

DIP Final Proposal

組員：103062240 蔡宗宇、103062224 劉哲宇

1. Problem Description

Background

現今的影像處理工具，如 PhotoShop 非常盛行，照片可以很輕易的被剪貼、加工，因此，在任何管道所看見的圖片都有可能經過一些修改，甚至是偽造，我們希望可以透過研究一些已知的方法，來了解並延伸如何確認該圖片是否經過人為處理，更希望能知道經過了何種處理等等。

Previous Work

我們搜索了一些 `image forgery detection` 領域的相關論文，找到一篇研究[1]在分析經過 `interpolation resizing` 會出現的特性，且該特性會在 `frequency domain` 中被顯現出來。在另一篇論文中[2]，將此方法延伸到 `interpolation rotation` 的偵測，並且可以由從 `frequency domain` 上的偵測結果，回推圖片的旋轉角度或是放大倍率。

Motivation

在這幾篇論文中，除數學推導外的關鍵技術主要在 `Discrete Fourier Transform`，而這堂課對此方法已做過許多介紹，我們亦在作業中實作過，因此我們想要使用 Matlab 將論文中的結果重現，以驗證其方法的成效，並且，我們可以將這個方法延伸應用到 `Image Splicing Detection` 問題，根據[3]的說明：“`Image splicing involves replacing of image fragments from one or more different images on to another image`”，`Image Splicing` 是將其他圖片的一部分剪貼到一張圖片上，由於剪貼的過程中經常會使用旋轉、放大等需要 `interpolation` 的操作，因此我們可以先將圖片分割成數區塊，對各區塊套用此偵測，找出影像中哪部分可能有透過 `interpolation` 修改過的痕跡。

2. Methods

A. Assumptions

我們假設經過修改的圖片，是使用 `interpolation` 進行影像局部或整體的 `rotation` or `resizing`，例如透過 `bilinear` 內插法來做影像的 `resize`。

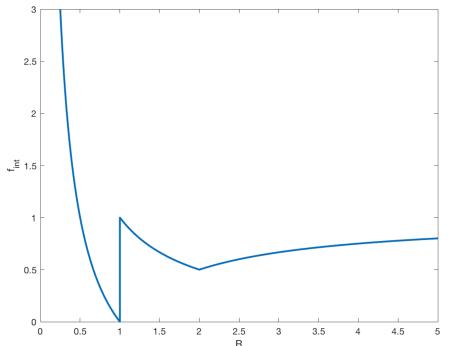
B. Mathematical Derivation

數學基本上由從[2]的結論來作延伸，從該篇論文的結論中我們可以知道：若一張圖片有經過 `interpolation` (`resize` or `rotate`)，則該圖片的 `frequency domain` 下的特定頻率會有異常的峰

值(peak value), 而峰值所發生的頻率與 interpolation 所使用的 parameter (例如放大倍率) 有直接關係, 因此我們可以藉由這些資訊, 從特定頻率回推 interpolation parameter, 對於 resizing 的 interpolation, 原論文給出了下面的關係式：

$$f_{int} = \begin{cases} 1 - \frac{1}{R}, & 1 < R \leq 2 \\ \frac{1}{R}, & R > 2 \end{cases}$$

其中 f_{int} 為峰值所發生的頻率。



透過上面所列的式子, 我們可以從已知 f_{int} 去回推可能的 R 。

而在[1]中, 我們可以利用類似方法回推旋轉時所使用的interpolation parameter, 旋轉的數學式子可以寫成：

$$\begin{cases} x' = x \cos \theta - y \sin \theta \\ y' = y \sin \theta + y \cos \theta \end{cases}$$

即旋轉矩陣, 單純考慮旋轉後的某一列(Row), 即我們固定 $y' = h$, 得到：

$$h = x \sin \theta + y \cos \theta$$

$$y = \frac{h - x \sin \theta}{\cos \theta}$$

代回得到：

$$\begin{aligned} x' &= x \cos \theta - \frac{h - x \sin \theta}{\cos \theta} \sin \theta \\ &= \frac{1}{\cos \theta} (x \cos^2 \theta - h \sin \theta + x \sin^2 \theta) \\ &= \frac{1}{\cos \theta} (x - h \sin \theta) \end{aligned}$$

可以觀察到, 在rotation過程中, 可以視作有factor $= \frac{1}{\cos \theta}$ 的resizing, 因此我們可以利用

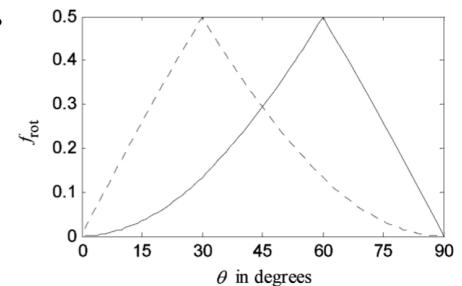
剛剛提到的方法, 即 R 在此為 $\frac{1}{\cos \theta}$, 則我們可以得到：

$$f_{int} = \begin{cases} 1 - \cos \theta, & 0^\circ < \theta \leq 60^\circ \\ \cos \theta, & 60^\circ < \theta < 90^\circ \end{cases}$$

透過相同的方法, 考慮旋轉後的某一行(Column), 我們可以得到：

$$f_{int} = \begin{cases} \sin \theta, & 0^\circ < \theta \leq 30^\circ \\ 1 - \sin \theta, & 30^\circ < \theta \leq 90^\circ \end{cases}$$

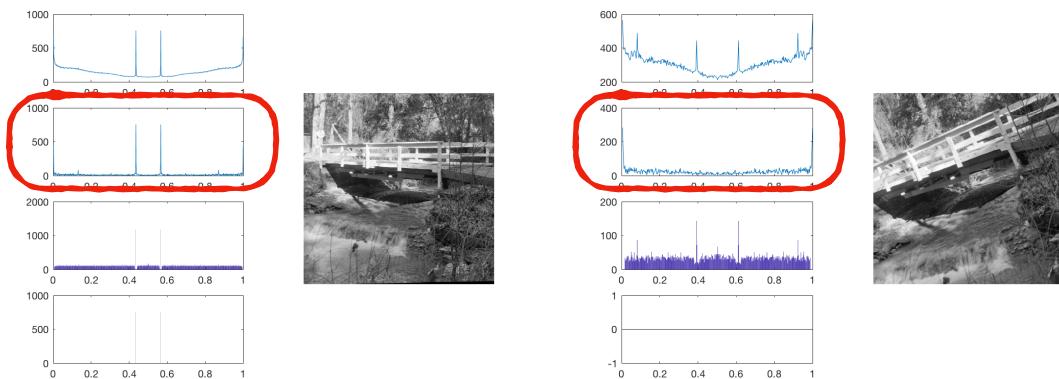
我們就可以透過 f_{int} 回推旋轉角度 θ 。



C. Limitations

雖然我們有了 interpolation parameter R 與峰值頻率 f_{int} 之間的關係，但因為從 R mapping 到 f_{int} 之間並非一對一的關係(one to one)，因此反函式並不存在，也導致回推過程中會產生 ambiguity。

另外，resize 與 rotate 經過推導後所得到的關係式是相同的（背後的原理相同），因此我們也需要能夠明確區分出該 interpolation 是因為 resizing 抑或是 rotation，在我們參考的資料[2]中有提出區分兩者的方法：



將所有 row 相加平均做 DFT 後，旋轉的情況下會沒有峰值，大致上的原因為：rotation 下 phase angle 會相當平均分配，使得峰值被抵消。

D. Algorithm Design (Interpolation Estimate)

1. Get an image with size $M \times N$

2. Apply 3x3 Laplacian filter

Use Matlab built-in Laplacian filter to get the second-order derivatives of image (edge map), denoted by E .

3. For each row of E , calculate its 1D-DFT

Denote row m of E by $v^{(m)}$, calculate their 1D-DFT $V^{(m)}$ using fft.

Note that in the following steps, we only use the left part of V (column 1~ $N/2$) because DFT is symmetric.

4. For DFT of all rows: peak counting

Initialize counter $c(n) = 0, \forall n \in [1, \frac{N}{2}]$.

For row $m = 1 \sim M$:

 For each frequency $n = 1 \sim N/2$:

 if $|V^{(m)}(n)| = \max_{i \in [n-\delta, n+\delta]} |V^{(m)}(i)|$, then $c^{(m)} := c^{(m)} + 1$

 End

End

Here we follow the paper to set $\delta = 5$, which is chosen empirically.

5. For DFT of all rows: peak detection

Normalize $c(n)$ such that $\sum_{n=1}^{N/2} c(n) = 1$

For each frequency $n = 1 \sim N/2$:

If $c(n) > T + \text{median}\{c(i), \forall i \in [n - W, n + W]\}$, record it as a pair $\{n, c(n)\}$

End

From all recorded pair, pick two with largest $c(n)$ and denote the one with smaller n as f_1 , and the other f_2 .

Here we follow the paper to pick $W = 5$ empirically but left T as a hyper-parameter to set by user.

6. For DFT of average row: peak detection

Along with step 3 and 4, we calculate the DFT of average of E along rows, i.e.

$DFT[\frac{1}{M} \sum_{m=1}^M E(m, n)]$, denoted by $c'(n)$

For each frequency $n = 1 \sim N/2$:

If $|c'(n)| > T + \text{median}\{|c'(i)|\}$ and $c'(n) = \max |c'(i)| \forall i \in [n - W, n + W]$, record it as a pair $\{n, c'(n)\}$

End

From all recorded pair, pick two with largest $c'(n)$ and denote the one with smaller n as f'_1 , and the other f'_2 .

7. Determine rotation or resizing by result of step 3, 4, 5

If no peak are found, we say the image is not interpolated, else if f_1, f_2, f'_1, f'_2 are matched, the image is resized. Else, the image is rotated.

8. Estimate its resizing/rotation factor by peak frequency

Estimate the interpolation factor by f_1, f_2 following the results of mathematical derivation.

E. Extended Application

We can extend our algorithm to perform the task of image splicing detection (interpolation is often in image splicing). If we can find out any region which is interpolated, we can find out whether any sub-region of an image is modified or not. Thus, we design the algorithm as follows:

F. Algorithm Design (Splicing Detection)

1. Divide image to overlapped blocks

Divide image with stride $L \times L$ into blocks with size $B \times B$, usually $L < B$, so there will be overlapped area between blocks.

2. For each block, perform Interpolation Estimation and get peak frequencies

For each block, perform the Algorithm mentioned in section D and get the f_1, f_2 if the block is marked as interpolated. Note the T in previous Algorithm will be calculated as $T = B \times T_{ratio}$, where user can specify the value of T_{ratio} .

3. Mark the blocks with populated peak frequencies as “suspicious”

From those blocks marked as interpolated, we collect their f_1, f_2 and make a statistics. For those blocks marked as interpolated and has f_1, f_2 being the most 2 frequencies, we mark them as “suspicious block”.

4. Remove isolated suspicious blocks

To reduce false alarm, we remove those blocks that have no suspicious neighborhoods.

3. Results

A. Verification of Mathematical Derivation

為了確認峰值存在，我們寫了個程式實際做旋轉、放大等操作，並將結果做成了影片：

旋轉 0~45 度：<https://drive.google.com/open?id=14JH2jygTplqKtPVm78WOA4EhbBs-BBLW>

放大 0.1~3.7 倍：https://drive.google.com/open?id=1ZVnE_LDUloefD4SjSJgCNOrkRFBkMz86

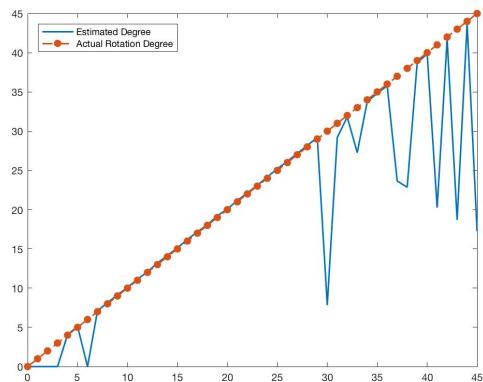
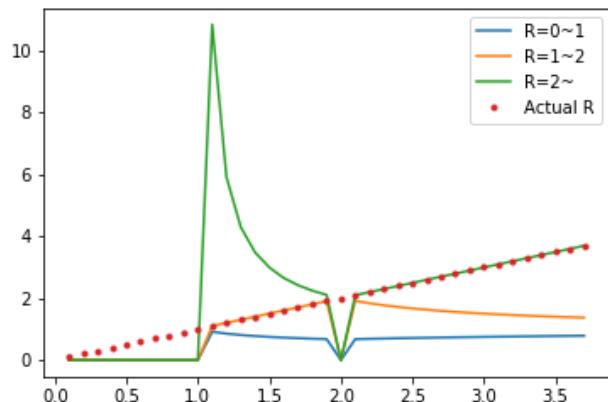
左邊四張圖分別代表：

- (a) Average of all rows' DFT
- (b) DFT of all rows' average,
- (c) Peak detection of a
- (d) Peak detection of b

右上方代表經過操作的圖，右下方則顯示了模型所預測的旋轉角度與放大倍率。

若右下方的值均為零，代表模型認為此圖 not interpolated。若旋轉角度第一個值為 NaN，

代表沒有偵測到 f_1 ，原因可能是峰值太小不易見，或是它被淹沒在自然影像的低頻區中，次時只需參考第二個值即可。若將程式的預測結果與實際使用的 factor 畫成圖表：



左：放大倍率與估計的圖表，實際放大倍率 R 為紅虛線，因 ambiguity，模型會估出三種值，需引入其他 prior knowlegde 才能決定要用何者。從圖中可見當 $R \in [0,1]$ 時，會判斷失敗，這點在論文[2] 中亦有提到： $R \in [0, \frac{2}{3}]$ 理論上找不到峰值、 $R \in (\frac{3}{2}, 1]$ 則很難有可觀測之峰值。在 $R > 1$ 情況下則預測結果就準確很多 (只看對應的值的話)，除了 $R = 2$ 等特例。

右：旋轉角度與估計的圖表，當旋轉角度小時我們的程式基本上沒有辦法判斷旋轉角度，而當度數越接近45度時判斷的準確率也有下降，其餘的度數判斷則相當接近真實的旋轉度數。

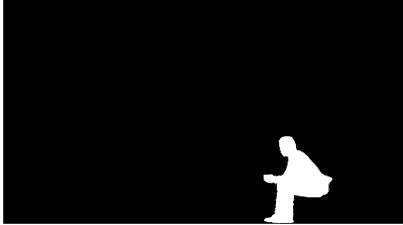
B. Extended Application (Splicing Detection)

為了數據化我們的結果，先定義每個 block 的 “correct label”：

If $\#pasted_pixels > 70\% \times B^2$, label of this block = 1, else 0.

也就是說，如果該 block 裡面有超過 70% 的 pixel 屬於被貼上的物件，則它應該要被 model 標示為 suspicious block。

Example 1

		
Spliced image	Splicing mask	Correct label
		
$B = 128, L = 32, T_{ratio} = 0.2$		$B = 64, L = 16, T_{ratio} = 0.28$
$Accuracy = 0.98$		$Accuracy = 0.971$
$Recall = 0.766$		$Recall = 0.349$
$Precision = 0.480$		$Precision = 0.348$

可以看出 block size 較小較可以找出比較細節的部分，但 false positive 也隨著增加了。

References

- [1] Wei, Weimin, et al. "Estimation of Image Rotation Angle Using Interpolation-Related Spectral Signatures With Application to Blind Detection of Image Forgery." *IEEE Transactions on Information Forensics and Security* 5.3 (2010): 507-517.
- [2] Li, Gouhui, et al. "A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD." *Multimedia and Expo, 2007 IEEE International Conference on IEEE 2007*.