ITCS381 Introduction to Multimedia Systems

Project: Forward Seam Carving

Group members (section 1)

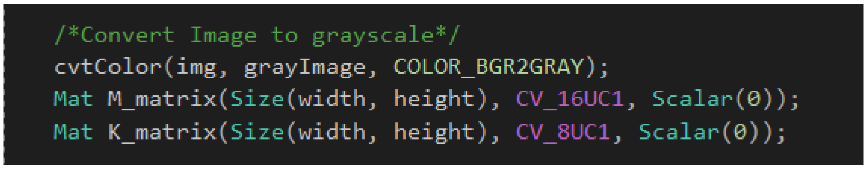
1. Vipawan Jarukitpipat	6088044
2. Dujnapa Tanundet	6088105
3. Thanakorn Pasangthien	6088109
4. Nontapat Pintira	6088118
5. Arada Puengmongkolchaikit	6088133

Introduction

Seam carving is an efficient method for resizing the digital images which can be used for both reduction and expansion of the images. It generates an energy map from gradient intensity of pixels and searches for seams, which is an optimal path of pixels on an image in a vertical path (from top to bottom), or horizontal path (left to right), where is defined by minimum energy areas of the image. Deleting or inserting pixel along the seams can change the ratio of an image, while still retaining all details of it.

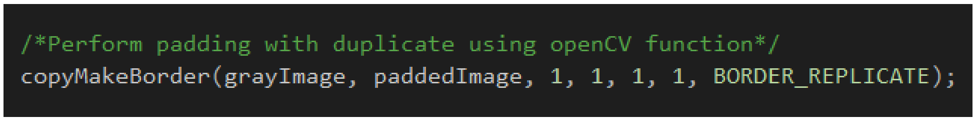
Implementation

1. **Convert image to grayscale:** convert image to grayscale in order to find M matrix of the image easier.



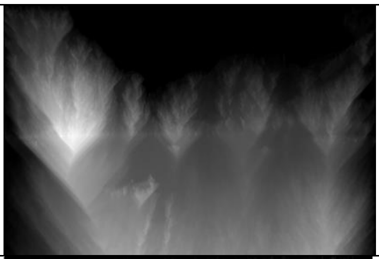
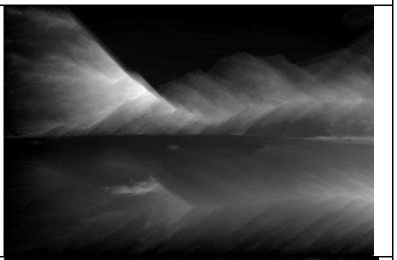
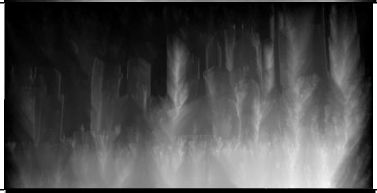
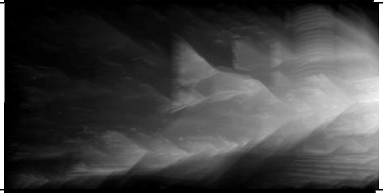
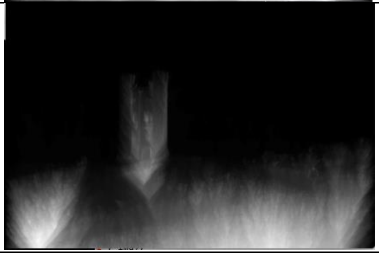
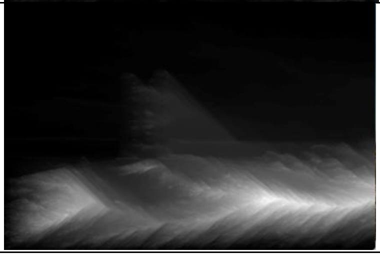
```
/*Convert Image to grayscale*/
cvtColor(img, grayImage, COLOR_BGR2GRAY);
Mat M_matrix(Size(width, height), CV_16UC1, Scalar(0));
Mat K_matrix(Size(width, height), CV_8UC1, Scalar(0));
```

2. **Padding with duplicate:** using the library function of OpenCV 'copyMakerBorder()' with the border type 'BORDER_REPLICATE' to get padding duplicate by 1.



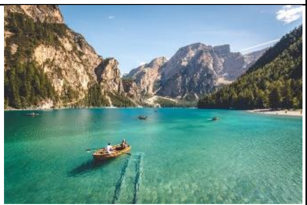
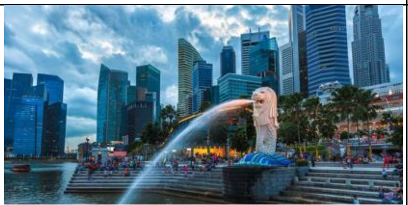
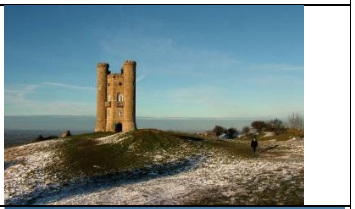
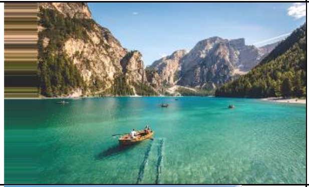
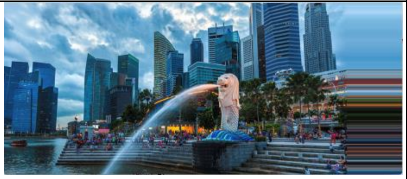
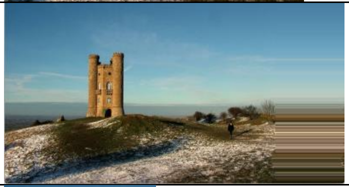
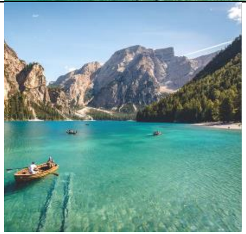
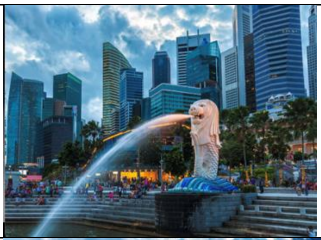
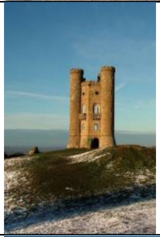
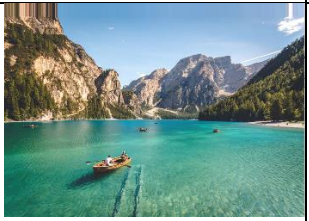
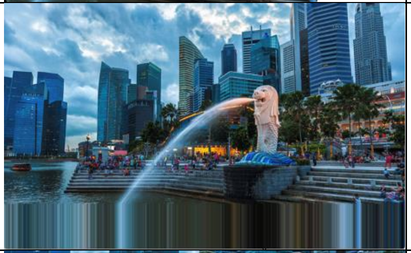
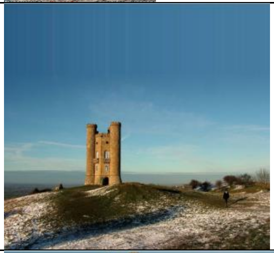
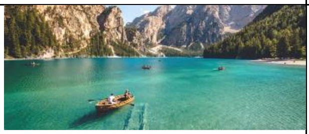
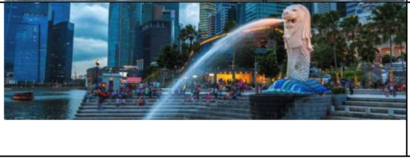
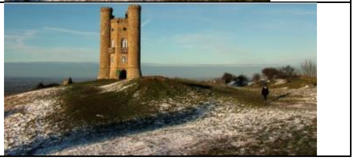
```
/*Perform padding with duplicate using openCV function*/
copyMakeBorder(grayImage, paddedImage, 1, 1, 1, 1, BORDER_REPLICATE);
```

3. **Calculate energy cL, cU, cR and Matrix M, K:** the nested loop is started from 1 because the image has a duplicate border. Within the loop, we calculate cL, cU, cR and then find the mL, mU, mR which is the addition between current value in image matrix and cL, cU, cR. After that, we find the minimum among these three values in order to store it in the M matrix. The offset of the choosing path will be stored inside the K Matrix.

	M matrix – Vertical Energy	M matrix – Horizontal Energy
		
		
		

4. **Find the optimal seam:** In order to find the best seam, we first create the matrix for stored the path of the best seam. After that, we find the least value in the bottom row of M matrix. Then, with the help of K matrix, we can generate the path of the best seam and stored it into the seam matrix in order to use for deletion or insertion later.
5. **For the horizontal direction:** the difference from the vertical direction is that we assign the value oppositely. Form the row is i-1, the column is j-1, to row is j-1, the column is i-1 instead.
6. **Deletion and Insertion:** we segment the image into 2 parts, to the left of the seam and to the right of the seam (the upper and the lower of the seam in case of horizontal direction). For the deletion, we merge the two segments together by using 'hconcat/vconcat' and delete the width/height. For the insertion, we add the seam before we merge the two segments together and increase the width/height.

Results

	Picture 1	Picture 2	Picture 3
Original Image			
Vertical Insertion			
Vertical Deletion			
Horizontal Insertion			
Horizontal Deletion			

Difficulty

1. Research openCV document

Our group spent a lot of time, research various built-in function from openCV in order to help us saving time and make our code easy to read. Examples for the built-in that we use such as copyMakeBorder, hconcat, and vconcat.

2. Language Difficulty

Most of our group member never have experience in C++ before. Therefore, the steep learning curve resulted in members have to research both the document and extend the functionality of C++. However, overall, it’s a fun learning experience.

3. Applied Logic to the horizontal direction

As the project required our program to perform seam carving in both vertical and horizontal direction. During the development of the program, we have the change the source code multiple time in order to make it work in a horizontal direction.

4. Datatype

OpenCV provides many data type for us to use which require the understanding of what kind of image that we want to produce or represent, which create multiple problems during the early stage of our development.

5. Development Compatibility

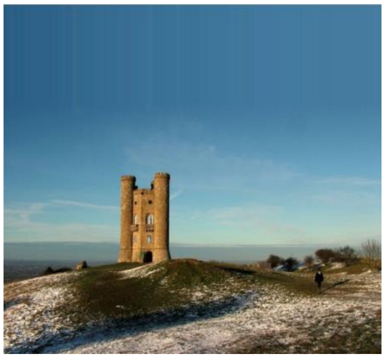
Among our group there are both Mac user and Window user, so the synchronize between the IDE and OpenCV in both systems give us the difficulty during the development.

Challenges and possible future developments

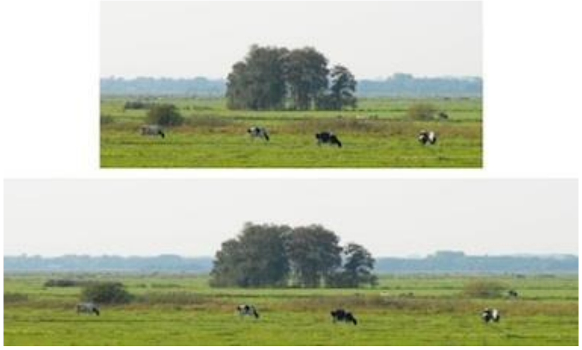
1. Smarter insertion

   We can further improve our insertion in both directions to insert the seam which results in a more natural way of the image.

   Our result

   Better result

2. Better energy function and application to video.

   We also can improve our energy function and use it in the video and the moving object.

3. Much faster removal of multiple seams

   We can further optimize our function to increase removal speed and number of the seam that we can find per computation.