

- **Lab Objective**

使用 Seam Carving Algorithm 來實作 image resizing, object removal and amplification。

- **Design Implementation**

我們首先解釋 Seam Carving 的核心想法：一般我們在做圖片 resizing 時，有兩個常見的方法，分別是 cropping(裁剪)或是 scaling(縮放)，但他們都會使生成的影像不太自然(ex. 物件被截掉、某個 dimension 變窄)，如下圖所示：

Original



Scaling



Cropping



Seam Carving 的想法在於我們找出圖片中的能量低的部分(我們稱作 **seam**，圖片中的一串不重要像素)，並在他周圍進行操作。由於這部分的能量比較低，在這附近作操作在視覺上比較不明顯，所以生成的圖片也比較自然。下面簡單講一下我們找 seam 的步驟：

1. Calculate Energy

我們對每一個 pixel 計算他的能量，這可以透過前幾個 Lab 的 gradient filter 來達成。

$$Energy = \left| \frac{dI}{dx} \right| + \left| \frac{dI}{dy} \right|$$

若一張圖片有多個 channel(ex. RGB)，則 $\left| \frac{dI}{dx} \right|$ = sum of absolute value of each channel's gradient.。

2. Find Seam by Dynamic Programming

我們可以使用 Dynamic Programming 的方式來找 Seam，以 vertical seam 為例：

我們使用一個額外的 matrix M ， M 的大小跟原先的 image 一樣。 $M(i, j)$ 代表以這個 pixel 為終點的 seam 所帶有的最小能量。注意 seam 中每一個 pixel 必須與為前一個 pixel 的右下、正下、左下，也就是 $I(i, j)$ 的下一個必須是 $I(i + 1, j - 1)$, $I(i + 1, j)$, $I(i + 1, j + 1)$ 。因此我們得到下面的關係式

$$M(i, j) = Energy(i, j) + \min\{M(i - 1, j - 1), M(i - 1, j), M(i - 1, j + 1)\}$$

透過 Dynamic Programming，我們可以從第一個 row 開始往下推，進而找到最後一個 row 的所有 $M(i, j)$ 。接著我們從最下方的 row 去找最小的 $M(i, j)$ ，接著根據 seam 的規則一路往上推，我們就能找到能量最低的 seam。

接著展示我們實作的成果

1. Seam Carving

這部分我們透過前面的方法找到 minimum energy seam，然後將它拿掉，重覆做直到圖片大小縮減到我們想要的大小。結果如下：

Scaling



Seam Carving

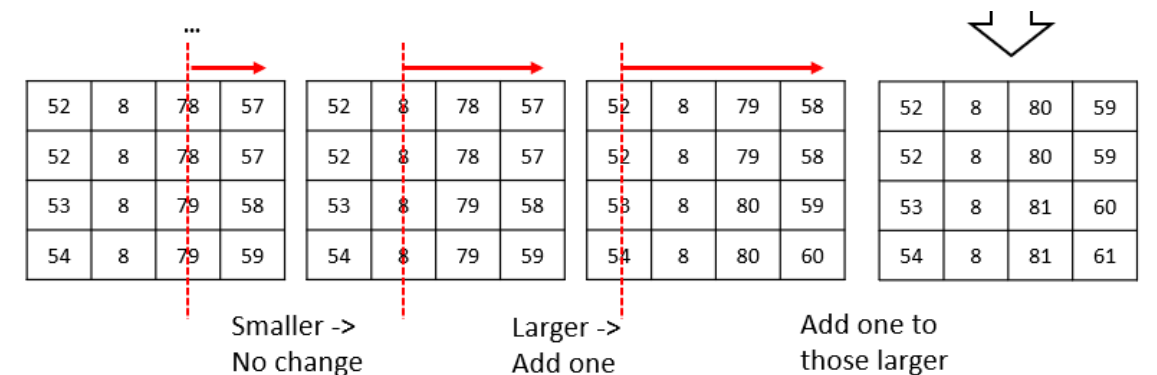


可以看出用 seam carving 做出來的結果中，圖片中的岩石比較符合原圖中的比例，運用 scaling 做出來的圖片中的岩石有種左右被壓扁的感覺。

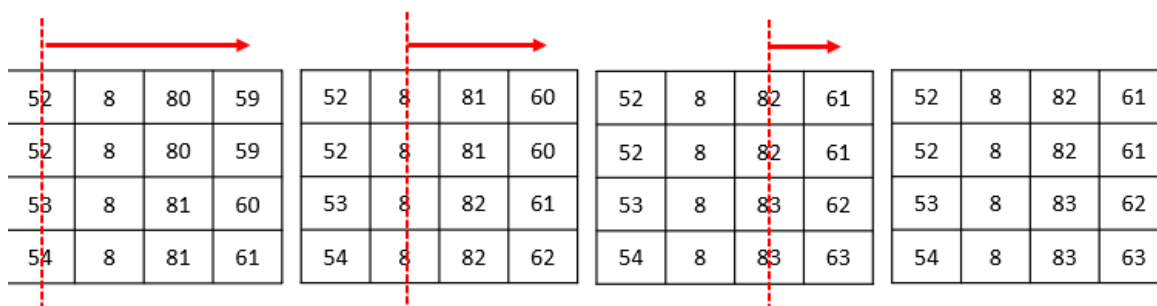
2. Seam Insertion

前一部分我們做了水平方向的壓縮，接著這部份我們來實現水平方向的拉伸。假設我們需要將圖片增加 N 個 column，我們照以下的步驟：

- 對圖片做 N 次 seam carving，找出 lowest, 2nd lowest ... nth lowest energy seam。
- 由於我們在做 seam carving 時，後面找到的 seam 的 column index 會受前面的影響，所以我們需要對他們做 index update，最右邊一行開始 update，如果比右邊的小則右邊的 row 要加一：



- 接著我們要計算要 insert 的 seam 的 column index，所以我們要在做一次 index update，這次我們要從最左邊這行開始做，因為我們會從 lowest energy seam 的地方開始做 insertion，如果右邊的行比較大則右邊的行要加一：



- 在我們計算好的 column index 處插入 seam，我採用的方法是使用其左右的 pixel 做 interpolation(ex. 取平均)。

實作結果如下：

Scaling



Seam Insertion



從圖片中央的岩石可以看出，**seam insertion** 的方式不會隨意使整塊石頭隨著等比例放大，幾乎保有原先圖片的尺寸，而是在其他處(ex.旁邊的海、右側岩石上的植物處)加上額外的 pixel，不使主物件失焦。而使用 **scaling** 的方式反而使圖片有些微模糊。

● Discussion

1. Insertion at same place

在 seam insertion 中，我們原先的做法是找最小的 N 個 seam，然後在這幾個位置去做 insertion，我們所需的時間複雜度約為 $O(row * N^2 + row * col)$ (打部分花在 index update 中)，如果 N 很大就會大幅增加運算時間。若我們改採用只找最小的 seam，然後在這邊做 N 次 insertion，那運算複雜度就會降低許多。但是這樣也會導致結果不太對，結果如下：



可以發現，一開始找到的 seam 位在圖片最左邊，若我們使用原先的 interpolation 方式，就會在這邊複製好幾個相同的 pixel，導致這種奇怪的現象。這種方法雖然運算較快，但生成的圖片不符合我們預期，因此不算是一個好方法。

2. Horizontal Seam Carving

這部分的做法跟前面的 Vertical Seam Carving 相同，只是我們在做 Dynamic Programming 改成從最左邊做到最右邊，關係式如下：

$$M(i, j) = \text{Energy}(i, j) + \min\{M(i-1, j-1), M(i, j-1), M(i+1, j-1)\}$$

實作結果如下，我們同樣去比較 scaling 和 seam carving 在將 row 變成一半的結果：

Scaling



Seam Carving



在 scaling 中，中間的岩石幾乎被壓扁，而 seam carving 則能保持中間岩石的完整性。但 seam carving 的結果也有其缺點，將 seam carving 的結果與原圖片相比，可以發現 seam carving 拿掉的是背景中天空、以及一部分的海洋，甚至使後方的天空與海平面交界不是一個橫線，出現相當不自然的地方。原因在於，我們找到的 lowest energy seam 不一定會是一直線，這導致某些物體的形狀、輪廓可能會因 seam carving 而被改變。

3. Object Removal and Amplification

這部分我們來實作 object removal 以及 object Amplification。首先是 object removal，我們會先指定一塊區域(mask)，我們的目的就是把這塊區域消除掉，我採用的步驟如下：

- 用 gradient filter 計算能量
- 為了讓 seam 保證通過 mask 所在的區域，我們根據 seam 的特性：Low energy，我們可以將 mask 處的 energy 改成一個很小的值(ex. -100)
- 做 seam carving 直到 mask 內所有的 pixel 被移除

假設我們想要拿掉圖片中藍色圈起來的區域



結果如下：



可以看到被圈起來的岩石確實被消除掉了，不過中間的岩石的一部分也被消除了而且圖片的 size 也改變了。未來如果要做效果更好的 object removal 或許可以再加上一塊 mask，這塊 mask 跟前面的 mask 剛好相反，是 protected(不能有 seam 通過)，這樣就能避免我們要保留的物件被刪除掉。另外，我們也可以透過在最後加上 seam insertion 讓圖片跟原先大小一樣，也不會因使用 scaling 而不自然。

接下來是做 object Amplification，我們希望強化圖片中物件的比重和強度，我採用的步驟如下：

- a. 將圖片用 scaling 的方式放大
 - b. 做 horizontal seam carving 和 vertical seam carving 讓圖片回復成原先大小。
- 實作結果如下：



可以看到圖片中的物件(岩石、岩壁、植物)都在圖片中佔有更大的比重，但也可以很明顯發現會發生如前面做 horizontal seam carving 時遇到的海平面彎曲的問題，圖片也因為 scaling 出現模糊的現象。

4. Horizontal v.s. Vertical seam carving in Object removal

在 object removal 中，我們比較 remove horizontal seam 和 vertical seam 的區別。我們用一樣的圖，一樣移除下方的岩塊。接著去比較成果差別以及移除的 seam 的數量。



結果如下：

Remove vertical seam (-102 seams)



Remove horizontal seam (-28 seams)



兩者都能有效的把指定物體移除，也都有些不自然之處(vertical：上方的岩石中間也被移除；horizontal：左方的沙灘被截去不少)。而去比較兩者移除的 seams 的數量，horizontal seam carving 移除的 seams 較少，因此在這個 case 中我認為 horizontal 會是比较好的方法因為他移除的 seams 較少，比較接近原先的圖片。而移除的 seams 的數量不同的原因在於，我所選擇的 mask 的形狀，以這個例子為例，我所選擇的 mask 的形狀是橫長條狀，column 比 row 多，這導致我們需要刪除比較多的 vertical seam 才有機會將整個 mask 刪除。

● Conclusion

這個 Lab 中我運用助教給的圖實作 seam carving algorithm。並嘗試使用這個技術來去時做 image resizing、object removal 以及 amplification。從結果可以看出 seam carving 可以讓我們 resize 後的圖片中的物件更自然，不會變形。然而，我們也會發現在某些 case 中，seam carving 會讓背景變形，甚至會不小心消去某些物件(從 object removal 中可以看到)。

● References

- 教授與助教的講義
- Intro to seam carving [Project 2: Image Resizing by Seam Carving | CMU 15-463](#)
- Seam carving [Seam carving - Wikipedia](#)
- Object removal [Seam Carving \(brown.edu\)](#)