具長時間音訊觀測之無線感測器網路設計

吳鎮安
屏東科技大學資訊管理系
碩士班學生
m9556030@mail.npust.edu.tw

蘇軍維
屏東科技大學資訊管理系
碩士班學生
m9656027@mail.npust.edu.tw

劉寧漢*
屏東科技大學資訊管理系
助理教授
gregliu@mail.npust.edu.tw

摘要

無線感測器網路雖然具有無線傳輸、不需事先佈線等特性，但其最大的限制在於電力保存的問題。無線感測器利用無線傳輸的耗電量遠大於內部運算，因此如果能在感測器上先做判斷分析，決定要不要將資料傳回到伺服器，會比一收到資料就傳回伺服器的方法更為省電。而目前無線感測器的處理能力並不強，如何在有限的記憶體容量與處理器速度下進行運算，且同時保有即時辨識與準確性，是一個值得討論研究的問題。本研究在無線感測器上嵌入音訊辨識模組，在感測器上利用音量差異總和函數先做音訊特徵值的擷取與分析，再決定要不要將資料傳回到伺服器，藉此達到延長感測器運作時間的目的。經實驗證明，在感測器上先做辨識的方法，的確可以有效節省感測器的電力消耗。

關鍵詞：音訊特徵擷取，音量差異總和函數，高斯混合模型，無線感測器網路。

Abstract

Wireless sensor networks can transmit data wirelessly, and they do not necessarily be sited in advance, but the biggest limit of wireless sensor networks is power saving. Sensors will waste much power with transmitting data wirelessly than computing inside. On the sensors, the data are analyzed and determined that they will be transmitted to server or abandoned, the power of batteries will be saved. The ability of sensors is not powerful, so it is worth researching how to keep analyzing speed and accuracy under restricted memory and processor. This research proposed an embedded audio processing module on the sensors. The module will extract audio features with sum of magnitude difference function and analyze in advance on the sensors, then it decide to transmit data to server or not. It will extend observing time of sensors. According to the results of experiments, it could save the power of sensors effectively.

Keywords: audio features extraction, sum of magnitude difference function, gaussian mixture models, wireless sensor networks

1. 前言

現代科技的快速發展，無線傳輸技術漸趨成熟，電子產品的開發也以體積小、方便攜帶為發展目標，這使得無線感測器網路(Wireless Sensor Networks, WSN)[11]的研究與應用也逐漸地普及。無線感測器網路具有可即時監控、不需事先佈線、無線傳輸等特性[1]，因此目前多用在偵測周遭環境的改變，如溫度、溼度、聲音等等。

無線感測器的低成本、用完即可棄置與任意擺放也能互相傳輸資料的便利性，使得研究者能將無線感測器網路應用在人煙稀少與人員不便到達的危險地區，以進行資料的收集與回傳，使用上不但極為方便，還可以省下龐大的佈線與維護費用。但無線感測器網路在應用上有個重要的限制需要解決，即電力耗損的問題。由於無線感測器的電力是由電池供應，電池供電的持續時間不算太長，而感測器一次無線傳輸所耗費的電力往往可提供感測器做數千到數萬次的內部運算[20]，若是感測器採用一接收到資料就全數回傳到伺服器端的方法，那麼電力耗盡的情況會常常發生，這時人員奔波兩地來更換電池就會變成繁瑣的工作。

目前把音訊處理技術應用在無線感測器網路上的研究仍不多，且由於音訊處理技術所耗費的記憶體空間與處理器資源甚大，對於處理功能並不強大的無線感測器[8]而言，要兼顧系統執行效能、即時運算與音訊比對的準確度將是一大挑戰。感測器上的簡單運算可以達到即時辨識的要求，也不需耗費太多的記憶體空間與系統資源，但準確度會受到不小的影響，相反地若是使用複雜的演算法，雖然可以獲得較佳的辨識率，但對系統資源是一大負擔，也無法適用在無線感測器網路上。

由於不同科別(family)的生物之間通常都有其獨特的叫聲，即使是同科但不同屬別

*為通訊作者

(genus)或不同種別(species)的生物叫聲也不盡相同。因此本研究針對這個生物特性，並選定棲息於台灣的數十種蛙類做為觀測對象，除了是因為現在聲音辨識技術多用在人類聲音上，對生物聲音的辨識研究還不是很多，另一個原因是蛙類除了會在繁殖季節的白天下雨天出現求偶外，其他時間多在晚上活動，與人類主要活動時間不盡相同，因此對觀測者而言，最好是能有個自動化的觀測及辨識系統來協助觀測生物的活動。

在本研究中，使用到的音訊辨識技術主要是音量差異總和函數(Sum of Magnitude Difference Function, SMDF)[10]，將偵測到的蛙叫聲先在感測器端進行音高等特徵值的擷取，之後利用音高的分佈型態來比較，以決定是否要將音訊特徵值傳回到伺服器做進一步分析，或是在感測端就把資料刪除。而在伺服器端則使用其他過去已提出的分類演算法，如高斯混合模型(Gaussian Mixture Models, GMM)[15][16]等來分析判斷這些聲音特徵值是哪一類的蛙類，以驗證簡單的音高分佈型態比較法是否能正確辨識聲音，藉由這種自動觀測的方式來收集生物的叫聲並自動做聲音的分析與判斷的方式，來減少感測器電力的消耗，延長感測器的運作時間，避免因經常更換電池造成的人力浪費。

本文其他章節結構如下：第 2 節為相關文獻的探討，針對目前無線感測器網路與音訊辨識技術的發展演進略加說明。第 3 節介紹本研究的整體架構與所使用的研究方法。第 4 節說明實驗環境設定與最終的實驗結果。最後第 5 節是實驗後得到的結論與未來的研究方向。

2. 相關文獻

無線感測器網路是指由許多的感測器(sensor)以及數個可移動式基地台(base station)所構成的網路系統[11][17]，感測器與基地台之間和感測器彼此之間的通訊方式是採用無線通訊方式[12][27]，由各個感測器負責收集鄰近地區的資料後，將資料依序傳回到基地台，基地台接收到資料後，再將運算結果傳給最終的工作站或伺服器。

過去無線感測器網路的應用主要是偵測感測器附近環境的改變，如感測人員出入活動的保全偵測、危險橋樑的監測控管、室內外溫濕度感應等等，由於近代科技發達，無線傳輸技術漸趨成熟，這使得無線感測器網路的研究與應用也逐漸地普及。

無線感測器網路的運行方式與傳統的無基礎架構網路(ad hoc network)非常相似，但因為無線感測器網路的感測器數量更多、網路拓樸更容易因感測器故障而發生變動，加上感測器的電力、運算能力、記憶體受到很大之限制，及無線感測器網路沒有共通的IP位置的識別證，使得現有的無基礎架構網路的通訊協定及演算法大都無法直接應用到無線感測器網路上[22]。

感測器主要有幾個資源限制：有限的溝通傳輸能力，電力耗損的問題，運算能力受限，與感測資料不穩定[27]。由於感測器的電力是由電池供應，電力壽命不算太長，加上感測器有著可隨意散佈的特性，大多將無線感測器使用在人員不便到達之地區，這使得在感測器電力耗損時，人員更換電池將會是個困難的任務。所以在設計上必須考量電力耗損的問題，不能讓感測器一直在工作滿載的情況下運作或進行不必要的資料傳輸而太早耗盡電力。如何保存感測器電力一直是無線感測器網路主要的設計重點[27]，以使用兩顆AA電池的MICA為例[8]，對感測器提供了 2000 毫安培的電力供應，而處於閒置狀態的感測器，電力約可維持一年，但若是一直處於滿載情況，電力在一週後即會耗盡。

而感測器是採用無線傳輸方式，因此傳輸資料的頻寬有限，這會造成傳輸時有著高變異性的問題[13]。除了感測器本身的因素，所在環境的影響或是佈建位置的不適當也會造成感測器偵測資料不正確，以負責偵測聲音的感測器為例，除了收集到想要的聲音外，也有可能會包含不必要的環境噪音[27]。在上述這些限制中，感測器的電力耗損問題是最為重要的，這也是無線感測器網路與ad hoc網路最大的差異[7][12]。

音訊處理(audio signals processing)的研究與應用，在過去數十年已有長足的進步。聲音在空氣中傳播，經過了麥克風之類的收集轉換器，會轉變成電的訊號，再經過類比轉數位轉換器，就會變成數位訊號，這時才能將數位訊號拿來作進一步的分析處理。目前音訊處理的應用技術有語音編碼、語音合成、語音辨認[5][24]等，另外在語言教學、去除噪音、比對聲紋[4]、門禁保全系統[25]，也都是目前音訊處理的應用領域。

一般聲音有幾個物理上的特性：音量(volume)、音高(pitch)、與音色(timbre)[10]。音量代表的是聲音的強度，也叫做能量(energy)，通常音量越大，訊號振幅也越大；音高是訊號振動的頻率，音高較低，聲帶的振動也越緩慢，而此頻率指的是基本頻率(fundamental frequency)，也就是基本週期(fundamental period)的倒數[3]，通常在分辨聲音特徵時，主要是看聲音的音高有無不同；音色是聲音的特質，主要受到發聲器官、共鳴器不同的影響。

聲音訊號是一種隨時間變化(time varying)的訊號，波形的變化非常快速，但如果將訊號以短時間來觀察，可以發現聲音的變化是很穩定的，因此在處理聲音訊號前通常會先將聲音切割成一個個小單位，即音框(frame)，以便找出訊號的週期或規律變化[26]。若音框切割的太大，無法看出音訊隨時間變化的特性；反之，若音框太小，也無法得到音訊的特性，通常音框要能包含數個音訊的基本週期。

目前在使用音訊處理技術比對聲音樣本方面，大致可分為兩部分，先對觀測到的聲音訊號進行特徵擷取，再將擷取出來的特徵向量傳給分類器進行分類[1][19]。擷取音訊特徵通常有下列步驟[9]：將音訊切成一個個音框，並擷取音框的特徵。在比對兩個聲音訊號時，並不是直接將兩個音訊拿來做比較，而是先將聲音做特徵擷取的工作，即計算出可以代表這個聲音的一些特徵值，如能量、音高、或過零率(Zero Crossing Rate, ZCR)[18][21]等。

一般求取音高的演算法有：自相關函數(Autocorrelation Function, ACF)，平均音量差異函數(Average Magnitude Difference Function, AMDF)，音量差異總和函數(Sum of Magnitude Difference Function, SMDF)等[10]。這三者的運算觀念類似，但SMDF不似ACF與AMDF牽涉到乘法與除法，運算上只使用到加法，比較不會耗費太多系統資源，適合應用在低階處理器上。

過零率(ZCR)的定義是每個音框中，訊號通過零點的次數，其具有下列特性[18][21]：雜訊及氣音的過零率比有聲音大；常用在端點偵測，尤其是用在預估氣音的開始點和結束點；可用來預估訊號的基頻。

特徵值擷取出來後，便開始進行分析，目前最常使用在音訊分類的是高斯混合模型[15][16]、隱藏馬可夫模型(Hidden Markov Models, HMM)[14]，及K近鄰演算法(K-Nearest Neighbor, KNN) [6]。

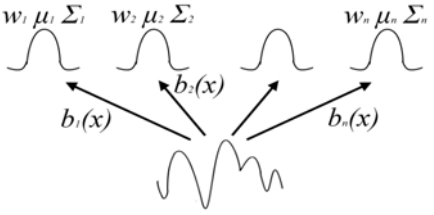

圖1. 高斯混合模型示意圖

隱藏馬可夫模型較高斯混合模型運作更為複雜，且需要更多的訓練語料[16]，因此高斯混合模型較適合應用在低階處理器上。一個完整的高斯混合密度可以用數個混合權重、平均值向量和共變異數矩陣來表示，如圖1所示，下方的波形為原本的聲音訊號，可分解成n個權重值不同的單一高斯分佈，其公式為：$p(x|\lambda)=\sum_{i=1}^{M} w_i b_i(x)$，其中$x$是D維的隨機向量，$b_i(x)$是基本密度，$w$是混合權重，$\mu$與$\Sigma$分別是該高斯分佈的平均值與變異數，$\lambda$是這些參數的集合：$\lambda=\{w_i,\mu_i,\sum_i\}, i=1,2,3,...,M$。高斯混合模型的主要特性是能夠平滑地接近任意形狀的資料分佈，原理為把同一聲音的聲音特性做分群，然後再把每一群的聲音特性用一個高斯分佈來描述。

K近鄰演算法是屬於傳統統計的辨識法之一，用最直覺的想法，決定每一個資料所歸屬的類別，即找最接近的鄰近點來判定屬於哪一類，K近鄰演算法也常用在聲音的分類上。其步驟為：設定初始值，尋找最近的鄰近點，更新編碼向量(codevectors)，重複以上步驟直到每個資料和編碼向量的平均距離小於自訂的門檻值[23]。

3. 系統架構與方法

本研究假設的實驗情境是一個略有雜音的野生環境，除了要觀測的蛙叫聲外，還有一些音量略小的風雨聲、樹葉飄落聲等。將數個無線感測器布置於其中，無線感測器在偵測到聲音後，會在感測器上先做聲音特徵值的擷取與判斷，如果判斷結果是蛙叫聲，會再將辨識結果與音訊特徵值傳回伺服器端記錄並做進一步地辨識。最後並驗證「感測器接收到聲音後即將資料傳回伺服器」與「感測器先做辨識後再決定要不要傳回伺服器」兩者間的耗電量有何不同。

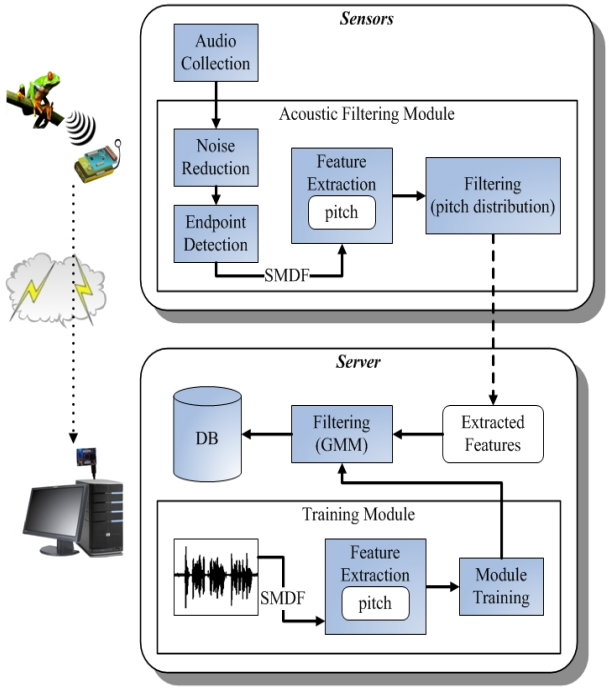

圖 2. 辨識系統架構圖

在佈建無線感測器網路前，要先製作訓練模組(Training Module)，系統架構如圖 2 所示。先將要辨識的蛙叫聲樣本資料做特徵值的擷取，在本研究中是使用經 SMDF 計算出來的音高值，之後再經由模組化的訓練，得到這些聲音的訓練模型，作為之後音訊比對的依據。

這些訓練過的辨識模組會分別放置在感測器端與伺服器端。在感測器端放置的是判斷較簡易的辨識演算法模組，如果觀測環境有異常聲響的產生，感測器會開始進行聲音的記錄，並開始執行聲音過濾模組(Acoustic Filtering Module)。這模組的工作步驟主要有雜訊去除(Noise Reduction)：去除不必要的背景噪音，只保留系統需要的聲音；端點偵測(Endpoint Detection)：決定系統要分析的聲音段落；與特徵擷取(Feature Extraction)的工作，然後再與之前建立好的音訊辨識模組在感測器端先做初步的判斷。

在經過感測器端的音訊辨識模型判斷後，若是偵測到的聲音可能為蛙叫聲，就會把已萃取過的聲音特徵值透過無線網路傳回伺服器端做更進一步地識別；而如果初步判斷不是系統所要的聲音，就會在感測器端直接將聲音捨去，並不會將聲音全部傳回伺服器端來進行判斷，這樣即可以減少因為感測器之間的傳輸過於頻繁而導致電力提早耗盡的問題。

伺服器端部分，由於處理速度較快、記憶體空間較感測器大，其辨識模組可以採用較複雜、辨識準確率較佳的模組，本研究中所採用的是高斯混合模型。在接收到感測器傳回已擷取出的音高後，會與伺服器內部之前已訓練好的辨識模組進行比對工作，這部分的比對工作將會比感測器端更為仔細，以求獲得更佳的辨識準確度，一旦系統判斷環境中有蛙類出沒，即會將偵測時間與地點記錄於資料庫中，以供生物學家日後查詢。

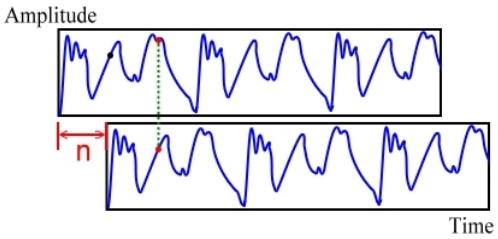

圖 3. SMDF 平移示意圖

3.1 從聲音檔中擷取特徵值

在收集到所有種類的蛙叫聲後，將這些聲音檔做音訊特徵擷取的工作，在本研究中選擇的音訊特徵值為經由 SMDF 擷取出的音高。

在擷取音訊特徵值前，必須先將整段聲音切割成一個個小單位的音框，而相鄰音框之間是可以重疊的。讓音框重疊的用意是希望相鄰音框之間的變化不會太大，即最後求得的音高曲線較有連續性，基本上音框長度必須要包含兩個基本週期以上，才能顯示出語音的特性。

接下來為了增加音框左右兩端的連續性，每一個音框都要乘上漢明窗，漢明窗的運算式為式(1)；N 為音框長度，W 為乘上漢明窗後的音框，一般取 a 為 0.46。接下來就可以從這些音框中找出該音訊的特徵值。

$$W(n,a) = (1-a) - a * \cos\frac{2\pi n}{N-1} \quad \text{.....(1)}$$
$$0 \le n \le N-1$$

由於SMDF的運算較簡單且不會耗費太多系統資源，比較適合本研究的作業環境，因此本研究將採用 SMDF 來偵測音高。SMDF 的運算式為式(2)；n 為音框往右平移的長度，$x(p)$ 為音框內各點振幅，M 為音框長度。

$$D_m(n) = \sum_{p=0}^{M-1} \left| x_m(p) - x_m(p+n) \right| \quad \text{.....(2)}$$
$$n = 0,1,2,3.....$$

SMDF 的計算含意為將音框每次向右平移 n 單位，如圖 3 所示，再和原本的音框重疊部分做兩點間的相減，接下來再進行取絕對值、

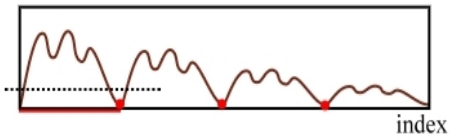

圖 4. SMDF 計算結果圖

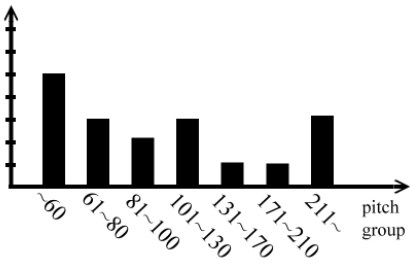

圖 5. 音高分佈狀況圖

加總的運算，重複 M 次後會得到 M 個內積值。最後運算的結果如圖 4，由圖可以看出 SMDF 的最小值是在起始點，其值為 0。因此可以尋求一個門檻值(虛線)，找出不包含起始點而且低於此門檻值的最低點(圓點)，再找出這些位置的索引值，取其最小者即為此音訊的音高。

3.2 訓練模組的建立

隱藏式馬可夫模型在音訊辨識的表現會比高斯混合模型好，但因為其系統較為複雜，而且高斯混合模型是隱藏式馬可夫模型的簡化，因此以高斯混合模型較適合使用於本系統。

在高斯混合模型的聲音識別中，假設每個聲音都有屬於自己的模型，即為所有參數的集合，也稱做基本密度。高斯混合密度是 M 個基本密度的加權總合(weighted sum)，其運算式為式(3)：

$$p(\vec{x} \mid \lambda) = \sum_{i=1}^{M} w_i b_i(\vec{x}) \text{.....(3)}$$

而每個 λ 的三個參數：混合權重 W_i (mean vector)、共變異數矩陣 $\sum_i$ (covariance matrix)和平均值向量 $\vec{\mu}_i$ (mixture weight)，如式(4)所示；M 為高斯分佈的個數。

$$\lambda = \{w_i, \vec{\mu}_i, \sum_i\} \quad i = 1,2,3,....M \text{.....(4)}$$

為了找出最能代表聲音特徵向量分佈的參數集合，在估算高斯混合模型參數的方法中，主要是用最大可能性估算法(Maximum Likelihood, ML)。假設有一訓練語句經過特徵參數擷取處理後得到 T 個訓練特徵參數向量，

表 1. 各種生物叫聲的音高分佈狀況

sound \ pitch	~60	61~80	81~100	101~130	131~170	171~210	211~
牛	7	6	5	4	0	2	1
羊	9	2	6	3	0	4	1
狗	3	4	2	7	4	3	2
馬	5	4	6	3	1	3	3
雞	7	5	2	1	6	0	4
小雨蛙	7	3	2	4	3	5	1
日本樹蛙	5	4	5	3	5	2	1
牛蛙	8	3	4	0	5	3	2
黑眶蟾蜍	6	7	3	4	2	2	1
盤古蟾蜍	7	5	3	4	0	3	3

其集合為 $X = \{\vec{x}_1, \vec{x}_2, ..., \vec{x}_T\}$，則高斯混合模型的概似值(likelihood)可寫為式(5)。

$$p(X \mid \lambda) = \prod_{t=1}^{T} p(\vec{x}_t \mid \lambda) \text{.....(5)}$$

這個式子中的參數集合 λ 是非線性的函數，所以不能直接用 $p(X \mid \lambda)$ 對 λ 微分等於零的方式，來找出 λ 的最佳解。因此，我們必須採取反覆地利用期望值最大化演算法(Expectation- Maximization, EM)來估算最大可能性的高斯混合模型參數，其作法為先找一個初始模型的參數 λ 來估算新的模型參數 $\overline{\lambda}$，使得 $p(X \mid \overline{\lambda}) \geq p(X \mid \lambda)$。這個新的模型參數 $\overline{\lambda}$ 變成初始模型參數 λ，反覆地利用期望值最優化演算法估算模型的參數，直到收斂為止。

而在感測器端的辨識模組，由於感測器的運算處理能力不強，因此在本研究中將利用 SMDF 求得的數個音高值加以統計分組，利用其分佈狀態的差異，來作為感測器端的簡易分類法。

因為一段觀測聲音大約會切割成數十個音框，所以經 SMDF 計算後會得到數十個音高，但若直接數十個音高值來統計，則各種聲音之間的差異性不容易被看出，因此先將數十個音高值進行分組，如圖 5 所示，實驗測試將會把組數控制在十組以下，用意是減少比對時資料量大所造成的運算負荷，同時也能增加各組音高間的差異性。

而表 1 則是五種蛙類與五種非蛙類的叫聲之音高分佈統計表(實驗設定：音框長度 256 點、重疊部分 128 點、向右平移 2 點)。可以從表中看出各種動物叫聲在音高分佈次數的不同，因此可以藉此來分辨出各種聲音，故本研究將用來做感測器端的簡易辨識。

經過許多筆聲音資料的特徵擷取與模組訓練後，會得到數組辨識用的參數，可以用來與

感測器接收到的數組音高做比對。

3.3 感測器端的即時特徵擷取

這階段工作可分為三個步驟，分別是標準化、端點偵測與擷取聲音特徵。感測器收集到聲音後，為了避免因為音量與聲音樣本的音量不同而造成之後辨識的錯誤，必須先進行標準化(normalize)，即將原始訊號做等比例地放大或縮小，使其取樣值都落在同一範圍。標準化的運算式可自行定義，本研究使用式(6)的計算方式。

$$\tilde{S}(n)=\frac{S(n)}{S_{Max}}*10 \quad n=1,2,\ldots,N \ldots\ldots\ldots(6)$$

接下來進行雜訊去除與端點偵測，因為偵測到的聲音多少都會含有背景微小的雜音，為了保留需要分析的聲音片段，需要將雜音部分給去除掉，端點偵測的常用方法可分為時域與頻域，因為時域的作法計算量比較小，比較適合本系統平台，所以本系統採用時域的作法，常用的方法有能量偵測、過零率等。

為了決定觀測聲的起點與終點，必須先計算各個音框的音量，在本研究中計算方式採用運算比較簡單的方法，即每個音框取絕對值的總和，如式(7)；*x(k)*為音框內各點振幅，*M* 為音框長度。

$$E_m=\frac{1}{M}\sum_{k=0}^{M-1}\left|x_m(k)\right| \ldots\ldots\ldots\ldots(7)$$

因此當該音框的音量大於某門檻值，且這樣的音量持續了一段時間，則可以視為有動物聲(蛙類或其他動物)的出現，便可以開始做該段音框的特徵擷取。

確定聲音的起迄點後，即可擷取這段音訊的特徵值，這一部分與 3.1 小節提及的步驟相同，即將這整段聲音切割成一個個小單位的音框，再利用 SMDF 計算各音框的音高。

3.4 特徵值的比對

在擷取完聲音樣本特徵值後，感測器端的初步辨識是採用 3.2 小節提到的音高分佈狀況比較法，當擷取出的音高值集中分佈於某幾個音高群組，或是某個音高群組資料極少，則有可能發生蛙叫聲。

開始辨識時，先讀取出各組蛙叫聲的平均次數。然後將 SMDF 計算出的數組音高，與表 2 的各蛙類音高分佈狀況做比較，看看與哪種蛙類較為接近，運算式如(8)與(9)：

$$R_i=\left|x_i-y_i\right| \quad if\ |xi\text{-}yi|>1,\ Ri=0\ else\ Ri=1 \ldots.(8)$$

$$P=\sum_{i=1}^{N}R_i \quad if\ P<3,\ it\ is\ not\ a\ frog \ldots..(9)$$

x_i 為未知聲音的音高分佈次數，y_i 為某種蛙叫聲的音高分佈次數，由(8)計算出與某種蛙類的組別差距，然後再從(9)去比較與哪一種蛙類的差距中，小於 1 的組別最多，則最有可能是該蛙類，反之若與所有蛙類的差距小於 1 的組數少於 3 組，則有可能是其他聲音，以此作為感測器端的簡易辨識。

經感測器端的簡易辨識後，若結果為蛙類叫聲，便將這組音高值以無線方式傳回到後端伺服器以進行更一步地比對；若在感測器端即辨識為非蛙類叫聲，就會捨棄該組音高值，而不做傳回伺服器的動作。

音高值傳回伺服器後會在這 *M* 個高斯混合模型中找出具有最大機率的聲音模型。經過前述的特徵擷取後，會得到一連串具有 D 維度的測試特徵向量 $\{x_1,x_2,\ldots,x_t\}$，再依照不同的輸入單位長度將所有測試的特徵向量進行部分重疊的片段，每個片段都視為一個獨立的測試句子，其單位長度為 *T* 個特徵向量，如下列式子所示：

$$\overbrace{x_1,x_2,x_3,\ldots,x_T}^{Segment1},x_{T+1},x_{T+2},\ldots,x_t$$

$$x_1,\overbrace{x_2,x_3,\ldots,x_T,x_{T+1}}^{Segment2},x_{T+2},\ldots,x_t$$

在辨識時，系統要考量所有的模型，找出最有可能的聲音模型，即符合最大可能性預估標準的模型。若以數學式來表示，辨識的過程可表示成找尋最有可能的模型 $\hat{S}$ ，如式(8)：

$$\hat{S}=\arg\max_{1\le k\le S}P(\lambda_k\mid X) \ldots\ldots\ldots\ldots(8)$$

其中 $X=\{x_1,x_2,x_3,\ldots,x_T\}$，表示一個分段的輸入聲音向量，輸入長度為*T*個音框，而 $\lambda_k, k=1,2,\ldots,S$ 是所有聲音模型的參數集合。

4. 實驗結果

4.1 實驗環境與參數設定

本研究所使用的聲音檔資訊為：樣本取樣率為 11025Hz 的單聲道 wav 檔；實驗平台為一

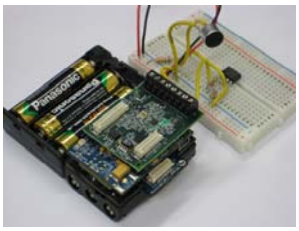

圖6. Crossbow公司生產的imote2

般個人電腦與 Crossbow 公司生產的 imote2 感測器(如圖 6 所示，左側為 imote2 與電池組，右側為加裝聲音感測器的麵包板，因為 imote2 沒有提供聲音感測的功能，所以必須加裝聲音感測器以進行聲音接收)，imote2 採用 32 位元的 XScale 處理器晶片，同時配置了 32Mb DRAM 與 32Mb Flash 的記憶體。

實驗從一個 imote2 感測器收集到環境聲音後開始，接下來在感測器上使用 SMDF 先做聲音特徵值的擷取，再利用音高分佈狀況比較法判斷是否為蛙叫聲。如果判斷結果是蛙叫聲，便將辨識結果與音訊特徵值傳回到有連接另一個 imote2 感測器的伺服器使用 GMM 做進一步地辨識並記錄之；如果在感測器端判斷結果為非蛙叫聲，便在感測器端把資料刪除。實驗項目除了分別針對「蛙叫聲與非蛙叫聲的辨識」、「各種蛙叫聲間的辨識」進行實驗外，也會對「感測器接收到聲音後即將資料傳回伺服器」與「感測器接收到聲音後，會先在感測器端做辨識後再傳回伺服器」兩方案進行感測器耗電量的比較。

表2. 各種蛙類音高分佈狀況表

pitch / sound	~60	61~80	81~100	101~130	131~170	171~210	211~
小雨蛙	7.1	3.7	2.9	4.8	3	5.8	1.2
日本樹蛙	5.2	4	5.5	3.1	5.6	2.7	1
牛蛙	8.6	3.1	4.2	0.7	5.7	3	2.1
黑眶蟾蜍	6.3	7.3	3.3	4.2	2.3	2.5	1.4
盤古蟾蜍	7.5	5.1	3.4	4	0.4	3.8	3
中國樹蟾	6.8	8.2	2.6	1.8	2.1	3.3	2
古式赤蛙	6.4	4.9	4.1	4.7	2.2	4.2	1.4
台北樹蛙	8.1	7.1	3.5	1.2	1.9	4	1.3
台北赤蛙	5.2	8.7	6.6	1.2	3	1.9	1.6
史丹吉氏小雨蛙	4	6	5.2	2	4.1	2.9	2.8
巴氏小雨蛙	5.6	2.7	2	3.6	5.3	5.8	3.7
橙腹樹蛙	7.4	5.2	6.3	4.9	2	0.1	1
澤蛙	7.1	3.9	6	3.7	1.4	2.1	3.6
白頷樹蛙	5.7	4.6	4.5	1.4	6.8	3	2
翡翠樹蛙	6.1	4.3	6	2.1	3	3.4	1
腹斑蛙	7.8	7.9	3.1	3.8	3.8	2.5	0.2
艾氏樹蛙	4.2	3	6.8	5	1.6	3.2	3
莫氏樹蛙	8.7	4.1	5.3	2.5	2.4	2	2.1
諸羅樹蛙	7.3	3.9	3.4	4.9	4	2.9	2
豎琴蛙	8	5.7	5	0.3	2.9	4	1.6

表 3. 實驗一：蛙類與非蛙類的辨識結果

偵測聲響	偵測數	辨識正確數	正確率(%)
非蛙類	40	29	73
蛙類	40	31	78

表 4. 實驗二：蛙類辨識結果

科別	種別	偵測數	辨識正確數	正確率(%)
蟾蜍科	黑眶蟾蜍	12	7	58
	盤古蟾蜍	13	8	62
樹蟾科	中國樹蟾	13	7	54
狹口蛙科	巴氏小雨蛙	13	8	62
	史丹吉氏小雨蛙	14	7	50
	小雨蛙	13	7	54
赤蛙科	腹斑蛙	12	9	75
	牛蛙	13	7	54
	古式赤蛙	13	8	62
	澤蛙	13	9	69
	豎琴蛙	12	8	67
	台北赤蛙	12	7	58
樹蛙科	日本樹蛙	14	7	50
	艾氏樹蛙	14	8	57
	莫氏樹蛙	12	7	58
	諸羅樹蛙	14	6	43
	白頷樹蛙	13	7	54
	翡翠樹蛙	12	7	58
	橙腹樹蛙	12	6	50
	台北樹蛙	13	8	62

在正式實驗前，先找出較適合用來做辨識實驗的參數設定，如音框大小等。所以先分別針對音框長度、音框重疊部分與 SMDF 平移點數等參數做初步實驗：候選的音框大小有 128ms、256ms 與 512ms；音框重疊部分則是 64ms、128ms 與 256ms；SMDF 平移點數則有 2 點與 4 點兩種。

結果初步實驗結果以音框大小256ms，重疊部分128ms，SMDF往右平移2點可以得到較大的分佈差異性，實驗結果如表2所示，故以此為本實驗的初始設定參數。其餘實驗設定為端點偵測以80分貝為門檻，並假定發生觀測聲時會持續0.3秒。

4.2 音訊辨識率實驗

第一部份的實驗結果如表 3，是針對「蛙類」與「非蛙類」的辨識，從結果可以看出在蛙類與非蛙類各 40 個樣本間的辨識結果皆在70%以上，能夠有效率地做兩者間的區別。

而第二部分的實驗結果如表 4，是細分成各種不同蛙類間的聲音辨識，這部分的辨識正確率已經下降到平均約 60%左右，與第一部份的實驗比較起來，效果不甚突出。

4.3 耗電量模擬實驗

最後是驗證在感測器端做初步辨識與否

表 5. 實驗三：感測器耗電量的比較

	接收的聲音數	蛙叫聲數目	剩餘電力
方案 1	50	30	88%
方案 2	50	30	76%

跟感測器耗電量的關係。這部分的實驗設定為：感測器組分別在兩個小時內進行聲音感測與運算，接收的聲音檔同為 50 個，其中有 30 個是蛙叫聲，使用的電池為 3 顆全新的 1.5V4 號鹼性電池，兩小時之後，再來測量電池組剩下的電壓。方案 1 為感測器接收到聲音後，會先在感測器端做辨識後再傳回伺服器；方案 2 是感測器接收到聲音後即將資料傳回伺服器。實驗結果如表 5。很明顯地，在感測器端先行辨識後再決定要不要傳回伺服器端的方法，會比接收到聲音後全部傳回到伺服器的耗電量減少許多，因此先在感測器端做辨識，可以有效地增加感測器的電池使用壽命。

5. 結論

過去生物的棲息地多可以從歷史發展演進去推測，但現代人類對環境的影響，使得生物為生存而遷徙至其他地方，也造成生物學家觀測研究生物活動的不方便。本研究基於不同的生物之間有著其獨特叫聲的生物特性，並結合音訊處理技術與無線感測器網路，可以讓生物學家透過這套系統去偵測生物出沒，不必自己費心於生物棲息地的尋找，而嵌在無線感測器上的自動化動物聲辨識模組，也可以減少過去直接將聲音傳回伺服器再做處理的電力消耗問題，增加感測器的觀測時間，免除人工辨識所花費的時間與精力，對生物學家的研究效率會有不小的助益。

本研究受限於感測器的處理能力，為了讓音訊辨識模組能在無線感測器上運作，僅利用簡單的音高擷取演算法 SMDF 來做特徵值的擷取，並利用簡單的統計方法-音高分佈狀況來分類，雖然在蛙類與非蛙類的辨識率不差，但不同蛙類間的辨識仍有改進空間，未來隨著無線感測器處理功能的進步，可以在感測器上做更精確地比對，以提升辨識的正確率。

參考文獻

[1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A Survey on Sensor Networks. *IEEE Communications Magazine, 40*(8), 102-114.

[2] Andersson, T. (2004). Master's Thesis : Audio classification and content description. *Department of Computer Science and Electrical Engineering Division of Signal Processing*.

[3] Cheveigne, A. D., & Kawahara, H. (2002). Yin, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America, 111*(4), 1917-1930.

[4] Doddington, G. R. (1985). Speaker recognition - identifying people by their voices. *Proceedings of IEEE, 73*(11), 1651-1664.

[5] Furui, S. (1994). An overview of speaker recognition technology. *Proceedings of the ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, 1-9.

[6] Gazor, S., and Zhang, W. (2003). Speech probability distribution. *IEEE Signal Processing Letters, 10*(7), 204–207.

[7] Hac, A. (2003). Wireless sensor network designs. *John Wiley & Sons, Inc*.

[8] Hill, J., & Culler, D. (2002). A wireless embedded sensor architecture for system-level optimization. *In UC Berkeley Technical Report*.

[9] Juang, B. H. (1998). The past, present, and future of speech processing. *IEEE Signal Processing Magazine,* 24-48.

[10] Jyh-Shing Roger Jang. Audio Signal Processing and Recognition. Retrived December 01, 2008, from http://www.cs.nthu.edu.tw/~jang

[11] Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D., & Anderson, J. (2002). Wireless Sensor Networks for Habitat Monitoring. *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*.

[12] Pottie, G. J. (1998). Wireless sensor networks. *IEEE Information Theory Workshop*, 139-140.

[13] Pottie, G., & Kaiser, W. (2000). Wireless integrated network sensors. *Communications of the ACM, 43*(5), 51-58.

[14] Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *IEEE Transactions on ASSP, 77*(2), 257-286.

[15] Reynolds, D. A., Quatieri, T. F., & Dunn, R. B. (2000). Speaker Verification Using Adapted Gaussian Mixture Models. *Digital*

Signal Processing, 10, 19-41.

[16] Reynolds, D. A., & Rose, R. C. (1995). Robust text-independent speaker identification using Gaussian mixture models. *IEEE Transactions on Speech and Audio Processing, 3*(1), 72-83.

[17] Ruiz, L. B., Nogueira, J. M., & Loureiro, A. (2003). MANNA: A Management Architecture for Wireless Sensor Networks. *IEEE Communication Magazine, 41*(2), 116-125.

[18] Scheirer, E., and Slaney, M. (1997). Construction and evaluation of a robust multifeature speech/music discriminator. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*, 2, 1331-1334.

[19] Soong, F. K., Rosenberg, A. E., Rabiner, L. R., & Juang, B. H. (1985). A vector quantization approach to speaker recognition. *Proceeding of IEEE ICASSP,* 387-390.

[20] Tilak, S., Abu-Ghazaleh, N.B., & Heinzelman, W. (2002). A taxonomy of wireless micro-sensor network models. *ACM SIGMOBILE Mobile Computing and Communications Review*, *6*(2), 28-36.

[21] Tzanetakis, G., and Cook. P. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing, 10*(5), 293-302

[22] Wang, A., & Chandrakasan, A. (2002). Energy-Efficient DSPs for Wireless Sensor Networks. *IEEE Signal Processing Magazine, 19*(4), 68-78.

[23] Wang, X., and Qi, H. (2002). Acoustic target classification using distributed sensor arrays. *IEEE International Conference on Acoustics Speech and Signal Processing, 4*, 4186.

[24] Wu, B. F., & Wang, K. C. (2005). A Robust Endpoint Detection Algorithm Based on the Adaptive Band-Partitioning Spectral Entropy in Adverse Environments. *IEEE Transactions on Speech and Audio Processing. 13*(5), 762-775.

[25] Wu, C. H., & Chen, J. H. (1997). Speech activated telephony email reader (SATER) based on speaker verification and text-to-speech conversion. *IEEE Transactions On Consumer Electronics, 43*(3), 707-716.

[26] Xing, B. et al. (2002). Short-time Gaussianization for robust speaker verification. *Proceedings of ICASSP, 1*, 681-684.

[27] Yao, Y., & Gehrke, J. (2003). Query processing for sensor networks. *Proceedings of the CIDR*, 233-244.