

TR—88—027

個人指紋鑑定法系統之設計
中央研究院資訊研究所
陶利守 黃俊雄

參 考 書
不 外 借

中研院資訊所圖書室



3 0330 03 000100 7

0100

中 央 研 究 院 資 訊 科 學 研 究 所
78.3.24
圖 書 室

一. 前言

指紋為人類身上最特殊的一項表徵，1988年英國人類學家葛爾登，佛蘭雪在實驗成果的發表中表示：世界上沒有二個人指紋是相同的。時至今日，此項特性已被證實無誤，且為世人廣泛地利用。諸如：犯罪指紋鑑定、指紋替代印鑑、利用指紋證明個人身份等等。關於指紋的特性，我們可歸納為：

(1) 獨特性 (Uniqueness)：

指紋人人不同，至今世上仍未發現不同人而擁有完全相同的指紋，就算是同卵雙胞胎的指紋也僅是紋路類似，仔細比對仍有很大的差別。

(2) 穩定性 (Stability)：

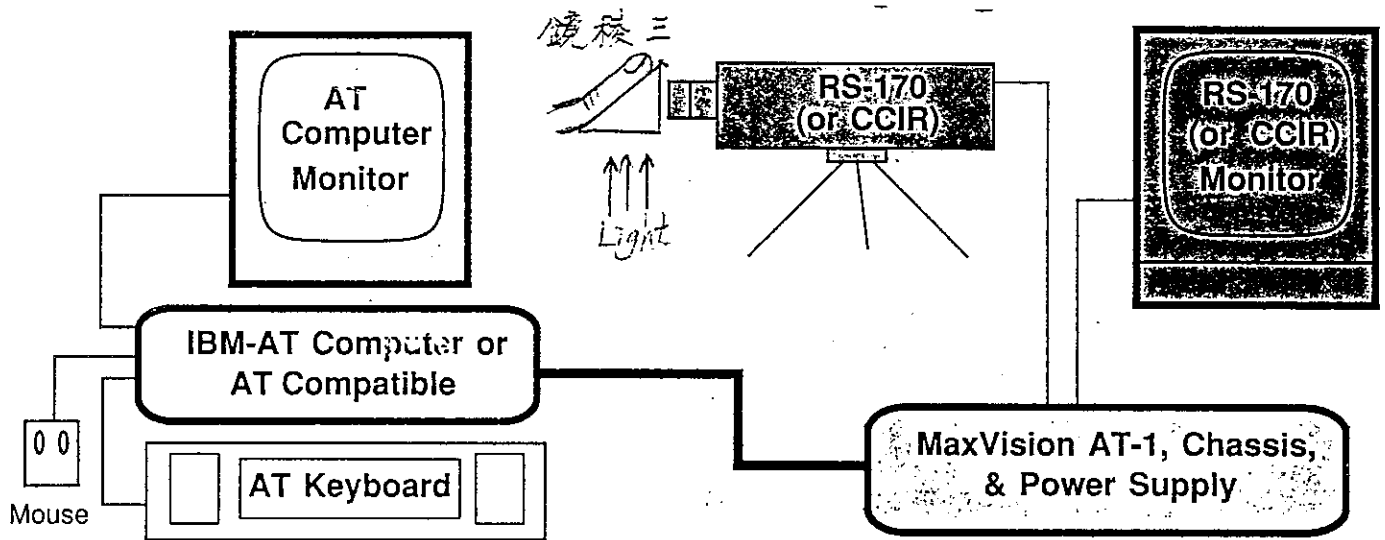
一個人出生到死，其指紋紋路特徵永遠不變，僅紋線間距離長短略有變化而已。而手指表皮若有損傷，一經傷癒，指紋亦恢復原狀。

本報告所要提出的是一種簡單而有效的指紋一對一辨認法，適用於小型系統，能在時下個人電腦上運作。這並不是用於在大量指紋中尋找特定對象的指紋工作，而是用在安全檢查或財物管理的個人指紋鑑定工作上。

由於罪犯指紋鑑定方法與本篇研究較為不同，故不予討論。

民國七十六年一月我們曾提出一篇相同的研究報告，但是由於辨認指紋的時間過長（約一小時五十分/每個指紋），無法適用於實際的應用上。故本篇延自上篇研究，利用特殊硬體，並將原方法略作更改，寫成此篇。關於上一篇報告見附錄[1]。

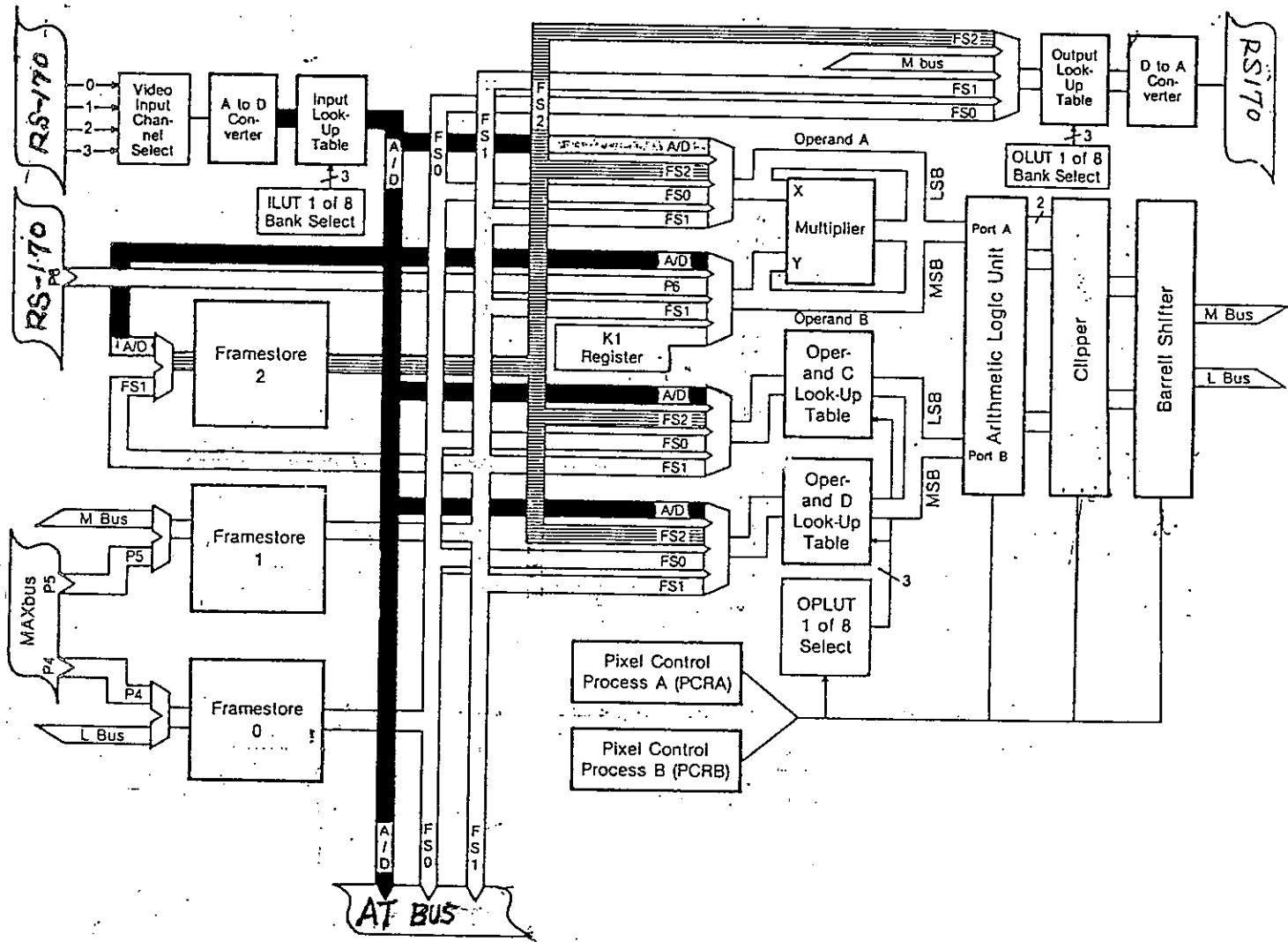
二. 系統設備介紹



圖一. 系統設備

如圖一，本系統係以 IBM PC 主機加上另一套專門處理影像的硬體，這套硬體包含攝影機、AT-1 處理器與螢幕監視器。指紋壓在三稜鏡上，經過攝影機攝取影像，呈現在 RS-170 螢幕上，經過 AT-1 處理器的計算，其中透過 PC bus 到 PC 主機上擷取或寫進資料，以便作更進一步的運算，最後將結果經過判斷，決定比對是否成功。

三. 硬體介紹



圖二. AT-1 處理器結構

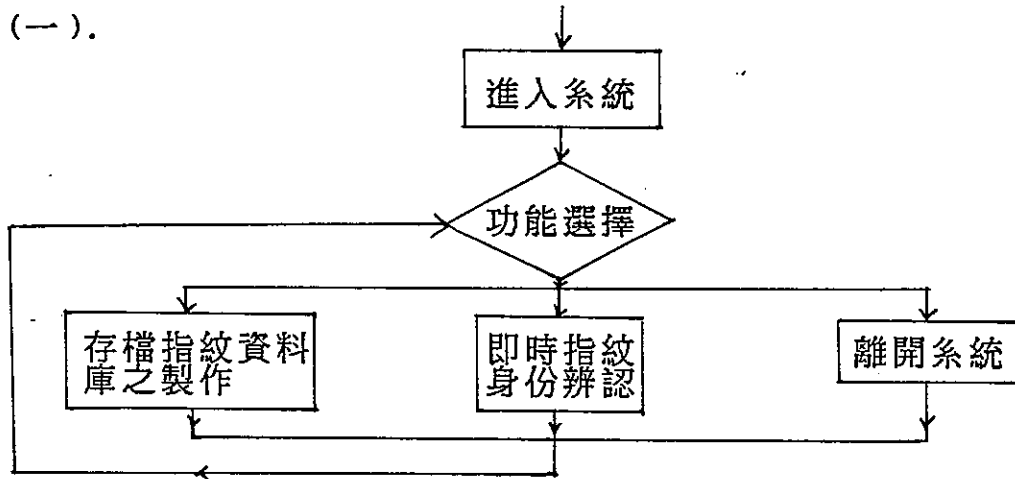
上圖二為 AT-1 處理器之結構，分別介紹如下：

- 輸入/輸出：使用 RS-170/CCIR 訊號、配備四個影像輸入通道可供選擇，輸入影像可自由控制增減，使用八位元的 A/D、D/A、有八個 ILUT 槽，八個 DLUT 槽。
- Framestore：有三個 $512 \times 512 \times 8$ 位元的 Framesstores (F2, F1, F0)，而其中 F1 與 F0 可合併為十六位元的 Framestore 使用。
- 數學運算：有一個十六位元的乘法器，二個運算 LUT (OPLUT)，十六位元的 ALU，十七位元的 Clipper，十六位元的 Barrel Shifter。
- 像素控制：為使用 Pipeline 點對點運算，運算速度 10MHz (100 nanosecond/ 每個像素)，可達到即時的效果。有二組可調式運算碼 (switchable operation, code SOC) 可供選擇作特定資料運算的工作。

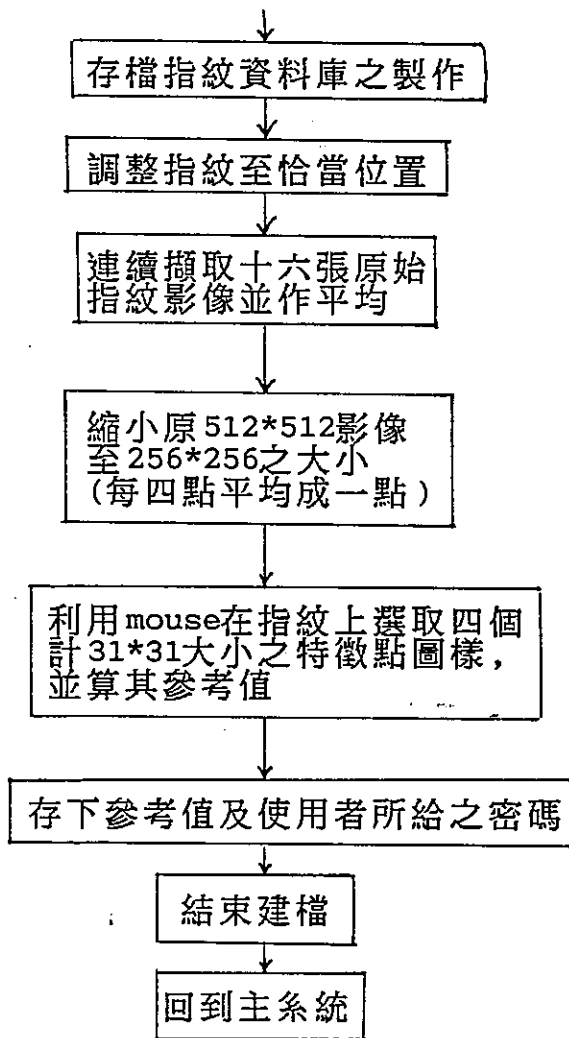
四. 系統介紹

系統流程如下：

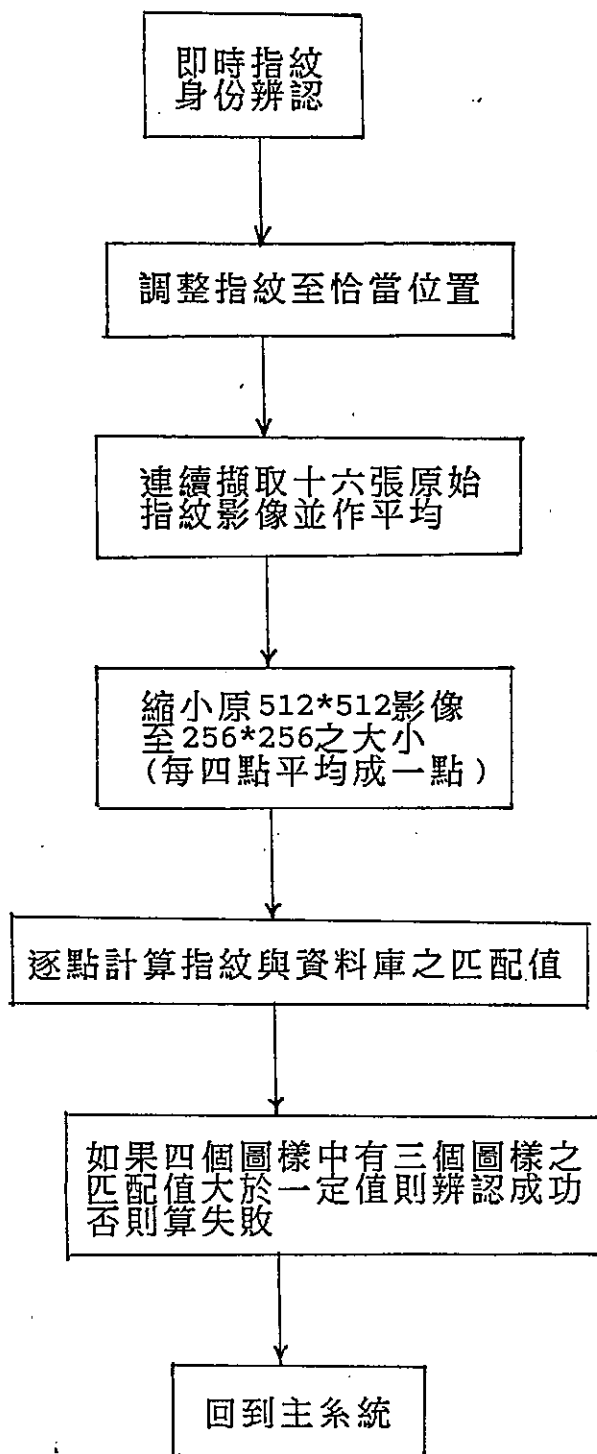
(一).



(二).



(三).



五. 圖樣辨認法

本系統中所使用的辨認方法為圖樣辨認法 (Template matching). 所謂圖樣辨認法就是以指紋影像的特徵點為中心, 取下一塊區域作為圖樣存起來. 要注意的是因為每次按指紋時, 總會因壓重壓輕或手指出汗, 而有些許的差異, 為避免這些差異造成影響, 我們在擷取指紋時, 連續擷取十六張原始影像, 而後將這十六張影像作一平均, 以減少雜訊.

此外, 所謂特徵點是指交叉點與端點. 見圖三所示. 在這裡我們是取四個圖樣, 而第一個圖樣通常是取中心點. 而圖樣的大小為 31×31 個像素. 見圖四所示.

除了圖樣的灰度值外, 再把各圖樣的中心點座標也存起來.

於是在辨認時, 如果是同一人的指紋, 原先存檔的該人指紋圖樣, 在所辨認的指紋影像裡應該都有, 而且其圖樣中心點座標, (換句話說即各圖樣的分佈位置) 也和存檔的中心點座標分佈狀況相同. 例如: 如果存檔的圖樣 1 中心點座標為 (X_{p1}, Y_{p1}) , 圖樣 2 的中心點座標為 (X_{p2}, Y_{p2}) . 而在被辨認的指紋圖像裡, 圖樣 1 所辨認到的中心點座標為 (X_{q1}, Y_{q1}) , 圖樣 2 的中心點座標為 (X_{q2}, Y_{q2}) , 則 P 影像與 Q 影像兩者的關係應該為

$$(X_{p2} - X_{p1}, Y_{p2} - Y_{p1}) = (X_{q2} - X_{q1}, Y_{q2} - Y_{q1})$$

$$\text{若令 } U_1 = X_{p2} - X_{p1}, U_2 = X_{q2} - X_{q1}$$

$$V_1 = Y_{p2} - Y_{p1}, V_2 = Y_{q2} - Y_{q1}$$

由於指紋按壓時變形產生誤差的關係, 我們只能要求

$$|U_1 - U_2| \leq t, \quad |V_1 - V_2| \leq t \quad \dots \dots \dots (1)$$

t 即在誤差下所允許的差量範圍, t 可由實驗得知. 接下來的問題是如何才能決定某圖樣是否辨認到它的位置, 這是測量函數的問題, 我們所

用的函數如下：

$$\text{匹配值} = \frac{\sum_{i=1}^N (P_i - \bar{P})(Q_i - \bar{Q})}{\left(\sum_{i=1}^N (P_i - \bar{P})^2\right)^{1/2} \left(\sum_{i=1}^N (Q_i - \bar{Q})^2\right)^{1/2}} \dots\dots\dots(2)$$

其中 P_i 為圖樣依序排列之各像素灰度值。

Q_i 為欲辨認指紋影像裡相對應之各像素灰度值。

N 為圖樣所包含之像素個數。

$$\bar{Q} = \frac{1}{N} \sum_{i=1}^N Q_i$$

$$\bar{P} = \frac{1}{N} \sum_{i=1}^N P_i$$

此匹配值對影像的 gray level 之 scaling 具有不變性，並且對 shift in gray level 具有不變性。

以實際所用之設備，所處理之影像為 512*512 像素之大小，經縮小至 256*256 大小（便於計算與減小誤差），而每個圖樣為以特徵點為中心，向上下左右各延伸 15 個像素所構成的區域，其大小為 31*31 像素，每個像素其灰度值介於 0 到 255 間， N 值大小為

$$N=31*31=961$$

匹配值所算出的結果介於 -1 和 1 之間，1 表示所欲辨認的影像區域和存檔圖樣完全吻合，或是各對應像素之灰度值都差同一放大倍數，其顯現圖形仍然完全一致；-1 的意義正好相反。當某個圖樣辨認到它的位置，即是在欲辨認影像裡對各像素都作一次匹配值計算，找出其值最高者即為該圖樣所辨認到位置，另一方面也要求匹配值需大於某常數 c ，否則該圖樣之辨認仍算失敗。

在此系統中，乃是對每個人的指紋取四個圖樣檔，取圖樣時須找指紋中最具特色變化最多的四個區域，亦即如前面敘述的特徵點。圖樣1乃找其紋路中心，往往也是最具特色的一點。

由於匹配值計算的繁複及像素灰度值輸出入的頻繁，若四個圖樣都作全面的逐點辨認，則將耗費許多時間，於是為了節省時間只對圖樣1作全面逐點辨認（實際作業上係要求使用者把指紋中心點按在鏡頭中心，如此圖樣1只在中心附近一小範圍內比對即可），其餘三個圖樣依其對圖樣1之相對位置，依式（1）在某差量範圍內逐點辨認即可。根據系統需求，我們對圖樣1所要求的匹配值是必須大於0.5，其餘三個圖樣其匹配值須大於0.45。

圖樣辨認法之演算法如下：

假設：

(X1, Y1) 為存檔圖樣 1 之中心點座標；

(X2, Y2) 為存檔圖樣 2 之中心點座標；

(X3, Y3) 為存檔圖樣 3 之中心點座標；

(X4, Y4) 為存檔圖樣 4 之中心點座標；

SCORE 之初值為 0。

演算法：

(1) 取出存檔圖樣1之資料對欲辨認指紋影像作全面逐點計算，取其最高匹配值 M1 及其座標 (a, b)。

(2) 若 $M1 < 0.5$ ，則至 (4)。亦即表該指紋非同一人。

(3) 對圖樣2，圖樣3，及圖樣4分別做下列程序：

$$(a) \quad U_i = X_i - X_1 \quad V_i = Y_i - Y_1$$

$$(b) \quad p = a + U_i \quad q = b + V_i$$

以 (p, q) 為中心，上下左右各延伸一個差量 t 所成的範圍內對圖樣 i 做逐點計算，取其最高匹配值 M_i 。

(c) 若 $M_i > 0.45$ 則 $SCORE = SCORE + 100$

(4) 若 $SCORE < 200$ 則不為同一人之指紋，否則為同一人指紋。

上述演算法之解釋如下：除圖樣 1 須匹配值須達 0.5 以上外，其餘三個圖樣也至有兩個圖樣在其相對位置差量範圍內匹配值達 0.45 以上才為同一人之指紋，否則即非同一人之指紋。（亦即表示四個圖樣必須至少有三個辨認成功，而第一個圖樣必須辨認成功，滿足上述條件者則判為同一人指紋，否則不為同一人之指紋）。其餘三個圖樣之所以不要求其匹配值達 0.5 以上，及因按指紋時即使再小心，也多少會有一點變形，為避免因此造成誤判，故只要求達 0.45 以上即可。

六. 演算法之修正

上節之演算法雖能滿足系統之需要，但費時甚鉅，無法做到即時辨認的地步。在第三節硬體介紹中，我們知道圖樣之計算及灰度值之存取，若經由AT-1處理器處理，速度遠較PC DOS運作迅速許多。另外，仔細觀察

(2)式匹配值之計算式，可發現： \bar{P} 和 $\sum(P_i - \bar{P})^2$ 因屬存檔資料，所以在計算時屬已知資料，而 \bar{Q} 和 $\sum(Q_i - \bar{Q})^2$ 則對每一像素均須重新計算再代入式(2)以計算出匹配值，但每一影像往往包含數百數千個像素，且每當位移一個像素，再作匹配值運算時，其中絕大部份的像素是重複的，只極少數不同，故每次經由下式計算 \bar{Q} 和 $\sum(Q_i - \bar{Q})^2$ 會浪費太多時間。

$$\bar{Q} = \frac{1}{N} \sum_{i=1}^N Q_i \dots\dots\dots(3)$$

$$S_n = \sum_{i=1}^N (Q_i - \bar{Q})^2 \dots\dots\dots(4)$$

為避免太多的重複計算及方便特殊硬體運算，我們特將(3)及(4)式修正如下：



圖五. 在欲辨認之指紋Q影像上，圖樣向右移動一個像素，計算匹配值。右方陰影部份表增加之像素，左方陰影部份表減少之像素。

假設在欲辨認之指紋圖樣向右移動一個像素，如圖五所示。而原先移動前之圖樣所計算出之 S_n 與 \bar{Q} 如(3)，(4)式所示。

而Gain表圖樣移動一個像素後所增加的一行(或一列)像素之灰度值的和，而Loss表圖樣移動一個像素後所減少的一行(或一列)像素之灰度值的

和。

則移動後，重新計算的 為

$$\begin{aligned} \bar{Q}' &= \frac{961 \times \bar{Q} + \text{Gain} - \text{Loss}}{961} \\ &= \bar{Q} + \frac{(\text{Gain} - \text{Loss})}{961} \dots\dots\dots(5) \end{aligned}$$

而

$$\begin{aligned} S_n' &= \sum_{j=32}^{961+31} (Q_j - \bar{Q}')^2 \quad (S_n' \text{'s definition}) \\ &= \sum_{j=32}^{992} \left(Q_j - \bar{Q} - \frac{(\text{G}-\text{L})}{961} \right)^2 \quad (\text{replaced by Eq(5)}) \\ &= \sum_{j=32}^{992} \left[(Q_j - \bar{Q})^2 - 2(Q_j - \bar{Q}) \frac{(\text{G}-\text{L})}{961} + \frac{(\text{G}-\text{L})^2}{961^2} \right] \\ &= \sum_{j=32}^{992} (Q_j - \bar{Q})^2 - \frac{2(\text{G}-\text{L})}{961} (961 \bar{Q}' - 961 \bar{Q}) + \frac{(\text{G}-\text{L})^2}{961} \quad (\text{remove } \Sigma) \\ &= \sum_{j=32}^{992} (Q_j - \bar{Q})^2 - \frac{2(\text{G}-\text{L})^2}{961} + \frac{(\text{G}-\text{L})^2}{961} \\ &= S_n - \sum_{j=10 \text{ s s}} (Q_j - \bar{Q})^2 + \sum_{j=\text{gain}} (Q_j - \bar{Q})^2 - \frac{(\text{Gain} - \text{Loss})^2}{961} \dots\dots\dots(6) \end{aligned}$$

試比較(5)，(6)式與原先之(3)，(4)式：原先之(3)式需重新計算961個像素灰度值之和，而(5)式僅需計算62個像素 (Gain和Loss各31個像素)灰度值之和，不僅省下加法計算時間，更由於擷取螢幕像素的灰度值需要花時間(I/O time)，估計(5)式所花時間約為(3)式的1/15。而(4)式需計算961次像素平方和的運算，(6)式卻僅需處理62次平方和的運算，(6)式所花時間亦為(4)式的1/15(大約)。由於本系統整體運作的時間為數秒至數十秒之間，而式(3)又為計算的核心，故式(5)與式(6)之修正，對於即時系統之改進裨益良多

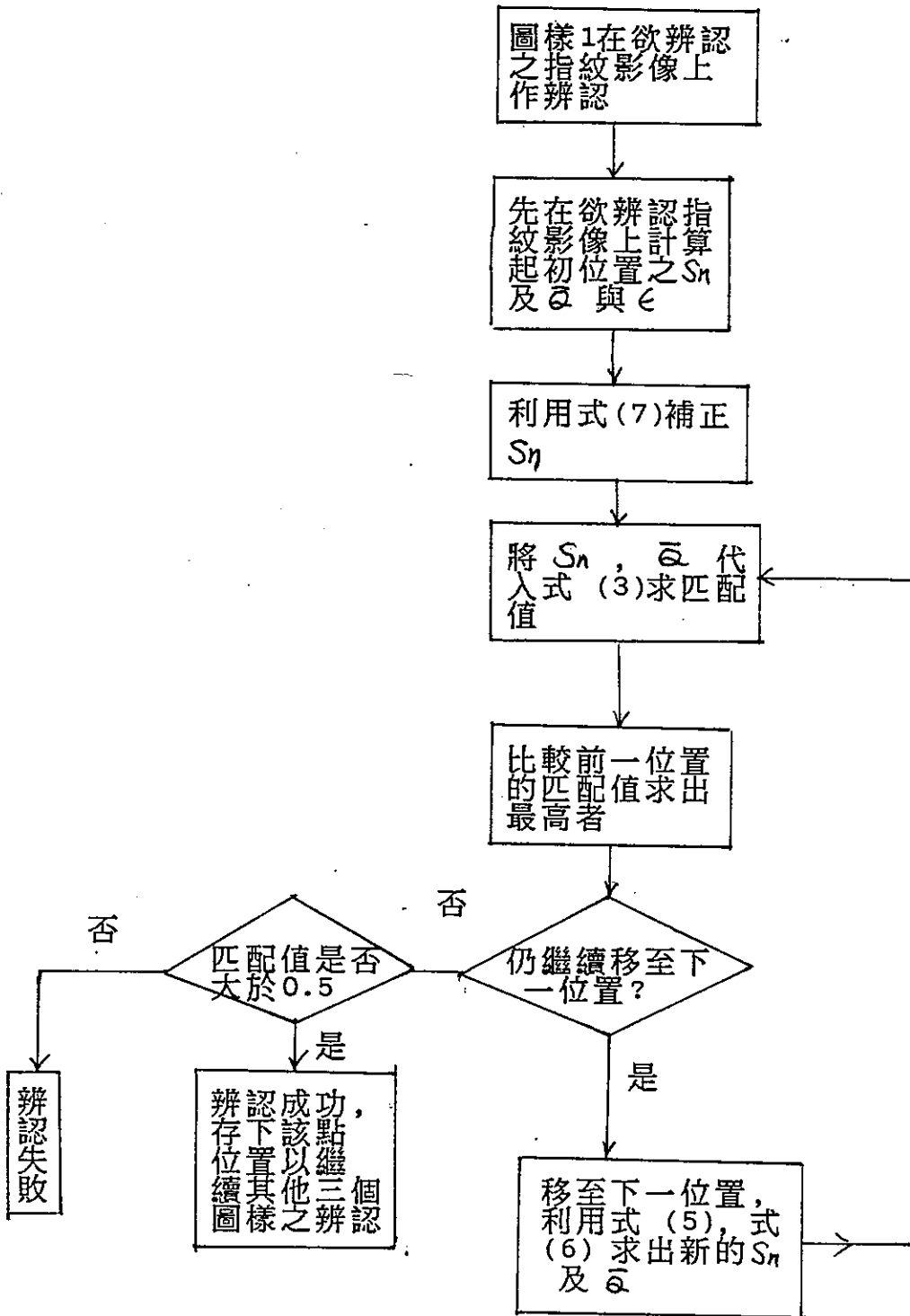
此外由於 $S_n = \sum_{i=1}^N (Q_i - \bar{Q})^2$ 我們是交由 AT-1 處理器的 framestore 先作一次相減： $Q_i - \bar{Q}$ 然後再自己相乘，將值取出相加而得出的。由於 framestore 僅儲存灰度值 0 至 255 的整數值，故經由 AT-1 所得出之 S_n 會因 \bar{Q} (為實數) 產生誤差，為作修正，我們計算如下：

若令 ϵ 表誤差， $\text{int}(\bar{Q})$ 表小於之最大整數， S_n' 表有誤差的結果， S_n 表實際值
則

$$\epsilon = \bar{Q} - \text{int}(\bar{Q})$$

$$\begin{aligned} S_n &= \sum_{i=1}^{961} (Q_i - \bar{Q})^2 \\ &= \sum_{i=1}^{961} (Q_i - \text{int}(\bar{Q}) - \epsilon)^2 \\ &= \sum_{i=1}^{961} (Q_i - \text{int}(\bar{Q}))^2 - 2\epsilon \sum_{i=1}^{961} (Q_i - \text{int}(\bar{Q})) + 961 \times \epsilon^2 \\ &= S_n - 2\epsilon [961 \times (\text{int}(\bar{Q}) + \epsilon) - 961 \times \text{int}(\bar{Q})] + 961 \times \epsilon^2 \\ &= S_n - 2\epsilon \times 961 \times \epsilon + 961 \times \epsilon^2 \\ &= S_n - 961 \times \epsilon^2 \dots\dots\dots(7) \end{aligned}$$

以上各式之應用，以圖樣1為例，計算之流程如下：



七. 實驗結果

系統完成後，經實驗結果證明：

(1) 不同人的指紋辨認率為百分之百，迄今尚未發現有誤認之情形，顯見系統之可靠度。

(2) 同人間的指紋辨認率受以下因素影響：

(i) 被辨認的指紋影像僅容許有限度的平移和旋轉。由於我們在三稜鏡上裝置一個指模，因此若非刻意斜放指紋，使其歪曲偏轉太多，系統均能正常辨認出該人指紋。

(ii) 系統為達即時的效果及指紋完整性，乃要求被辨人手指紋路中心必須落於螢幕中央的一小塊區域內，若偏離該區域則無法辨認出該人指紋。而螢幕中央區域為系統主觀認定範圍，範圍太小，則使用不便，範圍愈大，則辨認時間愈久，範圍若太大，則被辨人指紋之必要特徵點可能都脫離螢幕之外，而無法取得完整之指紋。故有必要規定一適當範圍。經實驗結果證實，凡中心點落內同一範圍內之同人指紋，均能成功地被辨認出來。

(iii) 有的人有所謂"汗手"的情形，亦即指紋常出現汗滴，若汗滴不至太多，沒有遮蔽重要的特徵點，則辨認不受影響，否則我們可要求手溼者或手過髒者重新擦拭後再行辨認。

經實驗結果證實：系統辨認所花時間最快僅需2秒，而慢則約需40秒，平均約花21秒的時間辨認一個指紋。

八． 發展方向及結論

由上面結果可知"圖樣辨認法"簡單可行，尤其百分之百的辨認不同人指紋的比率，更適合用於安全系統上，但由於耗時仍需約半分鐘左右，用於門禁管制上，仍有其不便之處，尤其門禁管制，人員進出頻繁，目前之磁卡，密碼等方式，足敷使用。故本系統較適用於使用不過於繁頻之安全系統上，例如，銀行之保險櫃，貴重物件之寄放箱，特殊建築物之進出等。

另外由於三稜鏡在辨認後需要擦拭乾淨，更由於會受霧氣，燈光熱度等因素，影響指紋之顯示，常常需要取下處理，殊多不便，未來若能改進輸入裝置，改由紅外光線，或其他更先進的輸入技術，則可節省許多的工作。



圖六．輸入之指紋影像



圖七．處理後之結果，由螢幕可看出處理時間為29秒。



圖八.系統設備圖.左為PC/AT.中為AT-1處理器及螢幕,右為三稜鏡及攝影機所組合之裝置.

九. 參考資料

1. 黃俊雄. 鍾克平: 可靠的個人指紋鑒定法, 中央研究院資訊所, 民76年(1987)1月, TR-87-003.
2. 許文星. 劉文卿. 鍾志裕: Automatic Fingerprints Identification System With Non-ink Input Method, 全國計算機會議, 民國74年(1985), 760-7
3. Malcom K. Sparrow and Penelope J. Sparrow "A Topological Approach to the Matching of Single Fingerprints: Development of Algorithm for Use on Rolled Impressions, " Computer Science and Technology, May 1985.
4. Malcom K. Sparrow and Penelope J. Sparrow "A Topological Approach to the Matching of Single Fingerprints: Development of Algorithm for Use on Latent Fingerprints " , Computer Science and Technology, October 1985.
5. A. Shimizu and M. Hase, "Entry Method of Fingerprint Image Using Prism", Trans, of IECE, JAPAN, Vol. J67-D, No.5, May 1984, pp627-628.
6. A. Shimizu and M. Hase, "Entry Method of Fingerprint Image Using Prism-A Comparison Between Total Reflection and Light Splitting", Trans.of IECE, Japan Vol. J 68-D ,No.3, March 1985.
7. "AT-1 maxvision manual", Data Cube, 1988, 3.