

線上魚類導覽系統

王建仁
崑山科技大學
資管系助理教授
cjlw@mail.ksu.edu.
tw

蔡哲民
崑山科技大學
資管系助理教授
tjm@mail.ksu.edu.
tw

陳世傑
崑山科技大學
資管系研究生
ilove581@yahoo.co
m.tw

周家弘
崑山科技大學
資管系研究生
samejack@msn.co
m

摘要

現今資訊科技和網際網路蓬勃發展，線上導覽也因此成為盛行的魚類介紹方式。目前線上導覽系統大部分是以文字搭配圖片或預先錄製的影片進行介紹，但影片常因容量的限制，造成播放內容貧乏，進而導致使用者的學習興趣低落。因此本研究製作了一套即時拍攝魚類的線上魚類導覽系統，本系統可將網路攝影機擷取之影像，透過物件追蹤影像處理技術取得畫面中所有魚的區域位置，再經由圖形辨識影像處理技術判別該魚為何種魚類。而系統使用 Java Applet 程式語言撰寫完成，使用者透過網頁瀏覽器連接即可觀賞，達到能即時觀察，並獲得所感興趣之魚類的相關資訊，讓使用者容易瞭解且提高學習興趣。

關鍵詞：即時線上導覽、物件追蹤、物件切割、圖形辨識。

Abstract

The contents of most existing on-line fish navigation systems were organized with texts, static pictures and pre-recorded videos. In this paper, we described an on-line web-based fish navigation system. Our system can track the fishes based on object tracking techniques in the real-time video caught from IP network camera. The interested fish can be chosen and the system would recognize the fish and retrieve the relevant information from database.

This system was implemented with Java Applet. Interested users can view the video, choose the interested fish and read the description through browsers easily.

Keywords: On-line Web-Based Navigation, Object Tracking, Object Segmenting, Pattern Recognition.

1. 前言

由於網際網路的普及化，大眾普遍使用網路進行資訊的查詢與收集，像是新聞、多媒體、產品資訊等。另外，有關生物的相關資訊也可以在網路上尋找，例如生物的壽命、體長、適合居住環境，而此類的網站或系統大部分是以文字方式，搭配一二張圖片進行描述[12]。對於使用者而言，由於以文

字方式無法充分表達生物成長過程、行為等特徵，導致使用者不容易深入了解這部份的資訊；而目前提供即時拍攝魚類影像的導覽平台較少，大多是將影片預先錄製好播放[2]。使用預錄影像常會因為容量大小限制影片解析度和長度，造成播放內容的貧乏，使用者的學習興趣因而低落。要讓使用者容易瞭解並有較高的學習興趣，以即時互動的方式觀察生物行為活動會有較好的效果[5]。

現今電腦數位影像處理技術快速發展，例如物件切割、物件追蹤、圖形辨識等技術，如今已被廣泛應用在各種環境之中，像是人員與運動員之追蹤[9]、生物行為量化[19]、影像辨識[1]與視訊壓縮[14]等。以上技術都能幫助使用者提升作業效率，但通常這些技術都是假設應用的環境是在固定的區域，比較少用網際網路做資料的傳遞，或跨網路做複雜的影像處理運算。因此，即便文獻中有許多魚類的追蹤辨識研究，但缺乏跨網際網路的即時魚類追蹤導覽系統。事實上隨著網際網路的速度愈來愈快，將影像追蹤與影像辨識技術應用在跨網際網路的服務上，也不再是遙不可及的夢想。

蔡哲民等人曾提出一套即時拍攝植物影像的教學平台[7]，使用者透過網頁瀏覽器連接便能即時觀察植物，且可依據使用者點選感興趣的組織部分給予知識。但此系統只能應用於移動速度較慢、活動範圍較小之生物，對於魚類、昆蟲等移動速度較快，且活動範圍也較廣的觀察對象，要利用該論文所提的動態輪廓模型(Active Contour Models)演算法[13]成功進行追蹤較為困難。

有鑑於上述的情形，本研究提出一套線上魚類導覽系統。使用一台個人電腦作為伺服器，並將一個網路攝影機(IP Network Camera)放置於一般家庭用水族箱前，將畫面發送給使用者端的系統程式。之後運用背景相減(Background Subtraction)[16]與區域成長演算法(Region Growing)[17]等影像處理技術將畫面中的魚定位出來；如有發生魚互相重疊的情況，利用魚的輪廓(Contour)及移動向量(Motion Vector)等資訊透過動態輪廓模型進行區分；最後使用顏色直方圖(Color Histogram)[15]判別該魚為何種魚類。本研究所使用之方法對於移動速度較快的魚類也能即時追蹤，而系統是以 Java Applet 撰寫而成，使用者只需透過網頁瀏覽器連線，便能即時觀察魚類的活動，瞭解魚類的活動特性。當使用者從影像畫面上點選感興趣的魚時，系

統便會顯示該魚類的相關資訊。這樣的系統不僅可以應用於線上教學，使學習過程更生動有趣，提升學習之成效；公私立之水族館亦可利用此系統，提供一般使用者隨時上網觀察魚類的活動情形與畫面，充分瞭解魚類的生活型態，甚至進行線上交易。

2. 系統架構

由於要將所拍攝到的畫面即時地發送，我們利用網路攝影機發送影像，並將其架設於一般家庭用之水族箱前。而為了減少Server的負擔、提高穩定性，我們使用Java Applet撰寫物件追蹤及圖形辨識等運算複雜度高的影像處理程式。Server端則負責提供魚類切割用的背景影像與魚類知識等相關資訊。使用者透過網頁瀏覽器連線至網站，系統即自動對攝影機所拍攝畫面中的魚類進行追蹤，當使用者於畫面上點選感興趣的魚時，便會從魚類資料庫中抓取相關資訊顯示給使用者。本系統架構如圖 1。

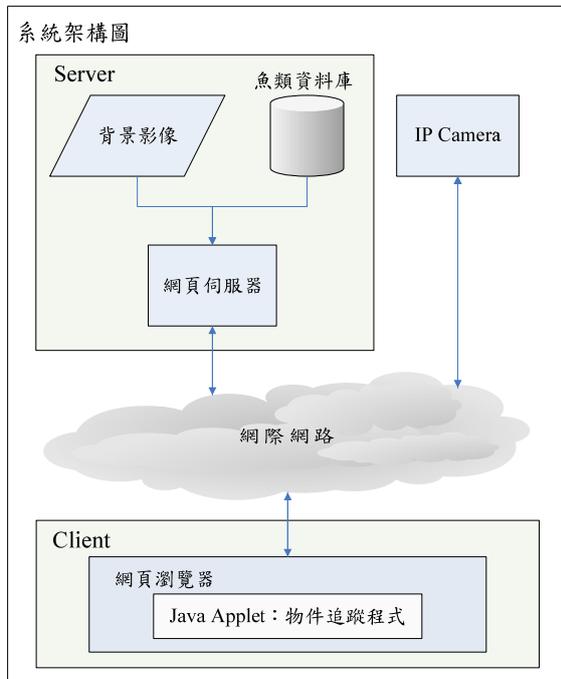


圖 1 系統架構圖

3. 方法描述

本系統之物件追蹤流程部分包含了物件搜尋程序、物件切割程序、重疊物件區分程序、物件辨識程序四個主要程序。在影像來源方面，將網路攝影機所拍攝的畫面傳送給Client端的Java Applet 程式，當影像資訊傳送成功後，系統便會即時做物件追蹤的動作，第一步為透過物件搜尋程序尋找物件可能存在之座標位置；接著以背景影像與目前影像透過背景相減與區域成長演算法將前景物件切割出來，並藉由鏈碼(Chain code)演算法[11]取得物件

的邊緣輪廓資訊。在背景影像取得方面，當使用者連上網頁時，會一併將預先儲存於Server端之背景影像傳送給Client端的Java Applet程式。由於背景影像為固定的畫面，我們事先將背景影像儲存於Server端以減少Client端重新建立背景影像的時間，進而縮短Applet載入到系統開始正常追蹤的時間。在物件切割程序後，如發生物件互相重疊的情況，利用物件移動向量與動態輪廓模型來區分不同的物件。最後以物件的顏色特徵利用顏色直方圖進行辨識，將資料庫中該魚類的資訊顯示給使用者。物件追蹤流程如圖 2 所示。

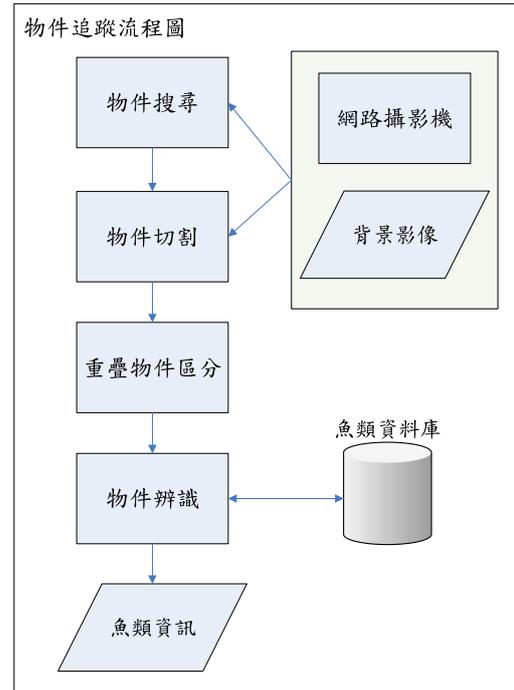


圖 2 物件追蹤流程圖

3.1 物件搜尋

傳統使用背景相減方法進行物件切割的方式，皆為對整張影像進行處理，所獲得之切割結果除了主體物件，可能也包含了許多雜訊。其方法需要進一步的後處理動作，例如使用低通濾波器(Low-pass Filter)將雜訊濾除，目前常使用的雜訊濾波器包含中值濾波器(Median Filter)、均值濾波器(Mean Filter)、形態學濾波器(Morphology Filter)、高斯濾波器(Gaussian Filter)等[3]。一般透過濾波器的處理之後，會失去原始物件資訊的準確性，且運算較複雜而導致處理速度較慢。我們為了得到更準確的物件資訊及更低的時間複雜度，先搜尋物件可能出現之座標位置，之後僅對這些座標的鄰近區域進行處理，且不需再做後處理的動作，相較於傳統背景相減演算法對整張影像做處理的方式，能降低許多運算複雜度。

以下為本文之物件搜尋演算法，由公式(1)判斷該座標位置是否為物件出現之位置。

$$C(x,y) = \sum_{i=-n}^n d(x+i,y) + \sum_{i=-n}^n d(x,y+i)$$

$$S(C) = \begin{cases} 1, & C = 4n + 2 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

當 $S(C)$ 為 "1" 時，代表座標點 (x,y) 為物件出現的座標位置。其中 n 為一常數值，可依據物件大小調整 n 值，當 n 值愈大可以濾除愈大的雜訊點。 $d(x,y)$ 函數為目前影像 $f(x,y)$ 與背景影像 $b(x,y)$ 之差異，並透過門檻值 T 進行二值化之結果，如公式(2)所示。

$$d(x,y) = \begin{cases} 1, & |f(x,y) - b(x,y)| > T \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

上述公式中， $|f(x,y) - b(x,y)|$ 代表目前影像與背景影像之差異值，可由下列公式(3)做計算。

$$|f(x,y) - b(x,y)| = |R_{f(x,y)} - R_{b(x,y)}| + |G_{f(x,y)} - G_{b(x,y)}| + |B_{f(x,y)} - B_{b(x,y)}| \quad (3)$$

其中 $R_{f(x,y)}$ 、 $G_{f(x,y)}$ 及 $B_{f(x,y)}$ 分別為目前影像之 RGB 像素值，而 $R_{b(x,y)}$ 、 $G_{b(x,y)}$ 與 $B_{b(x,y)}$ 則為背景影像 RGB 像素值，經由此公式即可獲得目前影像與背景影像之像素差異值。

上述之物件搜尋法，在空間概念上為透過十字線的方式尋找物件可能出現的座標位置。公式中的 n 亦可表示為十字線的大小，如圖 3 中紅色十字線所示。我們對圖片由左至右，由上至下以公式進行運算，將可獲得許多物件座標點。

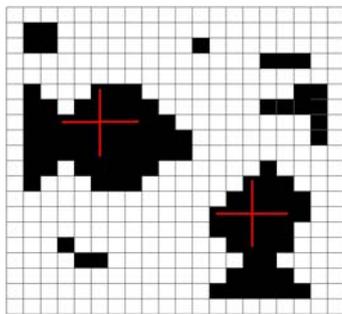


圖 3 物件搜尋示意圖

3.2 物件切割

經由物件搜尋程序後，將所獲得之物件座標點進行物件切割程序，分別對所有的座標點以背景相減結合區域成長演算法進行物件切割的動作。區域成長早期由 R. Adams 在 1994 年提出，在空間概念上以聯通因素為基礎，其為針對欲處理的像素點判斷上下左右四個鄰近像素點是否為物件，判斷方式由上一小節所提公式(1)完成，當四周任一像素判斷為物件時，便逐一繼續對其四周鄰近像素作判斷，以遞迴之概念完成物件切割。

在經由上一步驟之後可獲得物件之二值化影像，由於在進行拍攝時可能會受到光線、雜訊的影

響，或是物件本身的顏色與背景顏色相似，造成切割所獲得的物件有空洞的情形，如圖 4 中(b)圖所示。以往大部分的研究常使用形態學濾波器中所組成的閉合運算(Closing Operator)來填補空洞，此法必須對整張影像做運算，速度較慢且容易破壞原始物件之邊緣，並不適合用在即時的系統中。為了有較低的運算複雜度，本文所使用的方式是採用鏈碼演算法的概念。其方法為順時針沿著物體的邊界進行標記，將輪廓依序標記出來並且記下輪廓座標位置，找出物件輪廓後將該封閉區域填滿以修復空洞的情形。此一動作主要是為了提高重疊物件區分及物件辨識時的正確率。經由此步驟處理過後之效果如圖 4 之(c)圖。

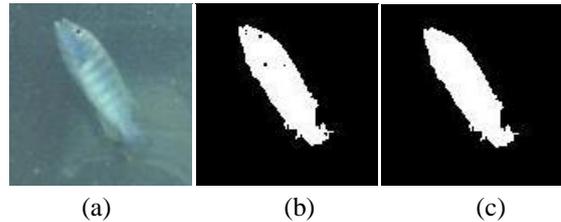


圖 4 (a)原始影像；(b)切出包含空洞區域之物件；(c)物件切割程序後之物件

3.3 重疊物件區分

處理物件重疊問題為實行多物件追蹤時容易遭遇的困難，如何正確地區分畫面中各個不同的物件便成為目前重要之研究。本文利用移動向量作為指標來比對連續畫面中的物件，並且配合物件面積資訊來偵測物件有無重疊的情況，在處理物件重疊時採用動態輪廓模型來劃分出各個物件的區域位置。下圖 5 為重疊物件區分之流程圖。

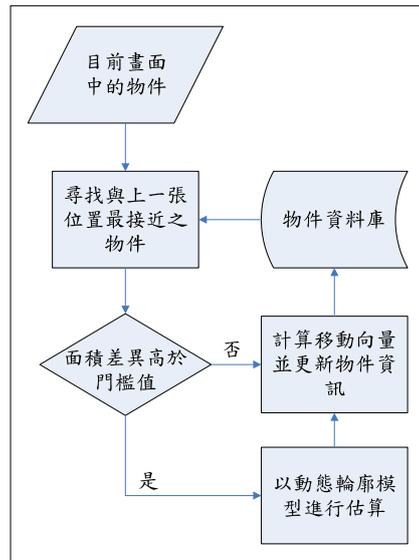


圖 5 重疊物件區分流程圖

首先將目前畫面中的各個物件，分別與前一張

畫面所儲存的所有物件逐一進行比對，計算出移動向量距離最為接近的物件，距離的計算方式為使用兩個物件的重心之歐基里德距離 (Euclidean Distance) 求得。重心座標的計算方式以空間概念為基礎，藉由統計物件座標計算求得，可以透過下列公式(4)求得。

$$M_{p,q} = \sum_x \sum_y x^p y^q f(x, y)$$

$$g(x, y) = \left(\frac{M_{1,0}}{M_{0,0}}, \frac{M_{0,1}}{M_{0,0}} \right) \quad (4)$$

我們以物件重心作為比對的指標，當從物件資料庫找出相對應的物件時，緊接著計算兩物件面積之比例。當物件面積比例大於門檻值 T_a 時，表示物件在目前畫面中發生重疊的情況，反之則為分離情況，判斷物件重疊之方式由以下公式(5)所示。

$$\begin{cases} \text{overlap} & , \frac{O_i - O_{db}}{O_{db}} > T_a \\ \text{non-overlap} & , \text{otherwise} \end{cases} \quad (5)$$

其中 O_i 為目前畫面中的物件， O_{db} 為從物件資料庫中比對出距離最近的物件，我們將門檻值設定為 0.3 會有較好的結果。當物件面積比例低於門檻值時，表示此物件在目前畫面中並無與其它物件重疊，則將物件輪廓及計算移動向量後之物件重心更新至物件資料庫中。重心移動向量可由公式(6)求得。公式(6)中 x_i 與 y_i 為目前畫面物件之物件重心座標， x_{i-1} 與 y_{i-1} 則為前一張畫面物件之物件重心座標。

$$(x_p, y_p) = (x_i - x_{i-1}, y_i - y_{i-1}) \quad (6)$$

反之，當發生物件重疊情況時，我們將物件資料庫中之物件輪廓進行移動向量之計算，使其輪廓更接近目前畫面中的物件，輪廓移動公式如公式(7)所示。

$$f(c) = f(p_n(x, y))$$

$$= (x + k(x_i - x_{i-1}), y + k(y_i - y_{i-1}))$$

$$c \in \{p_1(x, y), p_2(x, y), \dots, p_n(x, y)\} \quad (7)$$

公式(7)中 k 為一常數值，在經過多次實驗後，將 k 設定為 0.33 能夠獲得較好的準確性。而 c 表示為物件本身的輪廓，其包含了物件輪廓所有的座標點 $p_i(x, y)$ 。

在經過上述所提之方式移動物件輪廓後，對於目前畫面中實際的物件輪廓還是會有許多誤差，於是我們使用動態輪廓模型對物件輪廓進行收斂的動作，使得物件輪廓能更接近實際的物件邊緣。動態輪廓模型最早在 1988 年由 M. Kass 所提出，此演算法係為一種半自動的輪廓收斂方式，其必須事先由人工手動定義初始輪廓才能進行收斂。目前常被應用在各種醫學影像及物件切割的領域之中[10]。而動態輪廓模型在輪廓的收斂上，採用最小能量

(Energy Minimization) 的概念將輪廓上的控制點漸漸地移動至影像中實際的物件特徵位置。我們利用資料庫中所比對出的物件輪廓經過上述移動向量計算後之輪廓作為動態輪廓模型所需的初始輪廓。動態輪廓模型如下公式(8)所示：

$$E_{total} = \int_0^{-1} E_{snake}(v(s)) ds$$

$$= \int_0^1 (E_{int}(v(s)) + E_{ext}(v(s))) ds \quad (8)$$

其中 E_{total} 為整個輪廓之總能量， E_{snake} 為輪廓上某個控制點之能量，而總能量又包含了內部能量 $E_{int}(v(s))$ 與外部能量 $E_{ext}(v(s))$ ，其中 $v(s) = (x(s), y(s))$ 所指為控制點在影像中之資訊，可以由像素梯度值或影像強度值等資訊所組成。內部能量的定義為初始輪廓上各單一控制點所運算出來的能量，主要是為了維持輪廓之線性與平滑度，內部能量如下公式(9)所示：

$$E_{int} = (\alpha |V'(s)| + \beta |V''(s)|) / 2 \quad (9)$$

其中 α 與 β 為第一項與第二項的權重。 $|V'(s)|$ 與 $|V''(s)|$ 為控制點 V_i 的一階微分和二階微分，亦能以下公式(10)表示：

$$|V'(s)| = \left| \frac{dv_i}{ds} \right| = |v_i - v_{i-1}|$$

$$|V''(s)| = \left| \frac{d^2 v_i}{ds^2} \right| = |v_{i-1} - 2v_i + v_{i+1}| \quad (10)$$

$|V'(s)|$ 所表示的涵義為輪廓中相鄰各控制點之間的連續性 (Continuity)， $|V''(s)|$ 則是指輪廓中相鄰各控制點的平滑性 (Smoothness)。

而外部能量的定義為影像資訊所計算出來的能量，主要目的為促使輪廓收斂至物件實際的邊緣。外部能量的取得大多數是以影像的邊緣強度表示，可由下列公式計算求得。

$$E_{ext} = -\gamma |G(x, y)| \quad (11)$$

其中 γ 為外部能量的權重參數， $|G(x, y)|$ 為影像的梯度值，本文使用 Sobel 邊緣偵測來取得梯度值。Sobel 包含兩組 3*3 的矩陣，分別可得到橫向及縱向的梯度值。如果以 I 代表原始影像， G_x 及 G_y 分別代表經橫向及縱向邊緣檢測的梯度值，其公式如公式(12)：

$$G_x(x, y) = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I(x, y) \quad \text{and}$$

$$G_y(x, y) = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * I(x, y) \quad (12)$$

影像的每一個像素的橫向及縱向梯度值可由公式(13)結合，計算像素的總梯度值大小。

$$G(x, y) = \sqrt{G_x^2 + G_y^2} \quad (13)$$

透過上述之公式，在輪廓上各個控制點周圍 $m*m$ 的區域中，計算出最小能量出現的座標位置，進而將控制點移動至該座標位置，使輪廓達到逼近物件邊緣的效果。

下圖 6 中，圖(a)為兩物件沒有重疊時，物件切割程序後所獲得之物件區域。當兩物件互相靠近並發生重疊的情況，經由重疊物件區分程序後的結果如圖(b)。

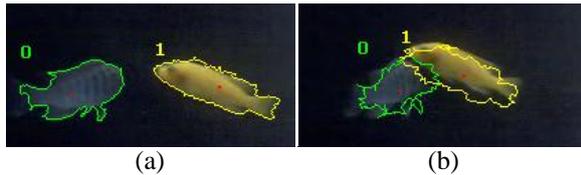


圖 6 (a)物件非重疊時物件切割情形；
(b)物件重疊時物件區分情形

3.4 物件辨識

透過以上程序後，可以得到各個物件的區域座標，我們擷取該區域範圍的顏色屬性，作為代表此物件的特徵值。由於 RGB 顏色空間對於光線的影響過於敏感，為了減少光線對辨識時的影響，我們將顏色空間轉換成 $YCbCr$ 顏色空間，其轉換公式如下公式(14)：

$$\begin{aligned} Y &= 0.2989 * R + 0.5866 * G + 0.1145 * B \\ C_b &= -0.1687 * R - 0.3312 * G + 0.5000 * B \\ C_r &= 0.5000 * R - 0.4183 * G - 0.0816 * B \end{aligned} \quad (14)$$

Y 值所代表的意義為亮度值，為了避免光線對辨識時造成影響，我們只採用 C_b 值及 C_r 值進行比對。

比對方式我們使用顏色直方圖進行，其第一步為統計每個像素點之 C_b 值及 C_r 值出現的次數，依最後統計的個數將可得到 C_b 值及 C_r 值的直方圖，接著計算 C_b 與 C_r 的加權平均數，再與魚類資料庫中的各個魚類之 C_b 與 C_r 的加權平均數兩兩相比，其差異最小之魚類即為此魚類物種。差異值之計算方式如下公式(15)所示。

$$Dif_F = \left| \frac{\sum_{k \in K} \#(H_O^b(k)) \times k}{N_O} - \frac{\sum_{k \in K} \#(H_F^b(k)) \times k}{N_F} \right| + \left| \frac{\sum_{k \in K} \#(H_O^r(k)) \times k}{N_O} - \frac{\sum_{k \in K} \#(H_F^r(k)) \times k}{N_F} \right| \quad (15)$$

其中 Dif_F 為此物件與魚類 F 的差異值， N_O 與 N_F 分別為物件與魚類 F 的像素個數。 $\#H_O^b(k)$ 與 $\#H_F^b(k)$ 分別為物件及魚類 F 的 C_b 直方圖在 k 值的個數； $\#H_O^r(k)$ 與 $\#H_F^r(k)$ 則分別為物件及魚類 F 的

C_r 直方圖在 k 值的個數。 k 的範圍介於 C_b 與 C_r 的最小值和最大值之間， C_b 與 C_r 的範圍都介於 16 至 240。

下圖 7 為使用者透過網頁瀏覽器連接時的畫面。頁面中包含了兩個方框，左方框內呈現網路攝影機目前所擷取到的畫面，而右方框則是註明了相關資訊。當使用者從左方框的影像畫面中點選了某條魚，右方框便會顯示該魚的相關資訊。



圖 7 上圖為連接時的畫面；
下圖為點選魚之後的畫面

4. 實驗結果與討論

本實驗中所使用的電腦設備皆為 CPU P-4 3.0Ghz、1G Ram，攝影機使用 AXIS 205 Network Camera，以解析度為 640*480、每秒 30 張之畫面進行測試。實驗中以長約 10 公分至 15 公分、寬約 2 公分至 8 公分之 4 種魚類，總共 6 隻魚進行測試，將其放置於一般家庭用之水族箱中，並且觀察魚在非重疊時與重疊時辨識的正確次數，以此測試系統的可行性。實驗中我們在不同情形分別辨識 100 次，統計全部 200 次的辨識結果，結果如下表 1 所示。辨識 200 次的結果中，有 171 次辨識正確，錯誤率 14.5%，每秒畫面處理速度(FPS)平均約 12 張。

表 1 實驗結果

有無重疊	辨識正確次數	錯誤率
無重疊	96	4%
有重疊	75	25%
總計	171	14.5%

由實驗結果得知，魚在沒有互相重疊的情況錯誤率只有 4%，表示本研究所使用之物件搜尋、物件切割與物件辨識的方法能正確地將魚定位出來且成功辨識。而在魚互相重疊的情況錯誤率有 25%，代表在重疊物件區分程序時用來區分物件的資訊可能不足，本研究只使用了移動向量與物件輪廓兩種資訊來區分物件。在過去的研究中曾利用物件的顏色進行區分的動作[4]，而此類方法必須在物件顏色差異甚大的前提下才能正確地區分出來。而本研究則是希望當物件顏色相似時也能區分出物件，因此沒有採用顏色資訊作為重疊物件區分的指標。

5. 結論

本文主要是提出一套線上的魚類導覽系統。系統是由 Java Applet 撰寫完成，使用網路攝影機作為影像的攝取裝置，使用者只需透過網頁瀏覽器連接，就能即時針對魚類的生態進行觀測。且結合物件追蹤及圖形辨識等影像處理方法，當使用者在影像上選取感興趣的魚時，系統便會自動顯示該魚類的相關資訊。此一系統將可幫助使用者觀察、瞭解魚類生活型態等，期望對於水族館導覽自動化及家庭水族箱管理會有很大的幫助。

在我們統計完魚互相重疊與非重疊時的辨識正確次數，魚在沒有重疊的情形下辨識錯誤的比例低於 5%，顯示出本研究所使用之物件搜尋、物件切割與物件辨識的方法能正確地將魚定位出來且成功辨識該魚為何種魚類；而在魚發生重疊的情形時，錯誤比例偏高，表示在重疊物件區分時所使用的資訊稍嫌不足。未來可將顏色特徵一併加入考量，當物件顏色相似時以物件輪廓進行重疊物件區分，反之則以物件顏色作為區分的指標。而畫面處理速度每秒約 12 張，當魚移動速度太快時將會造成追蹤錯誤，且影響使用者觀賞畫面的流暢度。如何降低演算法的運算複雜度是我們未來研究的主要目標。另外，如果考慮到 Client 端的電腦效能可能較低，如何將物件追蹤及圖形辨識等運算過程的部份運算改以 Server 端進行處理，以期進一步提昇系統效能，也是未來可以考慮的改進方案。

參考文獻

- [1] 李嘉紘、蔡哲民、王建仁、蔡文吉、吳明昆，**基於貝類正面特徵影像辨識檢索技術**，2007 資訊科技國際研討會。
- [2] 中央研究院生物多樣性研究中心，台灣魚類資料庫，2007 年 8 月 12 日取自 <http://fishdb.sinica.edu.tw/version.htm>
- [3] 徐曉珮 (譯) (2005)。A. McAndrew 著。**數位影像處理**。臺灣：新加坡商湯姆生亞洲私人有限公司台灣分公司。
- [4] 郭智誠，**智慧型水族館魚類導覽系統**。2006，崑山科技大學碩士論文。
- [5] 劉彥甫，**互動式網頁教學對運動技能學習效果之研究—以籃球上籃為例**。2002，國立台東師範學院碩士論文。
- [6] 蔡哲民、周家弘，**結合動態輪廓模型與移動向量之多物件即時追蹤系統**，2006 管理暨實務研討會。
- [7] 蔡哲民、傅耀賢、周家弘、賴峻廷，**具標示與追蹤能力之植物影像教學平台**，2006 台灣國際網路研討會。
- [8] B. Heisele, U. KreBel and W. Ritter, "Tracking Non-Rigid, Moving Objects Based on Color Cluster Flow," *Computer Vision and Pattern Recognition*, pp. 257-260, 1997.
- [9] C. J. Needham and R. D. Boyle, "Tracking multiple sports players through occlusion, congestion and scale," *Proceedings of British Machine Vision Conference (BMVA'01)*, Vol. 1, pp. 93-102, 2001.
- [10] F. Y. Shih, K. Zhang, "Locating object contours in complex background using improved snakes," *Computer Vision and Image Understanding*, pp. 93-98, 2007.
- [11] H. Freeman, "Boundary encoding and processing," *Picture Processing and Psychopictorics*, pp. 241-266, 1970.
- [12] J. T. Williams, "Division of Fishes," Retrieved August 12, 2007, from <http://www.nmnh.si.edu/vert/fishes/>
- [13] M. Kass, A. Witkin, D. Terzopoulos, "Snakes: Active Contour Models," *International Journal of Computer Vision*, pp. 321-331, 1988.
- [14] M. J. Chen, Y. P. Hsieh and Y. P. Wang, "Multi-resolution shape coding algorithm for MPEG-4," *IEEE Transactions on Consumer Electronics*, Vol. 46, pp. 505-513, 2000.
- [15] M. J. Swain, D. H. Ballard, "Color Indexing," *International Journal of Computer Vision*, Vol. 7, No. 1, pp.11-32, 1991.
- [16] N. Kehtarnavaz, F. Rajkotwala, "Real-Time Vision-based Detection of Waiting Pedestrians," *Real-Time Imaging*, pp. 433-440, 1997.
- [17] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, pp. 641-647, 1994.
- [18] S. Amarnag, R. S. Kumaran and J. N. Gowdy, "Real Time eye Tracking for Human Computer Interfaces," *International Conference on Multimedia and Expo*, Vol. 3, pp. 557-860, 2003.
- [19] S. Ye and W. J. Bell, "A simple video position-digitizer for studying animal movement patterns," *Journal of Neuroscience Methods*, Vol. 37, 1991, pp. 215-225.