



國立中山大學機械與機電工程學系碩士在職專班

碩士論文

應用多重粒子追蹤功能及雷射光鉗技術之

微流體自動細胞分類/計數系統

Microfluidic Cell Counter/Sorter Utilizing

Laser Tweezers and Multiple Particle Tracing Technique

研究生：林振成 撰

指導教授：林哲信 博士

中華民國 九十六 年 二月

誌 謝

猶記得一年多前，受到恩師教學熱忱的吸引，以及剛踏進生醫機電實驗室時所感受到的那股濃厚的研究氣息。如今，能在有限的時間內完成研究所的學業，這完全得要感謝指導教授 林哲信博士完全投入的無私指導。不僅鼓勵學生積極參與國際性論文期刊的發表，在實驗研究期間遭遇瓶頸時，更能傾聽問題並深入瞭解學生的顧慮，總在與恩師詳談之後，又燃起無限的希望與繼續向前衝刺的決心。

在每一個周末假日裏，當我忙於研究所的學業時，五歲大的宜萱却懂得靜靜陪在身旁，家人的支持著實扮演著最大的精神支柱。感謝內人佳憇的諒解與支持鼓勵、岳母的幫忙以及父母的栽培，你們都是我人生的避風港。在無數的深夜裏，也只有佳憇能體會我所承受的精神壓力，衷心的向妳說聲--謝謝妳。

在此，必須特別感謝中山生醫所 陳和瑟博士及其研究助理正忠於這段期間的協助，並使得活細胞樣本的操控得以持續的改善。感謝中山生醫所所長亦為口試召集委員的 洪文俊教授，於百忙之中撥冗參加學生的口試，並給予寶貴的意見，學生在此特別致上至高的謝意！

在實驗期間，沒有致暉的幫忙研究將無法繼續下去。此外亦必須感謝育生、敬修、世偉、鈞賢、承諺、志鵬、靖惠、書銘、岱洋、維仁以及助教宜莉於此期間言語上的鼓勵與實質上的幫助，也希望你們於日後的人生道路上，不管是事業或是繼續求學，都將更加一帆風順！

振成 于高雄中山大學

Feb.13 2007

目 錄

目 錄.....	I
圖目錄.....	V
表目錄.....	IX
中文摘要.....	X
Abstract.....	XII
第一章 緒論.....	1
1.1 前言.....	1
1.2 研究動機及目的.....	2
1.3 文獻回顧.....	4
1.3.1 粒子計數器.....	5
1.3.1.1 柯爾特粒子計數器(Coulter particle counters, CPC).....	5
1.3.1.2 微型光學偵測裝置之粒子計數器.....	7
1.3.2 流式細胞偵測技術.....	8
1.3.3 整合型微流體細胞分類系統.....	11
1.4 本文架構.....	13
第二章 全自動分類系統之基礎理論.....	15
2.1 影像擷取理論.....	15
2.1.1 影像擷取系統組成.....	15

2.1.2	影像運算與影像比對法則及原理.....	16
2.1.3	常規化相關匹配法(Normalized Cross Correlation, NCC)	17
2.2	光鉗之基本原理.....	19
2.3	微管道液態流體系統.....	24
2.3.1	電滲透流聚焦.....	25
第三章	全自動細胞分類控制系統整合與實驗架設.....	27
3.1	視覺系統軟硬體架構.....	31
3.1.1	視覺系統硬體架構.....	31
3.1.2	視覺系統影像辨識演算法.....	32
3.1.3	多重質點細胞影像辨識.....	35
3.1.4	細胞質點移動位置及速度之計算及應用.....	38
3.1.5	細胞質點即時粒子計數之概念及其方法.....	41
3.2	PC-Based 電腦控制及自動分類系統人機介面操作	43
3.2.1	PC-Based 電腦控制系統.....	43
3.2.2	自動分類系統人機介面操作說明.....	44
3.3	雷射光鉗光學系統.....	46
3.4	微流體晶片系統.....	47
3.4.1	微流體晶片設計.....	48
3.4.2	微流體晶片製程.....	51

3.4.3 晶片測試.....	53
第四章 結果與討論.....	55
4.1 微流體樣本流聚焦情形之數值分析與實際實驗結果圖	55
4.2 雷射光鉗有效捕捉力與粒子位移關係之實驗量測結果	57
4.3 樣本流聚焦寬度之控制.....	58
4.4 即時辨識分類操作畫面.....	59
4.5 影像即時辨識處理.....	60
4.6 不同目標粒子之混合樣品流分類結果.....	62
4.7 使用軟體偵測方式之粒子計數器.....	66
4.8 軟體計數器之性能分析.....	69
第五章 結論與未來展望.....	73
5.1 結論.....	73
5.2 未來展望.....	73
參考文獻.....	75
Appendix A. Image Capture Sub-Program.....	78
Appendix B. Image Display Sub-Program and Image Recognition Sub-Program	81
B.1 Program Initial.....	82
B.2 Image Display and Main Recognition Function	95

B.3 White Particle Software Counter Sub-Function.....	101
B.4 Black Particle Software Counter Sub-Function	103
B.5 Laser-Output Shutter Control Sub-Function	104
B.6 Laser-Closed Shutter Control Sub-Function	105
B.7 RS 232 CommPort Available Check Sub-Function	105
B.8 Miscellaneous.....	106

圖目錄

圖 1.1 柯爾特粒子計數器示意圖.....	6
圖 1.2 微型光學偵測裝置之粒子計數器示意圖	8
圖 1.3 利用流式細胞偵測技術(Flow Cytometry)以及“Imaging in Flow”或是 稱為”Flow Cytometric Imaging”取像方法，進行海洋有機浮游生物辨識、分 析及計數.....	10
圖 1.4 Optoelectronic Tweezers(OET)光電子鉗細胞分類器之工作原理示意圖	12
圖 1.5 利用雷射光鉗進行微流體細胞分類的操控	12
圖 2.1 樣本影像(Image Pattern)矩陣搜尋比對影像平面示意圖	19
圖 2.2 雷射光鉗聚焦原理示意圖.....	21
圖 2.3 光鉗作用力原理示意圖.....	22
圖 2.4 雷射光束入射透明球體之幾何光學模型示意圖	23
圖 2.5 電滲透流流場速度分佈示意圖	26
圖 3.1 微流體細胞分類系統之流體聚焦、軟體計數器功能配置、雷射光鉗 作用區及光學取像設計概念示意圖	28
圖 3.2 光鉗系統實體圖.....	28
圖 3.3 自動控制系統整體架構示意圖	30
圖 3.4 全自動光鉗操控之細胞分類系統各子系統之相互關係關聯作用圖	

.....	31
圖 3.5 利用動態連結函式庫 DLL(dynamic link library)的方式於記憶體開啟 一共用資源.....	32
圖 3.6 本研究影像辨識及分類計數系統控制流程圖	33
圖 3.7 “樣式比對(Pattern Match)”作動圖(a) 欲進行搜尋比對之影像樣本(b) 多質點影像辨識比對搜尋結果.....	36
圖 3.8 “簡易物件法(EasyObject)”作動圖(a)針對混合樣本進行目標質點粒 子分類示意圖(b)多質點影像辨識比對搜尋結果.....	38
圖 3.9 即時細胞粒子質點位置追蹤及移動速度計算結果(單位為 pixel)...	41
圖 3.10 粒子計數偵測位置之設計概念示意圖	42
圖 3.11 粒子計數偵測迴路設計概念示意圖.....	42
圖 3.12 PC-Based 電腦控制系統與周邊介面之控制區塊流程圖	44
圖 3.13 全自動光鉗操控之細胞分類系統(a) 即時影像辨識人機介面主畫面 及(b)即時影像辨識設定畫面.....	45
圖 3.14 微管道晶片之設計與製程流程圖示意圖	47
圖 3.15 (a)傳統型細胞分類器及(b)加入新式改良擴張閥設計之細胞分類器 之尺寸大小示意圖.....	48
圖 3.16 管道晶片設計示意圖.....	50
圖 3.17 光切換微流體細胞分類晶片製程示意圖	53

圖 3.18 (a)加入擴張閥設計之新式改良型晶片完成接合後之實體圖(b) 擴張閥結構顯微近視圖	53
圖 4.1 (a)(b)舊有光罩設計數值模擬計算結果與實際流場聚焦流線之比對。 (c)(d) 加入新式擴張閥之改良型光罩設計數值模擬計算結果與實際流場聚焦流線之比對。	55
圖 4.2 雷射光鉗捕捉之最大側向位移量測方法示意圖	56
圖 4.3 雷射光鉗捕捉之最大側向位移實際觀察量測影像圖	56
圖 4.4 於 80 $\mu\text{m/s}$ 、160 $\mu\text{m/s}$ 、320 $\mu\text{m/s}$ 等三種不同流速狀況下之雷射光鉗捕捉功率與最大側向位移關係圖.....	58
圖 4.5 使用 30 M Ω 高阻抗可變電阻進行樣本流聚焦寬度調整電路示意圖	59
圖 4.6 使用 30 M Ω 高阻抗可變電阻進行調整後之樣本流聚焦寬度，其寬度大約為 15 μm ~20 μm 之間.....	59
圖 4.7 本研究所自行開發之細胞/微粒子影像自動分類系統之人機操作介面 HMI (Human-Machine Interface).....	60
圖 4.8 電腦即時影像辨識處理之(a)灰階畫面與(b)二值化比對圖	61
圖 4.9 以 10 μm 為目標粒子之自動光鉗粒子分類連續圖像.....	63
圖 4.10 以 5 μm 為目標粒子之自動光鉗粒子分類連續圖像.....	65
圖 4.11 軟體計數器之灰階值掃瞄偵測作動連續示意圖	67

圖 4.12 使用自行開發之軟體計數器於微管道內之計數結果	68
圖 4.13 連續兩張取像之細胞質點影像恰都位於偵測線之兩側時，軟體計數器偵測線將無法感測到細胞質點通過時之灰階值變化。	69

表目錄

表 3.1 聚焦平凸透鏡 C260TM 光學特性規格表.....	47
表 4.1 於 1 sec 內軟體計數器偵測線所連續掃瞄到之灰階值變化以及影像辨識系統於連續兩張取像時間差之記錄值	71

中文摘要

本研究為一整合型細胞/粒子分類及計數之生醫光電系統，該系統整合即時多重粒子數位影像擷取及辨識系統，並利用紅外光光鉗控制及電驅式微流體晶片進行粒子補捉及分類。此研究提出利用影像控制光鉗之自動控制系統，其將藉由自行撰寫之數位影像辨識及追蹤軟體對微流道之粒子進行即時影像特徵之辨識及粒子計數。與傳統影像辨識系統之單一粒子辨識不同處在於本研究將同時對流道內之多個粒子進行影像辨識，並同時追蹤其流動軌跡來計算其行進速度，並藉由電腦判斷光鉗是否作動，及其作動之精確時機以及作動之時間長短，因而可以對微流道中之粒子進行連續式偵測、計數及分類。

本實驗所開發之微流體細胞分類器，其製作之微管道晶片皆採用成本低廉之鈉玻璃製程，利用電滲透流在十字管道內聚焦並驅動粒子，以數位攝影機，捕捉微流體管道內粒子流動之即時影像，配合實驗室開發之影像辨識軟體來對粒子進行判讀，並利用雷射光鉗改變目標粒子流線，達到分離粒子之效果。結果顯示，光鉗可成功的在流速為 $300 \mu\text{m}/\text{sec}$ 的 $5 \mu\text{m}$ 與 $10 \mu\text{m}$ 之混合粒子流中，經由影像辨識系統，成功辨識並啟動光鉗，使 $5 \mu\text{m}$ 以及 $10 \mu\text{m}$ 之微粒子產生約 $20 \mu\text{m}$ 之側向位移，流向後端收集槽儲存，達成分離與計數。

本研究所提之創新方法，係一非接觸式之微粒子操控法，其可容易地

被運用於樣品中特殊致病性細胞之捕捉、分類及計數，並且免除接觸式操作可能發生之交互感染，並因此可解決微全程分析系統領域內的細胞/粒子操控問題。

關鍵詞：微流體、光鉗、數位影像辨識、電滲透流、細胞分類/計數



Abstract

This study proposes a novel microfluidic system based on a computer controlled digital image processing (DIP) technique and optical tweezers for automatic cell/microparticle recognition, counting and sorting in a continuous flow environment. In the proposed system, the cells/microparticles are focused electrokinetically into a narrow sample stream and are then driven through the region of interest (ROI), where they are recognized and traced in real time using a proprietary DIP system. Synchronized control signals generated by the DIP system are then used to actuate a focused IR laser beam to displace the target cells from the main sample stream into a neighboring sheath flow, which carries them to a downstream collection channel where they are automatically counted. The proposed approach makes possible the continuous sorting and counting of cell samples without the need for any moving parts or embedded transducers. The experimental results show that the proposed system is capable of sorting 5 μm or 10 μm PS bead from a mixture of 5 μm and 10 μm samples in the flow speed 300 $\mu\text{m}/\text{sec}$. The proposed system provides a simple, low-cost, high-performance solution for cell manipulation in microfluidic devices.

Keywords: microfluidics, microparticle, digital image processing, electrokinetic focus, optical tweezers

第一章 緒論

1.1 前言

在生醫檢測或是細胞分析等研究中，粒子分類在微觀操作下一直是重要的研究方向。在大部分細胞分析的研究上，對於特定的細胞或是構成細胞之生物分子的分離及計數有其必要性。對於特定細胞或生物微小分子的分離及計數，至今已有多位學者提出許多種方法來對細胞及生物分子進行分類及計數。然而近年來，由於微機電製程技術的進展，使得許多原本龐大之元件得以微小化，而開始有學者利用此一技術，從事以往所無法進行之實驗，如細胞之操控以及蛋白質或 DNA 之分離等。而在眾多微機電研究領域中，將微流體元件應用於生醫檢測尤其受到重視[1]。其藉由微機電製程技術所生產之微流體生醫檢測晶片，不但具有高檢測效能、低樣品消耗量、低消耗能源、體積小以及成本低等優點，且其在整合微流體、即時反應以及同步分析之微全程分析系統中，具有不可忽視之發展潛力。且在生化反應的過程當中，不外乎是需要藉由一些外力來操控微流道中之生物樣本，使其於微尺度的空間裡進行樣本傳輸、混合、微閥開關或者是分類等工作。而現今大部分細胞分析的研究上，對於特定的細胞或是構成細胞之生物分子的分離及操控是有其必要性的。

此外，隨著電滲透流在各領域的應用快速發展[2]，以及電滲透流流速穩定、粒子在其中以等速度移動等特性，其應用於微管道內流體之驅動也

十分合適。因此本研究是利用光鉗非接觸式作用力的特性，並配合電驅動流體聚焦方式在微管道晶片底下進行粒子分類操作。

1.2 研究動機及目的

由於微機電系統(Micro-Electro-Mechanical-Systems, MEMS)製程技術具有將傳統大型生化分析或檢測設備微小化之能力，因此目前在醫學和生物細胞的研究上，微型化之流式細胞/顆粒計數器及分類感測器已廣泛地應用在生醫等相關之研究領域。除縮小其尺寸外，該微型化之流式細胞/顆粒計數器及分類感測器更可增加其分析速度與簡化人為操作程序，因此在分析化學與生物醫學方面之應用深受矚目。利用微系統技術將分析儀器微型化並整合於一小面積之生醫晶片上，不但可大幅降低檢測之成本、增加檢測分析之速度，另外也大幅增加偵測之靈敏度。

目前吾人所熟知的微粒子操控方法雖可達到分類、收集等功能，但各自仍具有其缺點與不便性，因而使得晶片系統的整合困難，工作效率不佳及整體製程造價昂貴。早期的微流體細胞分類裝置由於只能單獨使用雷射光鉗作為唯一的力量來攫取細胞粒子，並經過長距離的拖曳之後才能成功地將細胞粒子分類出來[3]。此方法雖可以精確地分類出目標細胞粒子，但必須以人工的方式，判斷細胞粒子的大小、形狀以及計算已分類的細胞數量。運用此方法分類的速度較緩慢，只能適用在少量(<100 pcs)的細胞粒子分

類，無法應用在較多數細胞粒子的連續分類需求上。

有鑑於此，本研究整合了一種非接觸式之微粒子/細胞分類法。該系統將結合雷射光鉗及微流體晶片。該晶片具備一聚焦機構可以進行流體聚焦，使樣品流中之粒子得以依序排列經過雷射光鉗的作動區來達到操控的目的。本實驗利用數位影像處理的方式，可以針對不同大小及形狀的細胞粒子，同時進行連續式的位置追蹤及相似度的比對，進而求出所欲分類之目標細胞粒子的瞬間速度及外形相似度。在通過雷射光鉗作用區之前，最終之瞬間速度將被擷取，並作為雷射光鉗作用時間長短之自動調整參數。

本研究目的為利用微機電系統製程技術，設計及製作微型流式細胞/顆粒計數及分類感測器(Micro flow cytometers)，結合螢光及雷射夾鉗之運用，以進行微流道晶片內之螢光蛋白質體之追蹤及分類等應用。其原理是利用流體於微流道內流動之特性及兩側快速的邊鞘流體之作用，將細胞聚集於中央特定位置，再利用螢光辨識方法偵測出有無轉殖成功之細胞，經雷射光鉗之捕捉，改變其細胞之行進流線來達到分類的目的。同時運用模擬光纖偵測器的原理，來計算出細胞之數量。並配合後端之電場及流場控制微流道內之無閥開關，以將成功轉殖之細胞收集至預定之樣品槽中，並藉此分離特定之細胞樣本。所欲達成之目標在於結合光鉗與螢光之微流體細胞分類裝置，架構一可適於無菌操作的雷射光鉗細胞分類系統，並應用於基因轉殖細胞之篩選。做為後續選擇轉殖細胞活性之分析，並建構螢光

細胞資料庫 (GFP-, RFP- HeLa cells, AGS cells etc.)。期能使之合適並應用於生物醫學之相關領域。

1.3 文獻回顧

過去的文獻曾有利用不同切換機構的晶片設計來進行粒子分類[4-6]。利用傳統的螢光激發細胞分類器 (FACSS) [7]，由於其分離之效率相當高，所以已廣為大眾所使用。此外，在微管道內利用微小電極組的設計，縮小其電極之間距，即可在高頻率和低電壓下產生非均一性電場，利用細胞在電場內被感應出不同的介電泳效應，而產生吸附或排斥的原理來分類細胞 [8]。或是，在微管道中利用壓電材料產生超音波進行微粒子在流體中分類的操控 [9, 10]。在計數器的應用上，有些學者利用在晶片上蝕刻出的微管道，將光纖伸入其管道內，以作為傳輸及接收雷射光的元件，藉此來對生物樣本進行計數，此舉手續亦相當繁雜且無法滿足量產的需求 [11]。還有學者利用昂貴的鍍膜製程，在晶片上製作電極，當粒子流過時，藉由阻抗的變化訊號來產生計數 [12]。然而，上述一些研究，在結構與製程手續上較為複雜，成本昂貴，使其成為商業化或大量生產之拋棄式晶片之可能性降低不少，因此並不適合製成可拋棄式晶片。過去雖有文獻進行雷射光鉗與微流體系統的整合應用 [3, 13]，但其所展示的實驗多是單一顆粒子的流動方向改變，至今尚未有一個真正連續且自動化的系統被發展出來。而利用光鉗 (Optical tweezers) 操作微管道中的粒子，甚至操作活細胞，在早期的研究文

獻[14-16]，已被證實其可行性；同時，這些實驗也展現在光鉗操作乾淨時無污染的特性。因為光鉗為一主動且非接觸操作，在晶片管道設計上，可採用較簡單的被動式晶片設計，不僅減低了結構的複雜度，同時可降低成本。Wang 等人於 2004 年，利用光學感測的方式來偵測被螢光標的的粒子，成功的開發出一連續式且高產量（high throughput）的粒子分類系統[17]，但卻無法針對不同大小及形狀的粒子進行連續式的分類及計數。

而早期為了穩定測量計數與分類生物細胞，均需使用大型傳統的儀器來進行分析的工作，其所利用的技術稱為流式細胞技術(flow cytometry)。流式細胞技術主要是利用雷射螢光激發的方式，快速偵測一顆接著一顆流動於液體水柱（fluid stream）中的顆粒或細胞，收集並統計被雷射光激發後所產生的光學訊號，再轉換成電子訊號並經由電腦分析細胞或顆粒的特性。流式細胞技術至今已是一項非常成熟且穩定的技術[18]，但其設備均需專人操作且費用昂貴[19]。

1.3.1 粒子計數器

1.3.1.1 柯爾特粒子計數器(Coulter particle counters, CPC)

在 1950 年代，Coulter 發現當血球細胞經過孔洞（orifice）時，會因排擠等張生理食鹽水溶液（isotonic saline solution）而提高電阻（electrical resistance）。如圖 1.1 為柯爾特粒子計數器(Coulter particle counters, CPC)之

操作原理示意圖。且藉由此原理，Coulter 設計製造了一種量測儀器，可以在液態流體系統(liquid system)內計數流動的細胞。大的細胞體積較大，所排出的等張生理食鹽水也較多，因此產生的電阻值亦將大於較小的細胞經過孔洞時產生的電阻。藉由觀察電阻值的改變次數以及大小之變化，即可量測並記錄通過的細胞數目以及細胞的相對大小及尺寸。因為此儀器可提供血液中紅血球及白血球的數目自動化的計數，在當時很快就變成醫院血液科不可或缺的機器。

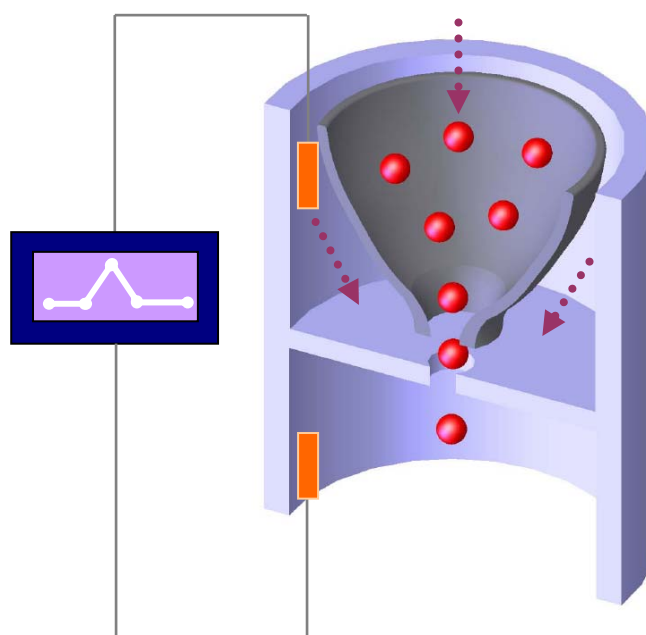


圖 1.1 為柯爾特粒子計數器示意圖。

由於此設備具備了流式細胞儀的主要特性：(1)可以使單一細胞一個接一個排列並快速通過孔洞，(2)以電子訊號來偵測細胞以及(3)具有自動訊號分析的功能。這些特性成了日後流式細胞儀發展演進的重要基礎。

然而，當時的技術無法完全確保液體在流動時，細胞能保持單一個接一個的順序，時常會有二個以上的細胞同時通過一個觀測點，影響量測訊號。因此 Crosland-Taylor 利用層流的原理設計了一個水流系統 (flow system)，使細胞能在水柱的中央內一個接一個地流動，而不會有二個以上的細胞同時通過觀測點的干擾現象，此即所謂的‘hydrodynamic focusing’系統。此系統亦成為日後大部份微流體裝置設計所必須整合的主要項目之一[20]。

1.3.1.2 微型光學偵測裝置之粒子計數器

細胞計數器乃現今生醫及環境檢測上所不可或缺之設備，但現行之設備均屬大型系統，該系統體積龐大且複雜度高，因此價格昂貴。利用微機電系統製程技術可將傳統大型流式細胞/顆粒計數及分類感測器微小化。

流式細胞/顆粒計數及分類感測器的操作原理則是利用水力所引發的流體聚焦效應(Hydrodynamic focusing effect)而進行細胞計數及分類。其基本結構是由三條平行進樣管道、噴嘴及集中流管道組成(如圖 1.2)。樣品由中間管道注入，兩側管道則注入邊鞘流體。若適當地控制各管道流體的注入流速，兩側管道流體在噴嘴口擠壓中間樣品流，使中間流體寬度變窄，兩側邊鞘流流速越快則集中效果愈佳。在適當的邊鞘流流速之下，可將樣品流之寬度縮減至單一細胞之尺寸。

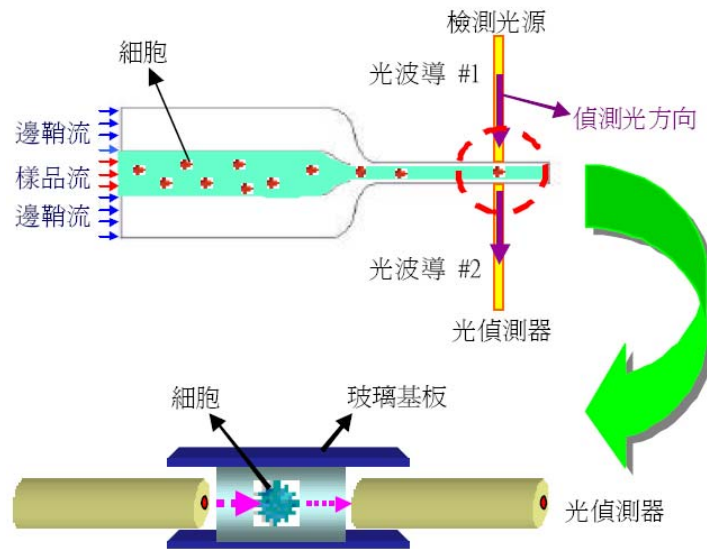


圖 1.2 微型光學偵測裝置之粒子計數器示意圖[21]。

於 2003 年，Lee 等人即利用上述之原理，整合微機電系統技術將上述微流體結構微型化，並將光學檢測機構整合於該微流體細胞計數器中，而免除傳統細胞計數器龐大複雜之光學元件，製作出輕薄短小之微型流式細胞/顆粒計數及分類器[21]。該細胞計數器利用微影技術於玻璃上製作光纖軸通道及光波導結構，檢測光可在其中以全反射進行傳遞而不衰減。當細胞流中之細胞通過該光波導檢測區時，將對檢測光造成阻擋或散射而造成光強度之變化，其光訊號可由另一端之光波導結構收集而加以檢測。該研究並成功地對微顆粒及全血進行流體聚焦並以該光學檢測機構完成細胞計數之功能。

1.3.2 流式細胞偵測技術

流式細胞技術是一種快速量測的專門科學。其主要運用於特殊細胞/質

點於連續流體內流動時的特性量測。此項技術最重要的特性在於其可於懸浮液中分析並操控單一細胞/質點。流式細胞技術不僅可與弧燈搭配運用，更可與雷射技術結合，並利用其對光線的散射及螢光的特性，進行多重參數的同時量測。目前此技術已被廣泛運用於生物學分析及醫學檢測等商業性用途上，並做為細胞/質點的純化分類及特性分析[22]。

於 1934 年，Moldavan 建議將靜態顯微鏡 (static microscopy) 轉換成流式系統 (flowing system)。他設計了一種顯微儀器，使得通過毛細管的白血球及被中性紅 (neutral red) 染上顏色的酵母菌細胞可以被觀察。為了記錄通過顯微鏡下的細胞形狀及數目，儀器還加上了光偵測器 (photodetector)。此種觀念後來形成了今日流式細胞儀發展的雛型，可以認定為是流式細胞技術的啓始。

於 1994 年，Wietzorrek 等人利用流式細胞取像技術[23]，進行海洋有機浮游生物辨識、分析及計數。如圖 1.3 所示，其所開發出之新式方法稱為“Imaging in Flow”或是稱為“Flow Cytometric Imaging”。其提出一新式概念，利用流式細胞偵測技術(Flow Cytometry)於光學感測區(Optical Sensing Zone)收集生物樣本之螢光散射訊號，若能即時分析其分佈之參數特性，則可以立即知道生物樣本之種類，並可計算光學感測區(Optical Sensing Zone)至取像區(Imaging Zone)之距離與樣本流速，即可於適當之時機觸發 CCD 進行取像並控制影像曝光時間，以便進行後續之樣本分析與計數。

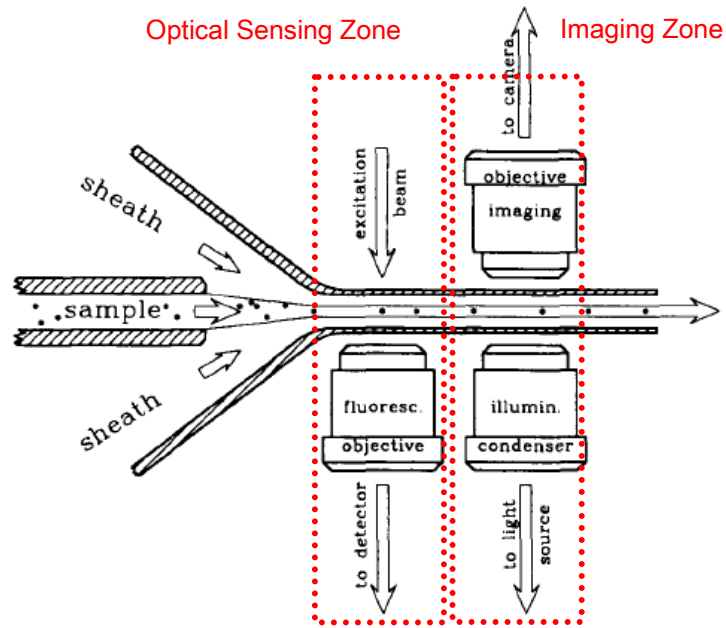


圖 1.3 利用流式細胞偵測技術(Flow Cytometry)以及“Imaging in Flow”或是稱為”Flow Cytometric Imaging”取像方法，進行海洋有機浮游生物辨識、分析及計數[23]。

在當時由於顯微鏡取像景深較淺，且影像處理與曝光時機不易控制，因而影像取像之品質不佳且分析難度較高。但時至今日已可利用立體顯微鏡來增加聚焦深度，且影像辨識演算技術已相當成熟並可於數十微秒即時完成取像及辨識，當初 Wietzorrek 等人所遭遇的問題已可獲得改善。

而本研究將利用此一流式細胞技術之概念，開發出一套整合雷射光鉗技術、影像辨識追蹤技術以及微機電製程技術之即時辨識，分析及計數之自動分類控制系統。

1.3.3 整合型微流體細胞分類系統

在 1970 年代，Ashkin 等人首先提出利用光壓(Optical Pressure)進行微小粒子操控的概念。

時至今日，愈來愈多的學者陸續提出許多創新的方法，進行微流道內生物樣本之操控。有些學者運用光電子鉗 Optoelectronic Tweezers (OET)的原理，利用光誘導介電泳(Light-induced dielectrophoresis)的方法，於靜電微電極產生非均勻性的電場，利用細胞在電場內被感應出不同的介電泳效應 (dielectrophoresis, DEP)，而產生吸引或排斥的原理,來對微管道內的生物樣本進行分類之操控。或是，運用微機電製作的技術將電磁元件整合至微流體管道內，利用電磁場的效應進行微粒子/細胞分類的功能[24]。在此，將針對較特殊的操作方法進一步的詳述如下：

於 2005 年 8 月，Chiou 等人提出一種 Optoelectronic Tweezers(OET)光電子鉗細胞分類器[25]。圖 1.4 說明其基本的工作原理。其主要是利用影像分析軟體分析微管道內之生物細胞樣本，藉由光電子鉗光學的虛擬電極方式產生介電泳作用力，並利用馬達持續定速移動 OET 裝置平台，進而對散佈之細胞樣本產生分類之效果。由於其虛擬電極樣式(Pattern)產生的更新速率目前只能達到 5 frames/sec，若平台移動的速率大於 15 $\mu\text{m}/\text{sec}$ 將產生遲滯的操作現象。因此目前只能達到 120 beads/min 的分類效能。

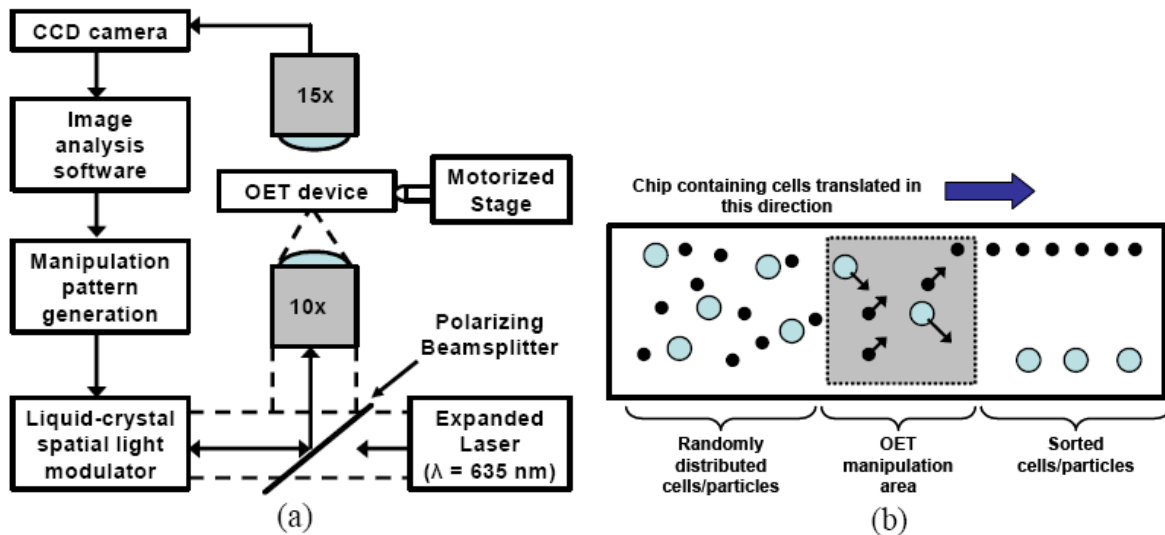


圖 1.4 Optoelectronic Tweezers(OET)光電子鉗細胞分類器[25]之工作原理示意圖(a)即時影像分析迴授控制與(b)分類操作方法。

Enger 等人亦於 2004 年，利用雷射光鉗進行微流體細胞分類的操控研究[3]。如圖 1.5 所示，該研究利用電驅動的方式驅動流體。其首先利用雷射光鉗對微流道內的酵母菌進行捕捉，將其搬運至側管來進行分類及收集的工作。此一研究所提出之方法，由於一次僅能搬運一個生物樣本，因此其分類之效能並不高，且無法進行長時間之連續式分類。

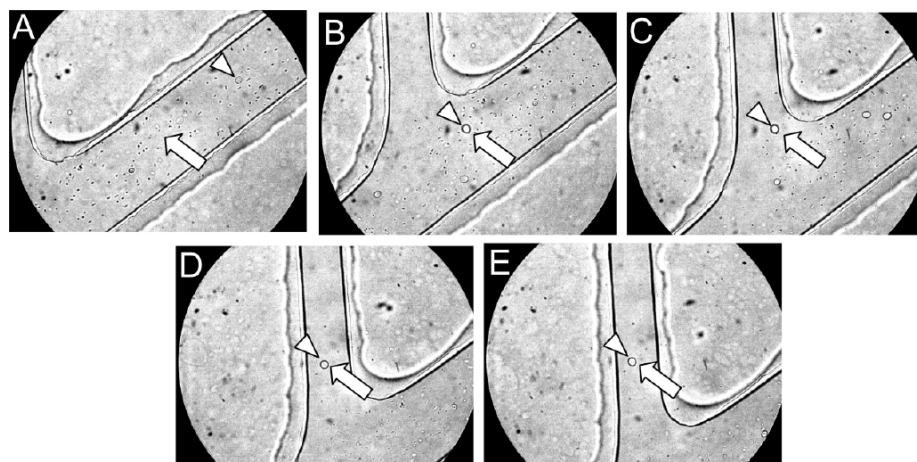


圖 1.5 利用雷射光鉗進行微流體細胞分類的操控[3]。

1.4 本文架構

本研究的主要目的是為了開發出一套結合影像辨識追蹤功能及結合雷射光鉗技術之自動化細胞分類/計數系統，以做為生醫檢測分析之特殊細胞純化等應用。由於使用影像辨識及結合電腦自動化控制，因此可以進行長時間及連續式之細胞分類。

首先，在第一章闡述本研究的研究背景、動機及目的，並且收集過去相關的研究主題作一文獻回顧。

在第二章闡述了影像辨識之基本原理，雷射光鉗之基本理論及其力學特性分析。此外還闡述了微管道液態流體系統之基本理論特性與原理，以及探討電滲透流電泳驅動力之現象。

而第三章則說明各單一模組之系統控制軟硬體架構，並且闡述本研究所使用之影像辨識方法，以及細胞質點粒子追蹤辨識及計數之程式設計概念。此外亦描述本研究所自行開發之影像分類辨識系統人機介面之使用及操作說明。還有本實驗所使用的晶片之設計概念、製作流程及自行製作之微流體晶片樣本，以及全自動光鉗分類系統之實驗架設。

第四章的結果與討論，分別探討了本研究所開發之全自動光鉗分類系統之性能分析，如：微流體樣本流聚焦情形之數值分析與實際實驗結果以及雷射光鉗有效捕捉力與粒子位移關係之實驗量測，並將其進行最佳化之測

試。最後，則連結整個自動分類系統，並針對整體分類功能進行自動化分類之測試。此外亦針對軟體計數器之性能進行其穩定性之效能分析。

最後一章，結論與未來展望則說明了本研究目前的貢獻以及將來尚待持續改善的計畫，並且對於更進一步的系統整合與系統商業化之穩定性提出有效的建議，期能對將來在此領域的發展有更進一步的貢獻。

第二章 全自動分類系統之基礎理論

2.1 影像擷取理論

2.1.1 影像擷取系統組成

組成一套完整的辨識系統其必須包含有影像擷取硬體、驅動程式以及自行開發或商用之套裝應用軟體。硬體提供影像擷取、時脈與控制功能，而驅動程式則是透過標準的應用程式介面（API）來存取硬體。應用軟體則專司負責數位影像顯示、辨識處理以及工作流程管控。

由於每張影像擷取後，必須經由 CCD 傳送至電腦做影像辨識。選用數位式 IEEE 1394 的傳輸介面做為影像訊號的資料傳遞，可以降低傳輸過程的干擾以及滿足快速取像的需求。

以一張灰階影像畫面而言，每一個畫素(pixel)其影像強度值可以介於 0~255 之數值來代表，亦即為一 8 bit 深度之影像。因此影像資料可以一個 $M \times N$ 的矩陣來表示，如式 (2-1) 所示。

$$f(x, y) = \begin{bmatrix} (0,0) & (0,1) & \dots & (0, N-1) \\ (1,0) & (1,1) & \dots & (1, N-1) \\ \dots & \dots & \dots & \dots \\ (M-1,0) & (M-1,1) & \dots & (M-1, N-1) \end{bmatrix} \quad (2-1)$$

本研究所使用的辨識演算法將包含形狀分析、區域分析(位置標定及區隔)、順序分析、特徵抽離以及偵測驗證。最後開發完成的應用程式將使用

這些演算出的位置及從背景分離出的形狀特徵，來進行速度、外形尺寸、形狀比對以及雷射光鉗作動時機之即時運算。

2.1.2 影像運算與影像比對法則及原理

常見的影像辨識方法可分為直接影像比對以及影像特徵值比對。且不同的影像辨識比對方法在執行效能以及運用的環境上亦不盡相同。

直接影像比對：使用常態性灰階關聯性比對法(Normalized Grayscale Correlation :NGC) ，針對影像中每個像素(pixel)進行差異比較計算，分析所擷取之影像內容以及圖形樣本(Template)間之相似程度。

影像特徵值比對：首先對於所擷取之影像內容分析其影像特徵資訊，找出可以作為圖形比對的參考。這些特徵資訊將包含有物件的大小、形狀以及其封閉外型之周圍等特徵，並可做為其與圖形樣本(Template)間相似程度之計算[26]。

在常態性灰階關聯性比對演算法的計算中，包括了許多重複性的迴圈運算式子，必須藉由高速執行速度的電腦來進行運算，才可以輸出正確的比對結果。在過去，若運用此影像演算法則於視覺檢測應用裡，是非常不切實際的。然而現今，電腦 CPU 的執行速度已突飛猛進，應用此原理所改良的新式圖形辨識軟體也陸續被開發出來。其軟體的功能性，已大大地優於傳統的二值化相關性辨識方法。

2.1.3 常規化相關匹配法(Normalized Cross Correlation, NCC)

在影像表示的二維空間中，欲進行特定標的物樣本的比對，可以將樣本視為一矩陣形式來進行原始影像之搜尋比對[27-29]。如圖 2.1 所示。假設原始影像為 $f(x, y)$ ，樣本影像為 $t(x, y)$ 。 $f(x, y)$ 為影像資訊且包含特徵樣本資訊 $t(x, y)$ 於 (u, v) 位置。因此，樣本比對的相關匹配公式(根據 squared Euclidean distance 距離量測方法)可表示為：

$$d_{f,t}^2(u, v) = \sum_{x,y} [f(x, y) - t(x-u, y-v)]^2 \quad (2-2)$$

將 d^2 展開後可得

$$d_{f,t}^2(u, v) = \sum_{x,y} f^2(x, y) - 2 \sum_{x,y} f(x, y)t(x-u, y-v) + \sum_{x,y} t^2(x-u, y-v) \quad (2-3)$$

對於影像比對之理想條件來說， $f^2(x, y)$ 以及 $t^2(x-u, y-v)$ 應為固定值。因此上式中 $2 \sum_{x,y} f(x, y)t(x-u, y-v)$ 將隨樣本矩陣的移動而變化，其值愈大，則 $d_{f,t}^2(u, v)$ 愈接近於零，亦即比對結果之相似度愈高。

因此可以定義影像比對之相似度相關係數 $c(u, v)$ 如下：

$$c(u, v) = \sum_{x,y} f(x, y)t(x-u, y-v) \quad (2-4)$$

當 c 值為最大時，即表示該處影像與樣本 t 之相似度最高。

儘管上式可以做為樣本比對相似度之計算依據，但其準確性將受到樣本影像 t 以及原始影像 f 亮度之影響而改變。因此可以藉由將原始影像與樣本影像向量常規化(Normalizing)成為單位長度，產生一類似餘弦之關聯性係數，即可消除並克服因打光以及光源照度不均勻而造成原始影像與樣本影像之取像品質不一致，因而導致比對之準確性不良。其關聯性係數公式(2-5)

如下：

$$r(u, v) = \frac{\sum_{x,y} [f(x, y) - \bar{f}_{u,v}] [t(x-u, y-v) - \bar{t}]}{\sqrt{\sum_{x,y} [f(x, y) - \bar{f}_{u,v}]^2 \sum_{x,y} [t(x-u, y-v) - \bar{t}]^2}} \quad (2-5)$$

其中 \bar{t} 為樣本特徵之平均值。 $\bar{f}_{u,v}$ 為原始影像於樣本特徵區域之平均值。上式即為常規化相關係數(normalized cross-correlation coefficient)，其可做為樣本比對之精確度指標。

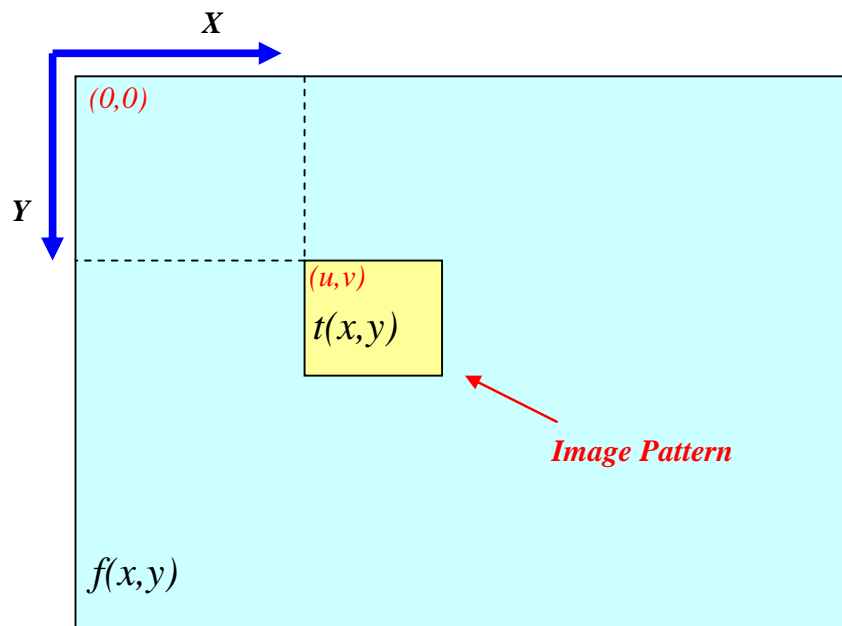


圖 2.1 樣本影像(Image Pattern)矩陣搜尋比對影像平面示意圖。

以上所述為所有影像辨識演算法之最基本理論。而現今之商業套裝軟體均以此為基本演算法則，並強化其邏輯運算能力。

2.2 光鉗之基本原理

於 1969 年，Ashkin 首先發現利用雷射光壓聚焦的原理可以移動水中直徑為數 μm 左右的塑膠球。於次年 1970 年，他將兩道相向的雷射光(Counter Propagating Laser Beams)加以聚焦，利用其相反方向的光壓構成一個穩定能量阱，鉗住直徑為 0.59 到 2.68 μm 的塑膠球[30]。1986 年，Ashkin 等人將單光束雷射聚焦後，共同發現於焦點處會產生與光前進方向相反的軸向吸力。並且粒子於雷射聚焦點附近具有橫向之牽引力，因此可將球固定在焦點處。此即所謂的『單束光梯度力阱』(Single-Beam Gradient-Force Optical

Trap)或是稱為『光鉗』(Optical Tweezers)[31]。利用此雷射光鉗的作用力，即可以將直徑大小約數十 μm 的塑膠球任意操控[32]。

由於雷射鐳夾能捕捉到的微粒子大小約在 $\text{nm} \sim \mu\text{m}$ 的等級，恰好與生物學中微物的尺度相符，因此在光鉗誕生之後，隨即被運用在生物學的研究上。

雷射光鉗具有捕捉、搬運、與操縱微小粒子的功能。一般而言，其力量大小在 10^{-12} 次方牛頓等級，因此還可以用來量測微小拉力。若配合特殊波長的雷射光源，非常適合應用在基礎分子生物的研究上，例如可以利用雷射光鉗來量測運動蛋白質(Motor Protein)的運動力，或是篩選細胞、微生物等之應用，而成為生物研究的利器。因為雷射光鉗是一種非接觸性與非侵入性的工具，目前已被廣泛的應用於細胞生物力學、分子生物學、藥物研發、細胞分群、微機械、微測量技術等，並且新的應用方式及應用範圍也仍持續被發現中[33]。

由於本研究使用之粒子大於雷射波長，因此使用幾何光學模型來解釋其基本原理[34]。若將一束雷射由上而下做高度聚焦，並在焦點附近放置一數 μm 大小的塑膠微粒。由圖 2.2 所示，雷射經由雷射光源射出後，通過一高 N.A. 值（數值孔徑）物鏡聚焦之後射往管道中粒子附近。由於粒子材料與周圍介質之折射係數不同，光子進入粒子後，會有部份的光子被介面折射或反射，即光子的方向受到改變，則光子的動量也會改變，又根據動量守

衡，粒子必需趨向平衡動量變化的方向，使粒子受到此一平衡趨勢而往雷射焦點移動，如圖 2.3 所示。此時，粒子上有兩種作用力——一是光子因折射造成動量改變而產生的光力，二是光子因散射造成動量改變而產生的光力。此一吸引力，分成兩個分量，一個分量是指向光軸的梯度力(Gradient force)，另一個分量則是指向光行進方向的散射力(Scattering force)。在兩作用力的總合使將使得粒子被鉗住於雷射光鉗 (Optical Tweezers or Laser Trap) 聚焦處。由於光束成高斯分佈，在靠近光軸處光束較強，即吸引力較大，而遠離光束處則光束較弱，吸引力亦較弱。而光鉗所產生之梯度力，即是本實驗用來吸引粒子，改變粒子流線之作用力。

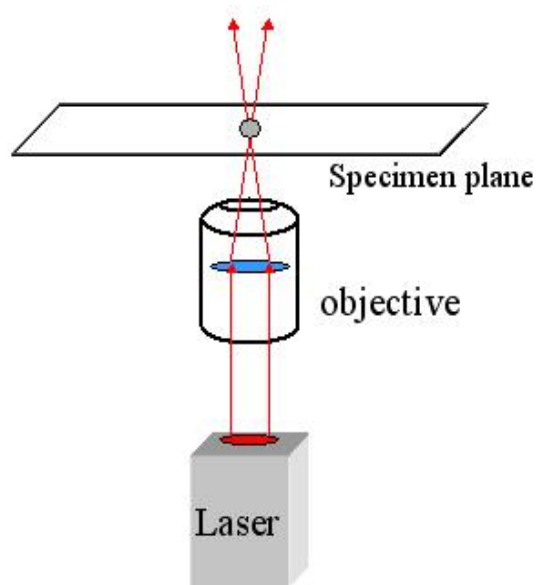


圖 2.2 雷射光鉗聚焦原理示意圖。

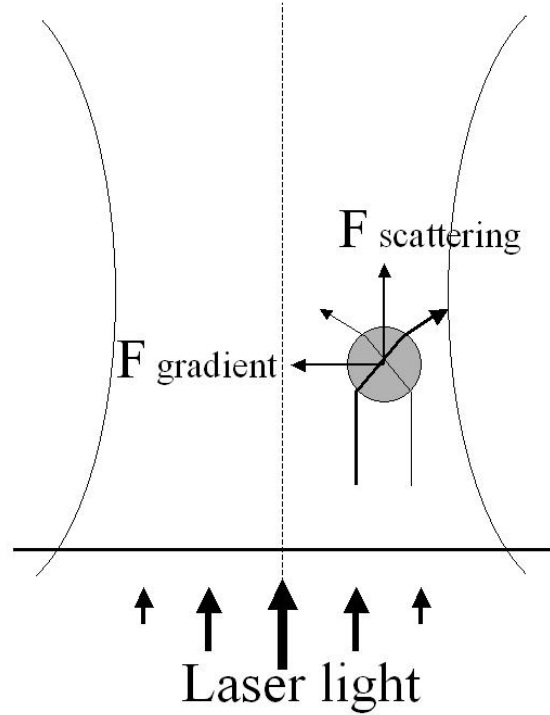


圖 2.3 光鉗作用力原理示意圖。

根據圖 2.4 之光學模型，基於動量守恆之推導，吾人可得到簡化後之雷射光鉗作用分力[35]如下所示：

$$F_{\psi} = \frac{nP}{c} \left\{ R \sin 2\theta - \frac{T^2 [\sin(2\theta - 2\theta_r) + R \sin 2\theta]}{1 + R^2 + 2R \cos 2\theta_r} \right\} \quad (2-6)$$

$$F_{\zeta} = \frac{nP}{c} \left\{ 1 + R \cos 2\theta - \frac{T^2 [\cos(2\theta - 2\theta_r) + R \cos 2\theta]}{1 + R^2 + 2R \cos 2\theta_r} \right\} \quad (2-7)$$

其中 F_{ψ} 是光束橫向牽引球體的力，與光束強度分佈有關，又稱為梯度力 ($F_{Gradient}$)。 F_{ζ} 是光束推球體的力，與光束行進的方向平行，又稱為散射力 ($F_{Scattering}$)。

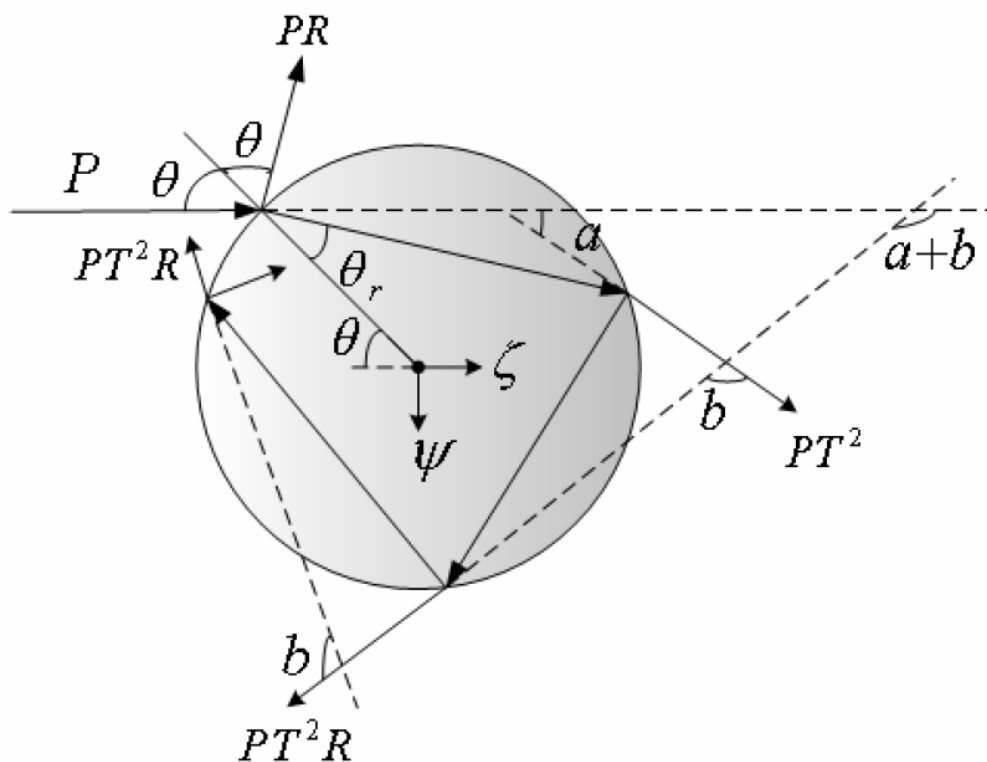


圖 2.4 雷射光束入射透明球體之幾何光學模型示意圖[35]。

其主要的基本概念是將一道雷射光束經由一高數值孔徑之物鏡聚焦後，在雷射聚焦處會形成一穩定的位能阱。 $F_{Gradient}$ 的方向是指向光強度較高處，因此對一高度聚焦雷射而言， $F_{Gradient}$ 即是指向光學焦點之光學嵌住力。當微小粒子流經此焦點周圍時，光鉗便利用此非機械接觸的光梯度力將微粒子吸引到焦點中心，進而達到三維捕捉與操控的目的。

而在本研究中，主要是探討如何利用光鉗之橫向捕捉力的特性來對微流道中的粒子進行分類的機制。因此亦對於橫向捕捉位移的效應進行實際之實驗統計量測。

2.3 微管道液態流體系統

目前微管道液體流已廣泛應用於工業上之微流體系統設計及微型散熱裝置上。在考量這些裝置的流體系統特性時，因其微管道尺寸已降至數 μm 至數百 μm 之間，流體在此管道雖可視為連續流體，但其與一般大尺寸管道的流體行為已不盡相同。管壁與液體間之界面效應將強烈地影響流體流動的行為。電雙層 EDL (Electric Double Layer) 的效應於微尺度液態流場內，其流體速度分佈、摩擦力係數、黏滯性以及熱傳導效應將不容忽視。由於大部份的固體表面均帶有靜電荷，亦即表面電位差。此電位差存在於管壁與液體間之界面效應，其主要來自於帶電介質與正負離子交互作用而形成，此電位差所形之物體力 (Body force) 將會影響流場特性，此稱為電動效應 (Electrokinetic effect) [36]。Mala [37] 和 Yang [38] 等人並以此效應來解釋流體在微管道的特性。

使用電滲透流的方式來推動流體，具備了不須額外搭配複雜的外接裝置即可精準的控制流體速度之優點，因而成了近幾年來較被廣泛使用的驅動力 [39]。本實驗亦使用此驅動力之方式做為細胞樣本操控之推力。在微尺度下，界面現象 (interfacial phenomena) 效應的影響對於流體傳輸而言必須加以考慮。以下將針對此一現象進行探討。

2.3.1 電滲透流聚焦

當矽基 (silica) 材料於水溶液中進行解離時，其管道壁面的 Si-OH 官能基在解離後，會使得壁面產生負電荷 (SiO⁻)，並使得電解液中帶正電荷的離子聚集於壁面附近。電雙層大致可分為兩部分：一為位於固定層 (Stern layer) 為管道壁面固定不動的帶正電荷離子。另一則為位於擴散層 (Diffuse layer) 之離管壁較遠且可移動的擴散離子。電雙層 (electrical double layer, EDL) 大多存在於固液界面 (solid-liquid interface) 之間。其 Zeta 電位勢[36] 與毛細管壁的電荷以及溶液的 pH 值有關。溶液的 pH 值越高表示電滲透流亦越大。

若於微流體管道內施加一電場，其流體會產生一種幫浦驅動力的現象，此即所謂的電滲透流 (Electro-osmotic Flow, EOF) 效應。其移動的狀態是由高壓電場朝向低電位方向以等速度流動。

電滲透流的流速 V_{eof} 定義如下[40]：

$$V_{eof} = \mu_{eo} E_{el} \quad (2-8)$$

μ_{eo} 是流體的電滲透泳動率，而 E_{el} 則為所施加的電場強度。由式子 (2-1) 可知當所施加的電場強度增加其電滲透流的流速亦會跟著增加。因此欲改變電滲透流的大小除了可改變所施加的電壓大小以外，尚可利用改變緩衝溶液的 pH 值、添加有機溶質或是界面活性劑等，進而改變並控制電滲透流流速的大小。如圖 2.5 所示，利用電滲透流可以降低微管道內壁面

分離紊流產生的機會。目前，應用此現象於樣品流之傳輸、聚焦 (Electrokinetic Focusing) 以及切換進料上之應用已相當廣泛[41-43]。

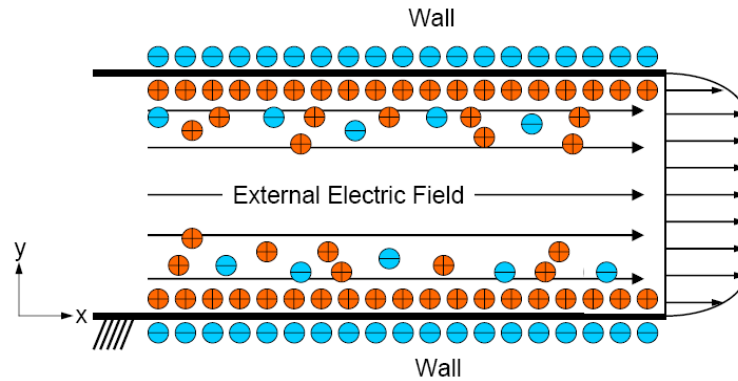


圖 2.5 電滲透流流場速度分佈示意圖[39]。

本研究利用施加直流電壓進行樣品驅動，並利用其兩側邊鞘流，將中央樣品流壓縮至單一細胞之寬度，細胞便可於微流體中依序排列流動前進。聚焦後之流線，可將細胞質點控制在一定的流動速度及移動密度下通過影像辨識區，以便於影像辨識、計數及雷射光鉗分類。其利用兩側邊鞘流，可將中央樣品流壓縮至單一細胞之寬度，細胞便可於微流體中依序排列流動前進，而不至發生同時有兩個以上之細胞通過影像辨識區，造成光鉗操控錯誤的情形發生。

第三章 全自動細胞分類控制系統整合與實驗架設

本研究提出之系統為一新型整合式控制系統，本實驗操作主要是結合雷射光鉗與微流體系統的特性來對微尺度中之細胞/粒子作操控，並利用影像辨識的功能事先對欲分類的目標設定條件，以達成光鉗系統之連續分類及計數之操作。此系統共整合了數位影像辨識追蹤功能、微流體晶片製程技術、電滲透流驅動理論、雷射光鉗應用以及視窗架構(PC-Based)電腦即時控制系統。其首先利用微機電製程製作出微流體晶片，並利用電滲透流的原理驅動細胞樣本，樣本流經過流體聚焦後，依序排列通過影像辨識區。經過自行開發之即時影像辨識演算系統進行標的細胞樣本辨認，之後透過PC-Based 電腦控制輸出同步訊號至雷射光鉗分類光學系統。近紅外光雷射經聚焦於微管道內樣本流之流經途徑上，經過位置以及速度運算控制，以便對標的細胞樣本進行連續式之分類切換。並於分類完成後利用軟體計數器統計分類數量。

為了搭配硬體周邊之控制，選擇使用 PC-Based 電腦架構來開發整個自動控制系統。因此在整個使用者圖控介面、影像顯示以及影像演算法辨識處理上較易整合。

本研究之細胞分類系統其流體聚焦、軟體計數器功能配置、雷射光鉗作用區及光學取像設計概念如圖 3.1 所示。

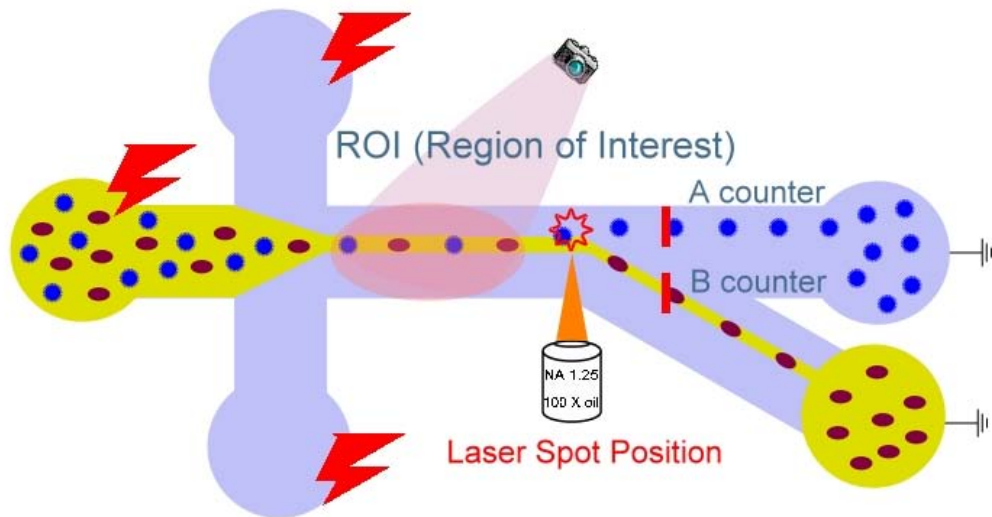


圖 3.1 微流體細胞分類系統之流體聚焦、軟體計數器功能配置、雷射光鉗作用區及光學取像設計概念示意圖。

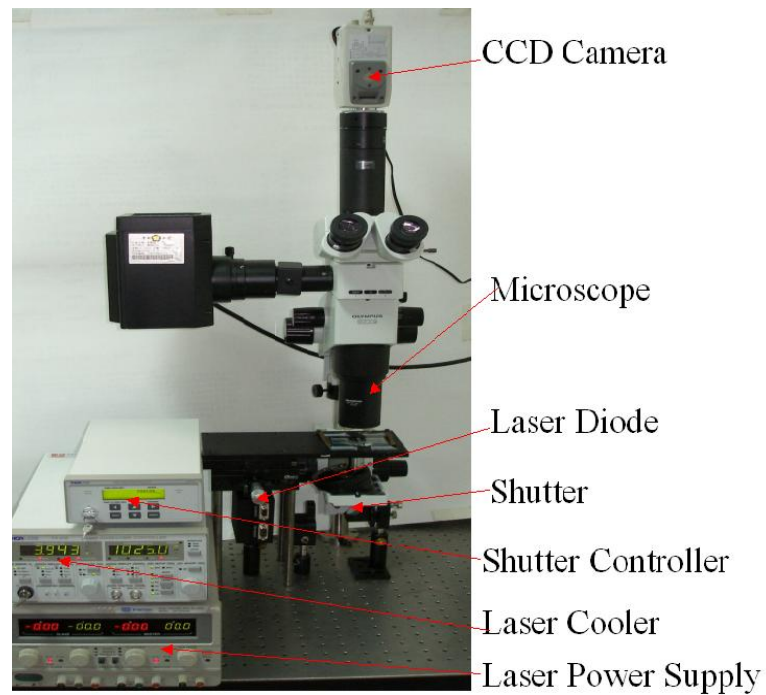


圖 3.2 光鉗系統實體圖。

整個實驗光路為自行架設之系統，實際系統全貌見圖 3.2。上半部的光路系統主要是用來直接顯微鏡觀測及影像擷取，其係由一實體顯微鏡所構成 (SZX9, Olympus, Japan)，並利用一高速影像擷取介面卡 (IEEE 1394) 以及 Basler CCD 以捕捉其微管道中之影像。下半部是雷射光鉗部份，雷射光源是使用半導體紅外光雷射(L808P1WJ, Thorlabs, USA)，最大輸出功率為 1 W，中心波長為 808 nm，配合一整合雷射冷卻與電源供應之控制器 (ITC502, Thorlabs, USA)操作，但因雷射之操作電流最大可達 2.5 A，整合控制器所能提供最大電流僅 1.5 A，因此又外接一電源供應器作為電流源。雷射通過一倍率 100X，N.A.值 1.25 之物鏡(Oil immersed, Olympus, Japan)聚焦後，輸出作動。共焦光路上方為一立體顯微鏡(SZX9, Olympus, Japan)，透過高速數位 CCD 及高速標準 IEEE 1394 介面將微管道中流動影像接收傳回電腦，並即時進行影像演算法及辨識處理。

實驗使用一高壓電源供應器(MP-3500, Major Science, Taiwan)來供應電滲透流驅動所需之電源。使用之晶片以一般鈉玻璃載玻片製成，尺寸為 25 mm×70 mm×1.0 mm，適合一般光學顯微鏡平台操作。用來實驗的粒子是 PS (polystyrene)的高分子球(Duke Scientific, USA)，尺寸分別是 5 與 10 μm 之螢光微粒子，粒子在晶片中流過觀測區的影像，以一高速數位 CCD (A311fc, Basler, Germany)擷取後傳回電腦，影像辨識系統依實際通過之粒子情形，再配合控制系統來作動快門控制器 (SC10, Thorlabs, USA)以

決定快門(SH05, Thorlabs, USA)之開閉，藉以控制雷射光鉗。

若以功能性來對整個自動控制系統加以分類，可以細分成如下四個子系統，請參考圖 3.3 所示之自動控制系統整體架構標示說明：

1. 視覺系統
2. PC-Based 電腦控制系統及自行開發之人機介面
3. 雷射光鉗與光學操控系統
4. 微流體晶片分類平台

各子系統間之相互關係關聯作用圖如圖 3.4 所示。以下將針對各子系統分別加以描述說明：

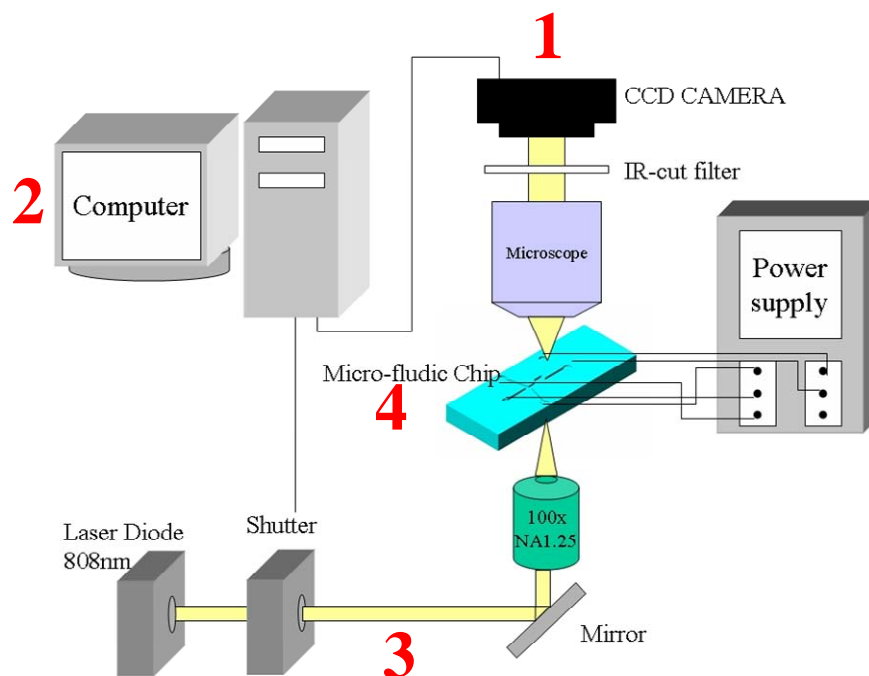


圖 3.3 自動控制系統整體架構示意圖。

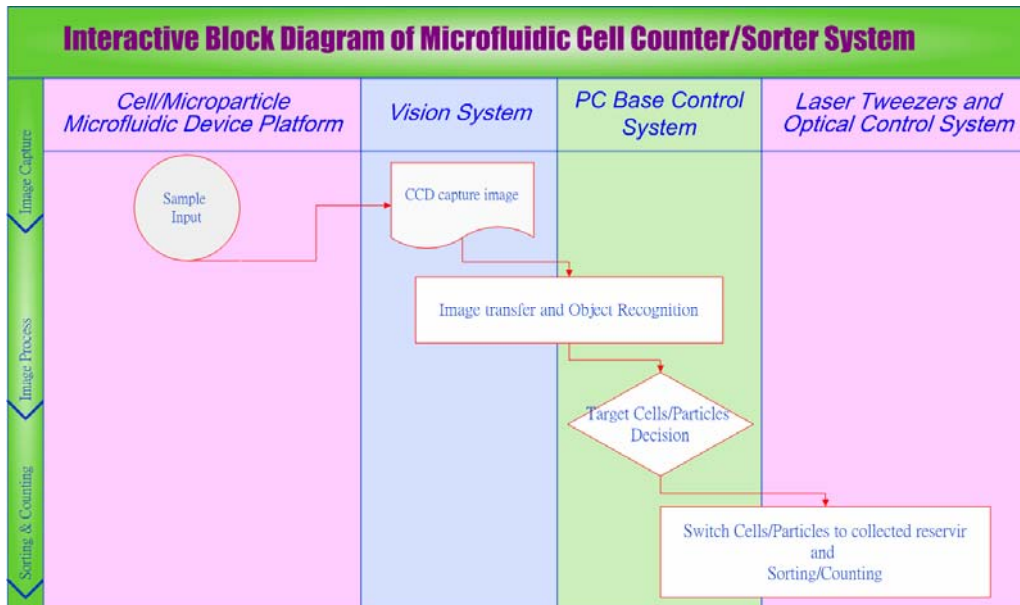


圖 3.4 全自動光鉗操控之細胞分類系統各子系統之相互關係關聯作用圖。

3.1 視覺系統軟硬體架構

3.1.1 視覺系統硬體架構

視覺系統硬體架構包含有光源、光路設計、CCD camera、影像擷取使用高速標準 IEEE 1394 介面。由於每張影像擷取後，必須經由 CCD 傳送至電腦做影像辨識，選用 IEEE 1394 介面的另一個好處是可以簡化系統的建置。只要電腦具備有標準的 IEEE 1394 介面，即可進行系統的連結，而不需要額外購置影像擷取卡。本實驗架構使用筆記型電腦做為 PC-Based 電腦控制系統及影像辨識核心，且具備 IEEE 1394 的通訊介面，因而大大簡化了系統的建置架構。CCD 使用德國 Basler 公司所出產型號為 A311fc 的數位式鏡頭，做為影像擷取裝置來捕捉微管道中之影像。其解析度為 656×492 pixel，frame rate 為 75 frame/sec。

3.1.2 視覺系統影像辨識演算法

為了加快開發的流程，本研究以市面上之商用影像辨識系統函式庫為演算法之基礎，再根據實驗系統的實際需求，來進行整個辨識系統之客製化開發。辨識基礎演算法為比利時 EURESYS 公司所出版的 eVision 套裝軟體搭配 Microsoft 公司所出版的 Visual Basic 6.0 以及 Visual Studio C++ 2003 來進行整個辨識系統及人機操控介面的開發，以設計符合辨識條件要求的程式規劃。

透過 CCD 擷取之影像經過 1394 界面傳輸至主電腦，立即被擷取至自行開發之記憶體共用資源區，並透過所撰寫之 Visual Basic 6.0 影像處理程式，將影像顯示之指標指向記憶體區塊，之後再進行影像處理。

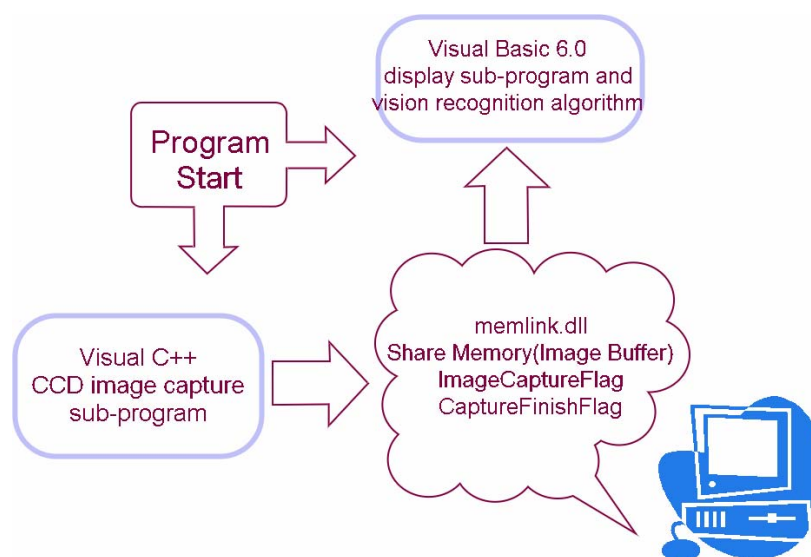


圖 3.5 利用動態連結函式庫 DLL(dynamic link library)的方式於記憶體開啟一共用資源。

由於 Basler CCD 的取像程式為 C 語言所編寫，無法與 Visual Basic 6.0 的辨識程式介面直接溝通，因此必須使用 C 語言自行撰寫一個取像程式，來擷取影像。並利用動態連結函式庫 DLL(dynamic link library)的方式於記憶體開啟一共用資源。利用此共用記憶體來做為 Visual C++與 Visual Basic 6.0 的溝通介面。所擷取的影像資料必須立即寫入此記憶體區塊，並利用軟體交握(Handshake)之掃描方式，告知 Visual Basic 6.0 辨識程式來抓取此影像資料 並進行影像辨識。當辨識完成後再利用軟體交握(Handshake)的方式告知取像程式進行下一張影片的擷取。其溝通示意圖如圖 3.5 所示。

Multiple Particles Vision Identification and Laser Tweezers Sorting System

Sequence Control Flow Chart

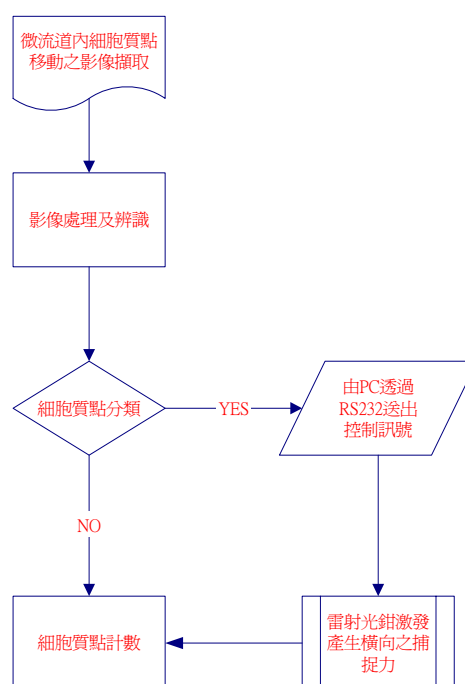


圖 3.6 本研究影像辨識及分類計數系統控制流程圖。

微流體粒子分類器系統取像辨識及分類計數流程圖，如圖 3.6 所示。當設定特定粒子篩選條件後，程式會依即時影像進行數位處理。若分辨為特定目標粒子，則控制並作動雷射光鉗，分離該粒子並通過 B counter 計數器。反之，則通過 A counter 計數器。

本研究所使用的辨識演算法將包含形狀分析、區域分析(位置標定及區隔)、順序分析、特徵抽離、偵測及驗證。最後開發完成的應用程式將使用這些演算出的位置及從背景分離出的形狀特徵，來進行速度、外形尺寸、形狀比對以及雷射光鉗作動時機之即時運算。經過抽離出來的質點物件資料，影像系統將賦與座標位置、相似度、面積大小、與原始樣本之夾角等特定資訊。利用這些有用資訊，PC-Based 電腦控制系統將可決定適當之控制時機與控制時間長短，以便對雷射光鉗系統進行自動控制，並針對符合條件之目標細胞質點進行攫取，因而改變其行進之方向及流線，進而達到分類之目的。

原始影像經取樣後，可轉換為灰階影像及二值化影像，以便於進階之影像處理。參數設定完成後，系統會對藍色辨識框格中的粒子進行比對辨識，並將粒子從背景分離出來以便進行位置之計算。依據其移動之相對座標，可立即計算出其於微流道內之行進速度，並即時將位置及速度資訊顯示於該粒子周圍。此相關資訊將提供後端光鉗反應時間之計算，並於粒子離開辨識框格後作動光鉗，以便正確捕捉欲進行分類篩選之粒子。光鉗作動時

間已被最佳化，其開啟與關閉由一與電腦控制系統連結之作動快門來控制。電腦控制系統與快門之控制是經由 RS 232 作為控制命令之通訊界面。粒子之流動速度會於計算程式內換算成速度因子，且被限定在 0.5~5 之間(可程式化)。作動開啟時間將等於基準控制時間(400 ms) 乘以(速度因子)。速度因子將隨粒子之移動速度而變化，並經由控制程式計算後輸出至快門以控制快門開啟時間。當粒子之流動速度愈快速度因子將愈小。例如:當速度因子等於 0.5 時，控制快門之作動時間將等於 $400\text{ ms} \times 0.5 = 200\text{ ms}$ 。

本研究所開發出之數位影像細胞辨識追蹤系統將分成「多重質點細胞影像辨識」及「細胞質點移動速度及位置追蹤偵測」等兩大類功能，以下將詳述其辨識方法與運算法則：

3.1.3 多重質點細胞影像辨識

在本研究中所自行開發出之影像辨識方法主要運用了兩種影像演算法：(a)樣式比對”(Pattern Match)”以及(b)簡易物件法“EasyObject”。此二種方法均可以於複雜的背景環境中，利用影像辨識比對的方法，只要經過適當的影像閾值二值化設定，即可將特定的質點目標物件抽離出來。

(a). “樣式比對(Pattern Match)”：當使用此方法之前，使用者必須於樣本影像內先決定欲分類目標質點之式樣，並加以學習(learn pattern)。影像系統將以此為學習範本，並於其後所攫取之每一幀影像畫面內搜尋相同或近似之質點。針對符合範本式樣之影像質點，影像辨識系統除了回傳基本

的質點所在之座標資訊外，並將提供”比對分數(Score)”及”大小比例(Scale)”之資訊。 ”比對分數(Score)”即代表相似度；”大小比例(Scale)” 即代表同一類型之細胞質點大小。於適當設定欲比對之”比對分數(Score)”及”大小比例(Scale)”等參數，“樣式比對(Pattern Match)”將可針對不同形狀及不同大小之細胞質點進行區別辨識。如圖 3.7 所示，影像辨識系統以 10 μm 為辨識範本，成功地於微管道內針對 10 μm 樣本進行目標質點辨識。

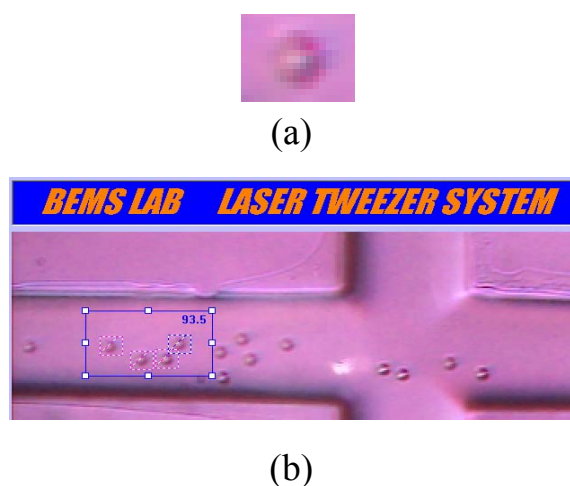


圖 3.7 “樣式比對(Pattern Match)”作動圖(a) 欲進行搜尋比對之影像樣本(b) 多質點影像辨識比對搜尋結果。

對於細胞質點種類的辨識可以區分成二個要素：1、細胞質點的大小。2、細胞質點的形狀。本實驗將使用影像處理軟體進行 Pattern Match 方法來比對細胞質點，其利用先行對粒子影像進行學習，並於操作時即時利用該學習之影像與樣本細胞進行相似度之比對，並即時判斷該粒子與學習粒子間之相似度之比例來決定細胞質點之大小，以及依據其與樣本之相似度百分

比做為形狀分類之判斷準則。此一作法不僅可以應用於不同形狀樣本粒子之辨識，更可以大幅改善傳統細胞計數器所使用之影像辨識分析法所造成之辨識錯誤等問題，因此可以提高標的細胞之辨識率。

(b) “簡易物件法(EasyObject)”：簡易物件法將直接利用影像辨識函式庫的功能，將特定的質點目標物件自複雜的背景環境中抽離出來。針對所分離出之細胞影像質點，影像辨識系統除了回傳基本的質點所在之座標資訊外，另外諸如細胞質點所佔之像素面積大小、幾何外型特徵、幾何中心等特徵資訊均可被回傳並加以運算。此面積大小相當有用，因為經過實際尺度換算後，它即可代表細胞質點之大小。本研究運用此種比對方法，只要設定其欲篩選目標細胞質點之大小範圍，即可進行二種以上不同大小混合樣本之自動辨識分類，且其辨識之準確度相當的高以及設定簡便，因而大大降低了操作系統使用之複雜性。如圖 3.8(b) 所示，針對 10 μm 與 5 μm 之混合樣本，影像辨識系統所辨識出之 10 μm 質點其所佔之像素為 42 pixels，5 μm 所辨識出之質點其所佔像素為 13 與 18 pixels。10 μm 與 5 μm 所佔之面積區間有很大的差異，假設以 35~45 pixels 為所欲分類之目標質點細胞範圍，則系統將可連續自動地針對微管道內 10 μm 進行目標質點攫取及分類。

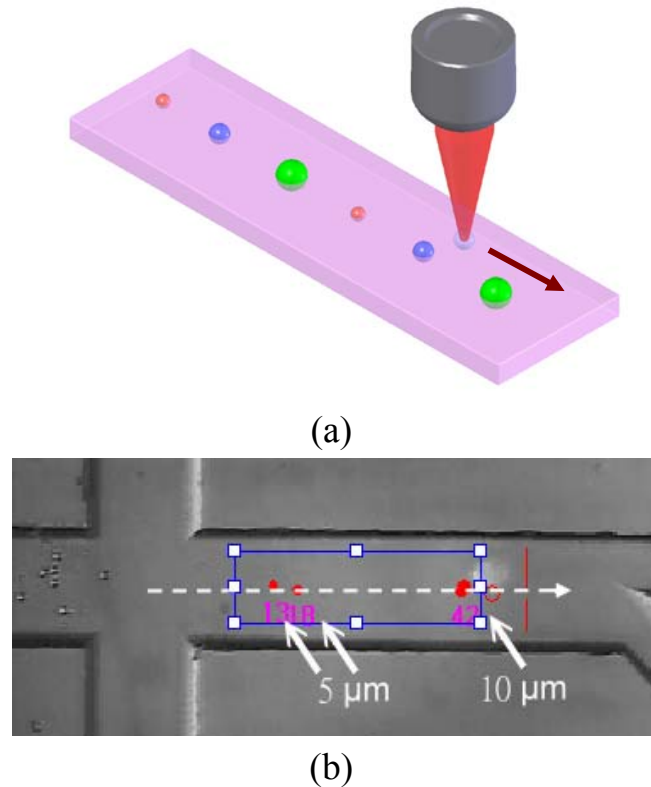


圖 3.8 “簡易物件法(EasyObject)”作動圖(a)針對混合樣本進行目標質點粒子分類示意圖(b)多質點影像辨識比對搜尋結果。

3.1.4 細胞質點移動位置及速度之計算及應用

本系統欲藉由影像辨識所計算出的座標值及質點之移動速度數據，經過數值運算轉換，可立即做為雷射光鉗控制子系統之控制參數。當辨識後之細胞質點被判斷為是標的粒子後，則雷射光鉗將適時地利用系統送出控制訊號激發。其激發之時機、激發模式(Continuous or Pulse Mode)及作用時間長短乃是依據細胞質點之移動座標值及速度而決定，主要之目的都是為了適當地控制雷射光鉗捕捉力，達到有效地細胞質點分類。因此細胞質點移動位置之追蹤於本系統扮演重要之角色，因唯有精確掌握細胞流動之軌

跡，方可以正確的送出雷射光鉗之訊號，進而正確收取標的細胞。

當影像中特定的質點目標物件自複雜的背景環境中抽離出來後，繼而就可以辨識出質點之絕對及相對座標位置，同時細胞質點所佔之像素面積大小、幾何外型特徵、幾何中心等特徵資訊均可被回傳並加以運算。為了控制雷射光鉗對”即將”通過的細胞質點於正確的適當時機攫取以及決定其光鉗開啟之時間長短，控制系統必須以”即將”通過的細胞質點之位置座標及其移動速度作為計算依據。但對於每一張影像內所分離出之質點座標，其所代表的只是”靜態”的位置資訊，若欲表現出細胞質點移動的速度，則必須使用兩張連續攫取的影像來進行位置比對。因本研究所開發出的自動系統強調影像辨識及操控的即時性，為了節省電腦系統運算的時間，目前只針對”即將”通過雷射光鉗攫取區之細胞質點之行進速度執行運算。

計算質點的水平 (u) 及垂直速度 (v) 之方法如下：

對於每一瞬間 t，所有自複雜的背景環境中辨識抽離出來的特定質點目標物件集合為：

$$\text{其座標資訊包含於} \in \{P_i(x, y)_i; i = 1, 2, \dots, n\} \quad (3-1)$$

其中“n”代表經由影像辨識系統於影像辨識區(ROI)內，所即時辨識出之所有特定質點目標物件總數。

因此最接近雷射光鉗之攫取聚焦點且”即將”通過雷射光鉗攫取區之細胞質點其行進速度可表示為：

$$\vec{V}_{(t)}(\vec{u}, \vec{v})_i = \left(\frac{d\vec{P}_i}{dt}\right)_x + \left(\frac{d\vec{P}_i}{dt}\right)_y; \quad i=n \quad (3-2)$$

其中

$\vec{V}_{(t)}$ 為每一瞬間 t ，細胞質點於 X 軸方向及 Y 軸方向之移動速度。

\vec{u} 為每一瞬間 t ，細胞質點於 X 軸方向之移動速度分量。

\vec{v} 為每一瞬間 t ，細胞質點於 Y 軸方向之移動速度分量。

因為在此系統架設之平台上，可隨時調整 CCD 攝影機於微管道內所擷取影像之水平角度，且細胞質於微管道內經過微型聚焦裝置後幾乎呈直線之一顆接一顆之流動狀態，可視為單一之 X 軸流動方向。因此，方程式(3-2)可簡化並表示為：

$$\vec{V}_{(t+\Delta t)}(\vec{u}, \vec{v})_i = \vec{u}_i = \left(\frac{d\vec{P}_i}{dt}\right)_x \quad (3-3)$$

對於相隣兩張連續影像取像之時間差，可以於各單張影像取像時，由程式讀取電腦系統之時間，再將兩張影像之取像時間相減，即可得到其時程差。於 Microsoft Windows 之 PC-Based 電腦控制平台架構下，其時間精度的單位可到達 1 ms。對於同一細胞質點之位移再除以其位移之時程差，即可得到段時程間細胞質點移動之平均速率。由於每兩張連續影像取像的時間差相當地短，約莫在 40 ms 左右，因此幾乎可以視為是該細胞質點之瞬時速度。因此，該速度可以如下之公式表示：

對於相隣兩張連續影像取像之時間差 Δt ，最接近射光鉗攫取焦點且”即將”通過雷射光鉗攫取區之細胞質點，其移動之瞬時速度(u)可由如下簡

化之計算式求出。其實際辨識計算結果如圖 3.9 所示。

$$u_i = \frac{P_{(t+\Delta t)}(x)_i - P_t(x)_i}{(t + \Delta t) - t} \quad (3-4)$$

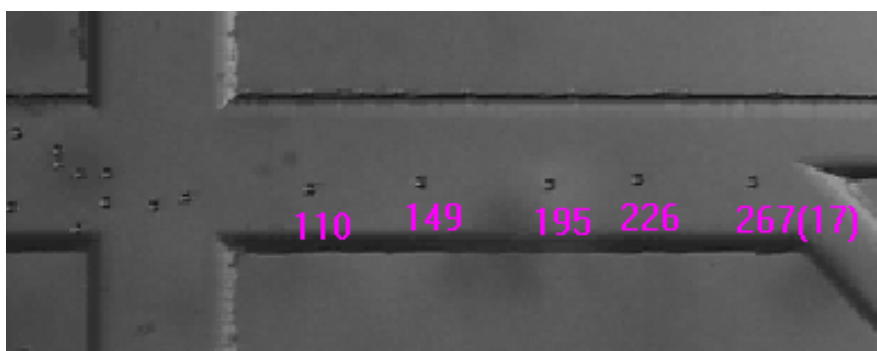


圖 3.9 即時細胞粒子質點位置追蹤及移動速度計算結果(單位為 pixel)。

3.1.5 細胞質點即時粒子計數之概念及其方法

本研究於微管道下游之細胞質點分類管道前，共設計了兩個軟體計數器，做為進入分類管道內細胞質點粒子數量之統計，如圖 3.10 所示。在程式設計內部使用了一個軟體迴圈不間斷地於計數偵測區對影像之灰階值變化進行掃描比對。其原理類似光纖偵測感應放大器之方法。當掃描到之影像灰階值變化大於預先所設定之閾值，則軟體計數器之偵測旗標即被設定為 ON。反之則被設定為 OFF。一次的 ON-OFF 循環即表示一個細胞質點粒子”正在”通過該軟體計數器。亦即當細胞質點正位於該偵測線上時，將造成該處之影像灰階值之變化，且其值將大於所設定之閾值。當該細胞質點通過該偵測線後，該處之影像灰階值將回復至閾值以下。此時軟體計數

器將自動累計加一，並進行軟體之即時粒子數量累計計算。圖 3.11 表示其偵測之方式及其連續對細胞質點粒子偵測之工作原理。

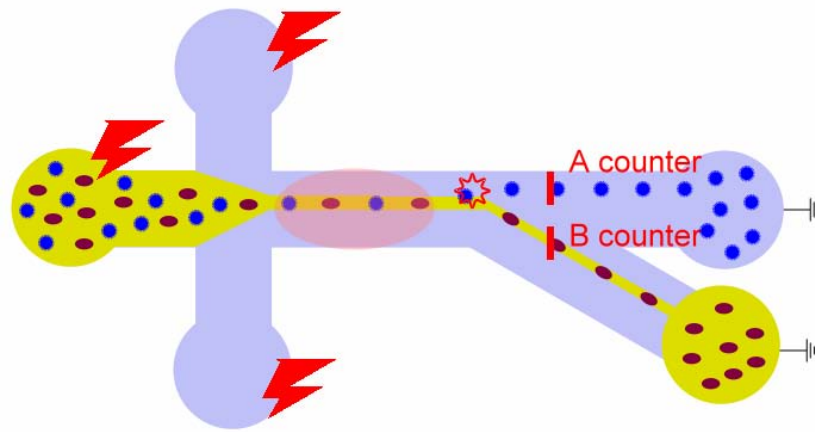


圖 3.10 粒子計數偵測位置之設計概念示意圖。

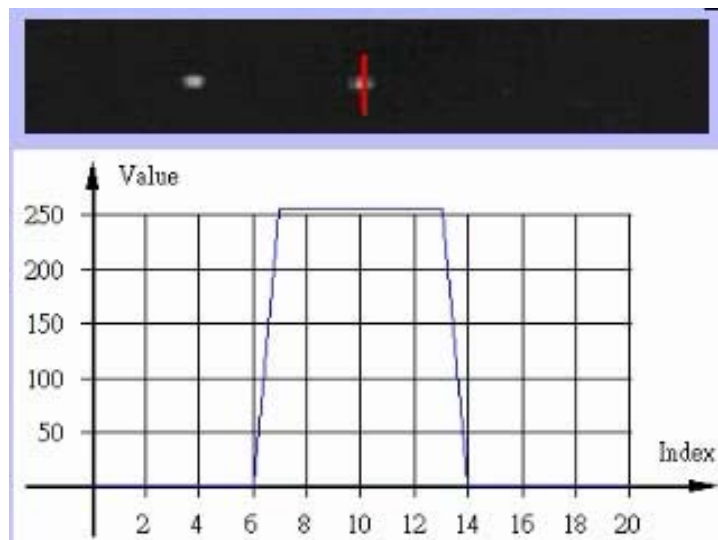


圖 3.11 粒子計數偵測迴路設計概念示意圖。

3.2 PC-Based 電腦控制及自動分類系統人機介面操作

3.2.1 PC-Based 電腦控制系統

以 PC-Based 電腦架構來開發一套新的控制系統，其具有如下之優點。

1)通俗性(Popularity)：因 PC-Based 電腦的廣泛使用已成研製主流，軟體、硬體資源取得容易。

2)相容性(Extensibility)：因系統持續升級，具有較佳的相容性。不會被單一控制器供應商限制硬體功能升級。

3)多樣化之人機介面功能(graphic Human-Machine Interface H.M.I.)：易於故障排除、易於維護以及易於使用者操作。可縮短研發設計時間及成本，並專注於重點研發設計。

4) 多工系統(Multi-Tasking) 及分時系統(Time-Sharing System)：同一時間於同一個 CPU 控制系統內可以同時執行兩個以上之程式。且可同時進行取像及週邊系統控制。

本實驗之系統使用 Intel P4 2.0G 之 CPU 做為影像處理之運算核心，不僅可使影像處理的速度大為增加，亦能增加整體控制之系統穩定性。圖 3.12 為 PC-Based 電腦控制系統與周邊介面之控制區塊流程圖。

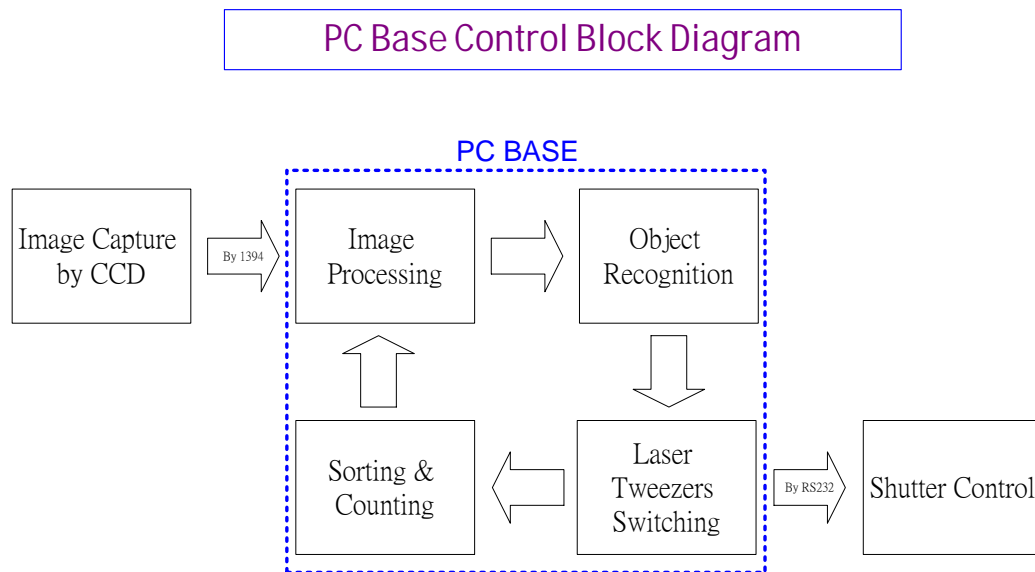
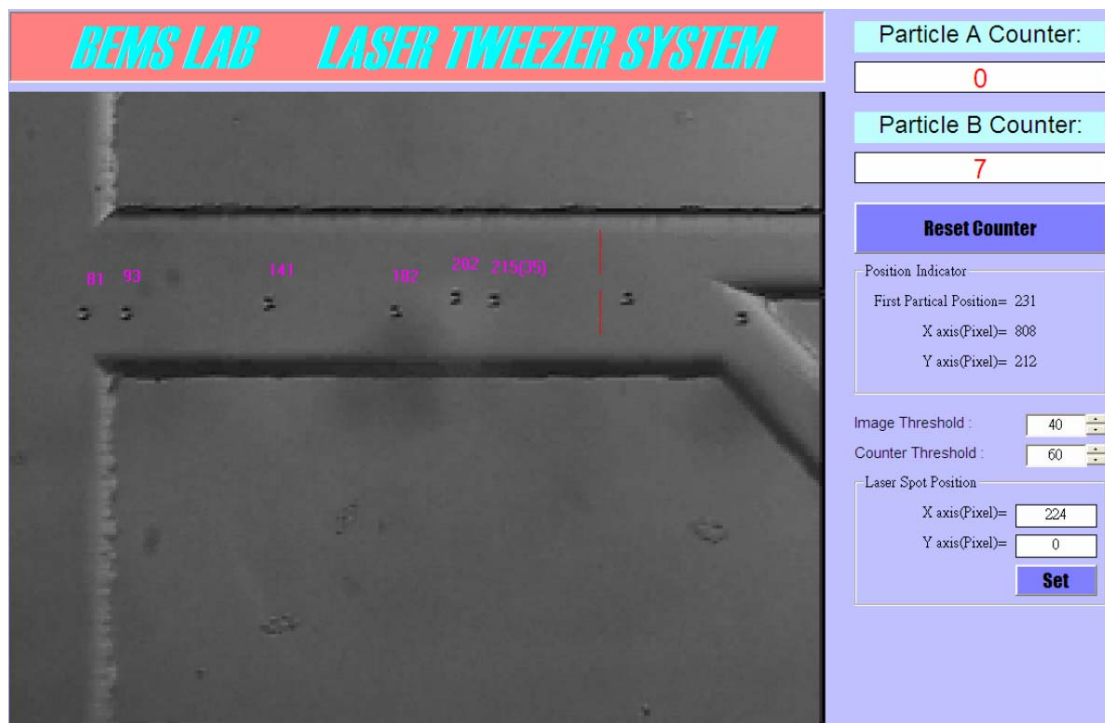


圖 3.12 PC-Based 電腦控制系統與周邊介面之控制區塊流程圖。

3.2.2 自動分類系統人機介面操作說明

在架設好雷射光鉗光路及微流體晶片後，即可準備進行欲篩選目標粒子條件之設定。首先於圖 3.13(a)之自行開發即時影像辨識人機主畫面上，先設定適當之影像篩選與軟體計數器之閾值。其次設定該雷射光鉗作用之聚焦點座標以作為觸發雷射訊號時機之參考。於圖 3.13(b) “Counter Line Position”處並設定軟體偵測器適合置放之位置，接著於”Display Option”可設定當細胞質點被辨識出後，於主畫面實際影像上欲伴隨質點所顯示之訊息種類。當為樣式比對”Pattern Match”或是簡易物件法”EasyObject”影像比對模式下，在此可選擇顯示速度及座標位置模式或是只顯示面積尺寸模式。若欲顯示樣式比對結果只可在樣式比對”Pattern Match”模式下選擇。於簡易物件法”EasyObject”影像比對模式下，還可另外設定欲篩選目標粒子條

件之大小面積範圍，如圖 3.13(b)所示。當經由影像辨識軟體所辨認出之質點粒子大小面積符合設定條件時，該粒子即被認定為欲篩選之目標粒子。



(a)



(b)

圖 3.13 全自動光鉗操控之細胞分類系統(a) 即時影像辨識人機介面主畫面及(b)即時影像辨識設定畫面。

3.3 雷射光鉗光學系統

本雷射光鉗系統為自行開發、組裝而成。此光學系統可分為上、下兩部分，下半部主要是輸入操控粒子之雷射光路徑，該雷射光源為 CW (Continuous Wave) 輸出之半導體雷射，最大輸出功率為 1 W。此實驗共使用二種波長(中心波長為 808 nm 及 975 nm)之 near-IR 雷射進行實驗測試。選用 near-IR 波長雷射之主要目的，是為了利用其部份可見光的波長特性，以便於觀察及光路的調整。由於雷射光路在安裝時，必須先行調節輸出功率及測試對粒子抓取之聚焦平面範圍。若使用 975 nm 波長之雷射光源，於進行光路調節與觀察抓取點時，因其幾乎沒有可見光之波長，因此並不易進行觀測及調節。所有最後選擇 808 nm 波長之雷射光源進行本研究之所有實驗。

雷射光源安裝在一個經過控溫之致冷晶片(TEC)溫度調節模組，因此可以在控溫的條件下長時間執行大功率的雷射輸出。

雷射光源經過控溫之 TEC 溫度調節模組發射後，必需先經過一倍率為 10 倍、焦距為 15.36 mm 之平凸透鏡聚焦，規格如表 3.1 所示。再經過一經由 RS232 透過電腦控制之機械快門來控制雷射光鉗之粒子抓取作用時機與作用時程。當欲分類之細胞/質點目標即將通過雷射聚焦作用區時，影像辨識系統會在適當時機輸出一作動訊號，並驅動快門開啟，使得雷射光通

過一反射鏡，經過反射之雷射光會垂直入射至高倍($\times 100$)/高 N.A.值(1.25)之聚焦物鏡，並聚焦在微流體晶片上欲捕捉細胞/質點目標之作用點處，進而產生高梯度之橫向捕捉力。

表 3.1 聚焦平凸透鏡 C260TM 光學特性規格表。

MOUNTED ITEM#	f (mm)	N.A.	CLEAR APERTURE (mm)	WORKING DISTANCE (mm)	OUTER DIAMETER (mm)	CENTER THICKNESS (mm)	EQUIVALENT MICROSCOPE OBJECTIVE MAGNIFICATION
C260TM	15.36	0.16	5.0	14.00	6.5	2.2	10X

3.4 微流體晶片系統

完成本研究之微管道晶片之設計與製程流程圖如圖 3.14 所示，大致可分為下列幾個步驟：

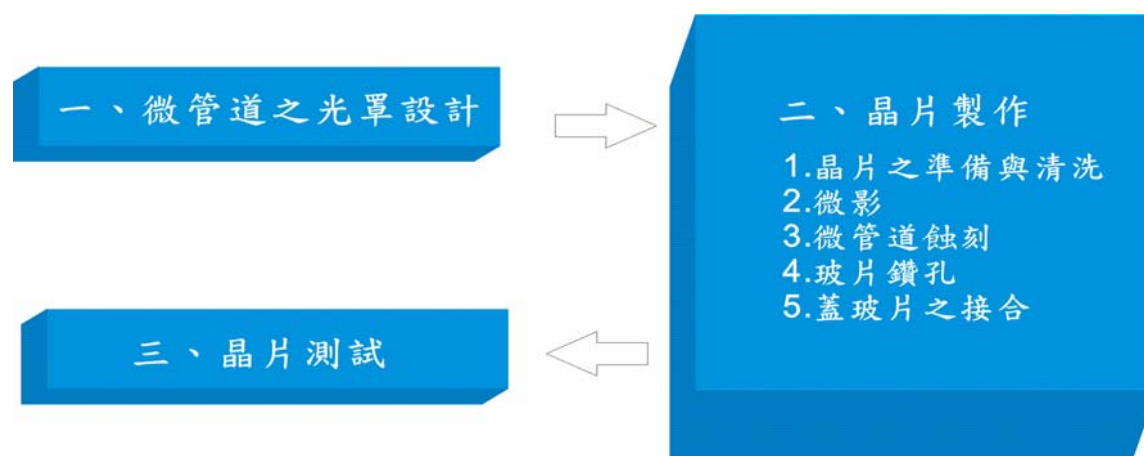
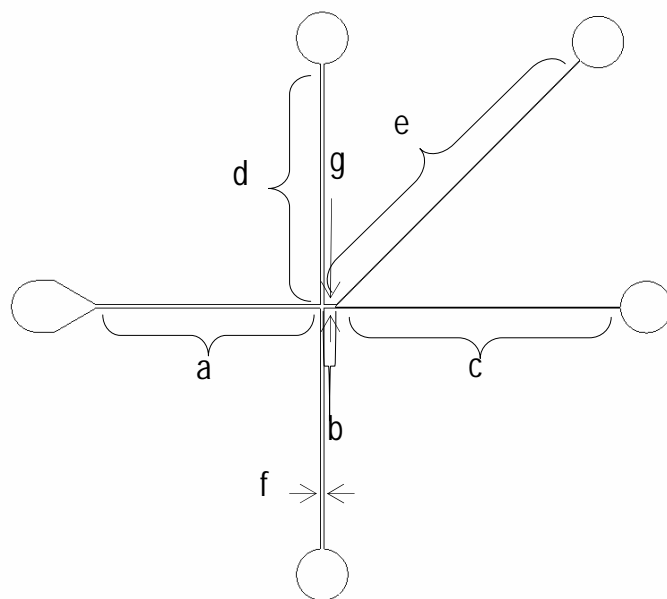


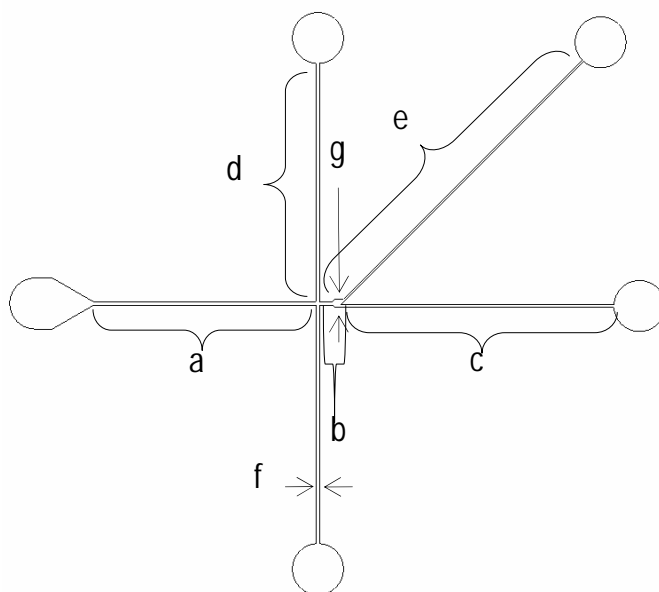
圖 3.14 微管道晶片之設計與製程流程圖示意圖。

3.4.1 微流體晶片設計



a	b	c	D	e	f	g
6.56 mm	370 μm	8.3 mm	7 mm	10 mm	160 μm	160 μm

(a)



a	b	c	D	e	f	g
6.56 mm	630 μm	8 mm	7 mm	10 mm	160 μm	300 μm

(b)

圖 3.15 (a)傳統型細胞分類器及(b)加入新式改良擴張閥設計之細胞分類器之尺寸大小示意圖。

本研究所使用的晶片，主要是利用電滲透流為液體的驅動力，以驅使液體帶動細胞流動，並且在微管道前端設計一十字形聚焦機構，利用邊鞘流聚焦效應，將樣品流之寬度縮小至十幾微米之範圍內，以限制其寬度僅允許單一細胞通過，因此細胞可於樣本流中依序流動，並通過影像辨識區以判斷是否讓光鉗作動來進行分類。當細胞受到光鉗之橫向捕捉力作動時，其會被吸引至另一流道，並透過微尺度中層流的現象，穩定地被傳輸至後方以進行分類儲存。

本實驗共設計兩種微流體晶片：一為傳統之晶片設計，如圖 3.15(a)所示。一為加入擴張閥增加粒子分離距離之新改良型設計，如圖 3.15(b)所示。設計概念上，在三個進口端施加電壓，用以產生電滲透流來驅動樣本，通過寬 $160\ \mu\text{m}$ ，深 $30\ \mu\text{m}$ 之微管道，利用十字聚焦管道設計，將粒子流聚成一寬度約 $15\sim 20\ \mu\text{m}$ 之聚焦流，此一寬度將可限制使同一時間僅單一粒子通過作動區，再經光鉗之側向抓取力，將粒子分入出口端之收集槽。

首先以 AutoCAD 軟體，在 $3.5\times 2.5\ \text{cm}$ 的範圍內設計微管道。本研究之晶片製程使用的是等向蝕刻，深寬度約 $30\ \mu\text{m}$ ，繪製時應考慮預留之寬度。同時因為雷射印表機之解析度，設計線寬應大於 $1\ \mu\text{m}$ 為限。

樣本槽與緩衝液槽為半徑 $750\ \mu\text{m}$ 之圓，十字管道設計部份，連接樣本槽的管道寬度約為 $160\ \mu\text{m}$ ，緩衝液之管道寬度亦為 $160\ \mu\text{m}$ 。如圖 3.16 中 a 區所示，目的在於利用兩旁緩衝液之較快流速將樣本液中之細胞流控

制在流道中央，且不產生兩個粒子同時通過的情形。圖 3.16 中 b 區的管道為一樣本流聚焦裝置的設計，以期樣本粒子能夠更有效被聚集排列，並依序進入影像辨識區。其後加入一擴張閥的結構，約擴張至 $300\ \mu\text{m}$ ，此一結構設計，可以造成液流不對稱的流入各分支中，並且使原先管道中的流線擴散，而使管道中兩分開粒子之間之距離被放大[44]，如此光鉗在作動時較不會發生如兩粒子同時通過之錯誤的現象。圖 3.16 中 c 區為管道末端之分流處，設計上，分流處的頂點，將會比粒子的流線高出約 $10\ \mu\text{m}$ 。如此，被聚焦之粒子便會往下管道流，辨識到欲分類之粒子時，光鉗再作動，粒子便被分入另一流道進入其分類之樣本槽。

新的光罩設計圖完成後，再用解析度 10000 dpi 之雷射印表機印在塑膠光罩上，準備用於製作玻璃晶片。整個晶片設計之示意圖如下圖 3.16。

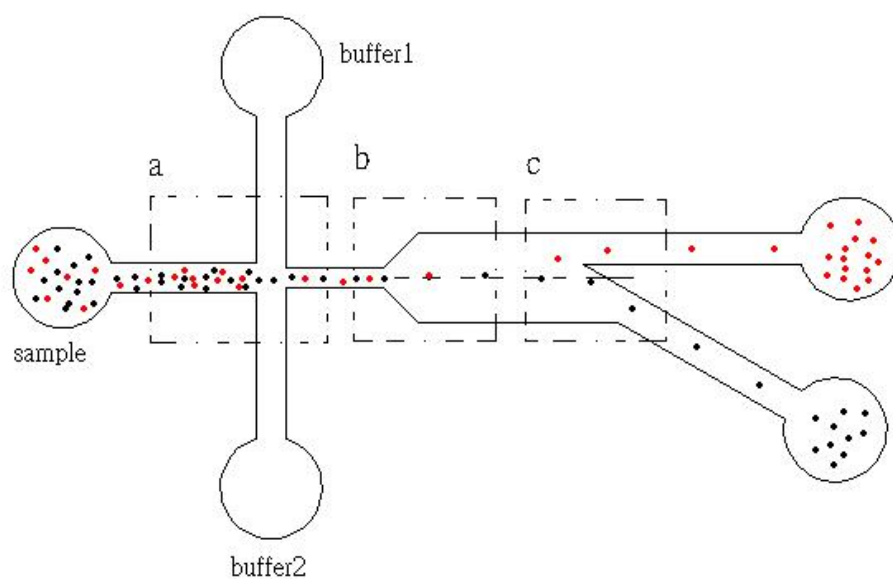


圖 3.16 管道晶片設計示意圖。

3.4.2 微流體晶片製程

由於此晶片是透過電動力 (Electrokinetic Force) 驅動，其晶片表面需要富有 -OH 官能基以產生電滲透流，因此本研究採用低成本之顯微鏡載玻片作為基材。所以晶片上，使用德國 Marienfeld 公司所生產之 25×75×1 mm 載玻片作為基材。

其製程之基本步驟是採用標準之光刻程序。本研究捨棄傳統昂貴之金屬蝕刻罩幕，而使用一厚度約為 3 μm 之 AZ4620 正光阻薄膜，作為玻璃晶片於 BOE 蝕刻槽中之蝕刻罩幕。為配合大 N.A. 值之高倍率物鏡之工作距離，並提高光學傳導率。本研究捨棄以載玻片為上蓋板之製程，而改採一厚度僅有 170 μm 之蓋玻片進行對位與接合，其整個晶片的製作過程可於十個小時內完成。

晶片製程詳細步驟可參考文獻[45]，此一製程最大特色即是成本較低且製程快速。微流體晶片製程的步驟示意圖如圖 3.17 所示，其過程詳述如下：

- 1、晶片之準備與清洗：玻璃晶片一開始應先經過 400°C 下退火四小時的處理來消去生產時的殘留應力，再放入約 200°C 之 Piranha 溶液中煮 10 分鐘，接著以 DI water 清洗之後，再以乾燥空氣吹乾，並放在 100°C 之電熱板烘烤 3 分鐘以完全去除水分子。

- 2、微影：微影包括了光阻塗佈、曝光、顯影等步驟來將光罩圖形轉到

光阻上。先以 hexamethyldisilazane (HMDS) 溶液滴上玻璃再放到 100°C 之電熱板烘烤 3 分鐘後，再把 AZ4620 正光阻以旋轉塗佈機塗在玻璃上，並再以 100°C 電熱板進行軟烤 3 分鐘。完成後開始進行曝光，約需使光阻吸收 180 mJ/cm² 的曝光能量。之後再將玻片浸入顯影液(AZ400K: DI water=1 : 3) 中約 70 sec 進行顯影，完成後再用乾燥的氮氣吹乾玻片，最後進行硬烤，先將電熱板升至 100°C，再將玻片放上並將電熱板設定加熱至 150°C，並在溫度達到 150°C 後十分鐘完成硬烤。

3、微管道蝕刻：將玻片放入蝕刻液中，此時一邊以超音波加快蝕刻效率，且每隔 5 分鐘便把玻片取出，以去離子水(DI water)清洗後，再放入 1 M 的 HCL 攪拌 10 秒以去除表面殘留物，再放回蝕刻液，如此反覆動作，經過 45 分鐘的蝕刻可得 40 μm 深之微管道。蝕刻後還要放入 80°C 之 KOH 稀釋液來去除光阻。

4、玻片鑽孔：玻片接合之前，要先以直徑 1.5 mm 之鑽石鑽頭鑽好微管道樣本液之進出口，過程中需要配合水流來降溫及帶走碎屑。

5、蓋玻片之接合：將蓋玻片和鑽孔完成的玻片做對位，並在 580°C 的高溫爐中熔融接合，持續約十分鐘後即可完成接合之程序。

完成上述步驟後晶片之製作即完成。加入擴張閥設計之新式改良型晶片完成接合後之實體圖與擴張閥結構顯微近視圖如圖 3.18 所示。

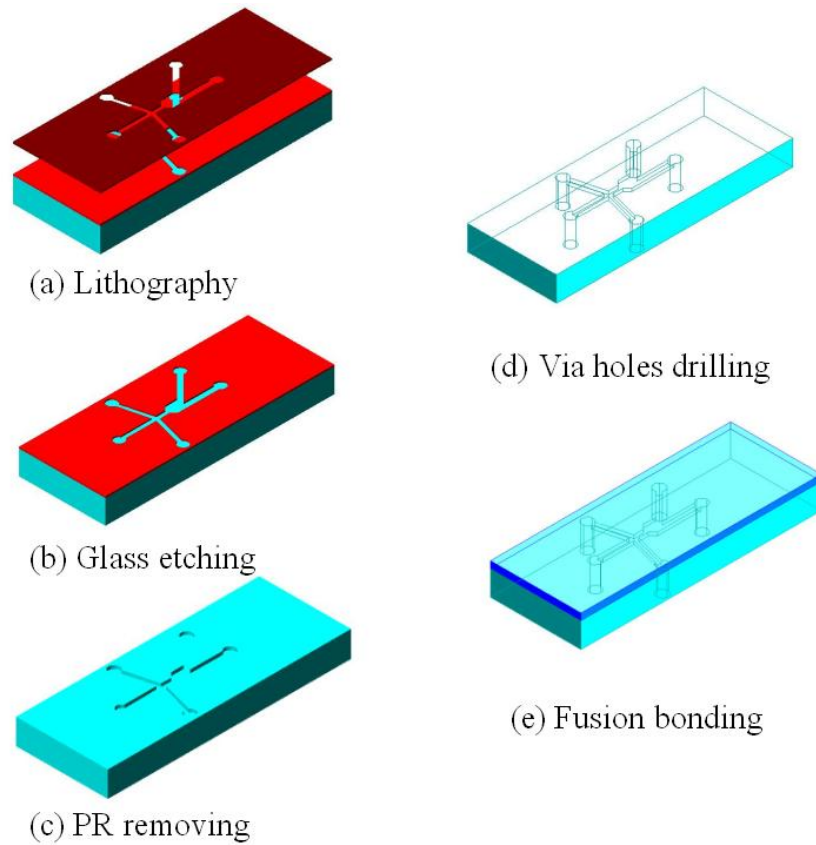


圖 3.17 光切換微流體細胞分類晶片製程示意圖。

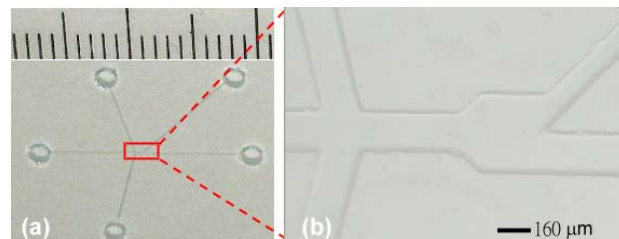


圖 3.18 (a)加入擴張閥設計之新式改良型晶片完成接合後之實體圖(b) 擴張閥結構顯微近視圖。

3.4.3 晶片測試

舊有的晶片主要是利用三個電源供應，對三個樣本槽施加電壓來驅動流體，本實驗室在先前的實驗中已可成功的讓 $10\ \mu\text{m}$ 的 PS 小球在樣品流中流

動，並流至末端收集入特定樣本槽。然而新的架構是以單一電源供應器，配合可變電阻構成的分壓電路來驅動流體，因此新的晶片完成後，尚需實際以 10 μm 之 PS 小球來測試，小球能穩定的在聚焦後的流線中順序流動，並確定不會發生同時有兩個以上的小球通過影像辨識區，造成不利光鉗作動的情況。最後將配合自動化的雷射光鉗細胞分類系統，進行實際的細胞分類測試。

第四章 結果與討論

4.1 微流體樣本流聚焦情形之數值分析與實際實驗結果圖

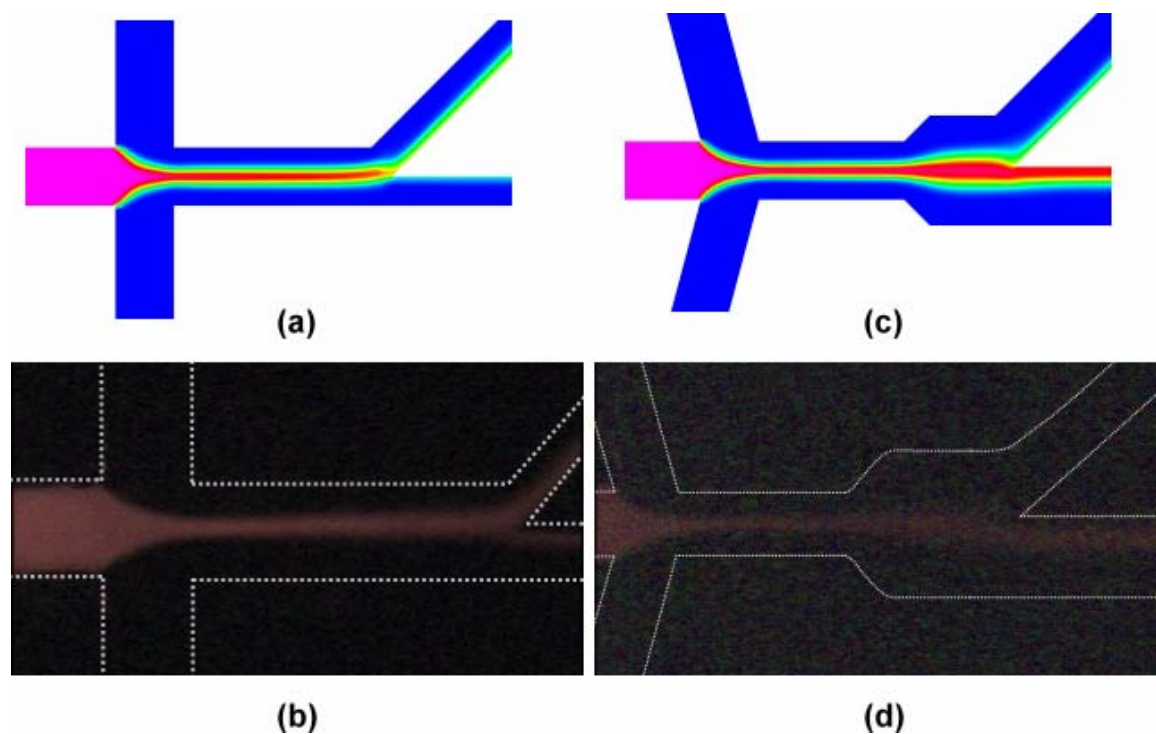


圖 4.1 (a)(b)舊有光罩設計數值模擬計算結果與實際流場聚焦流線之比對。
(c)(d) 加入新式擴張閥之改良型光罩設計數值模擬計算結果與實際流場聚焦流線之比對。

圖 4.1(a)(b)是本實驗室舊有光罩[39]的數值模擬及實際管道內聚焦之情形，數值模擬是使用模擬軟體 CFD-RC 2003 來進行模擬，所有邊界條件電壓皆依實際實驗數值等比例調整至最佳聚焦情形。從所模擬之流場分佈情形可發現，流體實際聚焦情形，和數值模擬之預期大致相同。樣本流經過十字聚焦後，成為寬度約 $15\sim 20\ \mu\text{m}$ 之聚焦流，並在末端分成兩支流進入兩收集槽。圖 4.1(c)(d)則是新光罩的模擬結果與實際聚焦情況之比對，管道

中加入了擴張閥的部份，從模擬結果之向量流線可以清楚看見流線之擴張，即光鉗在前段將粒子分離，則粒子將順著流線擴張，增大與其他粒子間之距離，並流入末端收集槽。同時，管道分岔端高於聚焦流線中心 $10\ \mu\text{m}$ 的設計，使聚焦流主流偏往下管流出，如此光鉗只要給予通過粒子向上之側向吸引力，即可將粒子分離。

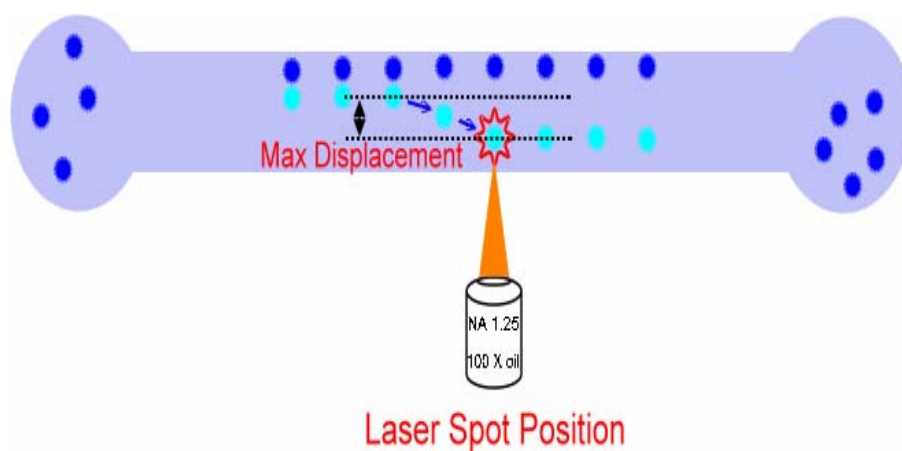


圖 4.2 雷射光鉗捕捉之最大側向位移量測方法示意圖。

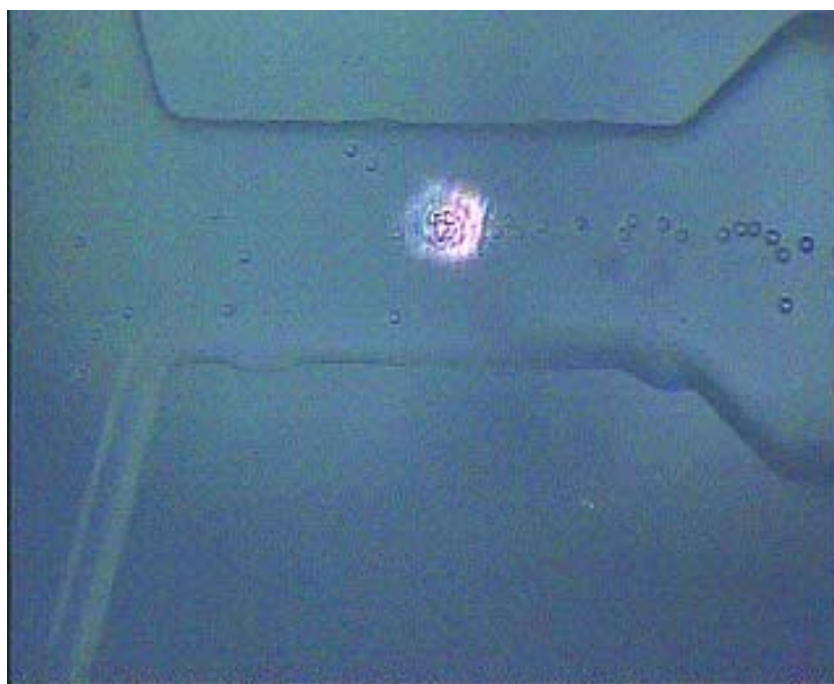


圖 4.3 雷射光鉗捕捉之最大側向位移實際觀察量測影像圖。

4.2 雷射光鉗有效捕捉力與粒子位移關係之實驗量測結果

本實驗將使用直徑為 10 μm 之 Polystyrene 塑膠小球進行雷射光鉗捕捉之側向位移量測。

首先將粒子放入微管道中以電滲透流驅動，且不驅動兩側之微管道聚焦裝置，此時粒子會隨層流穩定分佈於各流線，並以等速度前進，不均勻的分佈於微管道中，如圖 4.2 以及 4.3 所示。此時將固定功率之雷射光鉗聚於管道中心，通過光鉗周遭之粒子會受到一吸引力而產生側向位移，透過觀測並記錄下受光鉗吸引而產生側向位移至光鉗中心之最遠粒子，即可得知該流速及雷射功率下所能提供之”最大粒子位移”。隨著雷射功率的增加，可得雷射功率及粒子位移之關係曲線。圖 4.4 為雷射光鉗之功率在三種不同流速下對粒子產生之側向位移之關係圖表。本研究針對 80 $\mu\text{m/s}$ 、160 $\mu\text{m/s}$ 、320 $\mu\text{m/s}$ 等三種流速進行實驗，從關係圖可知，流速越緩慢，雷射光鉗之側向位移越為顯著，意即雷射光鉗作動之有效範圍較廣，分離時抓取效率較高。

然而，在一般情況下，雷射光鉗會將微粒子鉗住於雷射聚焦處，對於連續式分類之應用將造成中斷之困擾。因此，控制雷射光鉗作用時機並改變欲分類目標粒子之行進方向，為本實驗相當重要的關鍵因素之一。且雷射光鉗作用時間之長短亦會影響連續分類之順暢以及成功與否。目前所使用雷射光鉗作用時間與微粒子移動速度之最佳化比例值為 250 ms : 320 $\mu\text{m/s}$ 。

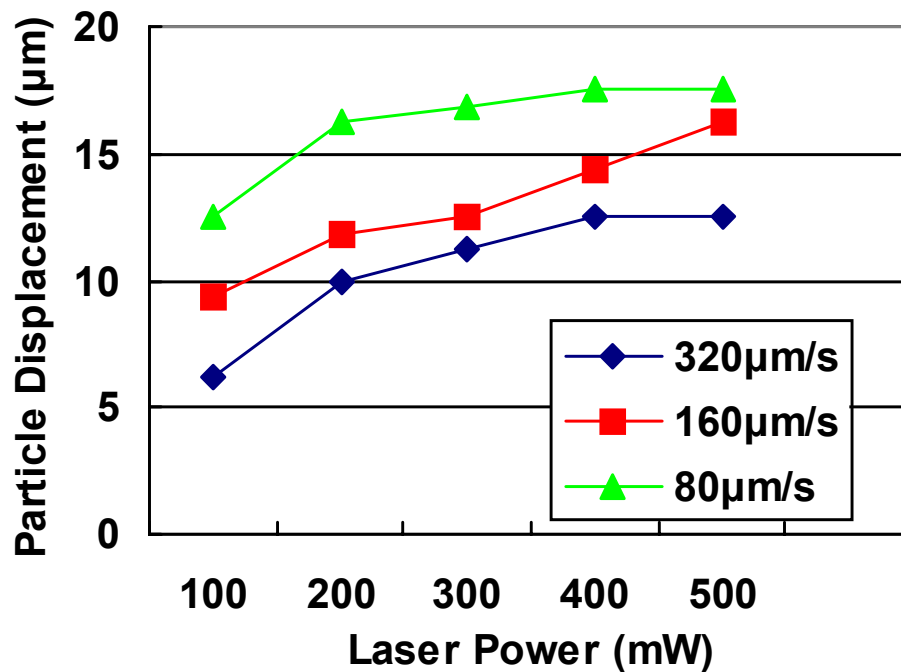


圖 4.4 於 80 $\mu\text{m/s}$ 、160 $\mu\text{m/s}$ 、320 $\mu\text{m/s}$ 等三種不同流速狀況下之雷射光鉗捕捉功率與最大側向位移關係圖。

4.3 樣本流聚焦寬度之控制

本實驗在進行連續式分類之前，必需先行利用兩側微流體聚焦裝置之邊鞘流，進行樣本流聚焦寬度之調整，使得微流體粒子得以一個接著一個依序排列進入影像辨識區及雷射光鉗分類區。其調整之方法乃是利用 3 個高阻抗 30 M Ω 之可變電阻，對樣品入口端及二個聚焦邊鞘流端各別進行分壓調整控制。其電路示意圖如圖 4.5 所示。調整後之樣本流聚焦寬度大約為 15 μm ~ 20 μm 之間，約莫等於單一樣本粒子之直徑大小。調整後之實際聚焦情形如圖 4.6 所示。

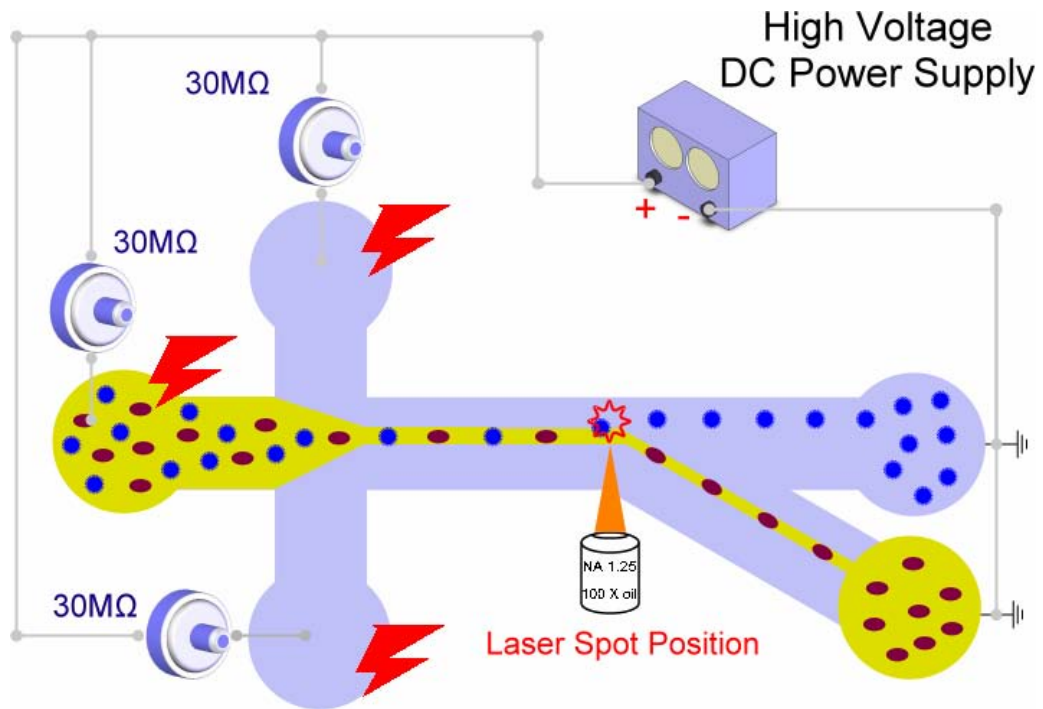


圖 4.5 使用 30 MΩ 高阻抗可變電阻進行樣本流聚焦寬度調整電路示意圖。

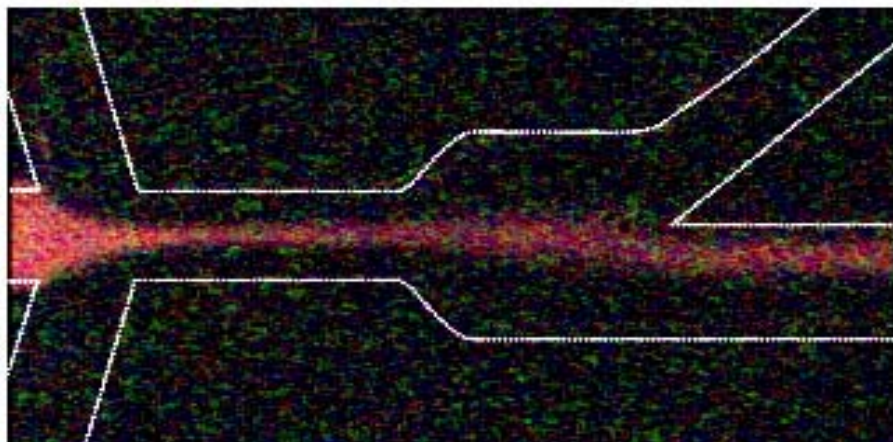


圖 4.6 使用 30 MΩ 高阻抗可變電阻進行調整後之樣本流聚焦寬度，其寬度大約為 15 μm ~ 20 μm 之間。

4.4 即時辨識分類操作畫面

如圖 4.7 所示為本研究所自行開發之細胞/微粒子影像自動分類系統之人機操作介面 HMI (Human-Machine Interface)。沿 X 軸方向所顯示之數值

代表各即時辨識出粒子之於影像平面上之座標位置資訊。為了增進影像辨識效率以及節省 CPU 運算資源，將只針對最靠近雷射光鉗作用區之辨識粒子進行移動速度之運算，其單位為 pixel/sec。當所辨識出之粒子離開辨識區時，此移動速度之資訊將做為決定雷射光鉗作用時間長短之依據。移動速度愈快，光鉗之作用時間愈短，以確保其後接續之微粒子有足夠之時間間隔，維持連續分類之穩定。

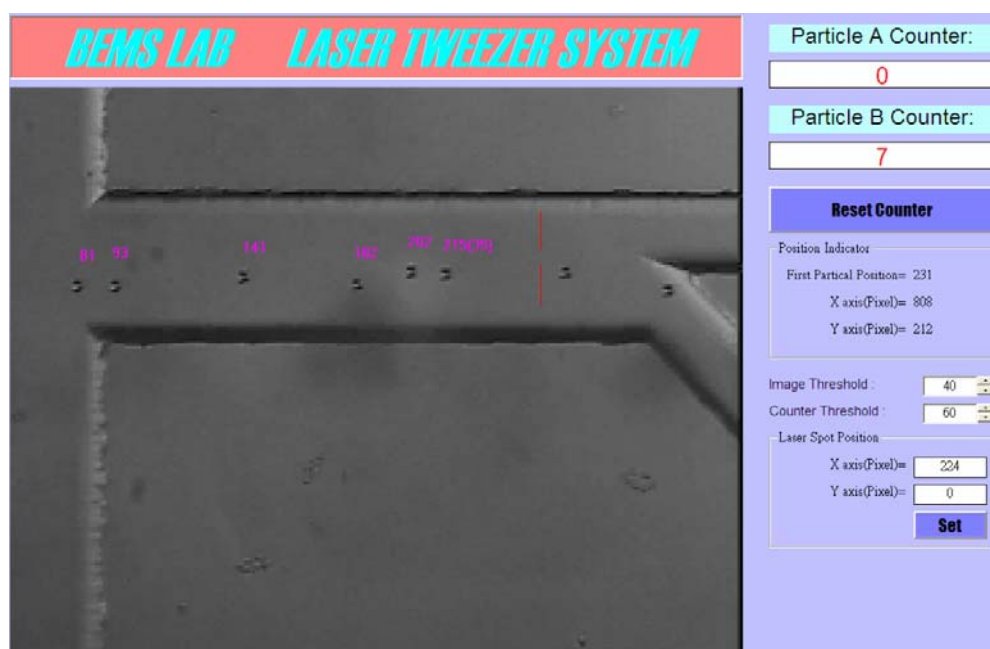


圖 4.7 本研究所自行開發之細胞/微粒子影像自動分類系統之人機操作介面 HMI (Human-Machine Interface)。

4.5 影像即時辨識處理

如圖 4.8(a)所示，樣本流為 5 μm (面積尺寸範圍 10~20 pixel)和 10 μm (面積尺寸範圍 35~45 pixel)微粒子混合液。10 μm 為所設定之自動分類目標粒

子。經過即時二值化影像處理，如圖 4.8(b)所示，微粒子質點隨即被分離出來，影像辨識軟體將自動標示出符合設定條件(35~45 pixel)之實體物件質點，如圖 4.8(a)(b)所示之紅色標示點，並可於其周圍即時顯示所辨識質點大小(單位為 pixel)之資訊，如圖 4.8(a)所示以及座標位置(單位為 pixel)和速度(單位為 pixel/sec)之資訊，如圖 4.8(b)所示。當被標示出之粒子即將進入雷射光鉗作用區時，其將觸發雷射光鉗訊號並進行分類之抓取。其雷射光鉗作用之時機將依所辨識出之座標位置決定。其作用時間之長短將依所計算出之移動速度來決定。

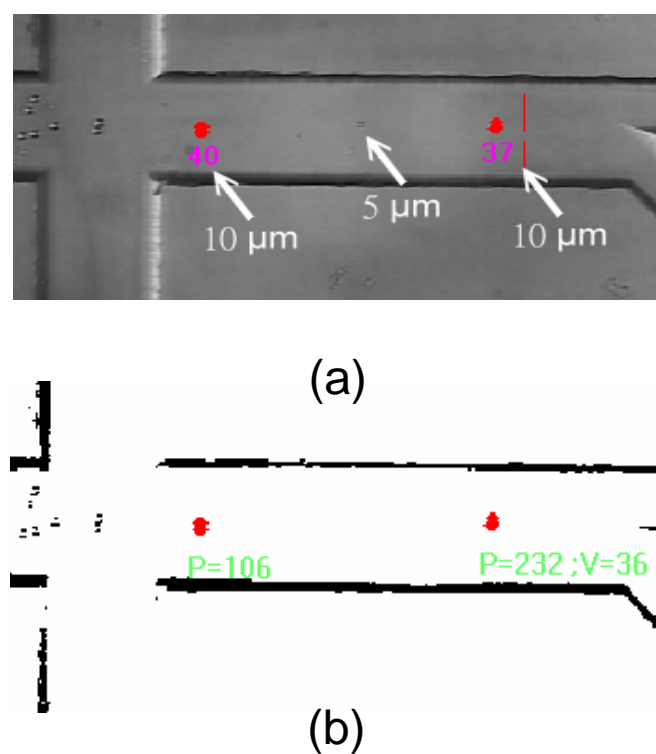


圖 4.8 電腦即時影像辨識處理之(a)灰階畫面與(b)二值化比對圖。

4.6 不同目標粒子之混合樣品流分類結果

圖 4.9 所示表示以 $10\ \mu\text{m}$ 為自動分類目標粒子之連續圖像，樣本流為 $5\ \mu\text{m}$ 和 $10\ \mu\text{m}$ 微粒子混合液，如圖 4.9(a)所示，粉紅色箭頭指向一 $10\ \mu\text{m}$ 之微粒子，以大約 $150\ \mu\text{m/s}$ 的速度順著流動方向前進，其餘黃色箭頭標示則為 $5\ \mu\text{m}$ 之微粒子。

當微粒子通過前段作動區，會受到邊峭流的效應，形成聚焦流，又因為微流體的層流性質，粒子會穩定的順著流層流動。同時，粒子通過前段作動區時，影像辨識軟體會依粒子的像素大小判定目標粒子是否需要被分類。由 4.9(b)(c)可見， $5\ \mu\text{m}$ 之微粒子通過作動區前段，因即時影像傳回電腦，辨識後確認並非需要分離之粒子，因此雷射並沒有作動，而粒子則隨之流入擴張閥，並穩定順著層流流往下放收集槽並被收集。如 4.9(d)所示， $10\ \mu\text{m}$ 的粒子被偵測，因此雷射開始作動，並使其產生約 $20\ \mu\text{m}$ 之側向位移，改變粒子的流層位置。如圖 4.9(d)(e)(f)所示， $10\ \mu\text{m}$ 粒子受到光鉗作動後，進入擴張閥，並順著流線流入上端收集槽，如圖 4.9(f)(g)(h)。如圖 4.9(c)(d)(e)所示，未受到光鉗作用之 $5\ \mu\text{m}$ 粒子，將穩定流往其中一收集槽。

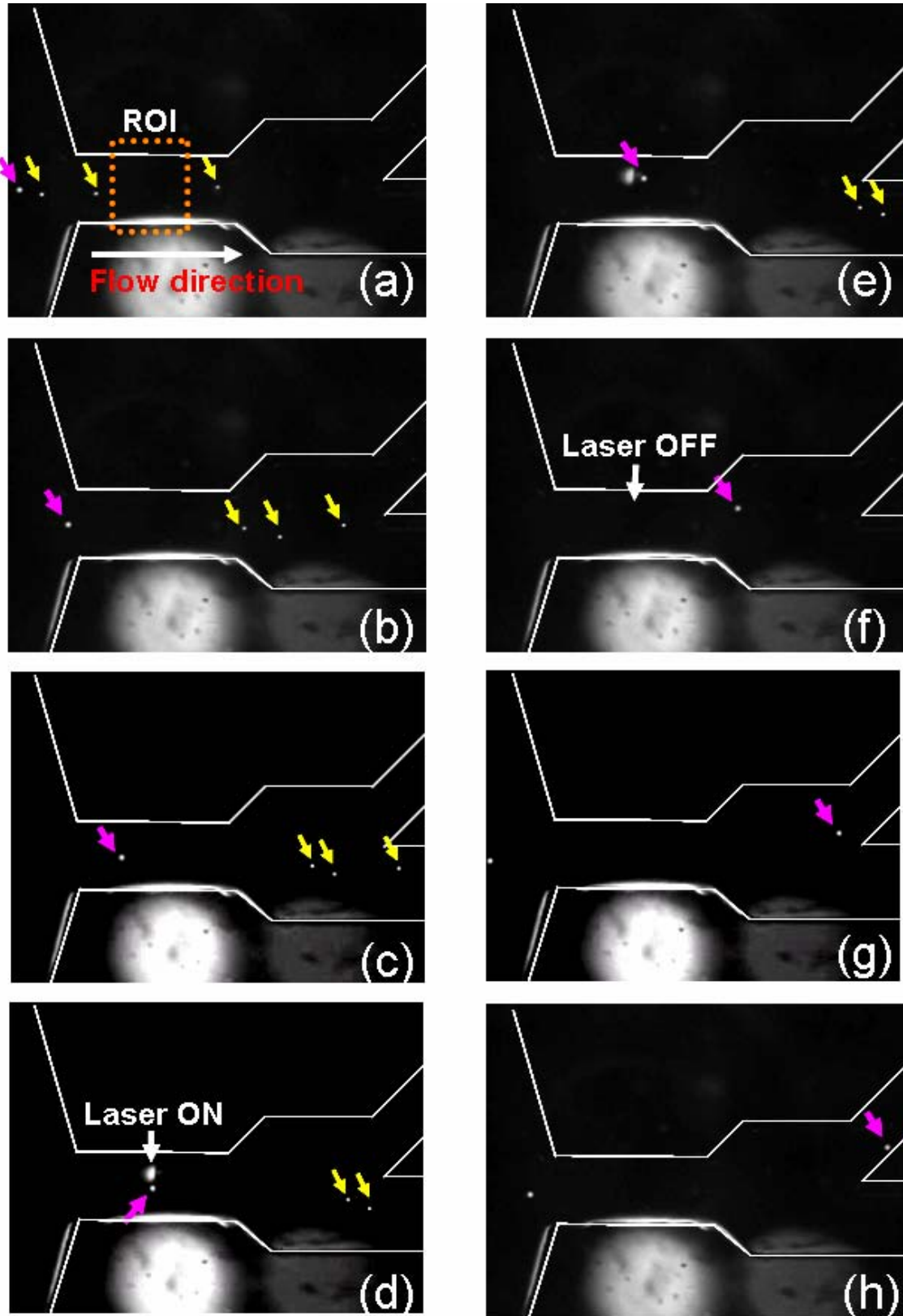


圖 4.9 以 $10\ \mu\text{m}$ 為目標粒子之自動光鉗粒子分類連續圖像。

如圖 4.10 所示為以 $5\ \mu\text{m}$ 為自動分類目標粒子之連續圖像，樣本流為 $5\ \mu\text{m}$ 和 $10\ \mu\text{m}$ 微粒子混合液，如圖 4.10(a)所示，黃色箭頭指向一 $5\ \mu\text{m}$ 之微粒子，以大約 $300\ \mu\text{m/s}$ 的速度順著流動方向前進，粉紅色箭頭標示則為 $10\ \mu\text{m}$ 之微粒子。

由圖 4.10(a)可見，當 $5\ \mu\text{m}$ 之微粒子通過作動區前段，因即時影像傳回電腦，辨識後確認為需要分離之目標粒子，因此雷射開始作動，使其產生約 $20\ \mu\text{m}$ 之側向位移，並改變粒子的流層位置。如圖 4.10(b)(c)(d)所示， $5\ \mu\text{m}$ 粒子受到光鉗作動後，進入擴張閥，並順著流線流入上端收集槽，如圖 4.10(e)(f)(g)所示。由圖 4.10(d)(e)可見， $10\ \mu\text{m}$ 之微粒子通過作動區前段，因即時影像傳回電腦，辨識後確認並非需要分離之粒子，因此雷射並沒有作動，而粒子則隨之流入擴張閥，並穩定順著層流流往下放收集槽並被收集，如圖 4.10(f)(g)(h)所示。

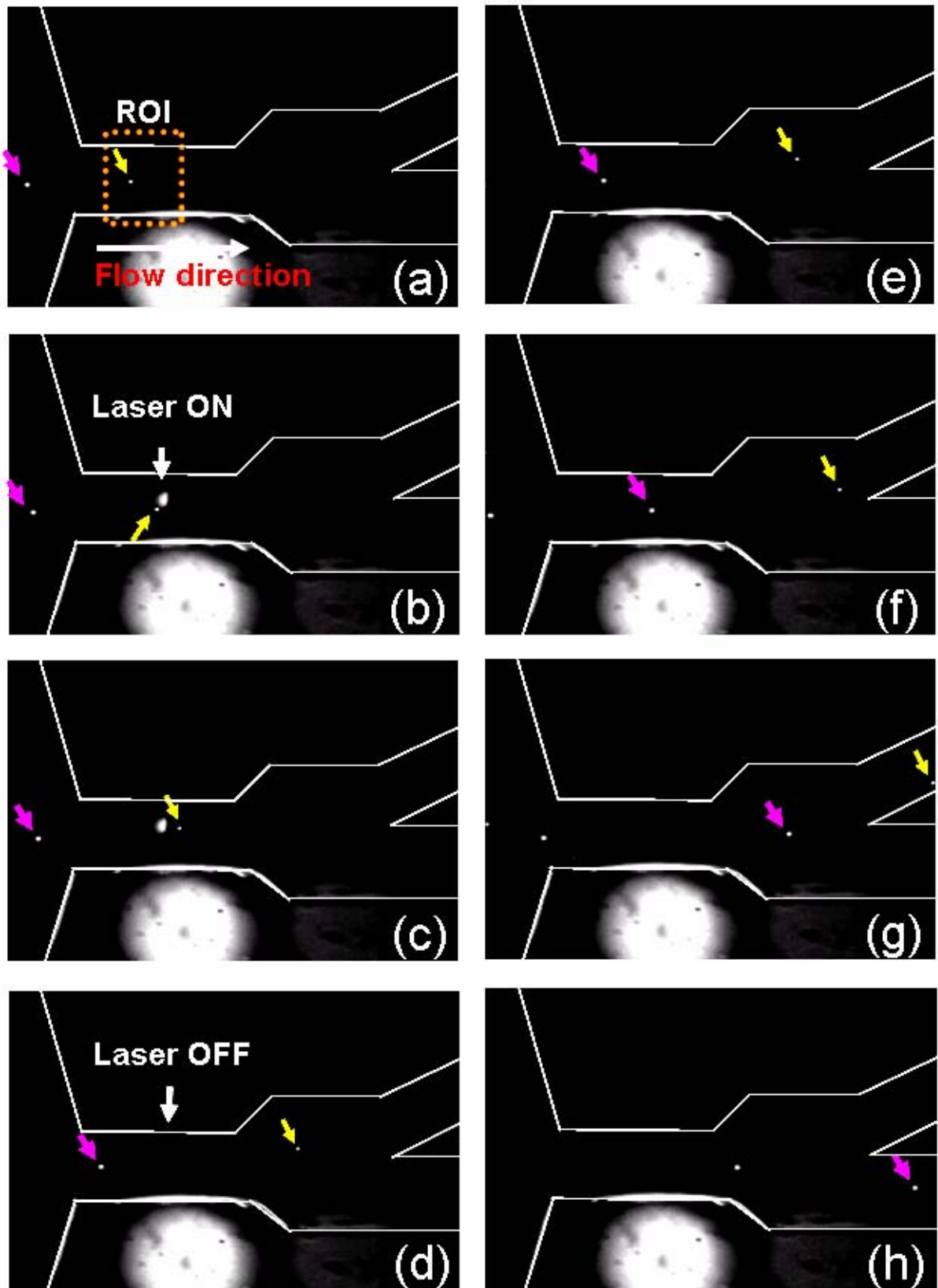


圖 4.10 以 5 μm 為目標粒子之自動光鉗粒子分類連續圖像。

此實驗成功展示了全自動影像辨識分類系統之分類能力。並可任意依據使用者之設定來分類不同大小之目標微粒子。受光鉗作用之粒子可成功被改變流層，流入另一端收集槽以完成分類、儲存。同時，整個分離過程皆由電腦迴授判讀，決定光鉗之作動與否，成功展示一全自動化即時影像辨識系統，結合雷射光鉗之細胞/粒子分類器，並成功的將混合粒子分離。

4.7 使用軟體偵測方式之粒子計數器

如圖 4.11 表示軟體計數器之灰階值掃瞄偵測作動連續示意圖。如圖 4.12 表示連續對 5 μm 以及 10 μm 之微粒子偵測計數之實驗結果。

針對利用灰階值之變化所進行之軟體偵測計數應用，為了瞭解其實際之偵測行為，本研究設計了如下之實驗：

實驗中實際將 5 μm 以及 10 μm 之 PS 小球放入微管道中進行實際分類，並於計數之子程式內特別設計一資料收集之副程式。對於所擷取之每一張影像，先將偵測之影像經過二值化處理，軟體計數偵測線所連續掃描量測到之最大變化灰階值將被所自行撰寫之程式自動收集下來，其影像擷取之系統時間亦將同時被記錄。之後再將所收集之灰階值資料以及其量測時之系統時間整理後，可得到如圖 4.12 之連續偵測結果。縱座標 Y 軸表示灰階值變化，橫座標 X 軸表示量測之時間區間。由於偵測之影像已經過二值化

處理，因此其量測之灰階值相當容易區別，只有 0 與 255 之偵測值，因此所設定之閾值可於 0 與 255 之間任意取一個值，即可達到計數偵測之效果。

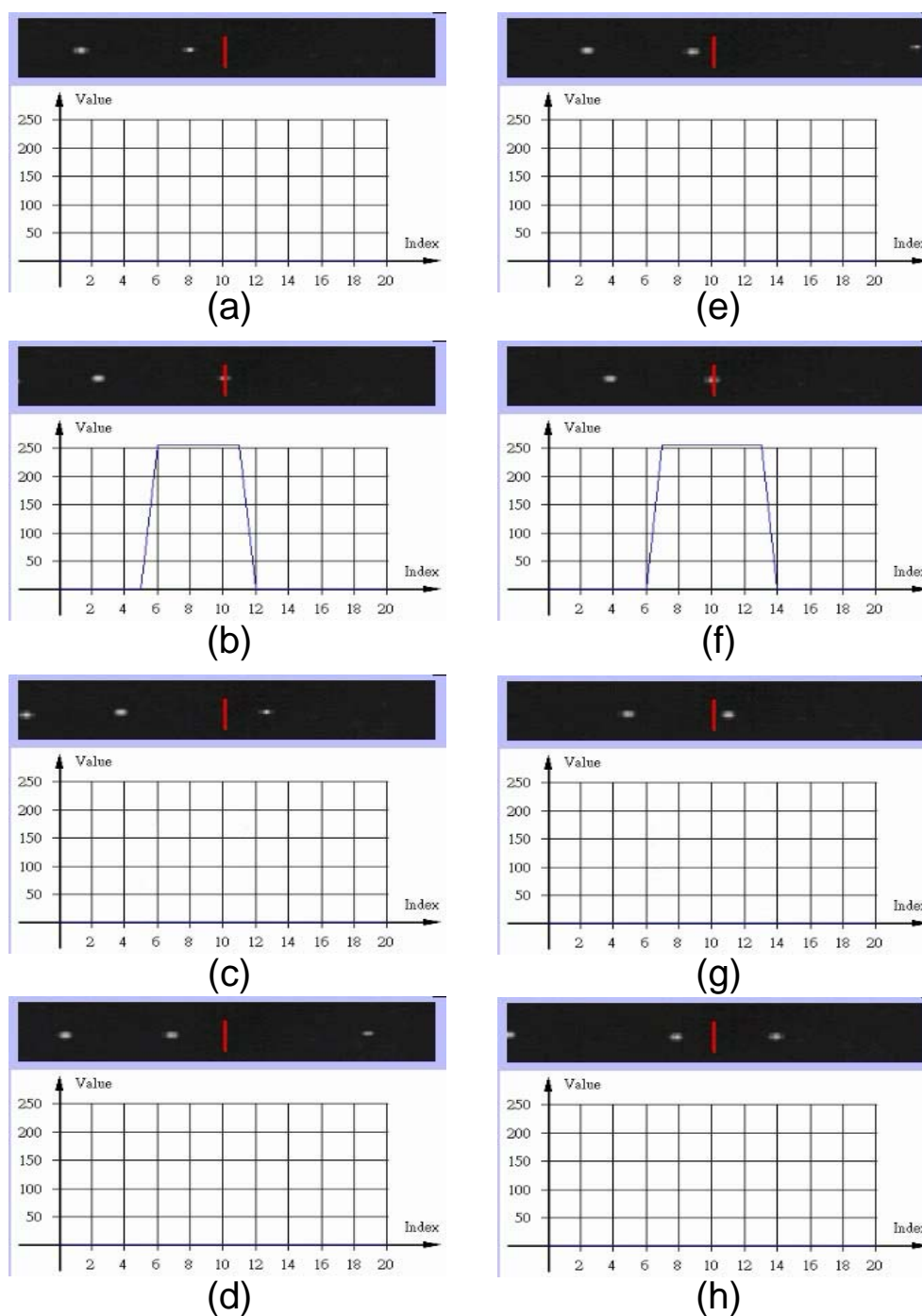


圖 4.11 軟體計數器之灰階值掃瞄偵測作動連續示意圖。

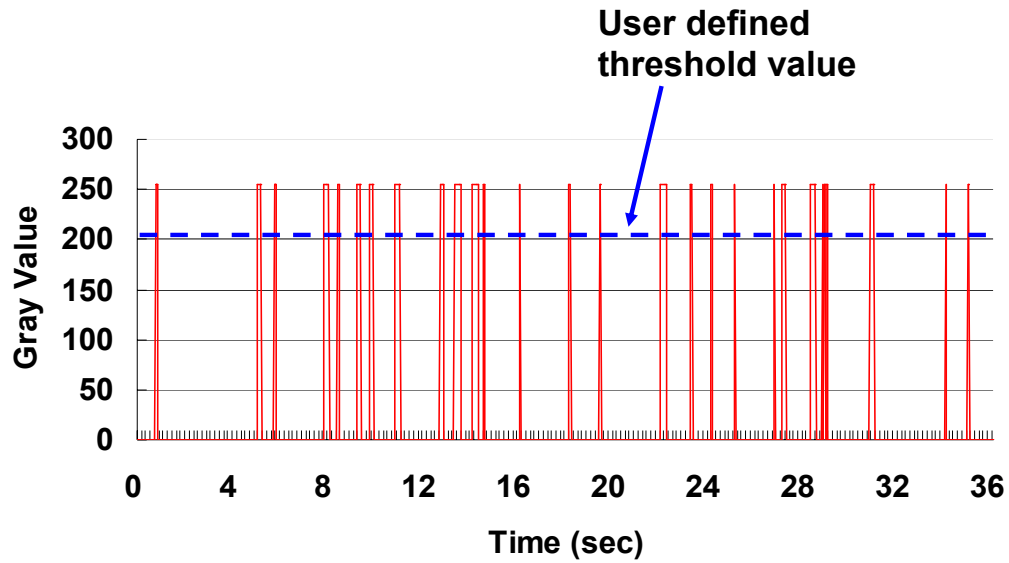


圖 4.12 使用自行開發之軟體計數器於微管道內之計數結果。

在進行本實驗時發現了一個現象。若細胞質點行進之速度過快，使得連續兩張取像之細胞質點影像恰位於偵測線之兩側時，由於軟體計數器偵測線只是一個像素(Pixel)所集成之直線，此時軟體計數器偵測線將無法感測並掃描到細胞質點通過時之灰階值變化，因而造成遺漏計數之錯誤。其解決的方法為：將電滲透流之驅動電壓降低，以便控制並降低細胞質點之移動速度。亦或是將偵測影像取像之立體顯微鏡倍率再調高，增加單一顆細胞質點之影像像素。使得其單一顆細胞質點之影像可穩定通過軟體計數器偵測線，計數之程式即可偵測其灰階值變化而進行計數之功能。

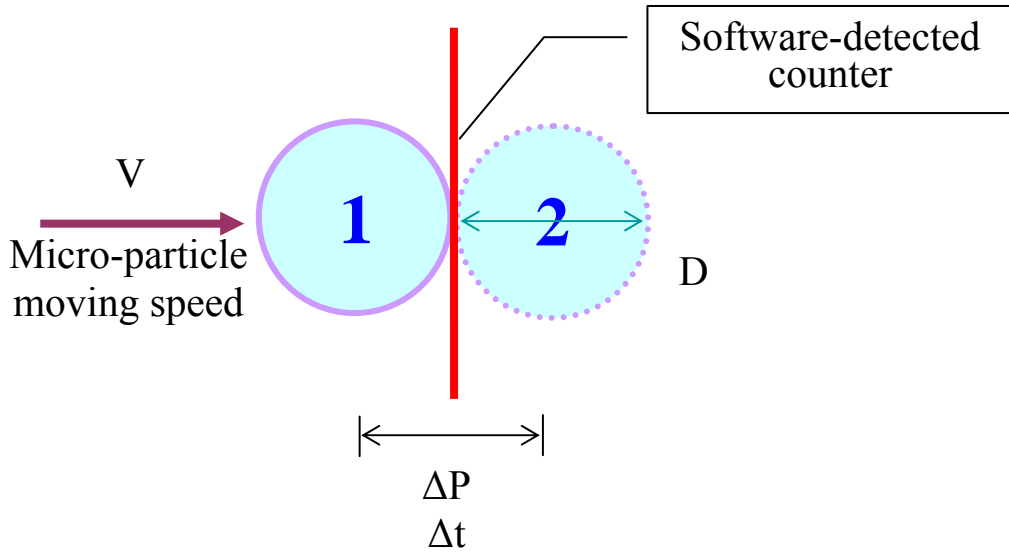


圖 4.13 連續兩張取像之細胞質點影像恰都位於偵測線之兩側時，軟體計數器偵測線將無法感測到細胞質點通過時之灰階值變化。

4.8 軟體計數器之性能分析

由於此現象之發現，吾人可藉此來計算此軟體計數器之偵測速度範圍。如圖 4.13 所示，欲穩定偵測單一細胞質點通過軟體計數器偵測線之灰階值變化，於連續兩張擷取影像內，單一細胞質點之移動位移不可大過其本身之影像直徑。亦即

$$\Delta P < D \quad (4-1)$$

又

$$\Delta P = V \times \Delta t \quad (4-2)$$

其中

ΔP : 單一細胞質點於連續兩張影像取像內之移動位移

V : 單一細胞質點之移動速率

Δt : 連續兩張影像擷取之時間差

D : 單一細胞質點之影像直徑

所以

$$V \times \Delta t < D \quad (4-3)$$

$$V < D / \Delta t \quad (4-4)$$

因此其最大速率限制取決於欲偵測之細胞質點之影像大小以及連續兩張影像擷取之時間差。

由於 Microsoft Windows 作業系統為多工之架構，每一個執行緒所分配到的系統資源並非總是一定，因此取像時的時間差亦並非相同。

表 4-1 顯示之資料內容為軟體計數器於 1 sec 內所收集之取像時間記錄，其內容包含軟體計數器偵測線所連續掃瞄到之灰階值變化以及系統時間差之記錄值，其偵測之細胞質點樣本為 5 μm 以及 10 μm 之 PS 小球。分析其取像之時間差可得知，最大之取像時間差為 52 ms。最小之取像時間差為 28 ms。

表 4.1 於 1 sec 內軟體計數器偵測線所連續掃瞄到之灰階值變化以及影像辨識系統於連續兩張取像時間差之記錄值。紅色字體標示表示軟體取像程式所擷取到之最小及最大時間差之記錄值。

System Timer(mini-sec)	Image Capture Timing(mini-sec)	Image Capture Period(mini-sec)	Gray Value
68276043	0	0	0
68276089	46	46	0
68276122	79	33	0
68276166	123	44	0
<i>68276194</i>	<i>151</i>	<i>28</i>	<i>0</i>
68276227	184	33	255
68276269	226	42	255
68276307	264	38	255
68276344	301	37	255
68276388	345	44	255
68276431	388	43	0
68276464	421	33	0
68276496	453	32	0
68276538	495	42	0
68276571	528	33	0
<i>68276623</i>	<i>580</i>	<i>52</i>	<i>0</i>
68276667	624	44	0
68276699	656	32	0
68276746	703	47	0
68276790	747	44	0
68276821	778	31	0
68276851	808	30	0
68276883	840	32	0
68276916	873	33	0
68276948	905	32	255
68276977	934	29	255
68277009	966	32	255
68277041	998	32	0
68277070	1027	29	0
68277103	1060	33	0

由式 4-4 可求出，當取像時間差為 52 ms 時，針對不同大小尺寸微粒子，影像辨識系統軟體計數器之其最大可偵測移動速率範圍限制 V_{max} 為：

對於 5 μm 之移動速率限制為 $5 \mu\text{m}/0.052 \text{ sec}=96 \mu\text{m}/\text{sec}$.

對於 10 μm 之移動速率限制為 $10 \mu\text{m}/0.052 \text{ sec}=192 \mu\text{m}/\text{sec}$.

由式 4-4 可求出，當取像時間差為 28 ms 時，針對不同大小尺寸微粒子，影像辨識系統軟體計數器之其最大可偵測移動速率範圍限制 V_{max} 為：

對於 5 μm 之移動速率限制為 $5 \mu\text{m}/0.028 \text{ sec}=179 \mu\text{m}/\text{sec}$.

對於 10 μm 之移動速率限制為 $10 \mu\text{m}/0.028 \text{ sec}=357 \mu\text{m}/\text{sec}$.

因此，由上述所推論，若能有效增加影像之取像頻率，以減少連續兩張影像取像之時間差，則軟體計數器之性能對於細胞質點之最大移動速率限制以及計數之正確率將大為提昇。

第五章 結論與未來展望

5.1 結論

本研究成功的使用雷射光鉗顯微鏡系統，配合即時影像辨識軟體，完成自動化混合粒子的分類。其共達成了如下之成果：

- 全自動影像辨識分類軟體之開發，並可任意依據使用者之設定來分類不同大小之目標微粒子。
- 利用改良微管道之幾何形狀，以達到系統分離效率之提升。
- 自動化之雷射光鉗架設及與 PC-Based 電腦系統之連結控制。
- 成功地以 5 μm 或 10 μm 為自動分類目標粒子，並將其自樣本流為 5 μm 和 10 μm 之微粒子混合液中分離出來。
- 針對自行開發之軟體計數器進行功能測試及其性能分析。

雷射光鉗系統是一種乾淨的操控方式，不僅能對微粒子做精準的操作，亦能夠避免許多接觸式操作可能產生的感染問題，對於實際應用於生醫領域將會有很大的助益。此外，簡單的微管道設計與低成本之材料，亦提供未來發展拋棄晶片之可能性。

5.2 未來展望

為了提昇軟體計數器之性能，並增加軟體計數器之偵測速度範圍，於現有之取像條件下，可以對於每一個軟體計數器，使用兩組偵測線來對單一

細胞質點進行灰階值變化之掃瞄，再利軟體程式進行條件式控制，以增加軟體計數器所能辨識之最大速度限制性能並提高計數之正確率。

對於座標、移動速度及粒子大小之影像與實際尺寸轉換，目前因必須時常調整焦距，轉換不易。將來參數固定後，可直接進行實際尺寸換算，即可立即得知微粒子之實際位移速度，其單位為 $\mu\text{m}/\text{sec}$ 而非目前之 pixel/sec 。如此，將可方便使用者進行性能及實際位移之比較。

光鉗運用在粒子分類，主要是運用光鉗對粒子所產生之橫向位移來完成。因此，如何增加其聚焦深度(depth-of-focus)使流道中任何流層的粒子都能夠被吸引，甚至利用調整聚焦方式，使雷射光點形成類似條狀(laser bar)來增加側向位移，亦是未來最佳化的發展方向。此系統目前已開始嘗試實際生物細胞樣本之操作，未來將可實現自動化以及快速化之生醫檢測目標。全自動化的即時影像辨識，若能配合細胞粒子雷射激發光收集偵測功能的改良，將可使整個細胞粒子操作過程更加精確且快速。其可拋棄式晶片的應用以及未來結合細胞粒子雷射激發光收集偵測功能的構想，將使得此系統有機會取代大型之流式細胞儀設備，並提高了未來商業化之可能性。

參考文獻

- [1] A. Manz, N. Graber, and H. M. Widmer, "Miniaturized total chemical analysis systems: A novel concept for chemical sensing," *Sensors and Actuators B: Chemical*, vol. 1, pp. 244-248, 1990.
- [2] D. J. Harrison, K. Fluri, K. Seiler, Z. Fan, C. S. Effenhauser, and A. Manz, "Micromachining a Miniaturized Capillary Electrophoresis-Based Chemical Analysis System on a Chip," *Science*, vol. 261, pp. 895-897, 1993.
- [3] J. Enger, M. Goksör, K. Ramser, P. Hagberg, and D. Hanstorp, "Optical tweezers applied to a microfluidic system," *Lab on a Chip*, vol. 4, pp. 196-200, 2004.
- [4] S. Fiedler, S. G. Shirley, T. Schnelle, and G. Fuhr, "Dielectrophoretic Sorting of Particles and Cells in a Microsystem," *Analytical Chemistry*, vol. 70, pp. 1909-1915, 1998.
- [5] A. Y. Fu, H. P. Chou, C. Spence, F. H. Arnold, and S. R. Quake, "An Integrated Microfabricated Cell Sorter," *Analytical Chemistry*, vol. 74, pp. 2451-2457, 2002.
- [6] P. C. H. Li and D. J. Harrison, "Transport, Manipulation, and Reaction of Biological Cells On-Chip Using Electrokinetic Effects," *Analytical Chemistry*, vol. 69, pp. 1564-1568, 1997.
- [7] M. G. Ormerod, *Flow cytometry - A Practical Approach*, 3rd ed: Oxford University Press, UK, 2000.
- [8] R. Pethig, "Dielectrophoresis: Using inhomogeneous AC electrical fields to separate and manipulate cells," *Critical Reviews in Biotechnology*, vol. 16, pp. 331-348, 1996.
- [9] H. Jagannathan, G. G. Yaralioglu, A. S. Ergun, and B. T. Khuri-Yakub, "An implementation of a microfluidic mixer and switch using micromachined acoustic transducers," *Proc. of IEEE 16th International MEMS Conference*, pp. 104-107, 2003.
- [10] G. M. Dougherty and A. P. Pisano, "Ultrasonic particle manipulation in microchannels using phased co-planar transducers," *Solid State Sensors and Actuators*, vol. 1, pp. 670-673, 2003.
- [11] E. Altendorf, D. Zebert, M. Holl, and P. Yager, "Differential blood cell counts obtained using a microchannel based flow cytometer," *Solid State Sensors and Actuators*, vol. 1, pp. 531-534, 1997.
- [12] K. H. Heng, W. Wang, M. C. Murphy, and K. Lian, "UV-LIGA Microfabrication and Test of an AC-Type Micropump Based on the Magnetohydrodynamic (MHD) Principle," *Proc. of SPIE*, vol. 4177, pp. 161-171, 2000.
- [13] M. Ozkan, M. Wang, C. Ozkan, R. Flynn, and S. Esener, "Optical Manipulation of Objects and Biological Cells in Microfluidic Devices," *Biomedical Microdevices*, vol. V5, pp. 61-67, 2003.
- [14] T. N. Buican, M. J. Smyth, H. A. Crissman, G. C. Salzman, C. C. Stewart, and J. C. Martin, "Automated single-cell manipulation and sorting by light trapping," *Applied*

- Optics*, vol. 26, pp. 5311, 1987.
- [15] A. Ashkin, J. M. Dziedzic, and T. Yamane, "Optical trapping and manipulation of single cell using infrared laser beams," *Nature*, vol. 330, pp. 769-771, 1987.
- [16] A. Ashkin and J. M. Dziedzic, "Optical trapping and manipulation of viruses and bacteria," *Science*, vol. 235, pp. 1517-1520, 1987.
- [17] M. M. Wang, E. Tu, D. E. Raymond, J. M. Yang, H. Zhang, N. Hagen, B. Dees, E. M. Mercer, A. H. Forster, I. Kariv, P. J. Marchand, and W. F. Butler, "Microfluidic sorting of mammalian cells by optical force switching," *Nature Biotechnology*, vol. 23, pp. 83 - 87, 2004.
- [18] Z. Darzynkiewicz, J. P. Robinson, and H. A. Crissman., *Flow cytometry*, 2nd ed. San Diego, Calif: Academic Press, 1994.
- [19] J. P. Freyer, M. E. Wilder, P. L. Schor, J. Coulter, and M. R. Raju, "A simple electronic volume cell sorter for clonogenicity assays," *Cytometry*, vol. 10, pp. 273-281, 1989.
- [20] 謝世良, "流式細胞技術 Flow Cytometry," 國立陽明大學 免疫學研究中心.
- [21] G. B. Lee, C. H. Lin, L. M. Fu, R. J. Yang, and B. H. Hwei, "Fabrication and Application of Micro Flow Cytometers," *Journal of Instruments Today*, vol. 131, 2002.
- [22] L. S. Cram, J. C. Martin, J. A. Steinkamp, T. M. Yoshida, T. N. Buican, B. L. Marrone, J. H. Jett, G. Salzman, and L. Sklar, "New flow cytometric capabilities at the national flow cytometry resource," *Proceedings of the IEEE*, vol. 80, pp. 912-917, 1992.
- [23] J. Wietzorrek, M. Stadler, and V. Kachel, "Flow Cytometric Imaging - a novel tool for identification of marine organisms," *Oceans Engineering for Today's Technology and Tomorrow's Preservation*, vol. 1, pp. I/688 - I/693, 1994
- [24] R. Rong, J. W. Choi, and C. H. Ahn, "A functional magnetic bead-biocell sorter using fully integrated magnetic micro-nano tips," *Proc. of IEEE 16th International MEMS Conference*, pp. 530-533, 2003.
- [25] C. Pei Yu, A. T. Ohta, and C. W. Ming, "Continuous optical sorting of HeLa cells and microparticles using optoelectronic tweezers," *Proc. of IEEE/LEOS International Optical MEMS Conference*, pp. 83-84, 2005.
- [26] 劉宜德, "軟性電路板自動化視覺導引鑽孔系統之研發," 國立中山大學 電機工程學系, 2001.
- [27] J. P. Lewis, "Fast normalized cross-correlation," *Vision Interface*, pp. 120-123, 1995.
- [28] B. H. Friemel, L. N. Bohs, and G. E. Trahey, "Relative performance of two-dimensional speckle-tracking techniques: normalized correlation, non-normalized correlation and sum-absolute-difference," *Proc. of IEEE Ultrasonics Symposium*, vol. 2, pp. 1481-1484, 1995.
- [29] K. Briechle and U. D. Hanebeck, "Self-localization of a mobile robot using fast normalized cross correlation," *Proc. of IEEE International SMC Conference*, vol. 4, pp. 720-725, 1999.

- [30] A. Ashkin, "Acceleration and Trapping of Particles by Radiation Pressure," *Physical Review Letters*, vol. 24, pp. 156, 1970.
- [31] A. Ashkin, J. M. Dziedzic, J. E. Bjorkholm, and S. Chu, "Observation of a single-beam gradient force optical trap for dielectric particles," *Optics Letters*, vol. 11, pp. 288-290, 1986.
- [32] A. Ashkin, "History of optical trapping and manipulation of small-neutral particle, atoms, and molecules," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 6, pp. 841-856, 2000.
- [33] 黃煥常, "雷射光鉗技術簡介及其應用," *國家衛生研究院電子報* 第 96 期, 2005.
- [34] A. Ashkin, "Forces of a single-beam gradient laser trap on a dielectric sphere in the ray optics regime," *Biophysical Journal*, vol. 61, pp. 569-582, 1992.
- [35] 林豐勝, "光學懸浮之軸向與橫向作用力之探討(Axial and Transverse Trapping Forces. In Optical Levitation)," 國立東華大學 電機工程研究所, 2002.
- [36] R. J. Hunter, *Zeta potential in colloid science: principles and applications*. New York: Academic Press, 1981.
- [37] G. M. Mala, D. Li, C. Werner, H. J. Jacobasch, and Y. B. Ning, "Flow characteristics of water through a microchannel between two parallel plates with electrokinetic effects," *International Journal of Heat and Fluid Flow*, vol. 18, pp. 489-496, 1997.
- [38] C. Yang and D. Li, "Electrokinetic effects on pressure-driven liquid flows in rectangular microchannels," *Journal of Colloid and Interface Science*, vol. 194, pp. 95-107, 1997.
- [39] 簡育生, "電驅動式微混合器及整合光鉗之細胞操控平台於生醫檢測之應用," 國立中山大學 機械與機電工程學系, 2004.
- [40] R. F. Probstein, *Physicochemical Hydrodynamics: An Introduction*, 2nd ed. New York: Wiley and Sons, 1999.
- [41] S. V. Ermakov, S. C. Jacobson, and J. M. Ramsey, "Computer Simulations of Electrokinetic Transport in Microfabricated Channel Structures," *Analytical Chemistry*, vol. 70, pp. 4494-4504, 1998.
- [42] S. C. Jacobson and J. M. Ramsey, "Electrokinetic focusing in microfabricated channel structures," *Analytical Chemistry*, vol. 69, pp. 3212-3217, 1997.
- [43] G. B. Lee, L. M. Fu, R. J. Yang, Y. J. Pan, and C. H. Lin, "MxN micro flow switches using electrokinetic forces," *Solid State Sensors and Actuators*, vol. 2, pp. 1895-1898, 2003.
- [44] J. Takagi, M. Yamada, M. Yasuda, and M. Sek, "Continuous particle separation in a microchannel having asymmetrically arranged multiple branches," *Lab on a Chip*, vol. 5, pp. 778-784, 2005.
- [45] C. H. Lin, G. B. Lee, Y. H. Lin, and G. L. Chang, "A fast prototyping process for fabrication of microfluidic systems on soda-lime glass," *Journal of Micromechanics and Microengineering*, vol. 11, pp. 726-732, 2001.

Appendix A

Image Capture Sub-Program (Programmed by using Microsoft® Visual C++)

```
//-----  
//2006:by Bio-Electro-Mechanical System Lab.  
//Department of Mechanical and Electro-Mechanical Engineering,  
//National Sun Yat-sen University,  
//70 Lien-hai Rd. Kaohsiung 804, Taiwan  
//-----  
/**  
 \file    Modified from ©Basler Bcam MiniGrabSample.cpp  
 \brief   grab a single image using Bcam-API calls  
*/  
  
#include "stdafx.h"  
#include "define.h"  
#include "mem_def_hmi.h"  
#include "resource.h"  
#include <bvcdib.h>  
#include <bcam.h>  
#include "BvcColorConverter.h"  
extern SHM *shm;  
using namespace std;  
using namespace Bcam;  
  
int main(int argc, char* argv[])  
{  
    create_share_memory();  
    open_share_memory();  
    int i=0;  
    if (CBcam::DeviceNames().size() == 0)  
    {  
        cerr << "No camera present!" << endl;  
        system ("pause");  
        return 1;  
    }  
}
```

```

}

// Get the devicename of the first camera
CString DeviceName = *(CBcam::DeviceNames().begin());

// Create the driver object and open the driver
CBcam Bcam;
Bcam.Open(DeviceName);
Bcam.Shutter.Raw=400;
Bcam.Brightness.Raw=255;
Bcam.Gain .Raw=500;

//=====
//Continuous Grab:DCAM Formate 7
// Parameter set for the fastest monochrome format (valid assumption only for Basler cameras)
const DCSVideoFormat VideoFormat = DCS_Format7;
const DCSVideoMode VideoMode      = DCS_Mode0;
DCSColorCode ColorCode           = DCSColor_Mono8;
CString ModelName = Bcam.Info.ModelName();
if(ModelName.Find("c") >= 0)

Bcam.SetVideoMode(VideoFormat, VideoMode);
Bcam.FormatSeven[VideoMode].ColorCoding = ColorCode;
// Set Area Of Interest to maximum
CSize ImageSize = Bcam.FormatSeven[VideoMode].MaxSize();
ImageSize.cx  = ImageSize.cx & ~3; // Beware : Windows Bitmaps have to have a DWORD aligned width :-(
CPoint AoiPosition(0,0);
CSize AoiSize(ImageSize);
Bcam.FormatSeven[VideoMode].Position = AoiPosition;
Bcam.FormatSeven[VideoMode].Size = AoiSize;

// Set full speed
unsigned long BytePerPacketMax = Bcam.FormatSeven[VideoMode].BytePerPacket.Max();
Bcam.FormatSeven[VideoMode].BytePerPacket = BytePerPacketMax;

// Create image buffer
const unsigned short BytePerPixel = 1;
const unsigned short DataDepth = 8;

```

```

// Allocate Resources and Grab the image
Bcam.AllocateResources(1,AoiSize.cx*AoiSize.cy*BytePerPixel);
Bcam.ContinuousShot = true;
Bcam.GrabImageAsync(shm->pBuffer, AoiSize.cx*AoiSize.cy*BytePerPixel, (void*)shm->pBuffer, false);
    unsigned long ErrorCode;
    FunctionCode_t FunctionCode;
    void *pContext=0;//&(shm->pBuffer);

    shm->RAM [1]=0;
    shm->RAM [2]=0; //exit flag
    shm->RAM [3]=0; //live trigger flag
    shm->RAM [4]=0; //image capture flag

    while (1){
        if (shm->RAM [2]) break;
        if (shm->RAM [3]==0){Sleep(20);continue;}
        if (shm->RAM [4]==0){
            Bcam.WaitForCompletion(&FunctionCode, &ErrorCode, &pContext, 3000);
            if (ErrorCode) break;
            //throw BcamException( ErrorCode );
            if (FunctionCode == AsyncGrabImage)
                Bcam.GrabImageAsync(pContext, AoiSize.cx*AoiSize.cy*BytePerPixel, pContext,
                    false);
            Sleep(10);
            shm->RAM [4]=1;
            Sleep(20);
        }
    }

    // Turn continuous shot off
    Bcam.ContinuousShot = false;
    Bcam.Cancel();

//=====
    close_share_memory();
    ExitProcess(0);
    return 0;
}

```

Appendix B

Image Display Sub-Program and Image Recognition Sub-Program (Programmed by using Microsoft® Visual Basic 6.0)

```
'-----  
'2006:by Bio-Electro-Mechanical System Lab.  
'Department of Mechanical and Electro-Mechanical Engineering,  
'National Sun Yat-sen University,  
'70 Lien-hai Rd. Kaohsiung 804, Taiwan  
'-----  
  
Option Explicit  
Dim MyBuffer As Long  
Dim geCurrentHandle As enumHandleNames  
Dim run_status As Boolean  
Dim eFileFormat As enumFileFormats  
Dim frame_no As Integer  
Dim particleA_counter As Integer  
Dim particleB_counter As Integer  
Dim optical_sensorA As Boolean  
Dim optical_sensorB As Boolean  
Dim image_threshold As Integer  
Dim counter_threshold As Integer  
Dim time_start As Currency  
Dim TimeIntervalBetweenTwoPosition As Currency  
Dim previous_position As Integer  
Dim IsNewParticle As Boolean  
Dim particle_start_position As Integer  
Dim first_particle_velocity As Double  
Dim SpeedFactor As Single  
Dim total_orgx As Single  
Dim total_ory As Single  
Dim sngCenterX As Single, sngCenterY As Single, sngAngle As Single  
Dim sngScaleX As Single, sngScaleY As Single, sngScore As Single  
Dim save_file_flag As Boolean
```

```

Dim gray_value As Integer
Dim gray_value_maxA As Integer
Dim gray_value_maxB As Integer
Dim gray_value_minA As Integer
Dim gray_value_minB As Integer
Dim intIndex As Integer
Dim laser_spot_positionX As Integer
Dim laser_spot_positionY As Integer
Dim LaserTriggerStartTime As Currency
Dim LaserTriggerOnLastingTime As Currency
Dim LaserTriggerOFFTimeDelayFlag As Boolean
Dim ParticleVelocityWhenLaserTrigger As Integer
Dim scale_factor As Single
Dim code_Xpos As Integer
Dim code_Ypos As Integer
Dim TheMostFrontXposition As Integer
Dim Size_Maximum As Single
Dim LiveCaptureFlag As Boolean
Dim InspectFromVideoFileFlag As Boolean
Dim ResetScaleFromFileFlag As Boolean
Dim ResetScaleFromLiveCaptureFlag As Boolean
Dim ImageScaleRate As Single 'scale rate between live capture image and video file image
Const imagepixel_W = 656
Const imagepixel_H = 492

```

B.1 Program Initial

```

Private Sub Form_Load()
Dim strMsg As String
Dim lngImagePointer As Long
Dim i As Long
Dim PIXPOINT As Long

shm_location = 0
shm_location = LinkMemoryBlock() + 307200 * 0

ImageScaleRate = 1
laser_spot_positionX = 460

```



```

laser_spot_positionY = 240
'Draw counter line position
counter_line_A_X1 = 502
counter_line_A_Y1 = 125
counter_line_A_X2 = 502
counter_line_A_Y2 = 165
counter_line_B_X1 = 502
counter_line_B_Y1 = 175
counter_line_B_X2 = 502
counter_line_B_Y2 = 205
ResetScaleFromFileFlag = False
ResetScaleFromLiveCaptureFlag = True
LiveCaptureInitialSetup

EBW8Image1.Refresh
EBW8Image2.Refresh
' Set the ROI 1 parent image
EBW8ROI1.ParentImage = EBW8Image1.object
EBW8ROI2.ParentImage = EBW8Image2.object

' Set ROI draw color(line)
EBW8ROI1.DrawColor = RGB(0, 0, 255)
' Set ROI fill style/color (handles)
EBW8ROI1.FillColor = RGB(255, 255, 255)
EBW8ROI1.FillStyle = eFSSolid

EBW8ROI2.DrawColor = RGB(0, 0, 255)
' Set ROI fill style/color (handles)
EBW8ROI2.FillColor = RGB(255, 255, 255)
EBW8ROI2.FillStyle = eFSSolid

'Set the ROI 1 coordinates
EBW8ROI1.ROIOrgX = EBW8Image1.ImageWidth / 4
EBW8ROI1.ROIOrgY = EBW8Image1.ImageHeight / 4
EBW8ROI1.ROIWidth = EBW8Image1.ImageWidth / 4
EBW8ROI1.ROIHeight = EBW8Image1.ImageHeight / 4
EBW8ROI1.DrawFrame eFrameOn, True

```

```

EBW8Image1.TextTransparent = True
EBW8Image1.TextColor = RGB(255, 0, 255)

EBW8Image3.TextTransparent = True
EBW8Image3.TextColor = RGB(100, 255, 100)

' Set the ROI 2 coordinates
EBW8ROI2.ROIOrgX = EBW8Image1.ImageWidth / 4
EBW8ROI2.ROIOrgY = EBW8Image1.ImageHeight / 4
EBW8ROI2.ROIWidth = EBW8Image1.ImageWidth / 4
EBW8ROI2.ROIHeight = EBW8Image1.ImageHeight / 4
EBW8ROI2.DrawFrame eFrameOn, True

ECodedImage1.WhiteClass = 1
ECodedImage1.BlackClass = 0
ECodedImage1.Threshold = 83
ECodedImage1.DrawColor = RGB(255, 0, 0)

EImageSequencel.State = eStateActive
EImageSequencel.NumRate = 10
EImageSequencel.DenomRate = 1
' The sequence will hold 512x512 EBW8 images
EImageSequencel.ImageWidth = imagepixel_W
EImageSequencel.ImageHeight = imagepixel_H
EImageSequencel.ImageType = eBW8
save_file_flag = False
'Slider1.value = Read_RAM_Parameter(5)

run_status = False
' Set angle unit to degrees
EasyMain1.AngleUnit = eDegrees

strMsg = "To build an AVI image sequence:" + vbCrLf
strMsg = strMsg + vbTab + "Open images from folder Images\EasyAVI." + vbCrLf
strMsg = strMsg + vbTab + "Select a destination file (*.avi)." + vbCrLf
strMsg = strMsg + vbTab + "Select a codec." + vbCrLf
strMsg = strMsg + vbTab + "The sequence is built automatically." + vbCrLf + vbCrLf
strMsg = strMsg + "To retrieve a sequence from an AVI file:" + vbCrLf

```

```

strMsg = strMsg + vbTab + "Open an AVI file (*.avi)." + vbCrLf
strMsg = strMsg + vbTab + "The sequence is played automatically." + vbCrLf + vbCrLf
strMsg = strMsg + "Required licenses: None"
'MsgBox strMsg, vbInformation

LiveCaptureFlag = False
error_code = 0
LDD10_ComPort = 1
check_COMM_Port
check_LDD10_error
isLaserSoftTrigger = False
LaserTriggerOFFTimeDelayFlag = False
LaserDiodeONTimer = 3 'unit: 0.1 sec; laser output timer base 300 ms.
IsWhiteParticle = True
InspectFromVideoFileFlag = False
MaxMatches = 5
MinScore = 0.9
MinScale = 90
MaxScale = 110
Particle_Size_Max = 50
Particle_Size_Min = 2
SpeedFactor = 1
image_threshold = 100
UpDown1.value = image_threshold
Text2.Text = image_threshold
DisplaySpeedFlag = True
DisplayAreaFlag = False
DisplayPatternMatchFlag = False
Timer_draw_mark.Enabled = True
Open App.Path & "\CounterPeak_Record.txt" For Output As #1
End Sub

Private Sub Timer_draw_mark_Timer()
    EBW8ROI1.DrawFrame eFrameOn, True
    EBW8ROI2.DrawFrame eFrameOn, True
    EBW8Image1.DrawCircle laser_spot_positionX, laser_spot_positionY, 8
    EBW8Image1.DrawLine counter_line_A_X1, counter_line_A_Y1, counter_line_A_X2, counter_line_A_Y2
    EBW8Image3.DrawLine counter_line_A_X1, counter_line_A_Y1, counter_line_A_X2, counter_line_A_Y2

```

```

    EBW8Image1.DrawLine counter_line_B_X1, counter_line_B_Y1, counter_line_B_X2, counter_line_B_Y2
    EBW8Image3.DrawLine counter_line_B_X1, counter_line_B_Y1, counter_line_B_X2, counter_line_B_Y2
    Timer_draw_mark.Enabled = False
End Sub

Private Sub Command1_Click()
'MsgBox EImageSequence1.ImageType
    Dim i As Integer
    Dim j As Integer
    Dim k As Integer
    Dim max_index As Integer
    Dim NumPositions_old As Integer
    Dim moving_distance As Single
    Dim total_orgx As Single

    Dim sngCenterX(10) As Single, sngCenterY(10) As Single, sngAngle(10) As Single
    Dim sngCenterX_old(10) As Single, sngCenterY_old(10) As Single, sngAngle_old(10) As Single, DX(10)
As Single
    Dim sngScaleX(10) As Single, sngScaleY(10) As Single, sngScore(10) As Single

    run_status = True
    moving_distance = 20
    'total_orgx = EC24ROI1.TotalOrgX
    ' Transform x and y from twips to pixels
    'total_orgx = ScaleX(total_orgx, vbTwips, vbPixels)

    EImageSequence1.Position = 0
    EMatch1.MaxPositions = 5
    EMatch1.MinScore = 0.93
    EMatch1.MinScale = 90# / 100
    EMatch1.MaxScale = 110# / 100
    For i = 1 To EImageSequence1.Count
        ' Retrieve the image
        max_index = 0
        'EImageSequence1.GetImage EC24Image1.object
        'EImageSequence1.GetImage EC24ROI1.Object
        ' Process it if needed
        ' ...

```

```

EMatch1.Match EC24ROI1.object
'On Error Resume Next
'Retreive informations about the first match
If EMatch1.NumPositions > 0 Then
For j = 0 To EMatch1.NumPositions - 1

    EMatch1.GetPosition j, sngCenterX(j), sngCenterY(j), sngAngle(j), sngScaleX(j),
sngScaleY(j), sngScore(j)

    'speed caculate
    If i > 1 Then
        For k = 0 To NumPositions_old - 1
            DX(j) = sngCenterX(j) - sngCenterX_old(k)
            If ((DX(j) <= moving_distance) And (DX(j) > 0)) Then Exit For
        Next k
    End If

    NumPositions_old = EMatch1.NumPositions
    sngCenterX_old(j) = sngCenterX(j)
    Label3(j).Caption = sngCenterX(j) + total_orgx
    'Label4(j).Caption = DX(j)
    Label3(j).Refresh
    'Label4(j).Refresh
    If j > 0 Then
        If sngCenterX(j) > sngCenterX(max_index) Then max_index = j
    End If
Next j

For k = NumPositions_old To 9
    Label3(k).Caption = 0
    'Label4(k).Caption = 0
Next k

Label2.Caption = sngCenterX(max_index) + total_orgx
Label2.Refresh
' Draw it
'EC24Imagel.Refresh
EMatch1.DrawStyle = eDot

```

```

    EMatch1.DrawColor = RGB(255, 0, 255)
    'EMatch1.DrawPositions EC24Image1.object

End If
' Sleep (50)
Next i
EC24ROI1.DrawFrame eFrameOn, True
End Sub

Private Sub Command10_Click()
check_LDD10_error
If ((error_code > 0) Or (buf = "")) Then
    'Exit Sub
    isLDD10DriverExist = False
End If
Form_Laser_Control.Show 1
End Sub

Private Sub Command11_Click()
'If Form_Laser_Control.CWButton1.Value = True Then Call Form_Laser_Control.Command4_Click
'Form_Laser_Control.MSComm1.PortOpen = False
End
End Sub

Private Sub Command9_Click()
Dim intIndex As Integer

    ECodedImage1.WhiteClass = 0
    ECodedImage1.BlackClass = 1
    'ECodedImage1.Threshold = 200
    ECodedImage1.DrawColor = RGB(255, 0, 0)
    ECodedImage1.BuildObjects EBW8Image1.object
    ECodedImage1.AnalyseObjects eObjArea
    ECodedImage1.SelectObjectsUsingLongFeature eObjArea, 0, 100, eObjRemoveLesserOrEqual
    'ECodedImage1.BuildObjects EBW8Image3.object
    ' Compute the Area feature
    'ECodedImage1.AnalyseObjects eObjGravityCenterX

```

```

'EcodedImage1.SortObjectsUsingFeature eObjGravityCenterX, eObjSortAscending

'EcodedImage1.GoToFirstObject
'Label14.Caption = EcodedImage1.NumberOfObjects
Label14.Caption = EcodedImage1.NumberOfSelectedObjects

' Loop around the objects list
For intIndex = 0 To EcodedImage1.NumberOfObjects - 1
'
    Label13.Caption = EcodedImage1.GetObjectLongFeature(eObjGravityCenterX)
    ' Go to Get next object of the objects list
    On Error Resume Next
    EcodedImage1.GoToNextObject
Next intIndex
' Compute the average and variance of the area feature of all currently selected objects
'Dim sngAreaAverage As Single, SngAreaVariance As Single
'EcodedImage1.FeatureVariance eObjArea, sngAreaAverage, SngAreaVariance
EcodedImage1.DrawObjects EBW8Image3.object, eObjSelectedTrue

End Sub

Private Sub EC24Image1_Refreshed()
    'EMatch1.DrawPositions EC24Image1.Object
    '    EC24ROI1.DrawFrame eFrameOn, True
End Sub

Private Sub OpenAVI_Click()
    ' Set filters
    CommonDialog1.Filter = "AVI Files (*.avi)|*.avi"
    ' Reset the name
    CommonDialog1.FileName = ""

    ' Display the Open dialog box
    CommonDialog1.ShowOpen

    ' Set file name to load
    If CommonDialog1.FileName = "" Then
        Exit Sub
    End If

```

```

'EImageSequence1.CodecFourcc = (D,I,B, )
EImageSequence1.NumRate = 15
EImageSequence1.DenomRate = 1
'EImageSequence1.KeyFrameRate = 5
EImageSequence1.StartLoading CommonDialog1.FileName

If Not InspectFromVideoFileFlag Then
    ResetScaleFromFileFlag = True
    ResetScaleFromLiveCaptureFlag = False
End If

InspectFromVideoFileFlag = True
FileCaptureInitialSetup

' eVision images are top-bottom while Windows images are bottom-up
EImageSequence1.FlipImage = True

' Adjust image size to the sequence size
EC24Image1.SetSize EImageSequence1.ImageWidth, EImageSequence1.ImageHeight

EC24Image1.ZoomX = 1
EC24Image1.ZoomY = 1
' Redraw Image
'EBW8Image1.Refresh

' Scroll through the sequence
Dim i As Long
Dim j As Integer
' For j = 1 To 5
EImageSequence1.Position = 0
Dim szCodecFourCC As String
Text1.Text = EImageSequence1.CodecFourccString

' Close the AVI file previously opened
'EImageSequence1.CloseFile
EImageSequence1.GetImage EC24Image1.object
EasyMain1.Convert EC24Image1.object, EBW8Image1.object

```



```

    EC24Image1.Refresh
    EBW8Image1.Refresh
    frame_no = 0
    EImageSequencel.Position = frame_no

End Sub

Private Sub OpenImages_Click()
    Dim current As Integer
    Dim i As Integer
    Dim iIndex As Integer
    Dim iStart As Integer
    Dim FileNames$()
    Dim sFname As String
    Dim iEndPath As Integer

    ' Set the common dialog to open multiple images
    With CommonDialog1
        .MaxFileSize = 2048 ' Set as appropriate
        .FileName = ""
        .Filter = "Image Files (*.tif, *.bmp, *.jpg)|*.tif; *.bmp; *.jpg|TIFF Files (*.tif)|*.tif|BITMAP
Files (*.bmp)|*.bmp|JPEG Files (*.jpg)|*.jpg|All Files (*.*)|*.*||"
        .Flags = cd1OFNAllowMultiselect + cd1OFNExplorer
        .ShowOpen
        sFname = .FileName & vbNullChar
    End With

    ' determine if multiple files were selected
    ' null delimiter is not inserted if only 1 file is selected
    iEndPath = 1
    If countDelimiters(sFname, vbNullChar) = 1 Then
        ' Set iStart to the index of the first letter of the first filename
        Do Until (iEndPath = 0)
            iStart = iEndPath + 1
            iEndPath = InStr(iEndPath + 1, sFname, "\")
        Loop

        ReDim Preserve FileNames(0)
    End If
End Sub

```

```

' determine if root directory was selected - preserve the "\"
If countDelimiters(sFname, "\" ) = 1 Then
    FileNames(0) = Mid(sFname, 1, iStart - 1)
Else
    FileNames(0) = Mid(sFname, 1, iStart - 2)
End If

Else
    ' Set FileNames(0) to the common path
    iStart = InStr(1, sFname, vbNullChar) + 1
    ReDim Preserve FileNames(0)
    FileNames(0) = Left(sFname, iStart - 2)
End If

' Set FileNames[1..NbSelectedFiles] to the filenames
iIndex = 1
For i = iStart To Len(sFname)
    If Mid(sFname, i, 1) = vbNullChar Then
        ReDim Preserve FileNames(iIndex)
        FileNames(iIndex) = Mid(sFname, iStart, i - iStart)
        iStart = i + 1
        iIndex = iIndex + 1
    End If
Next i

' Open the destination AVI file and select a codec
CommonDialog2.FileName = ""
CommonDialog2.Filter = "AVI Files (*.avi)|*.avi"
CommonDialog2.ShowOpen

EImageSequence1.StartSaving CommonDialog2.FileName, Me.hWnd

' Load the selected images and add them to the sequence
' eVision images are top-bottom while Windows images are bottom-up
EImageSequence1.FlipImage = True
' Retrieve FourCC code
Dim szCodecFourCC As String
Text1.Text = EImageSequence1.CodecFourccString

```

```

        'Close the AVI file
        EImageSequencel.CloseFile

End Sub

Function countDelimiters(ByVal sFiles As String, ByVal vSearchChar As Variant) As Integer
' Count the number of occurrences of vSearchChar in sFiles
Dim iCtr As Integer
Dim iResult As Integer

    For iCtr = 1 To Len(sFiles)
        If Mid(sFiles, iCtr, 1) = vSearchChar Then iResult = iResult + 1
    Next iCtr

    countDelimiters = iResult
End Function

Private Sub EBW8Image1_MouseMove(button As Integer, shift As Integer, x As Single, y As Single)
    ' Test if an image was created
    If EBW8Image1.Void Then
        Exit Sub
    End If

    If EBW8Image3.Void Then
        Exit Sub
    End If

    'If run_status = True Then Exit Sub

    ' Transform x and y from twips to pixels
    x = ScaleX(x, vbTwips, vbPixels)
    y = ScaleY(y, vbTwips, vbPixels)

    ' Drag the ROI borders (mouse down) or set the cursor (mouse up)
    Label5.Caption = x / scale_factor
    Label6.Caption = y / scale_factor
    'Label7.Caption = geCurrentHandle
    'Label8.Caption = eHandleNone

```

```

'On Error Resume Next
If (button = vbLeftButton) Then
    ' Mouse button down, drag the ROI border
    If (geCurrentHandle <> eHandleNone) Then
        EBW8ROI1.Drag geCurrentHandle, x, y
        EBW8ROI2.Drag geCurrentHandle, x, y
        ' Redraw Image
        EBW8Image1.Refresh
        EBW8Image2.Refresh
        EBW8Image3.Refresh
    End If
End If

' Change the cursor shape
On Error Resume Next
Select Case EBW8ROI1.HitTest(x, y)
    Case eHandleNorth: MousePointer = vbSizeNS
    Case eHandleSouth: MousePointer = vbSizeNS
    Case eHandleWest: MousePointer = vbSizeWE
    Case eHandleEast: MousePointer = vbSizeWE
    Case eHandleNorthEast: MousePointer = vbSizeNESW
    Case eHandleSouthWest: MousePointer = vbSizeNESW
    Case eHandleNorthWest: MousePointer = vbSizeNWSE
    Case eHandleSouthEast: MousePointer = vbSizeNWSE
    Case eHandleInside: MousePointer = vbSizeAll
    Case eHandleNone: MousePointer = vbArrow
End Select

Select Case EBW8ROI2.HitTest(x, y)
    Case eHandleNorth: MousePointer = vbSizeNS
    Case eHandleSouth: MousePointer = vbSizeNS
    Case eHandleWest: MousePointer = vbSizeWE
    Case eHandleEast: MousePointer = vbSizeWE
    Case eHandleNorthEast: MousePointer = vbSizeNESW
    Case eHandleSouthWest: MousePointer = vbSizeNESW

```

```

    Case eHandleNorthWest: MousePointer = vbSizeNWSE
    Case eHandleSouthEast: MousePointer = vbSizeNWSE
    Case eHandleInside: MousePointer = vbSizeAll
    Case eHandleNone: MousePointer = vbArrow
End Select

```

```

EBW8ROI1.DrawFrame eFrameOn, True
EBW8ROI2.DrawFrame eFrameOn, True
End Sub

```

```

Private Sub EBW8Image1_MouseDown(button As Integer, shift As Integer, x As Single, y As Single)

```

```

    ' Switch from the up to the down button position
    ' Test if an image was created
    'run_status = False
    If EBW8Image1.Void Then
        Exit Sub
    End If

```

```

    If EBW8Image3.Void Then
        Exit Sub
    End If

```

```

    ' Transform x and y from twips to pixels
    x = ScaleX(x, vbTwips, vbPixels)
    y = ScaleY(y, vbTwips, vbPixels)

```

```

    ' Get current handle under mouse cursor
    geCurrentHandle = EBW8ROI1.HitTest(x, y)

```

```

End Sub

```

```

Private Sub EBW8Image1_MouseUp(button As Integer, shift As Integer, x As Single, y As Single)

```

```

    'run_status = True
End Sub

```

B.2 Image Display and Main Recognition Function

```

Private Sub Timer_live_capture_Timer()

```

```

If InspectFromVideoFileFlag Then

EImageSequence1.GetImage EC24Image1.object
EasyMain1.Convert EC24Image1.object, EBW8Image1.object
'EC24ROI1.DrawFrame eFrameOn, True
  If frame_no = EImageSequence1.Count Then
    Timer_live_capture.Enabled = False
    'EImageSequence1.CloseFile
    'InspectFromVideoFileFlag = False
    LiveCaptureFlag = False
    Command16.BackColor = &HFF8080
  End If

  frame_no = frame_no + 1
  ' Close the AVI file previously opened

EC24Image1.Refresh
End If

On Error Resume Next
total_orgx = EBW8ROI1.TotalOrgX
total_ory = EBW8ROI1.TotalOrgY

' Transform x and y from twips to pixels
'total_orgx = ScaleX(total_orgx, vbTwips, vbPixels)
'total_ory = ScaleY(total_ory, vbTwips, vbPixels)

'Call Write_RAM_Parameter(1, 1)
If (Read_RAM_Parameter(4)) Then
  On Error Resume Next
  If Not InspectFromVideoFileFlag Then Call Write_RAM_Parameter(4, 0)

  'Pattern Match function
  '=====
  If DisplayPatternMatchFlag Then EMatch1.Match EBW8ROI1.object

```

```

If EMatch1.NumPositions > 0 Then
    EMatch1.GetPosition 0, sngCenterX, sngCenterY, sngAngle, sngScaleX, sngScaleY, sngScore
    Label2.Caption = sngCenterX + total_orgx
    'Label21.Caption = sngCenterY + total_orgy
    Label2.Refresh
    'Label21.Refresh

End If

```

```

'=====

```

```

EasyMain1.ThresholdBW8 EBW8Image1.object, EBW8Image2.object, image_threshold, 0, 255
EasyMain1.ThresholdBW8 EBW8Image1.object, EBW8Image3.object, image_threshold, 0, 255

```

```

If Option1(0).value = True Then EasyMain1.Laplacian4 EBW8Image2.object, EBW8Image2.object
If Option1(1).value = True Then EasyMain1.Gradient EBW8Image2.object, EBW8Image2.object
If Option1(2).value = True Then EasyMain1.Sobel EBW8Image2.object, EBW8Image2.object
If Option1(3).value = True Then EasyMain1.Roberts EBW8Image2.object, EBW8Image2.object
If Option1(4).value = True Then EasyMain1.Highpass2 EBW8Image2.object, EBW8Image2.object
EasyMain1.DilateDisk EBW8Image2.object, EBW8Image2.object, 1

```

```

ECodedImage1.Threshold = image_threshold '100
'ECodedImage1.BuildObjects EBW8Image3.object
ECodedImage1.BuildObjects EBW8ROI2.object
ECodedImage1.AnalyseObjects eObjArea
ECodedImage1.FeatureMaximum eObjArea, Size_Maximum

```

```

'ECodedImage1.AnalyseObjects eObjCentroidX
ECodedImage1.AnalyseObjects eObjGravityCenterX
ECodedImage1.AnalyseObjects eObjGravityCenterY
'ECodedImage1.GoToFirstObject
'ECodedImage1.GoToObjectByIndex 0
'For intIndex = 0 To ECodedImage1.NumberOfObjects - 1
    'Label113.Caption = ECodedImage1.GetObjectLongFeature(eObjArea)
    'Label113.Caption = ECodedImage1.GetObjectSingleFeature(eObjGravityCenterX)
    'Label113.Refresh
    'ECodedImage1.GoToNextObject
'Next intIndex
ECodedImage1.SelectObjectsUsingLongFeature eObjArea, Particle_Size_Min, Particle_Size_Max,

```

eObjRemoveOutOfRange

```
'ECodedImage1.SelectObjectsUsingLongFeature eObjArea, 100, 1100, eObjRemoveOutOfRange  
'ECodedImage1.SelectObjectsUsingLongFeature eObjArea, 1000, 2000, eObjRemoveOutOfRange  
'ECodedImage1.DrawObjects EBW8Image2.object, eObjSelectedTrue
```

```
'ECodedImage1.BuildObjects EBW8Image3.object  
'Label114.Caption = ECodedImage1.NumberOfObjects  
Label114.Caption = ECodedImage1.NumberOfSelectedObjects  
'Label124.Caption = Size_Maximum
```

```
'if there exist suitable size of particles  
'Compute the Area feature
```

```
If ECodedImage1.NumberOfSelectedObjects > 0 Then
```

```
    ECodedImage1.SortObjectsUsingFeature eObjGravityCenterX, eObjSortDescending  
    ECodedImage1.GoToFirstObject
```

```
    ' Loop around the objects list
```

```
    'For intIndex = 0 To ECodedImage1.NumberOfObjects - 1
```

```
    TheMostFrontXposition = Int(ECodedImage1.GetObjectSingleFeature(eObjGravityCenterX))
```

```
    TimeIntervalBetweenTwoPosition = (timeGetTime - time_start) * 10
```

```
    Label113.Caption = TheMostFrontXposition
```

```
    Label116.Caption = TheMostFrontXposition - previous_position
```

```
    Label115.Caption = TimeIntervalBetweenTwoPosition
```

```
    Label115.Refresh
```

```
    'calculate speed of first particle
```

```
    If (TheMostFrontXposition - previous_position) < 0 Then
```

```
        'If IsNewParticle = True Then
```

```
            particle_start_position = TheMostFrontXposition
```

```
            time_start = timeGetTime
```

```
            'IsNewParticle = False
```

```
    End If
```

```
    If ((TimeIntervalBetweenTwoPosition > 0) And (Val(Label116.Caption) >= 0)) Then
```

```
    first_particle_velocity = Int((TheMostFrontXposition - particle_start_position) /
```

```
    TimeIntervalBetweenTwoPosition)
```

```
    If first_particle_velocity < 1000 Then
```



```

        Label17.Caption = first_particle_velocity
    Else
        first_particle_velocity = 0
    End If

previous_position = TheMostFrontXposition

If first_particle_velocity Then SpeedFactor = 300 / first_particle_velocity
If SpeedFactor < 0.5 Then SpeedFactor = 0.5
If SpeedFactor > 5 Then SpeedFactor = 5

'put LDD-10 ON control here
'=====
If (((laser_spot_positionX - TheMostFrontXposition - total_orgx) >= (30 / ImageScaleRate))
And _
((laser_spot_positionX - TheMostFrontXposition - total_orgx) <= (50 / ImageScaleRate)) And
-
(Not (isLaserSoftTrigger))) Then

    LaserTriggerStartTime = timeGetTime 'start laser trigger ON time counting.
    ParticleVelocityWhenLaserTrigger = first_particle_velocity
    isLaserSoftTrigger = True

    'Trigger ON
    If (isLDD10DriverExist And isLaserSoftTrigger) Then
        'LDD10_DIODE_ON
        MSComm1.Output = "ens" & vbCr
    Else
        End If
    End If
End If
'=====

End If

'put LDD-10 OFF control here
'=====

```

```

'start trigger time counting
If isLaserSoftTrigger Then
    LaserTriggerOnLastingTime = (timeGetTime - LaserTriggerStartTime) * 100 'unit: 0.1 sec
    If ((LaserTriggerOnLastingTime >= (LaserDiodeONTimer - 2) * SpeedFactor) And (Not
LaserTriggerOFFTimeDelayFlag)) Then 'How many sec for Laser Trigger On.
        'Trigger OFF
        'LDD10_DIODE_OFF
        MSCCom1.Output = "ens" & vbCr
        LaserTriggerOFFTimeDelayFlag = True 'switch ON laser off time delay flag
    End If

    If (LaserTriggerOnLastingTime >= (LaserDiodeONTimer * SpeedFactor)) Then 'Trigger ON
time plus 0.2 sec: timing for switch off the soft trigger flag.
        isLaserSoftTrigger = False
        LaserTriggerOFFTimeDelayFlag = False 'switch OFF laser off time delay flag
    End If

End If

'Label21.Caption = isLaserSoftTrigger

If isLaserSoftTrigger Then
    Shape1.FillColor = vbRed
Else
    Shape1.FillColor = &H808080
End If

Label22.Caption = LaserTriggerOnLastingTime
'=====

EasyMain1.ImageToLineSegment EBW8Image2.object, EBW8Vector1.object, counter_line_A_X1,
counter_line_A_Y1, counter_line_A_X2, counter_line_A_Y2
'EasyMain1.ImageToLineSegment EBW8Image1.object, EBW8Vector1.object, counter_line_A_X1,
counter_line_A_Y1, counter_line_A_X2, counter_line_A_Y2

image_threshold = Val(Text2.Text)
counter_threshold = Val(Text3.Text)

If IsWhiteParticle Then 'white particle

```

```

        WhiteParticleCounterCalculation
    Else 'black particle
        BlackParticleCounterCalculation
    End If

Label10.Caption = particleA_counter
Label4.Caption = particleB_counter

EBW8Vector1.Refresh
EBW8Vector2.Refresh
' Redraw Image
EBW8Image1.Refresh
EBW8Image2.Refresh
EBW8Image3.Refresh

'scale_factor = EBW8Image2.ZoomX

'EasyMain1.ImageToLineSegment EBW8ROI1.Object, EBW8Vector1.Object, 0, 41, 0, 41
EasyMain1.Histogram EBW8ROI1.object, EBWHistogramVector1.object
EBWHistogramVector1.Refresh

If (save_file_flag) Then EImageSequence1.AddImage (EBW8Image1.object)

End If

End Sub

```

B.3 White Particle Software Counter Sub-Function

```

Private Function WhiteParticleCounterCalculation()
Dim i As Integer

gray_value_maxA = 0
gray_value_maxB = 0
'gray_value_minA = 255
'gray_value_minB = 255
    For i = 0 To 13

```

```

EBW8Vector1.GetAt Lin1_Average - 7 + i, gray_value

If gray_value > gray_value_maxA Then gray_value_maxA = gray_value
Next i

EasyMain1.ImageToLineSegment EBW8Image2.object, EBW8Vector2.object, counter_line_B_X1,
counter_line_B_Y1, counter_line_B_X2, counter_line_B_Y2
'EasyMain1.ImageToLineSegment EBW8Image1.object, EBW8Vector2.object, counter_line_B_X1,
counter_line_B_Y1, counter_line_B_X2, counter_line_B_Y2

For i = 0 To 13
EBW8Vector2.GetAt Lin2_Average - 7 + i, gray_value
If gray_value > gray_value_maxB Then gray_value_maxB = gray_value
Next i

Label9.Caption = "MaxA: " & gray_value_maxA
Label19.Caption = "MaxB: " & gray_value_maxB

'Print #1, "======"
'Print #1, "gray_value_maxA="; gray_value_maxA
'Print #1, "gray_value_maxB="; gray_value_maxB
'Print #1, "Time="; Now 'timeGetTime

If (optical_sensorA = True) And (gray_value_maxA <= counter_threshold) Then particleA_counter
= particleA_counter + 1
If (optical_sensorB = True) And (gray_value_maxB <= counter_threshold) Then particleB_counter
= particleB_counter + 1

If gray_value_maxA > counter_threshold Then
optical_sensorA = True
Else
optical_sensorA = False
End If

If gray_value_maxB > counter_threshold Then
optical_sensorB = True
Else

```

```

        optical_sensorB = False
    End If
End Function

```

B.4 Black Particle Software Counter Sub-Function

```

Private Function BlackParticleCounterCalculation()
Dim i As Integer

'gray_value_maxA = 0
'gray_value_maxB = 0
gray_value_minA = 255
gray_value_minB = 255

'Label20.Caption = Lin1_Average
    For i = 0 To 13
        EBW8Vector1.GetAt Lin1_Average - 7 + i, gray_value
        If gray_value < gray_value_minA Then gray_value_minA = gray_value
    Next i

    EasyMain1.ImageToLineSegment EBW8Image2.object, EBW8Vector2.object, counter_line_B_X1,
counter_line_B_Y1, counter_line_B_X2, counter_line_B_Y2
    'EasyMain1.ImageToLineSegment EBW8Image1.object, EBW8Vector2.object, counter_line_B_X1,
counter_line_B_Y1, counter_line_B_X2, counter_line_B_Y2

    For i = 0 To 13
        EBW8Vector2.GetAt Lin2_Average - 7 + i, gray_value
        If gray_value < gray_value_minB Then gray_value_minB = gray_value
    Next i

    Label19.Caption = "MinA: " & gray_value_minA
    Label19.Caption = "MinA: " & gray_value_minB

'Print #1, "=====
'Print #1, "gray_value_maxA="; gray_value_maxA
'Print #1, "gray_value_maxB="; gray_value_maxB
'Print #1, "Time="; timeGetTime

```

```

        If (optical_sensorA = True) And (gray_value_minA >= counter_threshold) Then particleA_counter
= particleA_counter + 1
        If (optical_sensorB = True) And (gray_value_minB >= counter_threshold) Then particleB_counter
= particleB_counter + 1

    If gray_value_minA < counter_threshold Then
        optical_sensorA = True
    Else
        optical_sensorA = False
    End If

    If gray_value_minB < counter_threshold Then
        optical_sensorB = True
    Else
        optical_sensorB = False
    End If
End Function

```

B.5 Laser-Output Shutter Control Sub-Function

```

Private Function LDD10_DIODE_ON()
Dim CLStr As String
'check if Laser Diod ON
'CheckIfLaserDiodON

'If Not isLaserDiodON Then 'ON
'    MSComm1.Output = "S" & vbCr
'    Sleep (70)
'    CLStr = MSComm1.Input
'End If

'control pulse ON instead of laser diode ON for temporary
CheckIfLaserDiodON

If Not isLaserDiodON Then 'ON
    MSComm1.Output = "F" & vbCr
    'Sleep (70)
    CLStr = MSComm1.Input

```

```
End If
```

```
'Shapel.FillColor = RGB(255, 0, 0)
```

```
End Function
```

B.6 Laser-Closed Shutter Control Sub-Function

```
Private Function LDD10_DIODE_OFF()
```

```
Dim CLStr As String
```

```
'If (isLDD10DriverExist) Then
```

```
    'check if Laser Diod ON
```

```
'    CheckIfLaserDiodON
```

```
'    If isLaserDiodON Then 'OFF
```

```
'        MSComml.Output = "S" & vbCr
```

```
'        Sleep (70)
```

```
'        CLStr = MSComml.Input
```

```
'    End If
```

```
'End If
```

```
'control pulse OFF instead of laser diode OFF for temporary
```

```
If (isLDD10DriverExist) Then
```

```
    CheckIfLaserDiodON
```

```
    If isLaserDiodON Then 'OFF
```

```
        MSComml.Output = "F" & vbCr
```

```
        'Sleep (70)
```

```
        CLStr = MSComml.Input
```

```
    End If
```

```
End If
```

```
'Shapel.FillColor = &H808080
```

```
End Function
```

B.7 RS 232 CommPort Available Check Sub-Function

```
Public Function check_COMM_Port()
```

```

'check LDD-10 exist and comm port available
'=====
On Error Resume Next
MSComm1.PortOpen = False
MSComm1.CommPort = LDD10_ComPort
MSComm1.PortOpen = True

If Err.Number <> 0 Then
'If Form_Main.MSComm1.PortOpen = True Then
    MsgBox "Comm Port not available!", vbExclamation + vbOKOnly, "SYSTEM MESSAGE"
    'Command10.Enabled = False
    isLDD10DriverExist = False
End If
'=====
End Function

```

B.8 Miscellaneous

```

Private Function CheckIfLaserDiodON()
Dim StrPos As Long
'MSComm1.Output = "s" & vbCr
'Sleep (50)
'buf = Trim(Form_Main.MSComm1.Input)
'StrPos = InStr(1, buf, "s")
'isLaserDiodON = Mid(buf, StrPos + 1, 1)

'use pulse for temporary
MSComm1.Output = "f" & vbCr
Sleep (50)
buf = Trim(Form_Main.MSComm1.Input)
StrPos = InStr(1, buf, "f")
isLaserDiodON = Mid(buf, StrPos + 1, 1)
End Function

```

```

Private Sub EBW8Image1_Refreshed()
Dim area As Long

```


'Will be called each time the image is redrawn

' If an error occurs continue execution

' See the On Error statement for more information about error handling

On Error Resume Next

' Test if an image was created

If EBW8Image1.Void Then

Exit Sub

End If

' Draw the selected (ECodedImage) objects, using the blue pen

EBW8Image1.DrawCircle laser_spot_positionX, laser_spot_positionY, 8 / ImageScaleRate

EBW8Image1.DrawLine counter_line_A_X1, counter_line_A_Y1, counter_line_A_X2, counter_line_A_Y2

EBW8Image1.DrawLine counter_line_B_X1, counter_line_B_Y1, counter_line_B_X2, counter_line_B_Y2

'EBW8Image1.DrawText (sngCenterX + total_orgx - (12 / ImageScaleRate)), (sngCenterY + total_ory - (40 / ImageScaleRate)), sngCenterX + total_orgx 'DX(max_index)

'Pattern Match function

'=====

If DisplayPatternMatchFlag Then EMatch1.DrawPositions EBW8Image1.object

ECodedImage1.GoToFirstObject

If DisplaySpeedFlag Then

EBW8Image1.DrawText 5, 20, "Position and Speed Mode"

For intIndex = 0 To ECodedImage1.NumberOfSelectedObjects - 1

code_Xpos = ECodedImage1.GetObjectSingleFeature(eObjGravityCenterX)

code_Ypos = ECodedImage1.GetObjectSingleFeature(eObjGravityCenterY)

If intIndex = 0 Then

'EBW8Image1.DrawText (code_Xpos + total_orgx - (12 / ImageScaleRate)), (code_Ypos + total_ory + (20 / ImageScaleRate)), "P=" & code_Xpos + total_orgx & " ;V=" & Int(first_particle_velocity) 'DX(max_index)

'EBW8Image1.DrawText (code_Xpos + total_orgx - (12 / ImageScaleRate) - 45), (code_Ypos + total_ory + (20 / ImageScaleRate) - 5 + 15), code_Xpos + total_orgx & "(" & Int(first_particle_velocity) & ")" 'DX(max_index)

```

        EBW8Image1.DrawText (code_Xpos + total_orgx - (12 / ImageScaleRate)), (code_Ypos + total_orgy
+ (20 / ImageScaleRate) - 5), code_Xpos + total_orgx & "(" & Int(first_particle_velocity) & ")"
'DX(max_index)

```

```

Else

```

```

        'EBW8Image1.DrawText (code_Xpos + total_orgx - (12 / ImageScaleRate)), (code_Ypos +
total_orgy + (20 / ImageScaleRate)), "P=" & code_Xpos + total_orgx 'DX(max_index)

```

```

        'EBW8Image1.DrawText (code_Xpos + total_orgx - (12 / ImageScaleRate) - 45), (code_Ypos +
total_orgy + (20 / ImageScaleRate) - 5 + 15), code_Xpos + total_orgx 'DX(max_index)

```

```

        EBW8Image1.DrawText (code_Xpos + total_orgx - (12 / ImageScaleRate)), (code_Ypos + total_orgy
+ (20 / ImageScaleRate) - 5), code_Xpos + total_orgx 'DX(max_index)

```

```

End If

```

```

ECodedImage1.GoToNextObject

```

```

Next intIndex

```

```

End If

```

```

If DisplayAreaFlag Then

```

```

ECodedImage1.DrawObjects EBW8Image1.object, eObjSelectedTrue

```

```

EBW8Image1.DrawText 5, 5, "Area Mode"

```

```

For intIndex = 0 To ECodedImage1.NumberOfSelectedObjects - 1

```

```

    code_Xpos = ECodedImage1.GetObjectSingleFeature(eObjGravityCenterX)

```

```

    code_Ypos = ECodedImage1.GetObjectSingleFeature(eObjGravityCenterY)

```

```

    area = ECodedImage1.GetObjectLongFeature(eObjArea)

```

```

    'If intIndex = 0 Then

```

```

        ' EBW8Image1.DrawText (code_Xpos + total_orgx - (12 / ImageScaleRate)), (code_Ypos +
total_orgy + (20 / ImageScaleRate)), "A=" & ECodedImage1.GetObjectIntegerFeature(eObjArea) & " ;V=" &
Int(first_particle_velocity) 'DX(max_index)

```

```

    'Else

```

```

        'EBW8Image1.DrawText (code_Xpos + total_orgx - (12 / ImageScaleRate)), (code_Ypos +
total_orgy + (20 / ImageScaleRate)), "A=" & area 'DX(max_index)

```

```

        EBW8Image1.DrawText (code_Xpos + total_orgx - (12 / ImageScaleRate)), (code_Ypos + total_orgy
+ (20 / ImageScaleRate) - 5), area 'DX(max_index)

```

```

    'End If

```

```

    ECodedImage1.GoToNextObject

```

```

Next intIndex

```

```

End If

```

```

        ' Draw ROI1 Rectangle with handles
    'EBW8ROI1.DrawFrame eFrameOn, True
End Sub

Private Sub EBW8Image2_Refreshed()
'Will be called each time the image is redrawn

    ' If an error occurs continue execution
    ' See the On Error statement for more information about error handling
    On Error Resume Next

    ' Test if an image was created
    If EBW8Image3.Void Then
        Exit Sub
    End If

    EBW8Image2.DrawCircle laser_spot_positionX, laser_spot_positionY, 8 / ImageScaleRate
    EBW8Image2.DrawLine counter_line_A_X1, counter_line_A_Y1, counter_line_A_X2, counter_line_A_Y2
    EBW8Image2.DrawLine counter_line_B_X1, counter_line_B_Y1, counter_line_B_X2, counter_line_B_Y2
    EBW8ROI2.DrawFrame eFrameOn, True
End Sub

Private Sub EBW8Image3_Refreshed()
'Will be called each time the image is redrawn

    ' If an error occurs continue execution
    ' See the On Error statement for more information about error handling
    On Error Resume Next

    ' Test if an image was created
    If EBW8Image3.Void Then
        Exit Sub
    End If

    ECodedImage1.DrawObjects EBW8Image3.object, eObjSelectedTrue
    ECodedImage1.GoToFirstObject

```

```

For intIndex = 0 To ECodedImage1.NumberOfSelectedObjects - 1
    code_Xpos = ECodedImage1.GetObjectSingleFeature(eObjGravityCenterX)
    code_Ypos = ECodedImage1.GetObjectSingleFeature(eObjGravityCenterY)
    If intIndex = 0 Then
        EBW8Image3.DrawText (code_Xpos + total_orgx - (12 / ImageScaleRate)), (code_Ypos + total_orgy
+ (20 / ImageScaleRate)), "P=" & code_Xpos + total_orgx & " ;V=" & Int(first_particle_velocity)
'DX(max_index)
    Else
        EBW8Image3.DrawText (code_Xpos + total_orgx - (12 / ImageScaleRate)), (code_Ypos + total_orgy
+ (20 / ImageScaleRate)), "P=" & code_Xpos + total_orgx 'DX(max_index)
    End If
    ECodedImage1.GoToNextObject
Next intIndex
End Sub

```

```

Private Sub UpDown1_Change()
Text2.Text = UpDown1.value
End Sub

```

```

Private Sub UpDown2_Change()
Text3.Text = UpDown2.value
End Sub

```

```

Public Function check_LDD10_error()
Dim StrPos As Long

'check_COMM_Port
On Error Resume Next
MSComm1.Output = "E" & vbCr
Sleep (50)
buf = Trim(MSComm1.Input)
StrPos = InStr(1, buf, "E")
error_code = Val(Mid(buf, StrPos + 1, 1))

```

```

If (buf = "") Then
    MsgBox "LDD10 didn't response! Please check all cables are connected!"
    isLDD10DriverExist = False
    'Unload Form_Laser_Control

```

```

ElseIf (error_code > 0) Then
    MsgBox "LDD10 error! All functions will be closed! error code= " & error_code
    'Unload Form_Laser_Control
    isLDD10DriverExist = False
ElseIf (error_code = 0) Then
    'MsgBox "LDD10 is ready! All functions are available!"
    isLDD10DriverExist = True
End If

```

```

End Function

```

```

Private Sub Command12_Click()

```

```

    ' Set filters
    CommonDialog1.Filter = "AVI Files (*.avi)|*.avi"

    ' Reset the name
    CommonDialog1.FileName = ""

    ' Display the Open dialog box
    CommonDialog1.ShowOpen

    ' Set file name to load
    If CommonDialog1.FileName = "" Then
        Exit Sub
    End If
    EImageSequence1.StartLoading CommonDialog1.FileName

    ' eVision images are top-bottom while Windows images are bottom-up
    EImageSequence1.FlipImage = True

    ' Adjust image size to the sequence size
    EBW8Image1.SetSize EImageSequence1.ImageWidth, EImageSequence1.ImageHeight

    ' Scroll through the sequence
    Dim i As Long
    Dim j As Integer
    For j = 1 To 2

```

```

EImageSequencel.Position = 0
  For i = 1 To EImageSequencel.Count
    ' Retrieve the image
    EImageSequencel.GetImage EBW8Image1.object
    EBW8Image1.Refresh
  Next i
Next j

' Retrieve FourCC code
Dim szCodecFourCC As String
Text1.Text = EImageSequencel.CodecFourccString

' Close the AVI file previously opened
EImageSequencel.CloseFile
End Sub

Private Sub Command13_Click()
save_file_flag = False
  ' Close the AVI file
EImageSequencel.CloseFile
If InspectFromVideoFileFlag Then
  ResetScaleFromFileFlag = False
  ResetScaleFromLiveCaptureFlag = True
End If
InspectFromVideoFileFlag = False
End Sub

Private Sub Command14_Click()
save_file_flag = True
  ' The sequence will hold 512x512 EBW8 images
On Error Resume Next
  EImageSequencel.ImageWidth = imagepixel_W
  EImageSequencel.ImageHeight = imagepixel_H
  EImageSequencel.ImageType = eBW8

' Open the destination AVI file and select a codec
CommonDialog2.FileName = ""
CommonDialog2.Filter = "AVI Files (*.avi)|*.avi"

```

```

CommonDialog2.ShowOpen

EImageSequence1.StartSaving CommonDialog2.FileName, Me.hWnd
' Open the destination AVI file and select a codec with the dialog box
'EImageSequence1.StartSaving "c:\AVI\Sequence.avi", Form1.hWnd

' Load the selected images and add them to the sequence
' eVision images are top-bottom while Windows images are bottom-up
EImageSequence1.FlipImage = True
End Sub

Private Sub Command15_Click()
' Perform learning
'EC24ROI1.ParentImage = EC24Image1.Object
EMatch1.MaxPositions = 15
EMatch1.MinScore = 0.8
EMatch1.MinScale = 100# / 100
EMatch1.MaxScale = 100# / 100
EMatch1.DrawStyle = eDot
EMatch1.DrawColor = RGB(255, 0, 255)

EMatch1.LearnPattern EBW8ROI1.object

' Perform pattern matching
EMatch1.Match EBW8Image1.object
'EC24ROI1.DrawFrame eFrameOn, True
' Redraw Image
EBW8Image1.Refresh

EMatch1.DrawPositions EBW8Image1.object
End Sub

Private Sub Command16_Click()
Call Write_RAM_Parameter(3, 0)
frame_no = 0
EImageSequence1.Position = frame_no

```

```

If LiveCaptureFlag Then 'switch OFF live capture
    'LDD10_DIODE_OFF
    LiveCaptureFlag = False
    Timer_live_capture.Enabled = False

    Command16.BackColor = &HFF8080

Else    'switch ON live capture

    Command16.BackColor = &HFF&
    time_start = timeGetTime
    LiveCaptureFlag = True

    If Not InspectFromVideoFileFlag Then
        LiveCaptureInitialSetup
        Call Write_RAM_Parameter(3, 1)
    End If
    Timer_live_capture.Enabled = True

    On Error Resume Next
    total_orgx = EBW8ROI1.TotalOrgX
    total_ory = EBW8ROI1.TotalOrgY

    EasyMain1.Histogram EBW8ROI1.object, EBWHistogramVector1.object
    EBW8Vector1.DrawColor = RGB(0, 0, 255)
    EBW8Vector1.DrawStyle = eSolid
    EBW8Vector1.SetYRange 0, 250
    EBW8Vector2.SetYRange 0, 250
    particleA_counter = 0
    particleB_counter = 0
    optical_sensorA = False
    optical_sensorB = False
    IsNewParticle = False
    previous_position = 99

    If IsWhiteParticle Then
        ECodedImage1.WhiteClass = 1
        ECodedImage1.BlackClass = 0

```



```

Else
    ECodedImage1.WhiteClass = 0
    ECodedImage1.BlackClass = 1
End If

ECodedImage1.Threshold = image_threshold '83
ECodedImage1.DrawColor = RGB(255, 0, 0)

End If
End Sub

Private Sub Command17_Click()
    Close #1
    Call Write_RAM_Parameter(2, 1)
End
End Sub

Private Sub Command19_Click()
    Form_Parameter_Setting.Show 1
End Sub

Private Sub Command4_Click()
    particleA_counter = 0
    particleB_counter = 0
    Label4.Caption = 0
    Label10.Caption = 0
End Sub

Private Sub Command5_Click()
    'Set the ROI 1 coordinates
    EBW8ROI1.ROIOrgX = EBW8Image1.ImageWidth / 4
    EBW8ROI1.ROIOrgY = EBW8Image1.ImageHeight / 4
    EBW8ROI1.ROIWidth = EBW8Image1.ImageWidth / 4
    EBW8ROI1.ROIHeight = EBW8Image1.ImageHeight / 4
    EBW8ROI1.DrawFrame eFrameOn, True
End Sub

Private Sub Command6_Click()

```

```

    laser_spot_positionX = Val(Text4.Text)
    laser_spot_positionY = Val(Text5.Text)
    EBW8Image1.Refresh
    EBW8Image3.Refresh
End Sub

```

```

Private Function LiveCaptureInitialSetup()
    EBW8Image1.SetSize imagepixel_W, imagepixel_H
    EBW8Image2.SetSize imagepixel_W, imagepixel_H
    EBW8Image3.SetSize imagepixel_W, imagepixel_H
    scale_factor = 1
    EBW8Image1.ZoomX = scale_factor
    EBW8Image1.ZoomY = scale_factor
    EBW8Image2.ZoomX = scale_factor
    EBW8Image2.ZoomY = scale_factor
    EBW8Image3.ZoomX = scale_factor
    EBW8Image3.ZoomY = scale_factor
    EBW8Image1.DrawColor = RGB(255, 0, 0)
    EBW8Image2.DrawColor = RGB(0, 255, 0)
    EBW8Image3.DrawColor = RGB(0, 255, 0)
    If ResetScaleFromLiveCaptureFlag Then ResetImageScale
    MyBuffer = Read_RAM_Parameter(0)
    EBW8Image1.SetImagePointer (MyBuffer)
End Function

```

```

Private Function FileCaptureInitialSetup()
    EBW8Image1.ZoomX = 1.25
    EBW8Image1.ZoomY = 1.25
    EBW8Image2.ZoomX = 1.25
    EBW8Image2.ZoomY = 1.25
    EBW8Image3.ZoomX = 1.25
    EBW8Image3.ZoomY = 1.25
    ImageScaleRate = imagepixel_W / EImageSequence1.ImageWidth
    If ResetScaleFromFileFlag Then ResetImageScale
End Function

```

```

Private Function ResetImageScale()
    If ResetScaleFromFileFlag Then

```

```

laser_spot_positionX = laser_spot_positionX / ImageScaleRate
laser_spot_positionY = laser_spot_positionY / ImageScaleRate
EBW8Image1.DrawCircle laser_spot_positionX, laser_spot_positionY, 8
'Draw counter line position
counter_line_A_X1 = counter_line_A_X1 / ImageScaleRate
counter_line_A_Y1 = counter_line_A_Y1 / ImageScaleRate
counter_line_A_X2 = counter_line_A_X2 / ImageScaleRate
counter_line_A_Y2 = counter_line_A_Y2 / ImageScaleRate
counter_line_B_X1 = counter_line_B_X1 / ImageScaleRate
counter_line_B_Y1 = counter_line_B_Y1 / ImageScaleRate
counter_line_B_X2 = counter_line_B_X2 / ImageScaleRate
counter_line_B_Y2 = counter_line_B_Y2 / ImageScaleRate
ResetScaleFromFileFlag = False
ElseIf ResetScaleFromLiveCaptureFlag Then
    laser_spot_positionX = laser_spot_positionX * ImageScaleRate
    laser_spot_positionY = laser_spot_positionY * ImageScaleRate
    EBW8Image1.DrawCircle laser_spot_positionX, laser_spot_positionY, 16

'Draw counter line position
counter_line_A_X1 = counter_line_A_X1 * ImageScaleRate
counter_line_A_Y1 = counter_line_A_Y1 * ImageScaleRate
counter_line_A_X2 = counter_line_A_X2 * ImageScaleRate
counter_line_A_Y2 = counter_line_A_Y2 * ImageScaleRate
counter_line_B_X1 = counter_line_B_X1 * ImageScaleRate
counter_line_B_Y1 = counter_line_B_Y1 * ImageScaleRate
counter_line_B_X2 = counter_line_B_X2 * ImageScaleRate
counter_line_B_Y2 = counter_line_B_Y2 * ImageScaleRate
ResetScaleFromLiveCaptureFlag = False

End If
Text4.Text = laser_spot_positionX
Text5.Text = laser_spot_positionY

Lin1_Average = (counter_line_A_Y2 - counter_line_A_Y1) / 2
Lin2_Average = (counter_line_B_Y2 - counter_line_B_Y1) / 2
'MsgBox Lin1_Average
End Function

```