行政院國家科學委員會專題研究計畫 成果報告

適用於監視系統之人類行為識別方法開發
研究成果報告(精簡版)

計 畫 類 別 ：個別型
計 畫 編 號 ：NSC 97-2221-E-343-007-
執 行 期 間 ：97 年 08 月 01 日至 98 年 09 月 30 日
執 行 單 位 ：南華大學資訊工程學系

計 畫 主 持 人 ：廖怡欽
共 同 主 持 人 ：賴榮滄
計畫參與人員：碩士班研究生-兼任助理人員：黃奕棻
碩士班研究生-兼任助理人員：顏斌傑
碩士班研究生-兼任助理人員：蘇怡璇
碩士班研究生-兼任助理人員：陳威志
大專生-兼任助理人員：林子翔
大專生-兼任助理人員：王士彥
大專生-兼任助理人員：吳誌軒
大專生-兼任助理人員：陳韋臣

報 告 附 件 ：出席國際會議研究心得報告及發表論文

處 理 方 式 ：本計畫涉及專利或其他智慧財產權，2 年後可公開查詢

中 華 民 國 98 年 12 月 02 日

行政院國家科學委員會補助專題研究計畫 ☑ 成 果 報 告
□期中進度報告

適用於監視系統之人類行為識別方法開發

計畫類別：☑ 個別型計畫　　□ 整合型計畫
計畫編號：NSC 97－2221－E－343－007－
執行期間：97 年 8 月 1 日至 98 年 9 月 30 日

計畫主持人：廖怡欽
共同主持人：賴榮滄
計畫參與人員：黃奕棻、顏斌傑、蘇怡璇、陳威志、王士彥、林子翔、吳誌軒、陳韋臣

成果報告類型(依經費核定清單規定繳交)：☑精簡報告　□完整報告

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
☑出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢
☑涉及專利或其他智慧財產權，□一年☑二年後可公開查詢

執行單位：南華大學 資訊工程學系

中　華　民　國 98　年　11　月　29　日

摘要

由於視訊監視系統的普及，分析監視內容的應用也越來越普遍。本計劃主要目的在針對固定式監視系統開發一套以動作歷史資訊(MHI)技術為基礎且不受動作速度影響的人類行為識別系統，本系統主要內容包括：行為特徵資料庫設計以及與不受動作速度影響之人類行為識別方法開發等部份。計劃執行完畢已完成一基本人類行為特徵資料庫的開發工作以及一與動作速度無關之人類行為識別方法。

關鍵字：監視器，行為識別，動作歷史資訊

Abstract

Due to the popularity of the surveillance systems, the requirement of analyzing video streams obtained from surveillance systems becomes more and more popular. The major purpose of this project is to develop a motion-history-information (MHI) based and motion speed independent human behaviors identification system for fixed surveillance systems. The major task of this project is to design the database for storing behavior features and to develop an effective method for recognizing human behaviors without influenced by motion speed. In the end of this project, we have completed the development of the database for storing behavior features and a motion speed independent human behaviors identification method.

Keywords: surveillance systems, behavior identification, motion history information

1.前言

由於監視系統的應用愈來愈廣泛，分析監視內容的工作也變得愈來愈重要，諸如交通監測、居家照護、遠端監測、犯罪偵查、工業生產、機器人視覺等都有判斷影像或視訊內容的需要。目前視訊內容的判斷工作大多仍需大量人力介入，無法有效快速的完成分析工作。自動視訊分析系統可快速分析視訊內容，找出視訊內容中感興趣的物件，完成視訊檢索或行為識別工作。本計劃主要目的在開發一套可自動分析固定式監視器內容判斷出人類特定異常行為的人類異常行為分析系統，此系統適用居家照護，保全監測，或犯罪偵測等應用，可自動偵測人類異常行為，適時提出警訊。本計劃從 95 年開始進行，95 年度與 96 年度分別完成適用於固定式監視系統之物件分割方法開發工作[1-3]與適用於固定式監視系統之物件追蹤方法開發工作[4-5]。

2.研究目的

本計劃本年度主要目的在萃取出適用的行為特徵，設計及建構適用的人類行為特徵資料庫，以及開發不受動作速度影響的人類行為識別方法。

3.文獻探討

人類行為識別技術[6]主要包括行為特徵萃取以及行為特徵比對兩部份，行為特徵萃取部份在於分析連續畫面中物件空間特徵隨時間變化情形，產生可供用來辨識人類行為的行為特徵；行為特徵比對部份則是由行為特徵資料庫中找到符合物件行為特徵的行為。常見的行為特徵有物件位置隨時間改變資訊(移動軌跡)[7]，物件形狀隨時間改變資訊[8,9]，人體模型中肢體隨時間改變資訊[10,11,12]，以及由動作歷史影像(Motion History Image)取得的行為特徵(如時間樣版、整體或區域移動方向、動作方向直方圖、移動速度、主要元素分析、動作幾何分佈等) [13-19]等。常用的人類行為比對方法主要有兩種[6]，一種是依狀態發生次序決定行為種類的狀態序列比對法[7,20]，另一種則是動作歷史影像特徵比對法[13-19]。

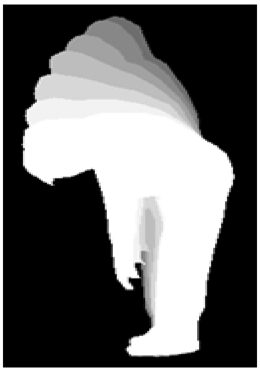
在狀態序列比對法中，人類物件的一個靜止畫面為一種狀態，一組特定的狀態序列用來表示一種特定行為。使用狀態序列比對方式的好處在於狀態序列中只需包含關鍵狀態(關鍵畫面)，關鍵畫面之間多餘的畫面會被忽略，不影響行為識別的結果，因此使用狀態序列比對方式，較不受物件運動速度影響。使用狀態序列比對方式缺點在於關鍵畫面比對過程費時以及行為的狀態序列資訊取得不易。

使用動作歷史影像特徵比對法[13-19]，首先必須從物件的動作歷史影像中取出行為特徵，然後再使用行為特徵匹配方法找出資料庫中相似的行為。可用的行為特徵匹配方法有時間樣版相似度比對法[13]、距離最接近行為比對法(K Nearest Neighbor) [18]、高斯混合模型[8,16]、隱藏式馬可夫模型(Hidden Markov Models)[14,21]、與類神經網路(Neural Network)[12,15,18]等方法。使用這種行為特徵比對法好處在於行為特徵比對速度快，容易定義新行為。缺點則是識別效果受到物件運動速度與動作複雜度影響。

為達到即時應用與簡化新行為的定義過程，本計劃將以動作歷史影像特徵比對法[13-19]為基礎，並且完成以下幾項工作：1.人類基本行為特徵選取；2.動作歷史影像組成畫面自動調整方法設計與實作；3.人類基本行為特徵資料庫建構；4.人類基本行為識別方法開發。

4.研究方法

圖一所示為一個動作歷史影像的例子，一個動作歷史影像是一張灰階影像，內含物件在一段時間內的行為歷程。影像中最上層的部份是最近取得的物件遮罩，往下則依序為離目前時間點較遠的物件遮罩。其中，最上層的物件遮罩以白色表示，離目前時間點愈遠的物件遮罩以愈暗的顏色表示。



圖一：一個動作歷史影像的例子

動作歷史影像內只會保留最近一段時間範圍內的物件遮罩。假設目前時間點為 t_c，時間點 t_i 所取得的物件遮罩為 M_{ti}。若一動作歷史影像內保留之物件遮罩的數目為 d，則該動作歷史影像內將包含 $M_{tc-d+1}, M_{tc-d+2}, ..., M_{tc}$ 等 d 個物件遮罩。其中，M_{tc} 位於最上層，物件遮罩 M_{tc-d+1} 位於最下層。物件遮罩 M_{ti} 在動作歷史影像內的顏色值 C_{ti} 如下：

$$C_{ti} = (t_i - t_c + d) \times \frac{255}{(d+1)} \tag{1}$$

4.1 人類基本行為特徵選取

一個動作歷史影像內含一個行為由開始到結束之間所有物件遮罩的集合，因此動作歷史影像可供作為行為的樣版(template)識別行為的種類。使用動作歷史影像進行行為識別，可事先取得所要識別之行為的動作歷史影像，然後由動作歷史影像中取出可用的行為特徵，再使用這些特徵作為行為識別的依據。

要判斷視訊中是否出現所要識別的行為，可產生目前畫面中物件的動作歷史影像，取出其特徵，再使用該特徵進行行為識別。常用的動作歷史影像行為特徵有以下幾種：

1.直接使用動作歷史影像作為特徵。
2.使用動作歷史影像中像素的坡度角度直方圖作為特徵。
3.使用整張影像的坡度角度作為特徵。
4.將影像分成多個區域，並使用各區域的坡度角度作為特徵。

以上幾種方法中，第一種方法易受物件運動速度與物件形狀影響，第二種方法內含太多無用特徵，且需要較多計算量。基於行為識別準確度與運算速度考量，本計劃採用 9 個區域坡度角度作為行為特徵。圖二所示為將動作歷史影像切割成 9 個區域的分割方式。

1	2	3
4	5	6
7	8	9

圖二：動作歷史影像區域分割方式

令 $f(x,y)$ 是動作歷史影像中座標為 (x,y) 的像素值，$G_H(x,y)$ 與 $G_V(x,y)$ 分別為像素 $f(x, y)$ 的水平與垂直坡度強度值，其中，$G_H(x,y)$ 與 $G_V(x,y)$ 分別可用以下 Sobel 遮罩計算取得：

$$S_H = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \tag{2}$$

$$S_V = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{3}$$

像素 $f(x,y)$ 的坡度角度 $\theta(x, y)$ 與坡度強度 $I(x, y)$ 可使用 $G_H(x,y)$ 與 $G_V(x,y)$ 計算得到，計算公式如下：

$$\theta(x, y) = \tan^{-1}\left(\frac{G_H(x, y)}{G_V(x, y)}\right) \tag{4}$$

$$I(x, y) = \sqrt{G_H(x, y)^2 + G_V(x, y)^2} \tag{5}$$

假設區域 R 的 x 座標範圍為 $[a, b]$，y 座標範圍為 $[c, d]$，則該區域坡度角度 θ_R 計算公式如下：

$$\theta_R = \frac{\sum_{x=a}^{b}\sum_{y=c}^{d}(I(x, y) \times \theta(x, y))}{\sum_{x=a}^{b}\sum_{y=c}^{d} I(x, y)} \tag{6}$$

針對每個動作歷史影像，我們將動作歷史影像依圖二所示將影像分成 9 個區域，然後使用公式(6)取出 9 個區域坡度角度 $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \theta_8,\ \theta_9]$。為了判斷每個區域坡度角度的可用性，可搜集許多行為的動作歷史影像及其相對應的 9 個區域坡度角度，然後使用分群方法[23-25]及群組分析方法[22]，產生最適數量的群組，再分析每個區域坡度角度於群組之間的差異性，去除群組間差異性小的特徵。針對本計劃所取得測試樣本，我們發現每個區域均有很高的可用性。因此我們保留所有區域的坡度角度，做為我們的行為特徵。

4.2 動作歷史影像組成畫面自動調整方法

動作歷史影像會受到物件動作速度影響，動作速度快，動作歷史影像內含物件遮罩張數少；反之，動作歷史影像內含物件遮罩張數多。本計劃使用整張影像的坡度角度與區域坡度角度作為行為特徵，可減輕動作歷史影像內含物件遮罩張數對坡度角度的影響，但是不知道物件動作速度，很容易產生內含物件遮罩張數不足或張數過多的動作歷史影像，以致影響行為識別效果。為解決以上問題，本計劃使用以下兩種作法：

作法一： 產生動作歷史影像時，先計算目前物件遮罩與參考物件遮罩的差異度。當差異度大於給定臨界值 TH_{diff} 時，才將目前物件遮罩加入動作歷史影像中，並將目前物件遮罩設為參考物件遮罩；否則，跳過目前物件遮罩。

作法二： 產生多個內含不同物件遮罩張數的動作歷史影像，若有任一動作歷史影像符合某一定義行為，則重設所有動作歷史影像。

由於針對不同的動作會有不同的最佳臨界值設定，既然物件的行為不可知，因此無法

有效設定 TH_{diff} 的值。為此，本計劃僅設定一個較小的 TH_{diff} 值(1%)，用以去除差異性小的畫面，目的在確保物件確實有動作。由於以上作法無法有效解決運動速度的影響，因此針對一未知行為的物件，除了使用以上方法去除差異小的畫面外，我們也產生多個內含不同物件遮罩張數的動作歷史影像。如此，不管動作速度為何，皆可找到一組合適的動作歷史影像來描述目前的動作。為避免部份行為因前面動作雷同而產生誤判動作，行為識別順序應由內含較多張物件遮罩的動作歷史影像開始識別。

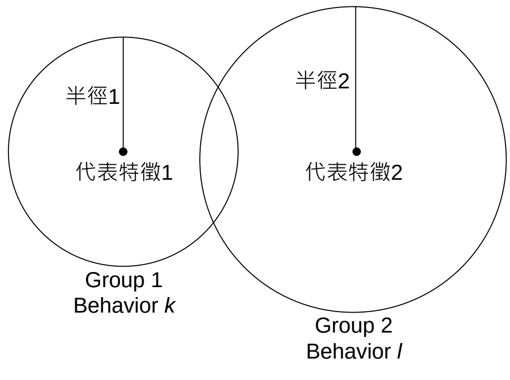
4.3 人類基本行為特徵資料庫

人類基本行為特徵資料庫用來儲存基本行為的行為特徵，由於人員或拍攝角度的不同進行相同的行為會產生不同的行為特徵，因此，針對每種基本行為，我們在資料庫中儲存多組行為特徵。為了加快行為特徵的比對速度以及取出多組相同行為的代表特徵，針對每種行為的多個行為特徵，我們使用分群方法[22-25]，將距離相近之相同行為的行為特徵當成同一個群組，然後再為每個群組產生一個代表特徵及群組半徑並儲存在資料庫中。

一個群組的代表特徵為該群組內所有行為特徵的平均值，群組半徑則為該群組內所有行為特徵中距離代表特徵最遠的距離。此處所使用的距離是 Euclidean 距離，令 B_1= $[\theta_1^1,\theta_2^1,\theta_3^1,\theta_4^1,\theta_5^1,\theta_6^1,\theta_7^1,\theta_9^1]$與 B_2=$[\theta_1^2,\theta_2^2,\theta_3^2,\theta_4^2,\theta_5^2,\theta_6^2,\theta_7^2,\theta_9^2]$分別是兩個行為特徵，則行為特徵 B_1 與 B_2 的距離計算公式如下：

$$D(B_1,B_2)=\sum_{i=1}^{9}\sqrt{(\theta_i^1-\theta_i^2)^2} \tag{7}$$

同一種行為的所有行為特徵一開始會被放在同一個群組中，群組半徑若超過一給定的臨界值 TH_{dist}（該臨界值同時也被使用在下一節的行為識別方法中）則該群組會被拆成兩半，本計劃的作法是使用 Direct Search Binary Splitting (DSBS)[26]方法，將群組分成兩半，如果分群後的群組半徑仍然大於給定的臨界值 TH_{dist}，則重覆分群直到群組半徑小於臨界值為止。此外，不同行為的群組範圍不能重疊，下圖所示即為兩個群組範圍重疊的例子。如果有發生群組範圍重疊的情況，必須將較大的群組分群，直到所有不同行為的群組都沒有發生重疊的情況為止。

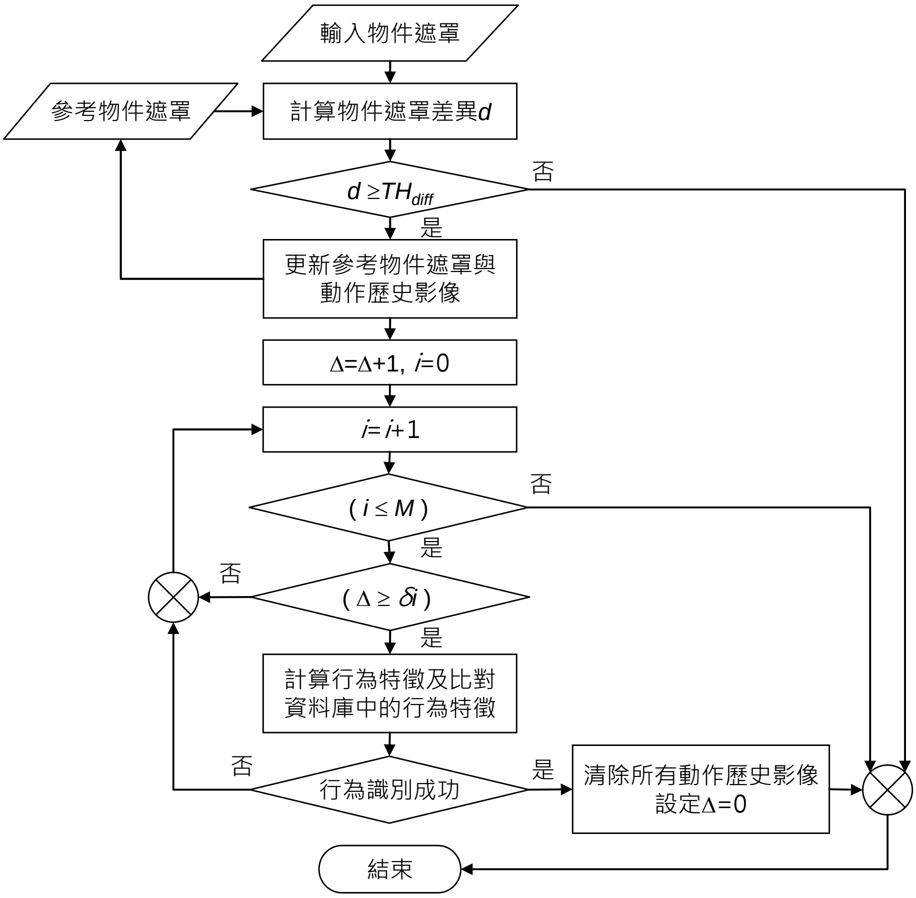


圖三：群組範圍重疊示意圖

4.4 人類基本行為識別方法

要判斷視訊中是否出現資料庫中定義的人類行為，首先必須產生視訊物件的動作歷史影像。為了減輕動作速度的影響，我們除了去除差異小的畫面外，也產生M份內含不同物件遮罩張數的動作歷史影像，令可用的物件遮罩張數集合為 $\{\delta_1,\delta_2,...,\delta_M\}$，其中 $\delta_1>\delta_2>...>\delta_M$。假設$\Delta$為輸入視訊中可用的畫面張數，其初始值為 0，若$\Delta\geq\delta_i$，表示內含$\delta_i$張物件遮罩的動作歷史影像內容搜集完成，可拿來跟資料庫中的行為代表特徵進行比對工作。行為特徵比對方式為計算行為特徵之間的距離（相異程度），距離計算公式如公式(7)所示。當距離小於臨界值 TH_{dist} 時，則認為視訊中出現該行為。由於我們使用多個δ，因此，

同一個時間點可能會有多個動作歷史影像內容搜集完成（$\Delta \geq \delta_i$），而可能必須針對多個動作歷史影像進行比對動作。圖四所示為本計劃所使用的人類基本行為識別流程。

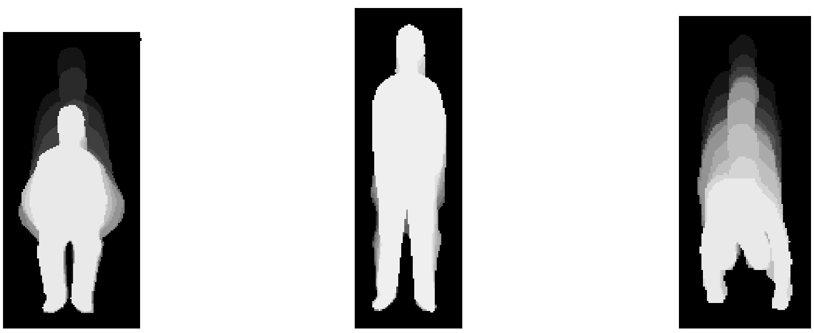


圖四：人類基本行為識別流程

行為識別的順序由內含較多張物件遮罩的動作歷史影像開始判斷，可避免當部份行為因前面動作雷同而產生誤判動作。由於連續輸入畫面之間的動作變化不大，因此使用連續輸入畫面所產生出來的動作歷史影像也會非常相似。也就是，當我們使用某個δ識別一個行為後，在接下來的幾個畫面中，也會識別出相同的行為。為了解決這個問題，當我們識別出一個行為後，如果接下來識別結果也是相同行為時，則略過不計。

5.結果與討論

為了測試所開發方法的效能，本計劃拍攝了兩個人坐下、起立、及跌倒等動作多組畫面（圖五所示為坐下、起立、及跌倒等動作的動作歷史影像的例子）。



(a) (b) (c)

圖五：人類基本行為動作歷史影像：(a)坐下, (b)起立, (c)跌倒

本實驗設定 TH_{dist}=30，使用所拍攝的畫面產生 11 組代表特徵，所產生之基本行為的行為種類以及其代表特徵詳列如下：

表一：資料庫中的 11 組代表特徵

行為	區域 0	區域 1	區域 2	區域 3	區域 4	區域 5	區域 6	區域 7	區域 8
起立	157	150	56	44	120	138	133	68	75
起立	250	197	60	119	140	89	102	142	126
起立	201	158	47	108	101	155	70	72	144
起立	230	149	43	113	165	114	81	101	173
起立	223	167	44	106	62	48	171	107	117
坐下	231	117	27	198	84	62	77	104	193
坐下	292	167	35	191	102	105	58	109	149
坐下	229	152	48	142	132	81	86	169	174
跌倒	221	132	26	149	81	27	79	137	85
跌倒	206	78	23	165	76	44	145	158	117
跌倒	212	82	38	156	52	41	133	80	141

為了測試行為辨識效果以及動作速度對行為識別的影響，我們拍攝一段內含多種行為以及不同動作速度的測試視訊，畫面尺寸為 704×480，測試視訊內含 150 張畫面。本實驗使用的物件遮罩張數集合為｛δ_1=40, δ_2=36, δ_3=32, δ_4=28, δ_5=24, δ_6=20, δ_7=16, δ_8=12｝。

實驗結果顯示，測試影片中的動作都可以被有效的識別出來，下表列出所識別出來之行為的發生時間點(以畫面張數表示)、行為種類、以及識別出行為時所使用之動作歷史影像的 δ(物件遮罩張數)。

表二：行為識別結果

畫面編號	識別到的行為	識別出行為時所使用的 δ
16	坐下	16
31	起立	16
43	坐下	20
54	起立	16
72	跌倒	20
101	起立	40
112	跌倒	24
132	起立	20

如表二所示，我們可以看到有些行為使用較大的 δ 即可成功識別，有些行為則需使用較小的 δ，且由於識別方式是由較大的 δ 先比對，若是動作速度不適合較大的 δ，則表示動作速度大於所使用的 δ，如此動作歷史影像會包含過多動作，而無法有效識別視訊中的動作，圖六所示為第 54 張畫面時，使用 δ=40 時所產生的動作歷史影像，由圖六可見，內含太多動作時，無法有效識別出動作。

圖六：第 54 張畫面使用 δ=40 所產生的動作歷史影像

相反的，如果動作速度太慢，δ 太小，畫面張數太少，會導致畫面中取得的物件遮罩不夠多，使得部份區域的角度資訊不正確。例如：坐下的動作如果太慢物件位移量小（如圖七)，則畫面大小會接近最後一張物件遮罩，以致動作歷史影像高度降低，如此會使得各區域所包含的物件內容改變，既然內容不同，所取得的角度資訊就會不同，因此易導致誤判情形發生。

(a)

(b)

圖七：第 72 張畫面所產生的動作歷史影像：(a) 使用 δ=12, (b) 使用 δ=20.

以上實驗在筆記型電腦上執行，該筆記型電腦配備 Intel Core 2 Due P8600 2.4 G CPU、4GB 記憶體，作業系統為 Windows Vista，程式開發環境為 Microsoft Visual C++ 2008。在該環境下，平均每一張畫面的處理時間約為 15ms。要加快處理速度，當動作歷史影像大小太大時，可先行將其影像縮小後再進行坡度角度的計算，由於本計劃使用區域資訊做為行為特徵，縮小影像大小對行為識別不會造成明顯的影響。

6.結論

本計劃已完成一不受動作速度影響的人類行為識別方法，所開發出來的人類行為識別方法使用歷史動作影像作為人類行為樣版，為了降低運動速度對行為識別的影響，本計劃分析前後張畫面物件的差異程度，去除差異程度小的畫面，僅使用差異程度大的畫面產生歷史動作影像，為了進一步降低運動速度的影響，本計劃針對視訊內容產生多個動作歷史影像。為了產生具代表性的行為特徵，本計劃也設計了一個適用的行為特徵資料庫，該資料庫針對每個基本行為，可儲存多組行為特徵及代表特徵。最後，本計劃也設計及實作一套人類基本行為識別方法，所開發出來的方法可有效識別人類行為且可有效降低動作速度的影響。

參考文獻

[1] 廖怡欽、陳易顯、黃崇仁、與賴榮滄, “針對視訊串流之多層式背景估測方法,” The Second conference on Digital Content Management and Applications, Tainan, Taiwan, June 2007.

[2] Yi-Ching Liaw, Bo-Shuan Chiu, Jim Z. C. Lai, and Patrick Huang, “A fast approach of object segmentation for video sequence,” The Ninth IASTED International Conference on Signal and Image Processing, Honolulu, USA, August 2007. (EI)

[3] 黃崇仁, 李政旻, 廖怡欽, 與賴榮滄, “視訊物件分割方法,” 中華民國專利, 2006/12 申請.

[4] Yi-Ching Liaw, "Improvement of the Fast Exact Pairwise-Nearest-Neighbor Algorithm," Pattern Recognition, Vol. 42, No. 5, May 2009, pp.867-870. (SCI; EI)

[5] Jim Z.C. Lai, Yi-Ching Liaw*, and Julie Liu, "Improvement of the k-means clustering filtering algorithm," Pattern Recognition, Vol. 41, No. 12, December 2008, pp. 3677-3681. (SCI; EI) (2007 Impact Factor:2.015, Ranking 23/210)

[6] Weiming Hu, Tieniu Tan, Liang Wang, and Steve Maybank, "A survey on visual surveillance of object motion and behaviors," IEEE Transactions on Systems, Man, and Cybernetics – Part C, Vol. 34, No. 3, pp. 334-352, August 2004.

[7] A. F. Bobick and A. D. Wilson, "A state-based technique for the summarization and recognition of gesture," International Conference on Computer Vision, Cambridge, pp.382-388, June 1995.

[8] Ashok Veeraraghavan, Amit K. Roy-Chowdhury, and Rama Chellappa, "Matching Shape Sequences in Video with Applications in Human Movement Analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 12, pp. 1896-1909, December 2005.

[9] 林金泉, 人類跌倒之行為分析與偵測, 國立中央大學碩士論文, 2004.

[10] K. Rohr, "Towards model-based recognition of human movements in image sequences," CVGIP: Image Understanding, Vol. 59 No.1, pp.94-115, 1994.

[11] Shanon X. Ju, Michael J. Black, and Yaser Yacoob, "Cardboard people: a parameterized model of articulated image motion," IEEE International Conference on Automatic Face and Gesture Recognition, Vermont, USA, pp. 38-44, October 1996.

[12] Y. Guo, G. Xu, and S. Tsuji, "Understanding human motion patterns," International Conference on Pattern Recognition, Jerusalem, Israel, pp.325-329, October 1994.

[13] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 3, pp.257-267, March 2001.

[14] G. R. Bradski and J. Davis, "Motion segmentation and pose recognition with motion history gradients," Machine Vision and Applications, Vol. 13, No. 3, pp 174-184, July 2002.

[15] 陳福泰, 以移動歷史影像為基礎之人類行為辨識, 中華大學碩士論文, 2005.

[16] R. Rosales and S. Sclaroff, "3D trajectory recovery for tracking multiple objects and trajectoryguided recognition of actions," IEEE Conference on Computer Vision and Pattern Recognition, Colorado, USA, June 1999, pp.117-123.

[17] J. W. Davis, "Hierarchical motion history images for recognizing human motion," IEEE Workshop on Detection and Recognition of Events in Video, BC, Canada, pp.39-46, 2001.

[18] R.V. Babu and K.R. Ramakrishnan, "Compressed domain human motion recognition using motion history information," International Conference on Image Processing, Hong Kong, pp. 41-44, April 2003.

[19] Hongying Meng, Nick Pears, and Chris Bailey, "A Human Action Recognition System for Embedded Computer Vision Application," IEEE Conference on Computer Vision and Pattern Recognition, MN, USA, pp. 1-6, June 2007.

[20] C. Bregler, "Learning and recognizing human dynamics in video sequences," IEEE Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, pp. 568-574, June 1997.

[21] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proceedings of the IEEE, Vol. 77, No. 2, pp. 257-286, February 1989.

[22] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis, "On Clustering Validation Techniques," Journal of Intelligent Information Systems, vol. 17, no. 2-3, pp. 107-145, 2001.

[23] Jim Z. C. Lai, Tsung-Jen Huang, and Yi-Ching Liaw, "A fast k-means clustering algorithm using cluster center displacement," Pattern Recognition, Vol. 42, No. 11, November 2009, pp.2551-2556. (SCI, EI)

[24] Jim Z. C. Lai and Yi-Ching Liaw*, "A Novel encoding algorithm for vector quantization using transformed codebook," Pattern Recognition, Vol. 42, No. 11, November 2009, pp.3065-3070. (SCI, EI)

[25] Yi-Ching Liaw*, Jun-Feng Lin, Shen-Chuan Tai, and Jim Z. C. Lai, "Fast exact pairwise-nearest-neighbor algorithm using groups and clusters rejection criteria," The Eleventh IASTED International Conference on Signal and Image Processing, Honolulu, USA, August 17-19, 2009.

[26] C. M. Huang and R. W. Harris, “A comparison of several vector quantization codebook generation approaches,” *IEEE Trans. on Image Processing*, vol. 2, no. 1, pp. 108-112, January1993.

計畫成果自評

■ 原計畫相符程度與達成預期目標情況

本計劃本年度預計完成工作項目如下：

1.人類基本行為特徵選取方法；

2.動作歷史影像組成畫面自動調整方法設計與實作；

3.人類基本行為特徵資料庫建構方法開發；

4.人類基本行為識別方法開發。

計劃執行完畢，已完成以上項目。

■ 研究成果之學術或應用價值

學術價值：本計劃已開發完成一些快速行為特徵分群方法，成果已發表於國際期刊[1,2]及國際學術研討會[3]；另外，我們也開發完成一與動作速度無關的人類基本行為識別方法，研究成果正準備投稿國際期刊或國際學術研討會。

應用價值：本計劃所開發出來的快速行為特徵分群方法與人類基本行為識別方法，可應用到監視器監控應用相關產業，分析影像內容，研究成果也可供申請專利，用以保護國內相關產業，迴避相關專利。

■ 影像處理人才培育

本計劃執行人員，主要包括四位碩士班學生以及多位大學部學生，碩士班學生主要負責實作本計劃所需的方法設計、程式撰寫、以及進行實驗，對其實作能力的提昇與實驗方法的熟悉有很大的幫助。大學部學生則幫忙拍攝影片，分割物件，以及協助撰寫程式等，參與人員對計劃與實驗的進行方式有更深入的了解，對程式實作能力也都能有效提昇。

■ 參考資料

[1] Jim Z. C. Lai, Tsung-Jen Huang, and Yi-Ching Liaw, "A fast k-means clustering algorithm using cluster center displacement," Pattern Recognition, Vol. 42, No. 11, November 2009, pp.2551-2556. (SCI, EI)

[2] Jim Z. C. Lai and Yi-Ching Liaw*, "A Novel encoding algorithm for vector quantization using transformed codebook," Pattern Recognition, Vol. 42, No. 11, November 2009, pp.3065-3070. (SCI, EI)

[3] Yi-Ching Liaw*, Jun-Feng Lin, Shen-Chuan Tai, and Jim Z. C. Lai, "Fast exact pairwise-nearest-neighbor algorithm using groups and clusters rejection criteria," The Eleventh IASTED International Conference on Signal and Image Processing, Honolulu, USA, August 17-19, 2009.

可供推廣之研發成果資料表

☑ 可申請專利　　☑ 可技術移轉　　　　　　　　　　日期：98 年 11 月 23 日

國科會補助計畫	計畫名稱：適用於監視系統之人類行為識別方法開發 計畫主持人：廖怡欽 計畫編號：NSC 97-2221-E-343-007-　　學門領域：資訊學門二
技術/創作名稱	快速視訊物件識別方法
發明人/創作人	廖怡欽
技術說明	中文： 由於視訊監視系統的普及，分析監視內容的應用也越來越普遍。本計劃主要目的在針對固定式監視系統開發一套以動作歷史資訊(MHI)技術為基礎且不受動作速度影響的人類行為識別系統，本系統主要內容包括：行為特徵資料庫設計以及與不受動作速度影響之人類行為識別方法開發等部份。計劃執行完畢已完成一基本人類行為特徵資料庫的開發工作以及一與動作速度無關之人類行為識別方法。
	英文： Due to the popularity of the surveillance systems, the requirement of analyzing video streams obtained from surveillance systems becomes more and more popular. This major purpose of this project is to develop a motion-history-information (MHI) based and motion speed independent human behaviors identification system for fixed surveillance systems. The major task of this project is to design the database for storing behavior features and to develop an effective method for recognizing human behaviors without influenced by motion speed. In the end of this project, we have completed the development of the database for storing behavior features and a motion speed independent human behaviors identification method.
可利用之產業 及 可開發之產品	可利用之產業：保全系統，交通監控，居家照護，網路視訊 可開發之產品：視訊物件識別方法
技術特點	適用固定式監視系統 快速識別視訊物件行為
推廣及運用的價值	可供用來識別視訊物件行為

※　1. 每項研發成果請填寫一式二份，一份隨成果報告送繳本會，一份送　貴單位研發成果推廣單位（如技術移轉中心）。

※　**2. 本項研發成果若尚未申請專利，請勿揭露可申請專利之主要內容。**

※　3. 本表若不敷使用，請自行影印使用。

出席國際學術會議心得報告

計畫編號	NSC 97-2221-E-343-006
計畫名稱	適用於監視系統之人類行為識別方法開發
出國人員姓名 服務機關及職稱	廖怡欽 南華大學副教授
會議時間地點	民國 98 年 8 月 17 日至民國 98 年 8 月 19 日
會議名稱	第 11 屆 IASTED 訊號與影像處理會議(Signal and Image Processing 2009)
發表論文題目	Fast Exact Pairwise-Nearest-Neighbor Algorithm Using Groups and Clusters Rejection Criteria

一、參加會議經過

此次出國主要目的是到美國檀香山參加 Signal and Image Processing 2009 會議(http://www.actapress.com)發表論文(附件一)，本人報告時間為民國 98 年 8 月 18 日下午 1:30。此會議由國際科學與科技開發協會(IASTED)舉辦，會議文章接受率大約 4 成，會議文章也經常被 EI 收錄，是一個很好的會議；會議地點在檀香山喜來登威基基飯店，開會地點氣候風景皆宜人，可有效調節開會氣氛及舒緩報告人的緊張心情。

會議 8 月 17 日(一)開始，本人搭乘華航班機 8/16（日）早上大約 6 點半到，對開會地點週遭環境及會議室的分佈先做個了解，第二次參加這個會議，開會地點週遭環境跟兩年前差不多，這兩年台灣夏天溫度提高不少，夏威宜的氣候則變化不大，均溫仍維持在 25 度左右。

會議第一天(8 月 17 日)早上 8:20 報到，人明顯較上次少，可能是受到經濟不景氣的影響，今年投稿期限延了很多次，由 3 月底延到 5 月，論文數仍明顯減少，較兩年前參加該會議時，論文數量減少了差不多一倍，不過論文接受率則維持差不多大約 4 成，顯示舉辦單位並沒有因為投稿論文數下降而降低篩選標準。，報到完成，拿到一個背包，內含一份會議議程及論文光碟，報到完成後，進 Niihou 廳聽大會主席介紹會議發展現況與未來展望，隨後聽取 Fuzzy 大師 Prof. Lotfi A. Zadeh 介紹他們最近所提出來的一個叫做 FLu 的理論，有別於傳統只接受精準輸入資料的 Fuzzy 理論，FLu 理論是一種建構在可接受非精準輸入資料的 Fuzzy 系統。

會議第二天(8 月 18 日)，我的論文安排在下午 1:30 報告，早上時間準備下午的報告，1:10 到達會場，原會場應有 8 人報告，實際上場報告的則只有我、會議主席、與兩個來自德國的報告者。由於報告人數減少，每個人可以有更多的時間可以跟與會人員討論，所以整場會議結束時間並沒有提前結束，報告過程順利，報告完成，獲得了一張證書，證明我確實有出席會議及上台報告。會議主席很年輕是新加坡的華人，目前在馬來西亞任教，會後交換彼此資料，希望後續能有跨國合作的機會。

會議第三天(8 月 19 日)聽了幾場有興趣的論文，由於是最後一天，會議出席人數不多，會議結束後也趁機看看夏威夷的風景及放鬆一下心情。8 月 21 日搭乘早上 9:45 華航經東京到台北的班機，回到台灣已經是 8 月 22 日下午 4:30。

在這次會議有限的休息及交流時間中，有幸認識一個來自德國從事聲音訊號處理的博士班學生，一個來自泰國從事市場經濟的教授，一個中國海洋大學的教授，兩個台灣來的博士班學生，以及一個明新科大的教授。

二、與會心得

由於已經出國報告許多次，對於上台報告的過程比較熟悉，因此在這一次的論文發表過程中，事前的準備時間大符縮短許多，也可以有較多的時間可以跟來至其它國家的學者交流，唯會議時間有限，無法跟太多學者交談。此行最大的收獲應是與來自馬來西亞的 Kok-Swee Sim 教授建立初步合作關係，獲邀擔任其博士班學生的審查委員；2009 年十月，本人參與 Sim 教授的跨國團隊，共同提出計劃書，申請法國 ICT 的研究計劃；本人也邀請 Kok-Swee Sim 教授於 2009 年 12 月上旬到本校(南華大學)及工研院演講，未來希望能透過出國發表論文的機會多跟國外學者交流，提高跨國合作機會。

FAST EXACT PAIRWISE-NEAREST-NEIGHBOR ALGORITHM USING GROUPS AND CLUSTERS REJECTION CRITERIA

Yi-Ching Liaw[1], Jun-Feng Lin[2], Shen-Chuan Tai[2], Jim Z. C. Lai[3]

[1]Nanhua University, Chiayi, Taiwan 622, R. O. C.
[2]National Cheng-Kung University, Tainan , 701, R. O. C.
[3]National Taiwan Ocean University, Keelung, Taiwan 202, R. O. C.

ycliaw@mail.nhu.edu.tw, jflin@mail.nhu.edu.tw, sctai@mail.ncku.edu.tw, zclai@mail.ntou.edu.tw

ABSTRACT

Pairwise-nearest-neighbor (PNN) is an effective method of data clustering, which can usually generate good clustering results, but with high computational complexity. In this paper, a new method is presented to reduce the computational complexity of the PNN algorithm through dividing clusters into groups of clusters and using projections of clusters on differential vectors of group pairs to reject impossible groups and clusters in the nearest neighbor finding process of a cluster. Experimental results show that the proposed algorithm can effectively reduce the computing time and number of distance calculations of the PNN algorithm for data sets from real images. It is noted that the proposed method generates the same clustering results as those produced using the PNN algorithm.

KEY WORDS

Pairwise-Nearest-Neighbor, Clustering, Fast algorithm, and Projection.

1. Introduction

Data clustering is a technique to classify data points into several clusters. To classify data points into clusters is not an easy task and usually requires a large amount of computations. There are many heuristic methods [1-3] available to categorize data points into clusters. Among these available approaches, the pairwise-nearest-neighbor (PNN) algorithm [3] can usually generate better clustering results [4] than other methods. For a data set of N data points, the computational complexity of the PNN algorithm is $O(N^3)$. It is impractical for a large data set.

To reduce the complexity of the PNN algorithm, Fränti et al. [5] presented a fast exact PNN algorithm (referred to as FPNN). The FPNN algorithm has a computational complexity of $O(\tau N^2)$, where τ is about 5 to 12 [5] for most cases. Comparing to the PNN method, FPNN can effectively reduce the computational complexity of PNN and generate the same results as those generated using the PNN algorithm.

However, the computational complexity of FPNN is still too high and can be further reduced. In our observation, we found that the FPNN algorithm spends much computing time on finding nearest neighbors for clusters. In this paper, we will present a new method to speed up the nearest neighbor finding process of FPNN.

This paper is organized as follows. Section II reviews the PNN and the FPNN algorithms. Section III introduces the algorithm developed in this paper. Experimental results are given in Section IV and concluding remarks are presented in Section V.

2. Background

2.1. PNN algorithm

Given a set of N data points, the pairwise nearest neighbor (PNN) algorithm [3] is used to cluster data points into M clusters through a series of merging processes. In the beginning, the number of clusters is set to N and each cluster contains only one data point. The merging process of PNN includes two steps: finding the nearest pair of clusters from all clusters and merging this nearest pair into a new cluster. After a merging process, the number of clusters will be decreased by one. The merging process is repeated until the number of clusters is equal to M.

To find the nearest pair of clusters, the distances for all pairs of clusters should be calculated. Let R_a and R_b be two clusters, C_a be the center of R_a, and C_b be the center of R_b. Let n_a and n_b be the numbers of data points in R_a and R_b, respectively. The distance between clusters R_a and R_b is defined as follows.

$$D_{a,b} = \frac{n_a n_b}{n_a + n_b} d\left(C_a, C_b\right) \tag{1}$$

where $d(C_a,C_b)$ is the squared Euclidean distance between C_a and C_b. Let $X=[x_1, x_2, ..., x_d]^t$ and $Y=[y_1, y_2, ..., y_d]^t$. The squared Euclidean distance $d(X, Y)$ is defined as

$$d(X,Y) = \sum_{k=1}^{d} |x_k - y_k|^2 \quad (2)$$

Let R_{ab} be the new cluster obtained through merging clusters R_a and R_b. The cluster center C_{ab} of R_{ab} and the number of data points in R_{ab} are evaluated as follows.

$$C_{ab} = \frac{n_a C_a + n_b C_b}{n_a + n_b} \quad (3)$$

$$n_{ab} = n_a + n_b \quad (4)$$

After a merging process, cluster R_a is replaced by R_{ab} and cluster R_b should be deleted.

2.2. FPNN algorithm

To reduce the computational complexity of the PNN algorithm, Fränti et al. used a nearest neighbor table *NN* and a nearest distance table *ND*, which respectively, records the nearest neighbor and the distance to the nearest neighbor for every cluster. It is noted that $NN[s]$ and $ND[s]$ denote the nearest neighbor and the distance to the nearest neighbor of cluster R_s, respectively. If two clusters R_a and R_b are merged, then tables *NN* and *ND* must be updated. Since the distance to the nearest neighbor of a cluster increases monotonically as the merging process proceeds [5], only items of tables corresponding to clusters with either cluster R_a or cluster R_b as their nearest neighbor should be updated. That is, $S=\{R_s: NN[s]=R_a \text{ or } NN[s]=R_b\}\backslash\{R_a, R_b\}$ is the set of clusters whose items in tables *NN* and *ND* should be updated.

3. Proposed Algorithm

In the proposed method, clusters are first divided into several groups of clusters using a fast cluster separation algorithm. Then, the rejection criteria and projections of clusters on differential vectors of group pairs are used to speed up the nearest neighbor searching process of a cluster. The cluster separation algorithm and fast nearest neighbor searching process are introduced in the following subsections.

3.1. Cluster Separation Algorithm

There are many methods [1-3, 6-7] could be used to categorize clusters into groups of clusters. Our objective here is to roughly and quickly divide clusters into several groups of clusters. To meet this requirement, a simple method for building a Kd-tree proposed by Kanungo et al. [6] is modified to generate the initial set of groups of clusters. Then, the General Lloyd Algorithm [1] is executed several times to refine these groups. The detail of the cluster separation algorithm is stated below.

Given a set of N clusters of dimension d, our problem is to divide N clusters into P groups of clusters. In this paper, we set $P=\sqrt{N}$. Let $C_q=[c_{q,1}, c_{q,2}, \ldots, c_{q,d}]^t$ be the centroid of a cluster R_q ($q=1, 2, ..., N$), S_G be the set of groups of clusters, and ng be the number of groups. In the beginning, we set $ng = 1$ and $S_G=\{G_1\}$, where $G_1=\{R_q: q=1, 2, ..., N\}$. Let $span_{i,k}$ be the span of kth axis of group G_i. The value of $span_{i,k}$ is defined as follows.

$$span_{i,k} = max\{c_{q,k}: R_q \in G_i\} - min\{c_{q,k}: R_q \in G_i\} \quad (5)$$

Then, the longest axis la_i of group G_i could be determined by

$$la_i = \arg\max_k \{span_{i,k}\}. \quad (6)$$

Let med_i be the median value of the longest axis la_i of group G_i. That is,

$$med_i = median\{c_{q,la_i} : R_q \in G_i\}. \quad (7)$$

The group G_i can be divided into two groups G_i and G_{ng+1} using med_i. Intially, group G_{ng+1} is empty. For a cluster $R_q \in G_i$, if c_{q,la_i} is greater than med_i, R_q is moved from group G_i to group G_{ng+1}. Otherwise, R_q stays in group G_i. Once the separation process of group G_i is completed, add group G_{ng+1} to S_G and increase the value of ng by one. If ($ng < P$), find a group with the largest span on its longest axis from S_G and repeat the above cluster separation process until $ng = P$. The clustering result could be further refined by executing the General Lloyd Algorithm (GLA) [1] several times. In this paper, the GLA is executed 3 times in consideration of the execution speed and the resulting quality.

3.2. Fast Nearest Neighbor Search Using Groups and Clusters Rejection Criteria

Given a set of groups of clusters $S_G = \{G_i: i=1, 2, ..., P\}$ and a query cluster $R_q \in G_L$, our task is to find the nearest cluster of R_q from groups of clusters in S_G. The nearest neighbor searching process for a cluster here is divided into two phases: the local nearest neighbor finding phase and the global nearest neighbor finding phase.

In the local nearest neighbor finding phase, a cluster R_l belongs to the group G_L and is nearest to the cluster R_q is determined using the following equation.

$$l = \arg\min_s \{D_{q,s} : R_s \in G_L \text{ and } R_s \neq R_q\}. \quad (8)$$

Since R_l and R_q belongs to the same group, the distance between R_l and R_q is expected to be small and is a good start to find the global nearest neighbor for cluster R_q. In our observation, the local nearest neighbor of a cluster has a large probability to be the global nearest neighbor of the cluster. From the observation on 32 real images, the average probability for the local nearest neighbor of a cluster is also the global nearest neighbor of the cluster is about 68%. Once the local nearest neighbor R_l of cluster R_q is found, the distance between clusters R_l and R_q is used in the successive global nearest neighbor finding phase to reject impossible groups and clusters.

In the global nearest neighbor finding phase, each cluster in groups other than the group contains the query cluster R_q should be checked to see if it is the nearest neighbor of cluster R_q. The global nearest neighbor R_g of cluster R_q is determined by

$$g = \arg\min_s \{D_{q,s} : R_s \in (G_i \cup R_l), G_i \in S_G, \text{and } i \neq L\} \quad (9)$$

The global nearest neighbor finding phase is very time consuming. To speed up this phase, differential vectors between group pairs and projections of every cluster on differential vectors of group pairs are pre-calculated and stored. Let $v_{i,j}$ be the differential vector between groups G_i and G_j. The differential vector $v_{i,j}$ is defined as follows.

$$v_{i,j} = \frac{(C_{G_i} - C_{G_j})}{|C_{G_i} - C_{G_j}|} \quad (10)$$

where C_{G_i} and C_{G_j} are the centroids of groups G_i and G_j, respectively. The projection of a cluster R_s on vector $v_{i,j}$ can be calculated using the following equation.

$$p^s_{i,j} = C_s \cdot v_{i,j} . \quad (11)$$

Let $d_{\min}$ be the distance between the query cluster R_q and its candidate nearest neighbor. In the beginning of the global nearest neighbor finding phase, $d_{\min}$ is set to the distance between cluster R_q and its local nearest neighbor R_l. Once $d_{\min}$ is given, every group $G_i \in S_G$ other than group G_L could be checked using the following inequality.

$$\left(p^q_{L,i} + p^{I_{i,L}}_{i,L}\right)^2 \times \frac{n_q}{n_q + 1} > d_{\min} \quad (12)$$

$$I_{i,L} = \arg\min_s \{p^s_{i,L} : R_s \in G_i\} \quad (13)$$

Where $I_{i,L}$ is the index of the cluster with minimum projection on vector $v_{i,L}$ in group G_i and $p^{I_{i,L}}_{i,L}$ is the minimum projection of clusters in group G_i on vector $v_{i,L}$. Let $P_{i,L}= p^{I_{i,L}}_{i,L}$ for a group $G_i \in S_G$ and $G_i \neq G_L$. Where $P_{i,L}$ and $I_{i,L}$ should also be pre-calculated and stored.
In the case that inequality (12) is satisfied, we know that every cluster in group G_i is impossible to be the nearest neighbor of cluster R_q and group G_i can be rejected. That is the distance calculations between cluster R_q and clusters in group G_i can be saved. If a group G_i cannot be rejected by inequality (12), clusters in group G_i can be further rejected using inequality (14).

$$\left(p^q_{L,i} + p^s_{i,L}\right)^2 \times \frac{n_q \times n_s}{n_q + n_s} > d_{\min} \quad (14)$$

where $p^s_{i,L}$ is the projection of a cluster $R_s \in G_i$ on vector $v_{i,L}$.

Once a cluster R_s cannot be rejected through using inequalities (12) and (14), the distance between cluster R_s and cluster R_q must be computed using equation (1) and $d_{\min}$ should be updated using the distance between cluster R_q and its updated nearest neighbor.

During the finding process of nearest neighbors for all clusters, the nearest pair of clusters R_a and R_b can also be determined. In the merging process, clusters R_a and R_b are merged into cluster R_{ab} and the cluster R_b should be deleted.

The merging process may cause that the nearest group of cluster R_{ab} differs from that of cluster R_a. Therefore, after a merging process of the nearest pair of clusters, cluster R_{ab} must be relocated. That is, we must find a nearest group for cluster R_{ab} and then move cluster R_{ab} to its nearest group. Moreover, the deletion of cluster R_b and the migration of cluster R_{ab} may induce the change of group centroids. To avoid relocating all clusters into groups, we do not update the centroids of groups after a merging process. However, after a merging process, the projections of cluster R_{ab} on differential vectors between the group containing the cluster R_{ab} and other groups must be calculated and the minimum projections for the groups of cluster R_a, R_b, and R_{ab} should also be updated. To speed up the updating process of the minimum projections, in the case of $I_{L,i}$ is not equal to the index of R_a or R_b and cluster R_{ab} still stay in group G_L, the updating process of the minimum projection for clusters in group G_L on vector $v_{L,i}$ for each group $G_i \in S_G$ and $G_i \neq G_L$ could be simplified by computing only the projection of cluster R_{ab} on vector $v_{L,i}$. If the projection of cluster R_{ab} on vector $v_{L,i}$ is less than the value of $P_{L,i}$, set $P_{L,i}$ equal to the projection of cluster R_{ab} on $v_{L,i}$ and set $I_{L,i}$ equal to the index of cluster R_{ab}.

The above proposed algorithm is applied on FPNN algorithm [5] to speed up the nearest neighbor finding process for a cluster whose items in tables *NN* and *ND* are going to be updated. Beside the above inequalities, one additional inequality can also be used to help eliminate impossible clusters in the updating process of *NN* and *ND*. Since $ND[s]$ is the nearest distance between cluster R_s to its nearest neighbor, if $ND[s] \geq d_{\min}$, cluster R_s is impossible to be the nearest neighbor of cluster R_q and the cluster R_s can be rejected. This inequality comes from that the distance between a cluster and its nearest neighbor will be monotonically increasing [5] as the cluster merging process proceeds.

4. Experimental Results

To evaluate the performance of the proposed algorithm, seven gray images (“Lena,” “Peppers,” “Baboon,” “Lake,” “Tiffany,” “F16,” and “Parrot”) with image size of 512×512 are used as data sets. Each data set consists of 16384 non-overlapping image blocks of size 4×4 pixels extracted from an individual image. Each 4×4 image block is a data point of dimension 16. Our task is to classify data points of each data set into 256 clusters using the PNN, FPNN and our proposed algorithms. These algorithms are compared in terms of the number of distance calculations and the computing time. All programs were implemented as console applications of Microsoft Visual C++ 2008 and were executed on an Intel Pentium Dual-Core 2.5GHz PC with memory of 2 GB under Windows XP Professional SP3.

Tables I and II, respectively, list the computing time and the number of distance calculations for PNN, FPNN and the proposed algorithm to generate 256 clusters using data sets from real images. All these methods generate the same results.

Table I. The computing time (seconds) for three algorithms to generate 256 clusters using data sets from real images.

Images	PNN	FPNN	Proposed
Lena	23218.6	34.2	4.7
Peppers	23418.8	35.5	5.2
Baboon	23173.9	34.4	10.8
Lake	22853.4	32.1	6.0
Tiffany	23331.2	36.2	6.9
F16	22932.9	30.4	5.4
Parrot	23088.8	33.3	4.4
Average	23145.2	33.7	6.2

Table II. The number of distance calculations per data point for three algorithms to generate 256 clusters using data sets from real images.

Images	PNN	FPNN	Proposed
Lena	44739072	46233	1335
Peppers	44739072	47412	1595
Baboon	44739072	49654	2501
Lake	44739072	46474	1902
Tiffany	44739072	49220	2245
F16	44739072	45499	2131
Parrot	44739072	46661	1297
Average	44739072	47307	1858

From table I and table II, we can see that our proposed algorithm has the better performance than PNN and FPNN both in terms of the computing time and number of distance calculations. Compared with the PNN algorithm, FPNN can reduce the average computing time and number of distance calculations significantly. While comparing to the FPNN algorithm, our method can further reduce the average computing time and number of distance calculations of FPNN by 81.6% and 96.1%, respectively.

5. Conclusions

In this paper, we presented a new method to reduce the computing time of Pairwise-Nearest-Neighbor (PNN) algorithm using groups and clusters rejection criteria. Through dividing clusters into groups and exploiting the projections of clusters on differential vectors of group pairs to reject impossible groups and clusters in the nearest neighbor finding process of a cluster, the computing time of PNN is reduced significantly. Experimental results show that the proposed algorithm can effectively reduce the computing time and number of distance calculations of PNN algorithm. Compared with FPNN, which is an effective fast exact PNN algorithm, our method can reduce the computing time and number of distance calculations of FPNN in average by 81.6% and 96.1%, respectively, for data sets from real images.

Acknowledgements

This work is supported by the National Science Council in Taiwan (NSC-97-2221-E-343-006).

References

[1] Y. Linde, A. Buzo and R.M. Gray, An algorithm for vector quantizer design, *IEEE Transactions on Communications*, *28*(1), 1980, 84-95.
[2] A. E. Cetin and V. Weerackody, Design vector quantizers using simulated annealing, *IEEE Transactions on Circuits and Systems*, *35*(12), 1988, 1550-1550.
[3] W. H. Equitz, A new vector quantization clustering algorithm, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *37*(10), 1989, 1568–1575.
[4] J. Shanbehzadeh and P. O. Ogunbona, On the computational complexity of the LBG and PNN algorithm, *IEEE Transactions on Image Processing*, *6*(4), 1997, 614-616.
[5] P. Fränti, T. Kaukoranta, D. F. Shen, and K. S. Chang, Fast and memory efficient implementation of the exact PNN, *IEEE Transactions on Image Processing*, *9*(5), 2000, 773-777.
[6] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, An efficient k-means clustering algorithm: analysis and implementation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(7), 2002, 881-892.
[7] Jim Z.C. Lai, Yi-Ching Liaw, and Julie Liu, A fast VQ codebook generation algorithm using codeword displacement, *Pattern Recognition*, *41*(1), 2008, 315-319.