

# 以電子浮水印為基礎的 著作權保護之研究

---

報 告 人：謝 欣 宏

指 導 教 授：王 金 印

楊 耀 波

# 大綱

---

※摘要

※浮水印的分類

※浮水印的隱藏方法

※實驗結果

※結論

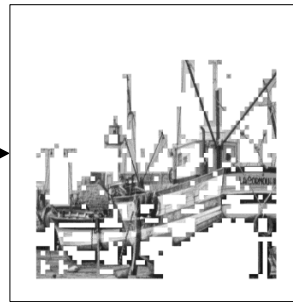
# 本論文的基本特性

- ※ 適用於一般任意灰階並且具有意義(例如指紋、標幟、符號、圖像等)的影像片段來當作浮水印。
- ※ 將浮水印隱藏在原始作品中特徵部分，藉以提高浮水印的強韌度及保持原圖的透明度。

浮水印



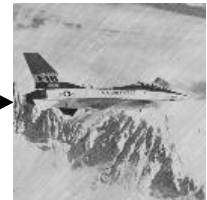
原圖



原圖特徵部分



藏有浮水印的原圖



取回的浮水印

# 本論文的基本特性

※ 以一串數值做為隱藏浮水印的鑰匙（參數），以錯誤的鑰匙取回的浮水印在視覺上像雜訊。

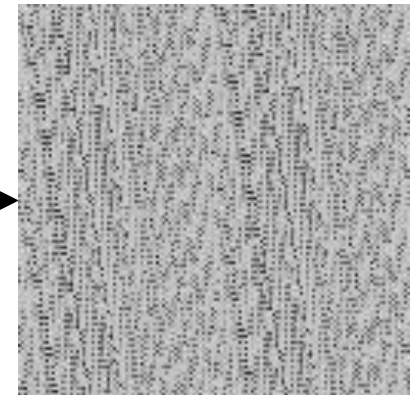


測試影像

錯誤的  
取回參數



浮水印  
取回程序



取回的浮水印像雜訊

# 本論文的達到的目標

---

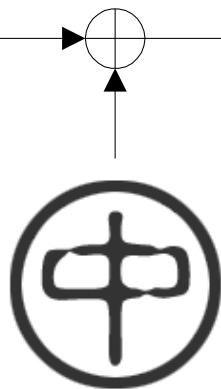
- ※ 原圖在不同的影像格式之間轉換時，浮水印仍然存在。
- ※ 可以有限度承受影像處理的攻擊，如模糊化、切割、尺寸變化等。
- ※ 改良了Toral Automorphisms 演算法（甜甜圈演算法），使該演算法的運算速度加快，原圖的面貌更不易被窺伺。
- ※ 僅需保留部分原始作品檔案，降低硬碟的儲存量。
- ※ 將甜甜圈演算法應用到影像辨識，增加了影像辨識的成功率。

# 浮水印的範例

※ 在數位媒體資料中隱藏一段秘密的資訊，藉此能辨別數位媒體資料的歸屬，保護合法權益。



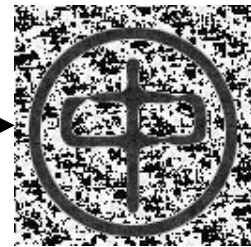
原圖



浮水印



藏有浮水印  
的原圖



取回的浮水印

# 浮水印的分類

---

- 依應用來區分
  - 以來源為導向的浮水印
  - 以目的為導向的浮水印
- 依本身特性來區分
  - 以有意義圖為浮水印
  - 具有統計特性的浮水印

# 浮水印的分類

---

- 以視覺品質來區分浮水印



看得見浮水印  
透明度差，強韌度好

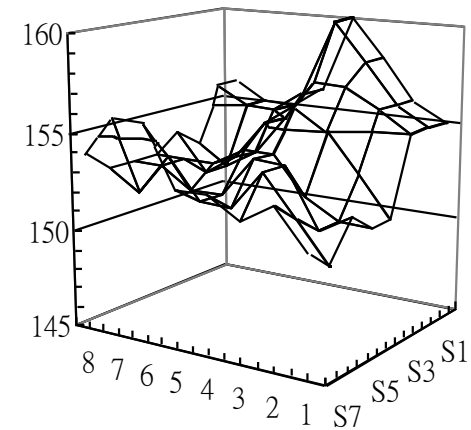


看不見浮水印  
透明度好，強韌度差

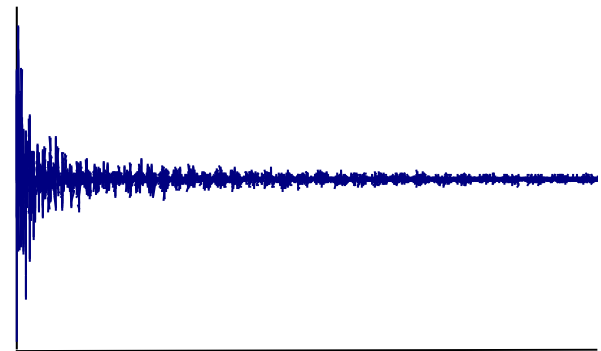


# 浮水印的隱藏方法

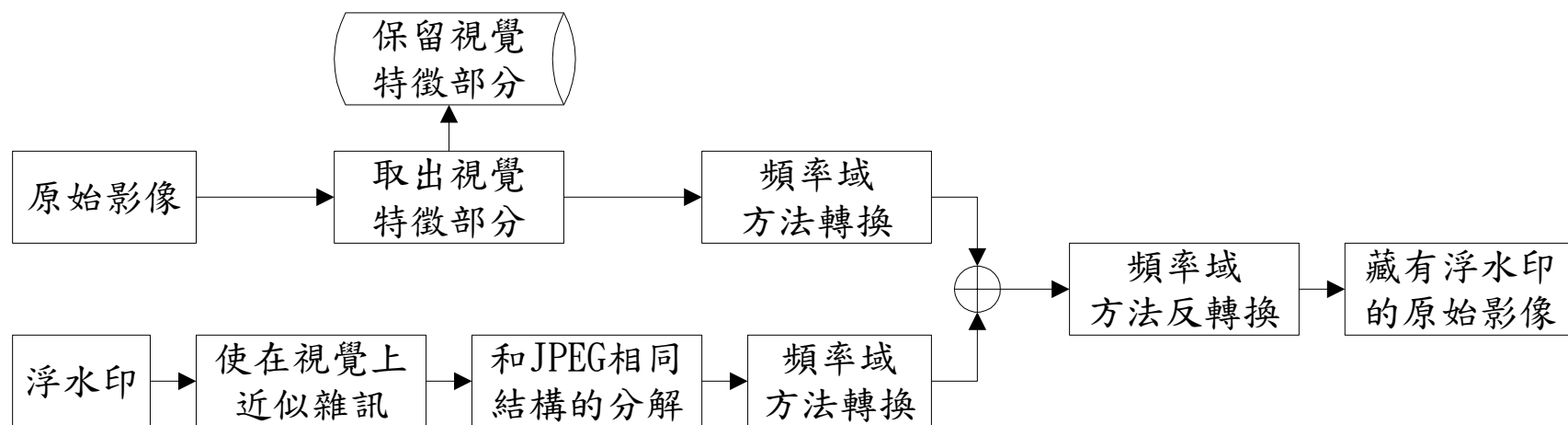
※ 空間定義域 (Spatial domain)



※ 頻率定義域 (Frequency domain)



# 浮水印隱藏流程

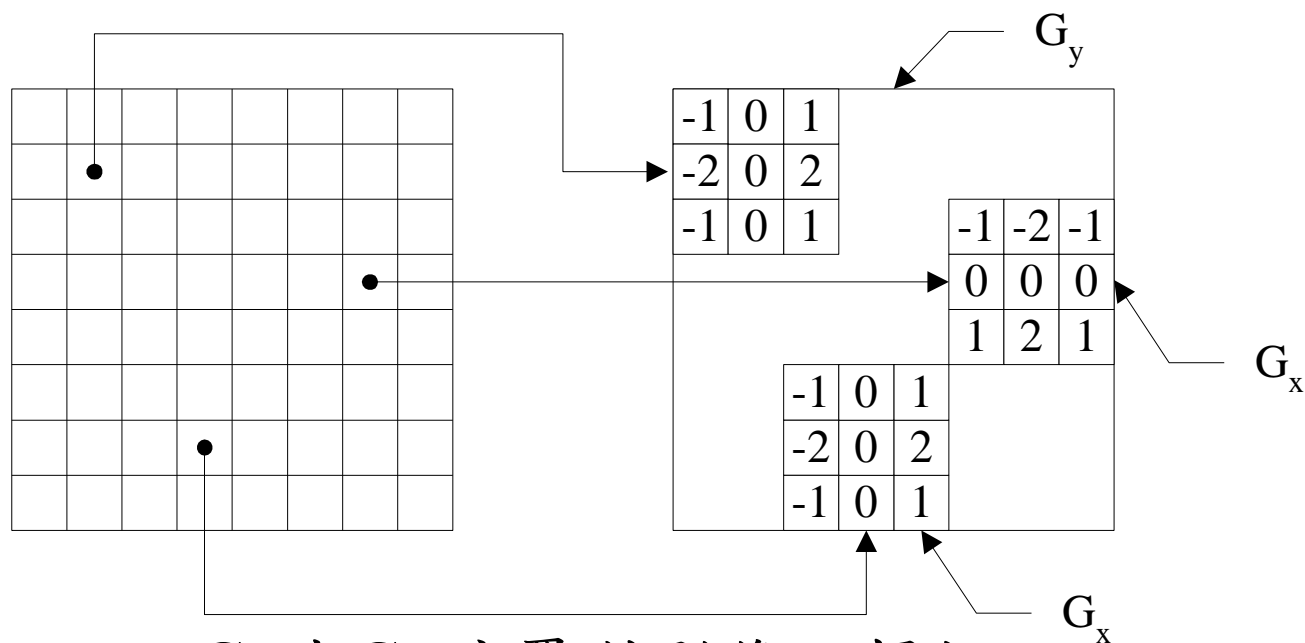


# 浮水印隱藏流程

---

- ※ 找出影像中的特徵部分（以邊緣偵測的方式），並儲存影像特徵部分區塊。
- ※ 使浮水印在視覺上近似雜訊（對浮水印執行甜甜圈演算法）。
- ※ 為抵抗影像失真壓縮（對浮水印執行和JPEG相同結構的分解）
- ※ 為避免在空間域容易被移除（將原始影像和浮水印在頻率域相加）

# 邊緣偵測



以  $G_x$  或  $G_y$  遮罩對影像做掃描

$$I(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2}, 1 \leq x < N-1, \quad y < N-1$$

# 邊緣偵測

---

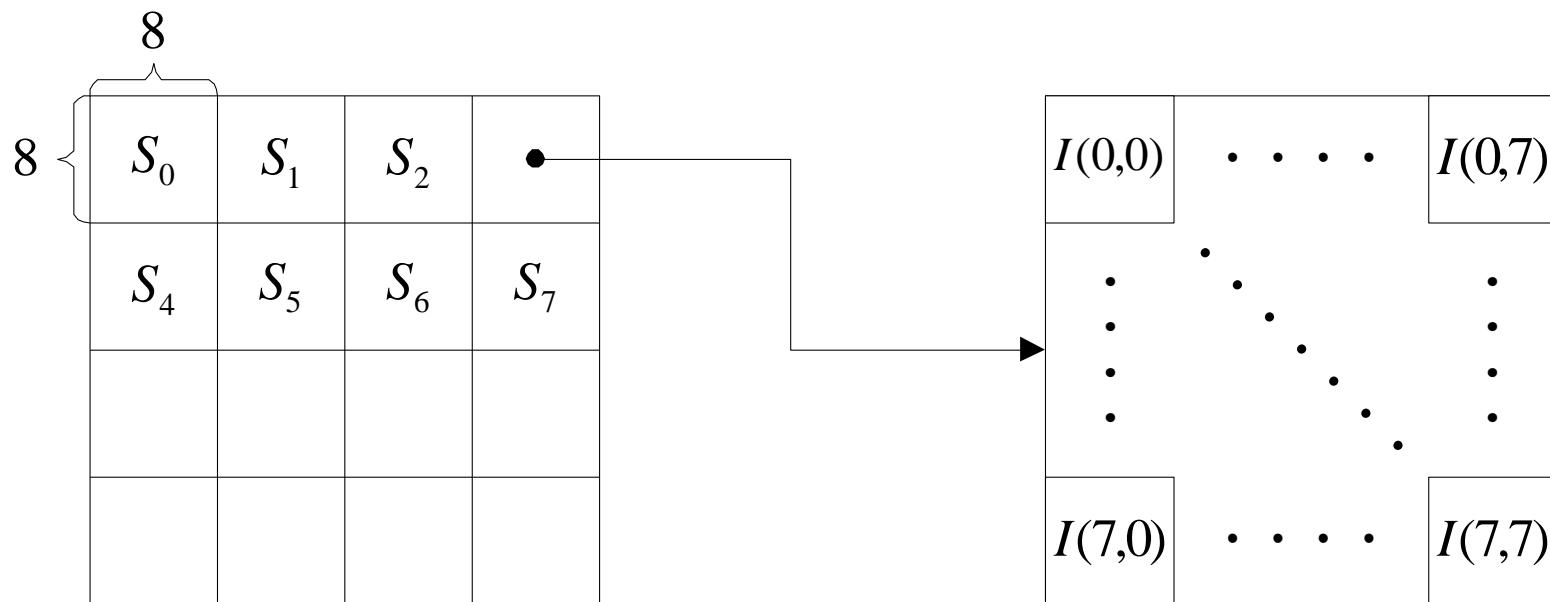


原始影像



經過邊緣偵測的結果

# 找出特徵區塊



$$S_i = \sum_{x=0}^7 \sum_{y=0}^7 I(x, y)^2, \quad i = 0, 1, 2, \dots, (N/8 * N/8) - 1$$

# 離散餘弦轉換

---

$$D(u, v) = \frac{1}{4} C(u) C(v) \left[ \sum_{x=0}^N \sum_{y=0}^N W(x, y) \times \cos \frac{(2x+1)u\pi}{16} \times \cos \frac{(2y+1)v\pi}{16} \right]$$

$$W(x, y) = \frac{1}{4} \left[ \sum_{u=0}^N \sum_{v=0}^N C(u) C(v) \times D(u, v) \times \cos \frac{(2x+1)u\pi}{16} \times \cos \frac{(2y+1)v\pi}{16} \right]$$

其中當  $u, v = 0$  時  $C(u), C(v) = 1/\sqrt{2}$  ；

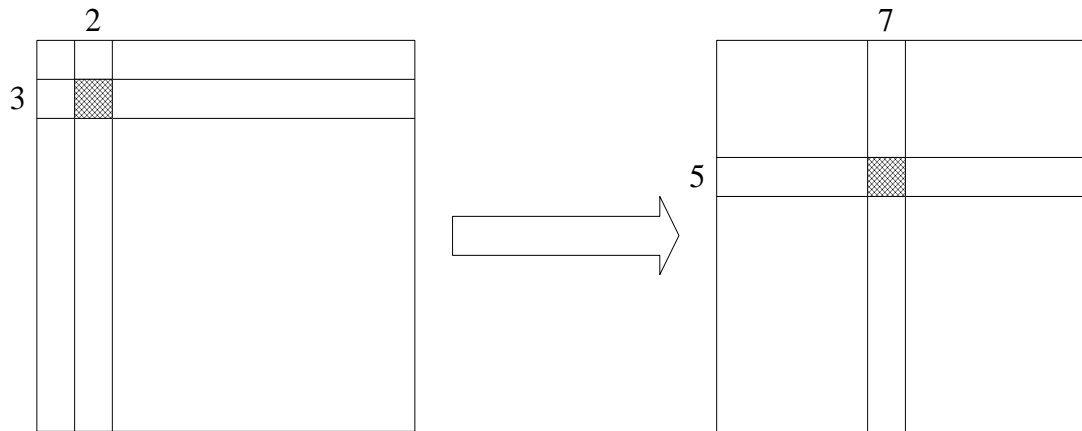
當  $u, v$  是其它情形時  $C(u), C(v) = 1$  ，

而  $N$  為影像的大小。

# 甜甜圈演算法

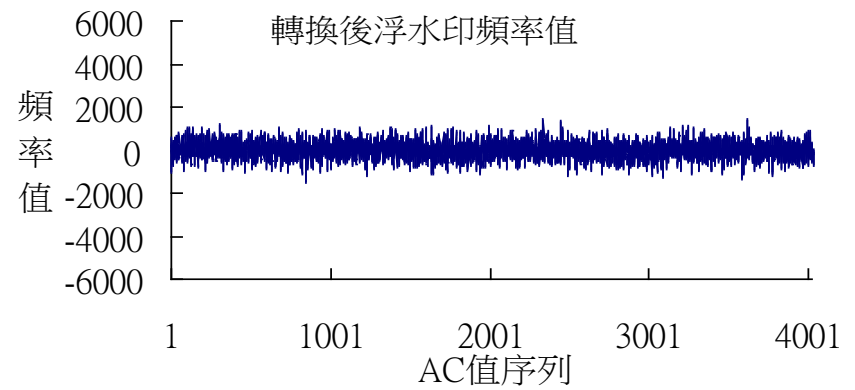
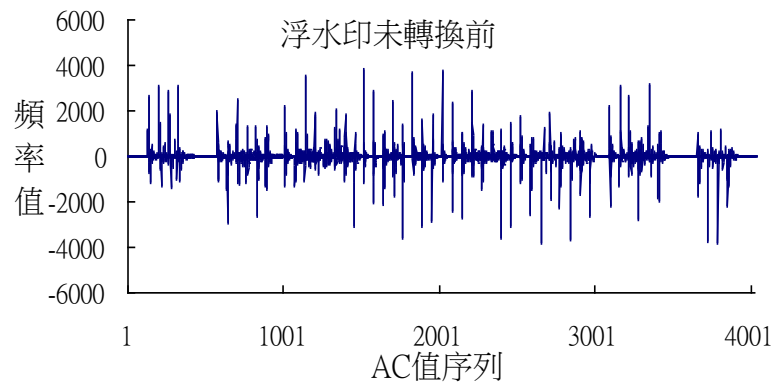
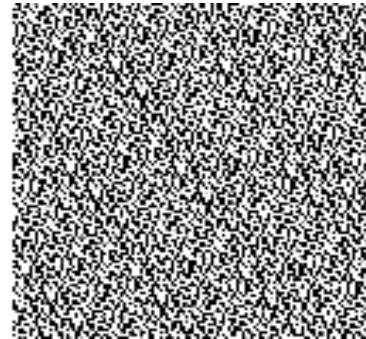
$$\begin{pmatrix} x_t \\ y_t \end{pmatrix} = \begin{pmatrix} \varpi + 1 & \varpi \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_{t-1} \\ y_{t-1} \end{pmatrix} \pmod{N}$$

$$\begin{pmatrix} x_t \\ y_t \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \end{pmatrix} \pmod{512} = \begin{pmatrix} 7 \\ 5 \end{pmatrix}$$

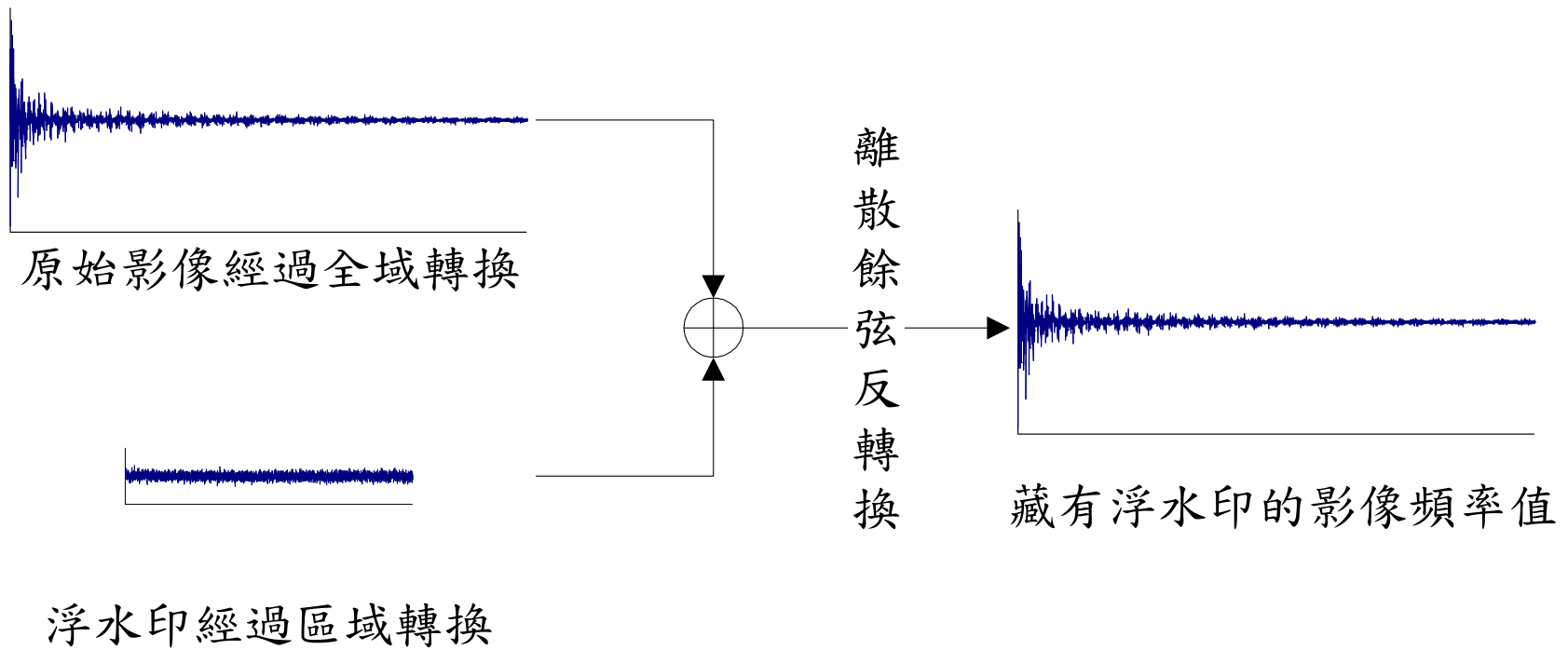




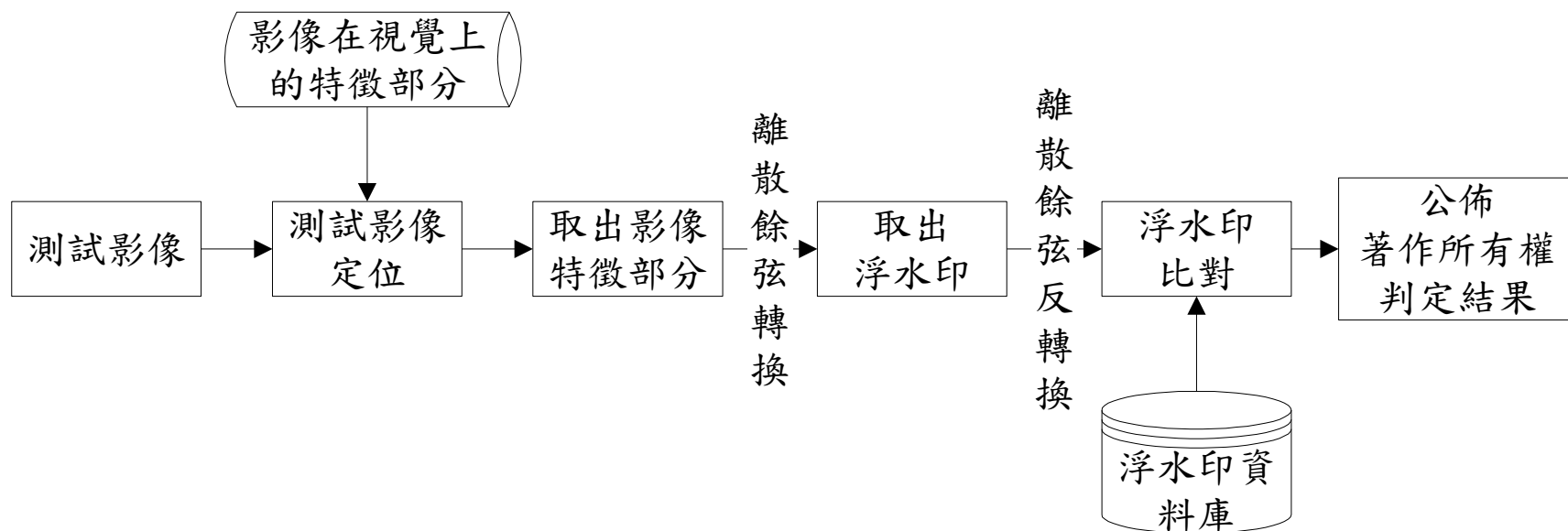
# 甜甜圈演算法的範例



# 隱藏浮水印



# 浮水印取回及辨識

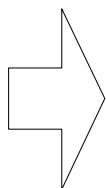


# 浮水印取回及辨識

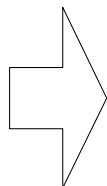
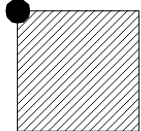
---

- ※影像定位（當測試影像為原始作品的部分影像時，為了取出正確的藏有浮水印的區塊）。
- ※以浮水印隱藏程序的反向操作取回浮水印。
- ※將甜甜圈演算法應用到辨識提高辨識成功率。

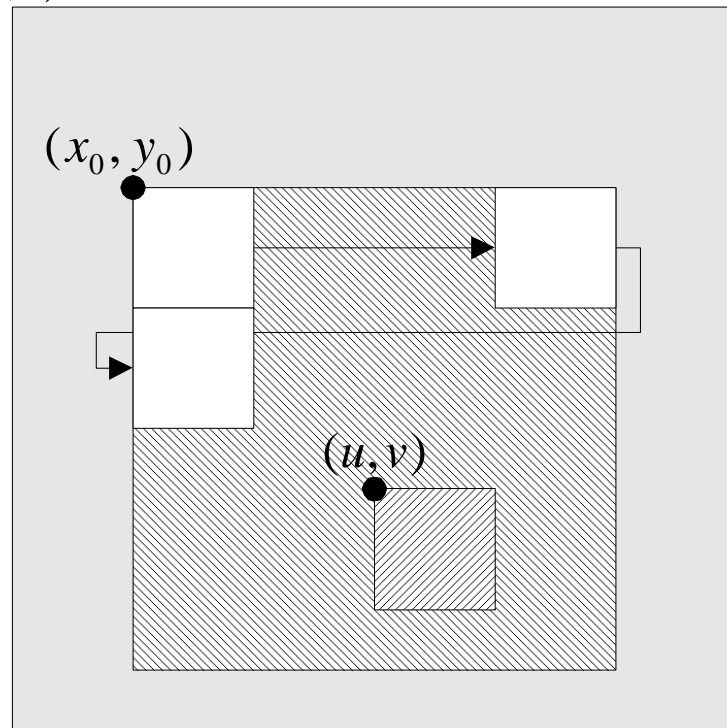
# 影像定位



$(x, y)$

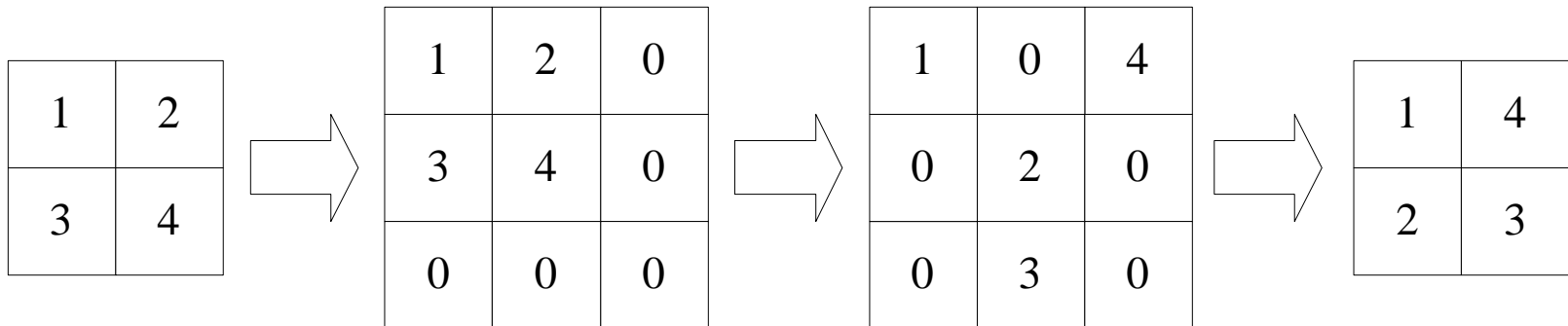


$(0,0)$



$$(x_0, y_0) = (x \div 8 - u, y \div 8 - v)$$

# 改良的甜甜圈演算法



$$\begin{pmatrix} 2 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \pmod{3} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 2 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \pmod{3} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 2 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \pmod{3} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 2 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \pmod{3} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

# 加快甜甜圈運算速度

※ 以對角化矩陣的觀念重新推導

$$\begin{pmatrix} x_t \\ y_t \end{pmatrix} = \begin{pmatrix} \varpi + 1 & \varpi \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_{t-1} \\ y_{t-1} \end{pmatrix} \pmod{N} \longrightarrow \begin{pmatrix} x_t \\ y_t \end{pmatrix} = \frac{1}{2^t} \begin{pmatrix} a + \varpi b & 2a \\ 2a & a - \varpi b \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \pmod{M}$$

其中當t為已知時，式中的矩陣  $\begin{pmatrix} a + \varpi b & 2a \\ 2a & a - \varpi b \end{pmatrix}$  為一常數

$t = \text{even number}$

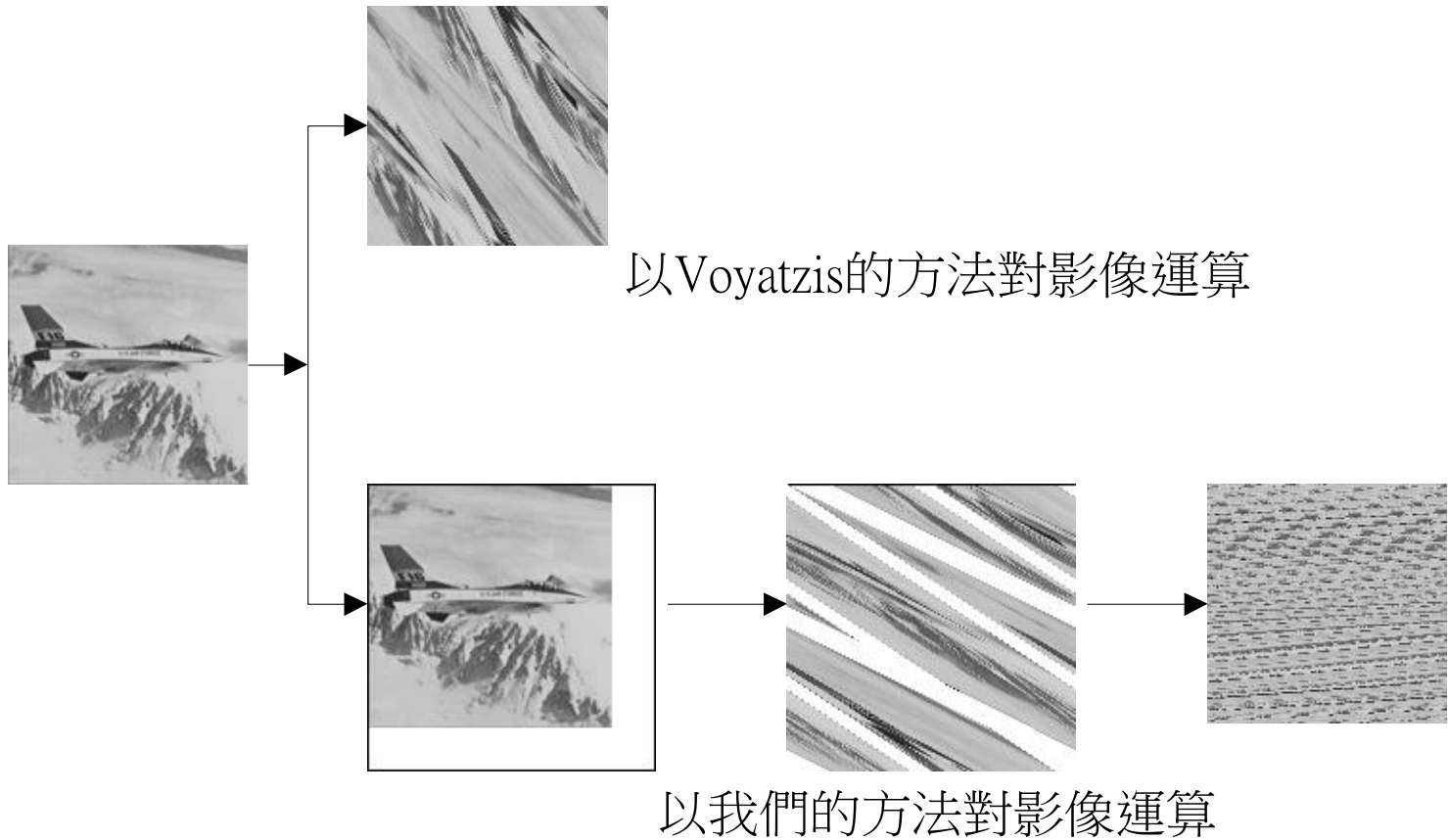
$$\begin{aligned} a &= \sum_{i=0}^{t/2} \binom{t}{2i} (\varpi + 2)^{2i} \Omega^{2i} + \sum_{i=2}^{t/2} \binom{t}{2i} (\varpi + 2)^{2i-4} \Omega^{4i-4} + \\ &\quad \sum_{i=6}^{t/2} \binom{t}{2i} (\varpi + 2)^{2i-6} \Omega^{6i-6} + \Lambda + \sum_{i=t-2}^{t/2} \binom{t}{2i} (\varpi + 2)^{2i-2} \Omega^{2i-2} + \frac{t}{t} (\varpi + 2)^0 \Omega^t \\ b &= \sum_{i=1}^t \binom{t}{2i-1} (\varpi + 2)^{2i-1} + \sum_{i=3}^t \binom{t}{2i-3} (\varpi + 2)^{2i-3} \Omega^2 + \sum_{i=5}^t \binom{t}{2i-5} (\varpi + 2)^{2i-5} \Omega^4 + \\ &\quad \sum_{i=7}^t \binom{t}{2i-7} (\varpi + 2)^{2i-7} \Omega^6 + \Lambda + \sum_{i=t-3}^t \binom{t}{2i-3} (\varpi + 2)^{2i-3} \Omega^{t-4} + \sum_{i=t-1}^t \binom{t}{2i-1} (\varpi + 2)^{2i-1} \Omega^{t-2} \end{aligned}$$

$t = \text{odd number}$

$$\begin{aligned} a &= \sum_{i=0}^t \binom{t}{2i+1} (\varpi + 2)^{2i+1} \Omega^{2i+1} + \sum_{i=2}^t \binom{t}{2i+1} (\varpi + 2)^{2i-1} \Omega^{2i-1} + \sum_{i=4}^t \binom{t}{2i+1} (\varpi + 2)^{2i-3} \Omega^{2i-3} + \\ &\quad \sum_{i=6}^t \binom{t}{2i+1} (\varpi + 2)^{2i-5} \Omega^{2i-5} + \Lambda + \sum_{i=t-3}^t \binom{t}{2i+1} (\varpi + 2)^{2i-1} \Omega^{t-3} + \sum_{i=t-1}^t \binom{t}{2i+1} (\varpi + 2)^{2i-1} \Omega^{t-1} \\ b &= \sum_{i=1}^t \binom{t}{2i} (\varpi + 2)^{2i} \Omega^{2i} + \sum_{i=3}^t \binom{t}{2i} (\varpi + 2)^{2i-2} \Omega^2 + \sum_{i=5}^t \binom{t}{2i} (\varpi + 2)^{2i-4} \Omega^4 + \\ &\quad \sum_{i=7}^t \binom{t}{2i} (\varpi + 2)^{2i-6} \Omega^6 + \Lambda + \sum_{i=t-2}^t \binom{t}{2i} (\varpi + 2)^{2i-2} \Omega^{t-2} + \sum_{i=t}^t \binom{t}{2i} (\varpi + 2)^{2i} \Omega^t \end{aligned}$$

# 改良的甜甜圈演算法範例

---





# 實驗結果

---

- ※ 以任意圖形做為浮水印

- ※ 尺寸變化 (Scaling)

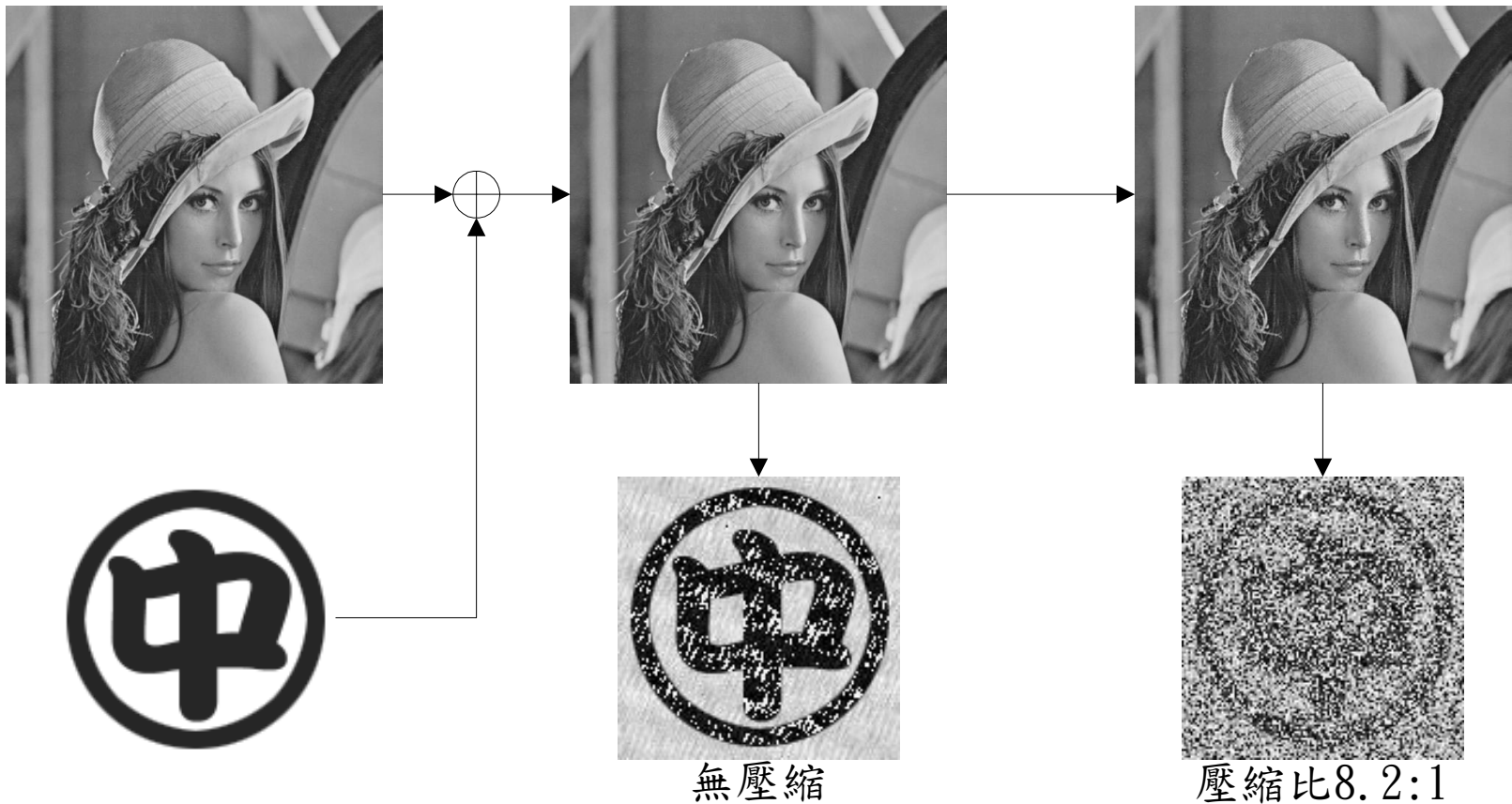
- ※ 平均化 (Blur)

- ※ 失真壓縮 (使用JPEG壓縮)

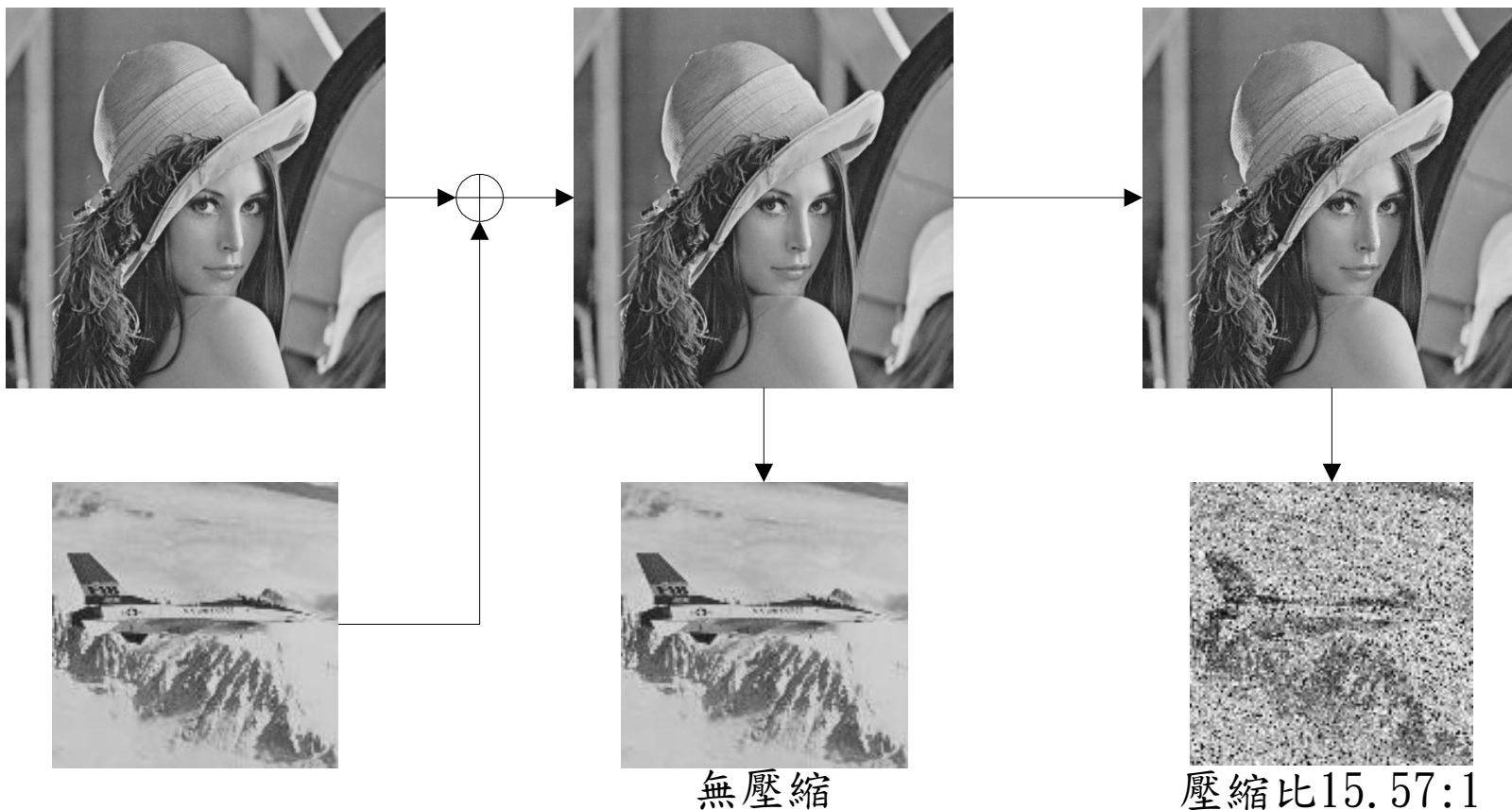
失真壓縮對隱藏在特徵區域和非特徵區域的的影響

隱藏在限制區域和全區域對只被盜用部分原圖影像的影響

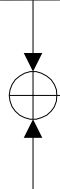
# 以自製灰階圖形為浮水印



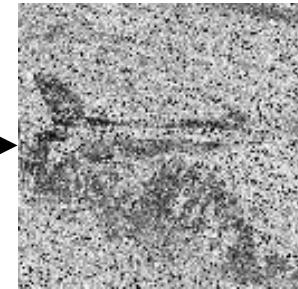
# 以戰機灰階圖形為浮水印



# 尺寸變化 (Scaling)

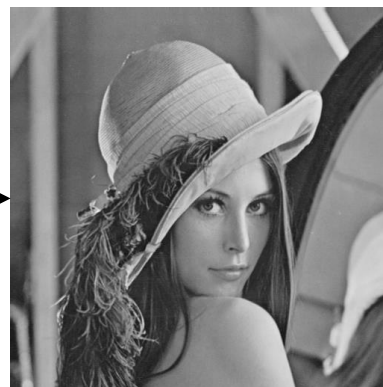


原圖縮小到 384x384  
再還原為512x512像素

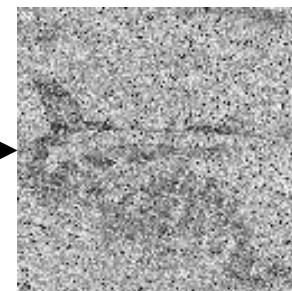


取回的  
戰機圖

# 平均化(Blur)

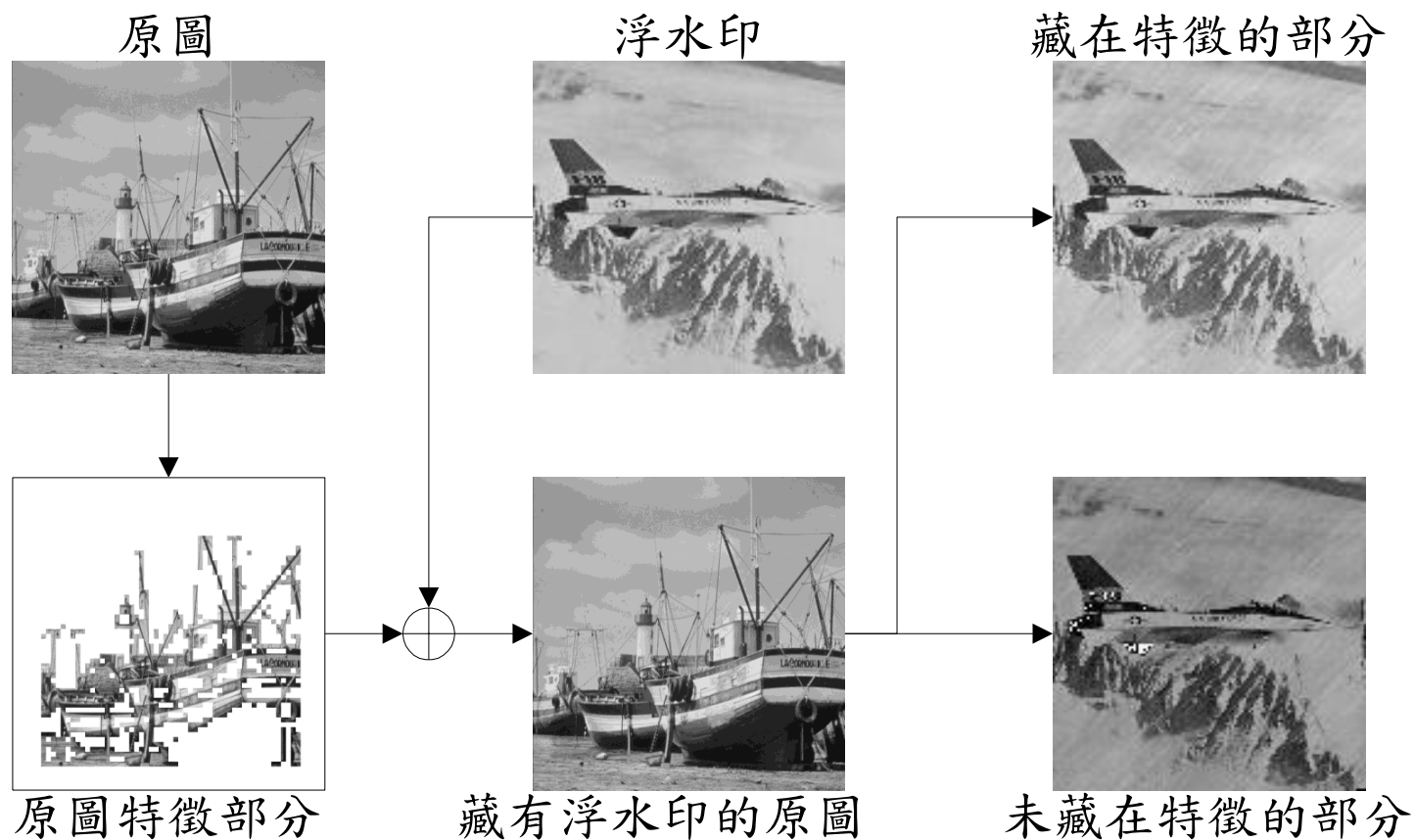


以3x3矩陣對原圖  
執行平均化處理



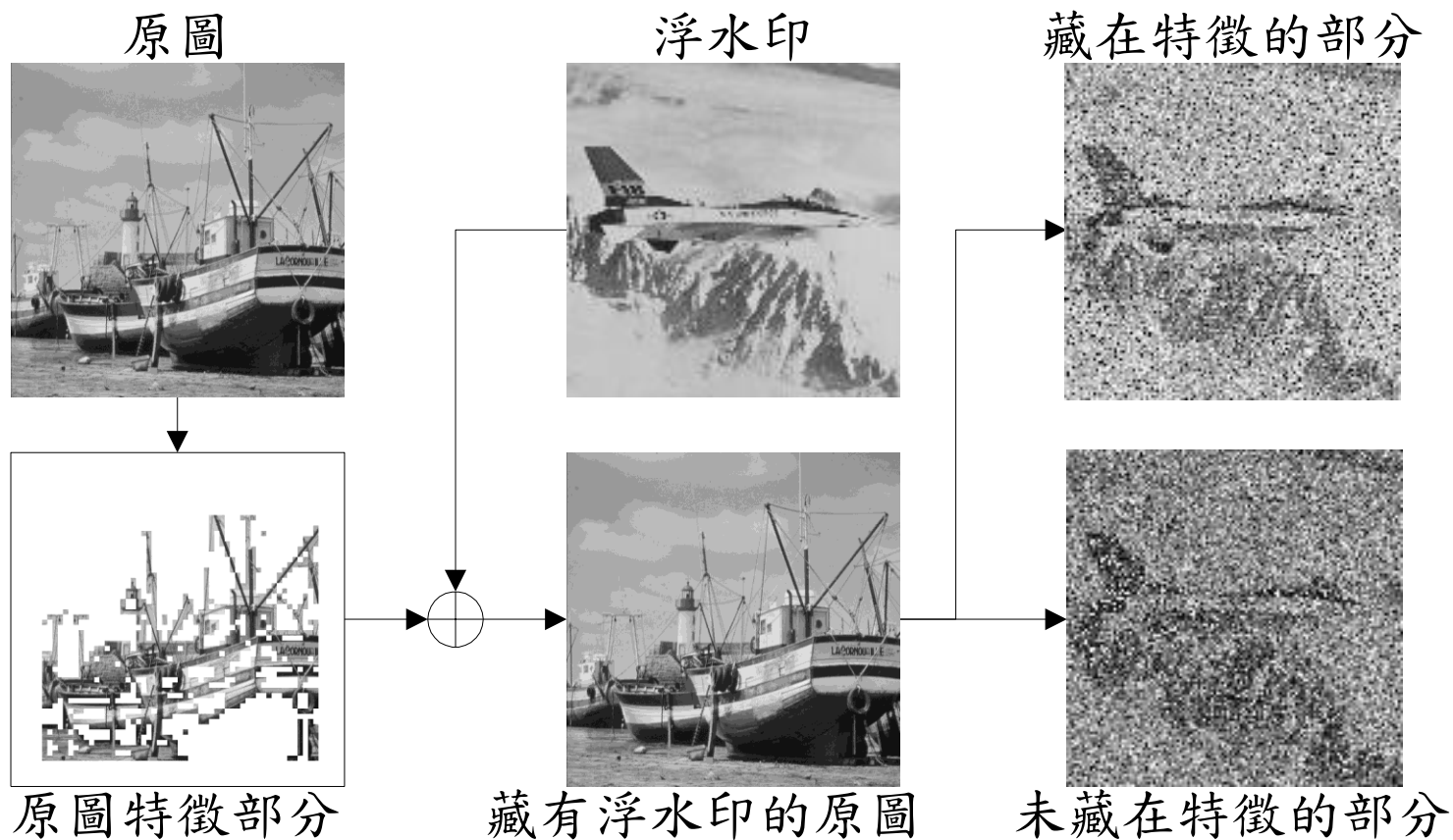
取回的  
戰機圖

# 隱藏在特徵區域（未壓縮）

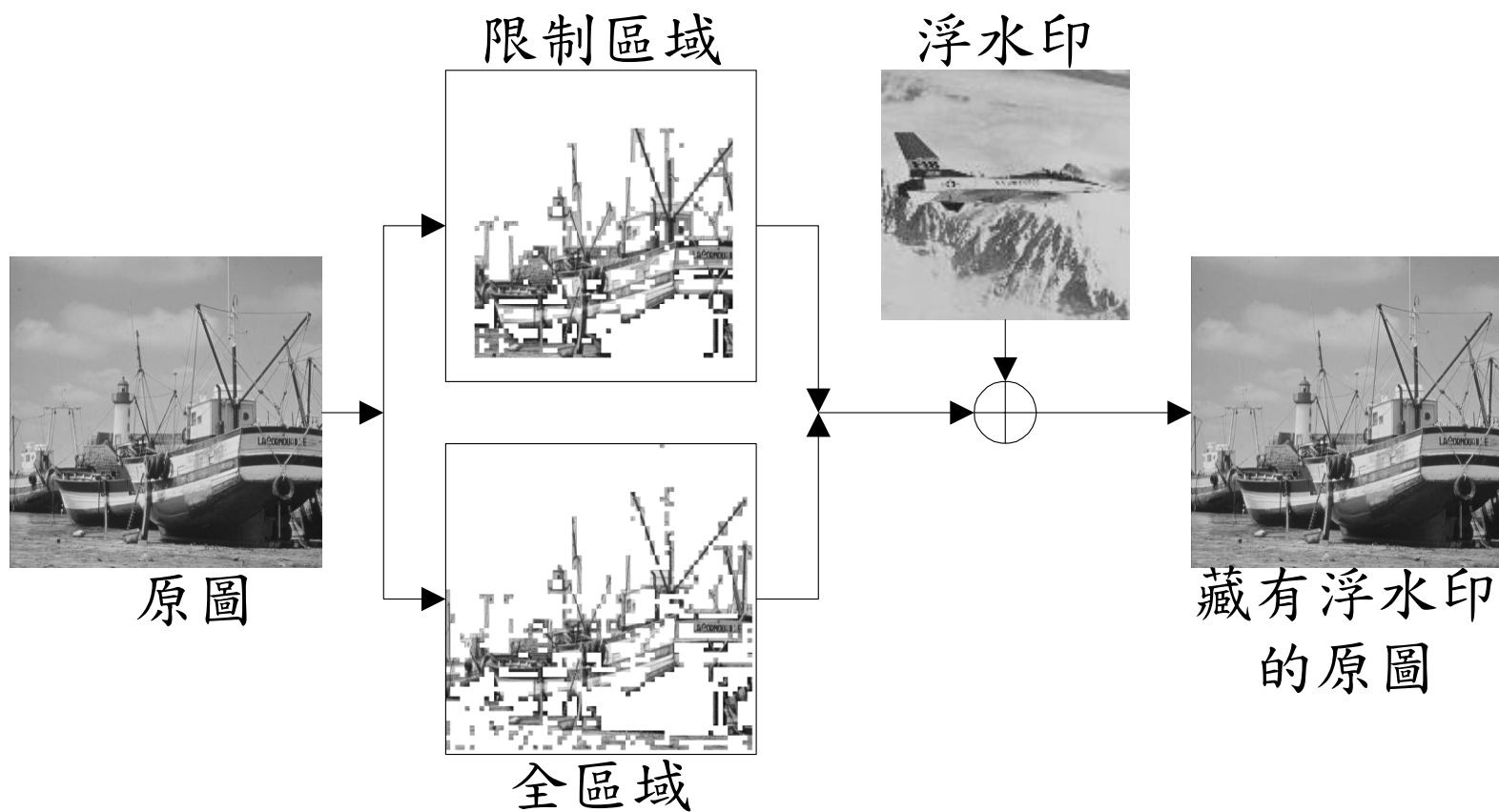




# 隱藏在重要區域（失真壓縮）

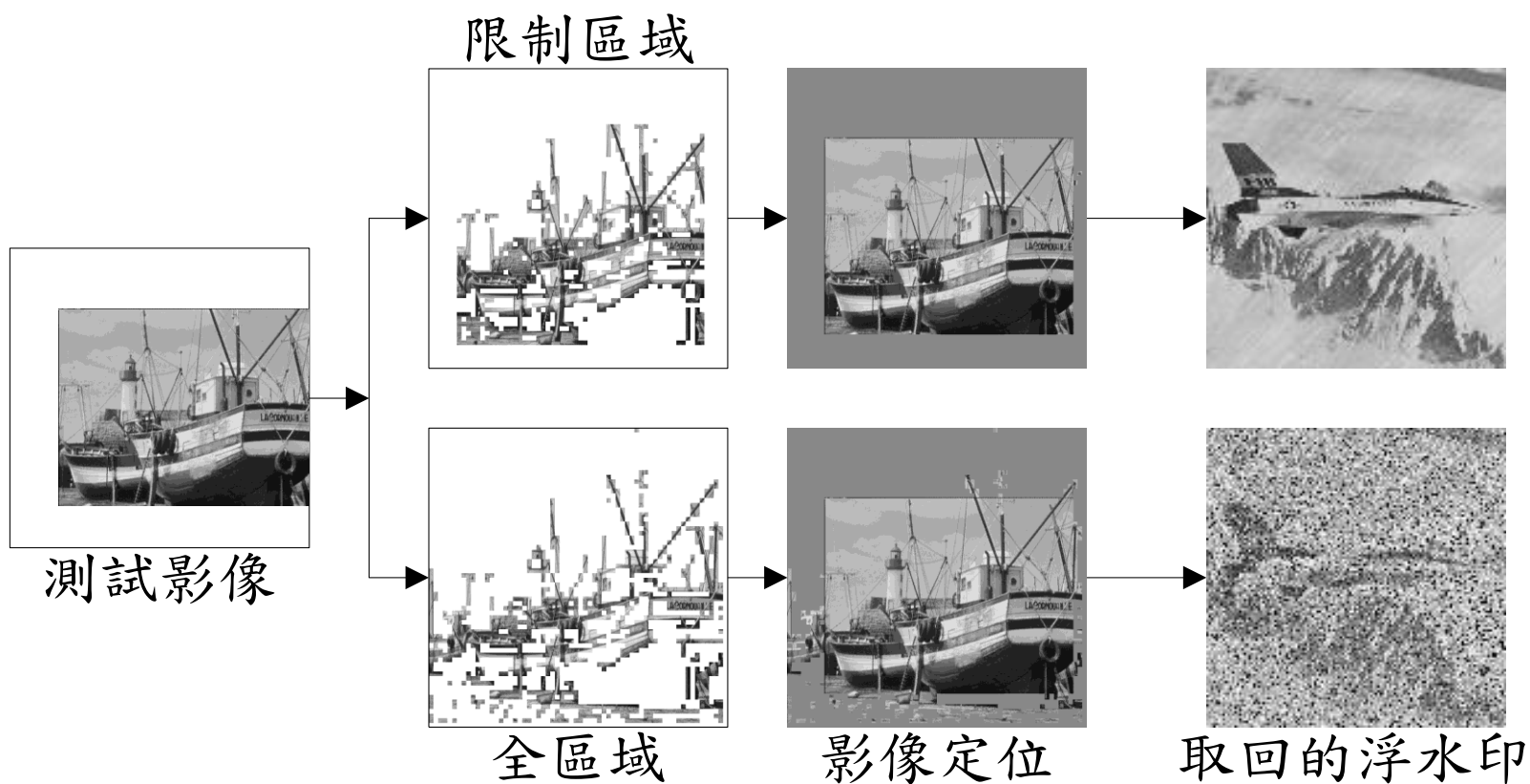


# 隱藏在限制區域和全區域的比較



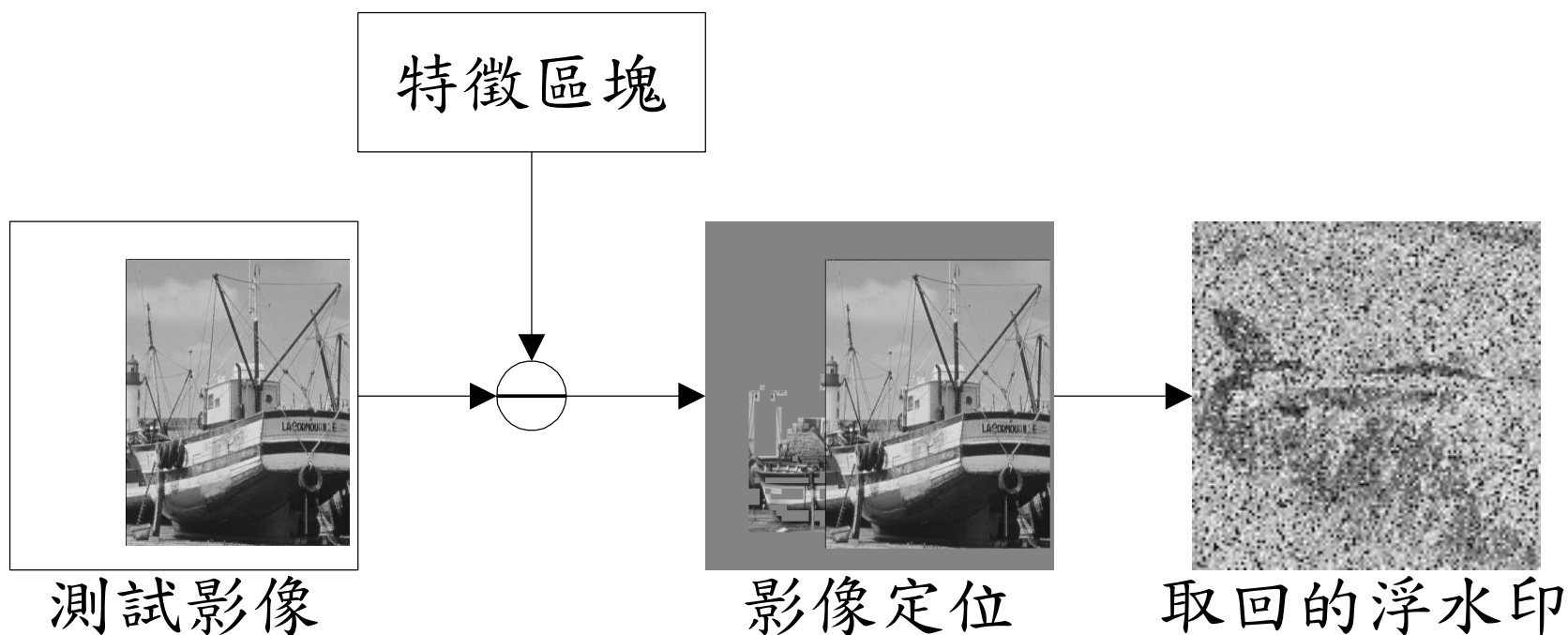


# 隱藏在限制區域和全區域的比較



# 隱藏在限制區域（部分）

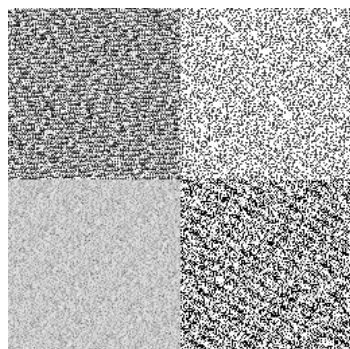
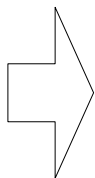
---



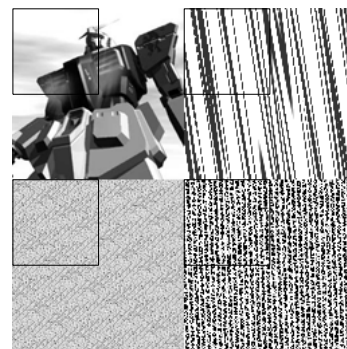
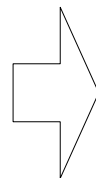
# 甜甜圈演算法應用到影像辨識



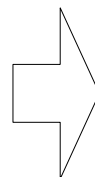
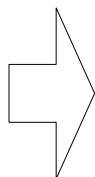
原始浮水印



存放在資料庫  
中的浮水印



以使用者鑰匙對資料庫中浮水印運算結果



1  
0  
0  
0

# 結 論

---

- ※ 本論文提出一個浮水印隱藏的整合方法，為了提高強韌度及透明度，我們將浮水印隱藏在影像中重要的部分。
- ※ 只保留重要部分的區塊資訊，使得我們的方法可以抵抗只有部分影像被盜用的情形，並且浮水印偵測只需要部分原圖，改善了浮水印偵測需要原圖的限制。
- ※ 更針對Voyatzis所提出的甜甜圈演算法加以改良並重新推導運算公式，使改良後的演算法具備較快速的運算速度、更混亂、更不易窺伺原圖面貌的特性。應用到浮水印辨識方面也有不錯的表現。