

# 國立交通大學

## 電機與控制工程研究所

### 碩士論文

數位照片手振還原



Restoration of Digital Photo with Motion Blur Caused by Hand

研究生：蘇群智

指導教授：李福進 博士

中華民國九十六年十一月

數位照片手振還原

Restoration Of Digital Photo With Motion Blur Caused By Hand

研 究 生：蘇群智

**Student : Chiun-Jr Su**

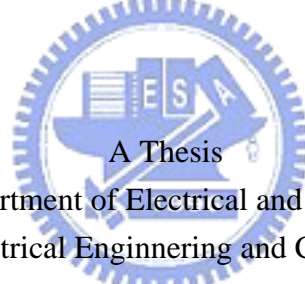
指導教授：李福進

**Advisor : Fu-Ching Lee**

國立交通大學

電機與控制工程研究所

碩士論文



A Thesis  
Submitted to Department of Electrical and Control Engineering  
College of Electrical Engineering and Computer Science  
National Chiao-Tung University  
in Partial Fulfillment of the Requirments  
for the Degree of Master  
in

Electrical and Control Engineering

November 2007

Hsinchu , Taiwan , Republic of China

中華民國九十六年十一月

# 數位照片手振還原

研究生：蘇群智

指導教授：李福進 博士

國立交通大學電機與控制工程學系碩士班

## 摘要

本論文的研究主題在於對現實發生的手振模糊照片，提供一個近似實際手振模糊的影像退化函數建立方案，以此為基礎設計還原濾波器，並且模擬各種現實上可能發生降低影像還原效果的不理想效應，討論其對影像還原的影響。並以實際數位像機拍攝到的手振模糊照片進行還原，以驗證此方案的實際效果。

# Restoration Of Digital Photo With Motion Blur Caused By Hand

Student : Chiun-Jr Su

Advisor : Fu-Ching Lee

Institute of Electrical and Control Engineering

National Chiao-Tung University



The purpose of this thesis is to provide a method to construct a degraded function approaching really motion blur caused by hand, take the function be basis of restoration filter design, and simulate in many situations that would reduce the results of restoration, discuss the effect to image restoration. Furthermore, restore the photo with motion blur caused by hand taken by digital camera to test and verify the result of the method.

## 誌 謝

這篇論文得以完成，首先要感謝我的指導教授李福進老師，在研究所的段時間以來，在理論和實作方面，不斷的給我寶貴的指導和建議，給予我相當多的幫助，在這裡表達十分的感謝。

另外，在這段期間中，有賴張駿嘉同學的互相幫助和提攜，和我度過這段研究生生活。還有在這段時間裡支持著我的父母，還有許許多多幫助過我的朋友們，非常謝謝你們。



# 目錄

中文摘要	iii
英文摘要	iv
誌謝	v
目錄	vi
圖目錄	viii
<b>第一章 序論</b>	<b>1</b>
1.1 背景	1
1.2 論文目的	1
1.3 論文架構	2
<b>第二章 手振模糊影像的建構</b>	<b>3</b>
<b>第三章 手振模糊影像的還原理論</b>	
3.1 數位世界的手振模糊影像的構成	6
3.2 手振模糊影像的還原理論	8
3.2.1 加入無效限制範圍的還原濾波	9
3.2.2 Wiener filter (minimum mean square error filter) 還原濾波	9
3.3 結論	11
<b>第四章 模擬手振模糊影像還原</b>	<b>12</b>
4.1 模擬手振模糊影像生成	12
4.2 無雜訊的手振模擬影像還原	15
4.3 加入四捨五入造成的雜訊的還原	16
4.4 加入額外雜訊的還原	22
4.4.1 加入強度 1 均勻雜訊的影像還原	23
4.4.2 加入強度 10 均勻雜訊的影像還原	24
4.5 有誤差的手振模型還原	25
4.5.1 有估測誤差的水平手振模糊還原	25
4.5.2 有估測誤差的 45 度手振模糊還原	30
4.5.3 有估測誤差的手振模糊還原	34

4.6	邊界效應 .....	37
4.6.1	去邊界後的模糊影像還原 例一 .....	38
4.6.2	去邊界後的模糊影像還原 例二 .....	42
4.7	結論 .....	49
<b>第五章 實際手振模糊照片還原</b>		<b>50</b>
5.1	移動被攝體造成手振模糊效果之還原.....	50
5.1.1	水平移動還原 .....	50
5.1.2	45 度移動還原.....	53
5.1.3	結論 .....	55
5.2	實際發生的手振模糊照片還原實例.....	55
5.2.1	實際發生的手振模糊照片還原 例一 .....	56
5.2.2	實際發生的手振模糊照片還原 例二 .....	60
5.2.3	結論 .....	63
<b>參考資料</b>		<b>64</b>
<b>附錄 --- 程式碼</b>		<b>65</b>
<b>簡歷</b>		<b>76</b>



# 圖目錄

圖 2.1	手振模糊示意圖	3
圖 4.1.1	手振模糊模擬使用的原始圖片	12
圖 4.1.2	水平手振模糊模擬	13
圖 4.1.3	手振模糊模型(水平)	13
圖 4.1.4	手振模糊模擬(22.5 度)	13
圖 4.1.5	手振模糊模型(22.5 度)	13
圖 4.1.6	手振模糊模擬(45 度)	14
圖 4.1.7	手振模糊模型(45 度)	14
圖 4.1.8	手振模糊模擬(曲線)	14
圖 4.1.9	手振模糊模型(曲線)	14
圖 4.2.1	無雜訊手振模擬影像還原( $K=10^{-3}$ )	15
圖 4.2.2	無雜訊手振模擬影像還原( $K=10^{-4}$ )	15
圖 4.2.3	無雜訊手振模擬影像還原( $K=0$ )	16
圖 4.3.1	在空間域下的四捨五入雜訊	17
圖 4.3.2	在頻率域下四捨五入雜訊的 magnitude	17
圖 4.3.3	四捨五入雜訊手振模擬影像還原(水平 $K=10^{-3}$ )	18
圖 4.3.4	四捨五入雜訊手振模擬影像還原(水平 $K=10^{-4}$ )	18
圖 4.3.5	四捨五入雜訊手振模擬影像還原(水平 最小 E 值)	18
圖 4.3.6	四捨五入雜訊手振模擬影像還原(水平 標準 Wiener filter)	18
圖 4.3.7	四捨五入雜訊手振模擬影像還原(22.5 度 $K=10^{-3}$ )	19
圖 4.3.8	四捨五入雜訊手振模擬影像還原(22.5 度 $K=10^{-4}$ )	19
圖 4.3.9	四捨五入雜訊手振模擬影像還原(22.5 度 最小 E 值)	19
圖 4.3.10	四捨五入雜訊手振模擬影像還原(22.5 度 標準 Wiener filter)	19
圖 4.3.11	四捨五入雜訊手振模擬影像還原(45 度 $K=10^{-3}$ )	20
圖 4.3.12	四捨五入雜訊手振模擬影像還原(45 度 $K=10^{-4}$ )	20
圖 4.3.13	四捨五入雜訊手振模擬影像還原(45 度 最小 E 值)	20
圖 4.3.14	四捨五入雜訊手振模擬影像還原(45 度 標準 Wiener filter)	20
圖 4.3.15	四捨五入雜訊手振模擬影像還原(曲線 $K=10^{-3}$ )	21
圖 4.3.16	四捨五入雜訊手振模擬影像還原(曲線 $K=10^{-4}$ )	21
圖 4.3.17	四捨五入雜訊手振模擬影像還原(曲線 最小 E 值)	21



圖 4.3.18	四捨五入雜訊手振模擬影像還原(曲線 標準 Wiener filter)	21
圖 4.4.1	在空間域下的均勻雜訊	22
圖 4.4.2	均勻雜訊頻譜的 magnitude	22
圖 4.4.3	加入強度 1 均勻雜訊的模糊影像	22
圖 4.4.4	加入強度 1 均勻雜訊的手振模擬影像還原( $K=10^{-3}$ )	23
圖 4.4.5	加入強度 1 均勻雜訊的手振模擬影像還原( $K=10^{-4}$ )	23
圖 4.4.6	加入強度 1 均勻雜訊的手振模擬影像還原(最小 E 值)	23
圖 4.4.7	加入強度 1 均勻雜訊的手振模擬影像還原(標準 Wiener filter)	23
圖 4.4.8	加入強度 10 均勻雜訊的模糊影像	24
圖 4.4.9	加入強度 10 均勻雜訊的手振模擬影像還原( $K=10^{-2}$ )	24
圖 4.4.10	加入強度 10 均勻雜訊的手振模擬影像還原( $K=10^{-3}$ )	24
圖 4.4.11	加入強度 10 均勻雜訊的手振模擬影像還原(最小 E 值)	25
圖 4.4.12	加入強度 10 均勻雜訊的手振模擬影像還原(標準 Wiener filter)	25
圖 4.5.1	使用正確的手振模糊模型的還原(水平 $K=10^{-2}$ )	26
圖 4.5.2	使用正確的手振模糊模型的還原(水平 $K=10^{-3}$ )	26
圖 4.5.3	使用正確的手振模糊模型的還原(水平 $K=0$ )	27
圖 4.5.4	建立模糊影像使用模型(水平 1:16pixel)	27
圖 4.5.5	還原時估測的手振模型(水平 1:17pixel)	27
圖 4.5.6	有估測誤差模糊模型的還原(水平 16pixel $\rightarrow$ 17pixel $K=10^{-2}$ )	28
圖 4.5.7	有估測誤差模糊模型的還原(水平 16pixel $\rightarrow$ 17pixel $K=10^{-3}$ )	28
圖 4.5.8	有估測誤差模糊模型的還原(水平 16pixel $\rightarrow$ 17pixel 最小 E 值)	28
圖 4.5.9	有估測誤差模糊模型的還原(水平 16 $\rightarrow$ 17 標準 Wiener filter)	28
圖 4.5.10	建立模糊影像使用模型(水平 1:16pixel)	29
圖 4.5.11	還原時估測的手振模型(水平 1:15pixel)	29
圖 4.5.12	有估測誤差模糊模型的還原(水平 16pixel $\rightarrow$ 15pixel $K=10^{-2}$ )	29
圖 4.5.13	有估測誤差模糊模型的還原(水平 16pixel $\rightarrow$ 15pixel $K=10^{-3}$ )	29
圖 4.5.14	有估測誤差模糊模型的還原(水平 16pixel $\rightarrow$ 15pixel 最小 E 值)	30
圖 4.5.15	有估測誤差模糊模型的還原(水平 16 $\rightarrow$ 15 標準 Wiener filter)	30
圖 4.5.16	使用正確的手振模糊模型的還原(45 度 $K=10^{-2}$ )	30
圖 4.5.17	使用正確的手振模糊模型的還原(45 度 $K=10^{-3}$ )	30
圖 4.5.18	使用正確的手振模糊模型的還原(45 度 $K=0$ )	31
圖 4.5.19	建立模糊影像使用模型(45 度 1:16pixel)	31
圖 4.5.20	還原時估測的手振模型(45 度 1:17pixel)	31
圖 4.5.21	有估測誤差模糊模型的還原(45 度 16pixel $\rightarrow$ 17pixel $K=10^{-2}$ )	32
圖 4.5.22	有估測誤差模糊模型的還原(45 度 16pixel $\rightarrow$ 17pixel $K=10^{-3}$ )	32
圖 4.5.23	有估測誤差模糊模型的還原(45 度 16pixel $\rightarrow$ 17pixel 最小 E 值)	32
圖 4.5.24	有估測誤差模糊模型的還原(45 度 16 $\rightarrow$ 17 標準 Wiener filter)	32
圖 4.5.25	建立模糊影像使用模型(45 度 1:16pixel)	33
圖 4.5.26	還原時估測的手振模型(45 度 1:15pixel)	33

圖 4.5.27	有估測誤差模糊模型的還原(45 度 16pixel→15pixel $K=10^{-2}$ ) -----	33
圖 4.5.28	有估測誤差模糊模型的還原(45 度 16pixel→15pixel $K=10^{-3}$ ) -----	33
圖 4.5.29	有估測誤差模糊模型的還原(45 度 16pixel→15pixel 最小 E 值) -----	34
圖 4.5.30	有估測誤差模糊模型的還原(45 度 16→15 標準 Wiener filter) -----	34
圖 4.5.31	手振模糊模擬影像 -----	34
圖 4.5.32	建立模糊影像使用模型 -----	34
圖 4.5.33	還原時估測的手振模型 -----	35
圖 4.5.34	無估測誤差模糊模型的還原( $K=10^{-2}$ ) -----	35
圖 4.5.35	無估測誤差模糊模型的還原( $K=10^{-3}$ ) -----	35
圖 4.5.36	無估測誤差模糊模型的還原( $K=0$ ) -----	36
圖 4.5.37	有估測誤差模糊模型的還原( $K=10^{-2}$ ) -----	36
圖 4.5.38	有估測誤差模糊模型的還原( $K=10^{-3}$ ) -----	36
圖 4.5.39	有估測誤差模糊模型的還原(最小 E 值) -----	37
圖 4.5.40	有估測誤差模糊模型的還原(標準 Wiener filter) -----	37
圖 4.6.1	手振模糊模型(水平) -----	38
圖 4.6.2	水平手振模糊模擬 -----	39
圖 4.6.3	去掉現實狀況所沒有的邊界後的水平手振模糊模擬影像 -----	39
圖 4.6.4	去掉邊界後的模糊影像的還原( $K=10^{-2}$ ) -----	39
圖 4.6.5	去掉邊界後的模糊影像的還原( $K=10^{-3}$ ) -----	39
圖 4.6.6	外加四捨五入雜訊的去邊界模糊影像還原( $K=10^{-2}$ )-----	40
圖 4.6.7	外加四捨五入雜訊的去邊界模糊影像還原( $K=10^{-3}$ )-----	40
圖 4.6.8	原始影像-----	40
圖 4.6.9	水平手振模糊模擬 -----	41
圖 4.6.10	去掉現實狀況所沒有的邊界後的水平手振模糊模擬影像 -----	41
圖 4.6.11	去掉邊界後的模糊影像的還原( $K=10^{-2}$ ) -----	41
圖 4.6.12	去掉邊界後的模糊影像的還原( $K=10^{-3}$ ) -----	41
圖 4.6.13	外加四捨五入雜訊的去邊界模糊影像還原( $K=10^{-2}$ )-----	42
圖 4.6.14	外加四捨五入雜訊的去邊界模糊影像還原( $K=10^{-3}$ )-----	42
圖 4.6.15	原始影像(四百萬畫素) -----	43
圖 4.6.16	水平手振模糊模擬(四百萬畫素) -----	44
圖 4.6.17	去掉現實狀況所沒有的邊界後的影像(四百萬畫素) -----	45
圖 4.6.18	去掉邊界後的模糊影像的還原(四百萬畫素 $K=10^{-2}$ ) -----	46
圖 4.6.19	去掉邊界後的模糊影像的還原(四百萬畫素 $K=10^{-3}$ ) -----	47
圖 4.6.20	外加四捨五入雜訊的去邊界模糊影像還原(四百萬畫素 $K=10^{-3}$ ) ---	48
圖 5.1.1	水平移動照片 -----	51
圖 5.1.2	水平移動照片還原結果 ( $K=10^{-1.5}$ )-----	52
圖 5.1.3	45 度移動照片 -----	53

圖 5.1.4	45 度移動照片還原結果 ( $K=10^{-1.5}$ ) -----	54
圖 5.1.5	水平移動照片預測手振模糊模型 -----	55
圖 5.1.6	45 度移動照片預測手振模糊模型 -----	55
圖 5.2.1	實際發生的手振模糊照片的預測手振模糊模型例一 -----	56
圖 5.2.2	實際發生的手振模糊原始照片例一 -----	57
圖 5.2.3	例一的還原結果( $K=10^{-1.5}$ ) -----	58
圖 5.2.4	例一局部#1 -----	59
圖 5.2.5	例一還原結果局部#1 -----	59
圖 5.2.6	例一局部#2 -----	59
圖 5.2.7	例一還原結果局部#2 -----	59
圖 5.2.8	實際發生的手振模糊照片的預測手振模糊模型例二 -----	60
圖 5.2.9	實際發生的手振模糊原始照片例二 -----	61
圖 5.2.10	例二的還原結果( $K=10^{-1.5}$ ) -----	62



# 第一章 序論

## 1.1 背景

由於電子科技方面的進步，最近幾年來數位照片已經漸漸取代傳統銀鹽底片的類比照片成爲主流，同時，對於照片的後製以及種種照片的劣化(如模糊失焦,色差,雜訊,手振模糊)的復原處理，數位照片也相對的來的方便且靈活度高，一般人在自己的家裡使用自己的電腦就可完成。

可以用來擷取數位照片的工具，也越來越多樣化，小型化，輕量化，附加在其他電子產品上，一般人取得照片也越來越方便。但是拍照的時候由於環境，人爲技術，以及機器本身的因素，造成取得的照片產生劣化的現象。拍照時因爲握著相機的手的震動使得相片產生模糊，稱爲手振模糊，就是劣化現象的其中一種，對於照片的破壞性相當的大。



## 1.2 論文目的

對於防止產生手振模糊，現在許多市售的機器已經搭載了光學防手振的系統，利用物理補償的方式使得攝入感光元件的影像儘量維持固定不動，如此可減小因手振對照片造成的影響。但是，對於超越光學防手振系統能解決的過度手振，或是沒搭載光學防手振系統的機器，當手振情況出現在已經紀錄下來的數位照片上，這時候就只能用軟體的方法來解決。

所以，本篇論文的目的是使用數位影像處理的方法，來解決已經發生手振情形的數位照片的手振問題。

### 1.3 論文架構

本篇論文的第 2, 3 章的部分是基礎理論架構的描述, 第 4 章的部分是模擬手振還原的實驗, 第 5 章是實際的手振模糊照片的還原。



## 第二章 手振模糊影像的建構

拍照時，手持相機的使用者按下快門之後，快門會持續一段時間開啓，在這段開啓的時間內被拍攝的景物的光線透過鏡頭，投影在相機內部的感光元件上，快門必須持續的開啓一段讓感光元件攝入足夠的光的時間，這是能拍出成功照片的必要條件。但如果攝影的環境過暗的話，快門開放的時間將相對的拉長，這時候手振的效果將會顯現出來。

在本文中將暫時不考慮鏡頭等因素造成的非理想性，把整個拍照視為一個固定大小的取景框在一個很大的畫面上抓取資訊，如下示意圖：

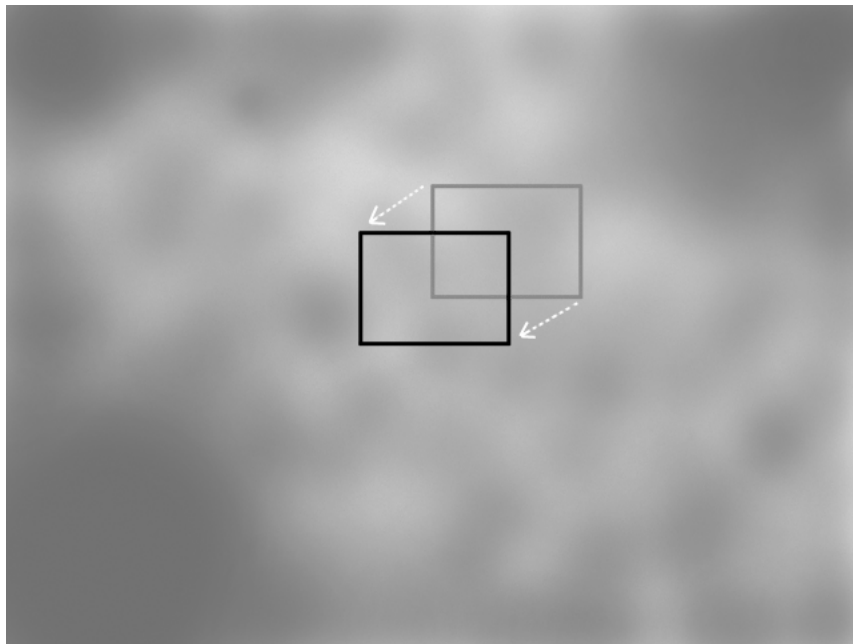


圖 2.1

在快門開放的時間內，如果沒有手振的情形，則我們得到的影像如下表示：

$$\begin{aligned} f(x, y) &= \int_{\frac{-T}{2}}^{\frac{T}{2}} f'(x, y) dt \\ &= f'(x, y)T \end{aligned} \quad (2.1)$$

其中  $f$  是我們取得的未退化影像， $T$  是快門開放時間

相對的，如果手振的情形發生了，這時候影像的表示式變成：

$$g(x, y) = \int_{-\frac{T}{2}}^{\frac{T}{2}} f'(x-\alpha(t), y-\beta(t)) dt \quad (2.2)$$

其中， $g$  表示手振模糊影像， $f'$  為單位時間內累積的影像強度值，而  $\alpha, \beta$  為時間函數，分別代表在影像中  $x, y$  方向的位移。在照相的概念中，影像  $f$  是在一段連續的時間內，累積  $f'$  而形成的。

接下來的部分，我們想要對手振模糊影像進行解析，首先對(2.2)式的左右兩邊做傅立葉轉換，結果如下：

$$\begin{aligned} G(u, v) &= \iint \left( \int_{-\frac{T}{2}}^{\frac{T}{2}} f'(x-\alpha(t), y-\beta(t)) dt \right) \exp\{-2j\pi(ux + vy)\} dx dy \\ &= \int_{-\frac{T}{2}}^{\frac{T}{2}} \left( \iint f'(x-\alpha(t), y-\beta(t)) \exp\{-2j\pi(ux+vy)\} dx dy \right) dt \end{aligned}$$

令  $a = x - \alpha(t)$ ,  $b = y - \beta(t)$

$$\begin{aligned} G(u, v) &= \int_{-\frac{T}{2}}^{\frac{T}{2}} \left( \iint f'(a, b) \exp\{-2j\pi(u(a+\alpha(t))+v(b+\beta(t)))\} da db \right) dt \\ &= \int_{-\frac{T}{2}}^{\frac{T}{2}} \left( \iint f'(a, b) \exp\{-2j\pi(ua+vb)\} \exp\{-2j\pi(u\alpha(t)+v\beta(t))\} da db \right) dt \\ &= \int_{-\frac{T}{2}}^{\frac{T}{2}} \exp\{-2j\pi(u\alpha(t)+v\beta(t))\} dt \times F'(u, v) \\ &= \int_{-\frac{T}{2}}^{\frac{T}{2}} \exp\{-2j\pi(u\alpha(t)+v\beta(t))\} dt \times F(u, v) \frac{1}{T} \end{aligned}$$

其中  $G$  為  $g$  的傅立葉轉換結果,  $F'$  為  $f'$  的傅立葉轉換結果

如此可求得退化函數  $H$  :

$$H(u, v) = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \exp\{-2j\pi(u\alpha(t)+v\beta(t))\} dt \quad (2.3)$$

如果手振為一個等速直線移動(速度為  $V$ , 移動方向的角度為  $\theta$ )的過程, 則:

$$\alpha(t) = V \cos \theta \times t, \quad \beta(t) = V \sin \theta \times t$$

$$\begin{aligned} H(u, v) &= \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \exp\{-2j\pi(V \cos \theta \times tu + V \sin \theta \times tv)\} dt \\ &= \frac{\sin \pi V (u \cos \theta + v \sin \theta)}{\pi V (u \cos \theta + v \sin \theta)} \\ &= \text{sinc}\{\pi V (u \cos \theta + v \sin \theta)\} \end{aligned} \quad [4] \quad (2.4)$$

以上為類比世界中的手振模糊影像的建構, 但是這個部分對於數位影像的處理是難以使用的。所以, 我們必須思考在數位影像處理的解決方法。



## 第三章 手振模糊影像的還原理論

### 3.1 數位世界的手振模糊影像的構成

雖然，造成數位影像手振模糊的現象，是現實世界中光線透過感光元件轉換成訊號，隨著時間累積而成。但是，成為數位訊號以後將會流失取樣的點以外訊號的資訊，我們所能取得的資訊只有成為數位照片的圖片檔的部分，我們所能利用的資訊也只有這張圖片檔而已。

所以，在這篇論文中我使用的方法是把手振造成影像模糊的整個過程以數位化的方式處理。考慮大小為  $M \times N$  的數位影像，假設在沒有雜訊的情況下，手振模糊的影像和未退化影像的關係式如下：

$$g(x, y) = a_0 f(x, y) + a_1 f(x - x_1, y - y_1) + a_2 f(x - x_2, y - y_2) + \dots + a_{n-1} f(x - x_{n-1}, y - y_{n-1}) \quad (3.1)$$

其中  $g$  表示手振模糊影像， $f$  表示未退化影像，在還原手振照片時，我們能取得的是  $g$ ， $f$  則是要的結果，其中：

$$a_0 \geq 0, a_1 \geq 0, \dots, a_{n-1} \geq 0$$

$$\sum_{k=0}^{n-1} a_k = 1$$

建立這個關係式的構想是，手振模糊影像是由未退化影像移動位置後加權累加而成。也就是手振模糊影像的其中的任意一點，是由(3.1)式中未退化影像相對位置的這些點加權累加而成。

此外， $g$  也可寫成以下的表示式：

$$g(x, y) = a_0 f(x, y) + \sum_{k=1}^{n-1} a_k f(x - x_k, y - y_k)$$

$$= \sum_{k=0}^{n-1} a_k f(x - x_k, y - y_k), \quad \text{其中 } x_0 = 0, y_0 = 0 \quad (3.2)$$

爲了求得  $f$ ，考慮(3.1)式在頻域(frequency domain)上做處理，已知：

$$f(x, y) \Leftrightarrow F(u, v)$$

$$f(x - x_k, y - y_k) \Leftrightarrow F(u, v) \cdot e^{-j2\pi(\frac{ux_k}{M} + \frac{vy_k}{N})}$$

其中“ $\Leftrightarrow$ ”是表示 2D DFT 的正反運算， $M$  和  $N$  分別表示影像的高度和寬度 (單位是 pixel)，又：

$$g(x, y) \Leftrightarrow G(u, v)$$



所以：

$$G(u, v) = a_0 F(u, v) + a_1 F(u, v) e^{-j2\pi(\frac{ux_1}{M} + \frac{vy_1}{N})} + a_2 F(u, v) e^{-j2\pi(\frac{ux_2}{M} + \frac{vy_2}{N})}$$

$$+ \dots + a_{n-1} F(u, v) e^{-j2\pi(\frac{ux_{n-1}}{M} + \frac{vy_{n-1}}{N})}$$

$$= F(u, v) \cdot (a_0 + a_1 e^{-j2\pi(\frac{ux_1}{M} + \frac{vy_1}{N})} + \dots + a_{n-1} e^{-j2\pi(\frac{ux_{n-1}}{M} + \frac{vy_{n-1}}{N})})$$

$$= F(u, v) \cdot (a_0 + \sum_{k=1}^{n-1} a_k e^{-j2\pi(\frac{ux_k}{M} + \frac{vy_k}{N})})$$

$$\begin{aligned}
&= F(u, v) \cdot \left( \sum_{k=0}^{n-1} a_k e^{-j2\pi \left( \frac{ux_k}{M} + \frac{vy_k}{N} \right)} \right) \\
&= F(u, v) \cdot H(u, v)
\end{aligned} \tag{3.3}$$

其中:

$$H(u, v) = \sum_{k=0}^{n-1} a_k e^{-j2\pi \left( \frac{ux_k}{M} + \frac{vy_k}{N} \right)}$$

所以, 手振模糊的程序可被模式化成一個退化函數。

實際上取得的照片的情況, 考慮到各種不理想因素, 以及數位化的種種誤差, (3.3)可修正為:

$$G(u, v) = H(u, v) \cdot F(u, v) + N(u, v) \tag{3.4}$$

在空間域上表示為:

$$g(x, y) = h(x, y) * f(x, y) + n(x, y) \tag{3.5}$$

在這裡  $h$  為在空間域中的退化函數,  $n$  為雜訊, 且

$$\begin{aligned}
&\overset{2Ddft}{h(x, y)} \Leftrightarrow H(u, v), \quad \overset{2Ddft}{n(x, y)} \Leftrightarrow N(u, v)
\end{aligned}$$

### 3.2 手振模糊影像的還原理論

根據(3.3), 求得  $f$  是一件單純的事情:

$$F(u, v) = \frac{1}{H(u, v)} G(u, v) = B(u, v) \cdot G(u, v)$$

其中  $B(u, v) = \frac{1}{H(u, v)}$  : 還原濾波器

但是, 實際上誤差和雜訊不可能不存在, 如(3.4)式的表示, 我們仍然用同樣的

方法還原的話:

$$\hat{F}(u, v) = B(u, v) \cdot G(u, v) = F(u, v) + \frac{N(u, v)}{H(u, v)}$$

在某些  $H(u, v)$  相當小的情況下,  $\frac{N(u, v)}{H(u, v)}$  將會相當的大, 而輕易的主宰

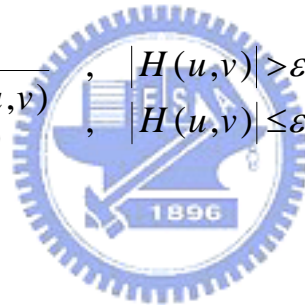
$F(u, v)$  的估測。

所以, 爲了迴避這種情況, 以下考慮兩種還原濾波的方法:

### 3.2.1 加入無效限制範圍的還原濾波

對於使反率波無效化的  $H(u, v)$ , 在某些  $\varepsilon$  的範圍內加以限制:

$$B(u, v) = \begin{cases} \frac{1}{H(u, v)} & , |H(u, v)| > \varepsilon \\ 0 & , |H(u, v)| \leq \varepsilon \end{cases}$$



### 3.2.2 Wiener filter (minimum mean square error filter) 還原濾波[1]

Wiener filter 尋求估測出  $\hat{f}$  使以下統計誤差最小化:

$$e^2 = E\{(f - \hat{f})^2\}$$

其中  $E$  是期望值運算子,  $f$  是未退化影像,  $\hat{f}$  是估測復原影像, 其表示式如下:

$$\begin{aligned}
\hat{F}(u, v) &= \left[ \frac{H^*(u, v) S_f(u, v)}{S_f(u, v) |H(u, v)|^2 + S_\eta(u, v)} \right] G(u, v) \\
&= \left[ \frac{H^*(u, v)}{|H(u, v)|^2 + S_\eta(u, v) / S_f(u, v)} \right] G(u, v) \\
&= \left[ \frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + S_\eta(u, v) / S_f(u, v)} \right] G(u, v)
\end{aligned} \tag{3.6}$$

$H(u, v)$  = 退化函數

$H^*(u, v)$  =  $H(u, v)$  的共軛複數

$S_\eta(u, v) = |N(u, v)|^2$  = 雜訊的功率頻譜

$S_f(u, v) = |F(u, v)|^2$  = 未退化影像的功率頻譜

註: 這個部分的內容出自於:

[1] Digital image processing, 2<sup>nd</sup> ed. 作者: Rafael C. Gonzalez, Richard E. Wood, 證明的部分請參考:

[3] Image Processing 作者: Maria Petrou, Panagiota Bosdogianni

在大部分的情況下,  $S_\eta(u, v)$  和  $S_f(u, v)$  是無法取得的, 所以在這種情況下有一個常用的近似式如下:

$$\begin{aligned}
\hat{F}(u, v) &= \left[ \frac{H^*(u, v)}{|H(u, v)|^2 + K} \right] G(u, v) \\
&= \left[ \frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + K} \right] G(u, v)
\end{aligned} \tag{3.7}$$

其中  $K$  是一個指定的常數

### 3.3 結論

根據實驗結果,“Wiener filter 反濾波”的方法在各方面較加”入無效限制範圍的反濾波法”來的優秀,所以本論文採用的方法就是“Wiener filter 反濾波法”。



## 第四章 模擬手振模糊影像還原

### 4.1 模擬手振模糊影像生成

本研究所模擬的部分使用的圖片檔是 $512 \times 512$  pixel 的尺寸, 256 階調的灰階圖片, 使用的圖片如以下表示:



圖 4.1.1

以下 4 組圖, 是使用上一章(3.3)式模擬出來的手振模糊情形, 左邊的圖是模擬結果, 右邊是手振的模型, 也就是模糊的照片中的, 一點是由相鄰的這些點加權相加而成, 其中 a~c 的每一點的加權係數皆相同, 只有 d 有些變化, 詳細的係數紀錄在 d 的右圖上

a.



圖 4.1.2

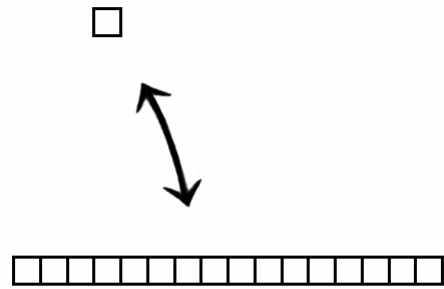


圖 4.1.3 手振模糊模型(水平)

b.



圖 4.1.4

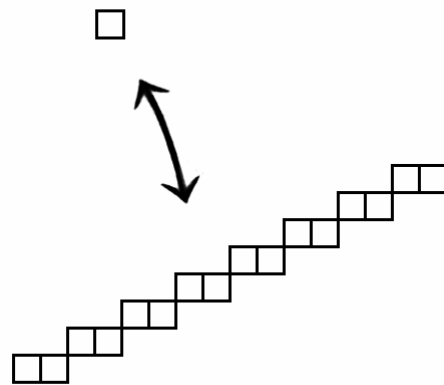


圖 4.1.5 手振模糊模型(22.5 度)



c.



圖 4.1.6

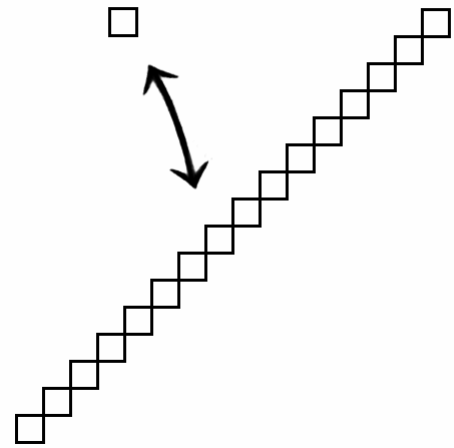


圖 4.1.7 手振模糊模型(45 度)

d.



圖 4.1.8

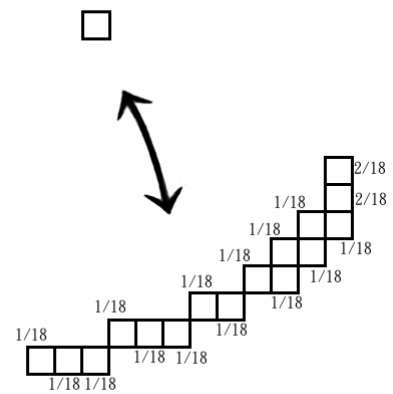


圖 4.1.9 手振模糊模型(曲線)



## 4.2 無雜訊的手振模擬影像還原

接下來，我們要對產生的模擬手振模糊影像進行還原，這裡選擇上一節的  $a$  作為樣本，測試在沒有雜訊下的還原結果：

這裡使用前章描述過的 Wiener filter 還原如下所示：

$$\hat{F}(u, v) = \left[ \frac{H^*(u, v)}{|H(u, v)|^2 + K} \right] G(u, v)$$

其中  $K$  是常數，此外這裡使用  $e^2 = E\{(f - \hat{f})^2\}$  統計誤差，用來評估實際的還原效果，為了方便表示，在這裡令  $E = e^2$

圖 4.2.1



$$K = 10^{-3}, E = 5.1513 \times 10^6$$

圖 4.2.2



$$K = 10^{-4}, E = 2.2954 \times 10^6$$

圖 4.2.3



$$K = 0, E = 7.4972 \times 10^{-23} \cong 0$$

由模擬的結果可以得知，當  $K$  值越小的時候還原的效果越好，當  $K=0$  的時候可以完全的還原為原來的照片，在這邊因為 MATLAB 計算誤差的關係  $E$  值並沒有完全歸 0, Wiener filter

的行為相當於直接反濾波： $B = \frac{1}{H}$

當  $K$  值不為 0 的情況下，還原的影像出現了沿著手振震動的方向，重複影像中邊界部分的邊界形狀的干擾。

### 4.3 加入四捨五入造成的雜訊的還原

4.1 的手振模糊照片的模擬，是使用：

$$G = H \cdot F \Leftrightarrow g = h * f$$

透過在頻域中的模糊函數  $H$  與原始影像的  $F$  相乘，得到手振模糊影像  $G$ ，轉換到空間領域上為  $g$ 。為了留下並且秀出處理的結果，我們必須把  $g$  做四捨五入到整數位，使其符合 8-bit,

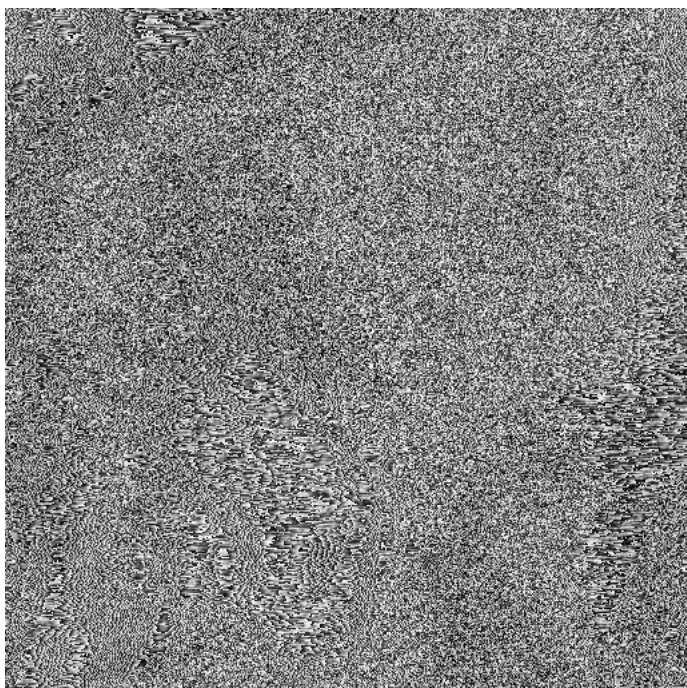
0~255 整數灰階值的數位圖片表示，這一點和我們使用數位像機攝影完成時，最後寫成數位相片的概念類似。這個部分在程式上的做法是，先把模糊運算完成的雙倍浮點矩陣轉成 8 bit 的圖片檔儲存，還原的時候讀取這個圖檔作為還原的影像來源(G)。這個部分對於手振模糊模擬的運算而言，相當於加入一個範圍在  $\pm 0.5$  的雜訊，當然在實際的情況下當然是無法得知這種雜訊的值的。

在這種情況下修正為：

$$G = H \cdot F + N \Leftrightarrow g = h * f + n$$

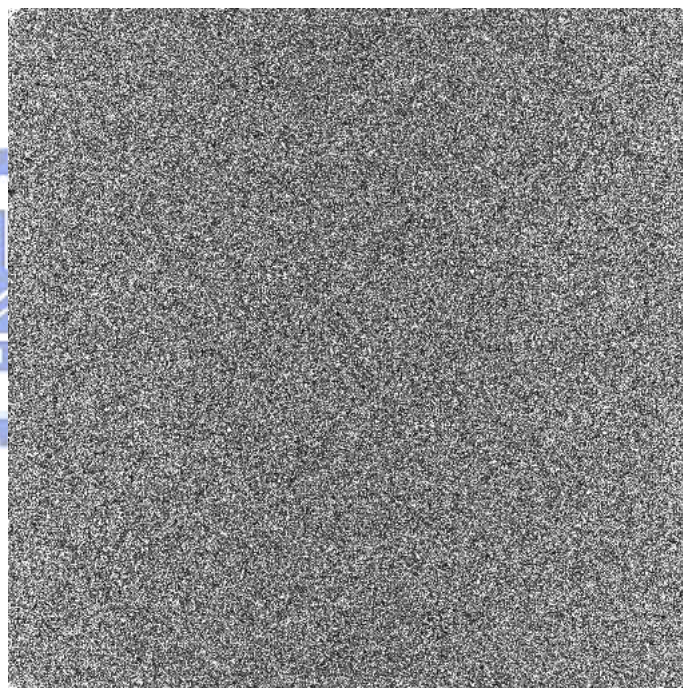
四捨五入造成的雜訊如下表示：

圖 4.3.1



在空間域下的四捨五入雜訊

圖 4.3.2



在頻率域下四捨五入雜訊的 magnitude

其中在空間域下的圖示的處理，是對每個點加上 0.5 使其全部的值都 shift 到正值，然後乘上 256 使其對應到 8-bit 能顯示的範圍內，而在頻率域下的部分則是沒有額外處理

接下來的部分是加入四捨五入雜訊的還原處理，和之前的順序相同，根據不同的參數依序還原 4-1 a, b, c, d

a. 圖 4.3.3

圖 4.3.4



$$K = 10^{-3}, E = 7.8239 \times 10^6 \quad K = 10^{-4}, E = 1.2291 \times 10^7$$



圖 4.3.5

圖 4.3.6



最小 E 值:  $K = 10^{-3.1}, E = 7.7416 \times 10^6$

標準 Wiener filter,  $E = 4.3637 \times 10^6$

b. 圖 4.3.7



圖 4.3.8



$$K = 10^{-3}, E = 6.7061 \times 10^6 \quad K = 10^{-4}, E = 1.2519 \times 10^7$$

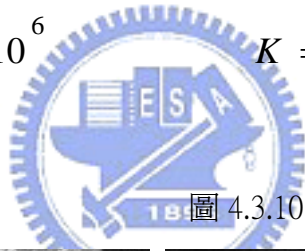


圖 4.3.9



圖 4.3.10



最小 E 值:  $K = 10^{-3.1}, E = 6.118 \times 10^6$

標準 Wiener filter,  $E = 3.4166 \times 10^6$

c. 圖 4.3.11



圖 4.3.12



$$K = 10^{-3}, E = 8.8461 \times 10^6 \quad K = 10^{-4}, E = 1.2528 \times 10^7$$

圖 4.3.13



圖 4.3.14



最小 E 值:  $K = 10^{-3.2}, E = 8.5462 \times 10^6$

標準 Wiener filter,  $E = 4.9236 \times 10^6$

d.

圖 4.3.15

圖 4.3.16



$$K = 10^{-3}, E = 1.1951 \times 10^6$$

$$K = 10^{-4}, E = 1.622 \times 10^6$$



圖 4.3.17

圖 4.3.18



最小 E 值:  $K = 10^{-3}, E = 1.1951 \times 10^6$

標準 Wiener filter,  $E = 5.9832 \times 10^5$



#### 4.4 加入額外雜訊的還原

這一節的模擬分成兩個部分，第一個部分加入強度為  $1(\pm 0.5)$  的範圍內的均勻雜訊，第二個部分改為加入強度為  $10(\pm 5)$  的範圍內的均勻雜訊，測試其還原結果。使用的手振模糊模型是 4.1 的 a, 水平手振模型

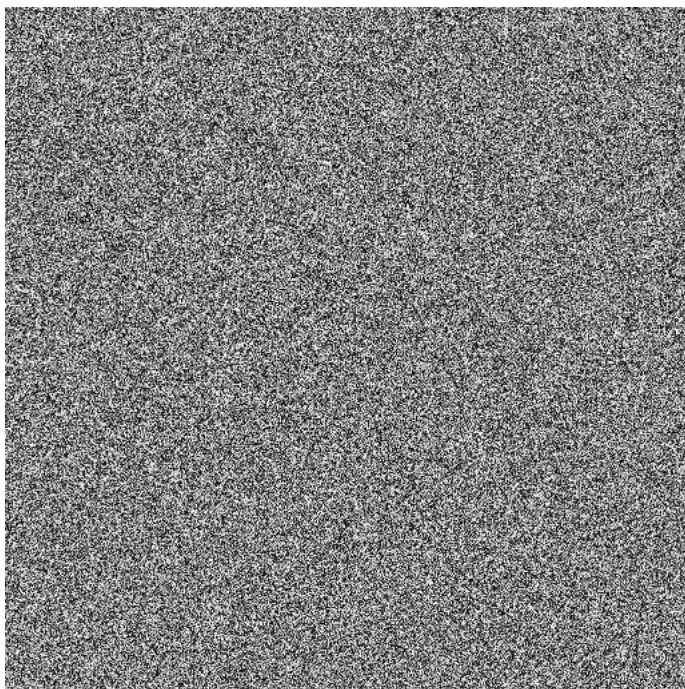


圖 4.4.1 均勻雜訊

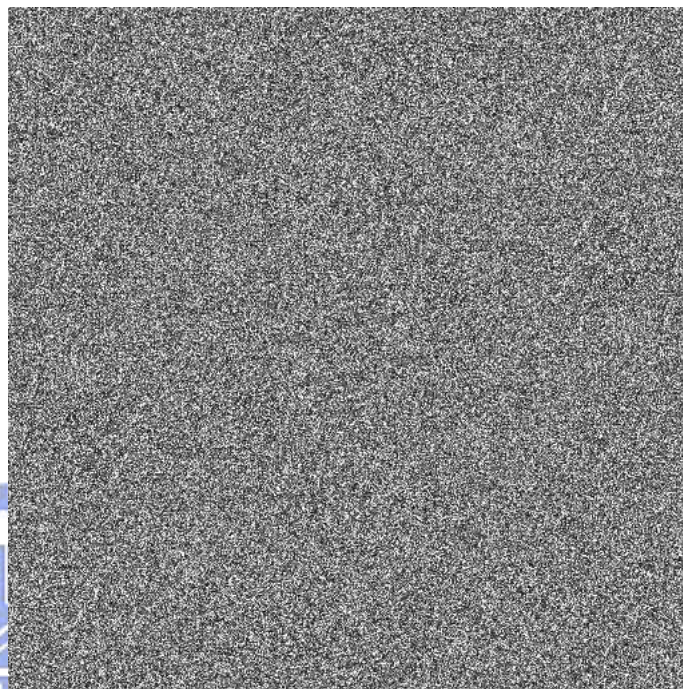


圖 4.4.2 均勻雜訊頻譜的 magnitude



圖 4.4.3 加入強度 1 雜訊的模糊影像

#### 4.4.1 加入強度 1 均勻雜訊的影像還原

圖 4.4.4



圖 4.4.5



$$K = 10^{-3}, E = 7.9468 \times 10^6$$

$$K = 10^{-4}, E = 1.2823 \times 10^7$$

圖 4.4.6



圖 4.4.7



最小 E 值:  $K = 10^{-3}, E = 7.8951 \times 10^6$

標準 Wiener filter,  $E = 4.3183 \times 10^6$

#### 4.4.2 加入強度 10 均勻雜訊的影像還原



圖 4.4.8 加入強度 10 雜訊的模糊影像

圖 4.4.9



$$K = 10^{-2}, E = 4.7169 \times 10^7$$



圖 4.4.10



$$K = 10^{-3}, E = 2.823 \times 10^8$$

圖 4.4.11



圖 4.4.12



最小 E 值:  $K = 10^{-1.6}$ ,  $E = 3.4677 \times 10^7$  標準 Wiener filter,  $E = 1.8115 \times 10^7$

從 4.3 和 4.4 以上實驗的結果我們可以發現到，無論是均勻雜訊或是四捨五入雜訊，對於還原影像最主要的影響是增加了雜訊這部分的影響。

和上一節的四捨五入雜訊比較起來，和加入強度 1 的均勻雜訊的還原的結果差別不大。四捨五入雜訊和強度 1 的均勻雜訊在頻譜上的分佈情形相當類似，都是不定強度的訊號分散在各頻率中，沒有明顯集中高頻或低頻上，所以用 Wiener filter 還原的影像效果相當類似，E 值也相去不遠。

加上強度 10 的均勻雜訊，對還原的結果影響相當大，還原後影像的雜訊有進一步被放大的情形。

#### 4.5 有誤差的手振模型還原

想要更加貼近實際上的手振模糊的情形，有一個必須考慮的要素，就是我們推測的手振誤差模型與實際上不是完全吻合。事實上實際的手振模糊的情形不是我們在數位化的世界中所能模擬出來的，再加上有更多現實上的非理想效應需要考慮，我們所能做的部分只能儘量的做出與現實相近的手振模型而已。為了進一步了解有估測誤差的手振模型所帶來的影響，在這裡做了以下模擬：

#### 4.5.1. 有估測誤差的水平手振模糊還原

這裡使用的同樣是 4.1 之 a 的水平手振模糊影像，然後做以下的模擬：

##### 4.5.1.1. 無雜訊，使用正確的手振模糊模型的還原

在這個部分中，建立模糊影像使用的手振模糊模型，和還原時使用的手振模糊模型是相同的，由於沒有任何的外加雜訊，所以可以完美的還原

圖 4.5.1



$$K = 10^{-2}, E = 1.6354 \times 10^7$$

圖 4.5.2



$$K = 10^{-3}, E = 5.1513 \times 10^6$$

圖 4.5.3

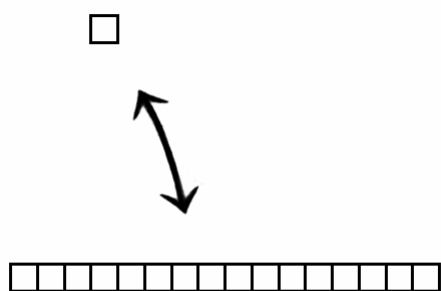


$$K = 0, E = 0$$

#### 4.5.1.2. 無外加雜訊，有估測誤差模糊模型的還原(16pixel→17pixel)

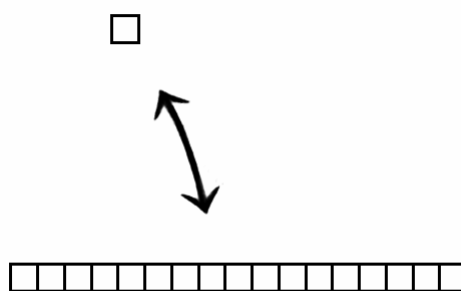
建立模糊影像的模糊模型每個 pixel 的加權值是 1/16, 還原時使用的模型為 1/17

圖 4.5.4



建立模糊影像使用模型(1:16pixel)

圖 4.5.5



還原時估測的手振模型(1:17pixel)

圖 4.5.6



圖 4.5.7



$$K = 10^{-2}, E = 2.2931 \times 10^7$$

$$K = 10^{-3}, E = 2.554 \times 10^7$$



圖 4.5.8



圖 4.5.9

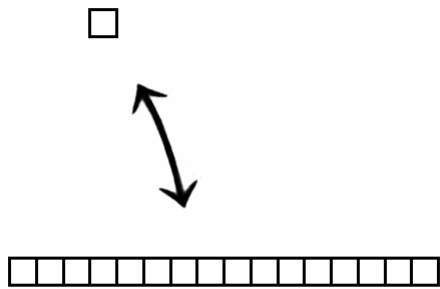


$$\text{最小 } E \text{ 值: } K = 10^{-2.4}, E = 2.0727 \times 10^7$$

$$\text{標準 Wiener filter, } E = 1.115 \times 10^7$$

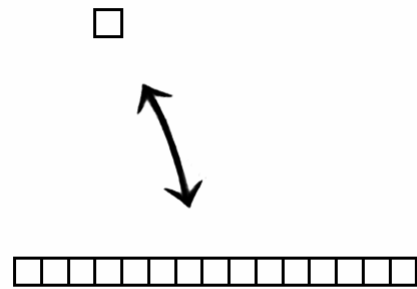
4.5.1.3 無外加雜訊，有估測誤差模糊模型的還原(16pixel→15pixel)

圖 4.5.10



建立模糊影像使用模型(1:16pixel)

圖 4.5.11



還原時估測的手振模型(1:15pixel)

圖 4.5.12



$$K = 10^{-2}, E = 2.307 \times 10^7$$



圖 4.5.13



$$K = 10^{-3}, E = 2.3738 \times 10^7$$



圖 4.5.14



圖 4.5.15



最小 E 值:  $K = 10^{-2.5}$ ,  $E = 2.0494 \times 10^7$       標準 Wiener filter,       $E = 8.9424 \times 10^6$

#### 4.5.2 有估測誤差的 45 度手振模糊還原

這裡使用的是 4.1 之 c 的 45 度手振模糊影像，然後做以下的模擬：

##### 4.5.2.1 無雜訊，使用正確的手振模糊模型的還原



圖 4.5.16       $K = 10^{-2}$ ,  $E = 2.0004 \times 10^7$



圖 4.5.17       $K = 10^{-3}$ ,  $E = 6.1642 \times 10^6$



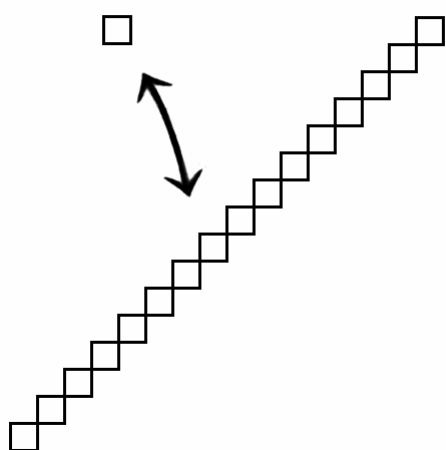
圖 4.5.18  $K = 0$  ,  $E = 0$

4.5.2.2 無外加雜訊，有估測误差模糊模型的還原

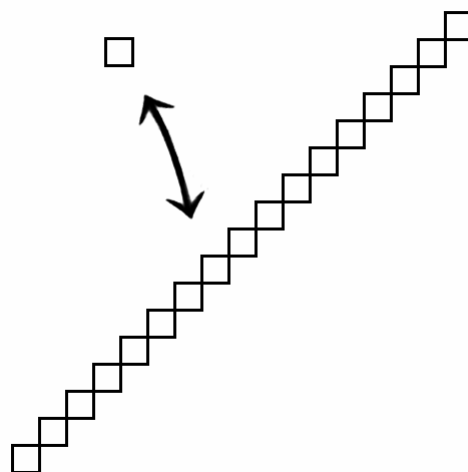


圖 4.5.19

圖 4.5.20



建立模糊影像使用模型(1:16pixel)



還原時估測的手振模型(1:17pixel)

圖 4.5.21



圖 4.5.22



$$K = 10^{-2}, E = 2.807 \times 10^7$$

$$K = 10^{-3}, E = 3.1773 \times 10^7$$



圖 4.5.23



圖 4.5.24

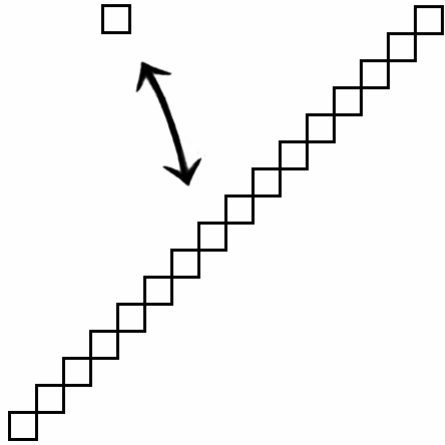


$$\text{最小 } E \text{ 值: } K = 10^{-2.4}, E = 2.5779 \times 10^7$$

$$\text{標準 Wiener filter, } E = 1.3061 \times 10^7$$

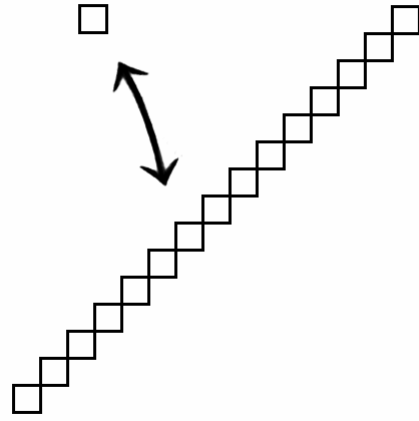
4.5.2.3 無外加雜訊，有估測誤差模糊模型的還原

圖 4.5.25



建立模糊影像使用模型(1:16pixel)

圖 4.5.26



還原時估測的手振模型(1:15pixel)

圖 4.5.27



$$K = 10^{-2}, E = 2.9191 \times 10^7$$

圖 4.5.28



$$K = 10^{-3}, E = 3.1424 \times 10^7$$

圖 4.5.29



圖 4.5.30



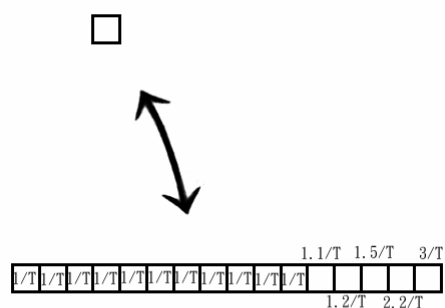
最小 E 值:  $K = 10^{-2.4}$ ,  $E = 2.654 \times 10^7$       標準 Wiener filter,       $E = 1.2274 \times 10^7$



4.5.3 有估測誤差的手振模糊還原

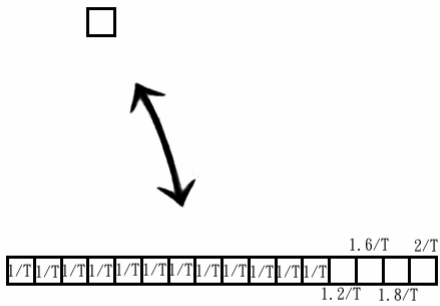


圖 4.5.31 模糊影像



其中,  $T=20$

圖 4.5.32 建立模糊影像使用模型



其中,  $T=18.6$

圖 4.5.33 還原時估測的手振模型

4.5.3.1 無外加雜訊，無估測誤差模糊模型的還原

圖 4.5.34



$$K = 10^{-2}, E = 3.4146 \times 10^6$$



圖 4.5.35



$$K = 10^{-3}, E = 7.3813 \times 10^4$$

圖 4.5.36



$$K = 0, E = 0$$

4.5.3.2 無外加雜訊，有估測誤差模糊模型的還原

圖 4.5.37



$$K = 10^{-2}, E = 5.8961 \times 10^6$$

圖 4.5.38



$$K = 10^{-3}, E = 7.0338 \times 10^6$$

圖 4.5.39

圖 4.5.40



最小 E 值:  $K = 10^{-2.3}$ ,  $E = 4.3312 \times 10^6$  標準 Wiener filter,  $E = 1.8787 \times 10^6$

4.5.1 和 4.5.2 的估測誤差對還原影像造成最主要的影響和 4.2 的情況有點類似，都是出現了沿著手振震動的方向重複影像中邊界部分的邊界形狀的干擾，只是在這裡更為嚴重。手振模糊模型的估計，長度差了一個 pixel 影響就相當大。而 4.5.3 的部分還原的效果相當好，甚至對於 4.2 的邊界狀干擾情形還有抵銷的作用。

## 4.6 邊界效應

對於手振還原來說，有一個相當重要影響手振還原的因素，就是手振模糊圖片的邊界效應的影響。

數位影像手振模糊的建的想法就是，模糊影像的的其中一點是，由未退化影像相對於這一點位置的周圍的數個點加權相加合成，如果考慮點的位置在影像邊界的時候，這個位置的模糊影像點需要超出邊界的未退化影像點來合成，也就是形成一張手振模糊影像需要比這張影像更大 size 的未退化影像。



在數位影像處理裡，在空間域中 “  $h * f$  ” 的 convolution 運算必須是 circular convolution 的情況，轉換到頻域才會是 “  $H \cdot F$  ”，相乘的結果，本篇論文的手振還原法就是基於這樣的條件下所建立的。本來，在模糊影像邊界的點是從超出邊界的未退化影像中取得的，用數位影像處理的方法的 circular convolution 來模擬手振模糊的情況下，整個影像視為同一張影像重複排列而成，也就是超出左邊的邊界時視為從影像的最右邊的點，超出下邊界時視同從影像的最上方，如此而形成模糊影像。但是現實中的模糊影像則不是如此，因為在現實的手振還原的運算裡，我們所面對的情況就是欲求得的未知數的數量大於能列出的式子的數量，這使的完美的還原變的更不可能。

在這節要測試的內容就是，邊界效應對於還原的影響

首先使用 4-1 的方法得到的模糊圖片，然後把 circular convolution 造成的效應的邊界去除，然後依照原來一樣的模糊模型建立合適 size 的 H，然後進行還原。在這一節的模擬中，全部使用的手振模糊模型皆為 4-1 的 a (如下圖所示)：

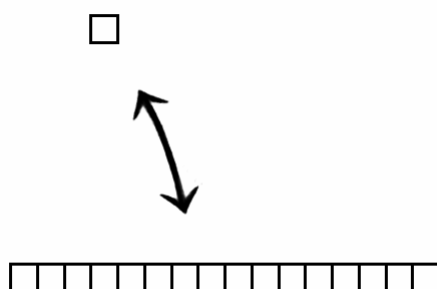


圖 4.6.1

#### 4.6.1 去邊界後的模糊影像還原 例一



圖 4.6.2 數位影像處理產生的手振模糊影像



圖 4.6.3 去掉現實狀況所沒有的邊界後的影像

使用去掉邊界後的模糊影像進行還原，結果如下：



圖 4.6.4  $k = 10^{-2}$

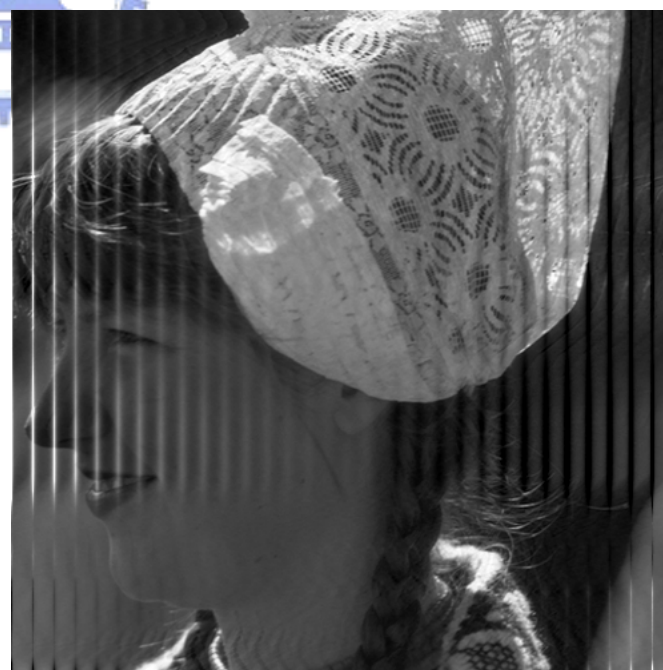


圖 4.6.5  $k = 10^{-3}$

使用外加四捨五入雜訊的去邊界模糊影像進行還原:



圖 4.6.6  $k = 10^{-2}$

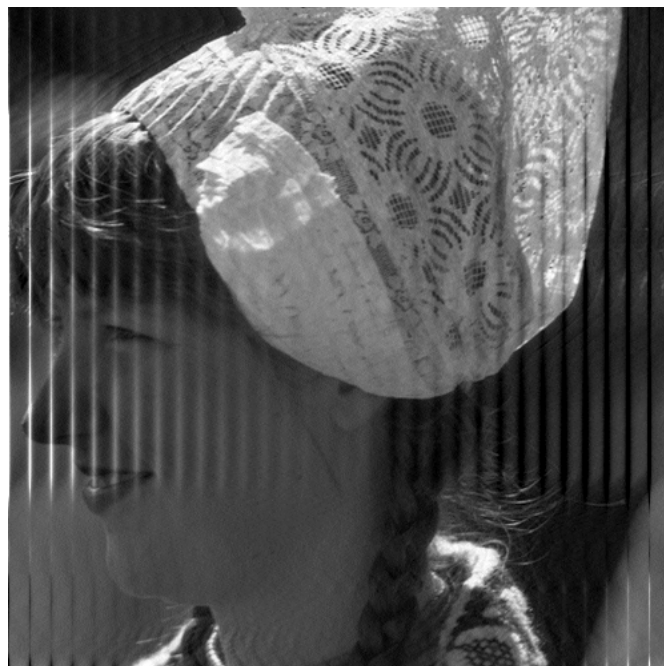


圖 4.6.7  $k = 10^{-3}$



使用另一張影像進行模擬，原圖如下:



圖 4.6.8 原始圖片



圖 4.6.9 數位影像處理產生的手振模糊影像



圖 4.6.10 去掉現實狀況所沒有的邊界後的影像

使用去掉邊界後的模糊影像進行還原，結果如下：



圖 4.6.11  $k = 10^{-2}$

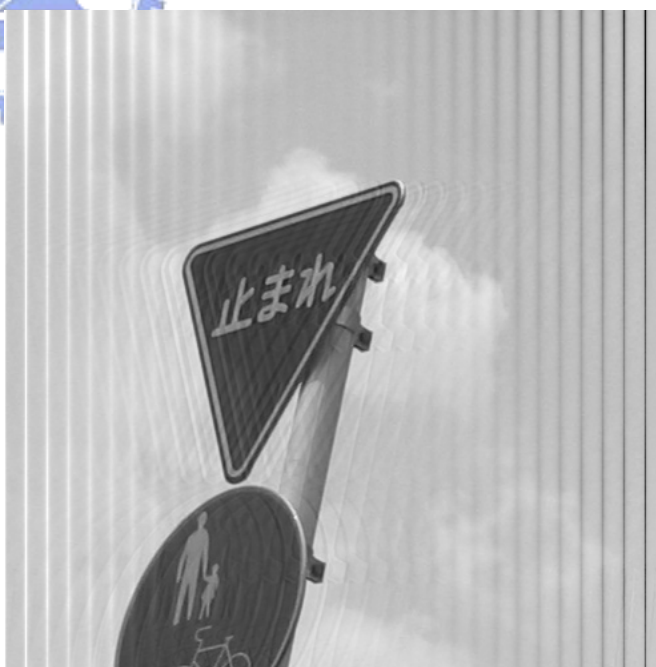


圖 4.6.12  $k = 10^{-3}$

使用外加四捨五入雜訊的去邊界模糊影像進行還原：



圖 4.6.13  $k = 10^{-2}$



圖 4.6.14  $k = 10^{-3}$



#### 4.6.2 去邊界後的模糊影像還原 例二

使用以上兩張  $512 \times 512$  的影像做去邊界還原模擬，由結果可以觀察到還原的影像的確受到相當大的影響。K 值小的時候雖然能得到較好的還原效果，但是相對的條紋狀的干擾就會變的更加嚴重。雖然如此，16 個 pixel 的水平手振對於這樣小尺寸的影像可能太過強烈了，所以下面的模擬換用較大的影像，使用相同的手振模糊模型，測試還原結果：

這裡使用的影像來源是 400 萬畫素(CANON PowerShot A80)的數位像機實際拍攝的照片，影像尺寸為：  $2272 \times 1704$

註：因為原始影像過大，為了方便表示起見，這裡的圖片皆經過 90 度角順時針旋轉處理



圖 4.6.15 原始影像



圖 4.6.16 數位影像處理產生的手振模糊影像



圖 4.6.17 去掉現實狀況所沒有的邊界後的影像





圖 4.6.18 使用去掉邊界後的模糊影像進行還原後的影像  $K=10^{(-2)}$



圖 4.6.19 使用去掉邊界後的模糊影像進行還原後的影像

$K=10^{(-3)}$



圖 4.6.20 外加四捨五入雜訊的去邊界模糊影像進行還原後的影像

$K=10^{(-3)}$

由以上的還原結果可以觀察到，即使在  $K = 10^{-3}$  的情況，條紋狀干擾只限於影像兩邊的邊緣部份，所以在這裡邊界效應對於整個還原影像的影響變的較不明顯。

在實際上的手振模糊影像還原裡，我們遇到的影像的尺寸通常是像這種大尺寸的，而且再加上更多的不理想因素的存在，使的  $K$  值沒辦法設的太小，太小的話還原影像的雜訊干擾勢必變的相當嚴重，所以邊界效應的影響就更不明顯了。

## 4.7 結論

大體來說，雖然標準的 Wiener filter 還原可以達成最好的還原效果，但是這再現實上是做不到的，因為不管是雜訊的數值或是正常影像的數值，都是拿到一張模糊照片時所不知道的，因此本章的標準的 Wiener filter 還原只是用來與其他的還原影像做比較用的。另外，尋找使  $E$  值最小的  $K$  值這個部分雖然需要正常影像來計算  $E$  值，但是實際這樣的  $K$  值也是有可能再我們尋找滿意還原影像時被我們選擇，而且  $E$  值最小的時候也不一定就是使用者所滿意的影像。

從還原影像大體上來看，當  $K$  值較接近 0 時，影像的細節等還原的效果會比較好，但是各種不理想效應也會變的更加嚴重，反而使還原影像的細節被遮蓋住，使得還原影像變的更糟糕。相對的，如果  $K$  值比較大的時候，還原影像會比較模糊，無法還原出細節，但是相對的非理想效應的影響會比較輕微。手振模糊這種情況對影像來說就像是一種低通濾波器的效果，而手振模糊還原則像是透過高通濾波器的處理，因此使得不理想效應變的更明顯。因此，如何抉擇大的  $K$  值或小的  $K$  值就看使用者個人的需求了。

# 第五章 實際手振模糊照片還原

在本章將對實際的手振模糊照片進行還原，和前一章的模擬部分比起來，存在一個相當大的困難點，就是”如何推測手振模糊模型”這個部分。雖然完全正確無誤差的手振模糊模型是不存在的，但是好的手振模糊模型的確有利於做成好的還原影像。

本章的實際手振模糊照片還原分成兩個部分，第一部分的照片是像機固定不動，移動被攝體所拍攝而成，這與實際上發生的手振的情況相當，都是由鏡頭與被攝體的相對移動所造成的。第二個部分就是實際拍攝時發生手振情形的照片，推測其手振模糊模型，然後進行還原。

本章實驗使用的數位相機:

5.2.2 CANON PowerShot A610, 其餘: CANON PowerShot A80

註: 因為原始影像過大，為了方便表示起見，本章的圖片皆經過 90 度角順時針旋轉處理



## 5.1 移動被攝體造成手振模糊效果之還原

首先在這一節裡，我想要測試的是實際上發生的等速水平手振和等速 45 度角手振，使用的模糊照片的取得的方法如下：

1. 將相機安裝於腳架上，確認相機機身保持水平的狀態
2. 調整相機延遲數秒拍攝
3. 將被攝物體(這裡使用的是書本)置於向機鏡頭前方的桌面上
4. 書面與相機機身保持水平，按下快門同時橫向拉動書本，盡量維持等速移動
5. 攝影完成後，調整腳架使相機機身與桌面成 45 度，然後重複 1~4

### 5.1.1 水平移動還原

圖 5.1.1 水平移動照片



圖 5.1.2 還原結果 (K=10<sup>-1.5</sup>)



5.1.2 45度移動還原

圖 5.1.3 45度移動照片





圖 5.1.4 還原結果( $K=10^{(-1.5)}$ )



### 5.1.3 結論

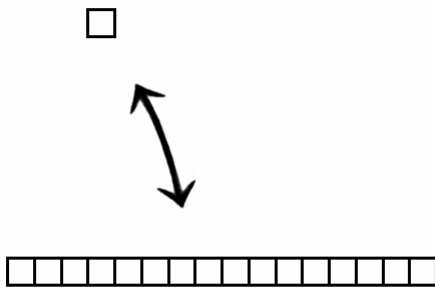


圖 5.1.5 “5.1.1”的預測手振模糊模型

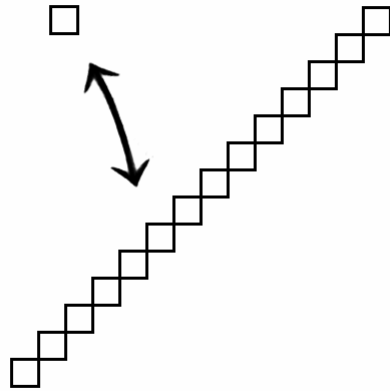


圖 5.1.6 “5.1.2”的預測手振模糊模型

上面兩個預測手振模糊模型，左圖為水平 16 pixel，右圖為 45 度 14 pixel

由還原的結果來看，還原的效果應該算是成功的。兩張原始的照片原本有一些無法辨識的比較小的文字，變的可以分辨出來了。



## 5.2 實際發生的手振模糊照片還原實例

要從一張發生手振模糊的照片中估測手振模糊模型，並不是一件簡單的事。在這裡我的做法首先是，尋找白點的拖曳軌跡。白點就是指在照片中存在的特別亮的單一點，在發生手振的照片中將會拖出一道特別亮的軌跡，這道軌跡的形狀就是我們的手振模糊模型的外形，軌跡各點的強度就是決定模型各點的加權值大小的依據，以此來決定還原照片用的手振模糊模型。不過這個也是大概的原則，因為即使是在正常照片中的一個白點，變成手振模糊照片時拖曳出來的軌跡也會受到這個點附近的其他點的數值所影響，比較理想的狀況是白點軌跡存在的地方的附近是非常相近灰階值的色塊。除此之外，在白點軌跡上亮度較高的地方常常會向外擴散成大範圍的狀況，並不是像前述的理論那樣是線性的，還有鏡頭本身造成的不理想效應，鏡頭距離被攝物的遠近、角度等，都會影響使用者對手振模糊模型的外形和各點加權值的判斷。一般來說判斷的基準位置是照片的中心附近位置或是對焦對到的地方。

在以下的實際範例中，我所推測的手振模糊模型不是用嚴謹的方法推測出來的，



圖 5.2.2 原始照片:



圖 5.2.3 還原結果( $K=10^{(-1.5)}$ )



## 還原結果(局部)

以下表示的是上面影像裁切下來的局部，可以明顯的看出還原效果的其中一部分。2 組 4 張圖當中，左邊的是原來的照片，右邊的是經過還原後的影像：

### 第一組



圖 5.2.4



圖 5.2.5

### 第二組



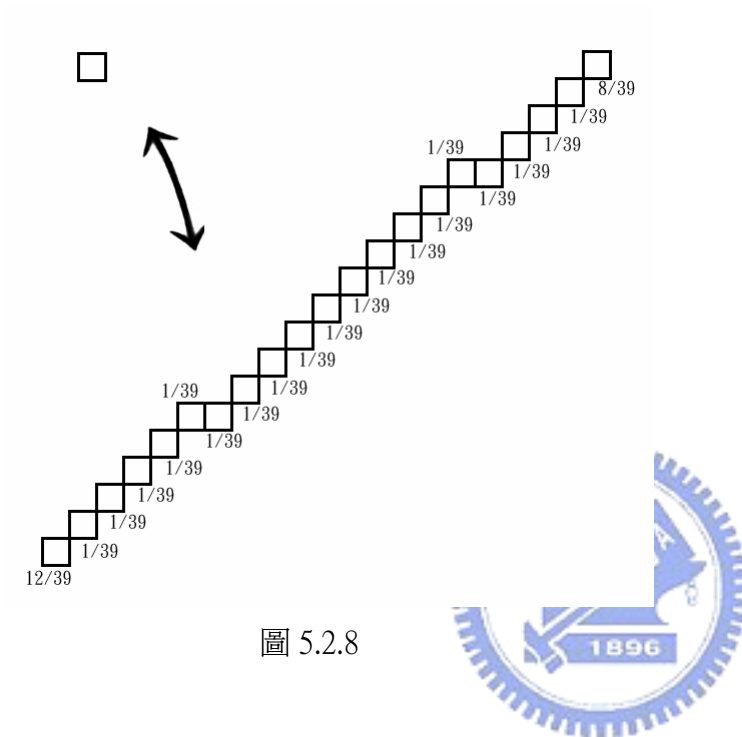
圖 5.2.6



圖 5.2.7

## 5.2.2 實際發生的手振模糊照片還原 例二

估測的手振模糊模型如以下所示:



結果如下:

圖 5.2.9 原始照片:

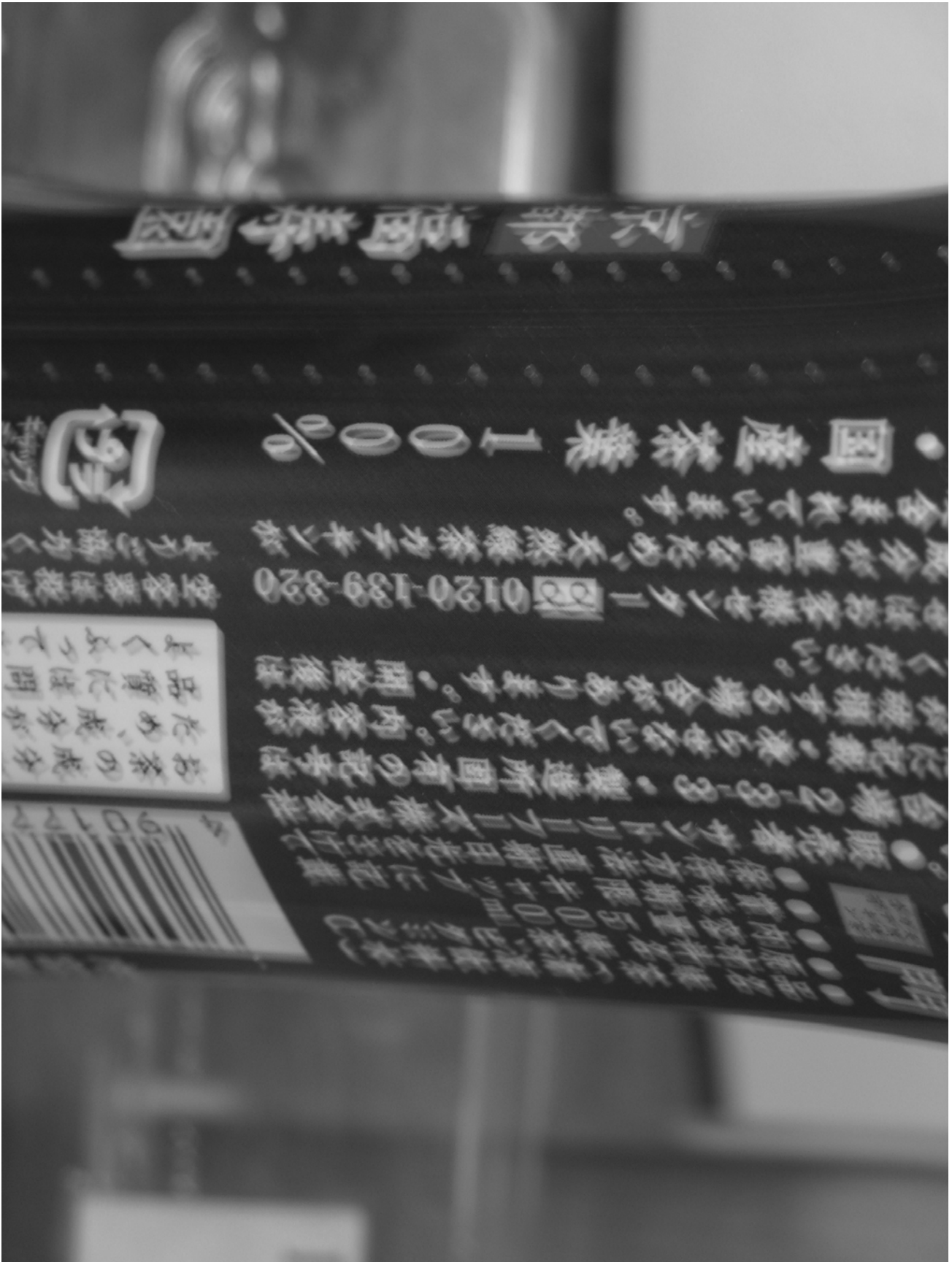
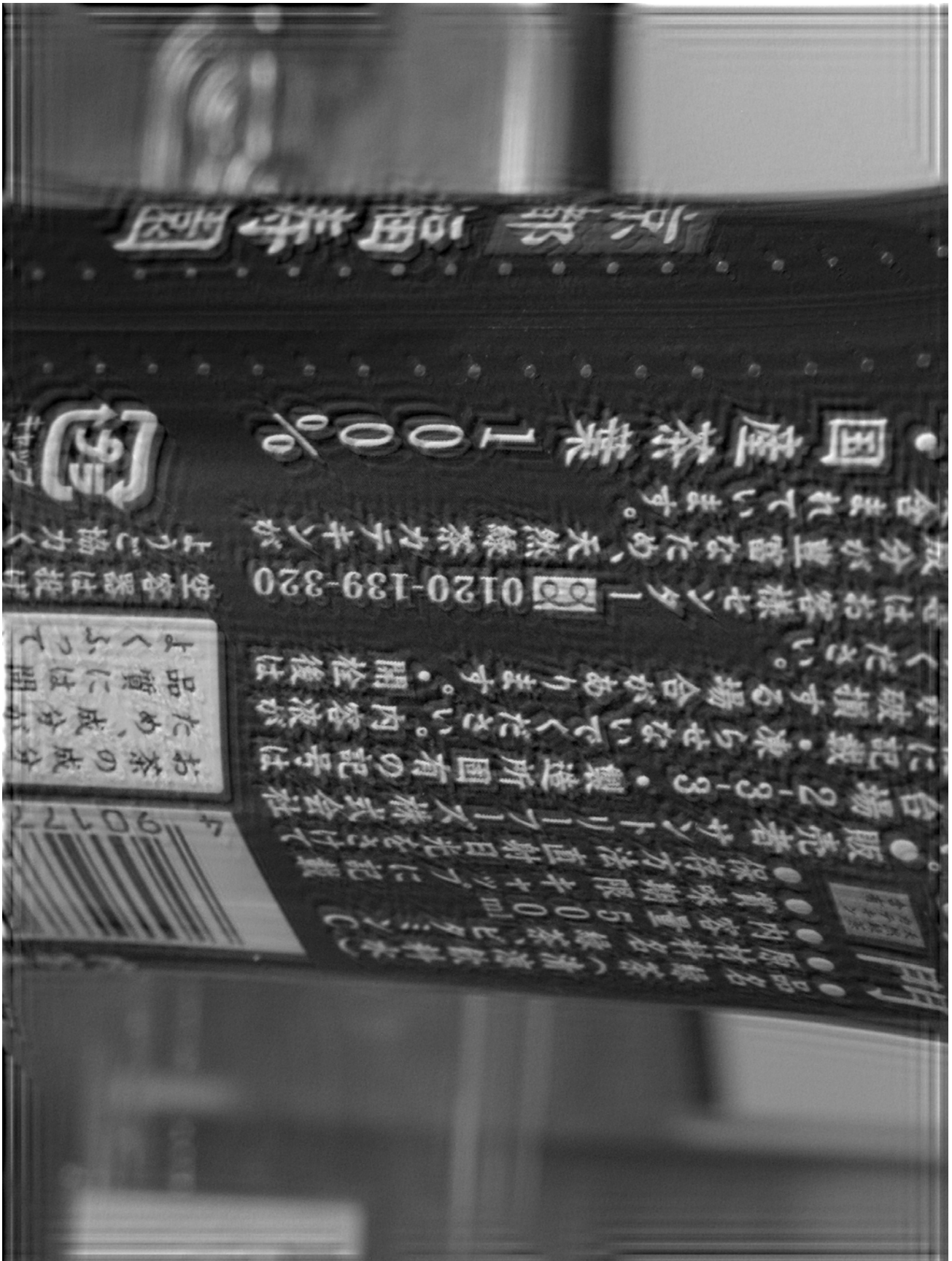




圖 5.2.10 還原結果( $K = 10^{-1.5}$ )



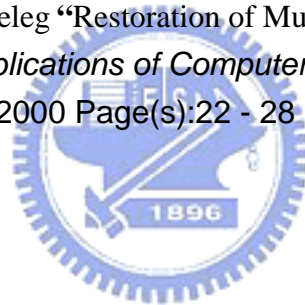
### 5.2.3 結論

首先看 5.2.1 的還原的結果，照片中物體的線條變的更加的清楚了，部分無法辨識的文字也變得可以辨識，照片右邊的本數葉空隙的白點的移動軌跡也大致上還原成一個點的樣子，證明這樣的還原方法和手振還原模型的推測是有效的。雖然如此，但是也和預期的結果一樣，原本天空部分輕微的雜訊變的更嚴重了，物體的邊界處也出現了擾動的現象，然後還有輕微的邊界效應的情形。5.2.2 的部分手振的軌跡拉得更加的長，大致上還原的結果把模糊掉的文字還原到能輕易辨識這些字了。



## 參考資料

- [1] Rafael C. Gonzalez, Richard E. Wood, “**Digital image processing, 2nd ed.**”, Pearson Education/Prentice Hall, 2002
- [2] Rafael C. Gonzalez, Richard E. Wood, Steven L. Eddins, ” **Digital image processing using MATLAB**”, Pearson Education/Prentice Hall, 2004.
- [3] Maria Petrou, Panagiota Bosdogianni, “**Image processing : the fundamentals**”, John Wiley & Sons, Ltd, 1999
- [4] 栗原崇, “手振模糊劣化影像的還原濾波器的評價(手ぶれ劣化画像の復元における復元フィルタ評価)”, 論文, 日本早稻田大學工學部情報學科, 2004
- [5] Alex Rav-Acha, Shmuel Peleg “Restoration of Multiple Images with Motion Blur in Different Directions”, *Applications of Computer Vision, 2000, Fifth IEEE Workshop on.* 4-6 Dec. 2000 Page(s):22 - 28



## 附錄 --- 程式碼 (MATLAB)

### A. 副程式的部分

#### A.1 read\_raw\_file()

```
%%%%%%%%%%%%%%  
% 讀取 raw 檔時使用的副程式  
%  
%%%%%%%%%%%%%%  
function [image]=read_raw_file(file)  
  
[FileSize,n]=size(file);  
Width=FileSize^0.5;  
Height=Width;  
image=zeros(Height,Width);  
k=1;  
for m=1:Height  
    for n=1:Width  
        image(m,n)=file(k,1);  
        k=k+1;  
    end  
end
```



#### A.2 H\_make2()

```
%%%%%%%%%%%%%%  
% 手振模糊函數 H 的生成副程式  
%  
%%%%%%%%%%%%%%  
function [H]=H_make2(l,a,Px,Py,image)  
  
[Height,Width]=size(image);
```

```

H=zeros(Height,Width);
for m=1:Height
    for n=1:Width
        for k=1:l

H(m,n)=H(m,n)+a(k)*exp(i*2*pi*(Px(k)*(m-1)/Height+Py(k)*(n-1)/Width));
            end
        end
    end
end

```

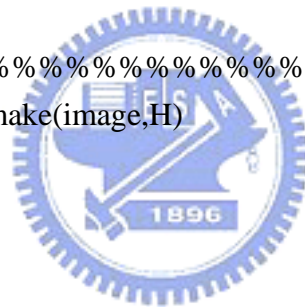
### A.3 BlurredImage\_make()

```

%%%%%%%%%%%%%%
% 生成手振模糊影像的副程式
%
%%%%%%%%%%%%%%
function [G]=BlurredImage_make(image,H)

[Height,Width]=size(image);
I=fft2(image);
for m=1:Height
    for n=1:Width
        G(m,n)=I(m,n)*H(m,n);
    end
end
end

```



### A.4 BlurredImage\_restore\_wiener()

```

%%%%%%%%%%%%%%
% 使用(3.7)式的近似 wiener filter 進行影像還原，生成還原影像的副程式
%
%%%%%%%%%%%%%%
function [B,I_r]=BlurredImage_restore_wiener(G,H,k)

[Height,Width]=size(G);

```

```

for m=1:Height
    for n=1:Width
        B(m,n)=conj(H(m,n))/((abs(H(m,n)))^2+k);
    end
end
for m=1:Height
    for n=1:Width
        I_r(m,n)=G(m,n)*B(m,n);
    end
end

```

### A.5 BlurredImage\_restore\_WienerStandard()

%%%%%%%%%%  
 %使用(3.6)式的標準 wiener filter 進行影像還原，生成還原影像的副程式  
 %

%%%%%%%%%%  
 function [B,I\_r]=BlurredImage\_restore\_WienerStandard(I,G,H,N)

```

[Height,Width]=size(G);

```

```

for i=1:Height
    for j=1:Width
        P(i,j)=(abs(N(i,j)))^2/(abs(I(i,j)))^2;
    end
end

```

```

for i=1:Height
    for j=1:Width
        B(i,j)=conj(H(i,j))/((abs(H(i,j)))^2+P(i,j));
    end
end

```

```

for i=1:Height
    for j=1:Width
        I_r(i,j)=G(i,j)*B(i,j);
    end
end

```



end

### A.6 error\_count()

%%%%%%%%  
% 計算 E 的副程式,

%其中  $E = e^2 = E\{(f - \hat{f})^2\}$

%%%%%%%%

function [e]=error\_count(Image,I\_r)

e=0;

[Height,Width]=size(I\_r);

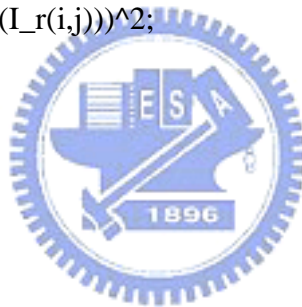
for i=1:Height

for j=1:Width

e=e+(Image(i,j)-real(I\_r(i,j)))^2;

end

end



### A.7 uint8noise\_make()

%%%%%%%%  
% 生成四捨五入誤差的副程式

%

%%%%%%%%

function [N]=uint8noise\_make(I,H,G);

[Height,Width]=size(I);

for m=1:Height

for n=1:Width

N(m,n)=G(m,n)-H(m,n)\*I(m,n);

end

end

## A.8 MinE\_search()

%%%%%%%%%%  
% 使用(3.7)式的近似 wiener filter 時，尋找使 E 值最小的 K 值，精確度到 t 的小  
% 數以下第 1 位，其中：

% K=10^(t)

%%%%%%%%%%

function [e2,t,i\_r2]=MinE\_search(image,G,H,t)

K=10^(t);

[B,I\_r]=BlurredImage\_restore\_wiener(G,H,K);

i\_r=ifft2(I\_r);

e1=error\_count(image,i\_r);

e2=e1+1;

while e1<e2

    e2=e1;

    i\_r2=i\_r;

    t=t-0.1;

    k=10^(t);

    [B,I\_r]=BlurredImage\_restore\_wiener(G,H,k);

    i\_r=ifft2(I\_r);

    e1=error\_count(image,i\_r);

end

t=t+0.1;



## A.9 Image\_cut ()

%%%%%%%%%%

% 剪裁圖片邊界的副程式，可輸入裁去的寬度

% 註：這個副程式裁去的是左邊界和下邊界

%%%%%%%%%%

function [image\_new]=Image\_cut(image,H\_cut,W\_cut)

[Height,Width]=size(image);

image\_new=zeros(Height-H\_cut,Width-W\_cut);

for m=1:Height-H\_cut

    for n=W\_cut+1:Width



```

        image_new(m,n-W_cut)=image(m,n);
    end
end

```

## B. 主程式部分

### B.1 手振模糊影像的生成

```

clear;
fid=fopen('E:\ImageProcess\Project\proj1A\image 93-2.raw');
file=fread(fid);
image=read_raw_file(file);

%%%%%%%%%%%% 手振模糊模型建立 %%%%%%%%%%%%%
l=16; %模型包含的 pixel(點)數
a=[1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16]; %點的加權值
Px=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]; %點在 x 座標的位置
Py=[0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 -11 -12 -13 -14 -15]; %點在 y 座標的位置
%%%%%%%%%%%%

H=H_make2(l,a,Px,Py,image);
G=BlurredImage_make(image,H);

g=ifft2(G);
imwrite(uint8(g),'E:\PROJECT2\pic\B_h.tif');

```

### B.2 加入四捨五入雜訊的手振模糊圖片的還原

```

clear;
g=imread('E:\PROJECT2\pic\B_h.tif');
g=double(g);

```

```
G=fft2(g);

l=16;
a=[1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16];
Px=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
Py=[0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 -11 -12 -13 -14 -15];
```

```
H=H_make2(l,a,Px,Py,g);
```

```
k=10^(-3);
[B,I_r]=BlurredImage_restore_wiener(G,H,k);
```

```
i_r=ifft2(I_r);
imwrite(uint8(i_r),'E:\PROJECT2\pic\r_c_0001w.tif');
%%%%%%%% E count %%%%%%%%%
fid=fopen('E:\ImageProcess\Project\proj1A\image 93-2.raw');
file=fread(fid);
image=read_raw_file(file);
E=error_count(image,i_r);
```



### B.3 無外加雜訊的手振模糊圖片的還原

```
clear;
fid=fopen('E:\ImageProcess\Project\proj1A\image 93-2.raw');
file=fread(fid);
image=read_raw_file(file);
```

```
l=16;
a=[1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16];
Px=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
Py=[0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 -11 -12 -13 -14 -15];
```

```
H=H_make2(l,a,Px,Py,image);
G=BlurredImage_make(image,H);
k=10^(-3);
[B,I_r]=BlurredImage_restore_wiener(G,H,k);
i_r=ifft2(I_r);
```

```

imwrite(uint8(i_r),'E:\PROJECT2\pic\pure\r_2.tif');
E=error_count(image,i_r);

```

#### B.4 加入強度 10 均勻雜訊的手振模糊影像還原

```

clear;
fid=fopen('E:\ImageProcess\Project\proj1A\image 93-2.raw');
file=fread(fid);
image=read_raw_file(file);
I=fft2(image);

l=16;
a=[1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16];
Px=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
Py=[0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 -11 -12 -13 -14 -15];

H=H_make2(l,a,Px,Py,image);
G=BlurredImage_make(image,H);
g=ifft2(G);
n=rand(size(image))-0.5; % union noise make

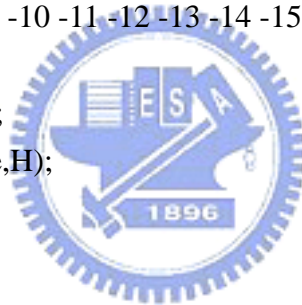
imwrite(uint8((n+0.5)*256),'E:\PROJECT2\pic\u_noise\noise.tif');
imwrite(uint8(abs(fft2(n))),'E:\PROJECT2\pic\u_noise\N.tif');

n=n*10;
N=fft2(n);
G=G+N;

k=10^(-2);
[B,I_r]=BlurredImage_restore_wiener(G,H,k);
i_r=ifft2(I_r);
E=error_count(image,i_r);
imwrite(uint8(i_r1),'E:\PROJECT2\pic\u_noise\r_10u_001W.tif');

t=-1;
[Emin,t,i_r1]=MinE_search(image,G,H,t);
imwrite(uint8(i_r1),'E:\PROJECT2\pic\u_noise\r_10u_minW.tif');

```



### B.5 手振還原模型估測錯誤情況下的還原

```
clear;
fid=fopen('E:\ImageProcess\Project\proj1A\image 93-2.raw');
file=fread(fid);
image=read_raw_file(file);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
l=16;
a=[1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16];
Px=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
Py=[0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 -11 -12 -13 -14 -15];

H=H_make2(l,a,Px,Py,image);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
l=17;
a=[1/17 1/17 1/17 1/17 1/17 1/17 1/17 1/17 1/17 1/17 1/17 1/17 1/17 1/17 1/17 1/17
1/17];
Px=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
Py=[0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 -11 -12 -13 -14 -15 -16];

H2=H_make2(l,a,Px,Py,image);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
l=15;
a=[1/15 1/15 1/15 1/15 1/15 1/15 1/15 1/15 1/15 1/15 1/15 1/15 1/15 1/15 1/15];
Px=[0 0 0 0 0 0 0 0 0 0 0 0 0 0];
Py=[0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 -11 -12 -13 -14];

H3=H_make2(l,a,Px,Py,image);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
G=BlurredImage_make(image,H);

G2=BlurredImage_make(image,H2);
N=G-G2;

k=0;
[B,I_r]=BlurredImage_restore_wiener(G,H,k);
i_r=ifft2(I_r);
```



```

E=error_count(image,i_r);
imwrite(uint8(i_r),'E:\PROJECT2\pic\mode_wrong\r_h_Wmin.tif');

k=10^(-2);
[B,I_r]=BlurredImage_restore_wiener(G,H,k);
i_r=ifft2(I_r);
[B,I_r2]=BlurredImage_restore_wiener(G,H2,k);
i_r2=ifft2(I_r2);
[B,I_r3]=BlurredImage_restore_wiener(G,H3,k);
i_r3=ifft2(I_r3);
e21=error_count(image,i_r);
e22=error_count(image,i_r2);
e23=error_count(image,i_r3);
imwrite(uint8(i_r),'E:\PROJECT2\pic\mode_wrong\r_h_001W.tif');
imwrite(uint8(i_r2),'E:\PROJECT2\pic\mode_wrong\r_h_17_001W.tif');
imwrite(uint8(i_r3),'E:\PROJECT2\pic\mode_wrong\r_h_15_001W.tif');

```

## B.6 手振模糊運算後的影像，裁去邊界後進行還原

```

clear;
image=imread('E:\PROJECT2\IMG_8349.tif');
image=double(image);
I=fft2(image);

l=16;
a=[1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16 1/16];
Px=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
Py=[0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 -11 -12 -13 -14 -15];

H=H_make2(l,a,Px,Py,image);
G=BlurredImage_make(image,H);
g=ifft2(G);
imwrite(uint8(g),'E:\PROJECT2\pic\B_IMG_8349.tif');

[g_new]=Image_cut(g,15,15);
imwrite(uint8(g_new),'E:\PROJECT2\pic\B_IMG_8349_cut.tif');
%%%%%%%%%%
g_new2=imread('E:\PROJECT2\pic\B_IMG_8349_cut.tif');
g_new2=double(g_new2);

```

```

G_new2=fft2(g_new);
H2=H_make2(l,a,Px,Py,g_new);
k=10^(-3);
[B,I_r]=BlurredImage_restore_wiener(G_new2,H2,k);
i_r=ifft2(I_r);
imwrite(uint8(g),'E:\PROJECT2\pic\B_big.tif');

```

## B.7 實際手振模糊照片的還原

```

clear;
g=imread('E:\PROJECT2\pic\IMG_2087.tif');
g=double(g);
G=fft2(g);

l=21;
s=39;
a=[12/s 1/s 1/s 1/s 1/s 1/s 1/s 1/s 1/s 1/s 1/s 1/s 1/s 1/s 1/s 1/s 1/s 1/s 8/s];
Px=[0 1 2 3 4 5 5 6 7 8 9 10 11 12 13 14 14 15 16 17 18];
Py=[0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 -11 -12 -13 -14 -15 -16 -17 -18 -19 -20];

H=H_make2(l,a,Px,Py,g);

k=10^(-1.5);
[B,I_r]=BlurredImage_restore_wiener(G,H,k);

i_r=ifft2(I_r);
imwrite(uint8(i_r),'E:\PROJECT2\pic\realR.tif');

```

## 簡歷

姓名： 蘇群智

出生地： 台灣台中市

學歷：

民國 86 年 私立衛道中學高中部

民國 89 年 國立交通大學電機與控制工程學系

民國 94 年 國立交通大學電機與控制工程研究所

