above image gives the hough curve as:



The intersections of the curves at the points represents the lines. Each line in the curve is for a pixel and if large number of pixels match it represents a line. Hence the intersection points here represents lines.

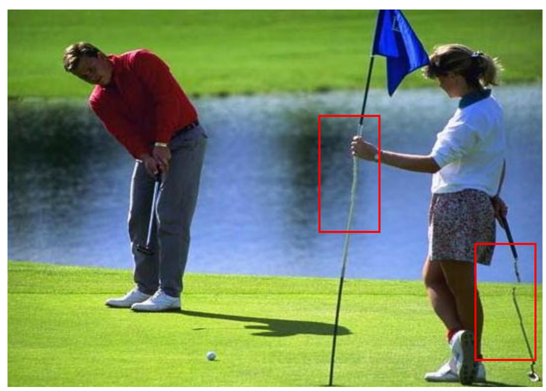
Now we can define the hough peaks and determine the lines with parameters as in Min Length and Fill Gap. Which determines what amount of gaps to fill while determining lines and what is the minimum length of lines that should be taken into consideration.

<u>Line Detection With Number of Peaks 20 and threshold 0 for both canny and sobel:</u>

In the above images , we have taken the fill gap and min length parameter to be default values of 7 and 20 respectively. The number of peaks determines the number of lines we want to detect. In this case we get 61 lines for Canny edge detection and 9 lines for sobel edge detection using 20 peaks.

As we can see that our purpose will be well filled by using the sobel edges for line detection in this case as the straight lines of the objects of importance are highlighted mostly where as in Canny edge detection method we can note that the lines in the grasses are also detected which are of little or no importance to us in this case.

Now an over all comparison of the traditional method and our method over this image:



As it can be seen that the distortions in the previous image are completely removed. As the distorted regions were visible due to the lines present which are now protected by giving them high energy values.

We can change several parameters for line detection and then apply it on the image to detect different types and numbers of lines. Once they are detected the seam carving can be applied.

Here is the comparison of line detection on the above image itself after changing few parameters:

The above images are obtained changing the number of peaks for each of the images in both sobel and canny. Starting form left the values are 5, 50 and 100 respectively sobel followed by canny.

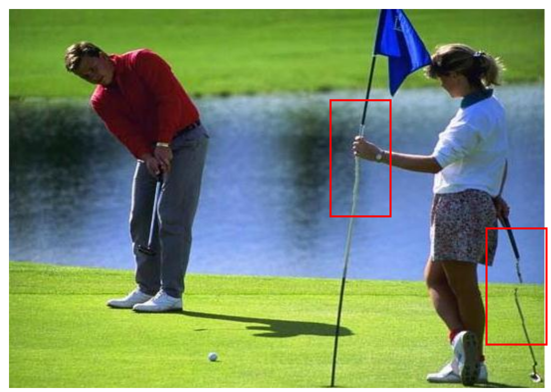
It can be observed that 100 threshold is of no use as it detects very minute lines and almost the entire Image including the non-interesting parts are also included. So according to the image we can define the line size and peak size to use this algorithm in the best possible use.

We have detected the lines in the image using appropriate parameters such that we get required useful lines only and not too many or too less... (parameters used: bin-size = 1, peaks = 10, threshold = 10, fillgap = default, length = default).

With the seams detected here we can see that too many seams do not pass through the straight lines detected in a small range(which is the main reason for visible distortion in the image).

(parameters: radius = 10, increment value = 300..)here radius tells us the area around which we have to increase the energy value to avoid seams passing through the reason and increment value provides the value to be incremented.

Lines with seams.

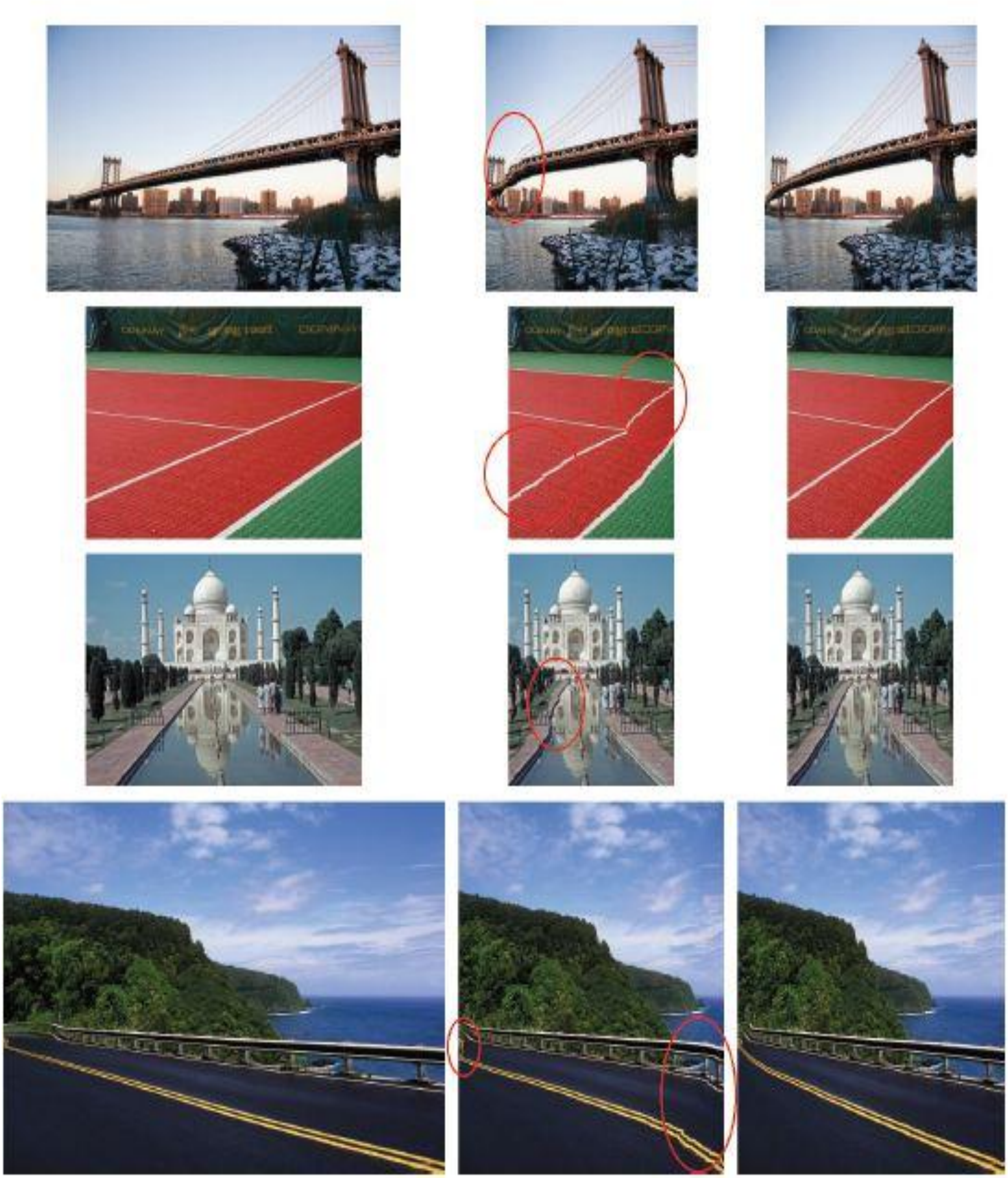


In this image we have removed the seams and on comparison with the previous image we can see that the distortion in the golf stick on the right is removed.

We can still see some distortion. So lets increase the radius.

Here with an increased radius, we significantly reduce the distortion.

More examples in which this method works better are:

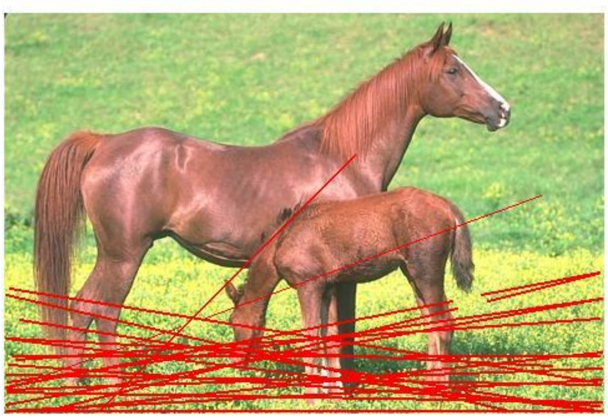


In all the above images we can see the distortions are removed by using this line detection algorithm along with the traditional seam carving method based on entropy. We have experimented with different parameters and had put only the optimal parameters output in this section.

There are certain failures of this approach too. In case there are various lines of almost the same length and peaks but are spread out in the image, then this method would fail. It will be bound to remove seams form continuous ranges. One such example is:

In this case we can see the distorted regions in the straight line. Hence this method fails for this image.

Another case where this method wouldn't work will be the images that doesn't have any significant straight lines.



Here the image will have no effect as we can see the lines detected are out of our region of interest.

The above image can be pre processed using saliency detection method followed by seam carving to protect the horse's head.

Visual Saliency

Normal gradient method just defines energy at edges. It does not include information regarding color of the image and causes unwanted distortions as well as loss of useful data.

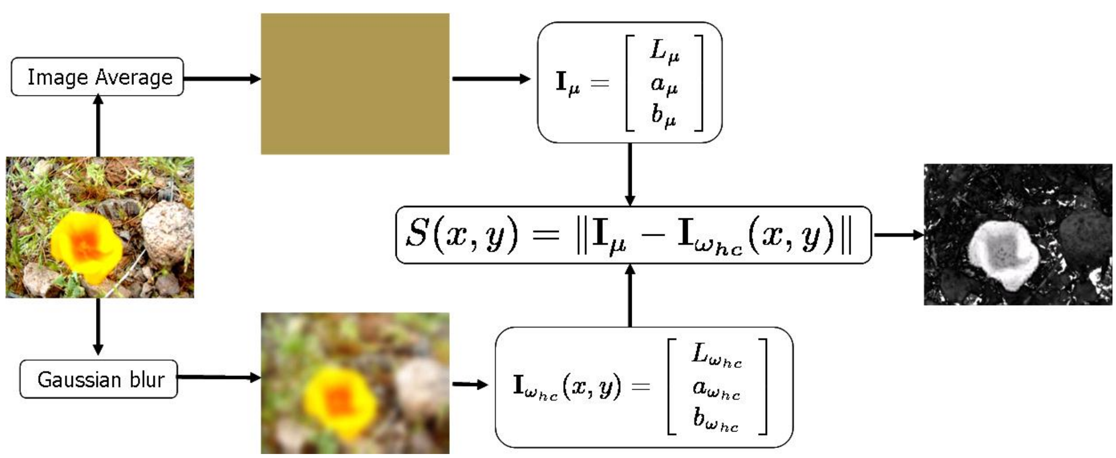
Visual saliency method takes into account the color information of the image. **Salient regions and objects stand-out with respect to their neighborhood.** The **goal** is to **compute the degree of standing out or saliency of each pixel** with respect to its neighbourhood **in terms of its color and lightness** properties.

Saliency detection methods take a centre-versus-surround approach.

One of the key decisions to make is the size of the neighborhood used for computing saliency. We use the entire image as the neighbourhood **(global saliency)** as well as smaller block sizes and combination of them (**local saliency)**. This allows us to exploit more spatial frequencies resulting in uniformly highlighted salient regions with well-defined borders.

The Algorithm

- We perform 3x3 or 5x5 Gaussian smoothing on the image.
- We take the color Lab transform of the image or a block of the image.
- In the lab color transform we define every pixel as vector with coordinates as L*, a* and b*.
- We find the centroid of the image or block in the L*, a*, b* coordinate system.
- For every pixel we find the distance from the centroid.
- We can have multiple distance matrix pertaining to different block sizes used.
- We do compression of range on lower value and expansion for higher value for all distance matrices except the global distance matrix.
- Summations of all the distance matrices give the new energy map.



Global saliency algorithm

Assumptions and reasons

- We have assumed that the image has a clearly defined foreground and background.
- The image has a global mean color pertaining to the background.
- The gausssian blurring is done to discard the high frequency components as our eyes are more tuned to low frequencies.
- The lab color space is comparatively more perceptually uniform than others and hence more effectively approximates human vision.
  Perceptually uniform means that a change of the same amount in a color value should produce a change of about the same visual importance.
- Global saliency gives us an energy map with a bright (high energy) foreground against a darker(low energy) background.
- Not always the entire background is uniformly low in energy and seams might interfere with the foreground. To avoid that we use local saliency. It tones the background further and also brings out details of edges and small changes in the foreground.
  (We came up with and designed the novel concept of improving the energy map attained from global saliency with local saliency)

Experiments and Observations:

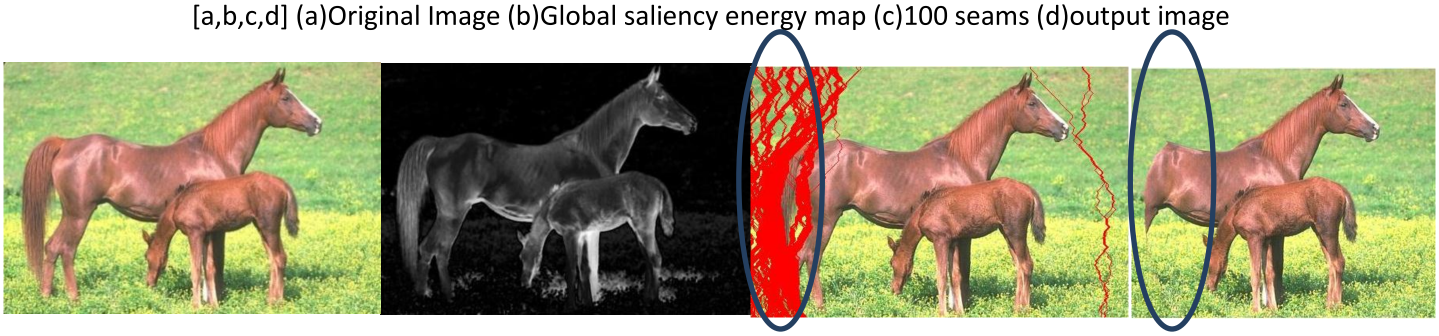
Results attained from using **only** global saliency.

[a,b,c,d] (a)Original Image (b)Global saliency energy map (c)100 seams (d)output image

- As we can see that the mage has a clearly defined foreground (the bear with fish) against a contrasting background (water) the energy map and output are up to mark.

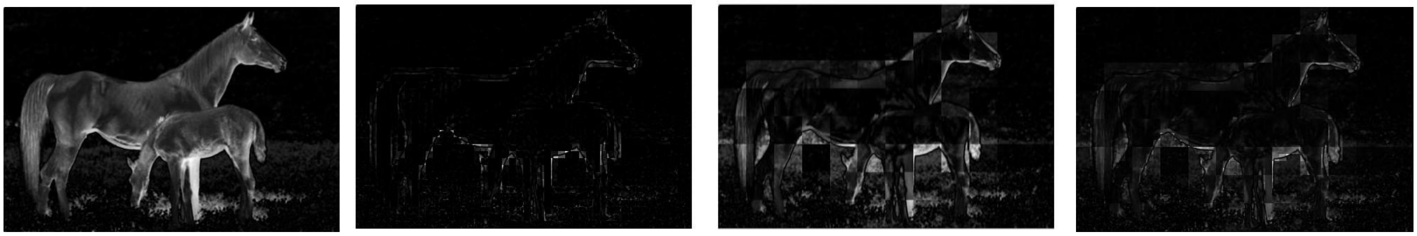
The method fails certain times when the foreground has color component close or similar to the background. Or when the background is very textured.

[a,b,c,d] (a)Original Image (b)Global saliency energy map (c)100 seams (d)output image



- We can see in the picture a part of the horse gets cut off. This unwanted.
- The global saliency method does not take into account local changes.

We can see that the yellow grass is very bright even though it’s part of the background. Using local saliency with samller block sizes we can undo that as then the mean color of the grass region will be closer to yellow.



[a,b,c,d](a)global saliency (b)local saliency block size 10x10 (c)50x50 (d)final E map

[a,b,c](a) Original Image 100 seams less: (b)Global Saliency (c) local + global saliency

<u>Improvement Of saliency method:</u>

1. **<u>Addition of Saliency Energy and Gradient energy</u>**

   We calculate the gradient energy map and rescale the range to match that of Saliency map.

   We added the two.

   <u>Reason –</u> Saliency energy maps used till now have not being taking in to account edge information of the image but only the color information. Edge information is important and we want to avoid distortion of edges.

a, b, c, d,

e,f

(a)Original Image (b)Gradient energy map (c) local saliency block size 40x40 (d) global + local saliency energy map (e) 100 seam carved image using ‘b’ (f) 100 seam carved image using ‘d’.

(We came up with and designed the concept of improving the energy map attained from addition of saliency and gradient edge detection.)

2. **Diffusion of energy of removed seams (Importance Diffusion)**

When a seam is removed from an image it creates new artefacts in the image. However, **repeated removal** of unimportant rows and columns often produces **unwanted artefacts**, since excessive removal of unimportant lines results in **discontinuous blocks of important parts**. This problem is caused by no preserving the context of removed parts during row/column removals.

To **resolve** this problem, the **neighbours of removed parts should evidently carry the information about the re-moved parts** as well as their own pixel information. In other words, the importance of the neighbours should increase reflecting the importance of the removed parts.

Algorithm used:

$vr(y') \leftarrow vr(y0) + w(y; y')*p(vr(y));$

$vc(x') \leftarrow vc(x0) + w(x; x')*p(vc(x));$

Here, vr(.)/vc(.) is importance (energy value) of ‘.’ for vertical or horizontal seams respectively.

w(a,b) is importance diffusion constant for diffusion of energy of ‘a’ over ‘b’.

$x' = (x+1)$ or $(x-1)$ and $y' = (y+1)$ or $(y-1)$

We have considered w(a,b) = 0.2 for all a,b.

[a,b,c] (a) Original Image (b)seam carved image w/o ID (c)Result with ID

- Without importance diffusion most of the seams tend to pass from from the same area of low energy, creating distortion.
- In the second image the distortion is very visible.
- Importance diffusion doesnot excessive number of seams to pass through the same area.
- The effect is visible in the 3rs image. The mountain slope is very smooth as compared to the first one.
- The result of seamcarving without importance diffusion shows distorted fea-tures. Because of the amount of information in the original image, unimportant regions are removed too much causing the distortion.
- However, importance diffusion avoids excessive removal of unimportant regions, so the result of seam carving with importance diffusion shows less artifacts.

3. **Colour component in forward energy terms**

For a vertical seam removal, the dynamic programming memorization table entry M(x; y) is given as:

$$M(x,y) = E_g(x,y) + min \begin{cases} M(x-1,y-1) \\ M(x,y-1) \\ M(x+1,y-1) \end{cases}$$

A removed seam may actually introduce more energy than it takes away because of previously non-adjacent pixels becoming neighbours. They therefore introduce forward energy criteria, such that the optimal seam found is one whose removal re-introduces minimum amount of energy.

$$M(x,y) = E_g(x,y) + min \begin{cases} C_L(x,y) + M(x-1,y-1) \\ C_U(x,y) + M(x,y-1) \\ C_R(x,y) + M(x+1,y-1) \end{cases}$$

We introduce the use of color information in the forward energy terms by using the corresponding vector distances in Lab space to obtain:

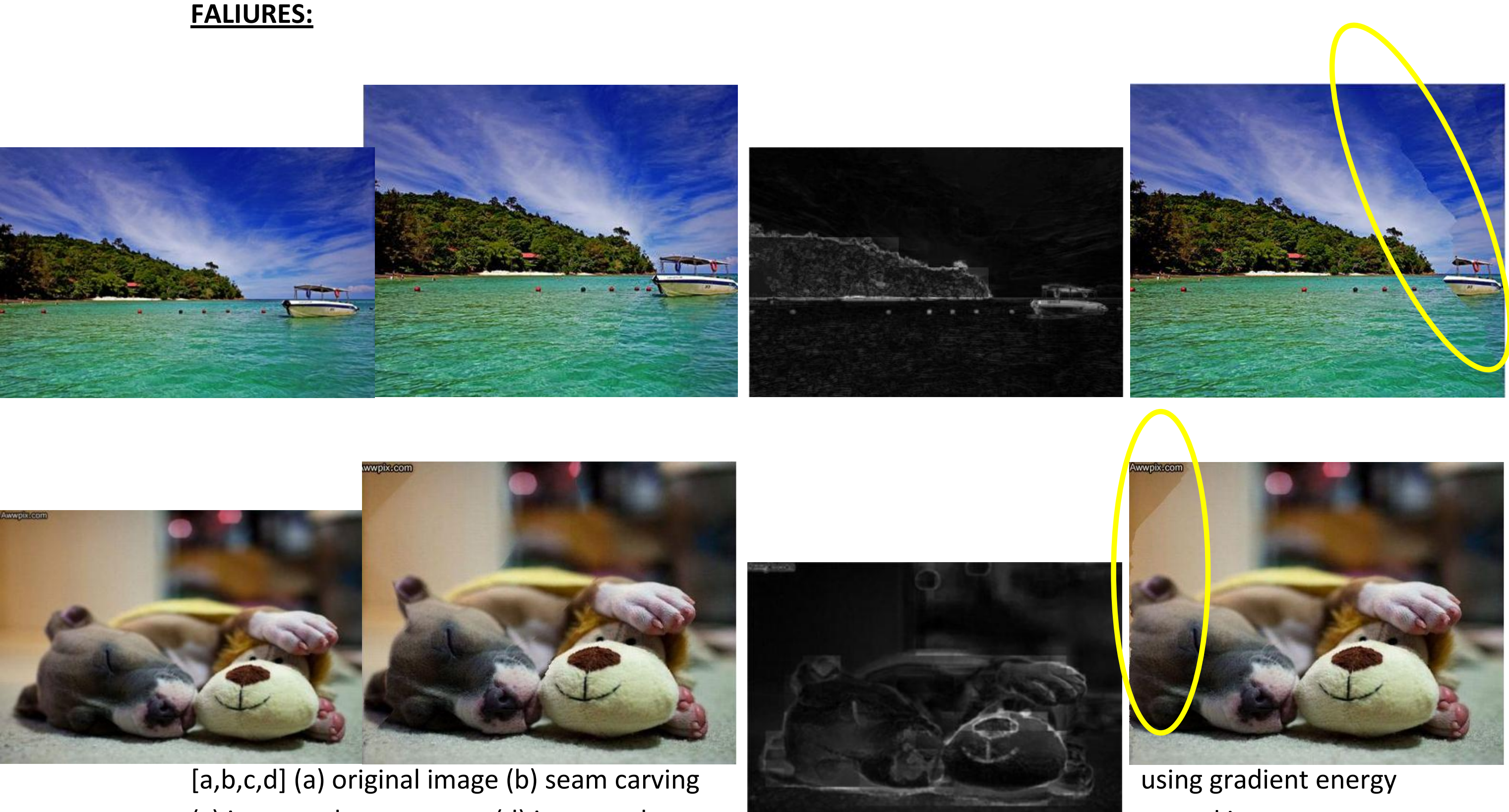
$$C_U(x,y) = \|\mathbf{I}(x+1,y) - \mathbf{I}(x-1,y)\|$$
$$C_L(x,y) = \|\mathbf{I}(x,y-1) - \mathbf{I}(x-1,y)\| + C_U(x,y)$$
$$C_R(x,y) = \|\mathbf{I}(x,y-1) - \mathbf{I}(x+1,y)\| + C_U(x,y)$$

Some results of improved seam carving (using the concepts of global and local saliency, gradient energy and importance diffusion):

(Results generated do not include importance diffusion as the algorithm is very time demanding)

[a,b,c,d] (a) original image (b) seam carving using gradient energy (c) improved energy map (d) improved seam carved image

FALIURES:



[a,b,c,d] (a) original image (b) seam carving using gradient energy
(c) improved energy map (d) improved seam carved image

- Improvements seem to break in certain images where simple gradient method was working fine.
- In the boat image is heavily distorted. The image includes two object of which one is very small compared to another. Also the saliency method tends to put in energy where there was no energy before as there was no defined edge (clouds).
- In the second image due to presence of text in the corner the seams can't go through there. Also the ears of the puppy have been cut off. This is because of the contrasting object (blurred cupboard in the back). As the cupboard is heavily smoothened out it doesn't tend to high energy in gradient method (no defined edge) but in the saliency method due to being of contrasting colour the energy of the region is high and seam don't pass from there.
- In the above two image the discontinuity can be removed by using importance diffusion method.
- For the second image there is no solution to save the ears of the puppy.
- No algorithm is fully complete and universal. Output varies from image to image.

BUT, on an average we can say the proposed algorithm is efficient and gives good results and accounts for almost all variations that can be present in an image.

Dynamic Programming is one way to perform seam carving on images. Another method which can be used for this purpose is Graph Cuts.

General Idea Behind Graph Cuts:

In graph theory, a cut is a partition of the vertices of a graph into two disjoint subsets. The cut-set of the cut is the set of edges whose endpoints are in different subsets of the partition. Edges are said to be crossing the cut if they are in its cut-set.

In a flow network, an s-t cut is a cut that requires the source and the sink to be in different subsets, and its cut-set only consists of edges going from the source's side to the sink's side (Figure 1). The capacity of an s-t cut is defined by the sum of capacity of each edge in the cut-set.

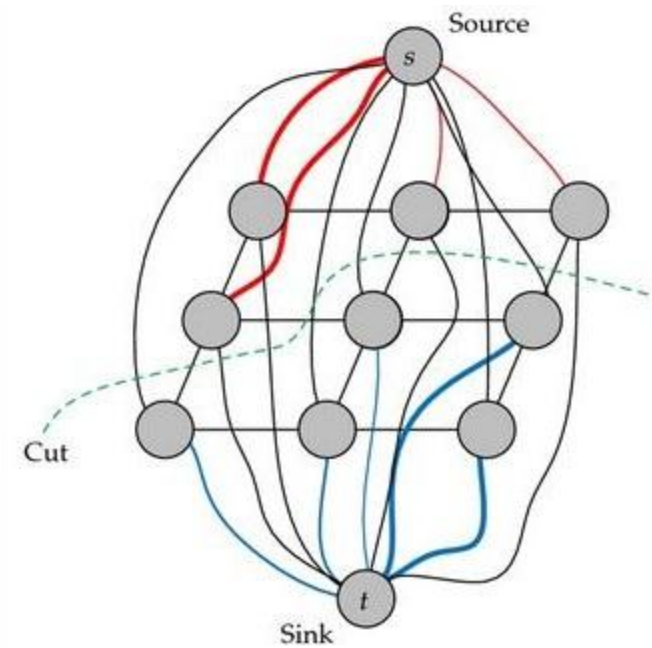


Figure 1

A cut C= (S, T) is a partition V of a graph G = (V, E).

An s-t cut C= (S, T) of a network N = (V, E) is a cut of N such that s ϵ S and t ϵ T, where s and t are the source and the sink of N respectively.
The cut-set of a cut C= (S, T) is the set {(u, v) ϵ E | u ϵ S, v ϵ T}.

The size of a cut C= (S, T) is the number of edges in the cut-set. If the edges are weighted, the value of the cut is the sum of the weights.

Graph cut method implemented in seam carving:

We formulate our image in a Graph representation. We refer to graph edges as arcs to distinguish them from edges in the image. We construct a grid-like graph from the image in which every node represents a pixel, and connects to its neighbouring pixels.

Virtual terminal nodes, S (source) and T (sink) are created and connected with infinite weight arcs to all pixels of the leftmost and rightmost columns of the image respectively

A "cut" is a partitioning of the nodes in the graph into two disjoint subsets S and T such that $s \in S$ and $t \in T$. To define a seam from a cut, we consistently choose the pixels to the left of the cut arcs. **The optimal seam is defined by the minimum cut which is the cut that has the minimum cost among all valid cuts**.

However, we cannot simply use any "cut" to form our seams. The reason is that a general cut does not define a valid seam for seam-carving, as it must satisfy two constraints:

- **Monotonicity**: the seam must include one and only one pixel in each row (or column for Horizontal seams).
- **Connectivity**: the pixels of the seams must be connected.

Graph Cuts for Image

In a standard grid graph construction, every internal node $P_{i;j}$ is connected to its four neighbours

$$Nbr(P_{i;j}) = \{ P_{i-1;j} , P_{i+1;j} , P_{i;j-1} , P_{i;j+1} \}.$$

We define the weight of arcs as the difference between the intensity of the corresponding pixels in the image either in the horizontal direction:

$$\partial x(i;j) = | I(i;j + 1) – I(i;j) |$$

Or in the vertical:

$$\partial y(i;j) = | I(i+1;j) – I(i;j) |$$

Under this formulation, Figure 2 (a) shows an optimal partition of the beach image into source and target parts. But this cut does not satisfy the seam carving constraints.

To keep the seam monotonic, the backward horizontal arcs are set to infinity (Figure 2(b)), while to keep it connected, diagonal infinity arcs are used (Figure 2(c)). The value of the energy difference calculated is stored on the forward horizontal arcs. This way, we've forced the minimum cut to formulate a connected and monotonic seam.

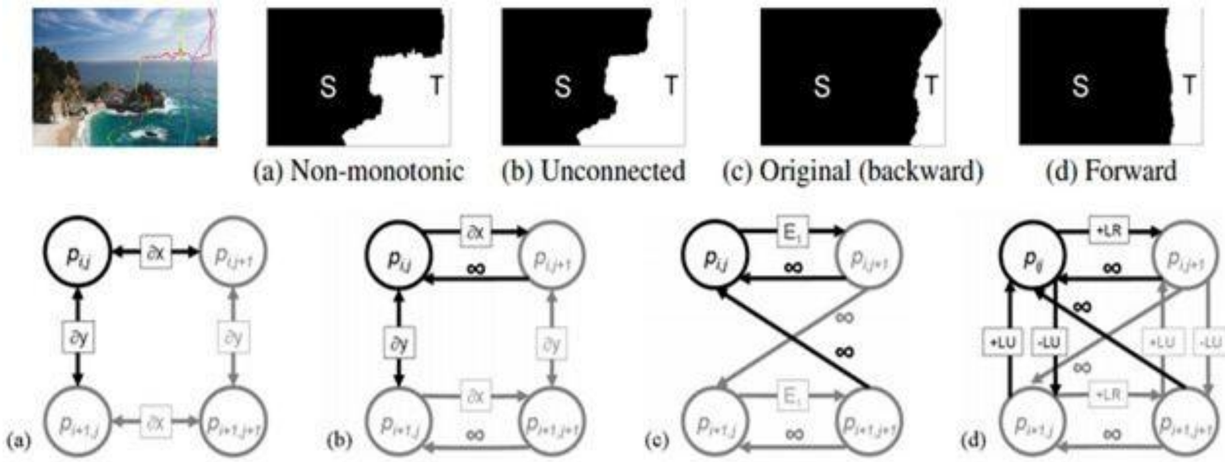
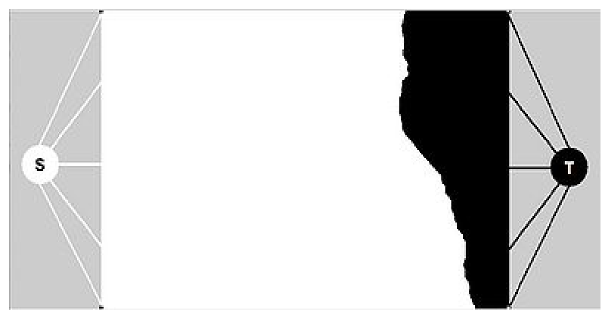


Figure 2

(a)



(b)

Figure 3: image (a) shows one seam detected using the graph cut algorithm using forward energy depicted in (b)

<u>Comparison between Graph Cuts and DP:</u>

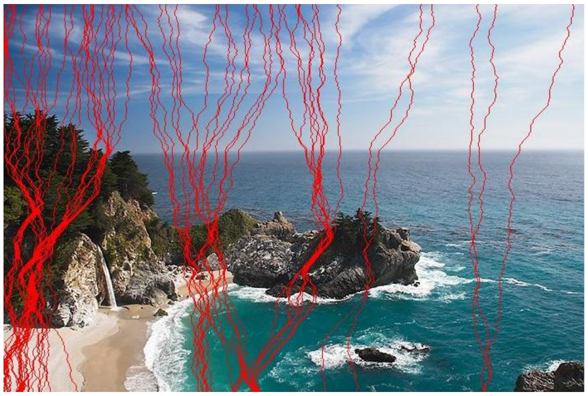
The algorithm allows graph cuts to find a slightly better path, but the difference is minimal.

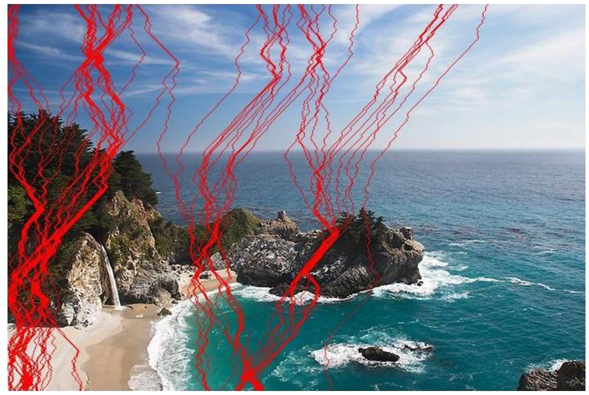
DP

GC

Following are the results on removing 50 seams in the vertical direction by both the Dynamic Programming and Graph Cut algorithm:

Dynamic Programing Graph Cuts





Conclusions:

- Dynamic Programming is much faster than Graph Cuts for still images. In the above examples we noticed that DP took around 5 seconds to remove 50 seams and the same operation was performed by Graph Cuts in 60 seconds.
- On an average the results the Graph Cut provides better results, but the difference is minimal as far as the images are concerned.
- Graph cuts method is more flexible and can be extended to videos.

Video Resizing

Static images are not the only media that is used on the varying devices these days. Videos have their place in this world and enjoy a much larger audience. Hence, the need to extend this method of seam carving to videos.

We can think about a video as a sequence of frames, and apply the seam carving for each video frame independently. This produces seams that are optimal for each frame, but are not guaranteed to be homogeneous between different frames. Consequently, the results contain jittery artifacts resulting in a choppy or jumpy video.

Figure 4: Seam carving on each frame independently creates locally optimal seams that can be totally different over time. This creates a jittery resized video.

So we need to think about a different approach that takes into consideration the successive frames as a unit, not just separated shots. Practically, we search for regions in the image that are of low importance in all video frames, generate a seam that passes through these regions, and remove that seam from the whole video. Seams calculated this way are called Static Seams, because they are constant through the whole video and don't change over time.

First, we calculate the energy for every pixel in each video frame separately. Then, assuming that the successive video frames form a cube, the frame width and height represent the two dimensions of a given cube face, and the third dimension is represented by the successive frames over time. Now, to find the least important regions in the whole video, we take each pixel in the X-Y plane, compare its energy with the corresponding pixels in the other frames, and take the maximum energy found. This way, we've found the maximum energy encountered in the video, but stored in a 2D view, reducing the problem back to image retargeting. We now compute the seam from that 2D grid, and apply that seam to each video frame, reducing the size of the whole video by one. We repeat this procedure to remove more seams.

Figure 5: Static seams for the golf video. The global energy function is shown using colour mapping from violet (low) to red (high). The actual static seams are shown for the golf sequence at the top.

<u>Advantages of Static Seam Carving:</u>

1) Simplicity & Speed.

2) Good results are given with stationary cameras, or when the foreground & background are separated.

<u>Drawback:</u>

If the camera is moving, or complex movements are present, Seams must change over time to adapt these movements.

Graph Cuts for Video Stream Context Aware Resizing

Extending graph cuts to video is pretty straightforward. We consider the X x T planes in the video cube and use the same graph construction as in X x Y. We connect the source and sink nodes to all left and right (top/bottom in the horizontal case) columns of all frames respectively.

A partitioning of the 3D video volume to source and sink using graph cut will define a manifold inside the 3D domain (Figure 4). Such a cut will also be monotonic in time because of the horizontal constraints in each frame that are already in place. This cut is globally optimal in the cube both in space and time. Restricted to each frame, the cut defines a 1D connected seam.

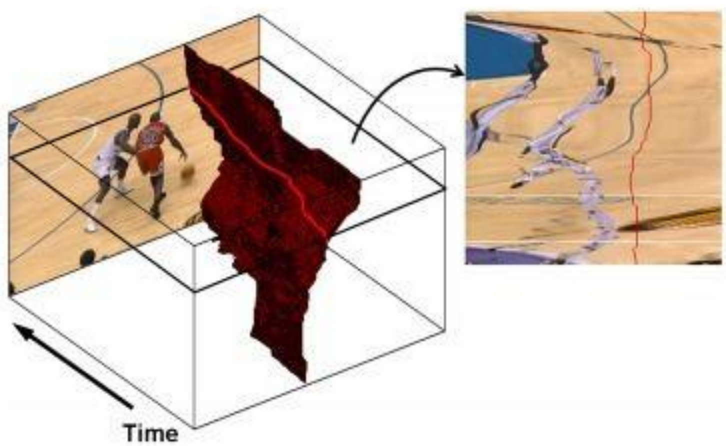


Figure 4: Intersection of every X x T plane with the seam surface

Seam carving is thus achieved on videos. However, this does not provide the best of results. This is because of the backward energy approach used. The fine lines and curves in the video will be jittery. Forward energy proposes to ensure better results in some situations. When we remove a seam from an image, the resultant image's energy may increase more for one seam removed as compared to another. The forward energy approach ensures that such a seam be chosen such that the resultant image energy is minimised.

Objective Analysis of the Seam Carved Images

We can also perform objective analysis of seam carving method by the following ways:

- Entropy measure

  In this method we will directly compute the energy of the total image and compare it with the energy of the original image. Since energy is the major factor that determines the optimal seams, this method would help us analyze the image mathematically.

- Difference in number of lines detected with the same parameter in the images.

  As the distortion in images is mostly due to the disfiguring of the regular patterns in the image, we can count the number of lines detected with a set of parameter in the image and compare it with the number of lines in the original image. If the number of lines have increased implies more amount of distortion was achieved and vice versa.

So far we have compared all the results using subjective measure only. In our next report we will be including these objective methods of analyzing seam carved Image.

➢ Energy based objective analysis.

As we know through the process of seam carving we remove the seams with the least energies from the image. Here the energy is defined by various factors like edge detail in an image, the global color, local color and entropy changes, diffusion of importance of energy of removed seam, etc.

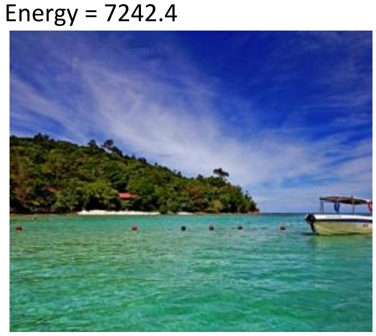
Here we compare the energies of cropped and seam carved images.

(a,b,c). [a] original image. [b]: seam caved image. [c]: cropped image.

Energy = 7261.6

Energy = 7242.4

Energy = 8052.8



Energy = 7467.1

Energy = 10823

Energy = 9812.5

Energy = 16972

Energy = 16900

➢ Correlation of images with original

Correlation of two signals determines how similar the signals are to each other. We use 2D correlation on output images of the two of the seam carving algorithm proposed (entropy energy and saliency) and the original image.

The closer is the value of the correlation to zero; the better is the output as it means the obtained image is closer to the input i.e. is the output image can be considered to have retained the most information in it.

➢ PSNR (Peak Signal to Noise Ratio)

The term determines the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. The PSNR is most commonly used as a measure of quality of reconstruction of lossy compression codecs (e.g., for image compression). The signal in this case is the original data, and the noise is the error introduced by compression. When comparing compression codecs it is used as an approximation to human perception of reconstruction quality, therefore in some cases one reconstruction may appear to be closer to the original than another, even though it has a lower PSNR (a higher PSNR would normally indicate that the reconstruction is of higher quality).

Here considering the output of the seam carving algorithms as the reconstructed signals and the original image as the original signal, we try to determine how similar are the two and how much the major information in the output have been saved.

The method has been used for comparison of entropy energy and saliency based image carving techniques.

Higher the PSNR worse is the output attained. Low PSNR determines good outputs.

- **Structural similarity analysis**

The structural similarity (SSIM) index is a method for measuring the similarity between two images. The SSIM index is a full reference metric, in other words, the measuring of image quality based on an initial uncompressed or distortion-free image as reference. SSIM is designed to improve on traditional methods like peak signal-to-noise ratio (PSNR), mean squared error (MSE) and correlation, which have proved to be inconsistent with human eye perception.

SSIM considers image degradation as perceived change in structural information. Structural information is the idea that the pixels have strong inter-dependencies especially when they are spatially close. These dependencies carry important information about the structure of the objects in the visual scene.

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Here,

μ_x the average of x;
μ_y the average of y;
σ_x^2 the variance of x;
σ_y^2 the variance of y;
σ_{xy} the covariance of x and y;
$c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ two variables to stabilize the division with weak denominator;
L the dynamic range of the pixel-values (typically this is $2^{\#bits\ per\ pixel} - 1$), here it is 255.
k1 =0.01 and k2 = 0.03 by default.

In order to evaluate the image quality this formula is applied only on luminance. The resultant SSIM index is a decimal value between -1 and 1, and value 1 is only reachable in the case of two identical sets of data.

Here although we have used structural dissimilarity. Which is

DSSIM = (1-SSIM);

The higher the value of DSSIM is the higher is the difference between the two images and hence we can say the worse off is the output. For better output we aim at lower DSSIM.

(a,b,c) [a]: original image [b] entropy energy seam carving [c] improved seam carving

Correlation value:	0.4119	0.4020
PSNR:	13.8444	13.7729
DSSIM:	0.2139	0.2205

Correlation value:	0.6993	0.7235
PSNR:	18.0659	17.8124
DSSIM:	0.4437	0.4240

Correlation value:	0.7631	0.7484
PSNR:	16.0634	15.9810
DSSIM:	0.4522	0.4237

Correlation value:	0.4749	0.2392
PSNR:	15.8218	14.0075
DSSIM:	0.2781	0.2195

Correlation value:	0.6751	0.3802
PSNR:	15.8247	13.0155
DSSIM:	0.3890	0.2607

Correlation value:	0.4322	0.3558
PSNR:	12.8518	12.3987
DSSIM:	0.6582	0.6321

Objective Analysis Using Line Detection:

- We can also provide some objective comments with the help of this method.
- As discussed earlier the major distortions are due to the broken lines in the image.
- Every point of huge distortion in the line makes it look like a separate line, thereby increasing the count of lines.

So we can conclude, if the number of lines in a seam carved image has increased, it has more distortions in it.

Original image (number of lines = 15)

no. of lines = 17 (stick represented through one line)

no of lines = 15 == same as the original image.. as it preserves all the straight lines.

Hence better results achieved.

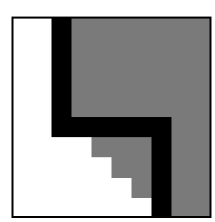
Conclusions based on objective analysis

- Here it can be noticed that in the **first set of images (boats) the entropy energy seam cropped image is actually better than the improved seam carved image.** Even though the correlation and PSNR value for the entropy energy seam carved is higher than the improved seam carved image, **the value for structural dissimilarity is lower for entropy energy seam carved image**.
- For the rest of the images improved seam carving gives perceptually better results. Also the value for structural dissimilarity for the rest is lower for improved seam carved image than entropy energy seam carved image.
- We can also notice that the difference between the values for DSSIM for the two seam carving for various images don't vary in magnitude largely owing to fact that both images are perceptually acceptable. **Except for the second last (mime) image. The value vary largely as the entropy seam carved image is completely lost.**
- Hence we can conclude that the DSSIM is a better objective parameter to determine how much perceptually good the seam carving has been performed.
- **Also it is visible that the proposed improved seam carving algorithm works effectively for a large variety of images.**
- **Line detection** algorithm gives very good results for images with lots of straight lines in them which otherwise would get significantly distorted.

General Drawbacks of seam carving method:

In general, we cannot always get an image which is optimally resized such that the "important aspects" stay the same size.

Our algorithm only minimizes along one dimension, so it can't decide to turn to follow a minimum-error surface orthogonal to the seam.
As an example, consider the above image.
The thick black line is the ideal seam (say) that should be removed from an image.
Because our algorithm is unable to "turn the corner" we get something less nice.



The seam that we get is represented by the boundary between the white and grey parts in the above example.
As a result, conditions may arise where a part of the image is “always” going to be distorted.

Another drawback of this method is that, when resizing the images, the “important” aspects are preserved but the non-important parts (generally the background) are carved out. This arises some visual discrepancies in the output. For example, we have a race car on the track, but our method carves out the road leaving behind only the car. The context of the image/video is lost in this way.

We propose an approach where resizing of images/videos for varied displays can be done by an optimal combination of conventional resizing (keeping the aspect ratio intact) and seam carving. By our method, not only are the key elements of the image/video preserved, but the context of this preserved part (aka. The background) is also partially preserved, which leads to better visual coherency.

Conclusions:

- In this report we experimented with various images and performed seam carving in both horizontal and vertical direction using Entropy, Image Gradient and Hog methods for Energy Mapping.
- This method works fine for almost all types of images if the number of seams removed is not high.
- This method works good for landscapes and images with a lot of flat regions in it.
- The quality of the image can be controlled by allowing more number of iterations.
- The computational cost is maximum in HoG followed by Entropy and least in Gradient method.
- Different methods are ideal for different images. As observed on our datasets the entropy method worked best for most of the images.
- “Regions of interest” can be protected by ensuring that the seams do not pass through the region (region with high energy). By this we can remove distortion in the interesting parts of the images.
- The concept of object protection has been extended to object removal.
- Object removal is trickier than object protection.
- In object removal we have also taken into consideration simultaneous insertion of seams in the removed region, keeping the image size same. This yielded better results.
- Seam Carving method has some constraints.
- If more than 50 % of seams are removed, the amount of distortion is high.
- The concept of seam removal can be reversed and extended to image expansion via seam insertion.
- Improved method of seam carving was implemented in this report.
- Line detection was used to minimize distortion and modify the energy content map.
- Line detection was also used for objective analysis of the image.

- Another important improvement was achieved by Saliency Detection, where we also considered the color mapping. It reduces significant distortion and also increases the focus on the interesting parts of the image.
- Another method of detecting seams was implemented named Graph Cuts.

- Dynamic Programming is much faster than Graph Cuts for still images.
- On an average the results the Graph Cut provides better results, but the difference is minimal as far as the images are concerned.
- Graph cuts method is more flexible and can be extended to videos.
- Primitive methods of applying seam carving to videos has been implemented but are not satisfactory. Graphcut method can be extended to videos for better results.
- Hence we can conclude that the DSSIM is a better objective parameter to determine how much perceptually good the seam carving has been performed.
- **Also it is visible that the proposed improved seam carving algorithm works effectively for a large variety of images.**
- **Line detection** algorithm gives very good results for images with lots of straight lines in them which otherwise would get significantly distorted.

References:

- S. Avidan and A. Shamir, Seam carving for content-aware image resizing. ACM Trans. Graph., 26(3):10, 2007
- P. Zarghan and S. Nassipour, Content-aware image resizing. Stanford 2012
- M. Rubinstein, A. Shamir and S. Avidan, Improved Seam Carving for Video Retargeting ACM Trans. Graph., 27(3):16, 2008
- M. Grundmann, V. Kwatra, M Han, I. Essa, Discontinuous Seam-Carving for Video Retargeting, IEEE CVPR, 2010
- Mansfield, P. Gehler, L.V. Gool and C. Rother, Scene Carving: Scene Consistent Image Retargeting, Microsoft Research
- J. Kiess, S. Kopf, B, Guthier and W. Effelsberg, Seam Carving with Improved Edge Preservation, Proc. of IS&T/SPIE Electronic Imaging (EI), 2010
- S. Cho, H. Choi, Y. Matsushita and S. Lee, Image retargeting using importance diffusion, Intl. Conf. on Image Processing, 2009
- R. Achanta and S. Susstrunk, Saliency detection for content-aware image resizing, IEEE ICIP, 2009

(all the above papers have been implemented in our project)

Work Done:

- Basic seam carving using gradient energy maps.
- Improvement of gradient energy based seam carving using entropy and histogram of gradient (HoG) methods.
- Object Protection and removal

- Saliency Implementation
- Innovative algorithm for improved seam carving using global, local saliency and gradient details.
- Improving seam carving with Importance diffusion and forward energy maps.
- Line detection and line preservation algorithm to improve seam carved outputs.
- Implementation of seam carving using Graph-Cuts.
- Extension of Graph-Cuts Algorithm for improved video retargeting.
- Objective analysis of output images using correlation, PSNR, DSSIM and line detection.

WHO DID WHAT?

- Apurva Kumar (201001083)
    - Object protection and removal
    - Implementation of Seam carving via graph-cuts.
    - Seam carving extended to videos.
    - Extension of Graph-Cuts Algorithm for improved video retargeting.
- Arushi Ladha (201030005)
    - Improvement of gradient energy based seam carving using entropy and histogram of gradient (HoG) methods.
    - Saliency Implementation
    - Innovative algorithm for improved seam carving using global, local saliency and gradient details.
    - Improving seam carving with Importance diffusion and forward energy maps.
    - Objective analysis of output images using correlation, PSNR and DSSIM.
- Masum Lodha (201001053)
    - Basic seam carving using gradient energy maps.
    - Line detection using Canny and Sobel gradient masks and their comparison.
    - Line preservation algorithm to improve seam carved outputs.
    - Objective analysis of output images using correlation, PSNR, DSSIM and line detection.

Nature of work done

- The nature of work done has been a mix-up of many things and a lot of experimentation.
- We have implemented all the research papers mentioned in the references above.
- We made our own algorithm for improved seam carving using for the first time local saliency data.
- We came up the methods for objective analysis of various seam carved, cropped, etc images.
- Our method uses line detection to tally the change in number of lines in the input and output image for objective analysis.