A Package for Seam Carving

Dong Liu[*]

University of Wisconsin-Madison
dliu328@wisc.edu

November 19, 2023

Abstract

Seam carving is a famous method for image resizing, in this paper we improve the Seam Carving method from some aspects, and in the end, we test the performance of several methods and original algorithm through the image classification test based on CNN.

I. Introduction

Seam carving is a quite famous method for resizing the image, it is a technique proposed by Avidan and Shamir1 for content-aware resizing. By using pixel removal, the algorithm removes paths of low energy pixels (seams) from top to bottom or from left to right to realize vertically or horizontally image resizing. However, shortcoming of seam carving has been there are several shortcomings for the seam carving because the simple energy map designation, brief energy dp function, disregard some special characteristics in the image etc. Specifically, we optimize the original seam carving model from the following aspects: applying bicubic interpolation when enlarging the image, using the LC algorithm to maintain the main characteristics of the image, applying the Cany line to augment the outline of the image , using the Hough transformation to detect the most significant line of the image and then augment the image, applying absolute energy function to improve the performance of the seam carving, designing dual energy function to implement another version of forward energy, in the end, we applied image classification experiments based on CNN to evaluate the performance of the seam carving.

[*]Welcome to my github: https://github.com/noakliu

II. Methods

There are the methods that we optimize the original models, and we will tell the details of them in later.

- apply bicubic interpolation when enlarging the image to smooth the image
- use the LC algorithm to maintain the main characteristics of the image
- apply the cany line detection to augment the outline of the image
- use the Hough transformation to detect the most the most significant line of the image and then augment the image
- apply absolute energy function to improve the performance of the seam carving
- use dual energy function to implement another version of forward energy
- use image classification experiments based on CNN to evaluate the performance of different enhancement methods of the seam carving[1].

III. Bicubic Interpolation when enlarging the image

I use double cubic policy for smoothing the value for the image. The algorithm uses the gray values of 16 points around the points to be sampled for cubic interpolation, which takes

[1]This will be used for evaluation

into account not only the gray influence of 4 direct adjacent points, but also the change rate of gray value between adjacent points.

The interpolation converts discrete data into continuous functions by an operation similar to convolution. General Meaning of this method is that according to the scale relationship, the floating-point pixel coordinate position (decimal representation) of the target pixel mapped in the original image is obtained. Take the surrounding 4x4 = 16 pixels, and calculate the weight according to the position relationship (distance). Finally, calculate the weighted sum of the surrounding 16 pixel values to obtain the pixel value of the target pixel. According to the resizing proportional relationship we can get that the corresponding coordinates of B(x,y), and then we can use the fomulas shown below to calculate the following result.

$$w(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1, & \text{if } x{<=}1 \\ a|x|^3 + 5a|x|^2 + 8a|x| - 4a, & \text{if } 1{<}x{<}2 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$B(X,Y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} * w_i * w_{ij} \tag{2}$$

For the pixel point (x, y) to be interpolated (x and y can be floating-point numbers), take the 4x4 neighborhood point (x_i, y_j) near it, i, j = 0,1,2,3. The interpolation calculation is carried out according to the following formula:

$$f(x,y) = \sum_{i=0}^{3} \sum_{j=0}^{3} f(x_i, y_j) * w(x - x_i) * w(y - y_i) \tag{3}$$

Below we will show some comparison examples.

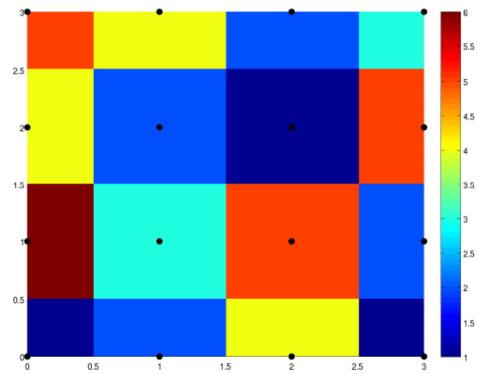


Figure 1: *original image*

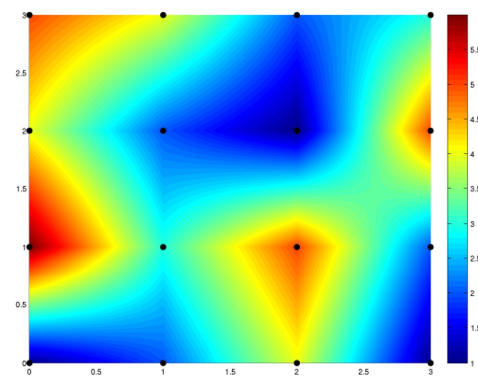


Figure 2: *bilinear interpolation*

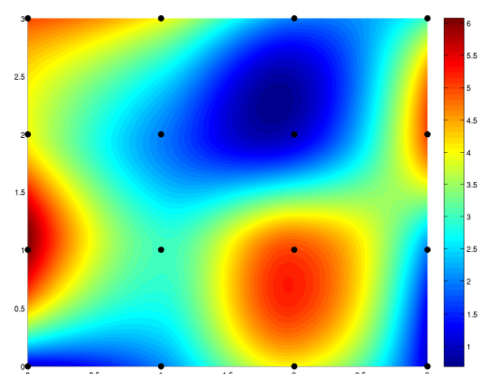


Figure 3: *bicubic interpolation*

IV. LC Policy

I use LC Policy to enhance the main characteristics of the image. Calculates the global contrast of a pixel over the entire image, that is, the sum of the distances of the pixel from all other pixels in the image in color as a significant value for the pixel. The significant value of is calculated as follows(note that I_i is grey scale):

$$SalS(I_k) = \sum_{\forall I_i \in I} ||I_k - I_i|| \tag{4}$$

Given an image, suppose that we know I_k for each pixel and I_k=a_m , frequency of a_n is f_n, then the above formula can be further reconstructed:

$$SalS(I_k) = \sum_{n=0}^{255} f_n ||a_m - a_n|| \tag{5}$$

Figure 4: *the outlier detected by LC algorithm*

V. Canny Line

I use Canny Line to perform a line-augmenting performance.

For canny edge detection filter

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{-x^2+y^2}{2\sigma^2}} \tag{6}$$

$$G(i,j) = \sqrt{G_x(i,j)^2 + G_y(i,j)^2} \tag{7}$$

$$\theta(i,j) = arctan\frac{G_x(i,j)}{G_y(i,j)} \tag{8}$$

Figure 5: *original picture*

Figure 6: *the edges detected by Canny detector*

VI. Hough Transformation

I use Hough Transformation for line-augmenting performance.

The key idea of the Hough transform is to vote evidence from the image domain to the parametric domain, and then detect shapes in the parametric domain by identifying local maximal responses. In the case of line detection, a line in the image domain can be represented by its parameters, e.g., slope, and offset in the parametric space. Hough transform collects evidence along with a line in an image and accumulates evidence to a single point in the parameter space. Consequently, line detection in the image domain is converted to the problem of detecting peak responses in the parametric domain.

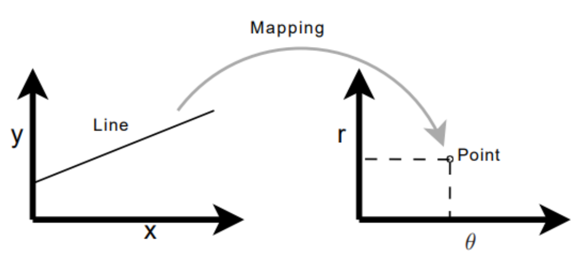
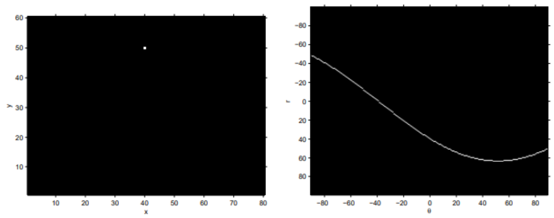


Figure 7: *mapping from space to points*



(a) Point p_0. **(b)** All possible lines through p_0 represented in the Hough space.

Figure 8: *mapping from points to space*

Figure 9: *original picture*

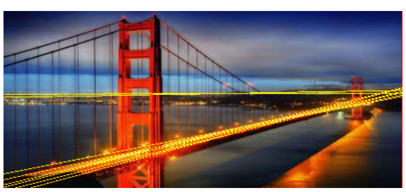


Figure 10: *the lines detected by hough transformation with the background*

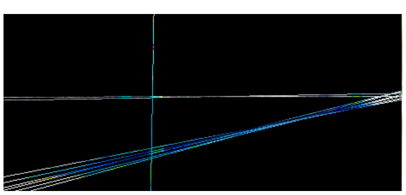


Figure 11: *the lines detected by hough transformation*

Algorithm 1 Hough Transformation for Lines

Require: *Image*: input digital image;
Ensure: (x, y): the coordinate of each pixel
grey: the grey value of each pixel in the picture

1: Edge detection, e.g. using the Canny edge detector
2: Mapping of edge points to the Hough space and storage in an accumulator.
3: Interpretation of the accumulator to yield lines of infinite length. The interpretation is done by thresholding and possibly other constraints.
4: initialize parameter space rs, thetas
5: Create accumulator array and initialize to zero
6: for each edge pixel
7: for each theta
8: calculate r = x cos(theta) + y sin(theta)
9: Increment accumulator at r, theta
10: Find Maximum values in accumulator (lines)
11: Extract related r, theta
12: Conversion of infinite lines to finite lines.

VII. Absolute Energy Equation

I use Absolute Energy Equation to solve the weak dp performance due to the weakness of one-direction dp transformation of the original method. The proposed absolute energy cost functional attempts to better handle images with high image detail concentration by taking into account the energy gradient along the seams being removed. This is accomplished by incorporating the energy gradient alongside the backward and forward energy functionals into the dynamic programming process, giving us the following cumulative energy matrix update formulation, The additional of the energy gradient cost functional is designed to ensure that seams do not cross through areas that contain local extrema. This is due to the fact that even minor local extrema require an energy gradient to be considered a minimum or maximum, and thus considering this energy gradient should help accentuate even seemingly insignificant extrema. As such, it can be said that the proposed absolute energy cost functional takes a very conservative approach when determining areas of the image to remove. It should be noted that a forward difference approximation for the energy gradient was chosen in the proposed absolute energy cost functional as other approximations would not allow an in-place update of the energy matrix . By taking a forward difference approximation, the amount of computation time and storage space required for the calculations will be greatly reduced.

$$s = \arg\min_{s}\{E(s)\} = \arg\min_{s}\{\sum_{i=1}^{n} e(I(s))\} \tag{9}$$

$$e(I) = |\frac{\partial I}{\partial x}| + |\frac{\partial I}{\partial y}| \tag{10}$$

$$\begin{cases} C_L(i,j) = |I(i,j+1) - I(i,j-1)| \\ C_R(i,j) = |I(i,j+1) - I(i,j-1)| \\ C_U(i,j) = |I(i,j+1) - I(i,j-1)| \end{cases} \tag{11}$$

Original energy function

$$\begin{aligned} e(i,j) = e(i,j) + \\ min \begin{Bmatrix} C_L(i,j) + e(i-1,j-1) \\ C_R(i,j) + e(i-1,j+1) \\ C_U(i,j) + e(i-1,j) \end{Bmatrix} \end{aligned} \tag{12}$$

Absolute energy function

$$\begin{aligned} e(i,j) = e(i,j) + |e(i,j+1) - e(i,j)| + \\ |e(i+1,j) - e(i,j)| + \\ min \begin{Bmatrix} C_L(i,j) + e(i-1,j-1) \\ C_R(i,j) + e(i-1,j+1) \\ C_U(i,j) + e(i-1,j) \end{Bmatrix} \end{aligned} \tag{13}$$

VIII. Dual Gradient Energy Equation

In the original method given by the baseline seam carving, the solution is Scharr operator to detect the edge. We can also achieve edge filter by numerical difference, and here we will consider dual gradient energy equation.

Consider Taylor expansion,

$$\begin{aligned} f(x+\triangle x) = f(x) + \triangle x f'(x) + \triangle x^2 \frac{f''(x)}{2!} + \\ \triangle x^3 \frac{f'''(\xi_1)}{3!} + \triangle x^4 \frac{f^{(4)}(x)}{4!} + \triangle x^5 \frac{f^{(5)}(\xi_1)}{5!}, \\ \xi_1 \in (x, x+\triangle x) \end{aligned} \tag{14}$$

$$\begin{aligned} f(x-\triangle x) = f(x) - \triangle x f'(x) + \triangle x^2 \frac{f''(x)}{2!} - \\ \triangle x^3 \frac{f'''(\xi_2)}{3!} + \triangle x^4 \frac{f^{(4)}(x)}{4!} - \triangle x^5 \frac{f^{(5)}(\xi_2)}{5!}, \\ \xi_2 \in (x, x+\triangle x) \end{aligned} \tag{15}$$

$$\begin{aligned} f(x+2\triangle x) = f(x) + 2\triangle x f'(x) + 4\triangle x^2 \frac{f''(x)}{2!} + \\ 8\triangle x^3 \frac{f'''(\xi_3)}{3!} + 16\triangle x^4 \frac{f^{(4)}(x)}{4!} + 32\triangle x^5 \frac{f^{(5)}(\xi_3)}{5!}, \\ \xi_3 \in (x, x+\triangle x) \end{aligned} \tag{16}$$

$$\begin{aligned} f(x-2\triangle x) = f(x) - 2\triangle x f'(x) + 4\triangle x^2 \frac{f''(x)}{2!} - \\ 8\triangle x^3 \frac{f'''(\xi_4)}{3!} + 16\triangle x^4 \frac{f^{(4)}(x)}{4!} - 32\triangle x^5 \frac{f^{(5)}(\xi_4)}{5!}, \\ \xi_4 \in (x, x+\triangle x) \end{aligned} \tag{17}$$

by calculation, we can get the $O(\triangle x^2)$ forward/backward difference approximation

$$f'(x) \approx \frac{-3f(x) + 4f(x+\triangle x) - f(x+2\triangle x)}{(2\triangle x)} \tag{18}$$

$$f'(x) \approx \frac{3f(x) - 4f(x-\triangle x) + f(x-2\triangle x)}{(2\triangle x)} \tag{19}$$

if we take $\triangle x = \delta = 1$, and we consider forward approximation

$$f_x(x,y) \approx \frac{-3f(x,y) + 4f(x+\delta,y) - f(x+2\delta,y)}{2} \tag{20}$$

$$f_y(x,y) \approx \frac{-3f(x,y) + 4f(x,y+\delta) - f(x,y+2\delta)}{2} \tag{21}$$

as for energy calculation, we add up the square of the energy for each channel in each direction and then calculate for the total energy.

$$\triangle_x^2(x,y) = R_x(x,y)^2 + G_x(x,y)^2 + B_x(x,y)^2 \tag{22}$$

$$\triangle_y^2(x,y) = R_y(x,y)^2 + G_y(x,y)^2 + B_y(x,y)^2 \tag{23}$$

$$energy = \sqrt{\triangle_x^2(x,y) + \triangle_y^2(x,y)} \tag{24}$$

IX. Result Evaluation

I use CNN to compare the result performance of the different method.

In this experiment, we realize image classification by construct a neural network, and evaluated the resizing effect by comparing the image classification results of different methods. The dataset for classification is from CIFAR-10 dataset.

the workflow of the experiment

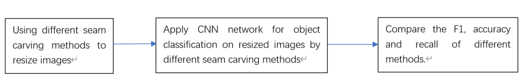


Figure 12: *workflow of experiment*

you can check the result of experiment in the bottom of the paper, and you can check the code in the evaluation module of the code section.

Figure 13: *error and accuracy of an sub-experiment Note: (results pictures of each sub-experiment can be found in the code part)*

Figure 14: *the picture processed by doublecubic method*

Figure 15: *the picture processed by absolute energy method*

Figure 16: *the picture processed by canny method*

Figure 17: *the picture processed by dual gradient energy method*

Figure 18: *the picture processed by Hough Transformation method*

Figure 19: *the picture processed by LC method*

X. Discussion

i. The shortcoming of bicubic interpolation

Shadows appear at upper and right of the image since the pixels near those two edges of this picture is darker than the other part.

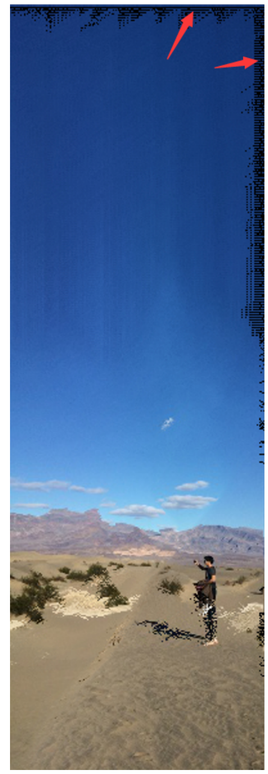


Figure 20: *the outlier of doublecubic method*

ii. The well application performance of Hough Transformation plus seam carving method

Hough Transformation plus seam carving method performs much better than the original seam carving method for images with more straight lines.

Figure 21: *Hough transformation plus seam carving method*

Figure 22: *original seam carving method method*

iii. some further ideas

Algorithm 2 GrabCut method for seam carving method

Initialize:

construct the image as a graph

background T_B, let foreground$T_F = \emptyset$,$T_U = \overline{T_B}$, $\alpha_n = 0$ where $n \in T_B$, $\alpha_n = 1$ where $n \in T_U$.

Iteration:

1. $k_n := \arg\min k_n D_n(\alpha_n, k_n, \theta, z_n)$
2. $\underline{\theta} = \arg\min_{\underline{\theta}} U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, z)$
3. $\{\alpha_n : n \in T_U\} = \arg\min_{\mathbf{k}} \mathbf{E}(\underline{\alpha}, \mathbf{k}, \underline{\theta}, z)$
4. Repeat Step 1.

the cut finding above is the seam for seam carving to delete or add

References

[Akshaya Mishra, 2007] Shai Avidan, Ariel Shamir Seam Carving for Content-Aware Image Resizing

[Johannes Kiess, 2014] Johannes Kiess. Seam Carving with Improved Edge Preservation study.

[Zijuan Zhang, 2015] Zijuan Zhang. The Comparison of Classic Image Resizing Methods study.

[Michael Frankovich, 2011] Michael Frankovich. Enhanced Seam Carving via Integration of Energy Gradient Functionals

Table 1: *The evaluation result of different methods*
(Note: TES: test set, VS: validation set,TRS: train set
ORG: original seam carving method
DI: doublecubic interpolation
HT:Hough Transformation
AE: absolute gradient function
DG: dual gradient function).

Model		Size		Epoches	F1	Accuracy			Recall		
		X size	Y size			TES	VS	TRS	TES	VS	TRS
ORG	exp1	300	600	10	93.2%	93.2%	93.2%	91.1%	92.3%	93.3%	95.9%
	exp2	400	500	8	94.1%	93.4%	93.8%	92.9%	92.8%	95.7%	93.6%
	exp3	500	800	11	91.8%	91.4%	91.2%	93.1%	93.3%	92.2%	92.7%
DI	exp1	300	600	10	95.3%	93.6%	92.5%	93.1%	98.3%	93.3%	91.2%
	exp2	400	500	8	94.2%	93.5%	92.4%	92.3%	95.8%	97.2%	94.2%
	exp3	500	800	11	95.5%	94.4%	90.1%	92.3%	97.1%	92.3%	92.8%
LC	exp1	300	600	10	95.3%	93.1%	94.1%	90.6%	97.2%	93.6%	96.1%
	exp2	400	500	8	94.2%	94.3%	96.6%	92.3%	94.1%	96.6%	93.1%
	exp3	500	800	11	94.3%	94.4%	91.2%	93.1%	94.3%	92.7%	95.9%
Canny	exp1	300	600	10	97.2%	97.2%	94.2%	91.1%	97.3%	94.3%	96.2%
	exp2	400	500	8	95.7%	94.2%	97.5%	92.3%	97.2%	98.8%	93.7%
	exp3	500	800	11	93.8%	94.4%	91.8%	93.8%	93.3%	95.7%	93.9%
HT	exp1	300	600	10	96.6%	95.2%	94.3%	93.1%	97.9%	94.1%	96.4%
	exp2	400	500	8	94.1%	96.4%	98.0%	91.4%	92.2%	96.2%	92.3%
	exp3	500	800	11	94.8%	96.4%	91.8%	97.1%	93.2%	91.7%	92.6%
AE	exp1	300	600	10	95.2%	97.2%	93.2%	96.7%	93.3%	98.0%	91.5%
	exp2	400	500	8	96.4%	97.4%	96.8%	98.9%	95.8%	97.7%	94.2%
	exp3	500	800	11	96.3%	96.1%	93.2%	94.4%	96.5%	91.6%	95.2%
DG	exp1	300	600	10	95.6%	97.2%	94.2%	91.1%	92.3%	97.3%	96.8%
	exp2	400	500	8	93.2%	92.4%	96.8%	92.9%	95.8%	93.2%	95.2%
	exp3	500	800	11	94.7%	96.4%	92.2%	92.5%	93.1%	95.8%	92.3%