

應用圖形處理單元於數位全像顯微術 之雜訊消除運算

陳炯良

真理大學 資訊工程學系 助理教授
clchen@mail.au.edu.tw

蔡秉賢

真理大學 資訊工程學系 研究生

摘要

本文提出運用圖形處理單元(Graphic Processing Unit, GPU)完成於數位全像顯微術(Digital Holographic Microscopy, DHM)消除雜訊的窗口傅立葉轉換(Windowed Fourier transform, WFT)技術。

DHM 系統架構來量測生物細胞的型態，擷取之相位資訊通常會潛藏著許多的雜訊干擾，必須在相位展開前將雜訊濾除，才能夠獲得正確的細胞型態。

WFT 技術能夠有效的對雜訊進行濾除，且參數的判斷對於濾波的效果影響甚大。因此，本文對 WFT 的參數設定進行分析，經由選取之參數來提升濾波的效果。最後使用最小平方相位展開法(Least Square phase unwrapping)的技術重建細胞的三維資訊。

由於 WFT 之技術計算複雜度相當高，本文應用 GPU 平行處理的能力，來提升數值計算的時間，並對提升速度之效果進行評估。經由 GPU 的計算提升，可將運算速度提高至 10 倍以上。

關鍵詞：數位全像顯微術、窗口傅立葉轉換、最小平方相位展開、圖形處理單元

highly perturbed by noise. The noise effect should be reduced before the phase unwrapping to obtain the correct morphology of cells. The Windowed Fourier transform is develop to filtering the noise and the parameter determination is crucial for the de-noise performance. Thus, filtering performance with related to the Windowed Fourier transform parameters is analyzed. The best selection of the Windowed Fourier transform filter parameters is then achieved. Finally, the weakness of the Windowed Fourier transform filter is high computational complexity.

Windowed Fourier transform is used to reduce the noise effect for the phase image of the digital holographic system and apply it to measure the morphology of the living cells. In advance, the Graphic Processing Unit is developed to increase the computation speed to achieve real-time processing.

Thus, we apply the Graphic processing unit to improve the computation speech and the computation efficiency is evaluated. As a result, one can improve the computation speed up to 10 times for most of the cases.

Abstract

The purpose of this paper to implement the Graphic Processing Unit (GPU) is developed to increase the computation speed to achieve real-time processing based on the Digital Holographic Microscopy (DHM) for to reduce the noising effect of the Windowed Fourier transform (WFT).

The Digital Holographic Microscopy system is apply to measure the cell morphology, however, the phase image is

Keywords : Digital Holographic Microscopy, Windowed Fourier transform, Least Square phase unwrapping, Graphic Processing unit.

1. 前言

由於罹癌的人數越來越高，醫學界希望能夠在癌症的治療判斷上得到一個突破，因此藉由觀測癌細胞的特性，進行相關的研究。針對此議題，能夠觀察到精準的細胞形態與結構便成為一個研究重心。

DHM 是利用光學全像的干涉來記錄細胞的完整資訊，並以數值計算的方式加以重建，然而在記錄全像資訊過程中受到雜訊的干擾，以及數值重建過程因反正切($\tan^{-1}$)函數運算，導至無法觀測到正確的細胞資訊。因此本文提出了雜訊去除以及相位展開的演算法來解決此問題，並對雜訊去除演算法中設定的參數加以分析，以便得到更精準的癌細胞資訊。

在面對高複雜度的雜訊去除演算法運算，本文也採取了使用 GPU 將之執行平行處理，能夠在實務操作上更加便利。

1.1 文獻回顧

相位展開分為兩種方向，一是在展開前，運用濾波技術，先將雜訊濾除，如此可使用較簡單的相位展開技術重建相位。另一種則是直接對含有雜訊的相位做展開，故此方向對於雜訊的容忍度需求較高，即為所謂的雜訊免疫型(noise-immune)相位展開技術，而本文架構是針對第一種方向來做思考。

濾波的領域可分為空間濾波(Spatial filtering)和頻域濾波(Frequency filtering)兩種。空間濾波最著名的是 A. Capanni[1]在 1997 年所提出的可適應性中值濾波器，利用局部性的直方圖分佈(local histogram)做濾波處理，避開了相位跳躍(phase jump)的地方平滑化的問題。接著 J. Meneses[2]於 2005 年也提出利用直方圖分佈將影像分割，避開相位跳躍的濾波方式。F. Qian[3]則在 2001 年提出利用調整濾波遮罩大小，來提升濾波的效果。在頻域濾波則多是採用傅立葉濾波及反傅立葉濾波的原理，來保留低頻項。除此之外，還有許多特殊的濾波方式，像 1995 年 Hong[4]提出的最小平方擬合法(least-squares fitting)，在 2000 年 Nico[5]也提出找出不連續濾點的所有組成結構，依結構的組成有其不同的處理。2008 年 J. Bioucas-Dias[6]運用局部多項式近似方式來位移遮罩，並利用 ICI 準則對遮罩適應性的調整。在 2006 年至 2008 年 Q. Kemao[7]提出許多利用窗口傅立葉轉換進行雜訊分離的方法。

圖形處理單元主要是利用硬體結構的特

性，達到平行處理的效果，由於硬體的成本低廉且加速的效果非常顯著，在近幾年的影像處理、信號處理等領域的研究中常被使用。在 2008 年 Tomoyoshi Shimobaba[8]提出了使用圖形處理單元，對數位全像顯微術的全像片做即時的重建，於 2009 年 Wenjing Gao[9]也提出利用圖形處理單元，執行二維的即時窗口傅立葉轉換之研究。

2. 研究方法

本文應用 DHM 系統來觀察細胞的形態，但因系統在全像資訊記錄的過程中容易受到雜訊的干擾，而這些雜訊可能來自數位重建時的訊號雜訊，及光學系統架構所產生的雜訊；在記錄全相資訊過程中所得到的雜訊統稱為散斑雜訊(speckle noise)[10,11]。

如圖 2.1 所示，為本文演算法之流程圖，從 DHM 系統取得重建後的相位資訊，而這個相位資訊是被抑制在 $-\pi \sim \pi$ 之間，且帶有散斑雜訊的包裹相位(Noisy wrapped phase)，首先對此包裹相位進行 WFT 濾波參數之分析，再利用 GPU 配合此參數，進行 WFT，對包裹相位有效的進行雜訊濾除，獲得去除雜訊的包裹相位(De-noisy Wrapped phase)，再使用最小平方相位展開法(Least Square phase unwrapping)將抑制在 $-\pi \sim \pi$ 之間的相位資訊，復原成連續的相位資訊，得到展開後的相位(Unwrapping phase)，可利用此資訊觀察到由 DHM 光學系統所拍攝細胞的二維以及三維之形態。

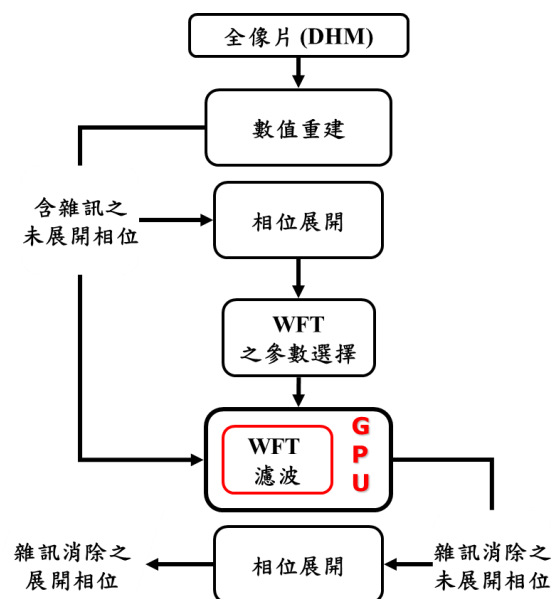


圖 2.1 雜訊去除及相位展開之流程圖

2.1 窗口傅立葉轉換(WFT)[12]

WFT 是傅立葉轉換延伸的一個演算法，讓我們聯想到利用濾波的方式來處理條紋圖或包裹相位圖的 WFT 頻譜，如圖 2.2 所示。

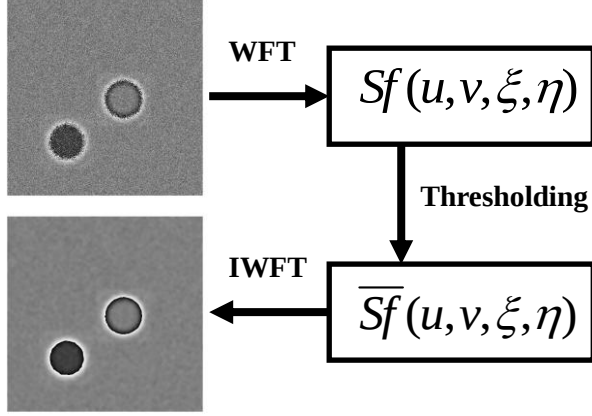


圖 2.2 WFT 之圖解

輸入資訊為含有雜訊的包裹相位，經由 WFT 轉換到頻率域上。一般而言，雜訊彌漫在整個頻率域上，其係數非常小，是由於雜訊在 WFT 的基底函數下是隨機性且鬆散不連續的，因此，可以利用一個閾值(thresholding)，當頻譜係數小於設定的閾值，就可以丟棄其頻譜係數，如此可以達到將雜訊濾除的效果，之後再做 IWFT 產生去除雜訊後的包裹相位。

WFT 之正轉換與反轉換，表示如下兩式：
WFT：

$$\bar{Sf}(u, v, \xi, \eta) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) * g(x-u, y-v) \exp(-j\xi x - j\eta y) dx dy \quad (1)$$

IWFT：

$$f(x, y) = \frac{1}{4\pi^2} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \bar{Sf}(u, v, \xi, \eta) * g(x-u, y-v) \exp(j\xi x + j\eta y) d\xi d\eta du dv \quad (2)$$

其中 $f(x, y)$ 為輸入影像，符號 $*$ 為執行旋積(convolution)， $g(x-u, y-v)$ 為高斯分布之窗口函數， $\bar{Sf}(u, v, \xi, \eta)$ 為經過閾值的 WFT 頻譜。

將之以快速傅立葉轉換(Fast Fourier Transform, FFT)的形式完成，以加快運算時間，表示如下式。

$$\bar{f}(x, y) = \frac{\xi_i \eta_i}{4\pi^2} \times \sum_{\eta_s = \eta_i}^{\eta_h} \sum_{\xi_s = \xi_i}^{\xi_h} \mathfrak{T}^{-1} \left(\mathfrak{T} \left\{ \mathfrak{T}^{-1} \left[\mathfrak{T}(f) \times \mathfrak{T}(h_{\xi_s, \eta_s}) \right] \right\} \times \mathfrak{T}(h_{\xi_s, \eta_s}) \right) \quad (3)$$

其中 $h_{\xi_s, \eta_s}(x, y) = g(x, y) \exp(j\xi_s x + j\eta_s y)$ ， $\mathfrak{T}$ 以及 $\mathfrak{T}^{-1}$ 分別表示為 FFT 之正轉換與反轉換。

2.2 圖形處理單元[14]

隨著 GPU 晶片的演進，如今已能夠做為 CPU 的輔助處理器，也由於硬體的特性，使得許多的應用能夠利用它來執行高速平行運算。

Nvidia 針對 GPU 提出了 CUDA(Compute Unified Device Architecture)的平行運算架構，並以 C 語言做為基礎環境，對於使用者自行開發更為便利。

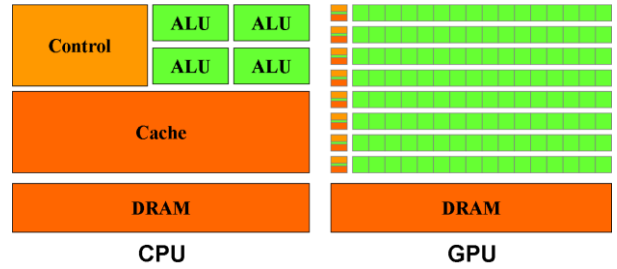


圖 2.3 CPU 與 GPU 架構比較圖[14]

2.2.1 CUDA 程式架構

在 CUDA 的程式架構下分為兩部分，Host 端與 Device 端。Host 端表示為在 CPU 上執行的部分，Device 端則表示在 GPU 上執行。程式運行在 Device 稱之為核心(Kernal)，執行的最小單位為執行緒(Thread)，通常核心會使用 `__global__` 做命名宣告，呼叫 Kernal 時使用 `<<<...>>>` 符號來指派 Threads 數量，即表示需要產生多少的 Threads 將 Kernal 做平行處理。

2.2.2 CUDA 內建常數配置

CUDA 平行結構分為兩層，第一層為執行緒網格(Grid)；一個 Kernel 只會使用到一個 Grid，第二層為執行區塊(Block)；如圖(2.4)所示，每個 Grid 中包含了數個 Block；每個 Block 中亦包含數個 Thread。在執行 Kernel 時，Grid 中各 Block 是平行執行的，且各 Block 之間沒有執行的順序，在 Block 中的 Thread 亦是如此特性。參考表(2.1) 中 Block 與 Thread 各自擁有內建的 ID，我們則使用這些 ID 來鎖定資料的位置，並且利用 ID 鎖定的位置來建立起二維甚至三維的空間，由此可以了解 GPU 優越的平行結構。

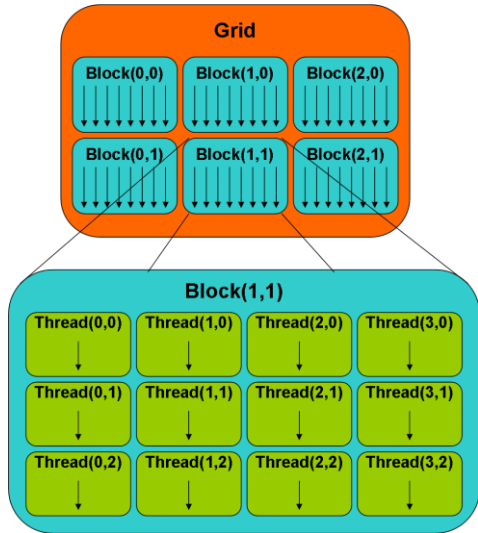


圖 2.4 Grid、Block、Thread 之關係圖[14]

表 2.1 CUDA 內建常數

ID	功能
GridDim	Grid 中包含 Block 數目
BlockDim	Block 中包含 Thread 數目
blockIdx	區塊之 Index
threadIdx	執行緒之 Index

2.3 最小平方相位展開法[13]

本文使用最小平方相位展開法對被抑制的相位做展開，此演算法是 Dennis C. Ghiglia 和 Louis A. Romero 在 1994 年所發表的“Minimum L^p - norm Phase Unwrapping”，針對原始相位存在不連續現象的相位展開問題加以克服。

展開前的包裹相位為 $\psi_{i,j}$ ，展開後的正確相位為 $\phi_{i,j}$ ，其關係如下式：

$$\phi_{i,j} = \psi_{i,j} + 2\pi k_{i,j} \quad (4)$$

其中 $k_{i,j}$ 為整數矩陣；

$$-\pi < \psi_{i,j} < \pi ;$$

$$i=0,1,\dots,M-1 ; j=0,1,\dots,N-1。$$

當連續的相位分佈取樣率足夠且無雜訊造成不連續點時，每一點與其正交四鄰任一點之相位差的絕對值必小於 π ，且已知展開後的正確相位差值 $(\phi_{i+1,j} - \phi_{i,j})$ 和展開前的包裹相位差值 $(\psi_{i+1,j} - \psi_{i,j})$ 控制在 $-\pi \sim \pi$ 的結果必定相同，其展開前相位之相位差 $\Delta_{i,j}^x$ 、 $\Delta_{i,j}^y$ 如下式表示：

$$\Delta_{i,j}^x = \begin{cases} W\{\psi_{i+1,j} - \psi_{i,j}\}, & i=0,\dots,M-2; j=0,\dots,N-1 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

$$\Delta_{i,j}^y = \begin{cases} W\{\psi_{i,j+1} - \psi_{i,j}\}, & i=0,\dots,M-1; j=0,\dots,N-2 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

其中， $W\{\}$ 為 wrap function，藉由加減 2π 整數倍，控制其值在 $-\pi \sim \pi$ 之間。

求解過程為：

$$(\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}) + (\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}) = \rho_{i,j} \quad (7)$$

$$\text{且 } \rho_{i,j} = (\Delta_{i,j}^x - \Delta_{i-1,j}^x) + (\Delta_{i,j}^y - \Delta_{i,j-1}^y) \quad (8)$$

接著採用 Discrete cosine transform (DCT) 收斂速度較快之解法：

Step1. 假設相位圖大小 $M \times N$ ，將公式(8)代入下式：

$$\hat{\rho} = \begin{cases} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} 4\rho_{i,j} \cos\left[\frac{\pi}{2M} m(2i+1)\right] \cos\left[\frac{\pi}{2N} n(2j+1)\right] \\ 0 \end{cases} \quad (9)$$

Step2. 將公式(9)代入下式：

$$\hat{\phi}_{i,j} = \frac{\hat{\rho}_{i,j}}{2\cos\left(\frac{\pi i}{M}\right) + 2\cos\left(\frac{\pi j}{N}\right) - 4} \quad (10)$$

Step3. 將公式(10)代入下式即可求得正確的相位 $\phi_{i,j}$ ：

$$\phi_{i,j} = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} w_1(m) w_2(n) \hat{\phi}_{m,n} \times \cos\left[\frac{\pi}{2M} m(2i+1)\right] \cos\left[\frac{\pi}{2N} n(2j+1)\right] \dots \quad (11)$$

3. 實驗與模擬

根據本文演算法之流程，從 DHM 系統取得重建後的相位資訊為未展開之包裹相位，對此包裹相位進行 WFT 濾波參數判斷，再利用 GPU 配合此參數進行 WFT，最後使用最小平方相位展開法得到展開後的相位資訊，經由展開的相位能夠觀察到由 DHM 系統所拍攝細胞的二維以及三維形態資訊。此章節將此架構以實驗模擬結果表示。

3.1 WFT 參數之分析

實驗根據觀察真實細胞結構特性來建造模擬細胞，以實踐較確切的模擬狀態。

本實驗進行分析的參數設定， ξ_l 、 ξ_h 、 η_l 、 η_h 固定於 $[-2,2] \times [-2,2]$ 的範圍內，濾波閥值設定為背景資訊標準差的三倍，取樣間隔為 $1/\sigma_x$ 及 $1/\sigma_y$ ，於以上設定條件下使用不同梯度且不同數量的模擬細胞，加入不同的雜訊變異數，在不同 σ_x 、 σ_y 之視窗遮罩，進行WFT濾波參數設置之分析。

不同梯度模擬細胞是使用真實取樣細胞之結構特性所建造，如圖 3.1、3.2 所示。在分析實驗中設定為 2、4、6、8、10 五種不同大小的視窗遮罩。

分析結果如圖 3.3、3.4 所示，分別為加入 0.0632 之雜訊變異數，單顆細胞分析之結果曲線以及梯度組合為 0.2、0.05 分析之結果曲線。

比較方式則是以均方差 (Mean Square Error, MSE) 來比較模擬細胞的原始相位與濾波後的相位經由相位展開，比較兩張影像正規化至 $[0 \sim 1]$ 後的差異程度，並以剖面曲線分布的趨勢來決定較佳參數的選取。均方差之公式如下式(12)所示。

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [\phi'_{i,j} - \phi_{i,j}]^2 \quad (12)$$

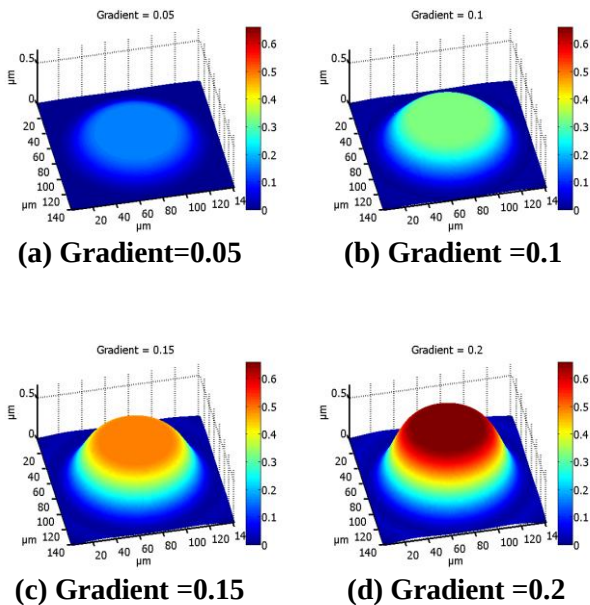


圖 3.1 不同梯度模擬細胞(單顆)

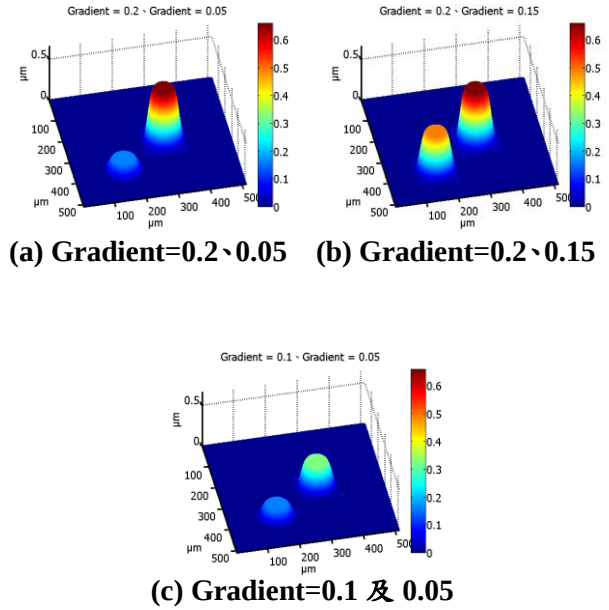


圖 3.2 不同梯度組合之模擬細胞(多顆)

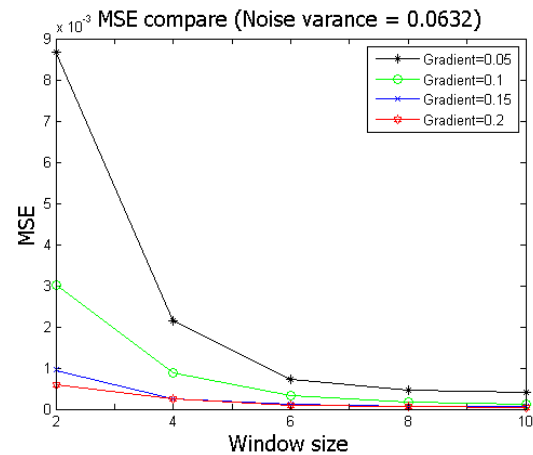


圖 3.3 單顆細胞分析之 MSE 分佈曲線

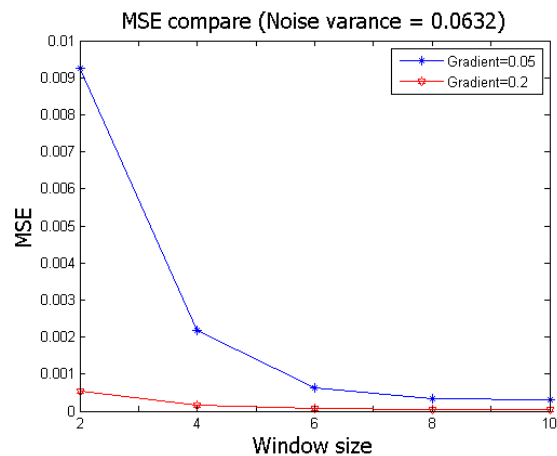


圖 3.4 多顆細胞分析之 MSE 分佈曲線

- ◆ 經由觀察分佈曲線趨勢，可得以下幾點：
- ◆ 梯度較低的細胞，選擇視窗遮罩越大，重建品質越好。
- ◆ 梯度較高的細胞，選擇視窗遮罩不需太大，即可獲得不錯的重建品質。
- ◆ 若同一平面擁有一個以上不同梯度的細胞，需以梯度較低的細胞做參數分析，為整張影像的較佳參數。

依照以上分析以及判斷條件，表 3.1 統整不同梯度區間內較佳視窗遮罩的參數範圍。

表 3.1 較佳視窗遮罩之參數範圍

梯度(Gradient)	σ_x 、 σ_y
0~0.1	10
0.1~0.15	8
0.15~0.2	6
0.2 以上	4

3.2 使用 GPU 執行 WFT

本文使用 GPU 來對高複雜度的 WFT 演算法做平行運算，以提昇濾波的速度。並針對 MATLAB、C、CUDA 三種程式語言，分別對 WFT 演算法程式執行時間做速度比較。C 語言執行 WFT 演算法中的 FFT 是使用 FFTW library[15]，FFTW 為目前最快之 FFT 的函式庫，許多開發軟體的 FFT 皆直接使用或改寫該函式庫，以加快程式執行 FFT 時間；CUDA 執行 FFT 則是使用 CUDA 平行處理中的 CUFFT library[16]，由於經過 CUDA 內建之優化，能夠達到更好的加速效果。

實驗使用的 GPU 為 NVIDIA 公司所推出的 GeForce GTX460 繪圖卡，並將程式運行於 Intel(R) Core(TM) i7 CPU 870 @ 2.93GHz、2.75GB 之記憶體、Microsoft Windows XP 32 位元的作業系統中。開發軟體使用 CUDA 2.3 版、visual studio 2008 Professional 版。

WFT 參數設置 ξ_l 、 ξ_h 、 η_l 、 η_h 固定於 $[-2,2] \times [-2,2]$ 的範圍內，濾波閾值設定為背景資訊之標準差的四倍，取樣間隔為 $1/\sigma_x$ 及 $1/\sigma_y$ ，且依不同的 σ_x 、 σ_y 選擇而定。在此實驗中設定了 2、4、6、8、10 五種不同大小的 σ_x 、 σ_y 。表 3.2 表示使用 512×512 大小的輸入影像，來執行速度之比較。

表 3.2 輸入影像為 512×512 之速度比較

σ_{xy}	MATLAB	C	CUDA	比較 (CUDA/C)
2	7.48	5.26	0.6	8.76
4	28.58	19.18	1.92	9.98
6	103.97	78.2	7.73	10.11
8	148.68	94.04	7.84	11.99
10	253.97	167.06	15.01	11.12

基於實際上之應用，在此針對執行 WFT 參數做調整，以便能夠達到加快處理速度且濾波後的結果差異不大的目的。其分析如表 3.3 所示，利用梯度為 0.2 之模擬細胞放至 512×512 大小平面中，並加入 0.0632(原始細胞全像圖估測之雜訊含量)之雜訊變異數進行分析。將濾波參數 ξ_l 、 ξ_h 、 η_l 、 η_h 調整為 $[-1,1] \times [-1,1]$ ， σ_x 、 σ_y 固定大小為 4，取樣間隔設置為 $1/4$ ，並以 MSE 以及 SNR 比較原始展開相位與濾波後展開的相位兩張影像，正規化至 $[0 \sim 1]$ 後的差異程度。

表 3.3 取樣頻率不同時之速度與品質的比較

ξ_l 、 ξ_h 、 η_l 、 η_h	MSE	SNR	CUDA (sec)
$[-2,2] \times [-2,2]$	0.00012	33.897	1.92
$[-1,1] \times [-1,1]$	0.00037	31.345	0.51

3.3 GPU 用於 DHM 之細胞影像處理

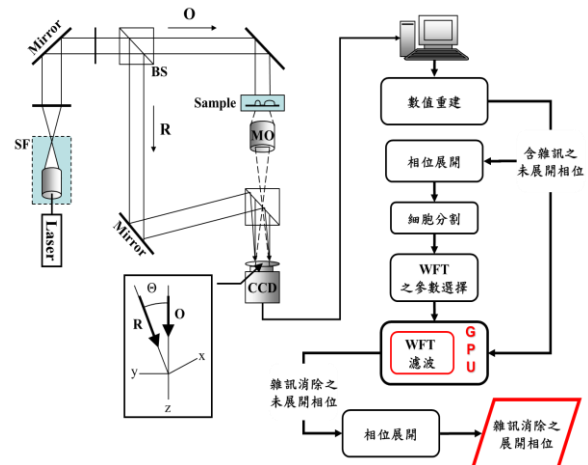


圖 3.5 GPU 用於 DHM 之細胞影像處理架構

根據提出之演算法，套用於真實取樣細胞來觀察展開後的形態。圖 3.5 為實驗所使用的真實細胞取樣 DHM 系統之架構，其中 Laser 為紅光雷射，其波長為 632.8nm；BS 為分光鏡 (Beam Splitter)；MO 為顯微物鏡 (Micro Objective)，利用它來放大樣本資訊；CCD 為電荷耦合裝置 (Charge coupled Device)，能感光並將影像轉成數位訊號的形式記錄下來；最後將 CCD 所記錄的數位全像資訊於電腦進行數值重建以及本文之演算法。

DHM 實驗架構中，由於是連續記錄動態的細胞資訊，所以對於 WFT 之參數選取只需計算第一次，而後的重建相位資訊皆套用此參數執行，接著將選取參數以及重建相位使用 GPU 進行 WFT 濾波，最後做相位展開來觀察細胞的三維形態。

實驗中 WFT 之參數配置 ξ_l 、 ξ_h 、 η_l 、 η_h 固定於 $[-2,2] \times [-2,2]$ 的範圍內，濾波閾值設定為背景資訊標準差的四倍，經由搜尋梯度最小之細胞，選擇較佳的 σ_x 、 σ_y ，可獲得取樣間格 $1/\sigma_x$ 及 $1/\sigma_y$ 。

以下為使用本文的演算法於真實取樣細胞來觀察展開後的形態。其中，標線表示為執行剖面比較的區間。

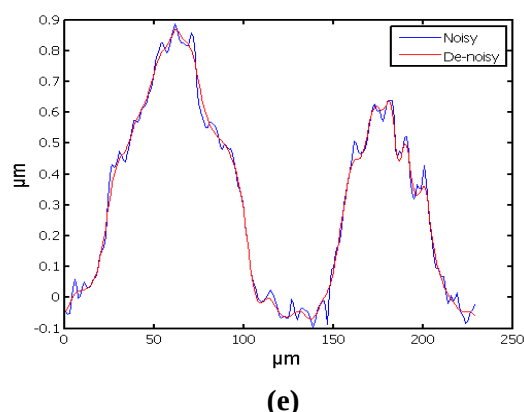
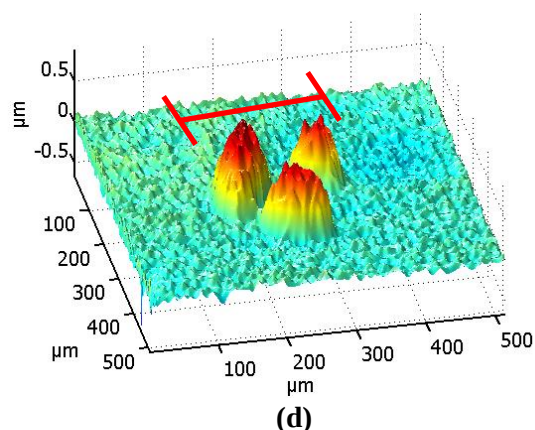
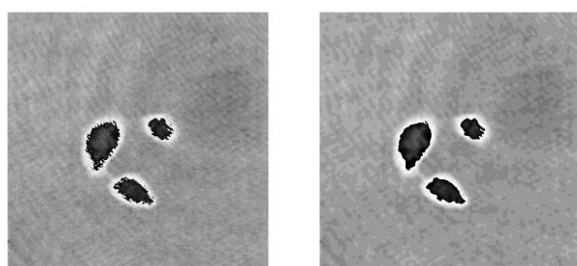
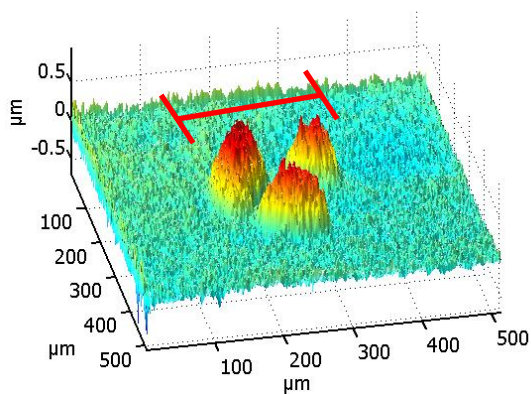


圖 3.6 細胞樣本



(a)

(b)



(c)

- (a) 含雜訊之未展開相位
- (b) 雜訊消除之未展開相位
- (c) 含雜訊之展開相位(3D)
- (d) 雜訊消除之展開相位(3D)
- (e) 剖面比較(含雜訊與雜訊消除之展開相位)

4. 結論與未來展望

本文提出的演算法能夠針對 DHM 架構下所面臨的雜訊干擾以及相位之問題加以解決，且將演算法成功的套用於實際取樣之細胞進行處理，並使用 GPU 硬體進行加速。在雜訊消除部分使用了 WFT，有效的將原始訊號與雜訊分離，且對 WFT 之濾波參數利用梯度最小的細胞資訊進行篩選，獲得品質更好的濾波結果。在面對高複雜度的 WFT 濾波演算法，使用了 GPU 硬體對之執行平行處理，大幅的降低了執行程式的時間。最後使用快速的相位展開演算法將濾波後的相位進行相位展開，能夠清楚的觀測到細胞三維之型態與結構。

本文於 WFT 之濾波參數選擇分析的方向是針對視窗遮罩大小來做篩選，已能夠獲得不錯的濾波效果，而未來希望進一步對更多不同

參數的組合進行分析，以便達到更好的濾波品質。

近年來，由於 GPU 擁有著成本低廉且增速效果明顯的優點，許多的研究皆利用此硬體設備來克服龐大的計算問題。本文也使用了此硬體於高複雜度的濾波演算法進行運算，確實具有不錯的增速效果。而在數位光學全像之領域中，仍有許多執行計算上需耗費大量時間的演算法，如何將之應用於 GPU，藉以縮短其執行時間，達成即時應用的結果，是在未來值得繼續研究的主題。

參考文獻

- [1] A. Capanni, L. Pezzati, D. Bertani, M. Cetica, and F. Francini, "**Phase-shifting speckle interferometry: a noise reduction filter for phase unwrapping**," *Opt. Eng.*, **36**, pp.2466-2472, (1997).
- [2] J. Meneses, T. Gharbi, and P. Humbert, "**Phase-unwrapping algorithm for images with high noise content based on a local histogram**," *Appl. Opt.*, **44**, pp.1207-1215, (2005).
- [3] F. Qian, X. Wang, X. Wang, and Y. Bu, "**Adaptive filter for unwrapping noisy phase image in phase-stepping interferometry**," *Opt. & Laser Tech.*, **33**, pp.479-486, (2001).
- [4] C.K. Hong, H.S. Ryu, and H.C. Lim, "**Least-square fitting of the phase map obtained in phase-shifting electronic speckle pattern interferometry**," *Opt. Lett.*, **20**, pp.931-933, (1995).
- [5] G. Nico, "**Noise-residue filtering of interferometric phase images**," *J. Opt. Soc. Am. A*, **17**, pp.1962-1974, (2000).
- [6] J. Bioucas-Dias, V. Katkovnik, J. Astola, and K. Egiazarian, "**Absolute phase estimation: adaptive local denoising and global unwrapping**," *Appl. Opt.*, **47**, pp.5358-5368, (2008).
- [7] Q. Kemao, H. Wang, and W. Gao, "**Windowed Fourier transform for fringe pattern analysis: theoretical analyses**," *Appl. Opt.*, **47**, pp.5408-5418, (2008).
- [8] Tomoyoshi Shimobaba, Yoshikuni Sato, Junya Miura, Mai Takenouchi, and Tomoyoshi Ito, "**Real-time digital holographic microscopy using the graphic processing unit**," *Opt. Express* **16**, 11776-11781 (2008)
- [9] Wenjing Gao, Nguyen Thi Thanh Huyen, Ho Sy Loi, and Qian Kemao, "**Real-time 2D parallel windowed Fourier transform for fringe pattern analysis using Graphics Processing Unit**," *Opt. Express* **17**, 23147-23152 (2009)
- [10] F. Charrière, J. Kuhn, C. Depeursinge, "**Influence of shot noise on phase measurement accuracy in digital holographic microscopy**," *Appl. Opt.*, **15**, (2007).
- [11] F. Charrière, T. Colomb, C. Depeursinge, "**Shot-noise influence on the reconstructed phase image signal-to-noise ratio in digital holographic microscopy**," *Appl. Opt.*, **45**, pp.7667-7673 (2006).
- [12] Q. Kemao, "**Two-dimensional windowed Fourier transform for fringe pattern analysis: Principles, applications and implementations**," *Opt. and Lasers Eng.*, **45**, pp.304-317, (2007).
- [13] D. C. Ghiglia, L. A. Romero, "**Robust two-dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods**," *Opt. Soc.*, **11**, pp.107-117, (1994).
- [14] CUDA Programming Guide 3.0, http://developer.download.nvidia.com/compute/cuda/3_0/toolkit/docs/NVIDIA_CUDA_ProgrammingGuide.pdf
- [15] FFTW, "FFTW library 3.2.2" <http://www.fftw.org/>
- [16] NVIDIA, "CUDA CUFFT library 2.3" http://www.nvidia.com/object/cuda_develop.html