

# 國立交通大學

## 資訊工程研究所

### 碩士論文

智慧型櫥櫃於居家環境之應用  
Smart Pantries for Homes



研究生：徐仲帆

指導教授：廖弘源 教授

中華民國九十五年六月

智慧型櫥櫃於居家環境之應用  
Smart Pnatries for Homes

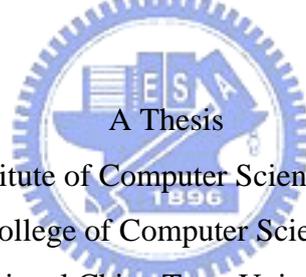
研 究 生：徐仲帆

Student：Chung-Fan Hsu

指 導 教 授：廖弘源

Advisor：Hong-Yuan Liao

國 立 交 通 大 學  
資 訊 科 學 與 工 程 研 究 所  
碩 士 論 文



A Thesis  
Submitted to Institute of Computer Science and Engineering  
College of Computer Science  
National Chiao Tung University  
in partial Fulfillment of the Requirements  
for the Degree of  
Master  
in  
Computer Science

June 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年六月

# 智慧型櫥櫃於居家照顧之應用

學生:徐仲帆

指導教授:廖弘源

國立交通大學資訊工程所碩士班

## 摘要

近幾年隨著高齡化和少子化社會的來臨，對獨居老人提供完善居家照顧的重要性也相對增加，但並不是每個家庭都有能力僱用看護提供他們完善的保護，因此在不增加人力負擔，以及使用方便且便宜的前提下，我們建構出一個智慧型家庭環境，使用電腦擔任照護的工作。本論文實作出一個智慧型櫥櫃，在櫃中物品有缺少的情況下，能自動辨識出缺少的物品，並通知商家補貨，在過去的研究文獻中，已有許多種有效的物品辨識方法，但這些方法各有優缺點，因此我們透過現有演算法的整合以及改善，提出最適合此環境及應用的辨識方法。本系統分為三個階段，第一步先從背景中將我們想辨識的物品分割出來，第二步找出物品的色彩資訊做初步的辨識，第三步使用物品的輪廓再做出更精確的搜尋，而得到最後的結果。

從實驗結果得知，經過測試資料庫後，在良好的解析度下，證實可以準確的辨識出所缺少的物品，但假若圖像品質變差，則系統無法做出正確的判斷。

因此假設環境及圖像解析度夠高，在此條件下，我們可以正確的辨識，進而達到自動補貨的目的。並希望能藉此為之後的居家照顧系

統建立基礎，並依照其特性推廣至其他如販賣機補貨系統等可能的應用。



# Smart Pantries for Homes

Student: Chung Fan Hsu

Advisor: Hong Yuan Liao

Institute of Computer Science and Engineering  
National Chiao Tung University

## Abstract

With the coming of aged society and the trend of fewer children, the importance of nursing solitary elders also increases. But not every family can afford the nursing of perfect protection. In order to decrease the burden of family, we design an intelligent, cheap, and easy-to-use computer system for nursing solitary elders.

In this thesis, we designed an intelligent cupboard. It can recognize the stuff we lack and notify the supermarket or grocery store to supply us automatically. In the past, we created many useful ways to recognize different stuff. But none of them is perfect. So we created an appropriated recognize method by modifying and integrating some existent algorithms.

There are three steps in this method. First, we subtract the background to find objects we want to recognize. Second, we use color to recognize objects. Third, we use shape to improve the precision of our algorithm and get the final results. We want use this techniques to build up the fundamental framework of home care system.



# 目錄

	頁次
中文摘要	i
英文摘要	iii
目錄	v
圖目錄	vii
一、緒論	1
1.1 研究動機	1
1.2 相關研究	2
1.2.1 特徵的選擇與描述	2
1.3 論文架構	5
二、前置處理	6
2.1 背景影像建立及二值化	6
2.2 數學形態學	7
2.2.1 膨脹	8
2.2.2 侵蝕	8
2.2.3 斷開與閉合	9
2.2.4 連通分量	9
2.3 陰影偵測	11
2.4 主軸校正	12
2.5 影像品質改善	14
三、特徵選擇與比對	15
3.1 顏色	15
3.1.1 主色	15

3.1.2	色彩配置	16
3.2	輪廓	18
3.2.1	邊緣偵測	18
3.2.2	log-polar histogram	20
3.2.3	角落偵測	21
3.2.4	匈牙利演算法	22
四、	實驗結果	24
4.1	物品影像樣本的取得	24
4.2	辨識結果	25
4.2.1	一般情況下的辨識結果	26
4.2.2	解析度與精確度之間的關係	28
4.2.3	所需時間	29
五、	結論與未來工作	32
5.1	結論	32
5.2	未來工作	32
	參考文獻	34



## 附圖目錄

	頁次
圖 1.1: 智慧型櫥櫃示意圖	2
圖 2.1: 背景剷除流程圖	6
圖 2.2: 二值化示意圖	7
圖 2.3: 膨脹示意圖	8
圖 2.4: 閉合示意圖	9
圖 2.5: 連通分量示意圖	10
圖 2.6: 使用數學形態學刪除雜訊的結果	10
圖 2.7: 使用 Sakbot 演算法去除陰影後的結果	12
圖 2.8: 主軸校正後的結果	14
圖 2.9: 最近鄰居決定法示意圖	14
圖 3.1: 顏色位移會造成的錯誤判斷	16
圖 3.2: 主色相同但顏色分布不同的各種圖形	16
圖 3.3: 使用色彩配置所得的部份結果	17
圖 3.4: 顏色分佈相同但形狀不相同	18
圖 3.5: 邊緣偵測的結果	20
圖 3.6: log-polar mask 示意圖	20
圖 3.7: 所得的 log-polar histogram 示意圖	21
圖 3.8: 角落偵測結果	22

圖 4.1: 一個物品的不同角度圖片	24
圖 4.2: 資料庫中的所有物品	25
圖 4.3: 普通情況的辨識結果	27
圖 4.4: 傾斜情況的辨識結果	28
圖 4.5: 各個解析度中辨識失敗的物品數	28
圖 4.6: 辨識率與解析度曲線圖	29
圖 4.7: 物品辨識所需時間以及加速後的成果	30



# 第一章

## 緒論

### 1.1 研究動機(Motivation)

隨著台灣出生率的下降，勞動人口的減少，高齡化社會已經漸漸來臨，六十五歲以上的人口比率即將超越十五歲以下所佔的人口比率，而對獨居老人的居家照護也越形重要，但為每一個獨居老人設置一個專職的看護不但浪費人力，而且一般家庭也無法負擔。因此設計以銀髮族為使用對象，容易操作且可靠的產品越來越受重視，藉由電腦的監控管理，利用龐大的計算能力，自動完成各種事情，而不需任何人力介入，以降低居家照護所需的成本。

這些產品不僅將可改善銀髮族的生活品質，而且也將為台灣的資訊與通訊產業提供一個新的發展契機。由於之前相關的老年照顧計畫多半著重於監控及照顧殘疾及瀕死的病人，但隨著醫療技術的日益精進，大部分的銀髮族都能夠獨立自主地活動，因此為了彌補在這方面的不足，中研院與新竹生醫科技園區合作，在一九九五年新增了 SISARL(Sensor Information Systems(Services) for Active Retirees and Assisted Living.)計畫，希望能運用各式各樣的技術，促進設計、生產、品質控制、及維護高品質、低成本的 SISARL 設備和服務。促使台灣的資訊產業活躍於老年照護技術工業的舞台。

一般而言，居家照護包含了許多部分，而有些部分只是為了增進人們的生活品質，而不是專為照顧退休老人而設計。例如物件定位系統(object locator)，在生活中，我們時常將一些像鑰匙，遙控器等小東西遺忘在家中的某處，而物件定位系統可以幫助我們快速的找出這些物品的位置。或是智慧型櫥櫃，在櫃中物品缺少的時候，能自動通知商家補貨，而不用自己出外採買。或是行人輔助器(walker' s buddy)，可以自動偵測地上是否有突起或凹下或其他的障礙物，並事先提出警告，以防走路時沒有發覺而被絆倒。或是智慧型藥盒，可以自動提醒使用者按照時程服藥，以及能偵測異常生理狀況如血壓、呼吸、心跳等重要生命徵象並及時通知相關的醫療院所或家人的監測系統，協助老年人做家事的智慧電器等。

而本論文的目的，是要探討如何使用影像處理及電腦視覺等相關技

術，來發展智慧型櫥櫃的實作方法。方法為將攝影機架設在櫃子正前方，隨時隨地擷取櫃子內部的情況，假設櫃中物品有了變動，我們即將得到的影像做分析，使用影像處理的演算法及電腦自動辨識技術，並與資料庫做比對，找出確實的物品為何。如下圖所示：

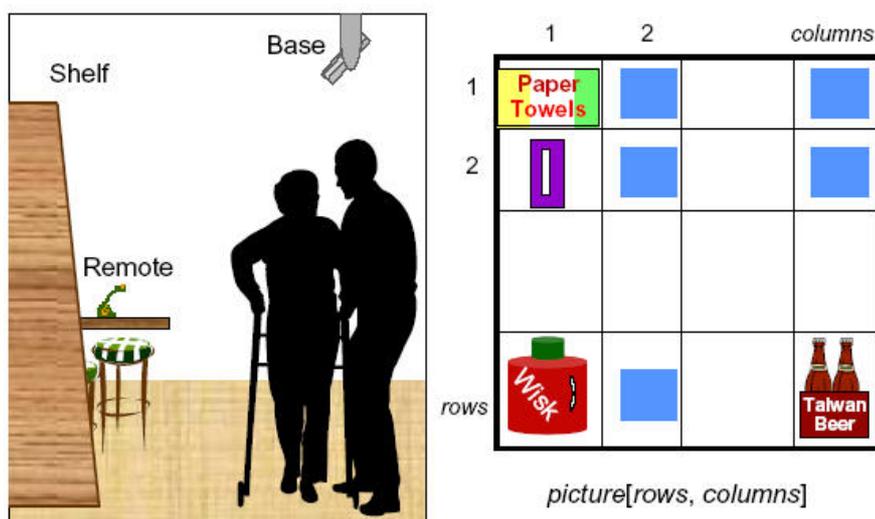


圖 1.1: 智慧型櫥櫃示意圖

我們發展此演算法，以期能為智慧型居家管理建立基礎。而此技術可進一步推廣至便利商店或是販賣機的自動補貨系統上，藉由攝影機 24 小時的監控，可以適時的找出目前何種商品已售罄，並立即通知廠商完成補貨，同時便利消費者與廠商。

## 1.2 相關研究

智慧型櫥櫃為一種物品辨識(Object recognition)系統於居家照顧的應用，有許多的方法可以達成這個目標。我們可以在櫥櫃中裝設 RFID reader，如此即可隨時隨地確認櫃子內有什麼物品，但 RFID 目前尚未普及，若應用在智慧型櫥櫃上會有成本過高的缺點，不符合期望。而退一步，我們可以使用物品條碼(bar code)，與 RFID 相同，櫥櫃藉由物品的條碼來識別櫃中的物品，但與購買物品時一樣，人們使用物品前必須先掃過條碼後才可拿取或擺入，會有不夠方便的缺點，最後一種方法使用圖形識別，利用所得的圖片做分析，以了解櫥櫃內的物品為何，其特點在於前處理(含影像處理)對於辨識的正確率影響相當大。物品辨識通常分為三個部份，第一部分為選擇適合的物品特徵，第二部份為描述我們剛才選擇的特徵，第三

部份則是拿我們剛才所描述的特徵與資料庫做比對，而得到最後的結果。由於描述的方法通常與所選擇的特徵是相關的，因此以下將這兩部分合併來介紹。

### 1.2.1 特徵的選擇與描述

除了前處理外，最重要的就是特徵抽取(feature extraction)，物品辨識分為 3D 以及 2D 兩種維度的特徵，由於本論文輸入的資料以影像為主，因此以下介紹也將以 2D image 的辨識方法為主。而最簡單的方法，我們使用物品的圖片作為特徵，稱為樣本比對(template matching)，將物品的圖片與整張圖片做比對，如果有一個地方與此物品圖片吻合，則確認此物品在圖片中。但此方法既費時且無法解決旋轉與縮放的問題。因此經過多年的研究，我們發展出相當多的特徵抽取方法，主要可以分為區域特徵與整體特徵兩種，兩者的差別在於區域特徵通常將影像分割為數個區塊，在每個區塊中擷取特徵，又稱為細胞式特徵(cellular feature)，整體特徵則多半藉由整體的轉換得到的數值來作為特徵，因此當物品的某部份被遮蔽或被切割，擷取特徵時會發生問題，我們通常依情境的不同來選擇。常使用的特徵有形狀輪廓(shape)，色彩分布(color)，邊緣分布(edge)，材質(texture)，或直接抽取物品上某些特徵點。

而描述的方法也隨著特徵有所不同，例如輪廓大致上分為以輪廓為基礎(contour-based)及以區域為基礎(region-based)兩種描述方法，以輪廓為基礎的方法僅以物品的邊緣做為描述的重點，捨棄其他的資訊，例如 Crim 使用富利葉描述法(Fourier descriptor) [1, 2]，將物品的周圍透過複利葉轉換(Fourier transform)而得到的係數來描述物體的形狀，由於這些係數不會因為物品旋轉或縮放產生變化，可以避免這些常見的問題。Mahn 使用鏈碼(chain code)[2]，用適當的網格來近似物體，以線段的方向來表示物品的輪廓，有四方向跟八方向兩種表示方法，但容易受雜訊影響，且無法解決轉動及比例的問題。多邊型近似法(polygonal approximation)，找出某些適當點作為頂點，頂點與頂點間的連線以多邊形線段來表示邊界，常用在線性片段的近似，缺點為較無法表示曲線。而 Mackworth 和 Mokhtarian 使用 CSS(Curvature scale space)[3]，對物品的輪廓做高斯平滑(Gaussian smooth)，直到成為一個平滑的曲線為止，在平滑化的過程

中，輪廓上的反曲點會逐漸消失，我們可繪製一個 CSS 影像，記錄每個反曲點與平滑係數之間的關係，並以此作為物品的特徵，但缺點為若輪廓本身並不存在太多的反曲點，如圓形或橢圓形以及矩形，則所得結果不會有差別，無法作為辨識之用。

而以區域為基礎的方法則是描述整個物品佔有的區域範圍，例如 Hu 使用動差(Hu moment)[4]，利用物品的座標值及像素值計算出七個動差值，並拿他們來作為比對的依據，可以避免旋轉及縮放的問題。另一個 Mpeg-7 所採用的特徵為 ART(Angular Radial Transformation)[5, 6, 7]，利用 ART 的基底函數以及影像的像素值和極座標(polar coordinate)上的座標值來計算並比對，但缺點為這些特徵較無法提供符合人們感官的相似度驗證，並不直覺，且為整體特徵，無法得到一些物品較細部的資訊。

除了輪廓之外，還有許多的特徵可以利用，例如 Park 與 Won 利用邊直方圖(edge histogram)，將圖片分為數百格，紀錄每一格內邊的方向，並累加起來，如此可同時獲得物品區域以及整體的邊資訊。或顏色(color)，與邊直方圖相同，我們使用顏色直方圖來記錄物品的顏色分布，在 Beitao 等人的研究中，還結合了色彩動差(color moment)，只要整張影像的色彩分類夠多，並取得足夠的動差資訊，即可大概的表示整張顏色的色彩資訊。在 MPEG-7 中，還有所謂的 scalable color[7]，先將 RGB 轉成較符合人類感官的色彩空間(color space)HSV，並將整個空間等分成 256 個部份，使用一個三維陣列表示，將圖片的像素依照我們所劃分的值來歸類至陣列中，即可得到圖片的 HSV 值分布，但以上皆為整體特徵，無法得到色彩與空間的關係。因此 Huang 使用 Color correlogram[8]，算出離一個像素各個距離色彩值為  $n$  的機率，可同時紀錄顏色及空間的資訊。而為了彌補單一特徵的不足，有些研究使用了兩種以上的特徵描述，例如 Diplos 等人結合了顏色和形狀進行物品辨識，藉由更精確的比對達到準確的辨識結果。

或使用一些其他的特徵點，Schmid 和 Mohr 使用角落(corner)作為辨識所需的特徵點，並進一步對其作描述以為影像檢索(image retrieval)之用。但會有縮放的問題而造成不準確，因此 Lowe 提出 SIFT(scale invariant feature transform)，先找出物品 DoG(difference of Gaussian)的區域最大及最小值作為特徵點，再針對取出的特徵點，以一個八乘八的梯度直方圖來描述，可以解決旋轉及縮放的問題，但缺點為特徵的維度太高，使計

算速度緩慢，因此 Sukthankar 提出 PCA-SIFT[9]與 Mikolajczyk 提出 GLOH[19]，兩者皆使用 PCA 來降低特徵點的維度以達到更精簡的表示方式，並有著更佳的效能。

### 1.3 論文架構

本篇論文共分成五章，第一章是簡介，包含了研究的動機以及相關的物品辨識技術，包含了最近的發展和應用，以及本篇論文的架構。

第二章是前處理，介紹本論文所使用的基本影像處理技巧，如數學形態學，邊緣偵測，陰影偵測等技術，去除背景及其中的雜訊，以取得圖片中物品所包含的資訊。

第三章為本篇論文的主要部份，介紹我們的主體演算法，使用第二章所得的處理結果來描述和辨識物品。

第四章為實驗結果，拿我們自己建立的居家用品資料庫來做測試，觀察我們的演算法能否得到理想的結果，並找出精確度與解析度之間的關係，以了解建構此系統需多少成本。

第五章是結論和未來的工作，針對本篇論文給予簡單扼要的結論，並提出一些看法與意見以及之後可能的研究方向。

## 第二章 前置處理

由於在系統中輸入的是直接由攝影機取得的影像，資訊太過雜亂，無法直接用於辨識系統，因此必須強化影像的品質，以提高後續模組的效率以及辨識率，在此我們使用一些基礎的影像處理方法做前置處理，找出物品在影像中的確實位置以及辨識所需的各項資訊。

而主要所使用的流程如下圖：

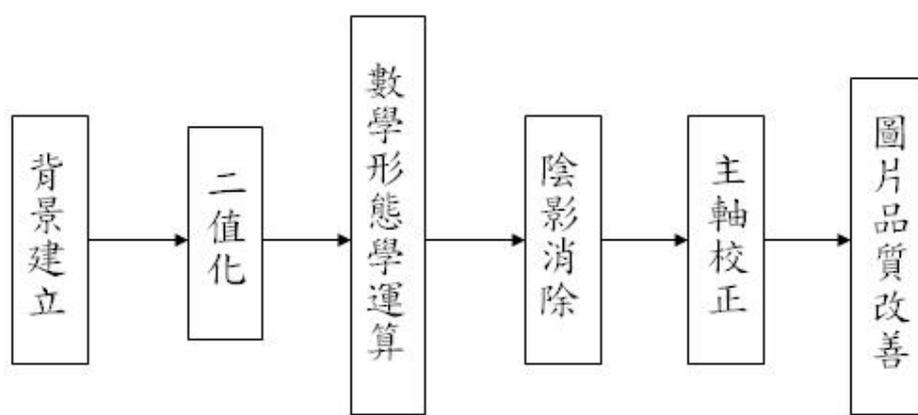


圖 2.1: 背景剷除流程圖

以下對每一步驟做詳細的介紹。

### 2.1 背景影像建立及二值化

前景物的萃取主要在將系統所感興趣的物體與背景分隔出來，因此在本系統中，必須建立一個可靠且穩定的背景影像。而背景的建立有許多種方法，例如先照數十張的背景，取每一張圖片的平均值後即將其設定為背景。或使用亮度值累積，累積變異數並統計資料，以動態更新背景的亮度資料等，主要視環境的不同而採用最適合的辦法。由於在本系統中設定的實驗場所為室內，不會受到風或太陽光源或背景擾動的影響，環境穩定，因此我們使用數張圖片的平均值做為我們的背景模型。

取得背景後，將輸入的影像與之相減，並做二值化，由於二值化後，原本圖片轉換成一張只有黑白顏色的影像，因此處理的資料量變小，速度也

會相對變快。而二值化的規則如下：

$$f(x, y) = \begin{cases} 1 & \text{if } |I(x, y) - B(x, y)| > T_{obj} \\ 0 & \text{otherwise} \end{cases} \quad (\text{公式 2.1})$$

其中  $B(x, y)$  以及  $I(x, y)$  分別為背景模型以及輸入的圖片在像素  $(x, y)$  中的灰階值，而  $T_{obj}$  為我們設定的門檻值(threshold)，以此為標準，將影像像素相減後，小於門檻值的設為零，大於門檻值的設為一，如此即可順利將大部份的前景切割出來，二值化的示意圖如下：

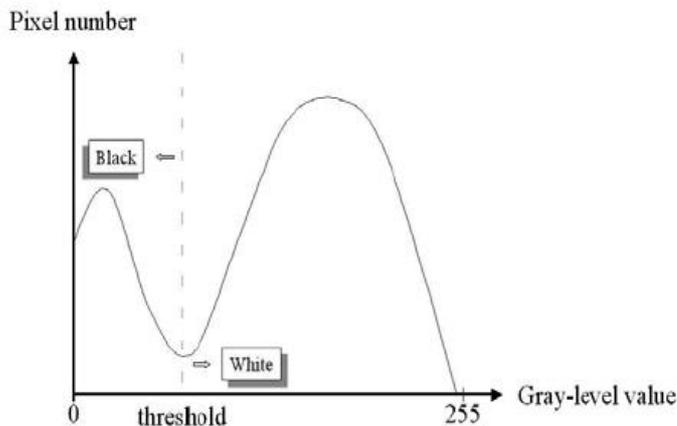


圖 2.2: 二值化示意圖

但實驗室的環境雖然穩定，還是存在著一些影響使得所得結果不只是前景，還包含了若干雜訊，因此我們再使用數學形態學以去除雜訊(remove noise)。

## 2.2 數學形態學

數學形態學是 Mathron 跟 Serra 於 1964 年所提出，其為基於許多簡單的集合理論而成的影像分析方法，是使用以數學為背景的方法來表示及描述區域形態，較常看到的應用為：邊緣偵測(Edge detection)，雜訊消除(Noise removal)，影像強化(image enhancement)，影像切割(image segmentation)等，其中有四個較為基本的運算，分別為：膨脹(Dilation)，侵蝕(Erosion)，斷開(Opening)，閉合(closing)，膨脹與侵蝕兩種運算均需要兩個部份，一個稱為主體影像(active image)，即為須在上面做運算的原始圖像，一個稱為核心影像(kernel image)，亦稱為結構元素(structuring element)，當然，膨脹與侵蝕的使用方法並不唯一，通常作

用於二元圖(Binary)，灰階圖(Gray Level image)與彩圖(Color image)。

### 2.2.1 膨脹(Binary dilation)

假設A與B為Z上的集合，A藉由B而膨脹，我們記為  $A \oplus B$ ，定義如下：

$$A \oplus B = \left\{ z \mid [(B)_{z} \cap A \neq \phi] \right\} \quad (\text{公式 2.2})$$

在系統中 A 為結構影像，B 為結構元素，簡單來說，A 藉由 B 膨脹即為所有可以使 B 在平移 z 後與 A 至少重疊一個非零元素的 z 的集合，也就是 B 在 A 內移動所能涵蓋到的範圍，如下圖所示即為 B 移動後所能涵蓋的最大範圍。Dilation 最簡單的二元圖應用是用來橋接縫隙，因為經結構元件處理過後，空缺會填滿且物體形狀增大，我們可以利用膨脹來填補物品之中的破洞，或把影像邊緣加粗來達到目的。

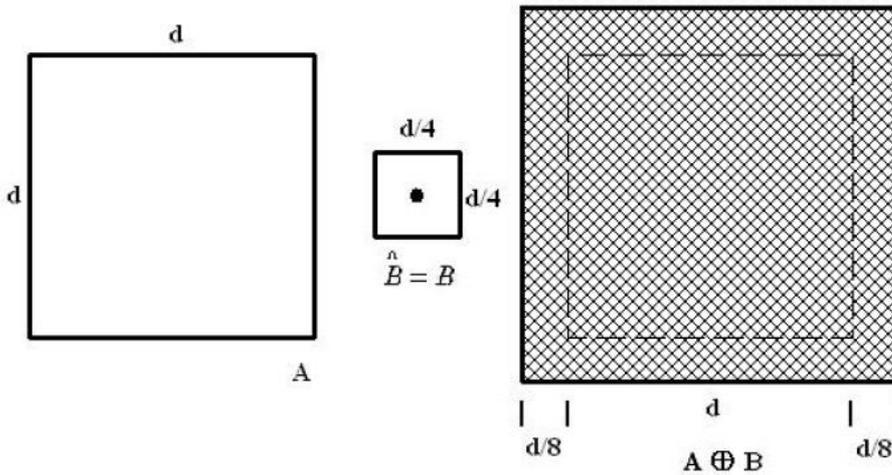


圖 2.3:膨脹示意圖

### 2.2.2 侵蝕(Binary erosion)

對於 Z 上的兩個集合 A 與 B，A 藉由 B 侵蝕，我們記為  $A \ominus B$ ，在此我們定義為：

$$A \ominus B = \left\{ z \mid [(B)_{z} \cap A \subseteq \phi] \right\} \quad (\text{公式 2.3})$$

與膨脹相反，侵蝕為 B 平移 z 後包含在 A 中所有點 x 的集合，即 B 在影像 A 的範圍中移動，B 的原點所涵蓋到的範圍。而經結構元件處理過後，可以將多餘且較細小的枝幹消除且將物體形狀縮小，因此可除去二元影像中不相

關的細節，達到去除雜訊的目的。

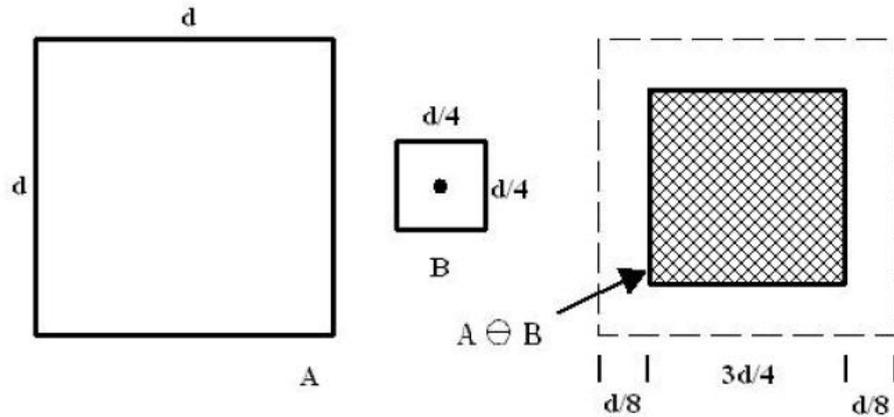


圖 2.4: 閉合示意圖

### 2.2.3 斷開與閉合(Opening and Closing)

斷開與閉合常為影像處理中兩個常用的型態運算，侵蝕與膨脹經常是一起搭配運算，而不是分開執行，而斷開與閉合只是侵蝕膨脹的前後順序不同，先做侵蝕再做膨脹稱為斷開，目的通常為平滑輪廓，截斷窄的細線，消除較細的分支，先做膨脹再做收縮則稱為閉合，目的為把周圍較窄的中斷部分和長的細缺口連接起來，消除內部小孔洞及填補輪廓上的缺口。斷開與閉合的定義如下：

影像 I 被結構運算子作斷開運算，記為  $I \circ S$  定義如下：

$$I \circ S = (I \ominus S) \oplus S \quad (\text{公式 2.4})$$

影像 I 被結構運算子作閉合運算，記為  $I \bullet S$  定義如下：

$$I \bullet S = (I \oplus S) \ominus S \quad (\text{公式 2.5})$$

一張影像若做完斷開的運算，則先收縮再膨脹後，圖形特徵會有雜點消失，骨架變瘦，周圍的細邊被除去等等的變化，此處理適合做毛邊去除的工作，若為閉合，由於經過膨脹，可將內部的小孔洞填滿，再經收縮即可將外型回復原狀，適合作為找尋輪廓的工具。

### 2.2.4 連通分量 (Connected component)

假設 Y 為在集合 A 中的一個連通分量(connected component)，假設我們已知 p 為 Y 中的一個點，則我們可用以下公式由 p 找出整個 Y 的連通分

量：

$$X_k = (X_{k-1} \oplus B) \cap A \quad k = 1, 2, 3, \dots \quad (\text{公式 2.6})$$

在此我們假設  $X_0$  為初始的點  $p$ ，而  $B$  為結構元素。使用遞迴計算，假如最後  $X_k = X_{k-1}$ ，則此演算法收斂，而可得到最後  $X_k = Y$  的結果。實際上我們輸入一張二值化影像，並在運算後輸出一張以符號表示的符號影像。通常根據四連通(4-adjacent)或八連通(8-adjacent)兩種規則來計算一個像素是否為另一個像素的鄰居。由於二值化後的結果中，細小的雜訊佔了大多數，而真正的物品面積通常較大，因此我們使用連通分量找出每個物品的面積大小，並設一個門檻值，假若面積低於此門檻值，則認定此區塊為雜訊並刪除。門檻值的選定則由圖片輸入解析度的大小而決定。

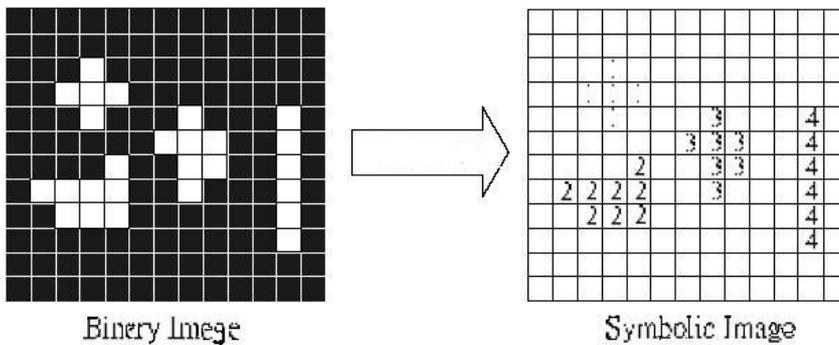


圖 2.5: 連通分量示意圖

使用數學型態學處理過後如下圖，可以看出大部分細小的，與物品不相關的雜訊已經幾乎完全去除了。

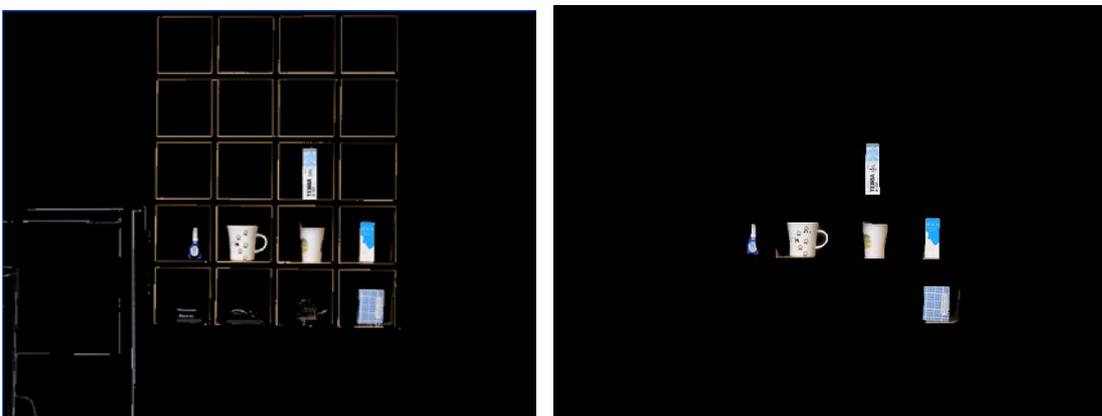


圖 2.6: 使用數學形態學刪除雜訊的結果

## 2.3 陰影偵測

從將所得資料做數學型態運算後的結果中可以發現，不管是戶內或戶外，由於光線的照射具有方向性，因此在每個物品的周圍，或多或少都會出現陰影(Shadow)的情況，輕微的陰影會在做背景相減時無法通過設定的門檻而被忽略，但太嚴重則無法直接去除，若不處理陰影的問題，則會對之後我們取得的物品資訊造成影響，進而使辨識的精確度下降。

因此我們使用 Andrea 等人所開發的演算法，稱為 Sakbot(Statistical And Knowledge-Based Object Tracker ) [13]。此演算法概念為，由於人類的眼睛可以輕易的分辨出陰影與正常物體的不同，因此首先將圖片由 RGB 轉為 HSV，使之較符合人類的感官，也較容易正確的偵測出陰影的所在。經觀察結果得知，假如圖片中某一個像素為陰影，則此像素的發光度會較之前背景來的低，且飽和度也會因為陰影的影響而降低，因此我們可以假設一個可偵測陰影點的遮罩  $SP_k$ ，而遮罩設定如下：

$$SP_k(x, y) = \begin{cases} 1 & \text{if } \alpha \leq \frac{I_k^V(x, y)}{B_k^V(x, y)} \leq \beta \\ & \wedge (I_k^S(x, y) - B_k^S(x, y)) \leq \tau_S \\ & \wedge |I_k^H(x, y) - B_k^H(x, y)| \leq \tau_H \\ 0 & \text{otherwise} \end{cases} \quad (\text{公式 2.7})$$

式中  $I_k^V(x, y)$ 、 $I_k^S(x, y)$ 、 $I_k^H(x, y)$  分別代表了輸入的前景圖片在座標點  $(x, y)$  中的強度(intensity)、飽和度(saturation)與色度(hue)的值。相同的， $B_k^V(x, y)$ 、 $B_k^S(x, y)$ 、 $B_k^H(x, y)$  則分別代表了我們設定的背景圖片在座標點  $(x, y)$  的強度、飽和度以及色度的值。由式...中可看到此公式有四個門檻值讓我們可以根據環境的不同作調整，其中  $\beta$  的主要作用是去除雜訊，以讓整個系統能更強固(robust)，而  $\alpha$  主要是考慮環境的光源亮度以及物品的反射係數或是否自成一個光源，由觀察得知，假如之前背景圖片的光源越強越亮，則  $\alpha$  的選擇必須越小，反之則越大，而  $\tau_S$  與  $\tau_H$  的值並沒有特別的依據，大部分依靠經驗法則決定。

下圖即為使用此演算法偵測並消除陰影後的結果：

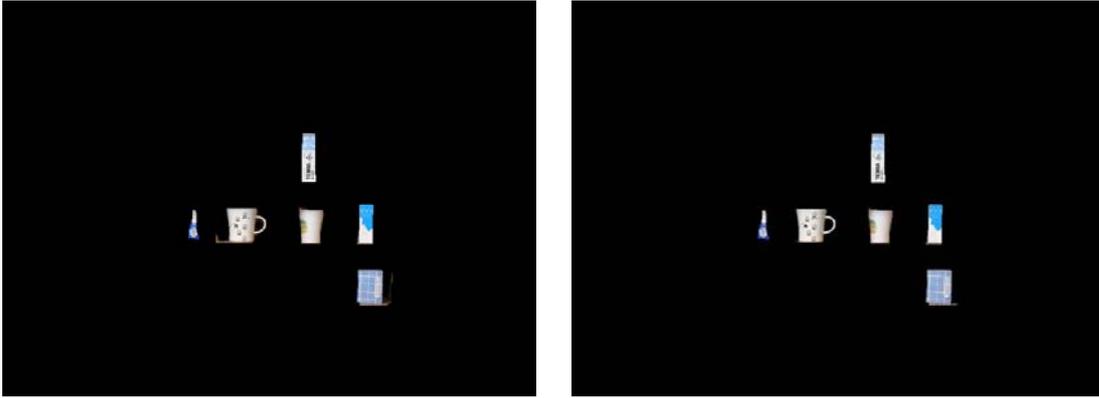


圖 2.7: 使用 Sakbot 演算法去除陰影後的結果。

其中可看出白色杯子邊緣與藍色陰影旁的陰影已被偵測出來並消除，但藍色盒子的邊緣可看出有些陰影沒有消除，這是因為門檻值的選擇所造成，但由於已消除了大部分陰影，因此對之後識別不會造成太大影響。

## 2.4 主軸校正

由於本系統主要作為居家照顧之用，鑑於老人手腳不靈活，在擺放物品到櫃子時，物品可能不慎倒下或傾斜，造成辨識困難，而這也是物品辨識中常發生的旋轉(translation)問題。對此通常使用兩種解決辦法，第一，在資料庫中新增每一個物品傾斜以及倒下的影像，即使所得的前景可能傾斜各種角度，資料庫中仍有相對應的資料可以做比對。但由於物品繁多，且物品角度改變後顏色及形狀也會產生相當大的改變，這種作法會使資料庫中的資料爆增數倍，是相當不切實際的。第二，使用動差，找出物品的質心及主軸角度，若主軸角度傾斜，則我們使用適當的方法校正，使之回復到正常的情況，如此便不用考慮一般傾斜角度的情況，只需在資料庫中新增物品在不同方向橫倒的資料即可，對系統的負擔較小。

動差為一描述形狀的特徵，常用於圖型識別(pattern recognition)中，公式如下：

$$m_{pq} = \sum_i \sum_j I(i, j) i^p j^q \quad (\text{公式 2.8})$$

其中  $I(i, j)$  為影像在座標  $(i, j)$  的像素值。由於在系統中我們使用二元圖，因此像素值分布如下：

$$I(i, j) = \begin{cases} 1 & (i, j) \in C \\ 0 & (i, j) \text{ otherwise} \end{cases} \quad (\text{公式 2.9})$$

其中  $C$  代表前景中被我們認定為物品之中的點。而下式中定義了物品區塊的質心位置  $(\bar{x}, \bar{y})$ 。

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad (\text{公式 2.10})$$

$$\bar{y} = \frac{m_{01}}{m_{00}} \quad (\text{公式 2.11})$$

其中  $m_{00}$  為物品面積， $m_{10}$  為物品  $x$  軸座標相加的結果，同理， $m_{01}$  為物品  $y$  軸座標相加的結果。

$$\mu_{pq} = \sum_i \sum_j (i - \bar{x})^p (j - \bar{y})^q \quad (i, j) \in C \quad (\text{公式 2.12})$$

可經由上式導出物品主軸傾斜的角度，公式如下：

$$\theta = \frac{1}{2} \tan^{-1} \left[ \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right] \quad (\text{公式 2.13})$$

如此即可經由這些運算找出主軸與  $x$  軸間角度的度量，求出物品主軸傾斜的角度後，我們即可使用仿射轉換(affine transform)來校正，使物品的主軸回覆正常。

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (\text{公式 2.14})$$

其中  $(x, y)$  為原圖片的座標，而  $(x', y')$  為經過仿射轉換校正後的座標。但物品的主軸可能為垂直或水平，因此我們必須加上條件限制，由觀察得知，當物品寬度大於長度時  $\mu_{20} > \mu_{02}$ ，若此條件成立，則認定物品的主軸為水平，反之，則認為主軸為垂直。而校正結果如下圖所示：

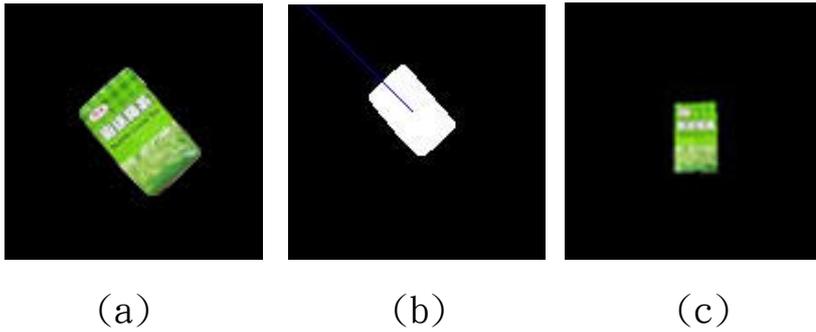


圖 2.8: (a)擺放歪斜的物品;(b)由動差求出的物品主軸;(c)主軸校正過後的物品

由圖可知，因為求得的主軸角度難免會有誤差，因此校正後的結果還是有點歪斜，與一般正常的擺放還是有些微的差距，但這並不妨礙之後的辨識步驟。

## 2.5 影像品質改善

為了達到降低成本的目的，圖片輸入的解析度可能較差，因此我們希望能對圖片品質做改善，使從中擷取的資訊能更加完整。

考慮到使用輪廓做為特徵時，由於相似度的計算方式，當圖片解析度縮小造成物品輪廓的改變，可能系統會將資料庫中顏色相近但輪廓較小的物品視為我們所查詢的物品，得到錯誤結果。為了改善這個問題，當我們從圖片擷取出物品後，依據所得圖片解析度的大小，將物品圖片使用內插法(interpolation)調整成普通解析度時的大小，試圖改善造成的影響。而依照用途不同，也有相當多的方法可以選擇，在此我們選擇較簡單的最近鄰居決定法(the nearest neighbor)，將放大後圖形的座標依比例投影到原圖形上，並選擇最接近的像素值作為該座標的像素值。如下圖所示：

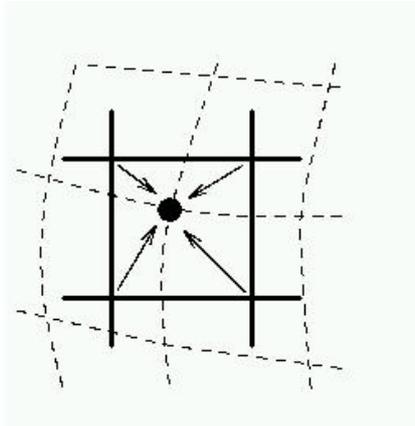


圖 2.9:最近鄰居決定法示意圖。

## 第三章 特徵選擇與比對

在系統切割出圖片前景後，下一步就是要辨識出前景中每個物品的確實身份。知道之後，就可以直接通知廠商做補貨等後續處理，不過這並不在本篇論文的討論範圍內，因此不再對此多做說明。

物品辨識通常會遇到兩大問題：一. 縮放(*scaling*)，由於攝影機離物品的距離可能有遠有近，而這會對物品的形狀造成影響，使取得的資訊與原本儲存的資料造成差異。二. 旋轉(*translation*)，由於物品只要稍有旋轉，看起來可能就會和一般情況完全不同，因而對辨識造成影響。旋轉的問題已經使用動差解決，而以下將介紹我們的演算法以及如何克服縮放所產生的問題。

在資料庫內包含許多物品，最終的目的是希望能快速且準確的辨識，因此我們採用階層式的架構，使用從粗糙到精細的搜尋比對策略，先以較簡單快速的低維度特徵刪除掉大部分資料庫中的候選物品，再使用較精細的高維度特徵做進一步的比對，以得到我們所需要的結果，由於不同的物品在顏色上通常有著不小的差距，因此我們先選擇顏色作為我們使用的特徵，得到初步的結果後，再使用輪廓作為進一步辨識所使用的特徵。

### 3.1 顏色(Color)

既然選擇了我們所使用的特徵，下一步即為描述，我們選擇了兩種描述顏色的演算法。分別描述了物品的整體以及區域顏色資訊。

#### 3.1.1 主色(Dominant color)

主色為我們使用的第一種色彩描述子(*color descriptor*)，首先在物品影像中，對色彩作量化切割(*quantization*)，並統計各種量化色彩出現的次數，作為此影像的色彩特徵描述，稱為色彩直方圖(*color histogram*)，直方圖的定義如下：

$$h_{R,G,B}(r:g:b) = N * \text{prob}(R = r, B = b, G = g) \quad (\text{公式 3.1})$$

其中  $N$  為整張圖片所包含的像素數量。

本系統中我們將直方圖均分為三十二格，並在每個波段中，找出所包含

點數最多的前三格作為代表物品的主色。求得主色之後，使用歐幾里得距離(Euclidean distance)來計算影像中物品與資料庫中物品的主色差距，假設結果低於某個門檻值(threshold)，則認定此物品為可能的候選者，並進行下一步的辨識。

而由於物品的顏色可能因為燈光的些微改變而產生差異，使比對時發生誤判，如下圖所示，因為顏色位移(color shift)，造成相似度的判斷錯誤：

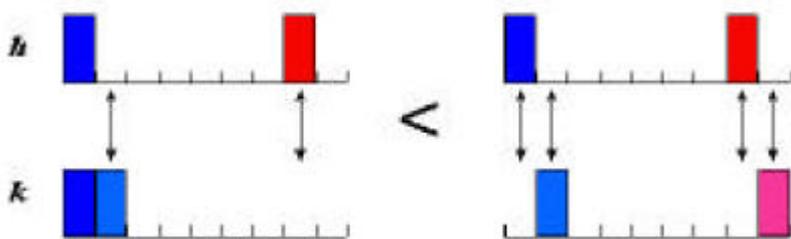


圖 3.1: 顏色位移會造成的錯誤判斷

因此我們再選擇了直方圖交叉(histogram intersection)作為相似性量測的基準，比對的公式如下：

$$D(H(I), H(I')) = \sum_1^n \min(H_j(I), H_j(I')) \quad (\text{公式 3.2})$$

其中  $n$  為直方圖的格數， $H_j(I)$  與  $H_j(I')$  為兩個物體的直方圖各自經過正規化(normalize)後的資訊。所得數值越大代表兩直方圖越相似。

### 3.1.2 色彩配置(Color layout)

由於主色是屬於整體特徵，因此最大的問題在於缺乏物品顏色和空間的相關資訊，如此會造成即使物品顏色分布很混亂，但只要整體的顏色分布相同，則會被判斷成相同的物品，如下圖所示：

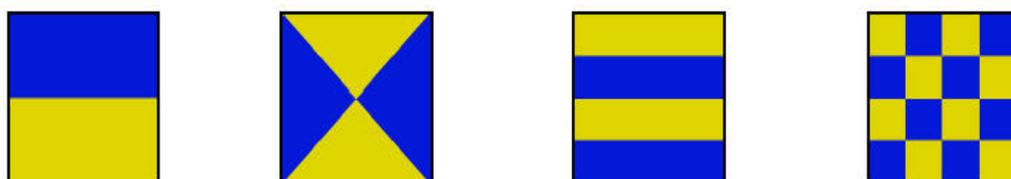


圖 3.2: 主色相同但顏色分布不同的各種圖形。

因此我們使用一個局部特徵來描述物品的區域性顏色資訊，使用簡單的方式來描述顏色與空間的相互關係，以便快速的擷取與瀏覽。在此我們採用的是 MPEG-7 所提供的的另一個描述子，稱為色彩配置(color layout)[14]。

色彩配置分為三個步驟：

1. 將單一個區塊切割成相同大小的四個區塊。
2. 找出每一個區塊中紅、藍、綠三個波段的主色。
3. 若已獲得足夠的空間與顏色資訊或區塊已太細小無法再切割，則停止動作，否則回到步驟一。

在實做上我們使用四元樹(quadratic tree)的資料結構來表示一個物品的色彩配置，比對時也是由每一個節點依序搜尋出每一小區塊的主色，來計算資料間的相似度。

色彩配置包含了許多優點，首先，他與影像解析度無關，即使物品可能因為攝影機的遠近或解析度而造成大小與資料庫中的物品有差異，但物品的顏色分布並不會因此而受到影響，所以色彩配置具尺度不變性(scale invariant)。第二，這是一個相當簡潔的描述子，將圖像之間的比對轉變成數字序列間的比對，因此具有相當高的搜尋效率以及較小的計算成本。由於圖片中物品並不大，因此在本系統中只切了兩層，共分成十六個區塊，而以下為我們對物品做色彩配置所得的部份結果：



圖 3.3:使用色彩配置所得的部份結果

## 3.2 輪廓(shape)

除了顏色之外，我們使用另一種特徵輪廓作為使用的特徵，由於物品顏色分佈也可能會很相像，如下圖：



圖 3.4: 顏色分佈相同但形狀不相同。

因此我們必須再找出另一種適合的特徵做比對以找出最後的結果。而我們使用 shape context[16,17]作為描述子。shape context 分為許多步驟，以下我們將詳細介紹每個步驟。

### 3.2.1 邊緣偵測

shape context 的步驟為，先找出物品外圍的輪廓，以從中取出能代表物品的參考點。因此在本演算法中，如何做好邊緣偵測是非常重要的，而這並不是件容易的工作，邊緣偵測有很多著名的演算法，如:Canny edge detection, Sobel edge detection, Prewitt edge detection...等，不同影像對於邊緣偵測有不同的需求，在本篇論文中，我們使用的是 Canny edge detection[18]，為的是能夠精確的找出影像中物品的輪廓作為辨識所使用的特徵。

Canny edge detection 是使用以下三點作為設計的基準：

- (1)良好的偵測能力：找出具灰階變化的邊線，要達此目的必須先去除雜訊，使訊雜比(Signal-to-Noise)增大。
- (2)良好的定位能力：希望真正灰階變化的位置和偵測找出的邊緣線間距離越進越好，距離小代表定位準確。
- (3)多重感應：一般邊緣線偵測器針對前兩項設計，但 Canny 認為應考慮單一邊緣產生多個感應的問題，這會造成同一個邊緣被視為多個邊緣的情況，因此 Canny 在此多加一個約制條件，即從單一邊緣線產生的多重感應中找出正確的邊緣線。

實作的方法分為四個步驟。首先，為了降低雜訊的影響，我們使用高斯

濾波器(Gaussian filter)與輸入的圖片做摺積(convolution)，公式如下：

$$S[i, j] = G[i, j; \sigma] * I[i, j] \quad (\text{公式 3.3})$$

$I[i, j]$  為原圖片在座標  $(i, j)$  上的像素值。 $\sigma$  代表了我們所選擇的高斯平滑係數，而高斯濾波器的定義如下：

$$G[i, j; \sigma] = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}} \quad (\text{公式 3.4})$$

在使用以上步驟消去影像中的雜訊之後，下一步驟我們利用以下公式，從影像中找出梯度大小及方向：

$$M[i, j] = \sqrt{P[i, j]^2 + Q[i, j]^2} \quad (\text{公式 3.5})$$

$$\theta[i, j] = \arctan(Q[i, j], P[i, j]) \quad (\text{公式 3.6})$$

其中  $P[i, j]$  與  $Q[i, j]$  分別為對  $X$  以及  $Y$  方向的偏微分。

但為了知道確實的邊的位置，我們還需依梯度方向對梯度大小作非最大值刪除(Nonmaximum Suppression)，即在梯度方向上邊緣點像素的大小值應該大於其臨近像素的大小值，所以只取區域(local)最大值為邊緣點。然後再實施一個附加遲滯性界定(Hysteresis Thresholding)的步驟，以刪除不正確的邊緣點，採用兩個臨界值，一個為高門檻值  $h T$ ，另一個為低門檻值  $l T$ 。任何一個像素的大小值只要大於  $h T$  則指定其為邊緣點，而連接此點的像素，只要其大小值大於  $l T$  也被指定為邊緣點，只要正確設定門檻值的大小，即可找出我們需要的邊。

在確實找出物品邊緣輪廓後，我們將所得的結果做取樣(sampling)，設定每隔數個點取一個參考點(control point)，希望儘可能的用最少的點數得到最好的效果，在時間以及辨識正確率中取得平衡，下圖顯示出剛才步驟的結果，由於是用取樣來代替物品的整個輪廓，因此即使我們在邊緣偵測時出了一些小差錯，對於結果並不會有太大的影響。

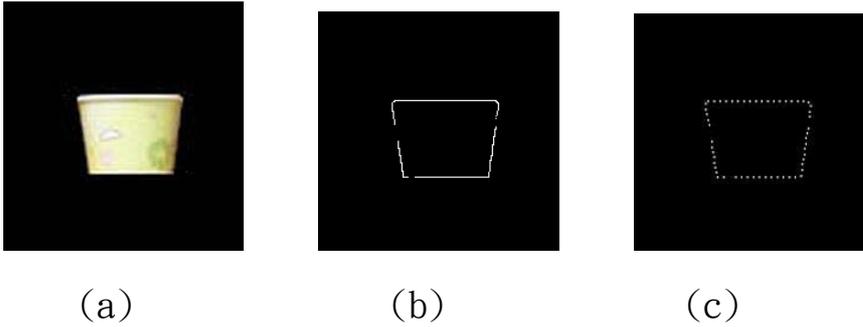


圖 3.5: 邊緣偵測的結果(a)原始圖片; (b)使用 Canny edge detector 所找出的邊; (c) 我們取的  $r$  個參考點。

### 3.3.2 log-polar histogram

邊緣偵測之後，我們使用 log-polar histogram 來表示出  $r$  個參考點之間的關係，為了獲得參考點之間的 log-polar histogram，我們將一個圓形的遮罩罩在每一個參考點上並計算出每個點之間的關係，而遮罩的形式由我們自己決定，在本系統中，我們將圓形三百六十度總共分成十二格，每一格涵蓋了三十度的範圍，並將圓的半徑取對數之後，平均的分為五格，其中遮罩的半徑由我們自行決定，如此即形成了下圖所看到的，一個總共具有六十格的遮罩：

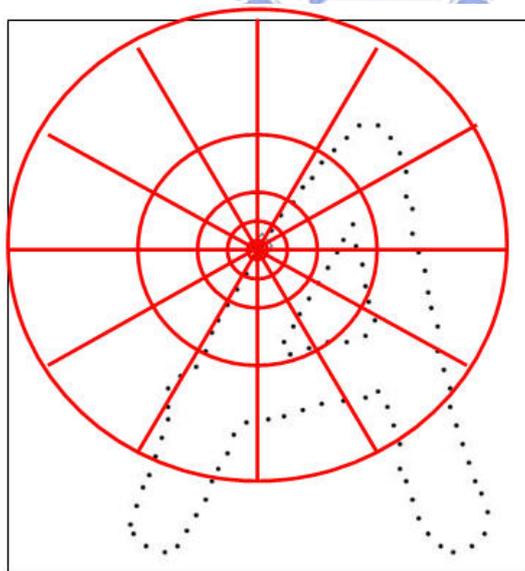


圖 3.6: log-polar mask 示意圖

而一個點的 log-polar histogram 我們可以表示如下，由圖中可看出，相對位置差不多的參考點可得到相近的直方圖：

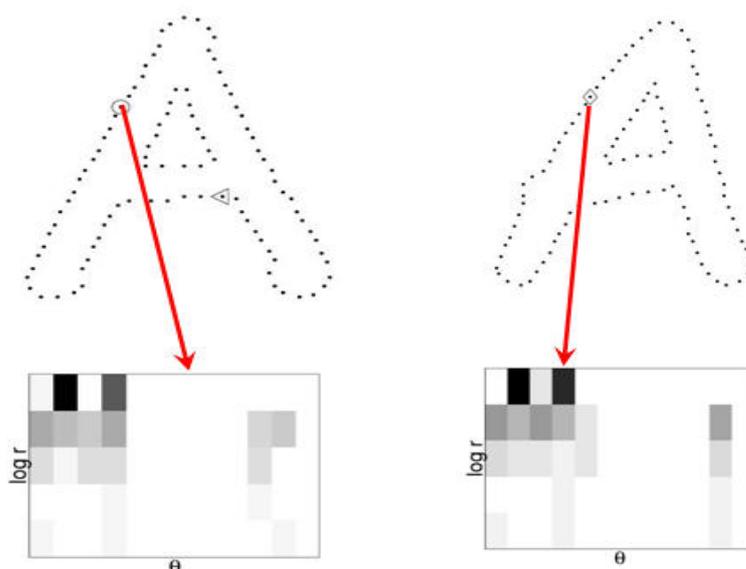


圖 3.7: 所得的 log-polar histogram 示意圖

### 3.3.3 角落偵測

找出  $r$  個參考點的 log-polar histogram 之後，由於精細辨識所需的時間相當長，為了改善後續的比對時間，我們想找出物品中的某些特徵點，先將這些特徵點的 log-polar histogram 與資料庫中剛才篩選過後的候選物品的特徵點作比對，假若所得的結果，兩者間有很大的差距，則認為此物品不可能是正確的結果，並從候選名單中刪除，若比對結果小於某個門檻值，則我們再對其餘的特徵點作進一步比對。

由於資料庫的物品中，多為矩形等有稜有角方方正正的物品，因此在此我們選擇的是角落作為預先比對的特徵點，角落在圖形辨識中是常常使用的特徵，與邊緣偵測一樣，已開發了許多不同的演算法，如拉普拉斯算子 (Laplacian operator)、KR' s method、Harris corner detector 等，通常依使用環境的不同選擇。由於資料庫中物品的形狀較簡單，因此使用 CSS(curvature scale space)[12] 的辦法，計算物品輪廓的曲率 (curvature)，假設在某一點的曲率大於某個門檻值，即認定為角落，但是此方法相當容易受到雜訊影響，因此在求取曲率之前，我們先對輪廓做高斯平滑，以改善區域最大值 (local maximum) 造成的問題，並參考特徵點附近的鄰居，以防止偵測出錯誤的角落。而求取曲率的公式如下：

$$k(t) = \frac{\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t)}{(\dot{x}(t)^2 + \dot{y}(t)^2)^{2/3}} \quad (\text{公式 3.7})$$

其中  $\dot{x}(t)$ 、 $\ddot{x}(t)$ 、 $\dot{y}(t)$ 、 $\ddot{y}(t)$  分別為對  $x$  以及  $y$  方向的一次及二次微分， $k(t)$  為所求的曲率。

下圖為我們對二值化圖運算得到的結果，藍色點即為所找出的角落：



圖 3.8: 角落偵測結果

找出角落之後，便可以開始做比對的動作，在此我們使用卡方檢定(chi square test)作為兩個點直方圖間的關連性計算方法，計算公式如下：

$$C_{ij} = \frac{1}{2} \sum_1^k \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)} \quad (\text{公式 3.8})$$

求出來之後，若兩個物品間角落的直方圖間差距太大，則從候選名單中捨去，否則進行下一個步驟。

### 3.3.4 匈牙利演算法

由於輪廓具有  $r$  個點，對兩個物品中每個點的直方圖依上一節所說的卡方檢定做比對，可以得到一個  $r \times r$  的矩陣，矩陣中之元素即為兩圖像中任意兩點參考點間之關連性。為了達到縮放不變性，我們選取  $r^2$  個值中的中位數(median)，並以此值對剛才所得的矩陣做正規化(normalize)，選擇中位數而不是平均數(mean)的原因在於，中位數較不受雜訊(outlier)的影響，因此較不會得到錯誤的數字使正規化的過程產生錯誤。我們將此問題視為尋找一個雙分圖間的最小成本配對，使用匈牙利演算法(Hungarian algorithm)來找出兩個物品  $r$  個參考點之間最適合的連結。

匈牙利演算法主要使用兩種性質，配對的矩陣同一行或同一列相減任一值後最佳解不變，以及矩陣中不相衝突的零組合就是最佳解。據此匈牙利演算法分為數個步驟：

1. 對矩陣做列運算，找出每一列的最小值，並將每一列的元素減去該列的最小值，以確保目前每一列至少有一個零。

2. 對每個零做標記，標記規則為，假設我們選擇了某個零，則與這個零同一行同一列的零就不能再選。
3. 若每一行每一列都有標記到零，則找到最佳解，零所在該位置的座標即為最佳配對的方式，結束此演算法。若沒有，則進行步驟四。
4. 把每一有兩個以上零的行標記出來，並標出這些行中包含零的列。
5. 將所有被標記的行以及沒被標記的列劃掉，並選出沒被劃掉的數字中最小的，每一沒被劃的橫列都減去此數，而每一被劃掉的直行都加上此數。
6. 回到步驟二。

找出  $r$  個點之間的最小配對後，我們即可由此計算出兩個物品輪廓之間的相似度。首先使用歐幾里得距離算出  $r$  對相配的点之間的距離，再將兩個輪廓調換，再重覆一次 shape context 的步驟，得到另一個結果，之後將兩個結果平均，最後的數值即為我們所要的相似度，若數值越大，則兩個物品的輪廓差距越大。所得最相近的物品即為查詢結果



## 第四章 實驗結果

在介紹過前置處理，以及我們所使用的特徵與描述之後，整個系統的架構已經浮現，本章將描述以上演算法的實驗方法及結果，並以實驗驗證本系統的可行性，並分析建構此系統所需的最小成本。

### 4.1 物品影像樣本的取得

由於本系統的主要目的為建構一智慧型的櫥櫃以達到自動偵測缺少物品並補貨的目的，所以我們也仿造一般家庭的環境，在實驗室內擺了櫃子以及攝影機作為資料的來源，以 Olympus C-8080 為攝影器材，設定影像解析度大小為 640\*480，65535 色全彩影像，取像設備及取像的角度皆固定，差別僅在於內部包含的物品不同，且物品擺放的角度也不同，模擬各種可能出現的情況，以求能確實的辨識出在資料庫中的物品。實驗環境如下圖所示：

為了測試本系統的可靠程度，我們建立了一個包含了 111 個物品的資料庫，且為了確定即使物品有橫向的旋轉，仍能清楚識別，我們每隔約二十度即擷取一張物品影像，如下圖所示：

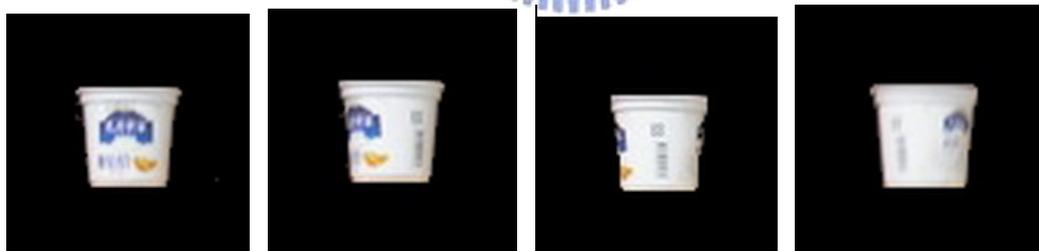


圖 4.1: 一個物品的不同角度圖片

而資料庫中包含的所有物品則如下圖：



實驗室的環境以及拍攝的圖片狀態，所得的一組較佳的門檻值。若希望系統在別的環境中作用，則必須在該環境中另取一組門檻值，否則可能無法達到預期效果。

#### 4.2.1 一般情況下的辨識結果

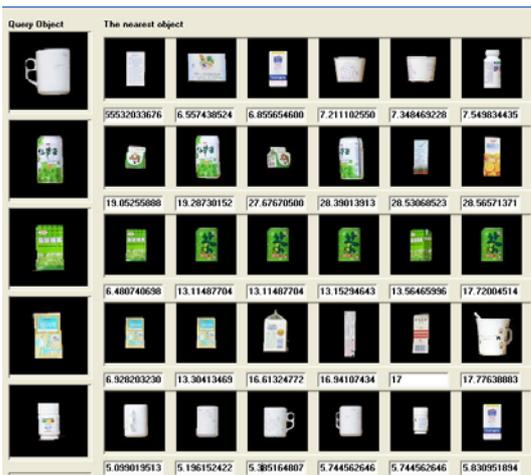
在實驗中，我們擺了五個物體在櫃子中，圖顯示了我們之前取得的背景，顯示了我們擺上五個物品後的櫃子圖像，圖顯示我們僅僅使用較粗糙的特徵(主色及顏色配置)得到的查詢結果，最左邊的圖像為我們所查詢的物品，而其餘的六個圖像代表資料庫中前六個最相似的物品，而由圖可知，除非物品的顏色分布相當特殊，否則使用粗糙特徵的結果並不好，但速度相當快。而圖為使用精細的特徵後的結果，雖然比對的速度較慢，但因之前粗糙的搜尋結果，已刪除了大部分資料庫中可能的物品，因此整體的速度仍然維持在可以忍受的範圍內。

本系統基礎的介面以及辨識結果則如下圖所示：



(a)

(b)



(c)



(d)

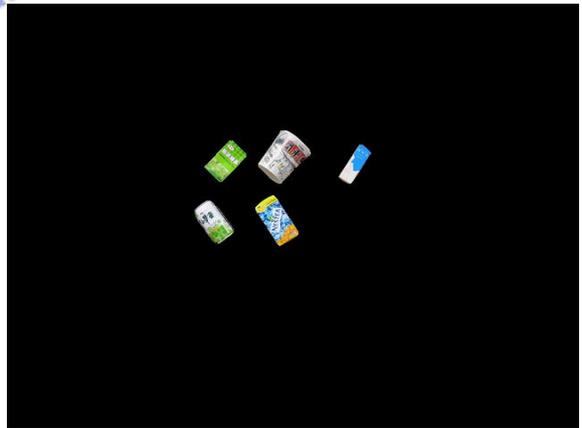
圖 4.3: 普通情況的辨識結果(a)我們建立的背景模型。(b)在櫃中擺了五個物品。(c)只使用顏色得到的結果。(d)再加上輪廓做辨識後得到的最後結果。

由圖(c)中可以明顯看出，第一步已篩選出資料庫中顏色分佈相近的物品，若顏色較特殊，則有可能直接找出正確的物品，並減輕之後步驟對系統執行時間造成的負擔。

以下是將物品隨便擺放，使用主軸校正後所得到的辨識結果：



(a)



(b)

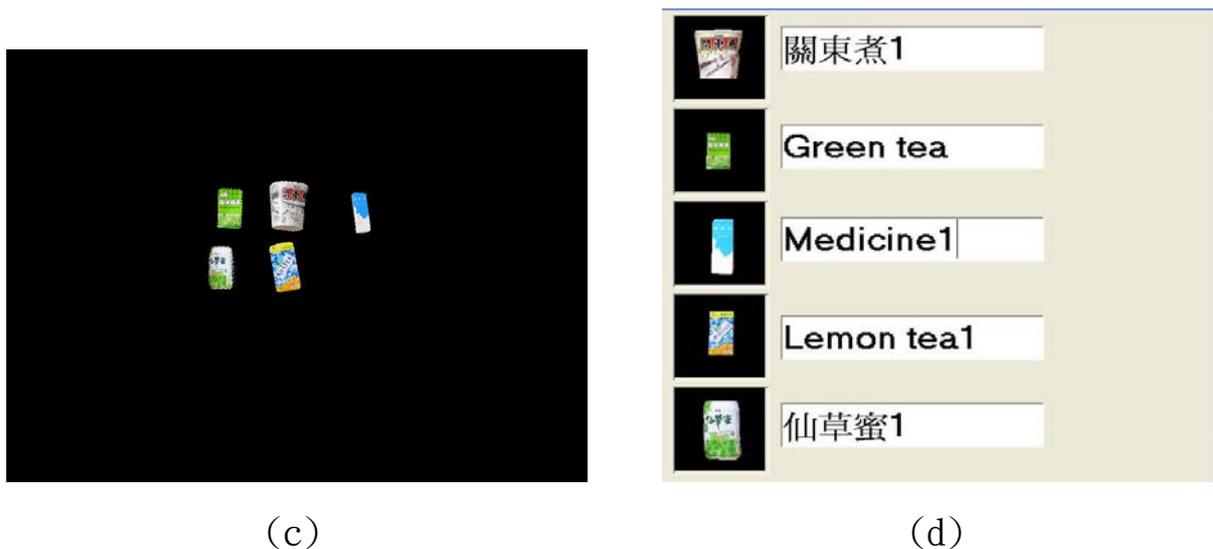


圖 4.4: 傾斜情況的辨識結果 (a)櫃子擺放五個物品的圖片。(b)背景剷除後的結果。(c)使用主軸校正後的結果。(d)最後辨識結果。

#### 4.2.2 解析度與精確度之間的關係

我們使用軟體降低取得圖片的解析度，以達到與使用較差攝影機相同的效果，並對其再做一次辨識，取得在各個解析度中辨識的正確率，正確率的算法為對資料庫中每一物品做辨識，最後將辨識成功的物品總數與資料庫中的物品總數相除而得。由於系統強調辨識率，而實驗在圖片解析度大小為 640\*480 時能夠識別出資料庫中的所有物品，數據令人滿意，因此我們以此為基準，考慮解析度與正確率間的相互關係。

以下列出在有和沒有使用內插法改善圖片品質的兩種情況下，系統辨識所得到的初步結果：

圖片解析度大小	櫃子中一格所佔 圖片大小	正常圖片辨識失 敗的物品數	圖片品質改善辨 識失敗的物品數
640 * 480	57 * 57	0	0
560 * 420	51 * 51	6	5
480 * 360	45 * 45	13	12
400 * 300	38 * 38	31	24
320 * 240	30 * 30	54	41
240 * 180	23 * 23	90	85

圖 4.5: 各個解析度中辨識失敗的物品數

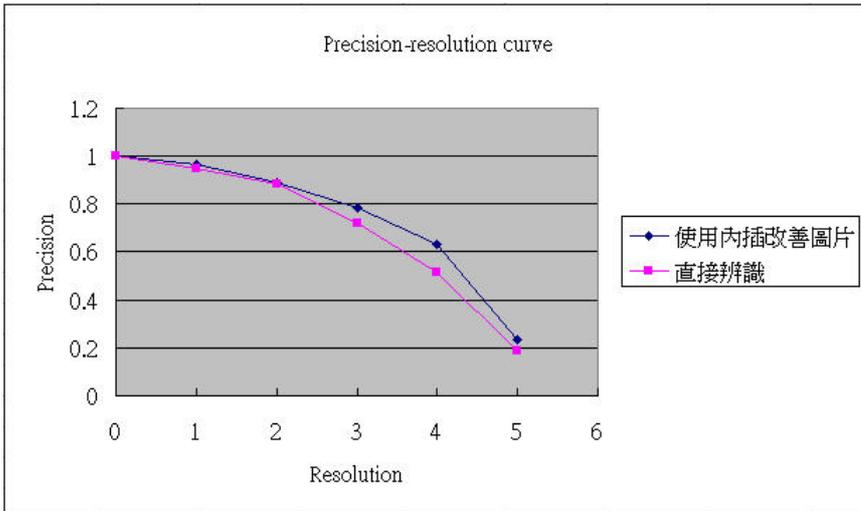


圖 4.6: 辨識率與解析度曲線圖

由圖表中可看出，使用內插法後對結果的確有些微的改善，在解析度不高也不低時會產生較明顯的效果，但以實際應用面來看，由於居家用品必須百分之百的辨識正確，以防廠商送錯貨品造成資源的浪費。為了改善此問題，我們嘗試使用雙線性內插法(bilinear interpolation)，將像素值依比例設定為最鄰近四個像素值的結合，以得到更為精緻的圖像，但此方法對顏色上有較好的效果，卻沒有對物品形狀上的問題有更明顯的改善，反而增加了所需的時間。因此若沒有使用其他能更加改善圖像品質的方法，則我們所開發的演算法目前只能作用在 640\*480 的解析度下，否則可能因為辨識錯誤而造成嚴重的後果。

#### 4.2.3 所需時間

由於本系統以正確率為主，因此採用 shape context 作為描述子，但所需時間就相對變長，無法運用在即時(real-time)系統上。依照物品性質的不同，辨識所需的時間也將產生極大的差異，因此以下列舉出幾個較具代表性的例子，並分析此系統對各個物品作用所需的時間。以及我們使用特徵點先比對做加速後所得成果。此結果受到資料庫的影響相當大，因此若資料庫中擺放的物品不同，可能會得到不同的結果。

查詢物品	所需時間(未加速)	所需時間(加速)
	17.659s	7.320s
	20.568s	8.926s
	16.329s	13.624s
	13.588s	9.257s
	6.935s	4.257s
	2.187s	1.562s
	2.565s	2.565

圖 4.7:物品辨識所需時間以及加速後的成果

從前四個物品可知，由於資料庫中淺色系物品相當多，因此當物品為白色或淡色時，所需時間即明顯偏高，若僅僅使用顏色做比對，可能的候選者通常都達到一百五十個以上，蘆筍汁的顏色分佈稍為特殊，也減少了辨識所需的時間，而最後蜜餞以及筆盒的顏色極為特殊，因此使用顏色即可刪除資料庫中大部分的物品，僅需兩秒左右即可完成辨識。

以上結果是在沒有使用加速的情況下測試的，而在資料庫中物品的限制下，辨識淡色物品所需時間相當長，假設我們先使用特徵點辨識，再進一步的篩選顏色比對後所得到的候選物品。則由上圖的結果，可以明顯的看出辨識時間依物品的不同而有不同的改善，由於前兩個物品的形狀特殊，因此我們得到的特徵點也與其他物品有相當大的差距，可刪除大部分的候選者，相當程度減輕了精細比對的負擔。但是同為白色的物體，第三和第四個物品減輕的幅度就不如之前的多，因為形狀太過簡單，只使用特徵點

無法具體表現出與其它白色物品之間的差異，刪除的反而是資料庫中形狀較特殊的物品，最後兩個物品其實也是相同的結果，但顏色已刪除大部分物品，因此加速後的結果改善反而並不是那麼的成功，尤其是筆盒，在辨識時間上完全沒有改善。由此也可知道，精細比對佔據了整個系統大部分的時間，若想再降低系統所需時間，或許可從使用更好的加速辦法，或從再增加簡單的特徵以刪除資料庫中更多的候選物品中著手。



## 第五章 結論及未來工作

### 5.1 結論

隨著時代的演進，台灣人口也將慢慢到達零成長的情況，老年人口的增多，青壯人口的減少，在在迫使現代國家戮力研究如何藉由結合資訊、電子、通訊等領域來建構「智慧型居家照顧系統」。

本文的目的，是要探討如何運用影像處理以及圖形識別等相關技術，來發展一套適合居家環境的物品辨識系統，以期能在生活用品不足時，可由廠商自動補充，藉此便利居家老人的生活，且相較於 RFID 等較不普及的產品，或條碼這種需要使用者介入較深的系統，我們使用便宜的照相機來建構一個智慧型的櫥櫃，以降低成本且節省人力。

本文提出的系統中，先利用背景削除來定位圖片中物品的位置，這個方法主要的概念是以數學形態學以及消除陰影以去除圖片中背景的雜訊。找出物品後，我們先求出物品的主軸，看擺放是否傾斜。若傾斜，則使用仿射轉換(affine transform)將其位置校正，去除物品旋轉的影響。且此時根據輸入圖片的解析度大小，將圖片做內插以改善低解析度時的物品圖像品質。

順利找出物品資訊後，為了能確實且快速的辨識出櫥櫃中的物品。我們使用了階層式的架構，結合兩種特徵，使用粗糙到精細的搜尋方法，依序使用了主色，顏色配置，以及 shape context 三種描述子，一步步的剔除資料庫中可能的候選者，最後找出最符合的物品。

最後由實驗驗證了這個智慧型櫥櫃系統的可行性，擁有可以辨認櫃中物品的能力，並且在影像解析度較高時，有著高正確辨識率。雖然隨著解析度降低，辨識率也緩慢下降，但希望能以此為居家智慧型櫥櫃建立基礎。也希望有更多的研究能投入在居家照顧方面的研發工作，提出更新穎的構想，讓大眾能早日享受到更直覺、更便利、以及更人性化的系統。

### 5.2 未來工作

本系統以辨識率作為主要的訴求，因此使用 shape context，利用大維度的特徵來描述物品的輪廓，因此在辨識率上也達到了一定的水準，但

shape context 為一複雜度為  $O(n^3)$  的演算法，在速度上並不令人滿意，雖然我們進一步的使用一些特徵點的比對來刪除形狀差距太大的物品，但除非物品形狀太特殊，否則所能達到的功效有限，且相似度的計算必須將兩個物品重複比對兩次，在資料庫中有數百張圖片，且恰好物品顏色大部分相近的情況下，會造成時間上的延遲，因此需要想辦法降低此演算法的複雜度，改善執行的時間，或使用更簡潔的描述子，但同時能保持相同的辨識率。加快處理時間，這是相當必要的。

另一方面，實驗室環境的光源相當穩定，但在一般家庭中，可能會有開一個燈，兩個燈，或完全黑暗的情況，雖然系統有對影像的灰階資訊做正規化，但只能免除小幅度的燈光改變，如果要實際應用在日常生活中，則對可能的光源強度改變還是需要有較好的改善方法。或是使用更好的建立背景的方法。

由於擺在櫃中的物品可能不只一個或不只一種，如果擺放的位置有些不好，則會造成物品之間互相遮蔽(occlusion)的現象。造成做完背景消除後，所得結果並不是單一物品的資訊，而是多個物品重疊在一起的影像，造成我們無法順利取得顏色及輪廓的資訊，進而造成辨識錯誤，但若加此限制，並不符合一般家庭平常的習慣，造成使用上的不方便，這也是以後需要考慮進去的因素。

## 参考文献 (Reference)

- [1] T.R.Crimmins, " A complete set of Fourier descriptor," IEEE Transactions on systems, Man, and Cybernetics, Vol.12, pp.236-258, 1982
- [2] S.Mahmoud, " Arabic character recognition using Fourier descriptors and character contour encoding," Pattern Recognition, Vol. 27, No. 1, pp. 815-524, 1994
- [3] F.Mokhtarian and A.Mackworth, " Scale-based description and recognition of planar curves and two-dimensional shapes," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.8, No. 1, pp. 34-43, 1986
- [4] Hu. M. K. , " Visual pattern recognition by moment invariants," IRE Transactions on information theory, vol.8, pp.179-187, 1962
- [5]W.K.Kim and Y.S.Kim, " A New Region-based Shape Descriptor," ISO/IEC MPEG99/M5472, Maui, Hawaii, December 1999
- [6] S. J. Park, D. K. Park, C. S. Won, " Core experiments on MPEG-7 edge histogram descriptor," MPEG document M5984, Geneva, May, 2000
- [7] B. S. Manjunath, P. Salembier, T. Sikora, Introduction to MPEG-7, Jone Wiley and Sons, Ltd., San Francisco, 2002
- [8] J. Huang, et al, " Image indexing using color correlograms," IEEE Conference On Computer Vision and Pattern Recognition, pp. 762-768, 1997
- [9] Y. Ke and R. Sukthankar, " PCA-SIFT: A More Distinctive Representation for Local Image Descriptors," IEEE Conference On Computer Vision and Pattern Recognition, pp. 511-517, 2004
- [10] T. Gevers and A. W. M. Smeulders, " Image indexing using composite color and shape invariant features," Proceeding of International Conference Computer Vision, Bombay, India, pp. 576-581, 1998
- [11] B. S. Manjunath, et al, " Color and Texture Descriptors," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, pp703-715, No. 6, June 2001

- [12] F. Mokhtarian, and R. Suomela, " Robust Image Corner Detection through Curvature Scale Space," IEEE Transaction Pattern Analysis and Machine Intelligence, vol. 20, number 12, pp. 1376–1381, 1998
- [13] A. Prati, et al, " Shadow detection algorithms for traffic flow analysis: a comparative study," IEEE International Conference on Intelligent Transportation Systems, Oakland, California, pp. 340–345, August 2001
- [14] E. Kasutani, A. Yamada, " The MPEG-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval," Proceeding of International Conference on Image Processing, vol. I, pp. 674–677, October 2001
- [15] R. C. Gonzales and R. W. Woods, Digital Image Processing, Addison–Wesley Publishing Company, 1992
- [16] S. Belongie, J. Malik, J. Puzicha, " Shape matching and object recognition using shape contexts," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 4, pp. 509–522. April 2002
- [17] G. Mori, S. Belongie, J. Malik, " Efficient Shape Matching Using Shape Contexts," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 11, pp. 1832–1837. November 2005
- [18] Canny, " A computational approach edge detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, pp. 679–698, November 1986
- [19] K. Mikolajczyk and C. Schmid, " A performance evaluation of local descriptors," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 10, October, 2005