

高階四元數 Zernike 矩的計算與 RGB 圖還原

鄧安文¹ 郭枝盈^{1*}

¹健行科技大學資訊管理系

摘 要

本文之目標主要在於發展有效、穩定且快速的演算方法計算四元數Zernike矩，並探討在RGB圖上的應用。計算Zernike矩的程序一般而言極為複雜，尤其在計算高階矩之際，如果再對於彩色圖RGB頻道分開計算的處理，所耗掉的時間是原先用Zernike矩處理灰階圖的三倍時間。我們將原先發展計算高階Zernike矩極為有效的方法，結合四元數的代數結構特性，推演出一套快速有效之演算方法。並以影像處理常用的六張512x512大小的RGB圖測試，根據實驗數據顯示，此法在計算高階四元數Zernike矩時，不僅在執行速度上並且在精確度上皆優於他法。

關鍵詞：Zernike矩，四元數Zernike矩，四元數，影像處理，simd向量化

Computation on High-Order Quaternion Zernike Moments and Reconstruction of RGB Images

An-Wen Deng¹ and Chih-Ying Gwo^{1*}

¹Department of Information Management, Chien Hsin University of Science and Technology

ABSTRACT

The objective of this research is to develop an efficient, stable and rapid algorithm to compute Quaternion Zernike moments (QZMs), and to study its application to the RGB images. The computations on Zernike moments are in general very complicated, especially on the high-order moments. When the calculation is performed on each individual RGB channel for the color image, the computational task makes three times of computing on Zernike moments for the gray-level image. In use of our previous method for computing Zernike moments, which is quite efficient, we combine the algebraic properties of Quaternion and derive a series of fast and stable algorithms to compute QZMs. Six most frequently color images of size 512×512 are used to evaluate the performance of proposed method. The experimental numerical results show that the proposed method is outperformed than other methods, not only on the speed but also on the accuracy.

Keywords: Zernike moments, Quaternion Zernike moments, Quaternion, image processing, simd vectorization

一、前言

Zernike 多項式最早是由諾貝爾物理學獎得主、荷蘭光學大師 F. Zernike 所提出來的[1]。而這些 Zernike 多項式具有互相垂直的性質，並於單位圓盤上 L^2 函數空間上形成一組基底。基本上 L^2 函數空間就是 Hilbert 空間[2]，是內積空間，具有相當豐富的數學性質，可以建立廣義的傅立葉級數(Fourier series)理論，其潛在的應用價值，就非常巨大。而 L^2 函數對這些基底的個別投影值所形成的係數，就是後來 Teague 所定義的 Zernike 矩(Zernike moment)[3]。自此，Zernike 應用處理許多面向的影像問題，如影像重建[4, 5]、影像分類、形狀或物件辨識[6]、邊緣偵測[7]、角膜表面建模[8]、人臉辨識[9]、數位浮水印之應用[10]、影像檢索[11]以及掌紋辨識[12]等不同的應用。

Zernike 矩在影像處理的應用面上，還是以處理二元圖(binary image)和灰階圖為主。然而隨著數位化器材的普及化，大部分所取得的影像卻是以彩色為主。雖然彩色影像，可以直接轉換成灰階圖再處理，但原先是 R、G、B 的資訊，經過轉換成灰階的過程中，許多重要的資訊與特徵，都有可能喪失。像是相同形狀，不同顏色的，轉灰階會具有相同灰階值的，如藍色、紅色與綠色，是根本無法辨識當中的不同。

另外一種不同思維，就是將彩色影像，分成 R、G、B 三個頻道，個別處理[13, 14]。相關頻道的 Zernike 矩計算，就個別三個頻道處理。如此，原本二度空間 Zernike 矩的複數計算問題，就必須以六度空間來處理。分開處理計算，就意味著，三個頻道間，原本可能相同的計算，就要分別進行三次，即使是採用平行處理，計算的 CPU 時間，還是以原先只要計算灰階圖的三倍時間起跳。就當中所造成的重複計算而言，這樣解決方案，並非最佳選擇。也有學者企圖採取其他手法，如透過幾何矩(geometric moments)切入[13]，但幾何矩的核心函數(kernel function)並不像 Zernike 多項式具有互相垂直的性質，這就產生資訊冗餘的現象。

最近幾年，原本在代數學上所考慮反對稱域(skew field)的四元數(Quaternion)[15]，被廣泛引進處理彩色圖問題[16]。以具有 RGB

資訊的影像為例，假設 f_B 代表 B 頻道上的強度(intensity)、 f_G 代表 G 頻道上的強度，而 f_R 代表 R 頻道上的強度。此時一個定義域為圖形像素點所成的集合、而對應域為四元數的影像函數 f 就可以定義為

$$f(x, y) = f_B(x, y)i + f_G(x, y)j + f_R(x, y)k \quad (1)$$

而傳統的傅立葉轉換的理論，可以成功地移植成四元數傅立葉轉[17]，原本的 Zernike 矩上的理論與計算方法，也可以移植到四元數 Zernike 矩上(Quaternion Zernike moment; QZM)[18]。然而由 19 世紀愛爾蘭人 William Hamilton 所創的四元數，並不像一般數系中的實數、複數乘法的計算具有交換律，四元數乘法交換律是不成立的。所移植過去的概念，必須區別是左邊還是右邊。

值得注意的是，四元數本身是四度空間的向量空間，也廣泛應用於三度空間的力學問題[19]，特別是在處理球面上的旋轉，有 Rodrigues 旋轉公式。四元數 Zernike 矩的定義，除了要考慮左邊還是右邊，也必須考慮所選取三度空間中的單位向量 μ 。在文獻[18]的四元數 Zernike 矩的理論探討或是實質計算，也都只限於 $\mu = (i+j+k)/\sqrt{3}$ 。一個較為完備的理論，以及系統化的計算方法，都值得更深入探討。透過 Rodrigues 旋轉公式，我們也在文中導出，在不同單位向量 μ 定義下四元數 Zernike 矩之間的轉換關係，藉由此，可定出合理四元數 Zernike 矩的旋轉不變量。我們也證明四元數 Zernike 矩的絕對值是個不變量，其值與不同單位向量 μ 的選取是無關的。

計算 Zernike 矩的方法有許多[20-23]，一般而言，由於複雜的積分公式，計算量非常龐大。採用定義公式直接計算，一方面是計算量龐大，運算緩慢，更不幸的是，由於定義 Zernike 多項式的整數係數，呈現正負互相交錯的現象，其最大絕對值的係數，又隨著階數成長而呈現指數成長。如此，計算誤差累積很快，在一定階數後，階數 45，所求得的 Zernike 矩根本不能採用。一般而言，低階的矩描述輪廓，高階的矩可以描述細節。在沒有採用有效運算之前，取得高階矩相關資訊可能可以帶來的應用，是根本無法產生的。比較有效的作法，

是間接採用 Zernike 矩上的遞迴關係式[20, 21, 23-27]，我們也將原本在計算 Zernike 矩非常成功的一系列演算方法[26, 28-31]，如運用二體群(Dihedral group)群作用的對稱性加速以及利用 Farey 數列法加速[26]，移植到四元數 Zernike 矩上，經實驗數據顯示，在大小 512x512 像素的彩色圖上，計算到非常高階，500 階，所得到的還原圖的品質，都有非常好的表現，當中的誤差都在控制之中。甚至與採 R、G、B 三個頻道分別處理的比較時，實驗數據也顯示有較小的誤差。

採用四元數 Zernike 矩處理彩色圖的優勢，除了前述誤差較小外，還至少其他兩個方面。一個是所需記憶體的量，原來採 R、G、B 三個頻道分別處理，需要六度的實數空間，而採用四元數 Zernike 矩，就是四元數的四度空間。在數學的描述上，也能較為緊致(compact)，這對資訊的存取，也較為有效。

另外一個優勢，就是在執行速度上。可以透過四元數的代數性質，與 R、G、B 的資訊整體結合，這在重複計算的冗餘性，可以有效地降低。由於四元數本身就是向量空間，透過有效向量化的處理，如透過 intel intrinsics 指令[32]，使用 AVX 256-bit 向量暫存器。這在不增加執行緒數目的情況下，執行速度也可以更為快捷。此處所發展的方法，也不僅限於四元數 Zernike 矩計算。

二、四元數 Zernike 矩

2.1 Zernike 多項式與 Zernike 矩

Zernike 多項式在單位圓盤 D 上的複數 L^2 空間中形成一垂直基底。考慮任一複數

$$z = x + iy = re^{i\theta},$$

(x 、 y 為實數， (r, θ) 為其極座標， $i = \sqrt{-1}$ 為純虛數)。Zernike 多項式可分解為 Radial 部份以及指數部份的乘積

$$\begin{aligned} V_{nm}(z) &= R_{nm}(r)e^{im\theta} \\ &= R_{nm}(r)(\cos(m\theta) + i\sin(m\theta)) \end{aligned} \quad (2)$$

其中 $R_{nm}(r)$ 為 Zernike Radial 多項式，階數 n 為非負整數，重複數(repetition) m 為整數滿足 $m \equiv n \pmod{2}$ 且 $|m| \leq n$ 。而 Radial 多項式

$$R_{nm}(r) = \sum_{\substack{k=|m| \\ k \equiv n \pmod{2}}}^n R_{nmk} r^k, \quad (3)$$

當中的各項係數可以二項次係數組合表達如下：

$$R_{nmk} = (-1)^{\frac{n-k}{2}} \frac{\left(\frac{n+k}{2}\right)!}{\left(\frac{n-k}{2}\right)! \left(\frac{k+|m|}{2}\right)! \left(\frac{k-|m|}{2}\right)!} \quad (4)$$

事實上，Zernike 矩 Z_{nm} 就是影像函數 $f(z)$ 與 Zernike 多項式 $V_{nm}(z)$ 的內積。複數版的 Zernike 矩 Z_{nm} 可定義為

$$Z_{nm}(f) = \frac{n+1}{\pi} \iint_D f(z) \bar{V}_{nm}(z) \frac{dz \wedge d\bar{z}}{-2i} \quad (5)$$

當中 $\bar{V}_{nm}(z)$ 為 $V_{nm}(z)$ 的共軛複數，而二階微分型為

$$\frac{dz \wedge d\bar{z}}{-2i} = dx \wedge dy = r dr \wedge d\theta \quad (6)$$

透過 de Moivre 公式，可以導出實數版與複數版 Zernike 矩的轉換公式：

$$\begin{aligned} RZ_{nm}(f) &= \frac{n+1}{\pi} \iint_D f(r, \theta) R_{nm}(r) \cos(m\theta) r dr d\theta \\ &= \frac{Z_{nm}(f) + Z_{n,-m}(f)}{2} \end{aligned} \quad (7)$$

以及

$$\begin{aligned} RZ_{n,-m}(f) &= \frac{n+1}{\pi} \iint_D f(r, \theta) R_{nm}(r) \sin(m\theta) r dr d\theta \\ &= \frac{Z_{n,-m}(f) - Z_{nm}(f)}{2i} \end{aligned} \quad (8)$$

根據一般 L^2 空間的理論，

$$f_M(z) = \sum_{n=0}^M \sum_{\substack{m=-n \\ m \equiv n(2)}}^n Z_{nm} V_{nm}(z) \quad (9)$$

以 L^2 的意義收斂於 $f(z)$ 當 $M \rightarrow \infty$ 。因為此性質， $f_M(z)$ 被用來當成重組圖函數。

2.2 四元數

四元數(Quaternion)為十九世紀愛爾蘭數學家 William Hamilton 所創，在三維空間的力學問題有極佳的應用[19]。四元數基本上是四維的實數向量空間，其正交基底為 $\{1, i, j, k\}$ ，基底向量間滿足以下關係等式

$$\begin{aligned} i^2 = j^2 = k^2 &= -1, \\ ij = -ji = k, \quad jk &= -kj = i, \quad ki = -ik = j, \end{aligned} \quad (10)$$

為一非交換除法環(non-commutative division ring)[15]。任何一個四元數皆是這些正交基底的線性組合：

$$q = q_0 + q_1 i + q_2 j + q_3 k \quad (11)$$

其中 q_0, q_1, q_2 與 q_3 皆為實數。 q 的共軛四元數 $\bar{q}$ 、其絕對值 $|q|$ 與其乘法反元素 q^{-1} (假設 $q \neq 0$) 可表為：

$$\bar{q} = q_0 - q_1 i - q_2 j - q_3 k \quad (12)$$

$$|q| = \sqrt{q\bar{q}} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \quad (13)$$

$$q^{-1} = \frac{\bar{q}}{|q|^2} = \frac{q_0 - q_1 i - q_2 j - q_3 k}{q_0^2 + q_1^2 + q_2^2 + q_3^2} \quad (14)$$

當中 q 的實部為

$$\text{Re}(q) = q_0 \quad (15)$$

虛部為

$$\text{Im}(q) = q_1 i + q_2 j + q_3 k \quad (16)$$

四元數上乘法交換律並不成立，針對於四元數

$x = x_0 + x_1 i + x_2 j + x_3 k$ ， $y = y_0 + y_1 i + y_2 j + y_3 k$ 的乘法 $z = xy$ ，可以矩陣乘法的方式表達：

$$z = \begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} y_0 & -y_1 & -y_2 & -y_3 \\ y_1 & y_0 & y_3 & -y_2 \\ y_2 & -y_3 & y_0 & y_1 \\ y_3 & y_2 & -y_1 & y_0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (17)$$

或者為

$$z = \begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} x_0 & -x_1 & -x_2 & -x_3 \\ x_1 & x_0 & -x_3 & x_2 \\ x_2 & x_3 & x_0 & -x_1 \\ x_3 & -x_2 & -x_1 & x_0 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad (18)$$

公式(17)與公式(18)分別對於右乘 y 、左乘 x 都可以再更進一步分解成四個純量乘法的向量和，可應用於 AVX 256-bit 向量暫存器的 simd 向量化程式實作[32]。以公式(17)為例，可導出

$$z = y_0 \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} + y_1 \begin{bmatrix} -x_1 \\ x_0 \\ x_3 \\ -x_2 \end{bmatrix} + y_2 \begin{bmatrix} -x_2 \\ -x_3 \\ x_0 \\ x_1 \end{bmatrix} + y_3 \begin{bmatrix} -x_3 \\ x_2 \\ -x_1 \\ x_0 \end{bmatrix} \quad (19)$$

針對單位四元數

$$\begin{aligned} g &= g_0 + g_1 i + g_2 j + g_3 k \\ &= \cos \frac{\theta}{2} + \left(\frac{g_1 i + g_2 j + g_3 k}{\sqrt{g_1^2 + g_2^2 + g_3^2}} \right) \sin \frac{\theta}{2} \end{aligned} \quad (20)$$

其中

$$\theta = 2 \cos^{-1} g_0 = 2 \sin^{-1} \sqrt{g_1^2 + g_2^2 + g_3^2} \quad (21)$$

在三度空間中的任一點 $x = (x_1, x_2, x_3)$ $= x_1 i + x_2 j + x_3 k$ 針對旋轉軸向量 (g_1, g_2, g_3) $= g_1 i + g_2 j + g_3 k$ 逆時針方向旋轉徑度 θ 至向量 $z = (z_1, z_2, z_3) = z_1 i + z_2 j + z_3 k$ ，可透過 Rodrigues 旋轉公式計算，即

$$z = g x g^{-1} \quad (22)$$

當中

$$g^{-1} = \bar{g} = \cos \frac{\theta}{2} - \left(\frac{g_1 i + g_2 j + g_3 k}{\sqrt{g_1^2 + g_2^2 + g_3^2}} \right) \sin \frac{\theta}{2} \quad (23)$$

其計算結果即左乘一代數群 SO(3) 中的 3x3 矩陣

$$M_g = \begin{bmatrix} 1-2g_2^2-2g_3^2 & 2g_1g_2-2g_0g_3 & 2g_1g_3+2g_0g_2 \\ 2g_1g_2+2g_0g_3 & 1-2g_1^2-2g_3^2 & 2g_2g_3-2g_0g_1 \\ 2g_1g_3-2g_0g_2 & 2g_2g_3+2g_0g_1 & 1-2g_1^2-2g_2^2 \end{bmatrix} \quad (24)$$

2.3 四元數 Zernike 多項式與四元數 Zernike 矩

考慮任何一個 μ 純虛數單位四元數，即 $\mu = \mu_1 i + \mu_2 j + \mu_3 k$ 且 $\mu_1^2 + \mu_2^2 + \mu_3^2 = 1$ 當中 μ_1, μ_2, μ_3 均為實數。藉由四元數版的 de Moivre 公式

$$e^{\mu m \theta} = \cos(m\theta) + \mu \sin(m\theta) \quad (25)$$

與 Zernike 多項式類似，四元數 Zernike 多項式定義為

$$V_{nm}^{\mu}(z) = R_{nm}(r) e^{\mu m \theta} \\ = R_{nm}(r) (\cos(m\theta) + \mu \sin(m\theta)) \quad (26)$$

而 RGB 影像函數 $f(z)$ 可以表示如下

$$f(z) = f_B(z)i + f_G(z)j + f_R(z)k \quad (27)$$

當中 $f_R(z)$ 、 $f_G(z)$ 與 $f_B(z)$ 各別為影像中的紅色、綠色與藍色頻道中的值。其相對應的右邊四元數 Zernike 矩定義為

$$Z_{nm}^{\mu}(f) = \frac{n+1}{\pi} \iint_D f(z) \overline{V_{nm}^{\mu}(z)} \frac{dz \wedge d\bar{z}}{-2i} \\ = \frac{n+1}{\pi} \iint_D (f_B(z)i + f_G(z)j + f_R(z)k) \\ \square R_{nm}(r) (\cos(m\theta) - \mu \sin(m\theta)) \frac{dz \wedge d\bar{z}}{-2i} \\ = Q_0 + Q_1 i + Q_2 j + Q_3 k \quad (28)$$

當中 Q_0, Q_1, Q_2 、以及 Q_3 均為實數，計算中主要處理右乘純虛數單位四元數 μ ，相關計算可透過矩陣公式(17)或是公式(19)進行。一般而言，一次四元數的乘法，需要 16 個乘法、12 加減法，由於單位四元數 μ 是純虛數，可將一次四元數的乘法縮減為 9 個乘法與 5 個加減法，其計算方式如下：

$$f(z)\mu = -f_B(z)\mu_1 - f_G(z)\mu_2 - f_R(z)\mu_3 \\ + i(f_G(z)\mu_3 - f_R(z)\mu_2) \\ + j(f_R(z)\mu_1 - f_B(z)\mu_3) \\ + k(f_B(z)\mu_2 - f_G(z)\mu_1) \quad (29)$$

這比起分別以 R、G、B 頻道分別各考慮個別對應的 Zernike 矩，需要考慮六維的實數空間，當然是更為方便。四元數 Zernike 矩的重組圖函數則是定義為：

$$f_M(z) = \sum_{n=0}^M \sum_{\substack{m=-n \\ m=n(2)}}^n Z_{nm}^{\mu} V_{nm}^{\mu}(z) \quad (30)$$

在評估重組圖好壞，特別是重組圖函數 f_M 與原影像函數 f 間的誤差，採用 NMSE (Normalized Mean Squared Error)，其定義為

$$\epsilon^2 = \frac{\iint_D |f(x, y) - f_M(x, y)|^2 dx dy}{\iint_D |f(x, y)|^2 dx dy} \quad (31)$$

公式(28)中四元數 Zernike 矩與分別以 R、G、B 頻道所得的 Zernike 矩之間的關係整理於下列定理：

定理 A. 符號如上，四元數 Zernike 矩為

$$Z_{nm}^{\mu}(f) = Q_0 + Q_1 i + Q_2 j + Q_3 k$$

則

$$Q_0 = \mu_1 RZ_{n,-m}(f_B) + \mu_2 RZ_{n,-m}(f_G) \\ + \mu_3 RZ_{n,-m}(f_R) \quad (32)$$

$$Q_1 = RZ_{nm}(f_B) - \mu_3 RZ_{n,-m}(f_G) \\ + \mu_2 RZ_{n,-m}(f_R) \quad (33)$$

$$Q_2 = RZ_{nm}(f_G) - \mu_1 RZ_{n,-m}(f_R) + \mu_3 RZ_{n,-m}(f_B) \quad (34)$$

與

$$Q_3 = RZ_{nm}(f_R) - \mu_2 RZ_{n,-m}(f_B) + \mu_1 RZ_{n,-m}(f_G) \quad (35)$$

證明：計算 $f(z)\mu\sin(m\theta)$ 透過公式(29)，並加上

$$f(z)\cos(m\theta) = f_B(z)\cos(m\theta)i + f_G(z)\cos(m\theta)j + f_R(z)\cos(m\theta)k \quad (36)$$

可得

$$Q_0 = \frac{n+1}{\pi} \iint_D (f_B\mu_1 + f_G\mu_2 + f_R\mu_3) \bullet R_{nm}(r)\sin(m\theta)rdrd\theta \quad (37)$$

$$Q_1 = \frac{n+1}{\pi} \iint_D (f_B\cos m\theta - (f_G\mu_3 - f_R\mu_2)\sin(m\theta)) \bullet R_{nm}(r)rdrd\theta \quad (38)$$

$$Q_2 = \frac{n+1}{\pi} \iint_D (f_G\cos m\theta - (f_R\mu_1 - f_B\mu_3)\sin(m\theta)) \bullet R_{nm}(r)rdrd\theta \quad (39)$$

以及

$$Q_3 = \frac{n+1}{\pi} \iint_D (f_R\cos m\theta - (f_B\mu_2 - f_G\mu_1)\sin(m\theta)) \bullet R_{nm}(r)rdrd\theta \quad (40)$$

透過轉換公式(7)、(8)，以及積分算子為線性的特性，整理後可得結果。 ■

值得一提的，定理 A 也可以用來計算四元數 Zernike 矩，前提是要先算出 R、G、B 三個個別通道實數版的 Zernike 矩，這樣的計

算方式，並沒有用到四元數計算的優勢。為方便計算，我們建議將四元數 Zernike 矩分解成餘弦部份與正弦部份的和：

$$Z_n^\mu(f) = C_n + f' \quad (41)$$

其中餘弦部份與 μ 無關，計算公式如下

$$C_{nm}(f) = \frac{n+1}{\pi} \iint_D \left(\frac{f(r,\theta)R_{nm}(r)}{\cos(m\theta)} \right) rdrd\theta = \frac{Z_{nm}^\mu(f) + Z_{n,-m}^\mu(f)}{2} \quad (42)$$

而正弦部份與 μ 相關，計算方式如下

$$S_{nm}^\mu(f) = -\frac{n+1}{\pi} \left(\iint_D \frac{R_{nm}(r)\sin(m\theta)}{f(r,\theta)rdrd\theta} \right) \mu = \frac{Z_{nm}^\mu(f) - Z_{n,-m}^\mu(f)}{2} \quad (43)$$

透過(41)-(43)，可得

$$Z_{n,-}^\mu(f) = C_n - f' \quad (44)$$

為了考慮理論的完整性，也討論左邊四元數 Zernike 矩，其定義為

$$L_{nm}^\mu(f) = \frac{n+1}{\pi} \iint_D \overline{V_{nm}^\mu(z)} f(z) \frac{dz \wedge d\bar{z}}{-2i} = \frac{n+1}{\pi} \iint_D (\cos(m\theta) - \mu\sin(m\theta)) R_{nm}(r) \square(f_B(z)i + f_G(z)j + f_R(z)k) \frac{dz \wedge d\bar{z}}{-2i} \quad (45)$$

右邊四元數 Zernike 矩與左邊四元數 Zernike 矩之間的關連，可推導出下列公式：

定理 B.

$$L_{nm}^\mu(f) = -\overline{Z_{n,-m}^\mu(f)} \quad (46)$$

證明：左邊四元數 Zernike 矩也可如四元數 Zernike 矩右邊分解成餘弦部份與正弦部份的和：

$$L_{nm}^\mu(f) = C_{nm}(f) + T_{nm}^\mu(f) \quad (47)$$

其中

$$T_{nm}^{\mu}(f) = -\frac{n+1}{\pi} \mu \left(\iint_D R_{nm}(r) \sin(m\theta) f(r, \theta) r dr d\theta \right) \quad (48)$$

計算

$$\begin{aligned} \mu f(z) &= -f_B(z)\mu_1 - f_G(z)\mu_2 - f_R(z)\mu_3 \\ &\quad -i(f_G(z)\mu_3 - f_R(z)\mu_2) \\ &\quad -j(f_R(z)\mu_1 - f_B(z)\mu_3) \\ &\quad -k(f_B(z)\mu_2 - f_G(z)\mu_1) \end{aligned} \quad (49)$$

比較 $f(z)\mu$ 在(29)的計算結果，可得

$$\mu f(z) = \overline{f(z)\mu} \quad (50)$$

因此可得

$$\begin{aligned} \overline{-\sin(-m\theta) f(z)\mu} &= \sin(m\theta) \overline{f(z)\mu} \\ &= \sin(m\theta) \mu f(z) \end{aligned} \quad (51)$$

可得

$$T_{nm}^{\mu} = -\overline{S_{n,-m}^{\mu}} \quad (52)$$

另外由於 $C_{nm}(f)$ 是純虛數 $\cos(\cdot)$ 是偶函數得

$$C_{nm}^{\mu} = -\overline{C_{n,-m}^{\mu}} \quad (53)$$

帶回(41)、(47)得證。■

由於四元數 Zernike 矩的值是與 μ 純虛數單位四元數的選取有關，用不同純虛數單位四元數所算出的四元數 Zernike 矩的值，其實可透過旋轉轉換公式求得，其結果整理於下列定理：

定理 C. 考慮兩個純虛數單位四元數 μ, ν ，滿足 $\nu = g\mu g^{-1}$ 其中 $g = g_0 + g_1 i + g_2 j + g_3 k$ 為單位四元數，則

$$Z_{nm}^{\nu}(f) = g Z_{nm}^{\mu}(f) g^{-1}$$

(54)

證明：考慮 ν 所定義出的四元數 Zernike 矩

$$\begin{aligned} Z_{nm}^{\nu} &= \frac{n+1}{\pi} \iint_D f(z) \overline{V_{nm}^{\nu}(z)} \frac{dz \wedge d\bar{z}}{-2i} \\ &= \frac{n+1}{\pi} \iint_D (f_B(z)i + f_G(z)j + f_R(z)k) \\ &\quad \square R_{nm}(r)(\cos(m\theta) - \nu \sin(m\theta)) \frac{dz \wedge d\bar{z}}{-2i} \end{aligned} \quad (55)$$

當中指數部份的計算（實數與四元數的乘積為純量乘法，是可交換的）

$$\begin{aligned} &\cos(m\theta) - \nu \sin(m\theta) \\ &= \cos(m\theta) - g\mu g^{-1} \sin(m\theta) \\ &= g(\cos(m\theta))g^{-1} - g(\mu \sin(m\theta))g^{-1} \\ &= g(\cos(m\theta) - \mu \sin(m\theta))g^{-1} \end{aligned} \quad (56)$$

代回式子(55)，整理後可得定理結果。■

四元數 Zernike 矩的絕對值是個不變量，其值與 μ 選取是無關。

定理 D. 考慮兩個純虛數單位四元數 μ, ν ， $Z_{nm}^{\mu}(f)$ 與 $Z_{nm}^{\nu}(f)$ 分別為不同四元數所訂出的四元數 Zernike 矩。則

$$|Z_{nm}^{\nu}(f)| = |Z_{nm}^{\mu}(f)| \quad (57)$$

證明：對於兩個純虛數單位四元數

$$\mu = (\mu_1, \mu_2, \mu_3), \nu = (\nu_1, \nu_2, \nu_3),$$

選取單位四元數

$$g = \cos \frac{\theta}{2} + \lambda \sin \frac{\theta}{2} \quad (58)$$

當中 $\lambda = \lambda_1 i + \lambda_2 j + \lambda_3 k$ 為單位純虛數且與 $\mu \times \nu$ 外積的向量平行，由於 μ 與 ν 是單位四元數，徑度則為

$$\theta = \cos^{-1}(\mu \bar{\nu}) = \sin^{-1}|\mu \times \nu| \quad (59)$$

根據 Rodrigues 旋轉公式(20)-(23)，可得
 $\nu = g\mu g^{-1}$ 。因此藉由定理 C 得

$$\begin{aligned} |Z_{nm}^{\nu}(f)| &= |gZ_{nm}^{\mu}(f)g^{-1}| \\ &= |g||Z_{nm}^{\mu}(f)||g^{-1}| \\ &= |Z_{nm}^{\mu}(f)| \end{aligned} \quad (60)$$

三、演算方法

3.1 離散化的計算公式

計算一個影像大小為 $N \times N$ 的四元數 Zernike 矩，以像素 $P(s, t)$ 小方格為例，其中心點的座標為

$$(x_s, y_t) = \eta(s, t) = \left(\frac{2s - N + 1}{N\sqrt{2}}, \frac{2t - N + 1}{N\sqrt{2}} \right) \quad (61)$$

為一寬度為 $\sqrt{2}/N$ 的正方形（當中 $s, t = 0, 1, \dots, N-1$ ）。如此可將整個影像嵌入單位盤 D 中的內接正方形：

$$A: \left[\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right] \times \left[\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right] \quad (62)$$

因此，定義域為 A 的影像函數 $f(x, y)$ 可經由公式(61) $f(\eta(s, t)) = P(s, t)$ 給定，而其函數對應域就是四元數中所有純虛數所形成的集合。採用的計算的方式為以下離散型四元數 Zernike 矩公式：

$$\begin{aligned} \hat{Z}_{nm}^{\mu} &= \frac{2(n+1)}{\pi N^2} \sum_s \sum_t f(x_s, y_t) R_{nm}(r_{st}) e^{-\mu m \theta_{st}} \\ &= \frac{2(n+1)}{\pi N^2} \sum_s \sum_t \begin{pmatrix} f_B(x_s, y_t) i \\ + f_G(x_s, y_t) j \\ + f_R(x_s, y_t) k \end{pmatrix} R_{nm}(r_{st}) e^{-\mu m \theta_{st}} \end{aligned} \quad (63)$$

其中 (r_{st}, θ_{st}) 為像素 $P(s, t)$ 中心點座標 (x_s, y_t) 的極座標 $(x_s = r_{st} \cos \theta_{st}, y_t = r_{st} \sin \theta_{st})$ 。

3.2 Radial 多項式的計算方式

根據我們在 Zernike 矩的計算經驗[26, 28-31]，直接採用 Radial 多項式的係數公式，如公式(3)，直接計算，不僅耗時（計算到所有階數小於等於 M 的 Zernike 矩，即使 Zernike radial 多項式的各項係數，如公式(4)中的階乘，其計算是利用查表法，其總計算複雜度為 $O(N^2 M^3)$ ），而且在 $M > 45$ 之際，因為其計算誤差累積太大，還原圖根本不能採用。因此，直接法、係數公式法、或是從係數公式所衍生的方法，在於高階矩的計算，強烈建議不予採用。要能精確並且快速的計算，採用 Radial 多項式的遞迴關係式的計算方法，就成為較佳的選擇。

其中一種遞迴方法，為許多數學學者所採用，稱之為 q 遞迴法 (q -recursive method)：

$$R_n(r) = \frac{K_{n-1} R_{n-1}(r) + K_n R_{n+1}(r)}{r} \quad (64)$$

其中 $m = n-4, -n-6$ ，或...

當中初始條件為：

$$R_m(r) = r^n, n \geq 0 \quad (65)$$

$$R_{n,n-2}(r) = nr^n - (n-1)r^{n-2}, n \geq 2 \quad (66)$$

係數滿足條件：

$$K_1 = \frac{(m+4)(m+3)}{2} - (m+4)K_2 + \frac{K_3(n+m+6)(n-m-4)}{8} \quad (67)$$

$$K_2 = \frac{K_3(n-m-4)}{4(n+3)} + (m+2) \quad (68)$$

$$K_3 = \frac{-4(n+2)}{(n+m+2n)}$$

(69)

其中 $n \geq 4, m \leq n-4$ 。

另外一類遞迴關係式，稱之為四項次遞迴式，源起於一個 Zernike Radial 多項式 DCT 表達式[21, 26, 33]，我們也給了組合方式的證明[31]，為本文所建議的方法，其公式如下：

$$R_{n,|m|}(r) = r(R_{n-1,|m-1|}(r) + R_{n-1,|m+1|}(r)) - R_{n-2,|m|}(r) \quad (70)$$

(當中 $n \geq 2$)

其初始條件為(65)。初始條件(65)中 $R_{nm}(r)$ 的計算，可由下列方式計算：

$$\begin{cases} r^0 = 1 \\ r^2 = \frac{(2s-N+1)^2 + (2t-N+1)^2}{2N^2} \\ r = \sqrt{r^2} \\ r^k = r^{k-2} \cdot r^2 \quad \text{當 } k = 3, 4, \dots, M \end{cases} \quad (71)$$

3.3 指數部份的計算方式

指數部份的計算，其實就是在計算三角函數 $\cos m\theta_{st}$ 、 $\sin m\theta_{st}$ 的值，其計算方式，可透過下列三角計算公式進行：先計算初始值

$$\cos 0 = 1, \sin 0 = 0, \cos \theta = \frac{x}{r}, \sin \theta = \frac{y}{r} \quad (72)$$

$$\tan \theta = \frac{2t-N+1}{2s-N+1} \quad (73)$$

$$\cos(2\theta) = \frac{1-\tan^2 \theta}{1+\tan^2 \theta}, \sin(2\theta) = \frac{2\tan \theta}{1+\tan^2 \theta} \quad (74)$$

再用和角公式以 for 迴圈計算：

For $m=3$ to M step 1

$$\begin{aligned} \cos(m\theta) &= \cos((m-2)\theta)\cos(2\theta) \\ &\quad - \sin((m-2)\theta)\sin(2\theta) \end{aligned}$$

$$\begin{aligned} \sin(m\theta) &= \sin((m-2)\theta)\cos(2\theta) \\ &\quad + \sin(2\theta)\cos((m-2)\theta) \end{aligned}$$

Next m (75)

由於我們避免使用系統內設的數值函數如 $\sin(\cdot)$ 、 $\cos(\cdot)$ 、 $\tan(\cdot)$ 等逼近三角函數、反三角函數的數值函數，而是採定義、線性遞迴關係是計算，所求得的三角函數數值精準，對於整體計算的誤差累積都能在控制範圍之內。

3.4 建議的演算法

計算四元數 Zernike 矩的演算方法，如圖 1 所示。第一個步驟就是先取得影像函數的值

```

取得影像函數的值  $f[s][t] = f(x_s, y_t)$ ，並計算所有四元數
乘法  $f\_mU[s][t] = f(x_s, y_t)\mu$  (for  $s, t=0,1,\dots, N-1$ )
for each ( $s, t=0,1,\dots, N-1$ ) {
     $(r, \theta) = (x_s, y_t)$  的極座標
    計算所有  $COS[m] = \cos(m\theta)$ 、 $SIN[m] = \sin(m\theta)$ 
    (for  $m=0,1,\dots,M$ )
    計算所有  $rr[k] = r^k$  (for  $m=0,1,\dots,M$ )
     $RnM1[0]=1$ 
     $Z[0][0]=Z[0][0]+2/(\pi N^2)*f[s][t]$ 
    for ( $n=1; n \leq M; n++$ ) {
         $c = 2(n+1)/(\pi N^2)$ 
         $(c0, c1) = c*(f[s][t], -f\_mU[s][t])$ 
         $Rnm = rr[n]$ 
         $RnM2[n-1] = RnM1[n-1]$ 
         $Z[n][n] = Z[n][n] + (Rnm * COS[n]) * c0$ 
        //m=n-1 stands for  $Z[n][-n]$ 
         $Z[n][n-1] = Z[n][n-1] + (Rnm * SIN[n]) * c1$ 
         $RnM1[n] = Rnm$ 
        if ( $n$  是偶數) {
             $Rnm = 2*r*RnM1[1] - RnM2[0]$ 
             $Z[n][0] = Z[n][0] + (Rnm * c0)$ 
             $RnM1[0] = Rnm$ 
        }
        for ( $m=n-2; m > 0; m-=2$ ) {
             $Rnm = r*(RnM1[m-1] + RnM1[m+1]) - RnM2[m]$ 
             $RnM2[m-1] = RnM1[m-1]$ 
             $Z[n][m] = Z[n][m] + (Rnm * COS[m]) * c0$ 
            //m-1 stands for  $Z[n][-m]$ 
             $Z[n][m-1] = Z[n][m-1] + (Rnm * SIN[m]) * c1$ 
             $RnM1[m] = Rnm$ 
        }
    }
}

```

圖 1. 建議計算四元數 Zernike 矩的演算方法。

$f(x_s, y_t)$ 並計算四元數乘法 $f(x_s, y_t)\mu$ ，分別

儲存於 Quaternion 陣列 $f[0:N-1][0:N-1]$ 與 $f_mU[0:N-1][0:N-1]$ 在索引 (s, t) 的元素。

對於每個像素的代表點 (x_s, y_t) (其中 (r_s, θ_s) 為極座標)，利用(72)-(75)的方法計算三角函數值 $\cos(m\theta)$ 、 $\sin(m\theta)$ 分別儲存於陣列 $COS[0:M]$ 與 $SIN[0:M]$ 。而實數冪次方，以(71)所示演算計算， r_{st}^k ($k=0,1,\dots,M$)就儲存於陣列

於 $rr[0:M]$ 。

四元數 Zernike 矩 Z_{nm}^μ 的處理方式，主要是計算 Zernike 矩的餘弦部份 C_{nm} 以及正弦部份 S_{nm}^μ ，如(41)-(43)所定義的，分別儲存於 Quaternion 陣列元素 $Z[n][m]$ 與 $Z[n][m-1]$ 。當中的 radial 多項式 $R_{nm}(r_{st})$ 值的計算，是採用四項遞迴公式(70)，並用到暫存記憶體 Rnm 、 $RnM1[0:M]$ 以及 $RnM2[0:M]$ 儲存計算過程中的值。

整個演算方法，雖然儘量降低最耗時的四元數乘法次數，但也至少要進行 N^2 次。其他四元數的計算如四元數加減法、純量乘法，都能以 simd 向量計算的方式進行。整體的計算複雜度為 $O(N^2M^2)$ ，為快速且精準的演算方法。

3.5 快速 Farey 數列計算法

透過 3.4 節所介紹的方法計算，已經是達到最快的數量級，但是利用代數群所產生的對稱性，可以大幅減少計算量，考慮由旋轉、鏡射所形成的群，如：

$$D_4 = \left\{ \tau_\theta, t_\theta \mid \theta = 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2} \right\} \quad (76)$$

當中 τ_θ 表逆時鐘方向旋轉 θ 而 t_θ 表在線上 $\theta/2$ 的鏡射。利用二面體群 (Dihedral Group) D_4 群作用所產生的對稱性，能有效地將量減到原計算量的 $1/8$ 。原離散計算公式(63)可進一步修改為

$$\hat{Z}_{nm} = \frac{2(n+1)}{\pi N^2} \left(\sum_{\substack{(x_s, y_t) \in H \\ x_s \neq y_t}} \sum_{\sigma \in D_4} f(\sigma(x_s, y_t)) R_{nm}(r_{st}) e^{-\mu m \sigma(\theta_{st})} \right) + \frac{1}{2} \sum_{(x_s, x_s) \in H} \sum_{\sigma \in D_4} f(\sigma(x_s, x_s)) R_{nm}(r_{ss}) e^{-\mu m \sigma(\frac{\pi}{4})} \quad (77)$$

當中 H 為圖 2 中所示正方形 A 中之 $1/8$ 大小的藍色直角三角形區塊，在對角線 $x=y$ 上的像素，如黃色像素 Q ，只有 4 個 D_4 共軛像素，其他在 H 內部的像素都會有 8 個 D_4 共軛像素。在公式(77)中，這些在對角線 $x=y$ 上的像素必須乘以 $1/2$ 。

由於像素的索引 s, t 都是非負整數，相同的徑度 $\theta_{st} = \tan^{-1}(y_t / x_s)$ 的三角函數數值， $(\tan \theta_{st} = (2t - N + 1) / (2s - N + 1))$ 沒有必要重複

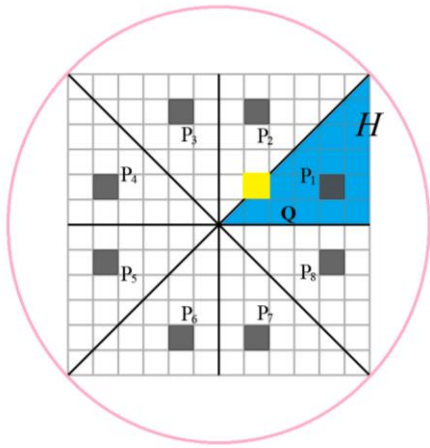


圖 2. 利用 D_4 對稱性加速計算示意圖。

計算，這時就可排除 $\gcd(x_s, y_t) > 1$ 的情況，此時 Farey 數列就可以派上用場。

當 N 為偶數時，就考慮階數 $< N$ 的 Farey 數列中分子、分母皆為奇數的子數列 $\hat{F}_N$ 。Farey 子數列的元素個數可由下列漸近公式估計 [26]：

定理 E. Farey 子數列的元素個數滿足下列漸近公式：

$$|\hat{F}_N| = \frac{N^2}{\pi^2} (1 + o(1)) \quad (78)$$

三角函數的總計算量，由於 H 上的像素代表

點有 $(N^2 + N) / 8$ 個。可以進一步減至使用公式 (77) 的 81.0569%。考慮 H 上的像素任一代表點 (x, y) ，得

$$(x, y) = d(p, q) \quad (79)$$

當中 $d = \gcd(x, y)$ 為正奇數且 $d \leq \lfloor (N-1)/q \rfloor$ ， $p = x/d$ 且 $q = y/d$ ， p/q 為分子、分母皆為小於等於 N 之正奇數的不可約分數，因此 $p/q \in \hat{F}_N$ 。因此 H 上的所有像素代表點與下列集合有一對一對應

$$I_N = \left\{ \left(\frac{p}{q}, d \right) \left| \begin{array}{l} \frac{p}{q} \in \hat{F}_N, \\ d \leq \left\lfloor \frac{N-1}{q} \right\rfloor \text{ 且 } d \text{ 為正奇數} \end{array} \right. \right\} \quad (80)$$

Farey 數列法基本上就是依 I_N 的順序計算 H 上的像素所有代表點。

3.6 建構 Quaternion Class

整個演算進行，當然可以實數計算進行，但是四元數本身擁有良好的代數結構，本身就是四維的實數向量空間，當中所有的算術計算都可以向量、矩陣運算的方式進行。

我們利用 intel intrinsics 指令 [32]，使用 AVX 256-bit 向量暫存器，實作了四元數向量加減法、四元數的純量乘法（如圖 3），以及利用公式(19)實作一般性的四元數乘法。由於 AVX 256-bit 向量暫存器一次可以處理 4 個 double 型態的數值，非常契合四元數的算術計算。在不使用 AVX 256-bit 向量暫存器的情況下，我們也實作了公式(29)。使用 AVX 256-bit 向量暫存器的 Quaternion Class，在處理四元數 Zernike 矩的計算，有非常顯著的加速效應。

```
// compute scalar multiplication
static inline Quaternion operator*( double
const & a, Quaternion const & b)
{
/*bb=(b0,  b1,  b2,  b3)  stands  for
b0+b1i+b2j+b2k */
    __m256d bb=_mm256_load_pd(b.q);
//aa=(a, a, a, a)
    __m256d aa=_mm256_set1_pd(a);
//pointwise multiplication of vectors aa & bb
    __m256d x0=_mm256_mul_pd( aa, bb );
    return Quaternion(x0);
}
```

圖 3. 利用 intel intrinsics 指令使用 AVX 256-bit 向

量暫存器實作純量乘法。

四、實驗設計與數值結果

4.1 實驗基本設定

演算法是以 C/C++ 程式語言撰寫實作程式，並搭配 intel intrinsics 指令，使用 AVX 256-bit 向量暫存器。我們採用 TMD-GCC 4.9.2 64-bit C/C++ release 編譯器、O2 優化，採用單一執行緒方式撰寫我們所發展的演算方法。所有實數相關的數值計算，全都是採用 64 位元格式的 double 處理。電腦的作業系統為 win 7，採用 Intel i7- 4790 quad-core 3.6 GHz 的處理器，Cache 有 8M。

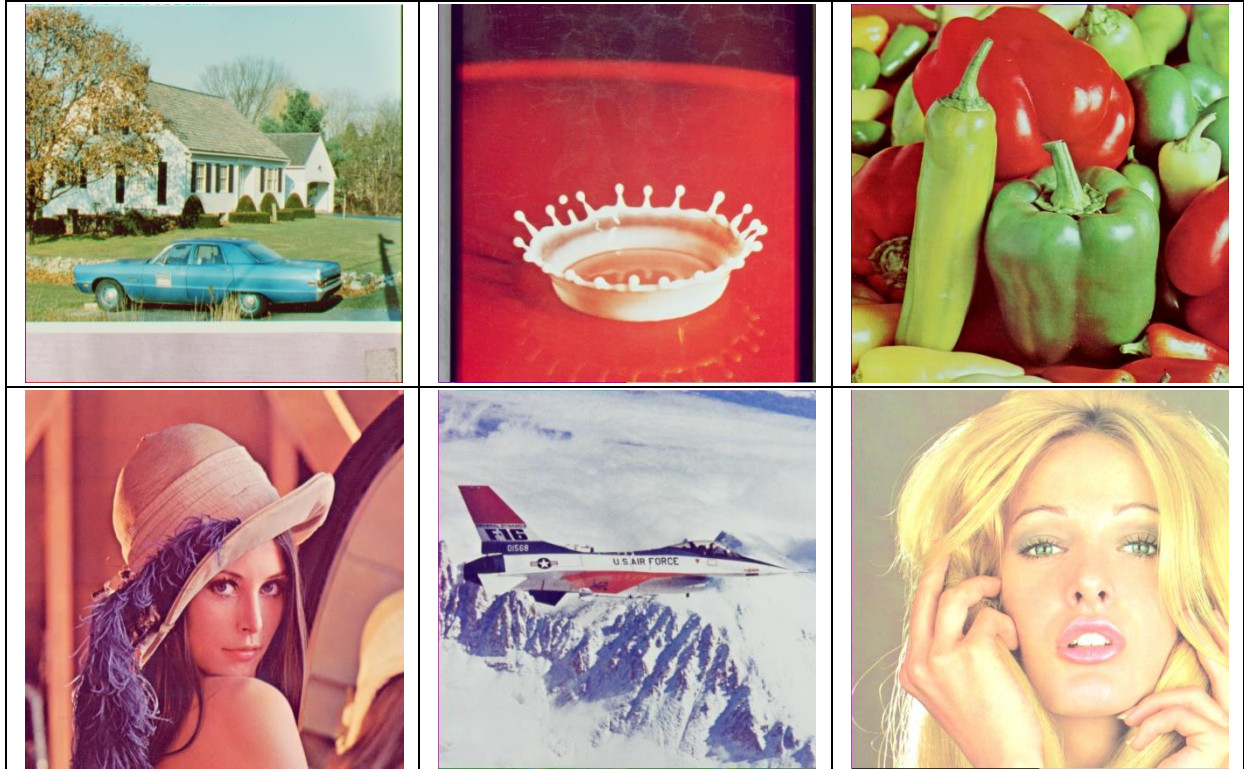


圖 4. 六張 512x512 大小的彩色測試圖。

所有的測試圖檔都是 512x512 的彩色圖 如圖 4 所示，分別為 house and car、splash、

peppers、Lena、F-16 以及 Tiffany。程式對於讀、寫一個圖檔的時間也不到 10 毫秒。比起計算高階四元數 Zernike 矩的時間，相對而言，非常微小的。我們實驗所呈現的執行時間 (elapsed time) 都是測試這六張圖的平均值。

4.2 不同階數誤差的實驗

假設純虛數單位四元數 $\mu = (i+j+k)/\sqrt{3}$ 。由於 RGB 彩色圖的高階矩計算龐大，所有實作的演算方法，皆以 Farey 數列計算法加速計算，我們考慮下列三種方法：

- A: 採用四元數計算，使用四項次遞迴式如(70)，加上 Farey 數列計算法加速。
- B: 採用 RGB 個別頻道分開計算處理，使用四項次遞迴式如(70)，加上 Farey 數列計算法加速。

- C: 採用四元數計算，使用 q 遞迴方法如 (64)-(69)，加上 Farey 數列計算法加速。

我們以 A、B 與 C 三法分別計算測試圖 Tiffany 在最高階數從 50 至 500 之間的還原表現。並計算其 NMSE 值於表 2，並且在表 1 展示其在高階 500 時的還原圖，使用法 A、法 B 時，除了在四角邊緣有可見白色的明顯誤差外，其餘部份幾乎與原圖一致，此與 Radial 多項式值在 $r \rightarrow 1$ 、高階 n 、較小絕對值的重複數 m 時震盪得很厲害，且 $R_{nm}(1) = 1 \forall n, m$ ，直接會有計算上的關聯性；可觀察 Radial 多項式 $R_{500,0}(r)$ 在閉區間 $[0.98, 1]$ 的函數圖形如圖 5 所示，其中的確是越接近 1 震盪得越厲害，其繪製方式是依照四項遞迴公式(70)計算描點而成。值得注意的是，使用法 A 的誤差，

表 1 測試圖 Tiffany 使用 A、B、C 等法計算在最高階 500 時的還原圖與 NMSE

		
(A) $\varepsilon^2 = 0.0017582$	(B) $\varepsilon^2 = 0.0017637$	(C) $\varepsilon^2 = 0.0021271$

表 2. 不同方法在不同最高階數還原圖的 NMSE 值

影像 階數	A	B	C
50	0.0112199	0.0111715	0.0111715
100	0.0060775	0.0060571	0.0060571
150	0.0044710	0.0044604	0.0044645
200	0.0034693	0.0034636	0.0035042
250	0.0028540	0.0028513	0.0029365
300	0.0023709	0.0023703	0.0025117

350	0.0020319	0.0020329	0.0022278
400	0.0017724	0.0017748	0.0020222
450	0.0016790	0.0016829	0.0019909
500	0.0017582	0.0017637	0.0021271

在數據上是更小於法 B，顯示使用採用四元數計算，比起採用 RGB 個別頻道分開計算處理，

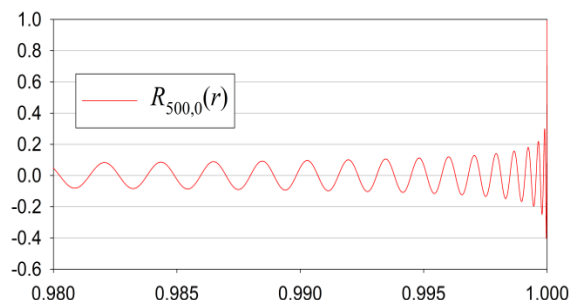


圖 5. Radial 多項式 $R_{500,0}(r)$ 在閉區間 $[0.98, 1]$ 的函數圖形。

有最佳的還原圖表現。法 C 採用 q 遞迴方法，由於該演算法在計算 Radial 多項式時，特別是在高階 n 、較小絕對值的重複數 m 時，在接近圓心處會有值為零的計算錯誤現象[26, 34]，這是在高階的四元數 Zernike 矩還原圖，出現有明顯漣漪的現象的最主要原因。類似在 Zernike 矩的計算經驗，以 q 遞迴方法為基礎的法 C，並不適用於階數超過 250 以上之四元數 Zernike 矩的計算。

4.3 不同測試圖誤差實驗

此實驗 $\mu = (i+j+k)/\sqrt{3}$ ，我們將法 A 作用到所有的測試圖，計算所有階數小於等於 500 的四元數 Zernike 矩，並計算還原圖之 NMSE 值，其實驗數據列在表 3。可以觀察到，這六張彩色圖的還原圖之 NMSE 是介於 splash 的 0.0017049 以及 peppers 的 0.0037278 之間。peppers 的還原圖如圖 6 所示。



圖 6. 測試圖 peppers 使用法 A 計算在最高階 500 時的還原圖。

表 3. 測試圖在最高階 500 還原所產生的誤差

測試圖	house	splash	peppers
NMSE	3.0806	1.7049	3.7278
測試圖	Lena	F-16	Tiffany
NMSE	2.6785	1.8413	1.7582

(單位: 10^{-3})

4.4 採用不同 μ 計算的實驗

在此實驗中，我們考慮法 A，使用不同的純虛數單位四元數 μ 計算四元數 Zernike 矩至最高階數 500。不同 μ 的選取如下：

$$\begin{cases} \mu^{(0)} = i \\ \mu^{(1)} = (i+j+k)/\sqrt{3} \\ \mu^{(2)} = i/\sqrt{3} + j/\sqrt{2} + k/\sqrt{6} \\ \mu^{(3)} = (i+j)/\sqrt{2} \\ \mu^{(4)} = i/\sqrt{2} + j/\sqrt{10} + k\sqrt{2}/\sqrt{5} \end{cases} \quad (81)$$

測試圖 Lena 的還原表現，並以其 NMSE 值取到小數點後八位，所有的值均是 0.00267849。顯示 μ 值的選取，並不影響還原圖的品質。並且比較法 B，其 NMSE 值為 0.00268286。經由不同 μ 值的選取比較，在數據上顯示，使用採用四元數計算，的確比起採用 RGB 個別頻道分開計算處理，有最佳的計算的精準表現。

4.5 測試不同計算方法速度的實驗

使用純虛數單位四元數 $\mu = (i+j+k)/\sqrt{3}$ 。

所有實作的演算方法，皆以 Farey 數列計算法加速計算，我們考慮四種不同的設定與方法計算四元數 Zernike 矩，並且比較兩種計算 Zernike 矩的方法：

- A0: 採用四元數計算，使用四項次遞迴式如(70)，加上 Farey 數列計算法加速。Quaternion Class 不使用 AVX 256-bit 向量暫存器。
- A1: 採用四元數計算，使用四項次遞迴式如(70)，加上 Farey 數列計算法加速。Quaternion Class 使用 AVX 256-bit 向量暫存器。
- C0: 採用四元數計算，使用 q 遞迴方法如(64)-(69)，加上 Farey 數列計算法加速。Quaternion Class 不使用 AVX 256-bit 向量暫存器。
- C1: 採用四元數計算，使用 q 遞迴方法如(64)-(69)，加上 Farey 數列計算法加速。Quaternion Class 使用 AVX 256-bit 向量暫存器。
- R: 使用四項次遞迴式如(70)，加上 Farey 數列計算法加速計算灰階圖上的 Zernike 矩。
- Q: 使用 q 遞迴方法如(64)-(69)，加上 Farey 數列計算法加速計算灰階圖上的 Zernike 矩。

法 A0、A1、C0、C1 計算四元數 Zernike 矩在最高階數 0-500 所花得平均時間如圖 7 所示。當中以最高階數 $M=500$ 為例，A0 需時 13.0653 秒，A1 需時 6.8563 秒，C0 需時 21.3156 秒，而 C1 需時 15.4625 秒。就計算速度而言，在最高階數介於 50 與 400 之間，實驗數據顯示，A0 最快、C1 次之、A1 第三，而 C0 最慢；

但是從最高階數 450 起，A0 的執行速度就追上 C1，成為第二。很明顯，以四項次遞迴公式為基礎的方法 A0、A1 是明顯快過以 q 遞迴為基礎的方法 C0、C1。

就採用四元數 Zernike 矩計算的速度評估，還需要和與他們相近的 Zernike 矩計算方法，一起比較。一般評估，如果是採用 Zernike 矩分別處理 RGB 三個頻道，計算時間應該是計算灰階圖的三倍時間。先就未使用 AVX 256-bit 向量暫存器的法 A0、法 C0 與相對應的法 R、法 Q 觀察計算時間比值 $A0/R$ 、 $C0/Q$ 。這些比值全部明顯小於 3，顯示採用四元數 Zernike 計算的方式，是有效的。以較慢 q 遞迴方法為例，特別是最高階數介於 100 至 450 之間，執行時間 $C0/Q$ 比值是介於 1.712 與 1.895 之間。反觀快速的四項次遞迴方法， $A0/R$ 比值較大，是介於 2.469 與 2.621 之間。這應該是四元數的算術計算總量，在四項次遞迴方法 A0，相對較慢的 q 遞迴方法 C0，有比較大的比重。

在採用 AVX 256-bit 向量暫存器計算的法 A1、C1，速度上有非常明顯的提昇。以執行時間比值 $A0/A1$ 為例，全數在 1.911 以上；而執行時間比值 $C0/C1$ 在階數小於等於 400 時，也有 1.570 以上的水準。利用 AVX 256-bit 向量暫存器加速計算，確實有相當的成效。

與相對應的 Zernike 矩計算方法比較，比值 $A1/R$ 在最高階數 100 以上，全數小於等於 1.296，也就是說，計算四元數 Zernike 矩只要將原計算 Zernike 矩時間，再加上額外不到 29.6% 的計算時間，就能夠達成。這種效應，在於較慢方法的 C1 更是明顯。執行時間比值 $C1/Q$ 在最高階數 250 與 300，甚至是 1.087 與 1.095，這是不過 8.7% 與 9.5% 的額外時間，就可以完成。利用 AVX 256-bit 向量暫存器向量化計算四元數 Zernike 矩，所能達到的成效，確實驚人。

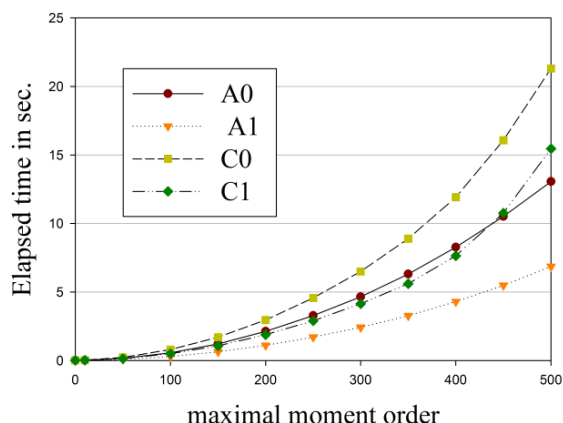


圖 7.法 A0、A1、C0、C1 計算四元數 Zernike 矩在最高階數 0-500 所花的平均時間。

表 4.不同方法針對 512x512 測試圖計算四元數 Zernike 矩在最高階數 50-500 所花得平均時間與相對應 Zernike 矩方法執行時間的比較

方法 階數	A0	A1	C0	C1	A0/A1	C0/C1	R	Q	A0/R	A1/R	C0/Q	C1/Q
50	0.160	0.082	0.222	0.135	1.941	1.642	0.061	0.109	2.621	1.351	2.034	1.239
100	0.561	0.288	0.794	0.500	1.949	1.589	0.218	0.423	2.573	1.321	1.876	1.181
150	1.210	0.633	1.698	1.071	1.911	1.586	0.497	0.923	2.434	1.274	1.840	1.160
200	2.129	1.103	2.952	1.883	1.930	1.568	0.868	1.648	2.453	1.271	1.792	1.143
250	3.274	1.702	4.559	2.895	1.924	1.575	1.319	2.663	2.482	1.290	1.712	1.087
300	4.650	2.415	6.498	4.139	1.925	1.570	1.872	3.779	2.484	1.290	1.719	1.095
350	6.304	3.263	8.887	5.597	1.932	1.588	2.540	5.072	2.482	1.285	1.752	1.104
400	8.266	4.294	11.918	7.636	1.925	1.561	3.320	6.592	2.490	1.293	1.808	1.158
450	10.512	5.477	16.074	10.756	1.919	1.495	4.234	8.483	2.483	1.294	1.895	1.268
500	13.065	6.856	21.316	15.463	1.906	1.379	5.292	10.857	2.469	1.296	1.963	1.424

時間單位：秒

五、結論

本研究對於四元數 Zernike 矩計算的探討是多面向的，除了理論性的探討，也涉及到演算法實作的議題，其結果可歸納如下：

- (1)我們對於四元數 Zernike 矩的一般性理論發展，給出一系列的證明，奠下四元數 Zernike 矩的理論基礎，並提出四元數 Zernike 矩合理的不變量。
- (2)發展一系列計算四元數 Zernike 矩的方法。這些方法在於高階矩的計算，也透過實驗結果證實非常有效，不僅是快速，而且精準。以 512x512 像素的 RGB 測試圖 Tiffany 為例，計算到 500 階的還原圖，其誤差 NMSE 僅 0.0017582。

- (3)透過四元數代數結構的向量計算，有效地使用 AVX 256-bit 向量暫存器，整合至所發展四元數 Zernike 矩演算方法，以 512x512 像素的 RGB 圖為例，計算到 500 階以降所有矩，僅需要 6.8563 秒。
- (4)與其他演算方法比較，不僅是在還原圖的品質，或是執行速度上，我們所提議的方法，都是明顯勝出。

參考文獻

- [1] Zernike, F., "Beugungstheorie des Schneidenverfahrens und seiner verbesserten Form, der

- Phasenkontrastmethode,” *Physica*, 1, pp. 689-704, 1934.
- [2] Abramowitz, M., and Stegun, I.A., “Orthogonal Polynomials, Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, ” 9th printing, Dover, New York, pp. 771-802, 1972.
 - [3] Teague, M., “Image analysis via the general theory of moments,” *J. Opt. Soc. Am*, 70, pp. 920-930, 1980.
 - [4] Pawlak, M., “On the reconstruction aspect of moment descriptors,” *IEEE Transactions on Information Theory*, 38, pp. 1698-1708, 1992.
 - [5] Pawlak, M., and Liao, S. X., “On the Recovery of a Function on a Circular Domain,” *IEEE Transactions on Information Theory*, 48, pp. 2736-2753, 2002.
 - [6] Belkasim, S. O., Shridhar, M., and Ahmadi, M., “Shape recognition using Zernike moments invariants,” in *Proceedings of 23rd Annual Asilomar Conference on Signals Systems and Computers*, pp. 167-171, 1989.
 - [7] Ghosal, S., and Mehrotra, R., “Detection of composite edges,” *IEEE Transactions on Image Processing*, vol. 1, pp. 14-25, 1994.
 - [8] Iskander, D. R., M. J. Collins, and B. Davis, “Optimal modeling of corneal surfaces with Zernike polynomials,” *IEEE Transactions on Biomedical Engineering*, vol. 48, no. 1, pp. 87-95, 2001.
 - [9] Farokhi, S., S. M. Shamsuddin, U. U. Sheikh, J. Flusser, M. Khansari, and K. Jafari-Khouzani, “Near infrared face recognition by combining Zernike moments and undecimated discrete wavelet transform,” *Digital Signal Processing*, vol. 31, pp. 13-27, 2014.
 - [10] Elshoura, S. M., and Megherbi, D. B., “Analysis of noise sensitivity of Tchebichef and Zernike moments with application to image watermarking,” *Journal of Visual Communication and Image Representation*, vol. 24, pp. 567-578, 2013.
 - [11] Singh, C., and Pooja, “Improving image retrieval using combined features of Hough transform and Zernike moments,” *Optics and Lasers in Engineering*, vol. 49, pp. 1384-1396, 2011.
 - [12] Nigam, A., and Gupta, P., “Designing an accurate hand biometric based authentication system fusing finger knuckleprint and palmprint,” *Neurocomputing*, vol. 151, pp. 1120-1132, 2015.
 - [13] Mindru, F., Tuytelaars, T., Gool, L. V., and Moons, T., “Moment invariants for recognition under changing viewpoint and illumination,” *Computer Vision and Image Understanding*, vol. 94, pp. 3-27, 2004.
 - [14] Chan, Y. K., and Chang, C. C., “A color image retrieval method based on color moment and color variance of adjacent pixels,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 16, pp. 113-125, 2002.
 - [15] Lang, S., *Algebra*, 3 ed., Springer-Verlag New York, 2002.
 - [16] Xu, Yi, Y., L.C., Xu, H.T., Zhang, H., and Nguyen, T., “Vector Sparse Representation of Color Image Using Quaternion Matrix Analysis,” *IEEE Transactions on Image Processing* vol. 24, no. 4, pp. 1315 - 1329, 2015.
 - [17] Hitzler, E., “Quaternion Fourier Transform on Quaternion Fields and Generalizations,” *Advances in Applied Clifford Algebras*, vol. 17, no. 3, pp. 497-517, 2007.
 - [18] Chen, B. J., Shu, H. Z., Zhang, H., Chen, G., Toumoulin, C., Dillenseger, J. L., and Luo, L. M., “Quaternion Zernike moments and their invariants for color image analysis and object recognition,” *Signal Processing*, vol. 92, no. 2, pp. 308-318, 2012.
 - [19] Kuipers, J. B., *Quaternions and rotation Sequences: a Primer with Applications to Orbits, Aerospace, and Virtual Reality*, Princeton University Press, 1999.
 - [20] Prata, A., and Rusch, W. V. T., “Algorithm for computation of Zernike polynomials expansion coefficients,” *Applied Optics*, vol. 28, pp. 749-754, 1989.
 - [21] Shakibaei, B. H., and Paramesran, R., “Recursive formula to compute Zernike radial polynomials,” *Optics Letters*, vol. 38, pp. 2487-2489, 2013.
 - [22] Singh, C., and Upneja, R., “Fast and accurate method for high order Zernike moments computation,” *Applied*

- Mathematics and Computation vol. 218, no. 15, pp. 7759-7773, 2012.
- [23] Kintner, E. C., "On the mathematical properties of the Zernike polynomials," *Optica Acta: International Journal of Optics*, vol. 8, pp. 679-680, 1976.
- [24] Chong, C.-W., Raveendran, P., and Mukundan, R., "A comparative analysis of algorithms for fast computation of Zernike moments," *Pattern Recognition*, vol. 36, pp. 731-742, 2003.
- [25] Chong, C.-W., Raveendran, P., and Mukundan, R., "An efficient algorithm for fast computation of pseudo-Zernike moments," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 06, pp. 1011-1023, 2003.
- [26] Deng, A.-W., Wei, C.-H., and Gwo, C.-Y., "Stable and Fast Computation of High-Order Zernike Moments Using a Recursive Method," *Pattern Recognition*, vol. 56, pp. 16-25, 2016.
- [27] Singh, C., and Walia, E., "Fast and numerically stable methods for the computation of Zernike moments," *Pattern Recognition*, vol. 43, pp. 2497-2506, 2010.
- [28] Deng, A.-W., Wei, C.-H., and Gwo, C.-Y., "Algorithms for Computing Zernike Moments and Image Reconstruction in Parallel Process," in *International Conference on Information Science and Control Engineering*, Shanghai, China, 2015.
- [29] Deng, A.-W., and Gwo, C.-Y., "Parallel Algorithms for Computing Zernike Moments." in *International Congress on Engineering and Information (ICEAI 2016)*, Osaka, Japan, pp. 1011-1019, 2016.
- [30] Deng, A.-W., and Gwo, C.-Y., "Parallel Computing Zernike Moments via Combined Algorithms," *The SIJ Transactions on Computer Science Engineering & its Applications (CSEA), The Standard International Journals*, vol. 4, no. 2, pp. 11-19, 2016.
- [31] 鄧安文、郭枝盈，四項次遞迴平行計算 Zernike 矩，*健行學報*第 36 卷，第 3 期，第 1-24 頁，2016。
- [32] intel Intrinsic Guide,
<https://software.intel.com/sites/landingpage/IntrinsicGuide/>.
- [34] Papakostas, G. A., Y. S. Boutalis, C. N. Papaodysseus, and D. K. Fragoulis, "Numerical error analysis in Zernike moments computation," *Image and Vision Computing*, vol. 24, pp. 960-969, 2006.
- [33] Janssen, A. J. E. M., and P. Dirksen,