

國立交通大學

工學院精密與自動化工程學程

碩士論文

機械視覺之加工補正研究

A study of manufacture compensation with machine vision

研究生：鐘正宜

指導教授：徐瑞坤 教授

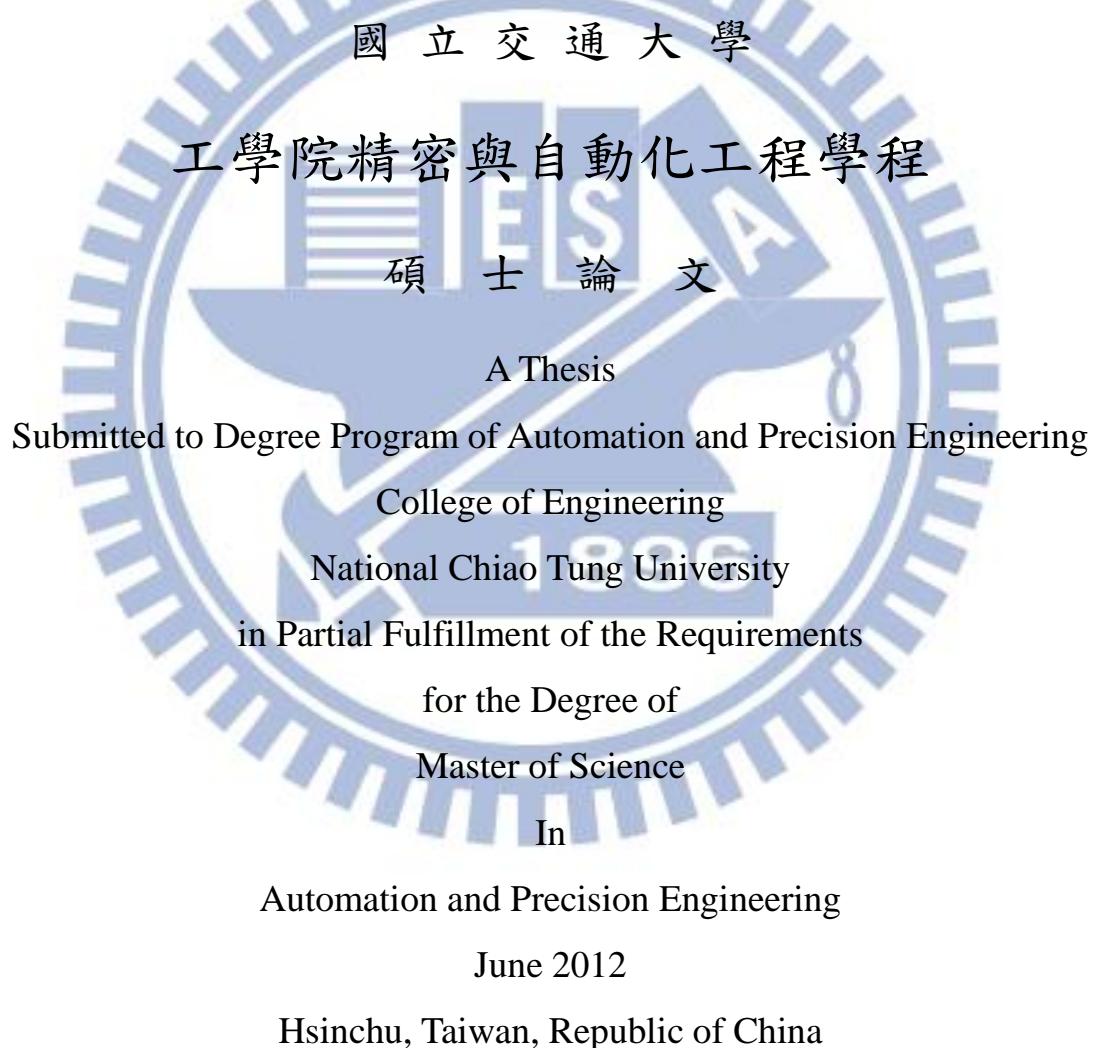
中華民國一百零一年六月

機械視覺之加工補正研究

A study of manufacture compensation with machine vision

研究生: 鐘正宜 Student : Cheng-Yi Chung

指導教授: 徐瑞坤 Advisor : Ray-Quan Hsu



中華民國 101 年 6 月

機 械 視 覺 之 加 工 補 正 研 究

學生：鐘正宜

指導教授：徐瑞坤

國立交通大學工學院精密與自動化工程學程

摘 要

本研究的目的是使用自由軟體，開發出在數控加工機上，運用自由軟體 OpenCV 機械視覺(Machine Vision)函式庫、dxfllib DXF 轉換函式庫與 OpenGL 繪圖函式庫，以商用開發工具 Visual Studio C++ MFC 為平台，加上 Dalsa Sepera LT CCD 開發函式庫，新代控制器 OCAPI 整合開發介面，整合撰寫出一套完整的加工補正系統。

數位噴印的產品製程中，產品特性是輕薄、印出的內容會有小變形，若要將印製的圖形以無治具的方式切下，傳統的 NC 加工的補正方式是達不到的，因為尋找加工基準將極為困難而無效率，原因就是工件極薄，上料後的位置，前後無法一致，若又遇到需將工件沿變形的輪廓切下，傳統的加工補正方式幾乎一籌莫展。

若以機械視覺做為加工補正的工具，可以達到無治具、小變形之加工補正。利用工件輪廓邊緣，印上定位點，做為工件位置定位與輪廓變形的參考依據。前述定位點以機械視覺辨識出位置後，計算出位置與加工路徑補正，加入圖形座標中。之後圖形座標轉出成為 NC 程式碼，直接透過新代控制器 OCAPI 函式庫，循序發出加工單節指令至控制器中進行加工。

關鍵字：自由軟體、機械視覺、治具、補正值

A study of manufacture compensation with machine vision

Student : Cheng-Yi Chung

Advisor : Ray-Quan Hsu

Degree program of Automation and Precision Engineering

College of Engineering

National Chiao Tung University

ABSTRACT

The study used freeware to integrate and write a comprehensive processing/compensation system for numerical control machines that employs freeware OpenCV machine vision library, dxflib DXF conversion library, and OpenGL graphic library. It also applies the commercial development tool Visual Studio C++ MFC as the platform, as well as the Dalsa Sepera LT CCD development library and the Syntec controller OCAPI for integrating and developing the interface.

During the digital print product manufacturing process, products have the characteristic of lightness and thinness and the printed content shows slight transformation. The compensation method of traditional NC processing cannot be used to cut printed images without gauges. Because of the extremely thin work pieces, and inconsistent positions after materials are loaded, identifying the processing datum is difficult and inefficient. We also cannot use the traditional processing/compensation method to cut work pieces along the outlines of distortions.

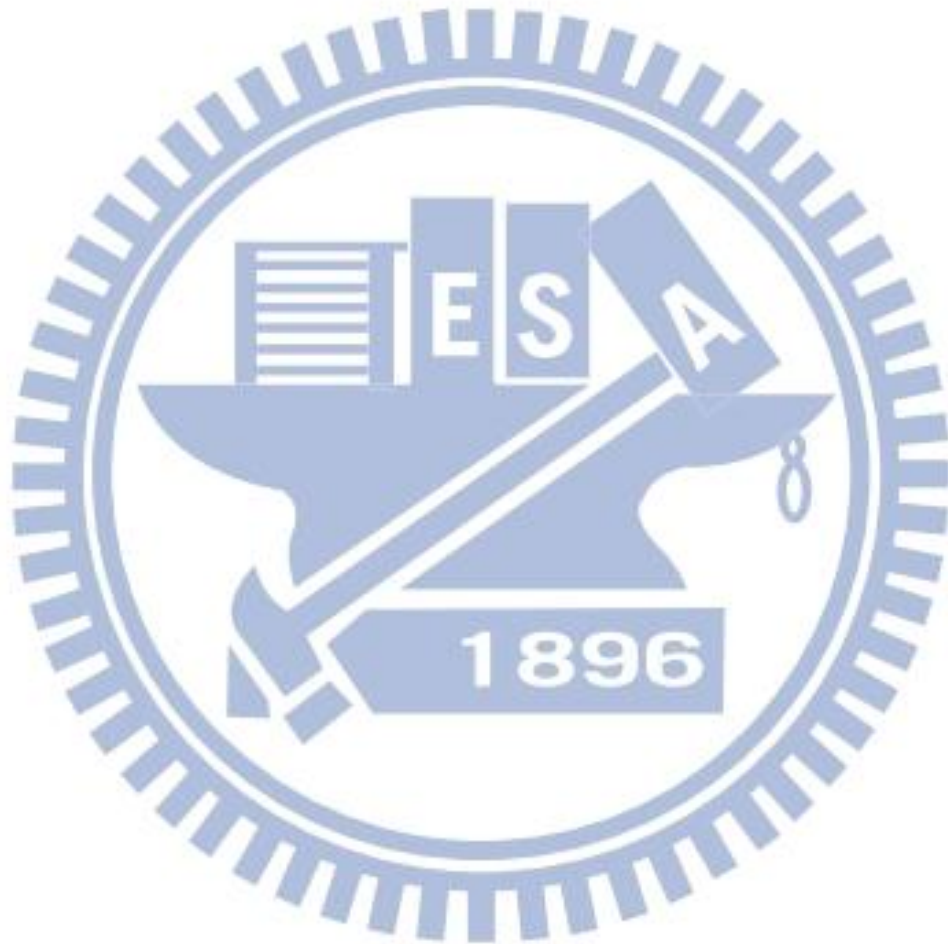
Machine vision can be used as a tool for processing and compensating to achieve slight distortion and no jig and compensation. We marked anchor points along the outline of the work pieces as a reference for their positioning and outline distortion. After the positions of the anchor points were identified using machine vision, the positions and processing path compensations were calculated and were added to graphics coordinates. Then, the graphics coordinates were converted into NC codes. Finally, we used the Syntec controller OCAPI library to send processing single-block commands to controllers in sequence for processing.

Keyword: freeware, Machine Vision, jig, compensation

致謝

本論文研究能夠完成要感謝徐瑞坤教授在論文研究過程中，給予研究方法的指導，以及華勝自動化股份有限公司陳長明先生與賴春億先生在控制器軟硬體上的支援。

最後當然也要謝謝內人素華在這段時間的支持，以及家父、家母不斷的鼓勵，讓我可以堅持到最後。



目錄

	頁次
中文摘要.....	ii
英文摘要.....	iii
致謝.....	iv
目錄.....	v
圖目錄.....	vii
表目錄.....	ix
第一章 緒論.....	1
1.1 研究背景.....	1
1.2 研究動機與目的.....	2
第二章 文獻回顧.....	4
2.1 名詞釋義.....	4
2.1.1 機械視覺(Machine Vision).....	4
2.1.2 圖素(Pixel).....	4
2.1.3 邊緣偵測(edge detect).....	4
2.1.4 模式比對(Pattern Match).....	4
2.1.5 FOV(Field Of View).....	5
2.2 機械視覺原理.....	5
2.2.1 影像檔格式(Image Format).....	6
2.2.2 點陣資料(Raster Image)格式與取樣(Sampling).....	9
2.2.3 影像過濾(Image Filtering).....	10
2.2.4 臨界值(Threshold).....	10
2.2.5 模式比對(Pattern Match).....	11
2.3 機械加工補正.....	13
2.3.1 工件置放補正.....	13
2.3.2 加工路徑補正.....	14
第三章 系統架構規劃與方法.....	18
3.1 硬體架構.....	18
3.1.1 選擇相機.....	18
3.1.2 選擇相機的影像擷取介面.....	18

3.1.3	機械視覺的光學與電腦系統.....	20
3.2	軟體架構.....	21
3.2.1	基於 MFC 之 SDI 介面.....	24
3.2.2	OpenGL.....	25
3.2.3	DXF.....	26
3.2.4	OpenCV.....	28
3.2.5	控制器與控制器 API.....	30
第四章	系統實作.....	31
4.1	NC 機器與視覺硬體.....	31
4.2	數位印刷與後製流程.....	31
4.3	軟體系統.....	34
4.3.1	DXF 輸入與資料庫建立.....	34
4.3.2	資料儲存與讀取.....	36
4.3.3	OpenGL 繪圖.....	38
4.3.4	CCD 取像與 OpenCV 定位點辨識.....	40
4.3.5	控制器 SDK.....	40
4.3.6	刀具與加工參數設定.....	41
4.3.7	NC Code 產生器.....	41
4.4	圖層管理.....	42
4.4	研究結果比較.....	42
第五章	研究結論.....	47
5.1	研究結論.....	47
5.2	未來的計畫.....	48
	參考文獻.....	49

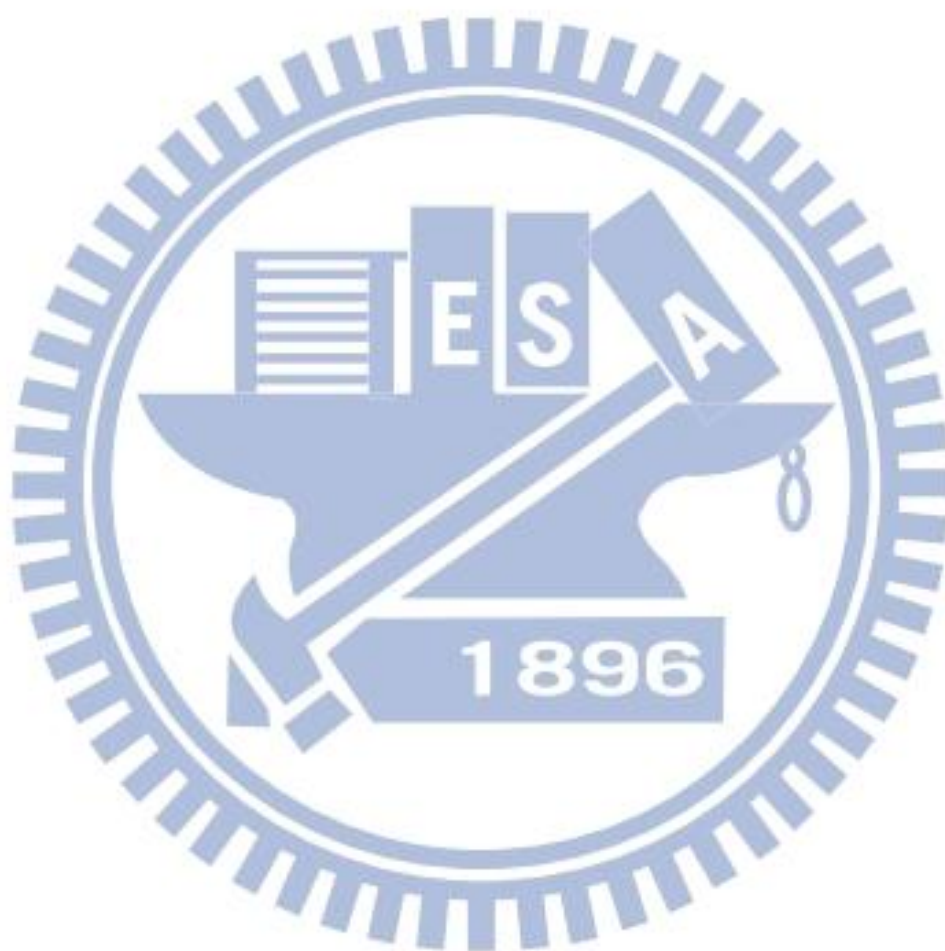
圖目錄

圖 1.1 應用範例.....	1
圖 2-1 邊緣偵測.....	4
圖 2-2 圓形偵測.....	4
圖 2-3 模式比對.....	5
圖 2-4 械視覺加工補正系統圖.....	6
圖 2-5 影像光強度方程式.....	6
圖 2-6 記憶體影像光強度.....	10
圖 2-7 影像過濾.....	10
圖 2-8 影像臨界值.....	10
圖 2-9 臨界值影像過濾.....	11
圖 2-10 模式比對.....	12
圖 2-11 工件置放補正.....	14
圖 2-12 原始未變形圖形.....	15
圖 2-13 旋轉變形與補正.....	15
圖 2-14 線性變形與補正.....	15
圖 2-15 扭轉變形與補正.....	15
圖 2-16 非線性變形與補正.....	15
圖 2-17 定位點的佈置.....	16
圖 3-1 Dalsa M640 相機.....	18
圖 3-2 影像資料擷取介面.....	19
圖 3-3 光學系統與電腦系統關係圖.....	21
圖 3-4 軟體架構圖.....	21
圖 3-5 機械視覺與加工補正系統流程圖.....	23
圖 3-6 MFC SDI 結構圖.....	24
圖 3-7 系統功能圖.....	25
圖 3-8 OpenGL 座標.....	26
圖 3-9 dxflib 函式架構.....	27
圖 3-10 OpenCV 基本架構.....	28
圖 4-1 系統總成簡圖.....	31
圖 4-2 數位印刷與後製流程.....	32
圖 4-3 無定位點的圖形.....	33

圖 4-4 具定位點的圖形.....	33
圖 4-5 B-Spline 的描繪.....	33
圖 4-6 OpenGL 2D 顯示.....	38
圖 4-7 $R>W/H$	39
圖 4-8 $R<W/H$	39
圖 4-9 CCD 取像與樣本比對.....	40
圖 4-10 移動控制介面.....	40
圖 4-11 刀具參數設定.....	41
圖 4-12 輸出之 NC Code.....	42
圖 4-13 圖層管理.....	42
圖 4-14 模式比對與邊緣偵測.....	43
圖 4-15 光線干擾下比對樣本圖檔.....	43
圖 4-16 光線干擾下模式比對結果.....	43
圖 4-17 光線干擾下邊緣偵測尋圓結果.....	44
圖 4-18 背景干擾下樣本圖檔.....	45
圖 4-19 背景干擾下樣本比對結果.....	45
圖 4-20 背景干擾下邊緣偵測尋圓結果.....	46

表目錄

表 2-1 影像與資料型態列表.....7



第一章 緒論

1.1 研究背景

距離 1950 年代 CNC(Computer Numerical Control)電腦數控機械問世已經超過一甲子。隨著電腦科學的日益精進，CNC 的應用層面也愈來愈廣泛。

在傳統 NC 加工中，加工路徑補正對於廣告數位印刷媒體及 3C 產品中網印印刷製品，在製程中的切割需求，傳統的切削補正無法達到製程上的需求。因為補正的需求須隨印製的變形與飄移而變化。此外由於產品的切割成型過程中，因為此類工件都十分輕薄，通常治具只是很間單的透氣背板，利用真空吸著，使工件附著於加工檯面上。所以定位的方法與傳統 NC 加工也不一樣。傳統 NC 加工利用尋邊器或者利用加工銑刀取得加工座標原點。而本研究的加工工件利用這兩種方法皆十分困難去取的加工程式原點。



圖 1.1(a)印刷圖型



圖 1.1(b)印刷圖型



圖 1.1(c)鍵盤

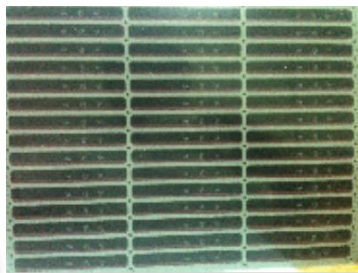


圖 1.1(d)筆電狀態燈工件



圖 1.1(e)筆電狀態燈(放大)

圖 1.1 應用範例

如圖 1.1(a)、(b)所示，為一個數位印刷的廣告圖形。由於數位印刷是非接觸式印刷，圖形會有一些變形，而且工件的定位，隨著手動放置，置放位置每一次都會有一些差異。而根據廣告心理學研究的結論，切除留白的圖形會比一張四四方方的看板具有較高的吸引力。為了這個目的當不使用機器切割時，往往使用人手利用刀片切除留白，通常曠日廢時，而且切割的品質也很不穩定。而圖 1-1(c)、1-1(d)、1-1(e) 為光學壓克力網版印刷的工件，因為網印的特性，會產生飄移與變形。為解決這個問題，若使用傳統 PCB 版的成形的工法，需在每一片的定位點上打 pin，一次一片加工，效率極差。

機械視覺 (Machine Vision) 的應用隨著 PC 的硬體性能大幅提升，單價大幅下降。逐漸在工業上被大量使用。但大多數是在產品的檢測應用上。

商用機械視覺軟體函式庫，較有名的如 Labview/Vision、Matrox MIL、Dalsa Sepera。應用機械視覺的功能，去尋找圖形位置的變化，作為補正的依據，讓我們可以跳脫傳統的補正方式，輕易的讓工件因印刷產生的變形，能依其輪廓的需求精確的切削出來。

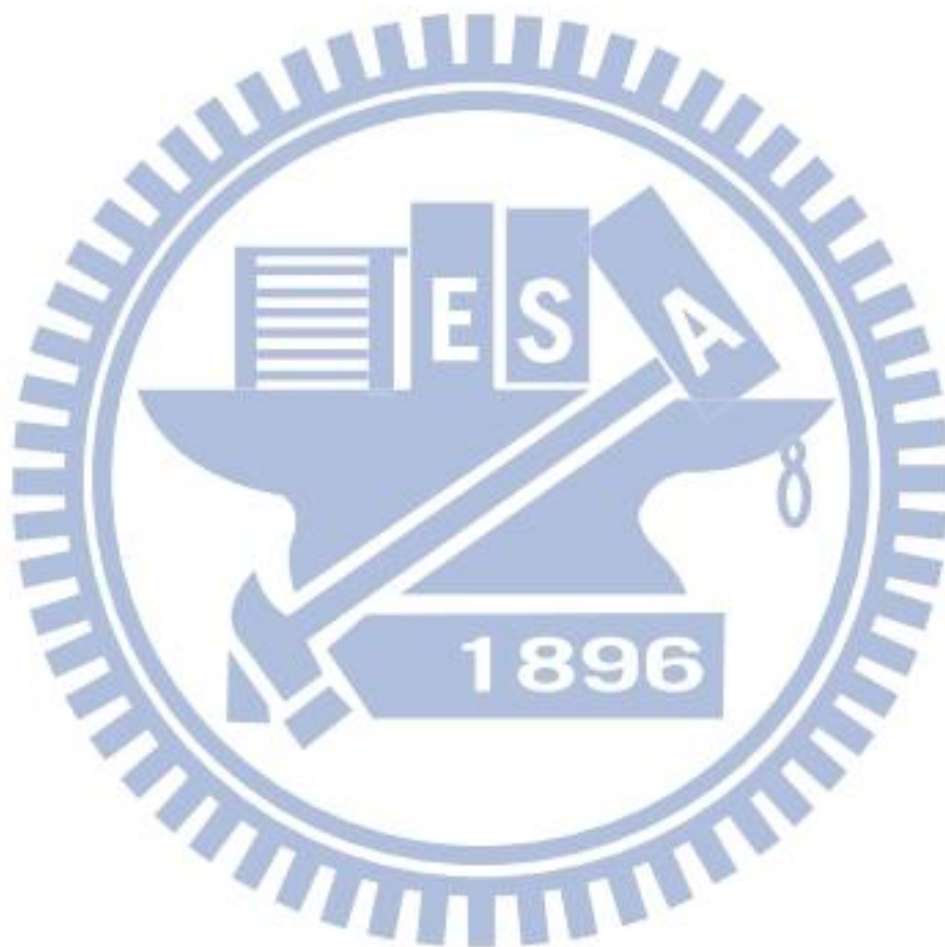
1.2 研究動機與目的

本研究題目的訂定是作者與工作中發現目前相關應用的 CNC 機台僅有少數國外的公司生產。而且結合機械視覺 (Machine Vision) 功能的設備價格十分昂貴，單單視覺軟硬體成本動輒近萬美元。若使用商用的機械視覺程式庫，邊際成本亦十分高昂。

本研究希望藉由自由軟體 OpenCV 與數控控制器間的整合，得到一個下述目的的機械視覺加工補正解決方案。

- 低成本：使用開放源碼的資源來開發系統。使用的 OpenCV、OpenGL 與 dxflib 均為自由軟體 (freeware)，可以使應用的邊際成本趨近於零。
- 跨平台：以上所用的自由軟體，均為跨平台的函式庫，可以輕易地移植到如 Linux、WinCE... 等作業系統平台上，未來可做不同的整合應用。
- 目標物多樣：利用模式比對 (Pattern Match)，使比對的目標物不一定是圓形。
- 抗干擾：模式比對演算法具較佳之抗干擾性。

- 易操作：模式比對演算法只需目標進入 FOV(Field Of View)即可進行比對。
- 速度快：快速的比對性能。
- 精度高：使用影像深度(image depth)為 32 bit float，提高比對的精度。



第二章 文獻回顧

2.1 名詞釋義

針對本研究中幾個關鍵的名詞，機械視覺、圖素、邊緣偵測、模式比對與 FOV 於本小節中加以說明如下：

2.1.1 機械視覺(Machine Vision)

機械視覺，利用攝影機和電腦來擷取、分析及解釋影像內容進而下達某些決策。也就是讓電腦有能力去”看”得到東西。

2.1.2 圖素(Pixels)

又稱像素，圖像顯示的最基本單位。如影像檔中可解析最小的點，或者是數位相機的影像感測器陣列的一個元素(element)。

2.1.3 邊緣偵測(edge detect)

邊緣偵測，依循影像中定義的路徑計算出可能的邊緣位置。邊緣偵測又可以區分為一般邊緣偵測及圓形偵測。如圖 2-1 為一般邊緣偵測，圖 2-2 為圓形偵測。

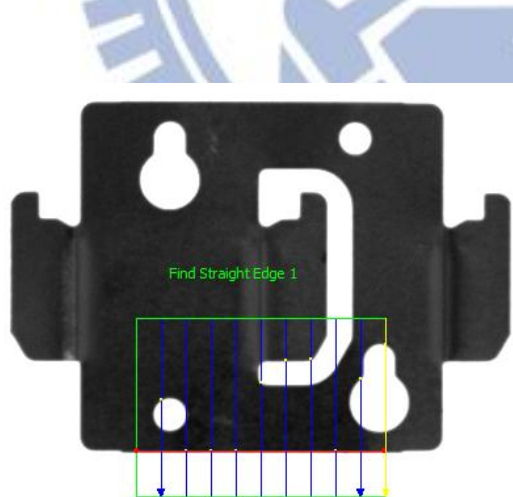


圖 2-1 邊緣偵測

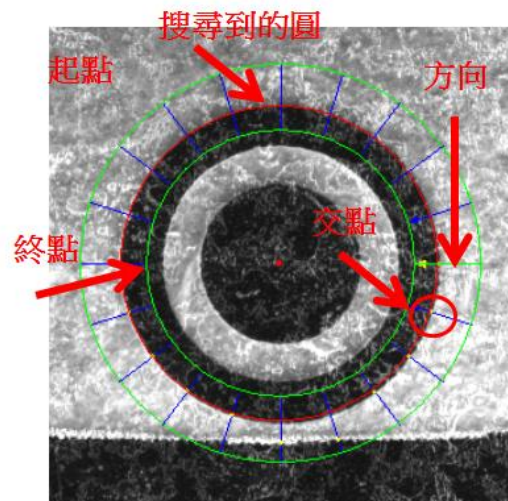


圖 2-2 圓形偵測

2.1.4 模式比對(Pattern Match)

利用樣本檔(Template File)，在影像檔中標定樣本檔的位置。如圖 2-3 所示，三個樣本檔在影像中標定出相同影像的位置。

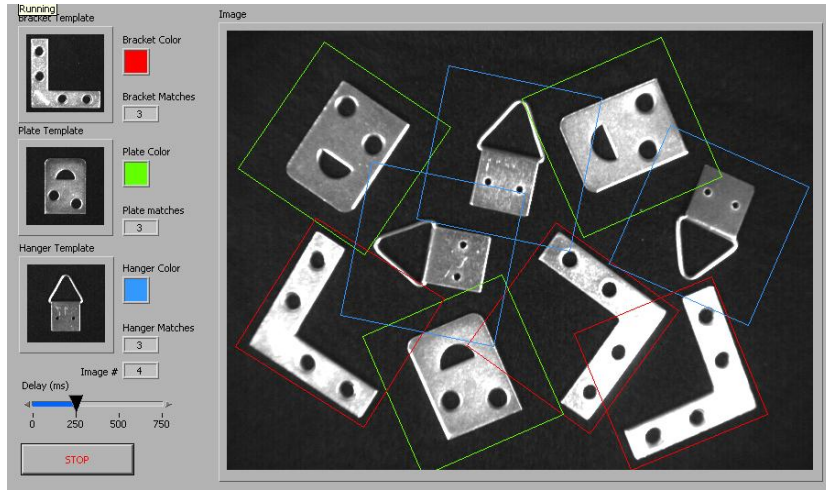


圖 2-3 模式比對

2.1.5 FOV(Field Of View)

數位相機中的可視範圍。FOV 的計算公式如下

$$FOV = \frac{\text{Camera Sensor Size}}{\text{鏡頭光學倍率}} \dots\dots\dots (1)$$

2.2 機械視覺原理

機械視覺系統的應用，整合了光學、機械、電路與計算機科學。而應用的領域非常廣泛，目前大部分集中在自動化檢測機械例如PCB焊道檢測、機械零件形狀檢測、熱軋鋼板尺寸檢測、木材瑕疵檢測…等。新的應用應用也不斷地推陳出新。

整個機械視覺加工補正的核心原理，架構在相機(CCD)的取像與取像後的影像處理與辨識。

機械視覺加工補正系統如下圖 2-4 所示，工件置於機台上，經由鏡頭取像，藉由 CCD 將光學影像資料轉成數位影像資料，而取得對位補正資料。而 CCD 介面將數位影像資料傳至電腦，在電腦中以機械視覺軟體計算對位資料，並判斷對位的結果及進行決策判斷。最終計算的結果透過 CNC 控制器與電腦間的資料介面傳送到 CNC 控制器。CNC 控制器最後將移動與加工的訊號傳送到 CNC 機械的主軸與伺服系統，整個系統的作動就完成了。

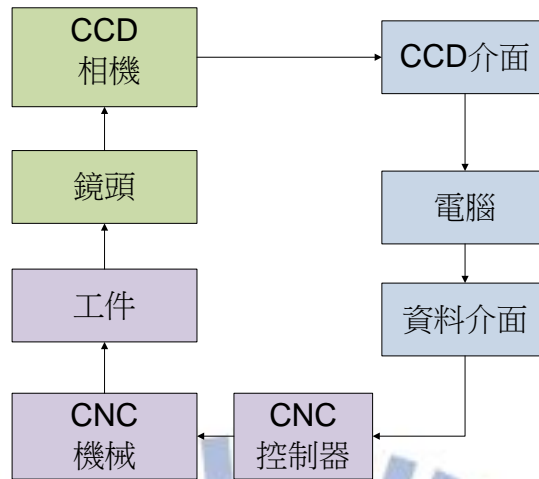


圖 2-4 械視覺加工補正系統圖

2.2.1 影像檔格式(Image Format)

一般來說，數位影像本身在紀錄的時候，有高、寬、深三個層面。影像高度和寬度是決定影像大小的關鍵，而深度則是決定影像層次的重要一環。我們可以把這樣的概念當作一個積木方塊來想像。積木的長與寬，也就是積木的平面面積越大，能夠容納的色彩資料也就越多。如果每一個積木本身是一個顏色的色塊，平面上的積木越多（也就是平面面積越大），我們能夠組合成的圖形就越細膩。

影像是以二維陣列來表示影像的光影強度大小。影像光強度方程式為

$$f(x, y)$$

x, y 分別代表二維陣列的行(column)與列(row)。

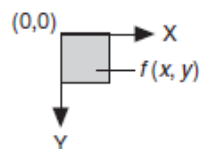


圖 2-5 影像光強度方程式[5]

數位影像處理式利用感測器，將影像轉換成離散的像素(pixels)，就是 $f(x, y)$ 所代表影像感測器(sensor)陣列的一個圖素(pixel)的強度。

表 2-1

影像與資料型態列表	
影像資料型態	單位 pixel 的 byte 數
1 bit , Line art	1/8 byte per pixel
1/8 byte per pixel	1/8 byte per pixel
16 bit , Grayscale	2 bytes per pixel
24 bit , RGB	3 bytes per pixel
32 Bit , CMYK	4 bytes per pixel
48 bit , RGB	6 bytes per pixel

表 2-1 列出影像格式 (Raster format) 與電腦中資料型態的對照，資料愈大所能描述的影像內容愈精確，相對地所占的記憶體也愈大，影像處理計算的時間也愈長，其代表另一種意義是耗費較多的成本。

常用的影像檔案格式如以下簡介：

Windows Bitmap(BMP)：這是微軟公司所提出的點陣圖格式，原本是專門用在 Windows 作業系統上，讓各軟體的圖形能彼此相容，像簡報、介面設計以及 Windows 桌面的底圖。BMP 檔雖然普遍，但有個壞處是雖然支援 RGB 全彩，卻無法壓縮全彩影像，存檔後會變得很大(影像佔用多少資料量，存檔後就有多大)，不適用於網路上。而 16 色、256 色和灰階圖片則可以使用 RLE 技術壓縮，壓縮後圖片不會失真，但儲存和開啟的速度會比較慢。而 48-bitRGB 全彩及 16-bit 灰階的圖片，在目前的影像類型無法存成 BMP 檔。

Graphic Interchange Format(GIF)：GIF 是網頁上最常用的影像格式了，原因是它可以製作透明圖(把背景變透明)、交錯圖(在瀏覽器中慢慢顯現)、和動畫(把許多 GIF 圖片連續重疊成一個檔案)，而且提供「非破壞性壓縮」，可以透過色盤數減少的方式將檔案縮小，圖片也不會失真，適合用

在遠距傳遞及網頁設計的影像應用上。但 GIF 最多只儲存 256 色的色彩數目，所以在儲存全彩的影像之前，必須將圖片轉為 256 色、16 色、灰階或黑白的影像類型，才能存成 GIF 檔。

Joint Photographic Experts Group(JPG/JPEG)：也是網頁常用的圖形格式，壓縮率非常驚人，原本 1MB 的圖片，存成 JPG 檔後可能只剩幾十 K(視影像的複雜度及設定的壓縮率而定)。由於 JPG 格式屬於破壞性壓縮，存檔時會捨棄一些不必要的像素(捨棄後就再也救不回來了)，因此可能造成圖片失真；不過一般而言，只要以正常的比例壓縮就算破壞了一些影像品質，肉眼是很難看出壓縮前後的差異的。儲存 JPG 檔時，當壓縮率愈高，影像的品質會降低；若壓縮率愈低，影像較不容易失真，但檔案也相對較大。JPG 格式支援 RGB 全彩、灰階等影像類型，但 16 色、256 色、黑白圖片則都無法存成 JPG 檔。

Portable Network Graphic(PNG)：由於 GIF 格式引發的權利金問題，所以 PNG 格式就成為免費的 GIF 代替品，在全球網際網路上作為非破壞性壓縮的影像格式。其與 GIF 不同之處在於：結合 GIF 可製作透明圖、交錯顯示、非破壞性壓縮、和 JPEG 支援全彩影像的優點，以及可製作透明度的 Alpha Channel，但目前尚無法儲存動畫。

Tagged Image File Format(TGA)：這是一種專業且常用的影像檔案格式，屬於非破壞性壓縮，支援 24 位元全彩與 32 位元全彩，含透明度資訊等兩種格式。所謂透明度資訊就是，可以使至少兩張以上的影像圖形互相產生半透明的疊合效果，業界也將這種透明度訊稱之為 Alpha 通道(Alpha Channel)，它是種隱性的影像資訊，也就是說，一般軟體是無法解讀這部分的資訊，必須透過如 PhotoImpact 及 Photoshop 專業像處理軟體，或者是如 MediaStudio Pro、After Effect 等的數位視訊軟體與 3D 動畫軟體，才能將 TGA 檔中的 Alpha 通道解讀出來加以應用。TGA 檔擁有其他影像檔案格式所沒有的 Alpha 通道，常被應用在影像合成、視訊合成與 3D 動畫製作的靜、動態影像合成上。

Photo CD(PCD)：這是柯達(Kodak)公司制定的相片光碟(Photo CD)格式。所謂相片光碟就是將照片存在光碟片裡面，不過可不是先洗成照片才掃描進電腦，而是直接將底片「沖洗」成數位影像，所以品質非常高，最高可達 4096X6144 個像素大小，而且還可以配上聲音及文字，做成多媒體光碟來播放。若使用者在開啟 PCD 檔時不要這麼大的圖片，也可以把它開

啟成 1024X1536、512X768 等大小，並可用全彩、256 色或灰階來檢視，而不影響檔案本身的內容。不過要注意的是，由於 PCD 是柯達公司獨家的技術，所以 PhotoImpact 並不能儲存 PCD 格式，但可以開啟 PCD 格式來使用、檢視。

Tagged Image File Format(TIF)：是多家公司共同製定的一種標準影像格式，也是影像處理界最普遍支援的圖檔格式，因為它可以跨平台、提供非破壞性壓縮，不會降低影像的品質，十分適用於印刷輸出，所以大多數的影像處理軟體及排版軟體都會支援 TIF 圖檔。TIF 支援 RGB 全彩、16 色、256 色、灰階、黑白影像類型，目前也唯有 TIF 檔能存成 16-bit 灰階與 48-bit 的全彩類型，如果有圖片要做為印用途，那麼存成 TIF 檔是最好的方式。

Ulead Photo Impact(UFO)：在 PhotoImpact 中所製作出來的影像，包含 PhotoImpact 專屬的物件系統及其相關的設定，為了要保留這些物件屬性及其設定，以便日後可加以重新編修，UFO 提供保留物件的功能，是 PhotoImpact 專屬的影像檔案格式，其它影像處理軟體並不支援 UFO 格式。不過如果要用別的軟體來開啟此圖片，必須先將其中的物件全部合併，存成一般類型的圖檔才行。

Bitmap 或 Raster 是一般所謂的影像的數位相片的格式。點陣圖(Bitmap)影像是儲存在電腦中的攝影機記憶體資料，每一點稱為一個圖素(Pixel)。

2.2.2 點陣資料格式(Raster Image)與取樣 (Sampling)

影像與感測器陣列如圖 2-6 所示，以點矩陣的方式來表示。要顯示 a 這個字母，圖右邊表示感測器陣列，每一個小方格代表一個圖素(pixel)，格內的數值表示影像深度。左側顯示依據圖素的深度來顯示字母 a 的影像。

每一個小方格代表一個圖素的深度表示通常是 0 到 1。也有以 0 到 255 或者是其它數值，其無關宏旨，主要能精確表示影像強度即可。

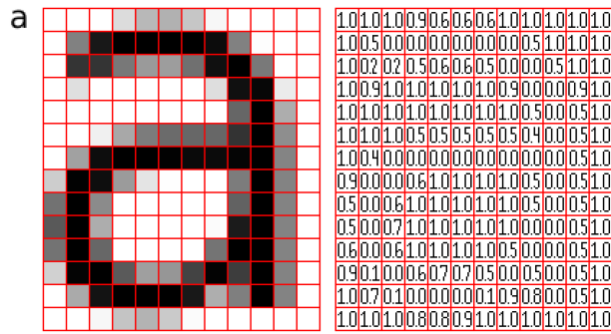


圖 2-6 記憶體影像光強度

2.2.3 影像過濾(Image Filtering)

由背景的雜訊可能會影響比對的精度，影像的過濾可以提高比對精度，但需考慮雜訊的特性。



圖 2-7 影像過濾

2.2.4 臨界值(Threshold)

臨界值的作用是當計算得到的灰階值大於或等於該門檻值時，將該點的值設為白色。如果計算得到的灰階值時，則該點的颜色設為黑色。這一個效果。如果當雜訊是可以用灰階的深度做為區別時，可以用臨界值將雜訊濾除，如圖 2-8。顯示灰階的影像強度臨界值的選擇，從 166 到 255。圖 2-9 是當使用臨界值後，從左邊的影像轉換到右邊的黑白圖形，一般也稱之為二值化(binary)。

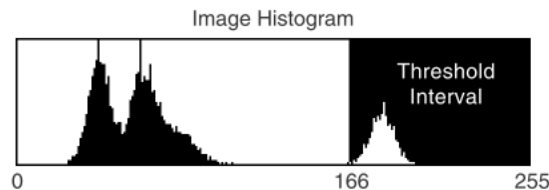


圖 2-8 影像臨界值[5]

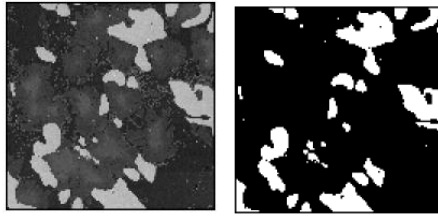


圖 2-9 臨界值影像過濾[5]

2.2.5 模式比對(Pattern Match)

模式比對是用來標定樣本檔(template file)在影像檔中的位置。模式比對找到吻合的樣本檔，其與樣本檔的光度、模糊、干擾、移動與旋轉無關。

使用模式比對需要先產生樣本檔，用來作為標定要找到的物件。然後機械視覺應用軟體，用這一個預先產生的樣本檔，去搜尋整個影像檔。每一個吻合的計算結果中會產生計算值(score)。這一個計算值可以知道這一個吻合計算結果有多靠近理想值。

在影像處理(image processing)中，模式比對演算法是其中最重要的一個函數。模式比對可以用在以下三種通用的應用上。

- 校準(alignment)：使用基準點在已知的物件上測定位置與方向。
- 測量(Gauging)：量測長度、直徑、角度及其他的重要尺寸。如果測量值，掉在精度範圍外，這個元件就會被拒絕。
- 檢測(Inspection)：檢查簡單的瑕疵，例如零件數量的短少或者漏掉的印刷。

如圖 2-10 所示，搜尋一個包含在印刷電路板上的影像或者更多的校準標的。這個機械視覺應用，利用對位標誌的校準來使的電子零件的插件機構，將電子零件置放在印刷電路板上。

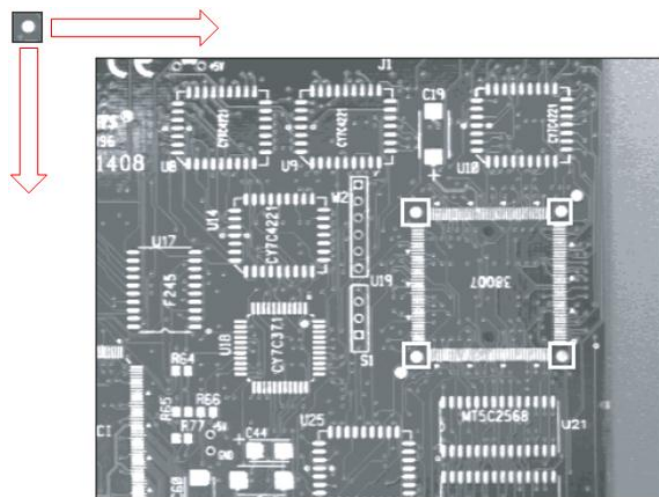


圖 2-10 模式比對

模式比對的數學式，如下述[1]：

Square difference matching method

$$R_{sq_diff}(x, y) = \sum_{x', y'} [T(x', y') - I(x + x', y + y')]^2 \quad (1)$$

Correlation matching methods

$$R_{ccorr}(x, y) = \sum_{x', y'} [T(x', y') \cdot I(x + x', y + y')] \quad (2)$$

Correlation coefficient matching methods

$$R_{ccoeff}(x, y) = \sum_{x', y'} [T'(x', y') \cdot I'(x + x', y + y')]^2 \quad (3)$$

$$T'(x', y') = T(x', y') - \frac{1}{(w \cdot h) \sum_{x'', y''} T(x'', y'')} \quad (4)$$

$$I'(x + x', y + y') = I(x + x', y + y') - \frac{1}{(w \cdot h) \sum_{x'', y''} I(x + x'', y + y'')} \quad (5)$$

$$Z(x, y) = \sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2} \quad (6)$$

$$R_{sq_diff_normed}(x, y) = \frac{R_{sq_diff}(x, y)}{Z(x, y)} \quad (7)$$

$$R_{ccorr_normed}(x, y) = \frac{R_{ccorr}(x, y)}{Z(x, y)} \quad (8)$$

$$R_{ccoeff_normed}(x, y) = \frac{R_{ccoeff}(x, y)}{Z(x, y)} \quad (9)$$

正常化法(Normalized methods)

加上正常化的程序，可以降低樣本檔與影像檔間，因光線造成的差異，所引起的比對效果差異。

$$Z(x, y) = \sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2} \quad (10)$$

Normalized Square difference

$$R_{ccorr_normed}(x, y) = \frac{R_{ccorr}(x, y)}{Z(x, y)} \quad (12)$$

Normalized Correlation coefficient

$$R_{ccoeff_normed}(x, y) = \frac{R_{ccoeff}(x, y)}{Z(x, y)} \quad (13)$$

2.3 機械加工補正

在機械加工補正上，使用的方法與傳統標準 ISO Code 在控制器上的補正方式有很大的不同。ISO Code 的補正是使用 G41、G42 作為刀半徑補正，G43、G44 作為刀長補正。刀具路徑可以使用絕對座標(absolute coordinate)G54、G55、G56、G57、G58、G59 作為加工座標的平移(shift)。G52 重設加工座標。

傳統的 NC 加工的定位與加工不能滿足前述加工的需求，因為此時已沒有治具，而且原始的加工圖形有可能已變形。所以需要有一個輔助系統可以滿足以下兩個目的。

- (一)工件置放補正(大板補正)
- (二)加工路徑補正

2.3.1 工件置放補正

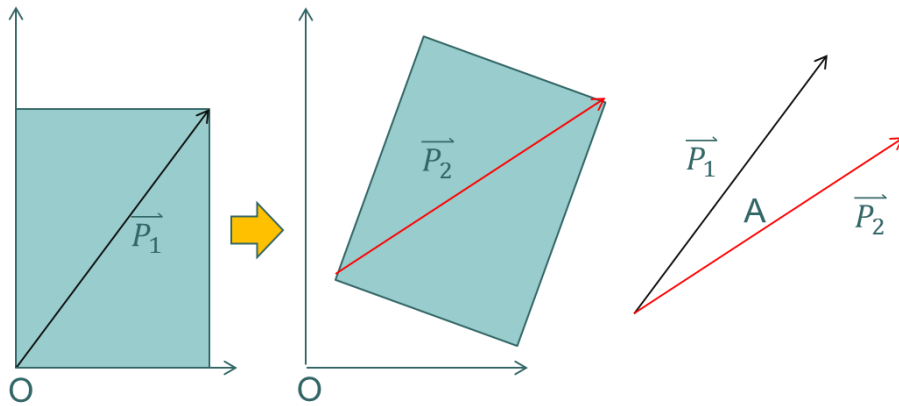


圖 2-11 工件置放補正

d_x : x 方向位移

d_y : y 方向位移

A : 旋轉位移

$$d_x = P_{2x} - P_{1x}$$

$$d_y = P_{2y} - P_{1y}$$

$$A = \cos^{-1} \frac{\overline{P_1} \cdot \overline{P_2}}{|\overline{P_1}| |\overline{P_2}|}$$

2.3.2 加工路徑補正

網版印刷與數位噴印的工件往往會有幾種變形的產生。旋轉變形、線性變形、扭轉變形、非線性變形。如果欲得到良好的裁切效果須將變形的結果修正。

如圖 2-12 為在影像檔中的圖形與刀具路徑的示意圖。圖 2-13 為單純的旋轉變形與補正，圖 2-14 為另一種單純的線性變形與補正，它只產生了 x 與 y 軸向的位移變形。圖 2-15 扭轉變形與補正。圖 2-16 變形為無特性的變形，亦可稱之為非線性變形及其補正後的示意圖。



圖 2-12 原始未變形圖形

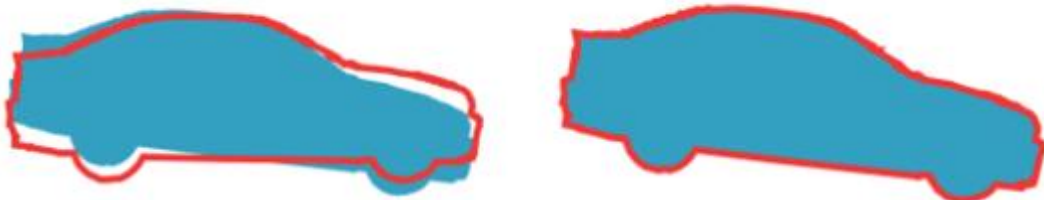


圖 2-13 旋轉變形與補正

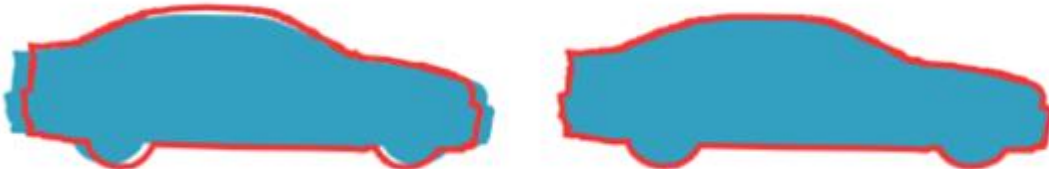


圖 2-14 線性變形與補正



圖 2-15 扭轉變形與補正

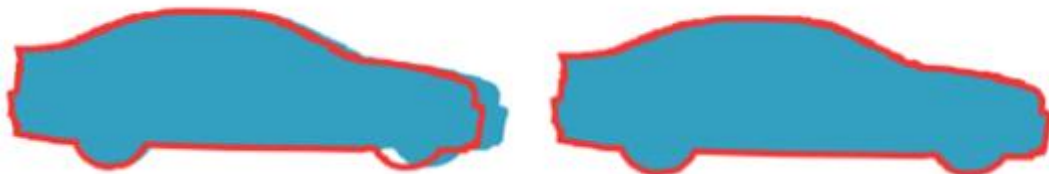


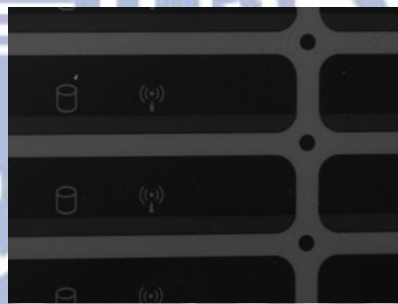
圖 2-16 非線性變形與補正

欲修正上述的變形，需要在圖形的適當的位置，預先印上定位的圖形，如圖 2-17(a)、(b)。目的是使定位圖形能隨著圖形的變形，能跟著一起變化。以後就能使用 CCD 取得圖形位置變化，作為補正的依據。

圖 2-17(a)是通用的補正方法，可以補正所有的變形方式，但是缺點是必須有許多的補正參考點，如果 CCD 一點、一點去取像，取得補正點的補正值，將耗用較多的時間。針對較小的工件可以使用單一的定位補正點，如圖 2-15(b)。只以單一補正點取得補正值後，將補正值加入整個圖形模塊的加工路徑座標中。



(a)廣告印刷



(b)電子產品

圖 2-17 定位點的佈置

定位點與印刷變形有相關性，可以平均誤差值作為補正值

$$\bar{x} = \frac{\sum_{i=0}^n (x_i - x_0)}{n}$$

$$\bar{y} = \frac{\sum_{i=0}^n (y_i - y_0)}{n}$$

\bar{x} ：X 軸的平均誤差

\bar{y} : y 軸的平均誤差

x_i : x 軸的實際定位座標

x_0 : x 軸的實際定位座標

y_i : y 軸的實際定位座標

y_0 : y 軸實際定位座標

共有 N 個取樣點，第 i 點 $0 < i \leq N$

$$\dot{x}_i = x_i + \bar{x}_i$$

$$\dot{y}_i = y_i + \bar{y}_i$$

x'_i = x 軸實際加工座標

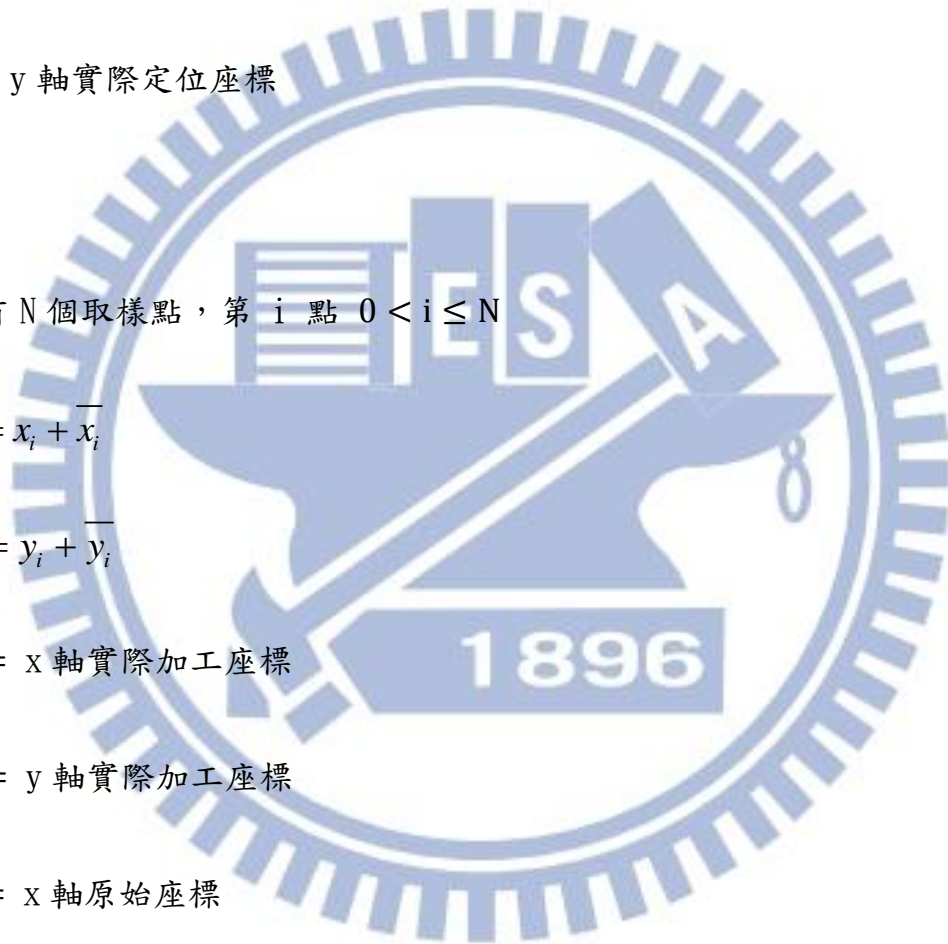
y'_i = y 軸實際加工座標

x_i = x 軸原始座標

y_i = y 軸原始座標

\bar{x}_i = x 軸座標平均誤差值

\bar{y}_i = y 軸座標平均誤差值



第三章 系統架構規劃與方法

系統架構設計可以大分類為兩大塊，第一部分是硬體架構，第二部分是軟體架構。結合這兩部分組成一個完整的機械視覺加工補正系統。

3.1 硬體架構

3.1.1 選擇相機

作為光感測器的相機(Camera)可分為 CMOS 與 CCD 兩大類。

CMOS：全名為 Complementary Metal-Oxide Semiconductor 「補充性氧化金屬半導體」

CCD：全名為 Charge Coupled Device 「感光耦合元件」

CCD 優點在於低雜訊、高感度、線路設計及製程單純且技術成熟，缺點為高耗電量、畫素無法隨機讀取、資料傳遞必須接近完整。CMOS 優點在於價位低、耗電量低、畫素可隨機讀取、單一晶片上可整合其他功能，缺點在於雜訊度較高、感度較差、晶片線路複雜。

由於 CCD 擁有較佳之感度，較低之雜訊雖然價格較高，一般工業級相機，多以 CCD 為主，本研究選擇使用 Teldyne 出品的 Dalsa Genie M640，作為取像的相機，其外形如圖 3-1 所示。



圖 3-1 Dalsa M640 相機

3.1.2 CCD 的影像擷取介面

CCD 影像擷取介面主要的規格如圖 3-2 所示。從較早期如圖 3-2(a)的同軸電纜與影像擷取卡，到近期主流規格使用一般 PC 上內建通用介面，如圖 3-2(c)USB、圖 3-2(d)IEEE1394 與圖 3-2(e)GigE 等。對某些對針對速度特別在意的應用範圍，Camera Link 如圖 3-2(b)Camera Link 影像擷取卡與接頭排線。

選擇何種介面需視影像應用的取像速度需求與成本考量，選擇一種較佳的性價比做為系統的取像介面。本研究將選擇 GigE 的介面，作為取像介面。

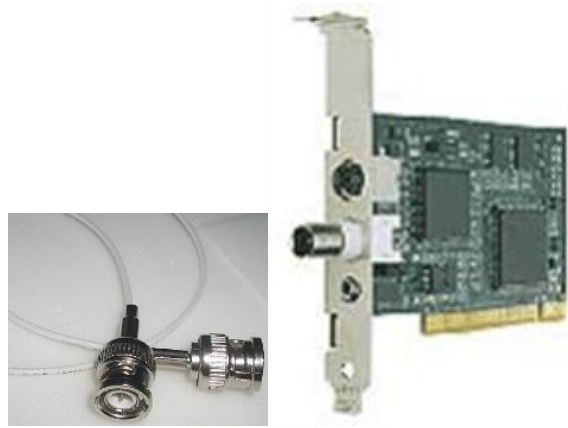


圖 3-2(a)同軸線與影像擷取卡



圖 3-2(b)Camera Link 影像擷取卡與接頭排線



圖 3-2(c)USB

圖 3-2(d)IEEE1394



圖 3-2(e)GigE

圖 3-2 影像資料擷取介面

GigE 或稱 GigE Vision 是在 2006 年推出的是一種高性能工業相機標準介面。它提供了一個乙太網絡的高速視訊和相關的數據控制傳輸架構。

該標準是由一組 12 家企業發起。委員會已發展到包括 50 個成員 12 個創始成員是：Admic、ATMEL、Basler AG、CyberOptics、DALSA、JAI A / S、JAI PULNiX、Matrox、National Instruments、Photonfocus、Pleora Technologiesc 和 Stemmer Imaging。

GigE Vision 是基於網際網路的標準。該標準是試圖統一機械視覺工業相機目前使用的通訊協定，並讓第三方組織開發相容的軟體和硬體。GigE Vision 標準是一個公開的標準，所以它是由歐洲許多國家承認的一個封閉的標準。它提供給任何一個組織使用僅收取象徵性的費用。

GigE Vision 標準的特點：

- 快速數據傳輸率-高達 1000Mbit / S。
- 數據傳輸長度可達 100 公尺（部分製造商宣稱可達 150 公尺）。
- 根據既定的標準，可與其他乙太網設備和電腦連線。
- 提供新一代基於主從乙太網路架構的乙太網路視訊應用架構。GigE 視覺主要有四個要素：
- GigE 視覺控制協議（GVCP）。在 UDP 協議上執行。該標準定義了如何控制和配置設備。指定發送圖像和配置相機和電腦之間的串流通道和機制。
- GigE 視覺串流協議（GVSP）。在 UDP 協議上執行。涵蓋資料型態定義和圖像可以通過 Gigabit 乙太網路的傳輸方式。
- Gigabit 乙太網設備搜尋機制。提供機制以搜尋 IP 位址。
- 一份 XML 描述文件，基於歐洲機器視覺協會的 GenICam 標準的架構定義。這是一份允許進入相機控制和影像串流的資料文件。

應用上的考量 GigE 的介面具有的優點是訊號線長度達 100 米，機台配線限制小，接頭使用 RJ45 是很平常的乙太網路線接頭。所以很容易購置，成本也很低。

3.1.3 機械視覺的光學與電腦系統

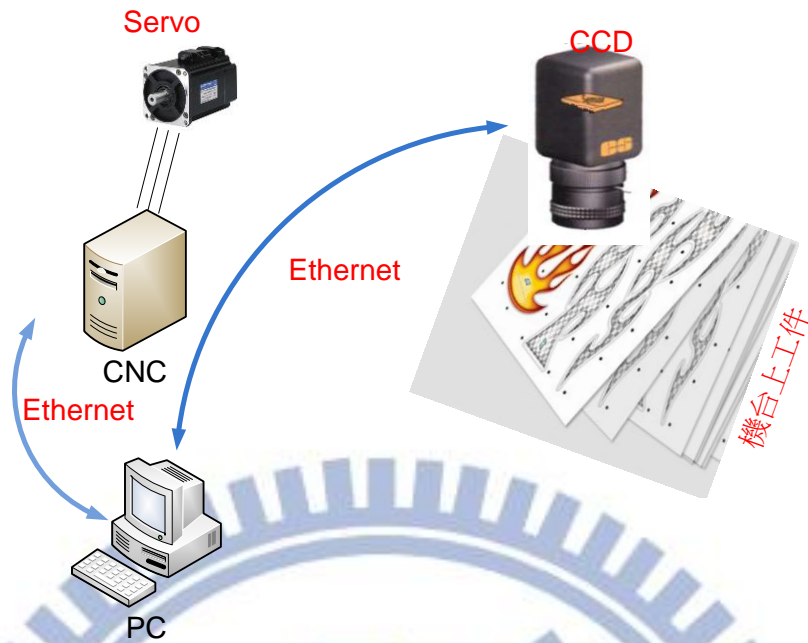


圖 3-3 光學系統與電腦系統關係圖

如圖 3-3 表示工件、CCD、PC 與 CNC 控制器加上伺服馬達間的關係。CCD 與 PC 間乙太網路線(Ethernet)連接，PC 與 CNC 控制器也是乙太網路線。CNC 控制器模組內的驅動器以訊號線連接伺服馬達，驅動器再驅動各個軸系運動。

3.2 軟體架構

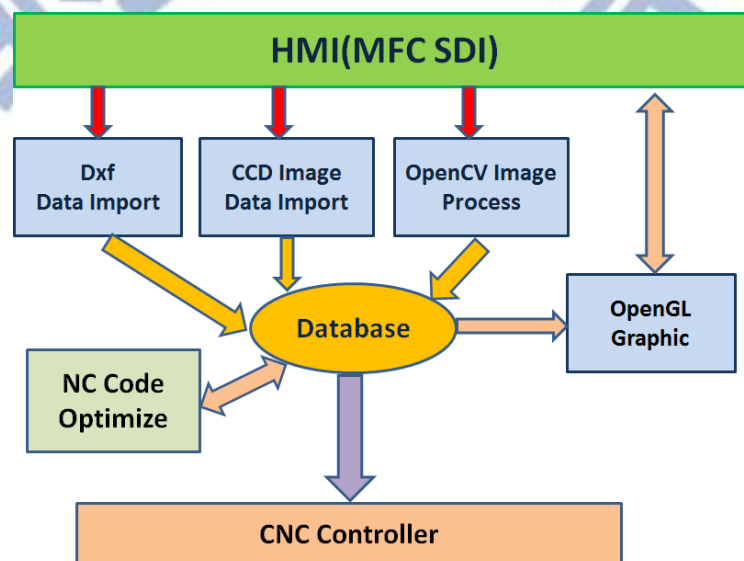


圖 3-4 軟體架構圖

圖 3-4 顯示整個軟體的結構，描述整個機械視覺加工補正系統各個功能模組間的關係。系統包括了 HMI(Human Machine Interface)、Dxf Data Import、CCD Image Data Import、Database、OpenGL Graphic、OpenCV Image Process、NC Code Optimize 與 CNC Controller。

HMI(Human Machine Interface):人機介面，人與機器間的圖形介面。負責使用者與程式指令間的圖形溝通架構。

Dxf Data Import：外部異質資料輸入模組，連接系統與外部 CAD 的異質資料，使用的是 DXF 格式。

CCD Image Data Import：負責機械視覺取像的部分，CCD 硬體必須藉由這個模組與系統溝通，模組建構在 CCD 原廠提供的 SDK(System Development Kit)上。

Database：資料庫，所有外部傳入的資料或者計算完成的資料，負責儲存的模組。

OpenGL Graphic：將圖形資料自 Database 中取出，並在視窗中繪出圖型。

OpenCV Image Process：專門做為機械視覺影像處理的部分，包括模式比對部分的功能，都是屬於這個模組。

NC Code Optimize：從資料庫將資料取出並計算加工補正的值，並產生新的加工資料。

CNC Controller：將最終的結果傳至控制器的功能。

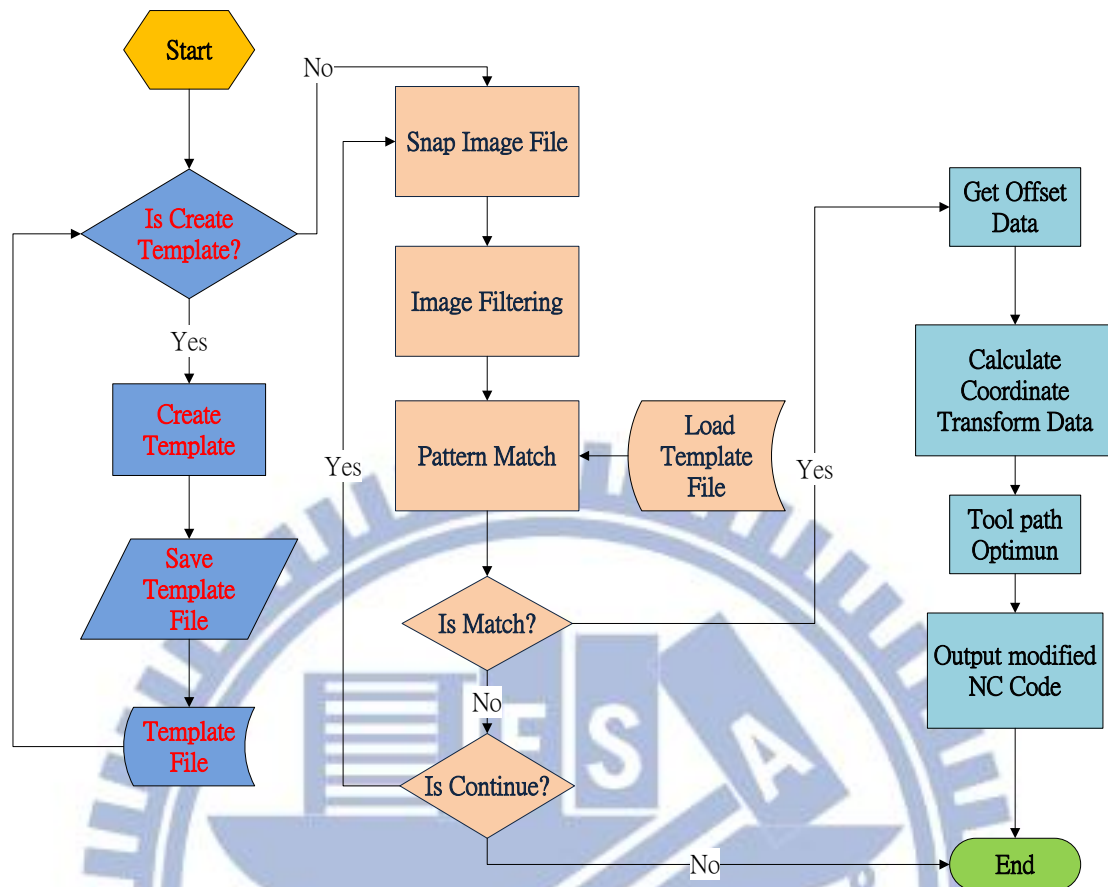


圖 3-5 機械視覺與加工補正系統流程圖

圖 3-5 機械視覺與加工補正系統流程圖說明整個樣本檔產生，樣本比對與補正值計算的流程。

流程圖從左邊”Start”開始，”Is Create Template”判斷是否需產生比對樣板(template)，因為每一次模式比對都需要一個樣本檔。如需產生樣本檔則進入”Create Template”，然後儲存樣本檔”Save Template File”，提供稍後模式比對之用。

”Snap Image File”當機台透過NC程式移動到定位點預定位置，補正程式透過CCD的SDK程式，抓取一張預訂定位點的影像。”Image Filtering”依據環境干擾的狀況採取對應的過濾對策。然後進行模式比對”Pattern Match”，”Is match?”模式比對有可能因為位置差異過大而無法取得結果，此時需手動移動機台，使CCD到達可視範圍內然後”Is Continue?”判斷跳回繼續重複一次”Snap Image File”、”Image Filtering”、”Pattern Match”流程。但是有可能有一種狀況是目標的影像干擾太大，影響比對精度，或者是樣本不對需重新建立樣本檔，此時要將補正流程結束，排除上述異常後再進要做的補正工作。

當樣本比對成功後，“Get Offset Data”取得樣本的偏置量。”
Calculate Coordinate Transform Data”依照偏置量計算出工件的位移與旋轉量。“Tool path Optimum”根據每一點的 CCD 取得的偏置量，計算 NC 程式補償所需的值。“Output Modified NC Code”當每一個單節的 NC 程式重新計算出來後，即時的輸入控制器，此時已完成補正的目的。

機械視覺加工補正系統的軟體，以 Microsoft Fundamental Class(MFC)的 Single Document Interface(SDI)為視窗架構。繪圖採用 OpeGL Library 為視窗繪圖架構。Dxflib 作為異質資料轉換函式庫。而 OpenCV 則是機械視覺函式庫，整個機械視覺加工補正系统的核心部分。透過 OpenCV 的模式比對功能，獲得補正的資料。

3.2.1 基於 MFC 之 SDI 介面

MFC (Microsoft Foundation Class Library)中的各種類別結合起來構成了一個應用程式框架，這是一種相對 Win32 SDK 來說更為簡單的方法，它的目的就是讓工程師在此基礎上來建立 Windows 下的應用程式。因為總體上，MFC 框架定義了應用程式的輪廓，並提供了使用者介面的標準實作方法，工程師所要做的就是通過預定義的介面把具體應用程式特有的東西填入這個輪廓。Microsoft Visual C++提供了相應的工具來完成這個工作:AppWizard 可以用來生成初步的框架檔（代碼和資源等）；資源編輯器用於直觀地設計使用者介面；ClassWizard 用來協助添加代碼到框架檔；最後，編譯，則通過類別函式庫實做了應用程式特定的邏輯目的。

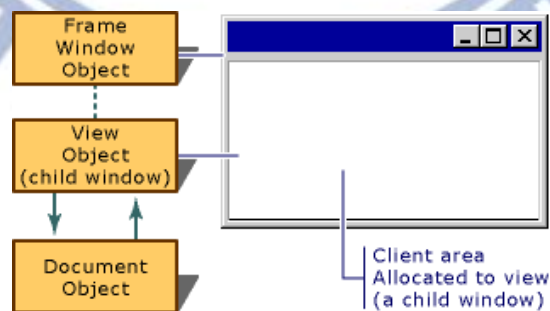


圖 3-6 MFC SDI 結構圖

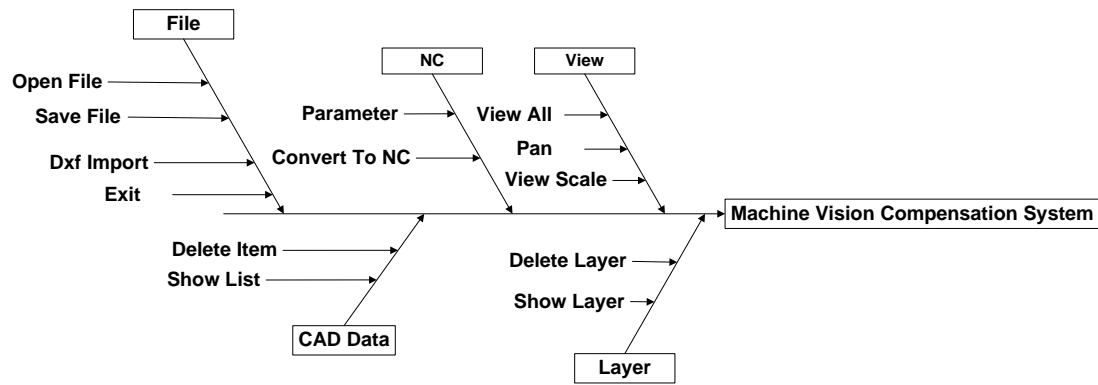


圖 3-7 系統功能圖

圖 3-8 顯示機械視覺加工補正系統，在 MFC SDI 架構下的功能魚骨圖。檔案(File)主功能下分開啟檔案(Open File)，開啟已產生的舊檔。存檔(Save File)，儲存新檔或修正的檔案。Dxf 輸入(Dxf Import)，以 Dxf 格式輸入異質資料庫。退出系統(Exit)。

主功能 CAD Data 用來管理輸入的 CAD 資料，刪除資料(Delete Item)刪除 Dxf 輸入中某些不需要的輸入的資料。主功能 NC 下，加工參數(Parameter)的設定，這包括了刀號、刀長補證號碼、刀半徑補正方式，刀半徑補正號碼、主軸轉速、加工進給率等加工參數。主功能圖層(Layer)，圖層顯示(Show Layer)顯示異質資料輸入的圖層架構。刪除圖層>Delete Layer)，刪除某一些異質資料庫輸入後，某些不需要的圖層。主功能視覺(View)，用來觀看輸入的資料，全部(All)表示顯示全部，移動(Pan)移動資料顯示的位置，顯示比例(View Scale)用來調整顯示的大小比例。

3.2.2 OpenGL

OpenGL (Open Graphics Library) 是個定義了一個跨程式語言、跨平台的程式介面(Application programming interface)的規格，它用於產生二維、三維圖像。這個介面由近三百五十個不同的函式組成，用來從簡單的圖元繪製複雜的三維景象。OpenGL 常用於 CAD、虛擬實境、科學視覺化程式和電子遊戲開發。

OpenGL 的高效能的實作(利用了圖形加速硬體)可執行於 Windows，很多不同版本的 UNIX、Linux 平台和 Mac OS。這些實作一般由顯示裝置廠商提供，而且非常依賴於該廠商提供的硬體。

OpenGL 規範由 1992 年成立的 OpenGL 架構評審委員會 (ARB) 維護。ARB 由一些特別興趣於建立一個統一的普遍可用的 API 的公司組成。根據 OpenGL 官方網站，2002 年 6 月的 ARB 投票成員包括 3Dlabs、Apple Computer、

ATI Technologies、Dell Computer、Evans & Sutherland、Hewlett-Packard、IBM、Intel、Matrox、NVIDIA、SGI 和 Sun Microsystems (Microsoft 曾是創立成員之一，但已於 2003 年 3 月結束)。

選取使用 OpenGL 的理由是，OpenGL 用於繪圖時期使用 3D 卡氏座標，右手定則。如圖 3-5 所示，觀測者自螢幕外直視螢幕，螢幕往右邊為 x 軸正方向，螢幕往上為 y 軸正方向。整個座標系與數學座標系一致。未考慮 Windows GDI+ 或者是 DirectX，原因是 Windows GDI+ 螢幕 X 向右為正，Y 軸向下為正，DirectX 座標使用卡氏座標左手定則兩者皆與常用數學座標系相違背，應用上較不方便。

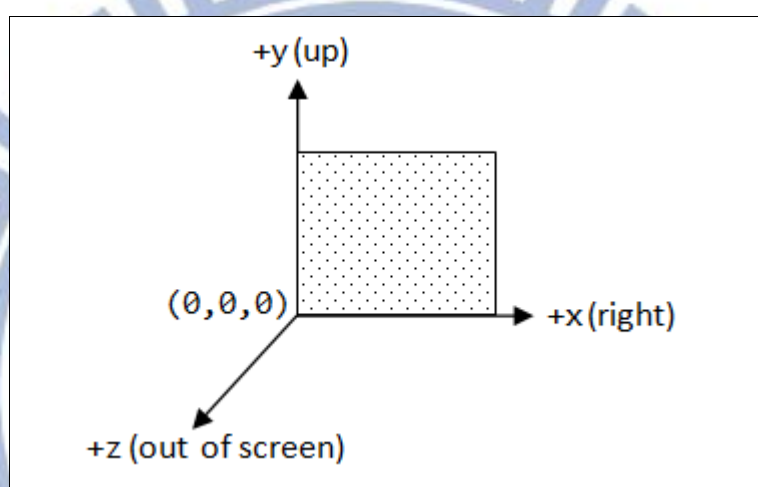


圖 3-8 OpenGL 座標

3.2.3 DXF

ASCII 版本的 DXF 可以用文字編輯器讀取。基本的 DXF 檔案結構如下：

- HEADER 部分 - 一般通用的繪圖資料。每一個參數有它的變數名稱與相關數值。
- CLASSES 部分 - 包括應用程序定義的類別的資料，這些實例將顯示在 BLOCKS、ENTITIES 以及 OBJECTS 部分。通常不包括用於充分用於與其他應用程序交互的信息。
- TABLES 部分 - 這部分包括命名條目的定義。
- Application ID (APPID) table
- Block Record (BLOCK_RECORD) table
- Dimension Style (DIMSTYLE) table

- Layer (LAYER) table
- Linetype (LTYPE) table
- Text style (STYLE) table
- User Coordinate System (UCS) table
- View (VIEW) table
- Viewport configuration (VPORT) table
- BLOCKS 部分- 這部分包括 Block Definition 實體用於定義每個 Block 的組成。
- ENTITIES 部分 - 這部分是繪圖實體，包括 Block References 在內。
- OBJECTS 部分 -- 包括非圖形對象的數據，供 AutoLISP 以及 ObjectARX 應用程序所使用。
- THUMBNAILEDIMAGE 部分 -- 包括 DXF 文件的預覽圖。
- END OF FILE

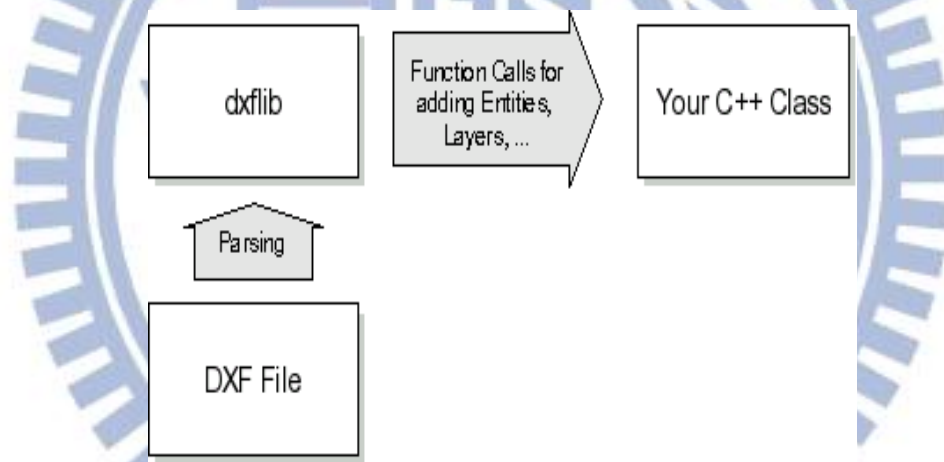


圖 3-9 dxflib 函式架構

dxflib 是一個開放源碼的 C++ 函式庫。是一個用來讀寫 DXF 檔的 C++ 函式庫。當讀 DXF 檔的時候，dxflib 分析檔並且呼叫使用者自己定義的函數來添加實體、圖層、等。需要注意的是，dxflib 並不保存任何實體或者資訊。它只是從 DXF 檔中分析可以識別的實體以及其他的物件。

使用 dxflib 函式庫讀 DXF 檔並不需要知道所有的 DXF 格式資訊。當然，像實體、屬性、層、段等基本概念還是需要瞭解的。如果是用 dxflib 庫寫 DXF 檔，則需要知道 DXF 檔是如何組成的。dxflib 完全基於 C/C++ 標準函式庫實作，不依賴任何其他的函式庫。

程式開發者

- Andrew Mustun (RibbonSoft) – Main developer and project coordinator
- Robert J. Campbell Jr. – Initial writer implementation
- Christian Ahmels (Fraunhofer Institut fuer Integrierte Schaltungen IIS) – Support for Traces and Solids
- Andreas Tschärner (Metromec Software AG) – Patches for Windows / Splines with Knot values

3.2.4 OpenCV

OpenCV(Open Source Computer Vision Library)，自 1999 年起是由英特爾(Intel)公司發起並參與開發的一個跨平台的機械視覺函式庫，以 BSD 許可證授權發行。可以在商業和研究領域中免費使用。

基本 OpenCV 的特點可以如下幾點描述：

- 以 BSD 許可證授權發行，可以在商業和研究領域中免費使用。
- OpenCV 是由英特爾公司發起並參與開發。OpenCV 可用於開發實時的圖像處理、計算機視覺以及模式識別程序。
- 最基本的程式全部由 C 寫，但是也完全支援 C++的程式介面。
- OpenCV 可用於解決如下領域的問題：

人機互動、物體識別、圖像分割、人臉識別、動作識別、運動跟蹤、機器人。

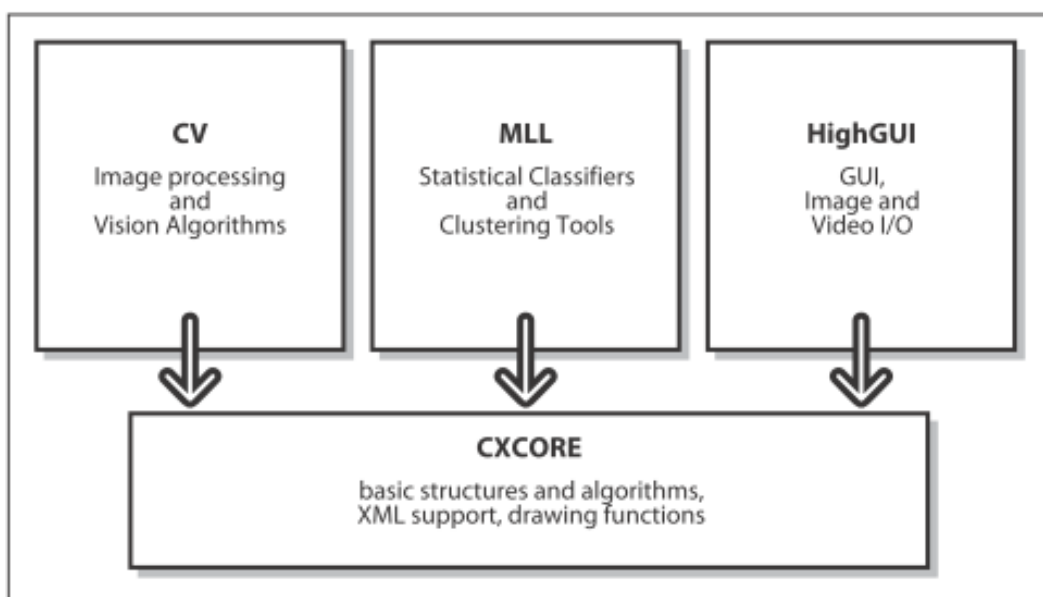


圖 3-10 OpenCV 基本架構

OpenCV 包括了 5 大部分的功能，圖 3-4 表示其中四大基本功能。CV 包含基本影像處理功能，及高階電腦視覺演算法。ML 是機械學習函式庫，MLL(Machine Learning Library)，ML 包含了許多的統計分類與叢集 (cluster) 工具。HighGUI 涵蓋了影像搜尋及載入錄影檔與影像檔的 I/O 程序與功能。CXCORE 包含基本的資料結構與內容。

圖 OpenCV 並未包括 CvAux，CvAux 包括兩個已作廢的區域，一個是 embedded HMM face recognition，另一個是實驗性質的演算法 background\foreground segmentation。

本研究在主要集中在樣本比對(template Matching)或者稱模式比對(Pattern Match)的功能 cvMatchTemplate()，功能上是依次計算樣本(Template File)與待測影像圖片的重疊區域的相似度，並將結果存入映射圖像 CvArr* result 中。

IplImage* 是 OpenCV 中定義的影像資料格式，所有輸入、輸出與解果的儲存均是以此格式定義。

例如：

```
IplImage* img;  
IplImage* temp;  
IplImage* ret;
```

cvMatchTemplate() 函式定義如下：

```
void cvMatchTemplate(  
    const CvArr* img,  
    const CvArr* temp  
    CvArr* result,  
    Int method);
```

img：樣本比對的目標影像。

temp：比對用的樣本檔。

result：比對結果資料。

method：所使用的演算法。

使用 cvMinMaxLoc() 函式取得結果資料中的最大值，函式定義如下：

```
void cvMinMaxLoc  
(  
    const CvArr* arr,  
    double*      min_val,  
    double*      max_val,  
    CvPoint*     min_loc = NULL,  
    CvPoint*     max_loc = NULL,  
    const CvArr* mask    = NULL  
);
```

從 result 中提取最大值，亦即相似度最大，以及最大值的位置，也就是 result 中該最大值 max_val 的座標位置 max_loc，即比對樣本滑行時樣本左上角的座標。

3.2.5 控制器與控制 API

一般 PC base 控制均會提供 API(Application Interface)，提供給使用者開發客製的應用程式，客製的應用程式與控制器間的資料交換與溝通須使用這個控制器提供的 API。

在研究中使用了新代控制器。新代控制提供了兩種的 API，第一種是微軟 DotNet 架構的 API “OpenCNC”，第二種是 C 架構的 OCAPI。這兩者均透過控制器的通訊架構 Dipole，在 CNC 控制器與 IPC 間傳遞資料與命令。

由於 CCD 使用的 API 為 C++ 的架構，而介面是採用了 MFC SDI，所以使用 OCAPI 是必要的選擇。當然 C++ 的架構還有執行效率的考量。

第四章 系統實作

4.1 NC 機器與視覺硬體

本系統由 NC 控制器，機械視覺硬體與機械本體所構成。NC 控制器採用新代控制器。機械視覺硬體使用的是加拿大 Teledyne Dalsa 公司的 Genie 系列的 Genie M640，GigE 架構的 CCD。

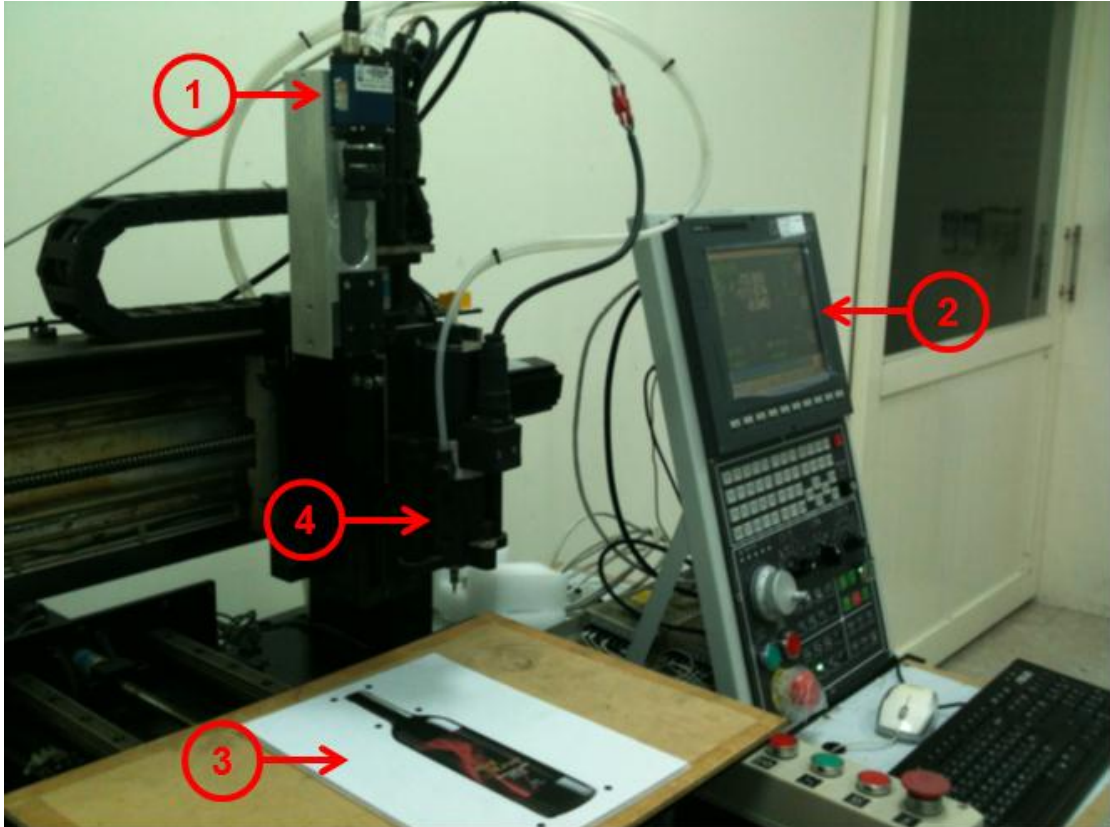


圖 4-1 系統總成簡圖

如圖 4-1 所示為論文研究的測試機台

1. CCD 含鏡頭。
2. 控制器顯示器與控制面板。
3. 加工檯面與檯面上之工件。
4. 加工主軸與刀具。

4.2 數位印刷與後製流程

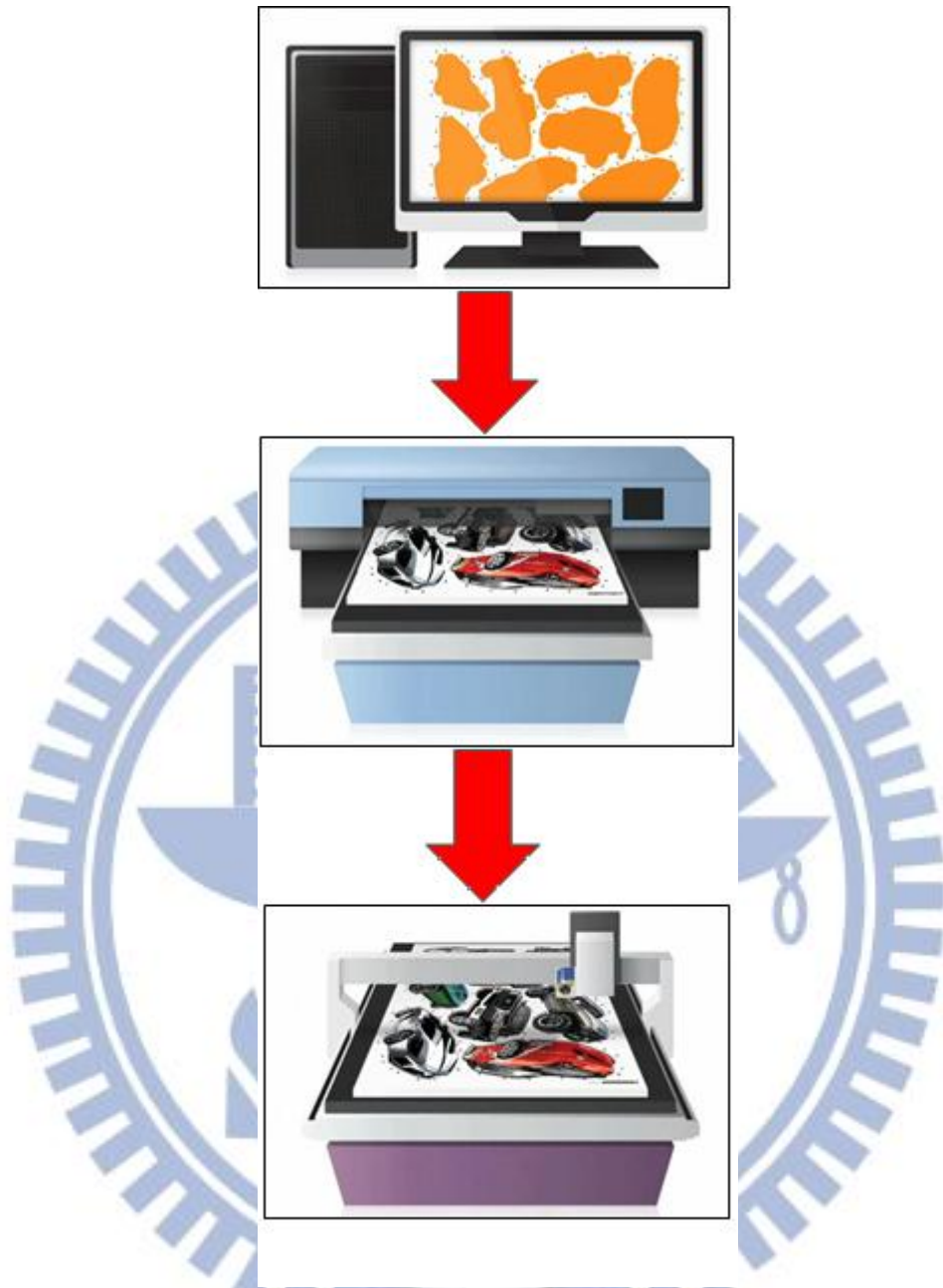


圖 4-2 數位印刷與後製流程

數位印刷的流程大體上如圖 4-2 所示。印刷前須經由照片檔，或手繪圖掃描檔案，讀入電腦。透過影像處理軟體如 Illustrator 或 CorelDraw 等進行印製前的處理。其中包括了顏色的調整處理，影像修改，大小的調整等工作。

影像處理完後通常亦是透過影像處理軟體進行列印的工作，因為影像處理軟體也具有極佳的列印設定功能。透過影像處理軟體的 B-Spline 線的描繪工具，將欲切除的邊界描繪出來，B-Spline 描繪工具，主要是提供線

段描繪時的平順化功能。平順化的目的是使的控制點交接處保持連續，如此，當轉成多折線(polyline)後線段與線段的接點就不會出現過大的轉折，使的加工時出現震動。

定位點即是在此步驟加入到圖形中。通常會在圖形變化較大的位置加入定位點。如圖 4-3 所示。



圖 4-3 無定位點的圖形



圖 4-4 具定位點的圖形

圖 4-3 為無定位點的圖形，圖 4-4 為加入定位點後的圖形。



圖 4-5 B-Spline 的描繪

圖 4-5 表示利用 B-Spline 工具，描繪出影像輪廓的外形線，

4.3 軟體系統

軟體系統包含了幾個重要部分，DXF 轉換，資料庫定義，資料顯示，CCD 取像，OpenCV 定位點辨識，控制器 SDK(System Development Kit)整合，刀具與加工參數設定與 NC code 產生器。

4.3.1 DXF 輸入與資料庫建立

DxfLib 程式庫的應用，須要繼承 `dl_creationadapter` 這一個類別。透過這個類別的虛擬函數如 `addLine(const DL_LineData&)`，`addArc(const DL_ArcData&)`...等解析出 DXF 檔內的圖形元素。

解析出的圖形元素存入資料庫等待後續工作使用。資料庫是以 C++ 標準函式庫 `list` 為基礎，以自訂的資料型態作為資料庫的分類。宣告的資料型態如下：

```
list<CadData> CadDataBase;
```

CadData 的定義說明如下：

```
typedef struct tagCadData
{
    unsigned long dataIndex;
    short dataType;
    long layerNo;
    Line2D line;
    Arc2D arc;
    Circle2D circle;
    BLOCK blockBoundary;
    bool layerActive;
    Point2D offsetValue;
    tagCadData()
    {
        dataIndex = -1;
        dataType = -1;
        layerActive = false;
    }
}
```

```
}
```

其中線段定義是：

```
typedef struct tag2D_Line
```

```
{
```

```
    Point2D Start;
```

```
    Point2D End;
```

```
} Line2D, *pLine2D;
```

圓弧的定義：

```
typedef struct tagArc
```

```
{
```

```
    Point2D Center;
```

```
    double Radius;
```

```
    double StartAngle;
```

```
    double EndAngle;
```

```
    double SweptAngle;
```

```
    short Direction;
```

```
    Point2D StartPoint;
```

```
    Point2D EndPoint;
```

```
    tagArc()
```

```
    {
```

```
        Radius = StartAngle = EndAngle=SweptAngle=0;
```

```
    }
```

```
} Arc2D, *pArc2D;
```

圓的定義：

```
typedef struct tagCircle
```

```
{
```

```
    Point2D Center;
```

```
    double Radius;
```

```
    tagCircle()
```

```

    {
        Radius = 0;
    }
} Circle2D, *pCircle2D;
2D 點的定義
typedef struct tag2D_Point

```

```

{
    double X;
    double Y;
    tag2D_Point()
    {
        X=Y=0;
    }
} Point2D, *pPoint2D;

```

4.3.2 資料儲存與讀取

當轉入的資料以 `stl::list` 架構儲存在記憶體中。為了資料能重複使用。需設計一個儲存的檔案結構。並撰寫一段讀取檔案的程式碼。

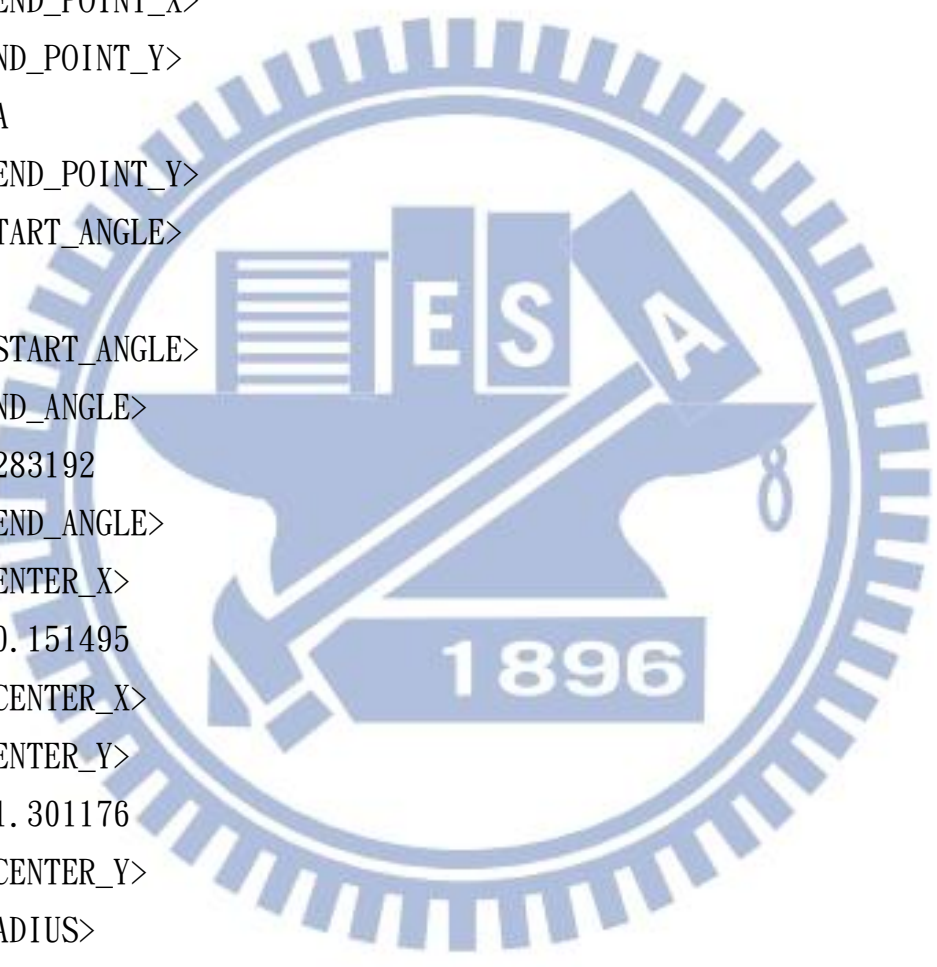
資料儲存的檔案採用 XML 的檔案結構，設計的結構如下例：

```

<?xml version="1.0" encoding="windows-1252"?>
<CAD_TABLE>
<SHAPE>
    <INDEX>
    12
    </INDEX>
    <DATA_TYPE>
    1
    </DATA_TYPE>
    <LAYER_NO>
    -1
    </LAYER_NO>
    <START_POINT_X>

```

N/A
</START_POINT_X>
<START_POINT_Y>
N/A
</START_POINT_Y>
<END_POINT_X>
N/A
</END_POINT_X>
<END_POINT_Y>
N/A
</END_POINT_Y>
<START_ANGLE>
0
</START_ANGLE>
<END_ANGLE>
6.283192
</END_ANGLE>
<CENTER_X>
120.151495
</CENTER_X>
<CENTER_Y>
281.301176
</CENTER_Y>
<RADIUS>
3.249414
</RADIUS>
<MAX_BOUND_X>
123.400909
</MAX_BOUND_X>
<MAX_BOUND_Y>
284.550590
</MAX_BOUND_Y>



```
<MIN_BOUND_X>
116.902081
</MIN_BOUND_X>
<MIN_BOUND_Y>
278.051762
</MIN_BOUND_Y>
</SHAPE>
<SHAPE>
:
:
</SHAPE>
</CAD_TABLE>
```

每一筆<SHAP></SHAP>中記錄一筆 CAD 的資料，亦即每一筆線段、圓或圓弧是以<SHAP></SHAP>做區隔。

4.3.3 OpenGL 繪圖

實作 OpenGL 的範例可以看如下圖例。

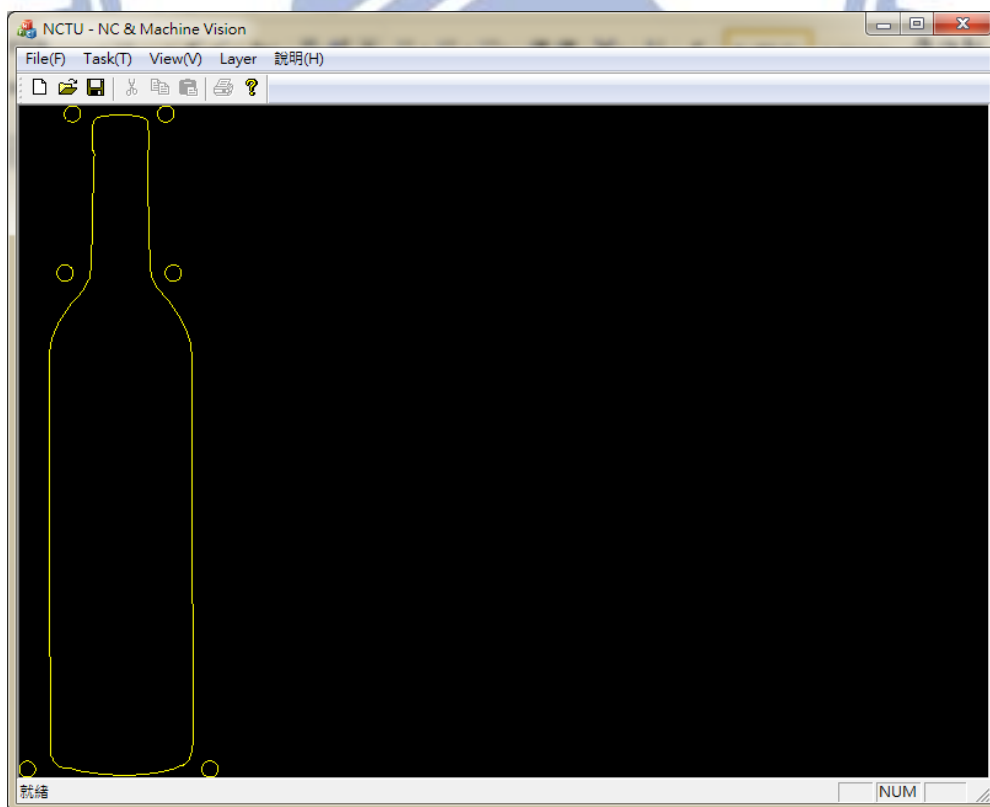


圖 4-6 OpenGL 2D 顯示

顯示圖形主要使用到的 OpenGL 指令有：

- `glClear(GL_COLOR_BUFFER_BIT)`：清除繪圖記憶體
- `glViewport(0, 0, WindowWidth, WindowHeight)`：設定視窗大小，`WindowWidth`是視窗寬，`WindowHeight`是視窗高。
- `glMatrixMode(GL_PROJECTION)`：顯示模式為投影。
- `glLoadIdentity()`：重設 OpenGL 繪圖。
- `gluOrtho2D(gl_Left, gl_Right, gl_Bottom, gl_Up)`：2D 投影實際的視野尺寸大小，`gl_Left`為真實世界左邊座標、`gl_Right`為真實世界右邊座標、`gl_Bottom`為真實世界上界座標、`gl_Up`為真實世界下界座標。
- `glColor3f(1.0,1.0,0.0)`：設定繪圖顏色。

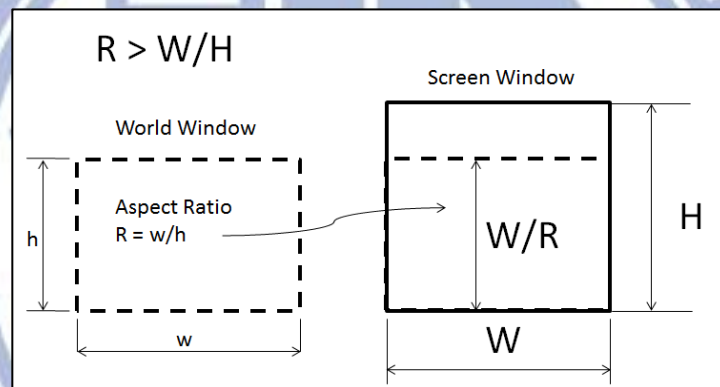


圖 4-7 $R > W/H$

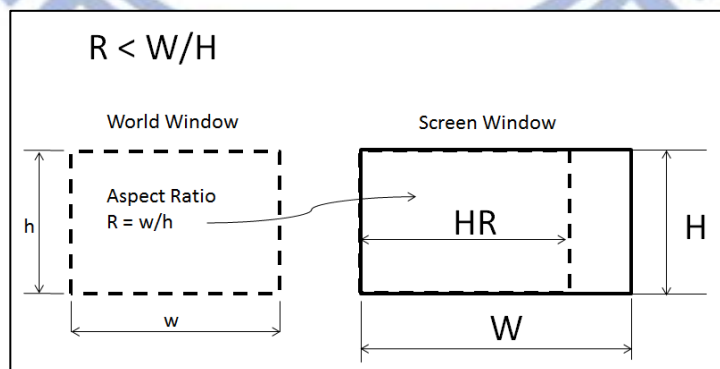


圖 4-8 $R < W/H$

OpenGL 繪圖時實際目視視窗(World Window)的外觀比例(Aspect Ratio)，跟電腦螢幕(Screen Window)的外觀比例通常是不一樣的。當比例不一樣時，OpenGL 的圖形顯示會有比例不同造成的變形。為了維持正常的

視覺比例，需要依照 $R < W/H$ 與 $R > W/H$ 這兩個條件去調整 gluOrtho2D 的輸入值。如圖 4-6 與圖 4-7 所示，代表真實世界視窗與電腦螢幕視窗之間的比例關係。

資料庫定義的 BLOCK BlockBoundary 儲存每一個圖形的矩形邊界，即是圖形最小的矩形外框。將所有的圖形的矩形外框累計並求得最大的邊界矩形，也就是實際目視時窗的大小。

4.3.4 CCD 取像與 OpenCV 定位點辨識

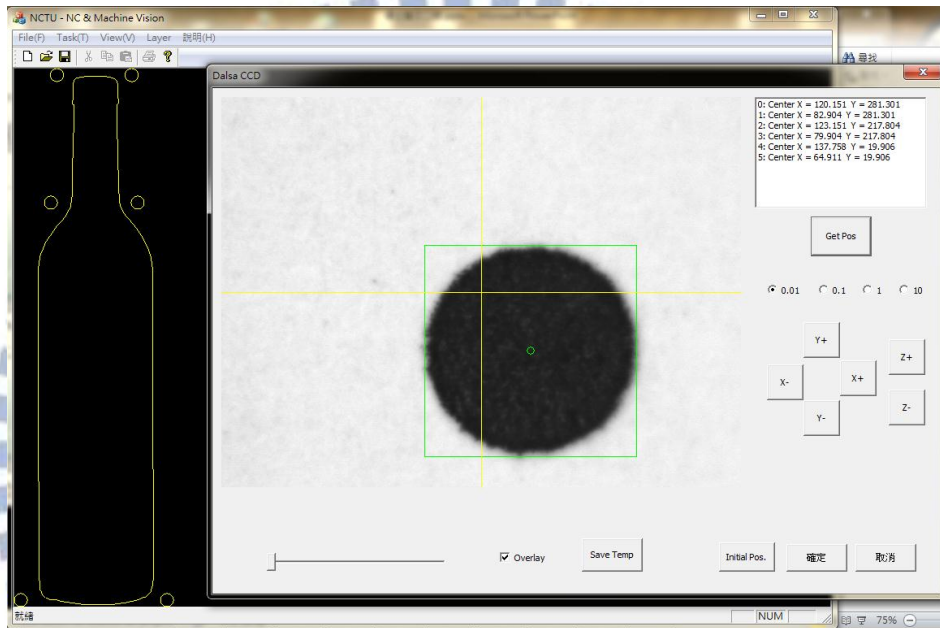


圖 4-9 CCD 取像與樣本比對

圖 4-8 呈現的是 CCD 的取像與樣本比對的結果。圖右上是 CCD 定位點圖層顯示清單。操作時依據顯示定位點，依序比對定位。程式中與原始資料比對得到位移補正值。此補正值稍後再加到 CAD 的圖形座標上。

4.3.5 控制器 SDK

右側圖 4-10 為圖 4-9 CCD 取像與樣本比對對話視窗部分的放大圖。特別取出此部分加以說明，是因為此處應用了新代控制器的整合函式庫 OCAPI。藉由視窗按鈕觸發事件，來執行 OCAPI 的移動指令。

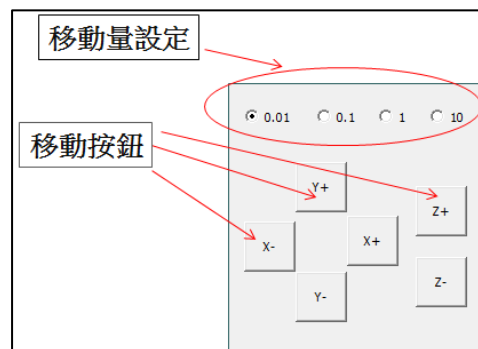


圖 4-10 移動控制介面

由於 CCD 取像時，必須同時取的即時的機械座標，所以必須整合控制器的 API 到加工補正系統中。如圖 4-9 顯示移動控制的界面。移動量設定是設定每一次按移動按鈕時要走的量。設定值從 0.01mm、0.1mm、1mm 到 10mm。X、Y 與 Z 按鈕每按一次即觸發一次移動命令。

4.3.6 刀具與加工參數設定

如圖 4-10 刀具參數設定，顯示刀具設定的對話框，其中的定義如下

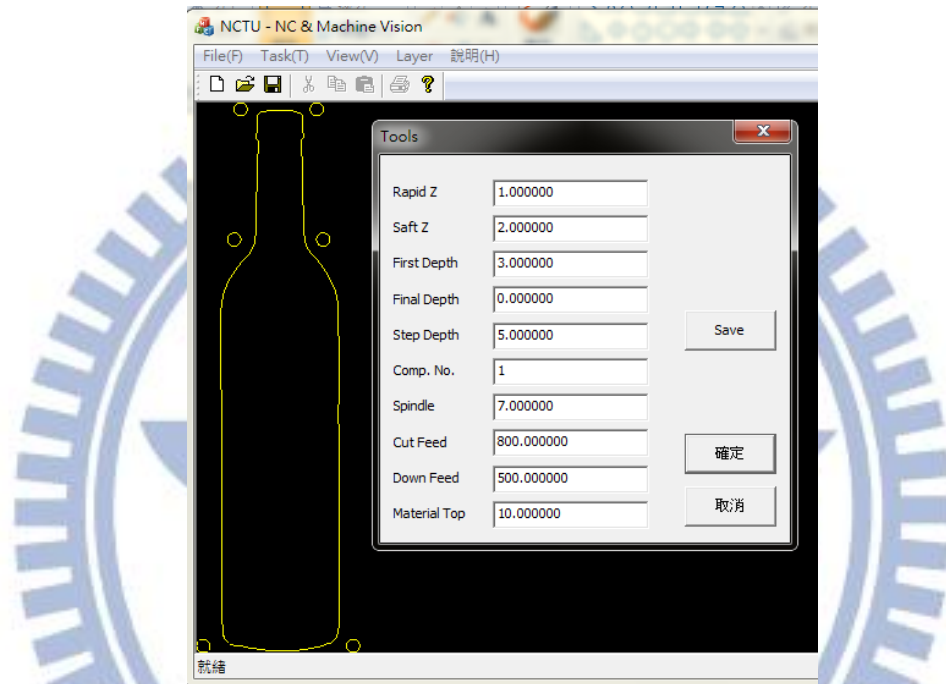


圖 4-11 刀具參數設定

Rapid Z: 快速位移高度，刀具自停止點快速移動到準備加工高度。

Safe Z: 安全高度，低於快速位移高度，高於加工材料高度，以不碰撞為原則。

First Depth：第一刀切削深度。

Final Depth：最終切削深度。

Step Depth：多階切削，每一階切削深度。

Copm. No.：刀具半徑補正號碼

Spindle：主軸轉速

Cut Feed：水平切削進給率

Down Feed：下切進給率

Material：材料高度

4.3.7 NC Code 產生器

```
G91G28Z0
G91G28X0Y0
G54
G00Z50.0
G00X101.306Y280.693
G01Z0.000F500
G01X107.418Y280.553F800
G01X108.206Y280.458F800
:
:
G01X94.100Y280.153F800
G01X94.368Y280.223F800
G01X94.878Y280.292F800
G01X95.394Y280.362F800
G01X97.837Y280.693F800
G00Z50.0
G91G28Z0
G91G28X0Y0
M30
```

圖 4-12 輸出之 NC Code

最終的程序，自 Database 中轉出成 NC 程式，並直接輸出至控制器中。

4.4 圖層管理

如下圖為圖層管理功能，圖層管理功能提供從 DXF 轉進來的資料顯示與整理功能。藉由刪除(delete)按鈕，可以將一些多餘的圖形刪除，降低系統操作的複雜度。

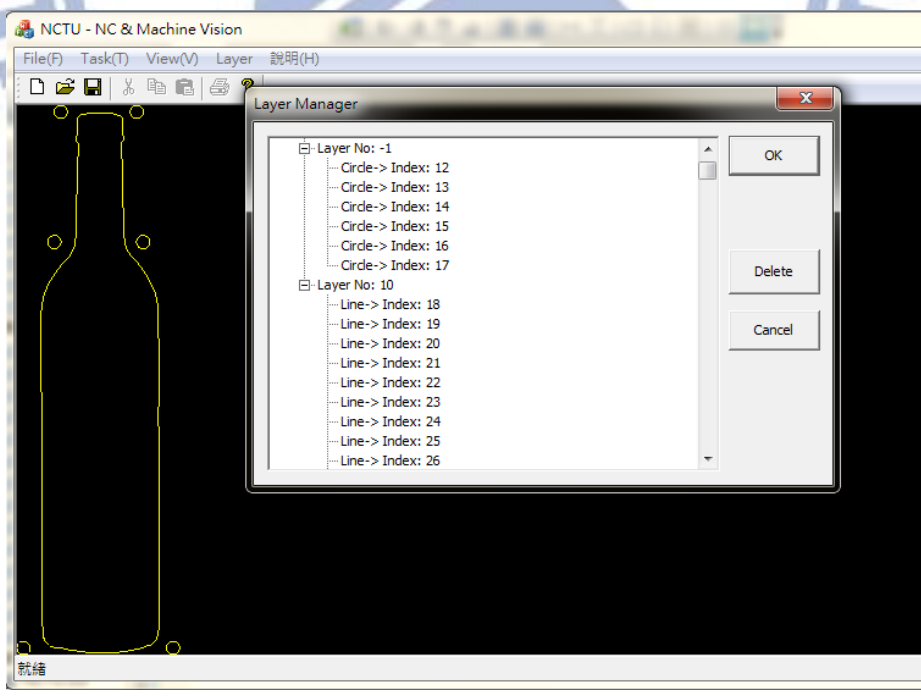


圖 4-13 圖層管理

4.4 研究結果比較

比較模式比對與邊緣偵測尋圓兩種演算法，圖 4-13 為此兩種演算法一般的比對結果圖形顯示。左圖將樣本標示在比對結果的位置上，右圖將尋得的圓型外框顯示在尋得的圖形上。

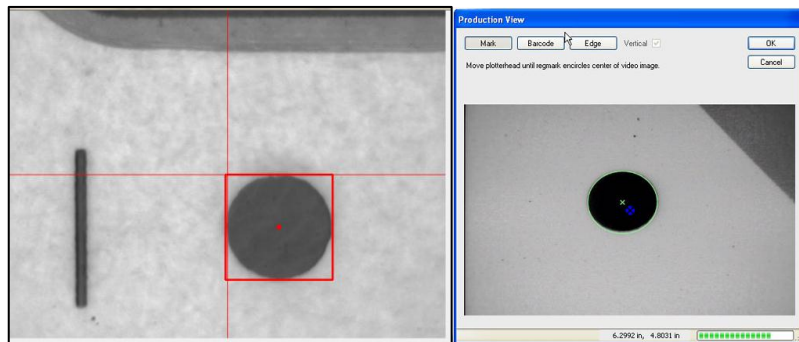


圖 4-14 模式比對與邊緣偵測

下圖 4-14 為 20 個樣本檔的縮圖，圖中的圓型圖形在 X 軸與 Y 軸各偏置了 1mm。利用這 20 個樣本，進行樣本比對與邊緣偵測尋圓，探討此二種演算法的比對差異，與優劣。

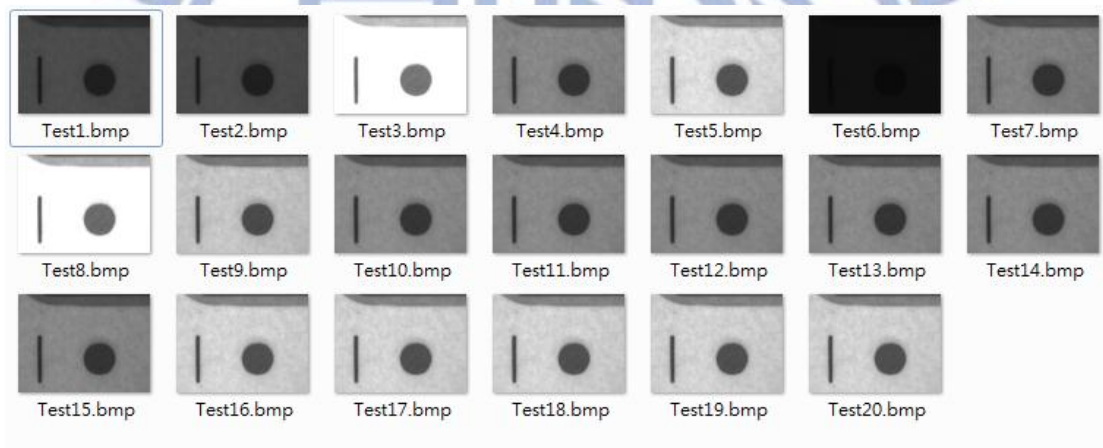


圖 4-15 光線干擾下比對樣本圖檔

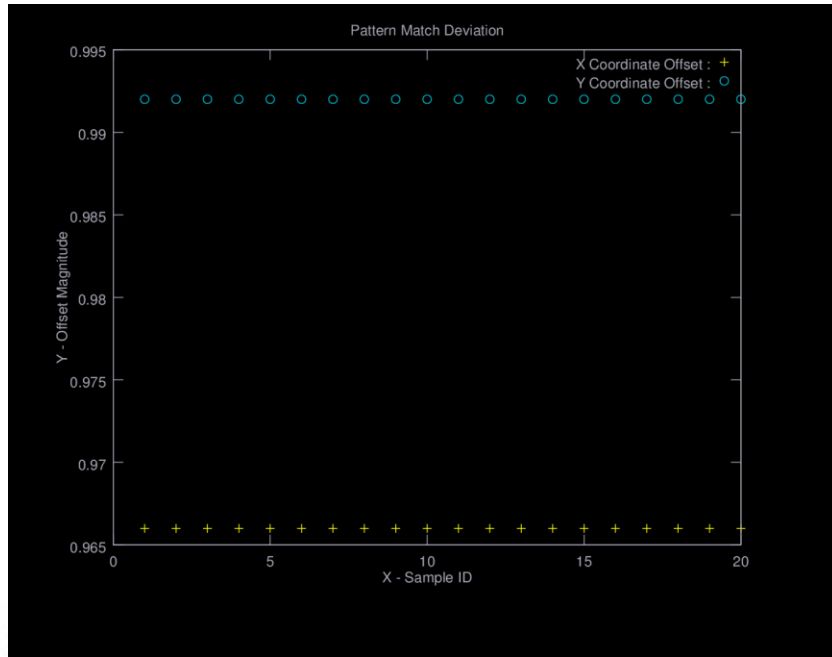


圖 4-16 光線干擾下模式比對結果

圖 4-15 為模式比對結果，X 軸偏置 0.966、Y 軸偏置 0.992，二十個樣本均得到相同的結果，不論樣本的光線明暗度為何。

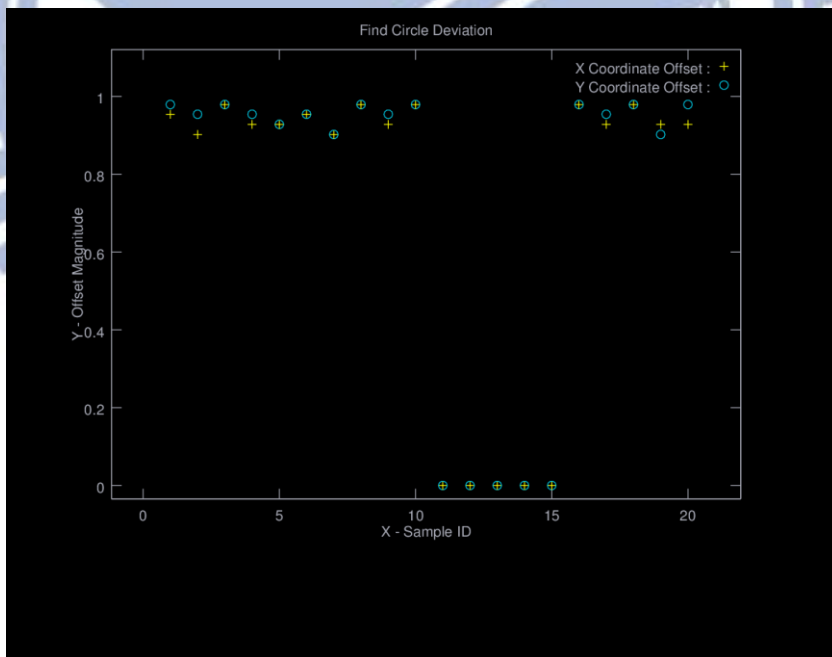


圖 4-17 光線干擾下邊緣偵測尋圓結果

圖 4-16 為光線干擾下邊緣偵測尋圓 20 個樣本的比對結果，其結果是 Test1, Test2 與 Test6 因光線太暗，偵測失敗。偵測成功的樣本 X 軸偏置值為 0.876~0.954mm，Y 軸偏置值為 0.902~0.979mm。

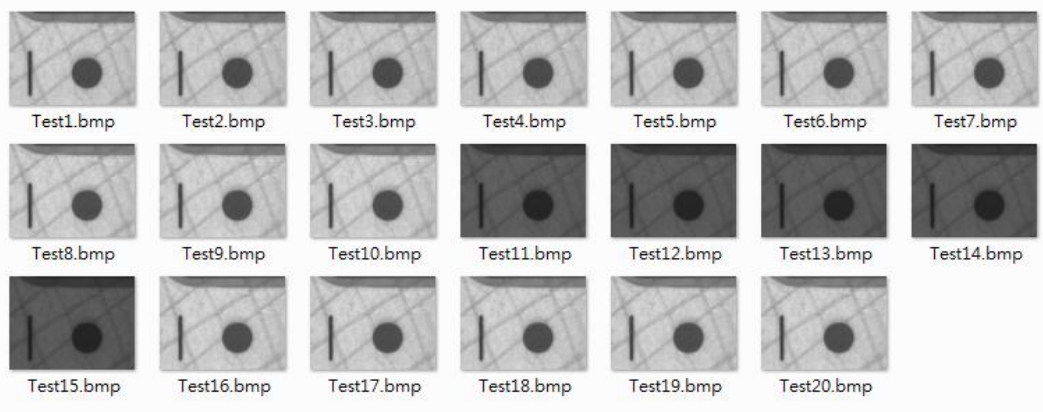


圖 4-18 背景干擾下樣本圖檔

圖 4-17 為 20 個樣本檔的縮圖，圖中的圓型圖形在 X 軸與 Y 軸各偏置了 1mm，並且增強了背景干擾。利用這 20 個樣本，進行樣本比對與邊緣偵測尋圓。

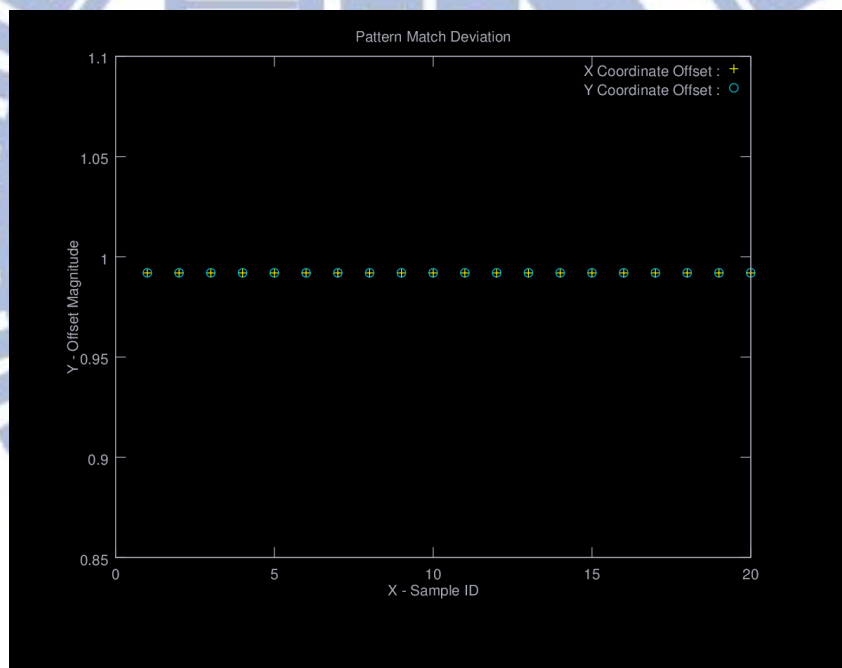


圖 4-19 背景干擾下樣本比對結果

圖 4-18 為模式比對結果，X 軸偏置 0.992、Y 軸偏置 0.992，二十個樣本均得到相同的結果，不論樣本的光線明暗度為何。

以上兩個比對的劇本，均顯示模式比對在對抗光線干擾，與影像干擾上都有較佳的表現。

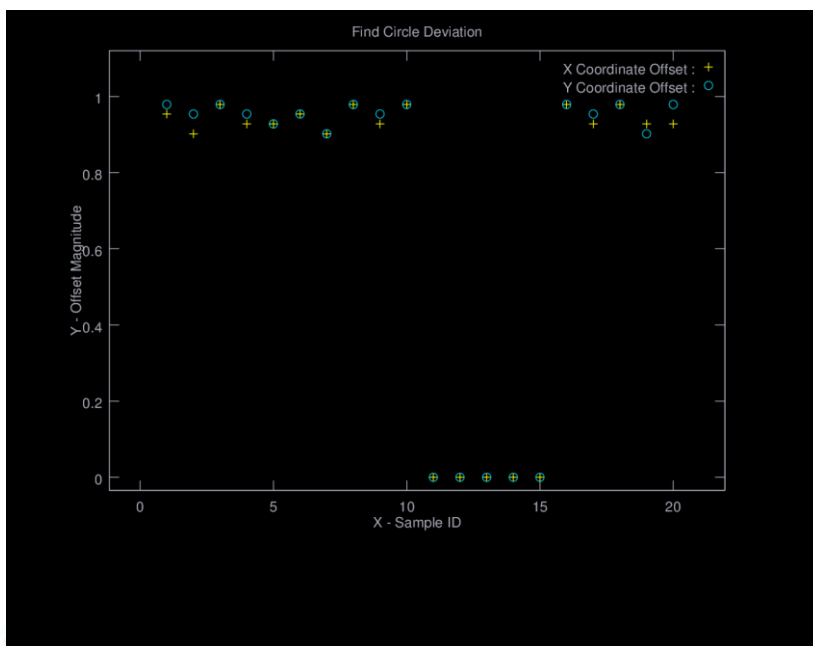


圖 4-20 背景干擾下邊緣偵測尋圓結果

圖 4-18 為光線干擾下邊緣偵測尋圓 20 個樣本的比對結果，其結果是 Test11、Test12、Test13, Test14 與 Test15 因光線太暗，偵測失敗。偵測成功的樣本 X 軸偏置值為 0.902~0.979mm，Y 軸偏置值為 0.902~0.979mm。

第五章 研究結論

5.1 研究結論

本研究的目的是在開發出一個建構在自由軟體(freeware)架構下，解決無治具，工件小變形，小位移，傳統 CNC 加工補正無法滿足的 CNC 切削加工製程。撰寫出一個低邊際成本，抗環境干擾的數控機械的加工補證系統。

本研究總結可以歸納如下：

1. 藉由自軟體 dxflib 函式庫，可以不需要花非常多的心力去開發自己的 DXF 解譯程式，節省相當多的時間。並經由應用 dxflib 的過程中，能理解 DXF 的架構與規格。
2. 經由 STL list 所定義的資料結構，list<CadData> CadDataBase，在實作中比較 STL list 與結構陣列(Structure Array)，前者要比後者應用上要簡單許多。作為 CAD 的資料庫使用 STL list 十分的簡潔而有效率。並且因 STL list 是一個公用的函式庫並不需要除錯的程序，但是自行定義的結構陣列除錯程序是免不了的。
3. OpenCV 是由計算機工業巨擘 Intel 所支持的開放源碼專案。非常多的國際知名的學術研究單位，諸如 MIT、史丹佛大學、哈佛大學...等在機械視覺相關的研究領域均採用這一個機械視覺函式庫做為基礎，從事各種研究專案。本研究也從這一個機械視覺函式庫的應用中得到相關的 CNC 加工應用的相關經驗。並驗證了模式比對與邊緣偵測尋圓的優缺點。驗證的結論證實了當光線明暗變化，與背景干擾增加時，幾乎不影響模式比對的結果，相反的，邊緣偵測尋圓卻造成了極大結果變化，在光源與背景干擾較大的應用上，模式比對的確是上選。
4. 由 SGI 所開發的繪圖函式庫 OpenGL，在本研究程式撰寫過程中花了相當多的時間在處理 Aspect ratio 的問題，當 Aspect Ratio 瓶頸尚未解決時，視窗大小變化呈現出來的圖形，一直出現變形的問題，最終還是導出 Aspect Ration 的演算邏輯。但也發現許多網頁與書本上針對 Aspect Ration 的處理邏輯，事實上存在有許多的謬誤。

5.2 未來的計畫

整個研究完成後，對機械視覺的應用有更深層的認識。基於商用的機械視覺函式庫使用的經驗，感到商用的函式庫，封裝的比較簡易使用，但是也令人對一些演算函式的應用原理，理解不夠深入，而影響系統開發方向判斷的正確性。

經本研究後對機械視覺與軟體架構設計有了較深入的理解，對後續的研究的計畫做了如下的規劃：

1. 計畫在 STL list 所定義的資料結構中，加入資料採礦演算法，加強資料搜尋的效率，擴充系統的應用領域到 CAD\CAM 系統。
2. 從機械視覺研究中引申出另一個機械學習的新領域，對數控機械應用例如刀具自動選用可以將人工智慧功能加入應用中，增加機器使用的方便性。
3. 可以將模式比對的定位應用，延伸應用至高透光性光學壓克力材料的加工，因為此類工件因高光學透光性，背景的干擾源十分強烈，正好符合模式比對的特性。
4. Ink Jet PCB 數位文字圖形噴印機，此類機型並非傳統機 CNC 切削加工機台。而是屬於數位印刷機，其具有無治具特性。需要利用機械視覺對待噴印工件進行定位，以決定起始印刷點。

参考文献

- [1] Gary Bradski , Adrian Kaehler, Learning OpenCV , O'Reilly, September 2008, First Edition.
- [2] Robert Laganiere, OpenCV 2 Computer Vision Application Programming Cookbook , Pocket Publishing Ltd., May 2011.
- [3] G. J. Awcock and Thomas Awcock, Applied image processing, McGRAW-HILL, 1996.
- [4] I.T. Young, J.J. Gerbrands and L.J van Vliet , Fundamentals of Image Processing, Delft University of Technology, 1998.
- [5] National Instrument, IMAQ Vision Concepts Manual, National Instrument, June 2003.
- [6] Roberto Brunelli , Template Matching Techniques in Computer Vision, John Wiley & Sons Ltd, 2009.
- [7] Ramesh Jain, Rangachar Kasturi, Brain G. Schunc, Machine Vision, McGRAW-HILL, 1995
- [8] Nafis Ahmad, A.F.M. Anwarul Haque, “Manufacture Feature Recognition Of Parts Using DXF Files” , 4th International Conference on Mechanical Engineering , Dhaka, Bangladesh, pp VI 111-115, Decmber 26-28 2001.
- [9] J.P. Lewis, “Fast Template Matching” , Vision Interface 95 , Canadian Image Processing Recognition Society, pp.120-123, May 15-19 1995.
- [10] Kai Briechle, Uwe D. Hanebeck, “Template Matching Using Fast Normalized Cross Correlation” , Proc. SPIE 4387, 95, March 20 2001.
- [11] Dave Shreiner, OpenGL Programming Guide, Pearson Education, Inc. 2010
- [12] R. Hewitt, ” Seeing With OpenCV, a Computer Vision Library” , SERVO magazine, 2007, pp. 62 - 66.
- [13] Frank D. Luna, Intrduction to 3D Game Programming with DirectX 10, Wordware Publish, Inc., 2008.

[14] Galina Istatkova, " Object-Oriented Scientific Visualization" ,
International Conference on Computer Systems and Technologies,
CompSysTech' 2003.

