

以簡單線性迴歸分析實作掌紋辨識系統

虞台文

大同大學資訊工程系

email: twyu@cse.ttu.edu.tw

李佳勳

大同大學資訊工程系

email: jamzero@gmail.com

摘要

現今，以生物辨識做為身分識別越來越被被人們所接受，其中手部辨識擁有高度方便性，未來勢必以手部辨識為主。

本研究將實作一個掌紋辨識系統，並且提出一些新的想法用於掌紋辨識，首先改進USAN這個擷取掌紋的方法，再提出一個快速搜尋掌紋主線的方法，並且利用簡單線性迴歸將掌紋特徵數值化。辨識系統使用USAN將原始掌紋影像轉換成一個三維空間，稱為USANarea，再使用threshold將掌紋擷取出來，擷取出來的掌紋經過去雜訊與close處理後，轉換成以線段表示的圖形，再搜尋出三條掌紋主線，掌紋主線分別經過簡單線性迴歸分析，最後整張手掌影像將以六個參數代表，大幅降低資料庫的複雜度，處理過程儘量減少複雜的運算，整個辨識過程可real-time完成。

關鍵詞：掌紋辨識 (Palmprint recognition)；掌紋主線 (Typical principal lines)；簡單線性迴歸分析 (Simple linear regression)；USAN (univalue segment assimilating nucleus)。

1 前言

在過去，身分辨識的主要方法是以鑰匙、密碼、磁卡、感應卡等工具為主，它們有很多缺點，例如：容易被複製、容易遺失或被竊、密碼則有遺忘的可能，在使用上有很多不方便。目前逐漸發展起來的，是以生物辨識為主的身分辨識系統，主要概念是以人體器官作為辨識的依據，因器官特徵具有唯一性、不變性、可攜性與持久性，可以彌補傳統身分辨識工具的不足，所以生物辨識的相關研究即成為熱門的研究題材。

目前生物辨識主要採用：手指（指紋）、手掌（掌形與掌紋）、眼睛（虹膜）等人體器官（特徵）做為辨識的依據。此外如：臉型、聲音、骨骼結構、靜脈掃描...等，也曾被研究當做生物辨識的特徵。

一個生物辨識系統是否被大眾接受，取決於：辨識速度、認證正確性、防偽性、是否受膚色影響、是否受環境影響（光、溫度、器材）、對於使用者是否方便，目前還沒有一個完美生物辨識技術，而基於器材的成本與辨識的方便度，使用者比較能接受利用手部進行身分辨識，因此很多專家學者投入研究手部特徵的生物辨識技術。

首先介紹手掌相關名詞，見圖 1，手掌中的紋路我們通稱為『掌紋』(Palmprint)，手掌彎曲的時候特別明顯。掌紋的深淺與走向，都是由遺傳來決定的，其中三條較粗的『主線』(Typical Principal Lines)分別稱為生命線、智慧線與感情線，它們在胎兒發展的頭幾個月就已經開始形成，主線在成年之後就不會有太大的變動；其他較細的『細紋』(Wrinkles)是隨著成長才慢慢出現；有時候一條主線會出現中斷的情況，中斷的部分稱為『間隙』(Gaps)。而手指的關節稱為『指節』(Knuckle)；手指之間相連的部分稱為『指蹼』(Finger-Web)；手掌的大小與形狀稱為『掌形』(Palm shape)。

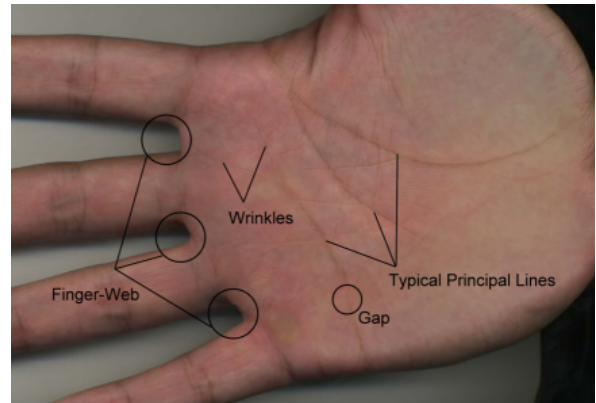


圖 1: 手掌與掌紋名詞介紹

掌紋辨識的相關研究於近十年蓬勃發展，香港理工大學的張大鵬 (David Zhang) 教授是此領域的權威，他於1998年發表了全球第一個掌紋辨識系統，並於2003年改進為高辨識率且低成本的掌紋辨識系統 [8]，主要概念是透過攝影機拍攝手掌影像，自動切割手掌心影像後，透過2D Gabor filtering偵測掌紋位置，最後以normalized Hamming distance作為比對數值的方法，實驗結果可達98%的正確接受率 (Genuine Accept Rate)，與0.04%的錯誤接受率 (False Accept Rate)。

張教授所領導的研究團隊建立了目前最大的掌紋資料庫—PolyU Palmprint Database [6] (蒐集了三百多人的手掌，共七千七百張手掌影像)，並且發表了很多關於掌紋辨識的論文：1999年首先發表的論文 [7] 是利用擷取掌紋線段的方式，將手掌的紋路，轉換為一系列的線段，經由線條比

對的方法，來驗證資料的正確性；2002年發表的論文[4]是利用Karhunen-Loeve transform將原始掌紋影像轉換成較小的特徵空間—“eigenpalms”，再以Euclidean distance classifier作為比對數值的方法。此外2005年提出了一個有效偵測掌紋的方法[3]，利用一個圓型的遮罩，於掌紋影像的每一個pixel運算一次，即可以簡單的threshold將掌紋偵測出來，並且比舊有的方法更有效。2004年則提出了一個將掌紋分類的論文[9]，利用指蹠與手指根部做為定位點，有效定位三條主線，進而分類成六類掌紋，包含斷掌、無生命線、無感情線…等特殊掌紋，若掌紋辨識加入此分類法，或許可以提高辨識率。

2006年新加坡南洋理工大學也發表了掌紋辨識的論文[2]，首先將掌紋線段從掌紋影像中擷取出來，再將線段以Line-based Hough Transform（類似Hough Transform的概念）分群，使得位置相近的線段落於同一群，以降低掌紋辨識的資料複雜度。

綜觀上述學術研究可以發現，掌紋辨識技術大多以主線特徵做為辨識的依據，若要達到比較高的辨識率，掌紋特徵必須取自手掌的相同區域，並且從手掌影像中有效擷取主線的特徵。本研究利用指蹠作為訂定辨識區域的依據，可以從同一人的不同掌紋影像中取得相同的辨識區域。在特徵擷取方面，我們改進一個曾經提出的掌紋擷取技術，去除其中複雜的運算，再加上去雜訊與close處理，得到的掌紋資訊經由我們所提出的新方法，快速判斷三條掌紋主線的位置。最後我們將三條掌紋主線分別進行簡單線性迴歸分析，以線性方程式的參數代表掌紋的特徵，此種方法只需要比較參數間的差異，即可判斷出身分，大幅降低運算的複雜度，加快辨識的速度，並且可以有效減少資料庫所需的儲存空間。本研究將以實作的角度，詳細說明掌紋辨識所需的技術，文末將顯示辨識率與執行時間，並與其他研究做比較。

2 掌紋特徵數值化

本研究在進行特徵比對的運算前，先將手掌影像中的掌紋特徵擷取出來，並且將特徵以數值表示。由於手掌並沒有固定於手掌影像擷取裝置上，因此每一次取得的手掌影像中，手掌位置沒有固定，因此必須先從手掌影像中切割一塊固定的區域以供後續處理。而手掌中的細紋會隨著年齡的增長而有所變化，因此我們只擷取主線的特徵，整個掌紋特徵數值化的過程主要分為四個步驟，敘述如下：

1. 取得手掌影像：採用 PolyU Palmprint Database[6]的手掌影像，此影像為灰階影像，解析度為 384×284 ，本研究只取用其中的右手影像。
2. 取得掌紋影像：從不同的手掌影像中訂定相同的辨識區域，並且將辨識區域從手掌影像中切割下來。

3. 掌紋擷取：偵測掌紋影像中的掌紋位置，並且大致去除細紋資訊，強化主線資訊，最後以binary image表示。
4. 數值化：從binary image中搜尋主線資訊，並且將主線特徵轉換為以數值表示。

2.1 取得掌紋影像

為了可以快速又有效的訂定辨識區域，本研究參考論文[8]的方法，以指蹠作為訂定辨識區域的依據，主要概念是搜尋手掌影像中的兩個指蹠位置，將這兩個指蹠位置連線以後，在連線的中垂線固定長度上取得一個正方形的範圍，這個範圍就是所需的辨識區域，我們將這個範圍的影像從手掌影像中切割下來，並且旋轉成正常的角度，稱為掌紋影像。

在此提出的方法完全由電腦程式自動處理，不需要人工的介入，詳細步驟如下：

1. 利用threshold，將原始影像（圖 2a）轉換為 Binary image（圖 2b），白色代表手掌影像，黑色代表非手掌影像。
2. 以 Connected Component Labeling[5]搜尋白色區塊的邊緣，即手掌的邊緣，並以線標示於圖 2c，利於後續的處理。
3. 以 boundary tracking追蹤手指間的線段 $(F_{ix_j}, F_{iy_j}), (i = 1, 2)$ ，並且定義兩點 $(F_{1x}, F_{1y}) | F_{1x} \geq F_{1x_j}, \text{ for all } j$ 、 $(F_{2x}, F_{2y}) | F_{2x} \geq F_{2x_j}, \text{ for all } j$ ，追蹤的方式是以中指與無名指之間的區塊為基準，這個區塊是屬於遮罩部分，位置不會改變，分別從這個區塊向上與向下搜尋，即可找到所需的線段，從線段上找到的兩點 (F_{1x}, F_{1y}) 、 (F_{2x}, F_{2y}) 就是我們所需要的指蹠參考點（圖 2d）。
4. 將 (F_{1x}, F_{1y}) 、 (F_{2x}, F_{2y}) 連線後，在其中垂線上固定距離切割一個 160×160 的掌紋區塊（圖 2e）。將影像轉正後（圖 2f），就是我們所需要的掌紋影像。

有些人的手掌因為指蹠位置高低不均，或是手掌比較小，造成取得的掌紋影像包含黑色遮罩的部分（如圖 3），這時候必須標示遮罩的位置，在進行後續處理的時候必須避開，否則手掌邊緣會造成後續處理的失誤。掌紋影像包含黑色遮罩並不會降低辨識率，因為掌紋影像還是包含了手掌中絕大部分的掌紋，只要同一人的掌紋影像永遠取自手掌的相同位置，就可以符合辨識的要求。

2.2 掌紋擷取

此步驟會將掌紋影像中的掌紋位置偵測出來，取得的掌紋座標資訊稱為掌紋資訊，本研究使用USAN [3]這個方法來擷取掌紋，可以取得掌紋的位置與寬度資訊，擷取的過程中無法分辨掌紋主線與細紋，我們會在取得掌紋資訊以後，將此資

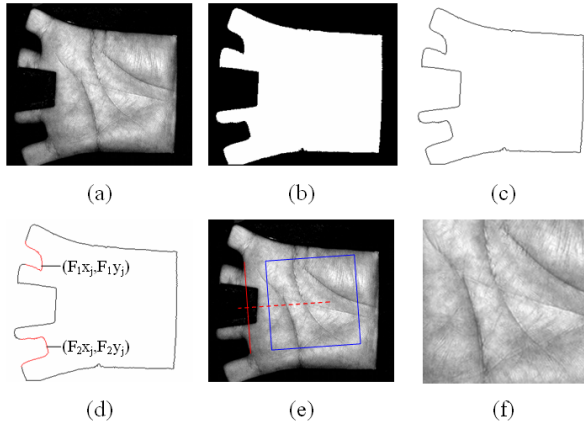


圖 2: 切割手掌影像：(a) 原始手掌影像, (b) Binary image, (c) 手掌邊緣, (d) boundary tracking, (e) 掌紋區塊, (f) 掌紋影像。

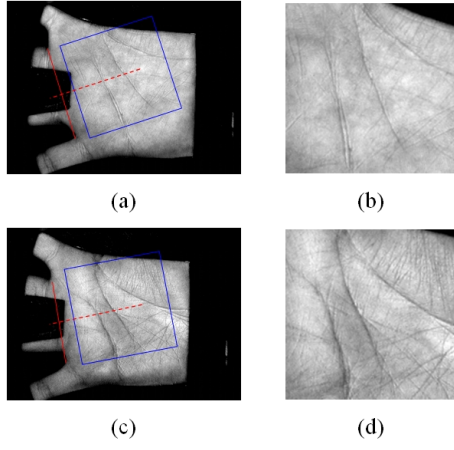


圖 3: 取得掌紋影像例外狀況：(a) 原始手掌影像, (b) 切割後的掌紋影像, (c) 原始手掌影像, (d) 切割後的掌紋影像。

訊以 Binary image 表示，再對 Binary image 進行去雜訊與 close 的處理，如此可以大致去除掌紋資訊中的細紋資訊，並且強化掌紋資訊中的主線資訊。

首先介紹 USAN 的基本概念，圖 4a 為掌紋影像範例，深灰色代表手掌上的掌紋，淺灰色則代表沒有掌紋的部分。此方法使用一個圓形的 mask，其圓心命名為『nucleus』，nucleus 將通過影像上每一個 pixel 座標（圖 4a 中的 mask A、mask B、mask C 代表其中三個位置），並且計算在 mask 內，有多少個 pixel 的灰階值（亮度）相似於 nucleus 的灰階值，而相似的 pixel 數量就叫做 USAN 值：當 mask 位於影像沒有變化的區域，USAN 值最大（如圖 4a 的 mask A）；當 nucleus 剛好位於影像中的線上，USAN 值最小（如圖 4a 的 mask B）；當 mask 位於線上，但 nucleus 沒有位於線上，USAN 值則介於最大及最小之間（如圖 4a 的 mask C）。

因此，『USAN 值較小的位置就是線的所在位置』，接下來只要以簡單的 threshold 就可以將線的位置擷取出來。為了方便觀察，我們建立一個三維座標空間（稱為 USAN area），Z 軸代表 USAN 值（以顏色表示），見如圖 4b。

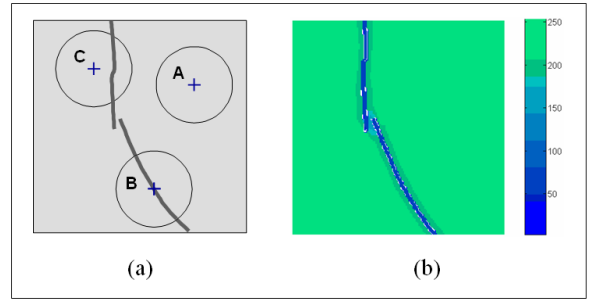


圖 4: USAN 主要概念：(a) 掌紋影像範例, (b) USAN area.

實際的掌紋影像（圖 5a）為灰階影像（灰階值 0 ~ 255）；解析度為 160 × 160，測得掌紋的寬度約為 1 ~ 3 pixel。經過實驗後發現，訂定 mask 半徑為 9，灰階相似度為 25（若 nucleus 與 mask 中的 pixel 差值小於 25，則定義為『相似』），可以擷取最多的掌紋資訊，完成的 USAN area 見圖 5b，其中顏色越深代表 USAN 值越小。接下來將 USAN area 以一個 threshold 做篩選，若 USAN area 中的最大值為 U_{max} ，則 threshold 訂為 $0.3 \times U_{max}$ ，低於這個值就代表是掌紋，高於這個值就不是掌紋，最後以一個 Binary image 表示，1（黑色）代表掌紋、0（白色）代表不是掌紋，見圖 5c。

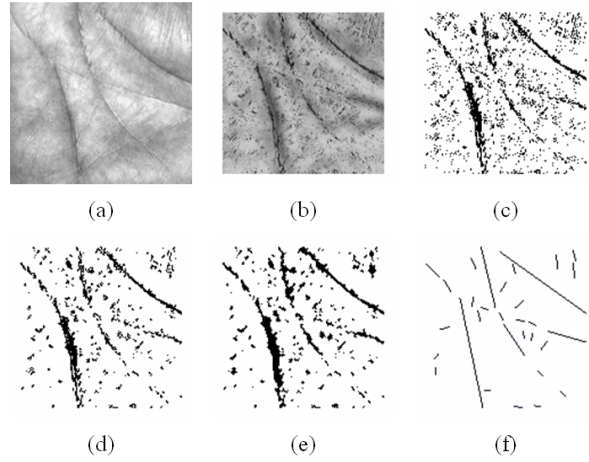


圖 5: 以 USAN 擷取掌紋：(a) 原始掌紋影像, (b) USAN area, (c) Binary image, (d) 經過去雜訊處理, (e) 經過去雜訊與 close 處理, (f) Lines map.

取得的 Binary image 中有一些很小的區塊，可能是 USAN 判斷錯誤的 pixel（非掌紋判斷為掌紋），或是手掌中很小細紋，在此使用一個 3 × 3 的 filter 執行去雜訊的動作，掃描 Binary image 中所有 pixel，若 filter 的範圍中，1（黑色）的個數少於 3 個，則將 filter 中心的 pixel 設為 0，可以抹除由 1 ~ 2 個 pixel 所構成的雜訊，處理後的成果見圖 5d）。

雜訊去除後，可以發現主線的部分有一些很小的間隙沒有被擷取出來，在此希望填補這些間隙，以連結成較大的區塊，利於後續的處理。我們使用影像處理常用的 close image，filter 設為半徑 1 的圓，不使用更大的 filter，是不希望做了 close 運

算後，主線跟鄰近的細紋連接起來，造成接下來分辨主線與細紋的困難，處理後的成果見圖 5e。

2.3 數值化

我們從掌紋資訊中擷取主線的資訊，並且利用簡單線性迴歸分析，將主線資訊以數值表示。處理的步驟是先將 Binary image 線段化，用直線線段來表示掌紋的位置與長度，接下來以線段的長度與相對的距離，判斷哪些線段屬於主線，找出三條主線分別由哪些線段表示以後，將線段對應回原來的 Binary image，取得完整掌紋資訊，再依序將掌紋的座標做簡單線性迴歸分析，線性方程式參數化以後的參數，就是最後的數值。

將 Binary image 線段化需要兩個步驟，首先以 Connected Component Labeling[5] 搜尋 Binary image 中相連的 pixel，所有緊鄰的 pixel 會被視為一個 Block (圖 6a)，再來要將每個 Block 以線段表示，方法是在每一個 Block 中取兩點，使得兩點連線的距離為最長，然後將取得的兩點以一條直線線段連接 (圖 6b)，所有線段的集合稱為 Lines map (圖 5f)。每一條線段會紀錄以下資訊 (圖 6cd)： L_k = 線段代號； $|L_k|$ = 線段長度； θ_k = 線段與正 X 軸夾角 ($0^\circ \sim 180^\circ$)； A_k = 線段端點 (Y 座標較小的)； B_k = 線段端點 (Y 座標較大的)， A'_k = 線段上與端點 A_k 距離為 2 的點； B'_k = 線段上與端點 B_k 距離為 2 的點； VL_k = 線段的斜率； VL'_k = 與線段垂直的斜率。

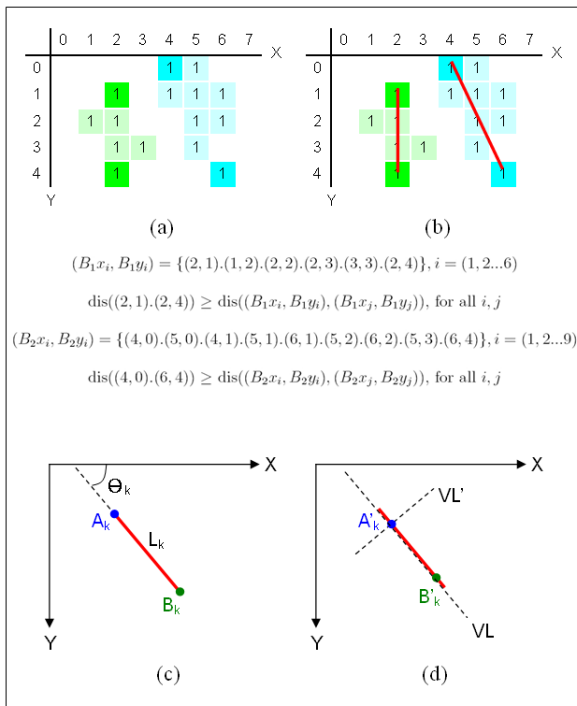


圖 6: Binary image 線段化範例：(a) Binary image 簡圖 (共有 2 個 Block)，(b) 以線段代表 Block，(c) 線段資訊，(d) 線段資訊。

Lines map 中包含主線線段與細紋線段，而較長的線段都是主線線段，在此必須判斷『哪些線段

是屬於同一條主線』。我們的做法是搜尋 Lines map 中最長的線段，將此線段定為基準線，並計算其他線段端點與基準線端點的距離與相對位置，然後取一條『最適合』的線段做為下一條基準線，以同樣的方法持續取得基準線，最後所有基準線就是一條主線的所有線段。所謂最適合的線段必須符合以下條件 (示意圖 7)：

1. 線段角度 θ_k 與基準線 θ 差值小於 30° 。
2. 使用 VL'_k 與基準線的 A' 點建立一條直線方程式，將基準線的 A 點與線段的 B 點分別代入方程式，若 2 個點位於直線的同側，則代表符合條件。
3. 使用 VL_k 與基準線上的任一點建立一條直線方程式，計算線段端點與基準線的距離 $disL$ ，若距離小於 5，則符合條件。
4. 若有多條線段符合以上三個條件，則計算基準線的 A' 點與線段 B 點的距離 $disP$ ，取 $disP + disL \times 5$ 最小的作為下一條基準線。

以上是朝向基準線的上方 (Y 座標較小) 搜尋下一條基準線的條件，若要朝下方搜尋，則將條件中的 A、B 點對調即可。我們先朝基準線的上方搜尋，直到沒有線段符合條件，再朝下方搜尋，當下方也沒有線段符合條件，就完成一整條主線的搜尋動作，接下來再搜尋 Lines map 中剩下的線段，將最長的線段訂為新的基準線，重複執行相同運算直到搜尋出三條主線為止。

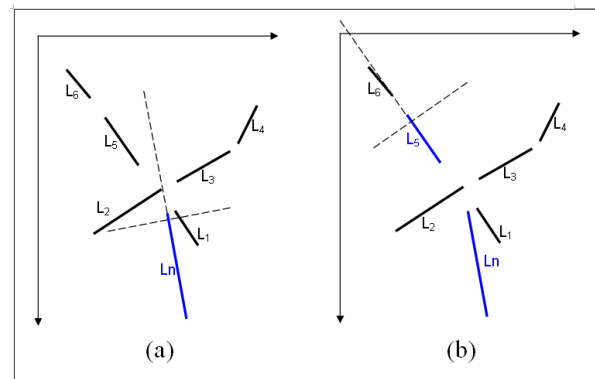


圖 7: 基準線與其他線段示意圖：(a) 最長線段 L_n 為基準線。(b) L_5 成為新的基準線。

從 Lines map 搜尋出來的主線線段 (顯示於圖 8a，三個顏色分別屬於三條主線)，對照 Binary image 取得相對應的 Block (圖 8b)，將 Block 內所有 pixel 座標點套用線性迴歸公式，即可得到線性方程式 $y = ax + b$ ，手掌中有三條主線，因此可得到三個線性方程式 (圖 8c 的黃色線)。為了方便計算線性方程式之間的差異，我們紀錄的數值不是方程式的係數，而是紀錄另外兩個參數：線性方程式與正 X 軸的夾角『 Θ 』；線性方程式的 X 軸截距『 I 』。我們也計算出三個線性方程式與 $y = 80$ (圖 8c 的黑色虛線) 的交點，依據交點的 X 座標值，依序編號為 Line1、Line2、Line3，參數則依序訂為 Θ_1 、 I_1 、 Θ_2 、 I_2 、 Θ_3 、 I_3 ，在辨識的時候，參數就可以跟相對應的參數比較。

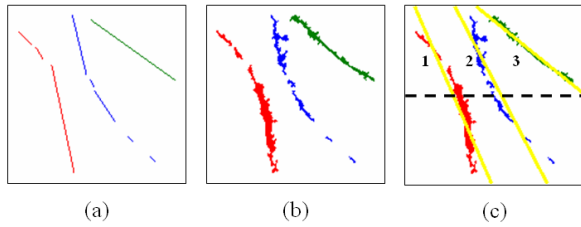


圖 8: 將各主線作線性迴歸：(a) 從Lines map分類出的三條主線, (b) 相對應的Binary image, (c) 三條線性方程式。

3 實驗結果

本研究以個人電腦進行實驗，硬體設備使用Intel Pentium4 CPU 2.0GHz與512MB RAM；作業系統為Windows XP；程式開發環境為.net 2003與MATLAB 7.0.4。close影像處理與線性迴歸分析使用MATLAB內建的函數，介面與其他處理是使用c++.net Form。從輸入手掌影像到輸出數值參數約需0.4毫秒，而整個辨識過程幾乎可於real-time完成。實驗分兩階段進行，分別建立20人與100人的資料庫，於100人的測試環境可達1%的FAR，與0.5%的FRR。

數值比對方式採用Euclidean Distance，此種相似程度的運算主要是直接計算兩參數之間的差異，計算後所得到的值越小代表兩參數差異越小，本實驗有兩種不同的參數（ Θ 與 I ），由於兩種參數的變化量不同，因此必須加上權值，計算公式如下所示：

$$d_E = \sqrt{\sum_{i=1}^3 (\theta_i - \theta_i^d)^2 + W \sum_{i=1}^3 (I_i - I_i^d)^2}$$

其中 d_E 代表Euclidean Distance， θ_i 、 I_i 代表輸入的待測參數， θ_i^d 、 I_i^d 代表資料庫儲存的參數， W 為權值，在此訂為0.1。 d_E 值越小，則代表待測參數與資料庫參數越相似。

本研究所有步驟的執行時間顯示於表 1，並與論文[8]比較，論文[8]的系統架構是將掌紋影像直接進行特徵擷取，然後以擷取出來的特徵進行比對，它也是目前唯一實際運作的掌紋辨識系統。本研究與論文[8]使用一樣的手掌影像資料庫，而系統架構是先將掌紋影像中的掌紋擷取出來以後，再執行數值化的動作，最後以數值參數進行比對。特徵比對與數值比對代表相同的動作，都是比對資料庫中的參數，在此都是使用100人的資料庫，執行時間是指1人的參數比對資料庫中100人的參數所需的時間；總時間則代表辨識系統從得到手掌影像，到送出辨識結果所需的時間，不包含取得手掌影像所需的時間。

在資料庫建立與比對數量方面，本研究分別建立20人與100人的資料庫，模擬家庭門禁與公司門禁系統，實驗所使用的手掌影像皆為右手手掌影像。第一階段以20個不同人的手掌影像建立資料庫，每人提供2張不同的手掌影像，其中一張影像經過掌紋特徵數值化的處理，取得參數後存入資料庫，最後資料庫共20組不同的參數，另一張影像也經過同樣的步驟取得參數，與資料庫參數

表 1: 執行時間比較.

Online. [8]		本研究		
CPU	500 MHz	CPU	2.0 GHz	
取得掌紋影像	538ms	取得掌紋影像	62ms	
特徵擷取	84ms	掌紋擷取	USAN	281ms
			threshold	15ms
			去雜訊	16ms
			close	5ms
		數值化	線段化	31ms
			搜尋主線	0.5ms
線性迴歸	7ms			
特徵比對	1.1s	數值比對	0.1ms	
總時間	1.7s	總時間	0.4s	

進行比對，並且判斷是否為同一人。第二階段則建立100人的資料庫，每人提供3張不同的手掌影像，其中一張用於建立資料庫，剩下兩張與資料庫進行比對。比對的結果如表 2，第一階段的實驗可以達到100%的辨識率，20組測試參數都可以正確比對；在第二階段的實驗中，200組測試參數有3組無法正確比對，其中2組為False Accept，1組為False Rejection。

表 2將本研究的辨識率與論文[1][4][8]比較，其中論文[4]是以左手加右手一起辨識，可得到很高的辨識率，論文[8]則是目前測量最大的論文，每人需要從三張掌紋影像擷取特徵後儲存於資料庫，資料庫人數與測試人數中，一半提供右手影像，另一半提供左手影像。

4 結論

本研究實作了一套掌紋辨識系統，並且可以在real-time完成辨識。系統中使用簡單線性迴歸分析將掌紋特徵數值化，只需要用6個floating points即可代表掌紋特徵，大幅降低了資料量與數值比對的時間，由實驗結果可以發現同一人在不同時間所擷取的掌紋影像，數值化以後的參數都非常接近，而不同人的之間的數值差異量也夠大，因此使用簡單線性迴歸分析用於掌紋數值化是可行的方法，但是因為數值化以後的參數較少，比較適合50人以下的環境，以達到較好的辨識率。

在實際應用場合中，若需要更高的辨識率，可以從掌紋資訊中再擷取一些有用的特徵存入資料庫，例如主線間隙的位置或主線的寬度等特徵。在未來若可以結合多項生物辨識系統（例如掌紋、手掌靜脈與指紋），更可有效提高辨識率。

Table 2: 辨識率比較.

文獻		[4]	[1]	[8]		本研究	
特徵 (擷取方法)		eigenpalms feature	crease points	Gabor filter		linear regression	
比對方法		Euclidean distance		Hamming distance		Euclidean distance	
資料庫	人數	191	20	50	100	20	100
	每人影像張數	4×2	1	3	3	1	1
	總影像張數	1528	20	150	300	20	100
測試	人數	191	20	386	386	20	100
	每人影像張數	4×2	1			1	2
	總影像張數	1528×2	20	7152	7152	20	200
辨識率	FAR %	0.03		0.1	0.3	0	1
	FRR %	1				0	0.5
	Accurate Rate %		95	99.5	98		

參考文獻

- [1] J. Chen, C. Zhang and G. Rong, "Palmpoint recognition using crease", International Conference on Image Processing, vol. 3, pp. 234-237, 2001.
- [2] F. Li and M.K.H. Leung, "Hierarchical Identification of Palmpoint using Line-based Hough Transform", International Conference on Pattern Recognition, vol. 18, pp. 149-152, 2006.
- [3] L. Liu and D. Zhang, "Palm-Line Detection", International Conference on Image Processing, vol. 3, pp. 11-14, 2005.
- [4] G. Lu, D. Zhang and K. Wang, "Palmpoint recognition using eigenpalms features", Pattern Recognition Letters, vol. 24, pp. 1463-1467, 2003.
- [5] J.M. Park, C.G. Looney and H.C. Chen, "Fast Connected Component Labeling algorithm Using A Divide and Conquer Technique". Conference on Computers and Their Applications, pp 373-376, 2000.
- [6] "PolyU Palmpoint Database", <http://www.comp.polyu.edu.hk/biometrics>.
- [7] D. Zhang and W. Shu, "Two novel characteristics in palmpoint verification: datum point invariance and line feature matching", Pattern Recognition Letters, vol. 32, pp. 691-702, 1999.
- [8] D. Zhang, W.K. Kong, J. You and M. Wong, "Online Palmpoint Identification", IEEE Transactions on Instrumentation and Measurement, vol. 25, pp. 1041-1050, 2003.
- [9] X. Wu, D. Zhang, K. Wang and B. Huang, "Palmpoint classification using principal lines", Pattern Recognition Letters, vol. 37, pp. 1987-1998, 2004.