

2012 年臺灣國際科學展覽會 優勝作品專輯

編號：110003

作品名稱

影像式條碼辨識系統

得獎獎項

大會獎：四等獎

作者姓名：伍從心

就讀學校：國立新竹女子高級中學

指導教師：徐以誠、王先念

關鍵字：條碼辨識、影像處理、ISBN

作者簡介



我是伍從心，今年十八歲，目前就讀新竹女中高級中學三年級。我從國中就一直對科學抱持很大的興趣與熱情，進入新竹女中後有幸就讀數理資優班，參與了專題研究。在專題課程中，我不但學習到許多專業的知識，也學習到與同伴合作的重要。在高中曾經參加過全國科展、遠哲科學趣味競賽、實驗科學能力競賽、智慧鐵人、TRML 數學競賽……等。很期待這次要參加的國際科展，希望能有很好的表現，並且能結交更多國內外的科學知音。

影像式條碼辨識系統

摘要

以隨手可得拍攝影像的工具（如智慧型手機、個人數位助理、網路攝影機等）及網際網路的便利性，讓使用者能隨時透過書籍條碼影像向遠端資料庫查詢書籍的相關訊息。本研究以「ISBN 書籍條碼」為例，經過條碼影像的擷取、偵測並分割出影像中的條碼、歪斜校正以及辨識後，透過網際網路，從 ISBN 書籍資料庫中查詢出該書籍的相關資訊，讓我們體驗行動通訊科技所帶來的便利。

在程式的實作上，除了使用現有影像處理技術中的霍夫轉換與邊緣偵測外，也搭配上自行設計的變化量分配圖，使得 ISBN 條碼的辨識能力大為提升。此外，為了將使用者查詢時等待回應的時間降至最低，採用了記憶體直接存取的方法來處理影像外，也針對程式執行時的瓶頸部分，以平行處理的機制來加速，經過在 Intel i5 M560 四核心處理器上的實際測試，一幅 900×675 像素的影像，平均只要花費 1.08 秒即可辨識出 ISBN 條碼，相當的迅速。

An Image-Based Barcode Recognition System

Abstract

With handy image shooting tools (smart phones, personal digital assistants, online webcams, etc.) and the convenience of Internet, users may search for the information of books from a remote database through ISBN barcode at any time. The research takes the ISBN barcode system as an example. Through capturing, detecting, and splitting the barcode from an image, we may enjoy the convenience of the mobile communication technologies by searching for information of books from the ISBN database via Internet.

In the program, besides the present image processing technologies such as Hough Transform and Sobel Edge Detection, we also use a self-designed Variation Histogram which greatly enhances the effectiveness of recognition. Also, in order to shorten the waiting time of the users, we use Direct Memory Access to process the image. As for trying to solve the bottle-necked part while running the program, we also design a mechanism of parallel process to increase the speed. It only takes an average of 1.08 seconds to test a 900*675 pixel image on an Intel i5 M560 4 cores CPU for identifying the ISBN barcode, which proves a convincing result.

壹、研究動機與目的：

現今幾乎所有的物品都有一個專屬的條碼，而這些條碼經過條碼機的掃描後，就能夠確認物品的資料訊息，無論在生產的控制管理、貨物的銷售管理及倉儲物流的配銷上，均可以加速作業的進行。最近發現手機上也有條碼辨識的應用程式，有別於傳統必需使用條碼機掃描的方式，而是採用手機鏡頭來掃讀條碼，這樣利用影像處理的技巧來辨識條碼，使得條碼的應用更為廣泛與便利，因為目前能夠拍攝影像的工具實在太普遍且隨手可得了。

目前手機上最熱門的條碼辨識程式當屬金揚資訊的 QuickMark，經過實際的使用測試，在正常使用的情況下，可以正確的辨識出條碼，但是若遇到光線過暗、拍攝過遠、背景雜亂或是角度傾斜時，就無法辨識出條碼的資料了。我們希望藉此研究，以國際標準書號 ISBN 條碼為例，發展出能力更佳的影像式條碼辨識系統，以便於書籍的查詢。

貳、研究設備器材及軟體：

- 一、個人電腦、筆記型電腦。
- 二、網路攝影機、照相手機。
- 三、Ulead PhotoImpact。
- 四、Visual Basic.Net 2008。
- 五、各式國際標準書號 ISBN 條碼。

參、研究過程與方法：

一、ISBN 國際標準書號條碼

條碼可分為一維條碼與二維條碼，目前一維條碼的使用最為普遍，常見的一維條碼有 UPC 碼、EAN 碼、CODE39 碼、Interleaved 2/5 碼等。ISBN（International Standard Book Number）國際標準書號條碼屬於 EAN 碼的一種，是為因應圖書出版、管理的需要，並便於國際間出版品的交流與統計所發展的一套國際統一的編號制度。

ISBN 條碼表示了 13 個數字碼，由左至右分別為 1 個導入碼、6 個左側數字碼、5 個右側數字碼與 1 個檢查碼。除了導入碼外，其餘 12 個數字碼由黑白條紋相間的碼元（黑條紋表示 1；白條紋表示 0）來表示，碼元分為 A、B、C 三類，左側數字碼使用 A 與 B 類碼元的組合來表示，右側數字碼與檢查碼則是使用 C 類碼元來表示，若碼元相加為奇數即為 A，相加為偶數即為 B 或 C。導入碼本身不以碼元來表示，而是由左側數字碼所使用 A 與 B 類碼元的組合情形來決定。

導入碼	左側數字碼的碼元類形組合
0	A A A A A A
1	A A B A B B
2	A A B B A B
3	A A B B B A
4	A B A A B B
5	A B B A A B
6	A B B B A A
7	A B A B A B
8	A B A B B A
9	A B B A B A

導入碼與左側數字碼的碼元組合之對應



圖 1 ISBN 條碼

數字碼	A 類碼元	B 類碼元	C 類碼元
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

數字碼的各類碼元組合

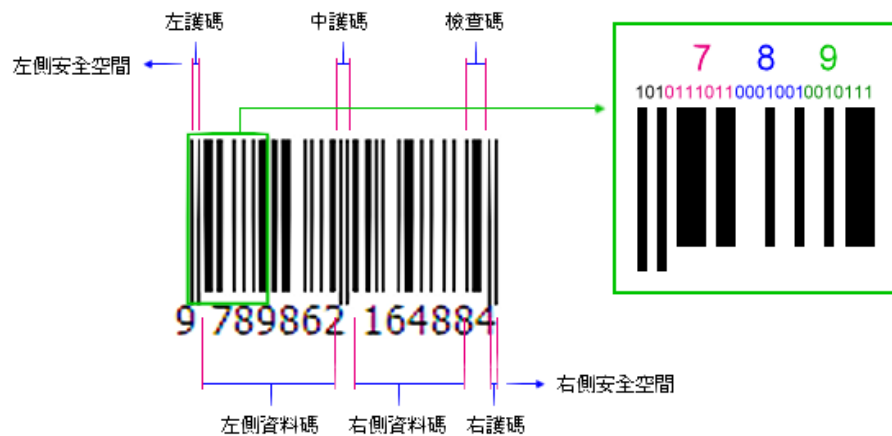


圖 2 ISBN 條碼的結構

ISBN 條碼在結構可分為 8 個部分，左側安全空間、左護碼（3 個碼元 101）、左側資料碼（42 個碼元）、中護碼（5 個碼元 01010）、右側資料碼（35 個碼元）、檢查碼（7 個碼元）、右護碼（3 個碼元 101）與右側安全空間。圖 2 中 ISBN 條碼代表的數字碼為 9789862164884，導入碼為 9，故左側數字碼 789862 的碼元類形組合為 ABBABA，7 的 A 類碼元為 0111011，8 的 B 類碼元為 0001001，9 的 B 類碼元為 0010111，如此依序完成各數字碼的碼元編碼，最後加上左、中、右護碼即可以黑白條紋繪製出該條碼。

二、條碼的辨識方法



圖 3-1 光罩式掃瞄器



圖 3-2 雷射式掃瞄器



圖 3-3 手機掃讀條碼

條碼的辨識主要是使用條碼掃描裝置，包括了掃描器、解碼器及傳輸裝置，常見的掃描器類型有雷射式與光罩式兩種，其價格依功能由數千元至數萬元。光罩式的掃描器價格較便宜且壽命較長，但缺點為掃描器與條碼間的距離不能太遠；雷射式的價格較貴，但其掃描距離可以較遠。

隨著智慧型手機的日益風行，相關的應用程式蓬勃發展，最近也出現了利用手機鏡頭來掃讀，運用影像處理的技巧來辨識條碼的方法，不過經過實際測試，其辨識能力並不盡理想，還有很大的進步空間。

三、一種便利且辨識能力較佳的查詢方式



圖 4 一種便利的書籍查詢方式

將辨識能力較佳的影像式條碼辨識程式放在手機上執行，並非是很理想的做法，因為現階段的手機，無論在處理器的速度及記憶體的资源上，均與伺服器等級的電腦相去甚遠，況且條碼辨識出來後仍需要透過網路與資料庫相連，才能獲得更多有用的資訊，所以，較好的做法是把條碼影像的辨識與資料庫查詢工作交給網路上的伺服器，手機僅負責拍攝條碼影像、傳送條碼影像及顯示查詢結果。

我們的構想如圖 4 所示，首先，使用者利用可以拍攝影像的設備（如照像手機、相機、攝影機、網路攝影機等）取得書籍上 ISBN 條碼的影像，再透過網路，將條碼影像傳送給條碼辨識伺服器，辨識出條碼後將 ISBN 書號導向 ISBN 資料庫查詢，最後，將查詢結果傳回使用者端顯示。

四、影像式條碼辨識的主要過程

影像式條碼辨識的主要過程分成條碼影像擷取、偵測並分割出影像中

的條碼、條碼歪斜校正、條碼辨識、條碼資料庫的查詢等 5 個部分，分述如下。

1. 條碼影像擷取

理論上，條碼影像的品質越好，越能夠正確的辨識，但是在現實中，因為數位科技的發達，能夠拍攝條碼的設備越來越多且隨手可得，但是擷取設備的良莠不一，加上拍攝時環境的光源影響，通常不見得有品質良好的條碼影像。

2. 偵測並分割出影像中的條碼

辨識系統要能夠從條碼影像中偵測出條碼真正的所在位置，並且將其分割出來，以供後續的條碼分析與辨識。電腦程式不像人類的視覺系統，可以很輕易的從影像中找出條碼的所在，由於條碼是黑白相間的條紋組成，參考資料[2][4]中的 Sobel 邊緣偵測法或許可以幫忙找出影像中的條碼位置。

先將影像作邊緣偵測，再將邊緣影像作垂直投影及水平投影 (Histogram 分佈圖)，就可以利用條碼的邊緣影像在垂直投影變化量大，而在水平投影累積量多的特性來偵測與分割出影像中的條碼，或許有數塊可能是條碼的區域，我們可以選擇面積最大的一塊，如圖 5 所示的青色區域。

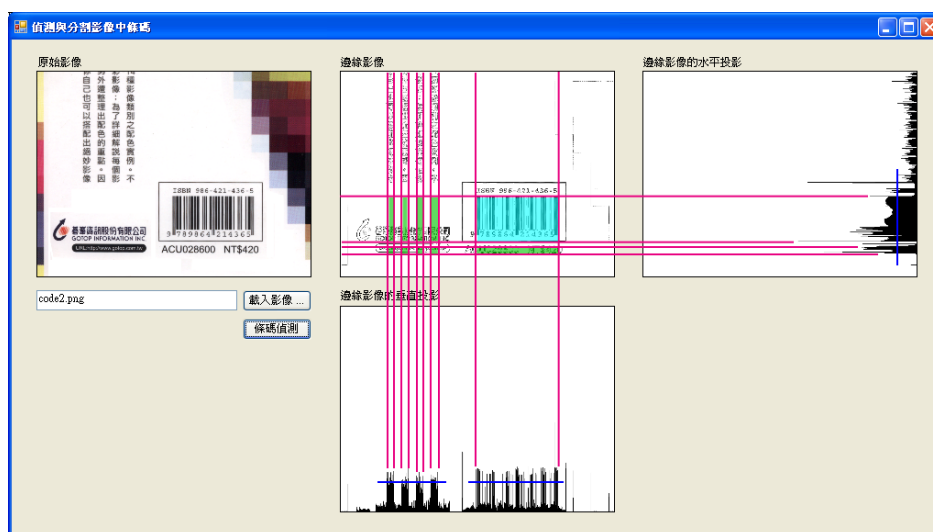


圖 5 利用邊緣偵測、垂直投影與水平投影來偵測條碼位置

3. 條碼歪斜校正

在擷取條碼影像時，條碼的編碼方向能夠越水平越好，但在真正擷取時，多少會有歪斜的情形產生，所以辨識系統要能夠將歪斜的條碼影像校正。參考資料[4]提到的霍夫轉換（Hough Transform），可以從尋找出影像中的直線，我們可將具有相同斜率的直線作數量上的統計，由於條碼中黑色條紋的邊緣都是平行直線，他們的斜率都相同，所以在該斜率的直線數量統計上一定蠻大的，只要尋找出有較大統計量的斜率，利用其斜率平均值來校正歪斜的影像。



圖 6 利用霍夫轉換計算出條碼歪斜角度並將之校正

4. 條碼辨識

透過觀察條碼的圖形，發現條碼的前後一定是空白的安全空間，而與安全空間緊鄰的是左護碼（碼元 101 黑白黑）及右護碼（碼元 101 黑白黑），所以我們就利用這個特性來確認條碼的開始與結束，以及由左護碼起始碼與右護碼間的寬度計算出每個碼元的平均寬度，再利用此平均寬度及編碼表來辨識出條碼中的每個數字。若是擔心碰到部分受污損的條碼，可採用多條的掃描線，最後以投票的方式來做決定。另外，由於左側數字碼與右側數字碼的碼元組合不一樣，所以中護碼（碼元 01010 白黑白黑白）的確認也很重要。



圖 7 條碼辨識示意圖

5. ISBN 條碼資料庫的查詢

目前所知國內有兩個網站提供了書籍 ISBN 條碼查詢的資料庫[5][6]，只要輸入書籍的 ISBN 書號的 HTTP Request，就會將該書籍的相關資訊回傳，相當的方便。



圖 8-1 翻書客 Findbook 網站



圖 8-2 行動博客來網站

五、偵測與分割出影像中的條碼

1. 邊緣影像的水平變化量

觀察條碼的邊緣影像發現，水平投影的累積量並非表達條碼位置的最好方式，因為原始影像中條碼有個外框，經過邊緣偵測後會產生水平邊緣，在水平投影上會有不小的累積量（圖 9-1 中可以看出），所以我們改採邊緣影像的水平變化量來表示，更能突顯出條碼的位置。水平變化量是將邊緣影像做一條條的水平線掃描，每條水平線由左至右的掃描並累計該

線上的點由黑變白或由白變黑的總數。



圖 9-1 邊緣影像的水平變化量

在邊緣影像的水平變化量中表示的是邊緣在水平線上的變化程度，正好符合條碼的特性，我們可以設定一個量的門檻（程式設定為平均值的 0.8），高於此門檻者才有可能是條碼的位置。另外，還可以設定一個高度的門檻（程式設定為影像高度的 $1/9$ ），小於高度門檻者將其排除，如此可以消除一些零碎的空間。在圖 9-2 中的紅色區域，就符合這兩個門檻的要求。

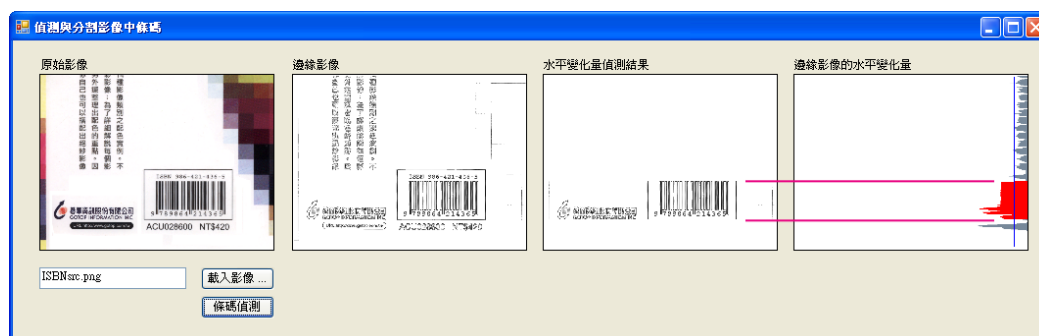


圖 9-2 邊緣影像的水平變化量偵測結果

2. 邊緣影像的垂直變化量

條碼在垂直的投影上有很密集的變化，所以我們可以同樣的概念，由切割出來的邊緣影像的垂直變化量來偵測條碼的位置。垂直變化量是將切割出來的邊緣影像做一條條的垂直線掃描，每條垂直線由上至下的掃描並累計該線上的點與其右邊鄰點黑白不同色的總數。

由於只和右邊的鄰點作比較，並不足以表達整個條碼區域在垂直投影上的密集變化，可將垂直變化量作平滑處理，更能表達密集變化的特性。同樣的，我們可以設定一個門檻（程式設定為平均值的 0.48），高於此門檻者代表變化量密集，很可能是條碼的位置，另外，也可以設定一個

寬度的門檻（程式設定為影像寬度的1/9）來消除一些零碎的空間。

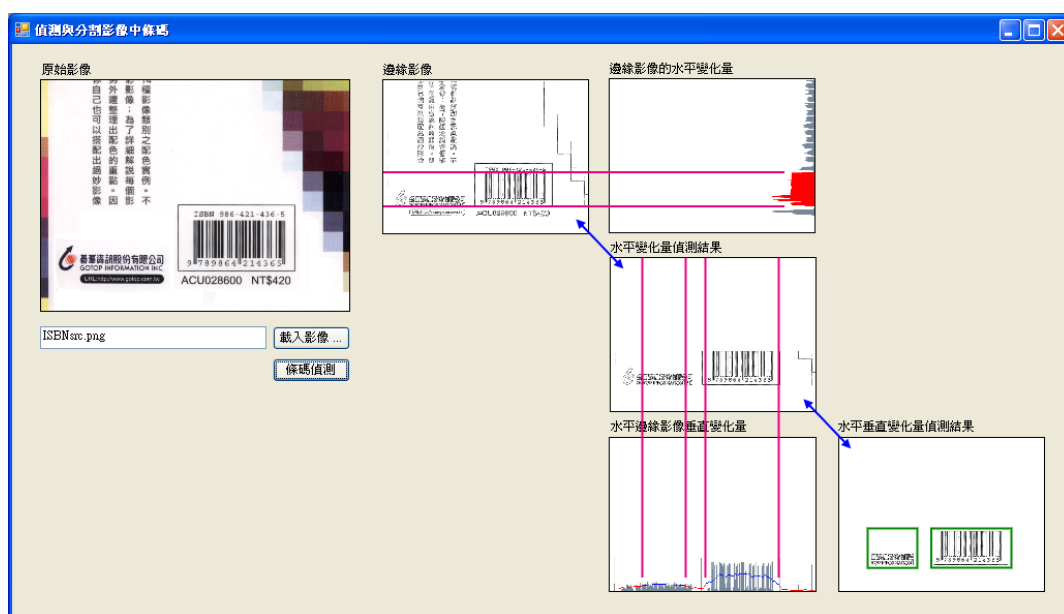


圖 10 邊緣影像的水平與垂直變化量偵測結果

3. 偵測與分割影像中的條碼



圖 11 偵測與分割影像中的條碼

邊緣影像經過水平變化量與垂直變化量偵測後，原始影像中可能是條碼的部分會被分割出來，或許有數個候選的區域，我們選擇面積最大的區域為偵測的結果，由圖 11 看來，我們的方法還蠻成功的。

六、條碼歪斜校正

我們採用了參考資料[4]所提到的霍夫轉換（Hough Transform）將條碼影像做歪斜校正，歪斜的角度由 -25° 偵測至 25° ，不過條碼影像經過偵測與分割後，可能會有部分的條碼被切除，直接套用霍夫轉換的話，可能會造成在判斷歪斜角度上的錯誤。為了克服這個問題，我們可先將條碼影像旋轉 90° 後，再交由霍夫轉換去判定歪斜角度，最後，再將條碼的原影像依此角度做歪斜校正。我們準備了 13 幅不同歪斜角度的條碼影像，分別以原有的方法及修正後的方法測試，由實驗結果得知，修正後的方法有較佳的歪斜校正能力。

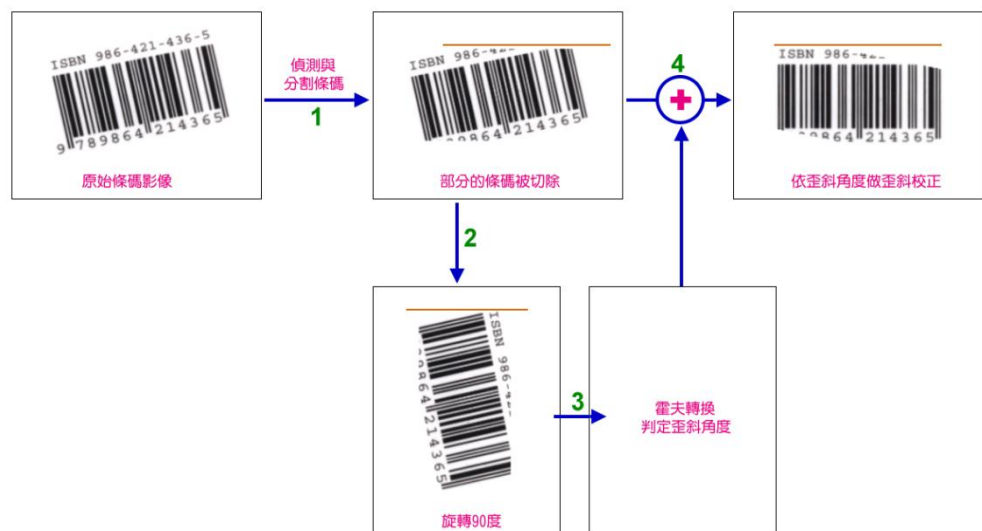


圖 12 利用修正後的霍夫轉換來做條碼歪斜校正

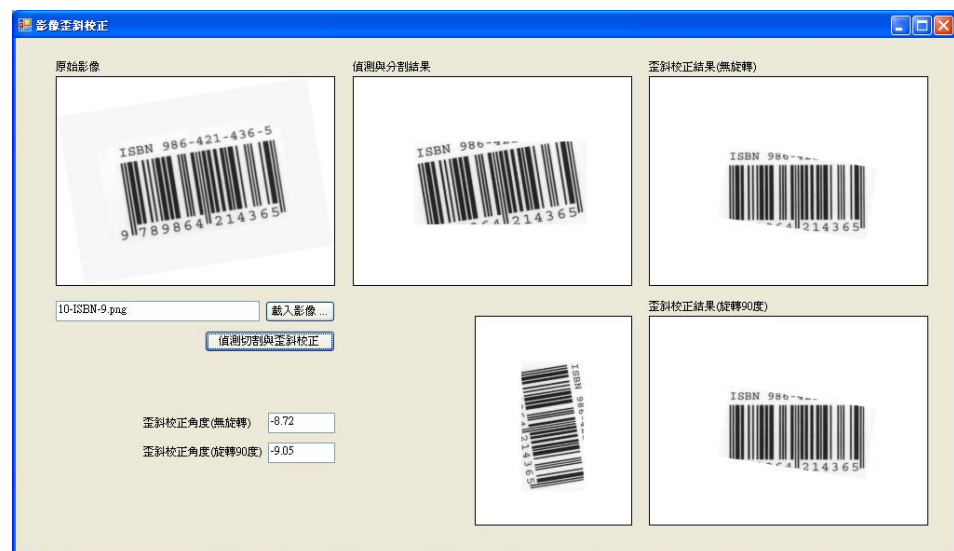


圖 13 修正後的霍夫轉換有較佳的歪斜校正能力

條碼真實歪斜角度 單位：度	霍夫轉換		修正後的霍夫轉換	
	偵測歪斜角度	誤差	偵測歪斜角度	誤差
15.00	14.79	0.21	14.97	0.03
13.00	13.03	0.03	12.92	0.08
11.00	10.85	0.15	10.90	0.10
7.00	7.17	0.17	6.92	0.08
4.00	4.01	0.01	3.96	0.04
1.00	0.88	0.12	0.99	0.01
0.00	0.05	0.05	0.00	0.00
-3.00	-2.90	0.10	-3.04	0.04
-5.00	-4.91	0.09	-5.03	0.03
-9.00	-8.72	0.28	-9.05	0.05
-11.00	-10.97	0.03	-11.05	0.05
-13.00	-12.90	0.10	-13.06	0.06
-15.00	-14.96	0.04	-15.09	0.09
平均誤差		0.11		0.05

七、條碼辨識

條碼影像經過歪斜校正後，原來偵測出來的範圍可能會有異動，所以在辨識條碼前，我們再做一次偵測與分割，分離出條碼影像後，將其水平切出 5 條掃描線（程式設定為條碼影像高度的 2/8、3/8、4/8、5/8、6/8 處），最有可能的條碼由此 5 條掃描線投票決定，並經由編碼規則解譯出此 ISBN 條碼表示的數字碼，程式執行畫面如圖 14 所示。

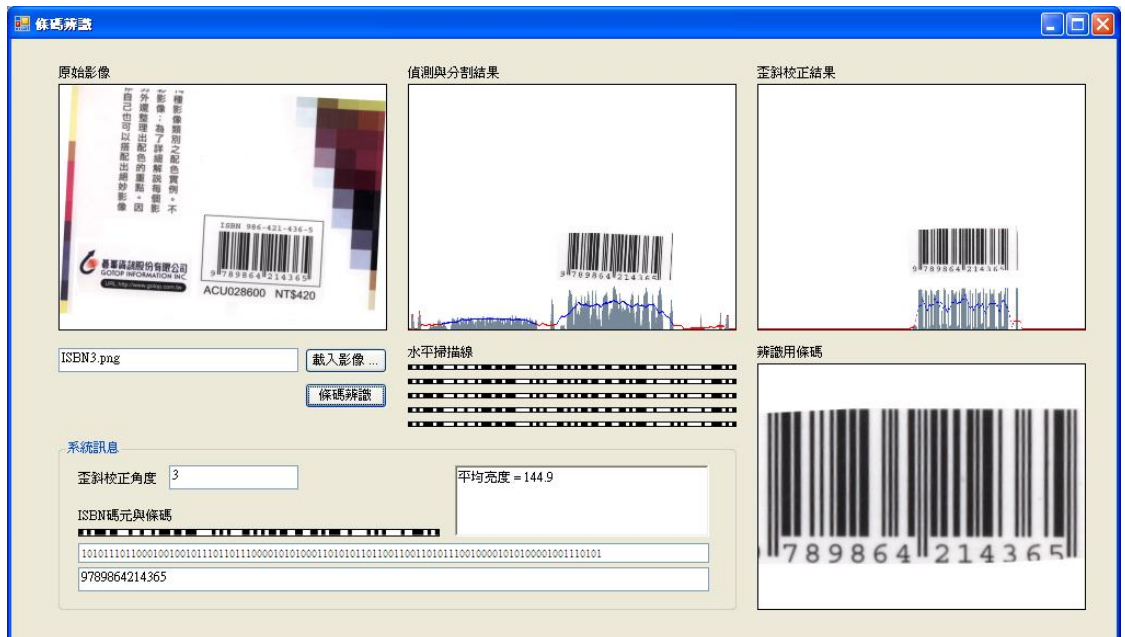


圖 14 條碼辨識實作

八、條碼資料庫查詢

目前所知國內有翻書客與博客來兩個網站提供了書籍 ISBN 條碼查詢的資料庫[5][6]，圖 15 是以翻書客網站為例的查詢畫面，只要輸入關鍵字（包括書籍條碼、作者、書名、簡介等），就可以快速的找到您所想要找的書，除了有內容簡介外，也列出了哪裡可以購買、租用、借閱，相當的方便。



圖 15-1 利用 ISBN 條碼數字查詢碼翻書客網站所提供的書籍資料庫



圖 15-2 輸入 ISBN 條碼數字



圖 15-3 網站提供了簡介與比價的功能

肆、研究結果：

一、影像式條碼辨識系統

綜合前面的研究結果，我們成功地開發了影像式條碼辨識系統，可分為手機端的應用程式與伺服器端支援條碼辨識的 Web Service 兩部份，分述如下。

1. 手機端的應用程式

現今的手機雖較以往有很長足的進步，但對於影像處理這類需要更多系統資源的程式上，執行起來仍略顯吃力，所以分配給手機端應用程式的工作比較單純，主要是負責將攝影機取得的條碼影像[3]，透過網路傳送給伺服器進行條碼辨識，然後等待伺服器回傳辨識出來的 ISBN 號碼，再呼叫瀏覽器以此 ISBN 號碼向翻書客或博客來網站進行查詢，並顯示查詢結果。

以 Visual Basic.Net 2008 進行實作，並將程式傳入 HTC Touch Diamond 2 手機中，其作業系統為 Windows Mobile 6.1 Professional，經過實際測試，確實可以正常執行，操作過程如圖 16 所示。



圖 16-1 執行條碼辨識程式

圖 16-2 開啟相機

圖 16-3 拍攝 ISBN 條碼



圖 16-4 條碼辨識中

圖 16-5 依據回傳查詢書籍

圖 16-6 顯示書籍查詢結果

2. 支援條碼辨識的 Web Service



圖 17 支援條碼辨識的 Web Service 在瀏覽器中所顯示的畫面

Web Services[3][1]是一種程式元件，它透過 Web 的通訊協定及資料格式的開放式標準（例如 HTTP、XML 及 SOAP 等）來為其他的應用程式提供服務，它可用來建構分散式系統，實現分散式架構的動態整合、平衡負擔、單元升級等優點。

同樣地，我們以 Visual Studio 2008 進行實作，並將設計出來支援條碼辨識的 Web Service 放在 Acer AR-320 的伺服器上，當接收到手機端傳送過來的條碼影像後，隨即進行條碼辨識的影像處理工作，完成後，將 ISBN 號碼回傳給手機端。圖 17 為 Web Service 在瀏覽器中所顯示的畫面，手機端的應用程式可以使用 UploadImage 的方法將條碼影像上傳，並獲得一組辨識出來的 ISBN 號碼回傳。

二、辨識能力的評估

我們隨機挑選了 10 本書（附錄一），以其 ISBN 條碼當做測試樣本，每本書的 ISBN 條碼影像有五種類型，所以共有 50 幅條碼影像，詳見附錄二。經過辨識系統的實際測試，由下列表格的實驗結果得知，ISBN 條碼辨識的平均正確率為 100%，意味著我們的影像式條碼辨識系統的辨識能力相當優秀。

編號	條碼類型 ISBN 號碼	類型 1	類型 2	類型 3	類型 4	類型 5
1	9789574996568	正確	正確	正確	正確	正確
2	9789861813240	正確	正確	正確	正確	正確
3	9789575275211	正確	正確	正確	正確	正確
4	9789574936731	正確	正確	正確	正確	正確
5	9789866275258	正確	正確	正確	正確	正確
6	9780470845622	正確	正確	正確	正確	正確
7	9789861818443	正確	正確	正確	正確	正確
8	9789861811437	正確	正確	正確	正確	正確
9	9789868271265	正確	正確	正確	正確	正確
10	9789862412534	正確	正確	正確	正確	正確

三、與現有的手機條碼辨識程式比較



圖 18 與熱門的手機條碼辨識程式 QuickMark 比較

我們也與目前手機上最熱門的條碼辨識程式（金揚資訊的行動條碼 QuickMark）做個比較，由於條碼影像的拍攝品質，諸如清晰度、對比度、亮度、歪斜變形等，直接影響了程式的辨識能力，這與相機的解像力、對焦、增益補償有關，所以我們採用愛比科技（IPEVO）的 P2V 網路攝影機當作取像設備，拍攝條碼影像後分別交由我們的影像式條碼辨識系統與 QuickMark 進行辨識（圖 18 所示）。

經過附錄一中 10 本書之條碼的實際測試比較，我們的影像式條碼辨識

系統可以正確的辨識出所有的條碼，但是令人訝異地，QuickMark 幾乎無法辨識出任何一本書的條碼。分析其原因，可能是 QuickMark 必需在條碼影像幾乎佔滿標示區域後，方可執行辨識的工作，也就是相機必需很貼近條碼，這樣的作法會有兩個問題產生，第一，若是相機的近拍功能不夠好的話，影像容易較模糊；第二，相機貼近條碼時會遮擋不少的光線，除了影像的亮度分佈不均外，若是相機自動增益補償不佳，影像的對比度也不好。這兩個問題嚴重地影響了 QuickMark 的辨識能力，也反映在測試結果上。



圖 19-1 QuickMark 必需在條碼影像佔滿標示區域後方可執行辨識

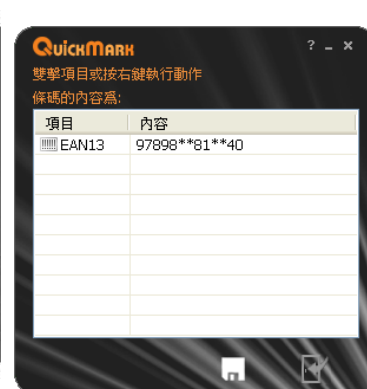


圖 19-2 QuickMark 的辨識結果



圖 20-1 我們的辨識程式可以較遠的距離拍攝

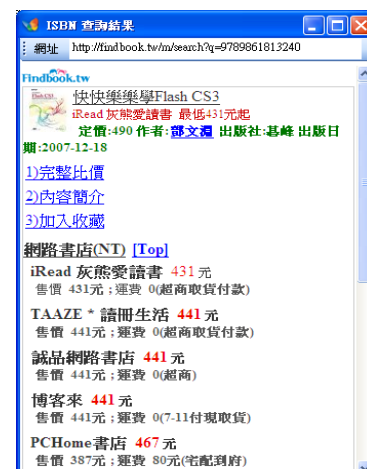
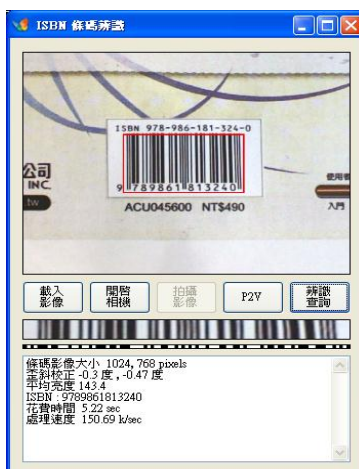


圖 20-2 我們的辨識結果

四、程式執行速度的提升

在微軟的系統中存取影像的像素，主要的方法有兩種，一種是利用 System.Drawing 命名空間對 GDI+ 基本繪圖功能所提供的存取指令 SetPixel / GetPixel，另一種是先使用 LockBits / UnLockBits 將影像鎖定在記憶體內，再將其拷貝 (Marshal.Copy) 至一維陣列，然後利用設定指標位置的方式直接存取一維陣列 (記憶體)，最後再把一維陣列拷貝回原影像。剛開始，為

求簡單，採用了 GDI+ 的 SetPixel / GetPixel 來設計程式，發現程式的執行速度真是可用牛步來形容，讓使用者等待的都是去耐心了，改用直接記憶體存取的方法後，程式的執行速度超乎想像地大為提升。

是否還可以更快呢？分析程式的各個主要程序，發現影響程式執行速度的瓶頸在邊緣偵測部分，若是可以將影像分成數個部分，分別交由不同的處理器來同時進行邊緣偵測處理，最後再將個別的結果合而為一，如此一定可以加快程式執行的速度。現今的多核心處理器也提供了非常適合的環境，經過程式實作，以圖 21 中 10 幅影像（大小為 900 像素 × 675 像素）進行測試，平行處理果真可以提升程式的整體執行速度，下列表格為不同的程式設計方法其各主要程序所花費的平均時間，測試用筆電的處理器為四核心的 Intel i5 M560，其運算時脈為 2.67GHz。另外，圖 22 為程式在循序處理與平行處理時 CPU 之使用紀錄，可以明顯看出，平行處理確實有動用了四個核心。

主要程序 設計方法	歪斜校正	邊緣偵測	水平變化	垂直變化	其他	總時間
GDI+	1.22 秒	10.34 秒	5.27 秒	1.80 秒	2.04 秒	20.67 秒
直接存取	0.11 秒	1.66 秒	0.10 秒	0.08 秒	0.37 秒	2.33 秒
平行處理	0.11 秒	0.41 秒	0.11 秒	0.09 秒	0.37 秒	1.08 秒



圖 21 測試程式執行速度用的 10 幅影像

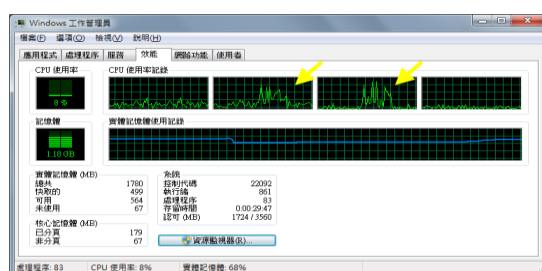


圖 22-1 循序處理程式執行時之 CPU 紀錄

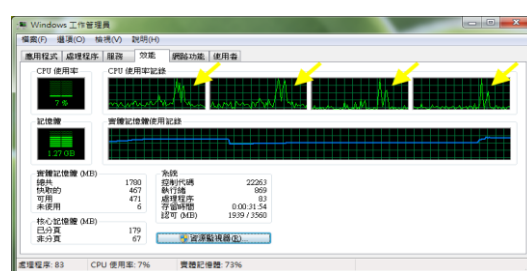


圖 22-2 平行處理程式執行時之 CPU 紀錄

伍、結論與應用：

在當今社會，條碼系統已經十分的普及。舉凡一般消費、圖書館借書、甚至飛機的登機，都需要條碼以及條碼辨識的配合。傳統的條碼辨識系統的掃描器分為光罩式掃描器和雷射式掃描器兩種，但因成本高昂且攜帶不方便，現在市面上已經發展出以相機為掃描器的條碼辨識系統，但這些系統仍有些缺點限制，如必須光線充足、解析度佳且影像要清晰、拍攝背景單純、距離不可太遙遠等。

本研究為了解決上述缺點，發展了一個能校正歪斜條碼，且能在背景雜亂的情況下辨識條碼的程式。經過與其他條碼辨識系統的比較後，證實本程式具有較高的辨識能力。在程式中我們也加入了書籍條碼查詢的功能，將辨識出的條碼送至翻書客（FindBook）查詢，然後將結果回傳顯示。我們更將程式撰寫到 Web Services 中使速度大幅提升，也讓程式能在一般電腦以及手機上使用。

陸、參考資料：

- 一、江家頡,“Windows Mobile 6 應用與程式開發”,文魁資訊股份有限公司, 2008。
- 二、鍾國亮,“影像處理與電腦視覺”,台灣東華書局股份有限公司,2006。
- 三、董大偉、許雅婷,“Visual Basic.2005 程式設計與案例剖析”,旗標出版股份有限公司,2006。
- 四、繆紹綱譯,“數位影像處理”,台灣培生教育出版股份有限公司,2003。
- 五、<http://findbook.tw> 「翻書客 Findbook」網站。
- 六、<http://m.books.com.tw> 「行動博客來」網站。

評語

題目較老舊，但所提出之研究動機具說服力，也有實用價值，只是在連結網路資料庫方面之電腦實現瓶頸較高，整體而言是一佳作。

1. 報告完整性不錯，架構清晰。
2. 可以整體實物做現場演示。
3. 霍夫轉換及修正後對整體系統效能的益處有待釐清。
4. 步驟的 performance data 需條列以利對照。
5. 實驗環境的光線控制及硬體需求須注意。
6. 整體系統效能應包含影像擷取、上傳、伺服器處理、回傳結果。
7. 主題選定可再加強。
8. 業界及學界相關資料查詢。
9. 實驗取樣範圍可再加強。