中華民國 第49屆中小學科學展覽會

作品說明書

高中組　數學科

佳作

040412

碎形萬花筒

學校名稱：國立馬祖高級中學

作者：	指導老師：
高一 陳亭妤 高三 林艷麗 高一 邱晨芳 高一 洪育群	胡裕仁

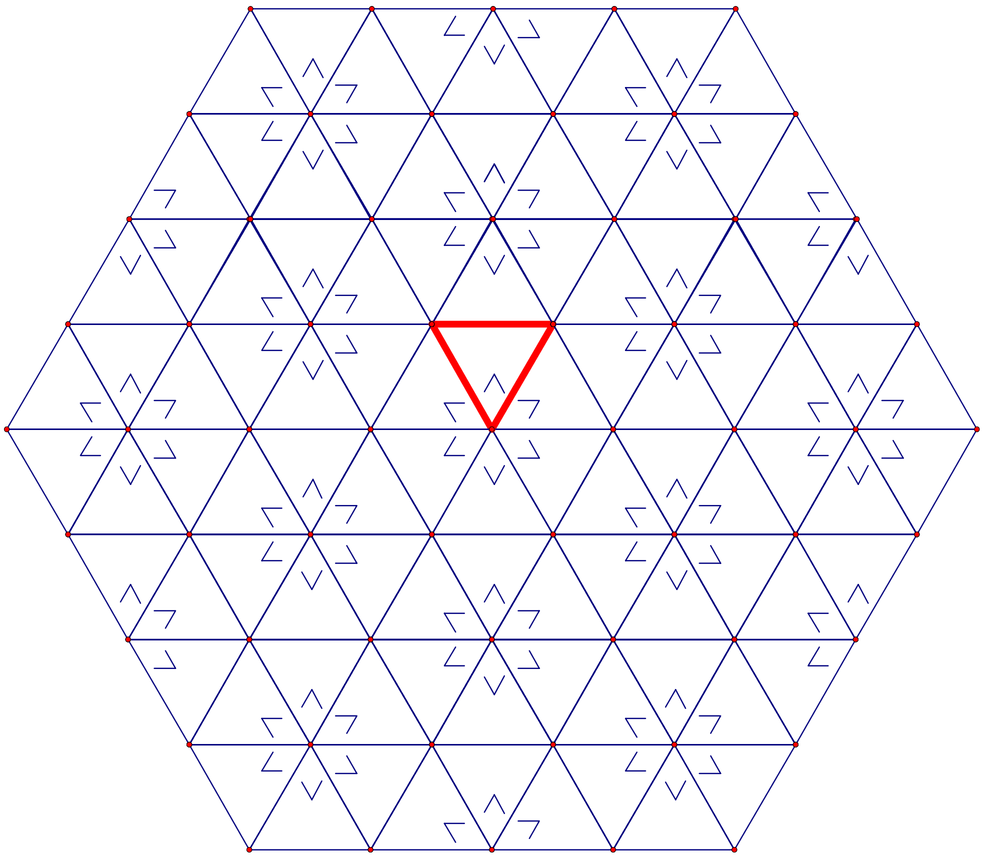
關鍵詞：碎形　、　渾沌　、　遍歷轉換

摘要

本文旨在探討萬花筒中的美麗圖形的現象原理。過程中透過碎形理論及迭代函數系統進行分析，並對其美麗圖形的建構組成，找出背後的數學理論。

文中在制式萬花筒實物模型及軟體的模擬中，找出萬花筒中圖形背後的數學函數對映關係。發現其結果是依循三稜鏡所構成的頂點為中心，並以三頂點中心等間隔鏡射生成的平面，複製出對稱的圖形進而產生六芒星，並在三稜鏡面上所生成的鏡射平面上，再進行以六芒星為基礎圖複製的無理旋轉而成，最終使鏡射平面上佈滿稠密的六芒星。

而該六芒星中心點會以三稜鏡的三個頂點為圓心的夾角，等間距地生成於起始位置周圍，而轉動萬花筒所產生的美麗圖形，正是基本圖在圓筒內進行遍歷的鏡射所生成，結果如下。



壹、研究動機

本組有成員去親戚家時，看到小朋友在玩萬花筒，自己也因好久沒玩而向他借來把玩。無意間對那似曾相識的對稱圖樣感到驚艷，此時腦海中立刻湧現不久前學校老師在數學專題討論所提到碎形的美麗圖樣，基於對此種規律又對稱的美麗圖樣感到好奇，發現有些和萬花筒類似，便引起我們對這個題材的研究興趣。

雖然萬花筒在現代科技發展昌盛的時代中，是一個不起眼又過時的小玩具；然其背後所潛藏的數學原理卻不像魔術方塊那般受人注意及研究。然而萬花筒發明者的智慧之高不但令人驚嘆，其被埋沒的價值與意義亦令人感到惋惜。

貳、研究目的

本研究期望能找出萬花筒中美麗圖形的形成原理，因此希望先行找出其背後數學方法來分析，並提醒現代人不要荒廢古人的智慧。文中借用碎形分析的相關方式，並結合遍歷理論及高中的矩陣數學，以萬花筒中三稜鏡為邊、三稜鏡交點為項點進行相關分析，藉以檢驗眼睛透過萬花筒所看到的美麗的圖形其實是一種渾沌現象。因此，本文以 Devancy 及 Li-Yorke 兩位數學家所定義的數學渾沌現象方法去分析解釋萬花筒的渾沌行為，並將研究結果設計出具有數學教育意義的多功能萬花筒以呈現更多美麗的鏡射圖像。

參、研究設備及器材

一、電腦壹台。

二、數學方程式符號編輯器 Equation、GSP、Word、Visual Basic、Visio、Photoshop 軟體各壹套。

三、彩色印表機一台。

四、不同種類萬花筒數筒、鏡片數面及色紙數張。

五、數位相機一台。

肆、文獻探討及研究方法

碎形(fractals)發展是1975年由IBM 研究所的Mandelbrot提出，它泛指一個外貌複雜的形體，但其結構具有尺度不變性(scaling invariance)，即自相似性(self-similarity)的圖形。透過電腦本身快速計算與繪圖功能，可將一個簡單的機制指令，經由不斷地迭代(iteration)，而產生一連串的分岔複製(bifurcation)，最後在電腦螢幕上出現一個複雜形體，這就是一個人工虛擬碎形。而數學碎形是指利用迭代函數系(iterated function system) 所獲得的碎形，例如Cantor 集、Sierpinski三角形（如圖 4-1）、Von Koch 雪花（如圖 4-2）…等等。數學碎形具有嚴格的自相似性， 而自然界也有很多碎形的實例，但是多半以隨機自相似(stochas-tic self-similarity) 呈現[3]。

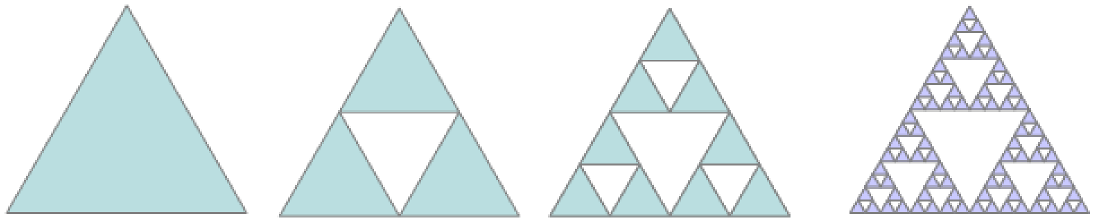


圖 4-1：Sierpinski 三角形

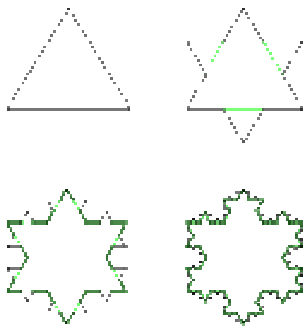


圖 4-2：Von Koch 雪花

自相似性碎形

泛指一個碎形的任意一個片段可以和整體相似。Mandelbrot為碎形定義：當是具有某種自相似性的集合[1]，因此自相似性是碎形所需的必要條件和一種普遍存在的特

徵。而此項特性，正好是萬花筒內鏡射出美麗圖形的一種共同存在的特性。

迭代函數系統 Iterated Function System (IFS)

IFS 主要用在二維空間碎形設計的一種方法。它利用壓縮映射原理，由簡單的構圖開始，經過幾回合的一系列的轉換而建立，如 Sierpinski 三角形（如圖 4-3）；一些**非碎形的構圖**及其它維度亦可使用此一方式設計（如圖 4-4）[7]，或附件(三)。

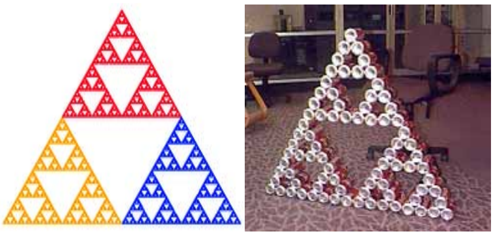
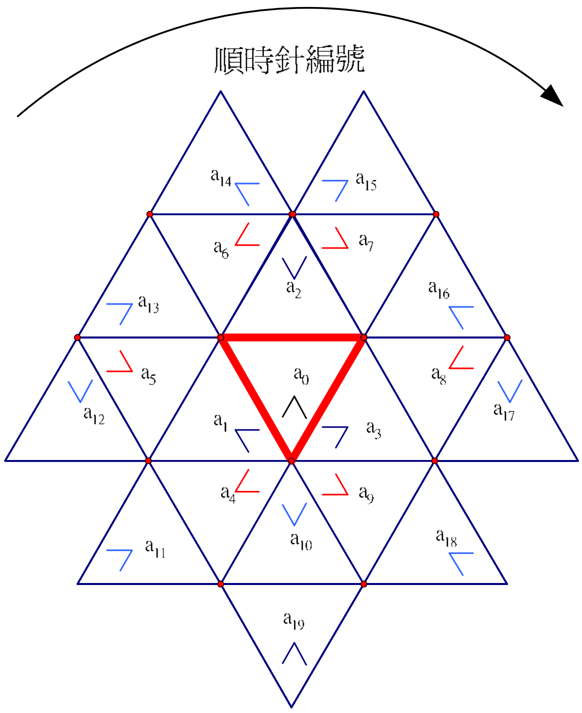


圖 4-3：IFS 設計的 <u>Sierpinski 墊圈</u>　圖 4-4：參考附件(三)由 IFS 設計的<u>電綿羊</u>

IFS 的數學原理<u>應用於萬花筒內鏡射圖形之設計</u>

若把自相似性集合看作是若干個相似映射的作用下的不變集合，則由我們研究結果如圖 4-5 及圖 4-6 所定義出萬花筒內鏡射的子圖其內部傳遞的集合關係結構。



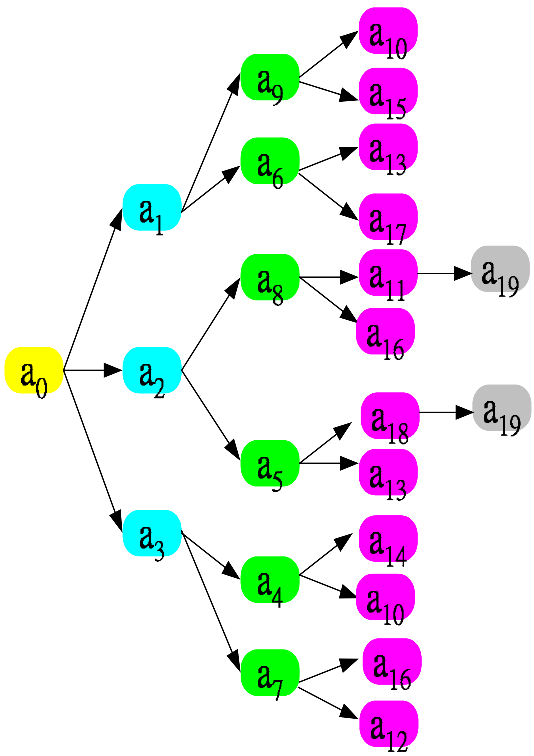


圖 4-5：3 次迭代生成的六芒星圖形集合　　圖 4-6：3 次迭代的數值傳遞

如同是由 a_0 至 a_{10} 經過三次迭代生成的一個完整的六芒星。端點完全對稱六芒星生成關係 $\{a_0\} \rightarrow \{a_1、a_2、a_3\} \rightarrow \{a_9、a_6、a_8、a_5、a_4、a_7\} \rightarrow \{a_{10}、a_{15}、a_{13}、a_{17}、a_{11}、a_{16}、a_{18}、a_{13}、a_{14}、a_{10}、a_{16}、a_{12}\} \rightarrow \{a_{19}\}$，其中三層箭頭代表三次迭代作用。

本文改良傳統分析數學碎形的壓縮映射方式，將原本存在碎形中的壓縮映射的關係性略作修改，並限定壓縮比為定值 1，以避免圖 4-7 至圖 4-11 的伸縮現象發生，而三稜鏡邊長縮小，其對映生成的基本映射圖將會在平面上呈現更多及更緊緻，最終將

圖 4-7：邊長 30 像素

圖 4-8：邊長 25 像素

圖 4-9：邊長 20 像素

圖 4-10：邊長 15 像素

圖 4-11：邊長 10 像素

完整接近覆蓋整個鏡射生成的平面。因此將壓縮映射改為**蓋射映射**，可簡化分析出**萬花筒內鏡射圖形定理 1**。

> **萬花筒內鏡射圖形定理 1**：設 $\mathrm{f}=\{f_1,\ldots,f_m\}$ 是在 m 個 $\mathbb{R}^n$ 中有蓋射映射，則一定存在一個唯一的閉集合 F，使得 $F=\bigcup_{j=1}^{m} f_j(F)$。

定理 1 說明：令 $S(E)=\bigcup_{j=1}^{m} f_j(E)$ 這裡（$\forall\ E\subset\mathbb{R}^n$），則定理 1 的結論可寫作 $F=S(F)$，即 F 是 S 的不變集。假定 F 是關於蓋射映射組 $\mathrm{f}=\{f_1,\ldots,f_m\}$ 的非空不變集，此時可定義 F 的相似性維數，可完整覆蓋整個平面。

若假定 F 中的映射中，只限於位移與相似性變換的複合，且滿足條件：$H^s(f_i(F)\cap f_j(F))=0$、$(i\neq j)$，則稱 F 是自相似集。而若無此條件，則代表在某個交

集 $f_i(F)\cap f_j(F)$ 中失去自相似性的特徵[1][7]。由於在碎形定理中的豪斯多夫維數計算式 $d=-\lim\limits_{R\to 0}\dfrac{\log(N)}{\log(R)}$，因受壓縮比的介於 $(0,1)$ 開區間的關係所以 $R\to 0$，但本研究在定理 1 時，已將壓縮比修正為定值 1（即 a=1），因此得出**萬花筒內鏡射圖形定理 2**。

萬花筒內鏡射碎形定理 2：若 F 是一個自相似集，則它的相似性維數等於豪斯多夫維數，即 $d=-\lim\limits_{R\to a}\dfrac{\log(N)}{\log(R)}$。

定理 2 說明：豪斯多夫維可以給一個任意複雜的點集合賦予一個維度，而對於簡單的幾何目標比如線、長方形、長方體等，豪斯多夫維通常等於幾何維度或拓撲維度。一般來說，一個物體的豪斯多夫維可能會是一個非整數型態的有理數或者無理數。

例如：

1. 正方形：一個正方形由 9 個長寬都只有它三分之一的小正方形組成，那麼 $d=\dfrac{\log(9)}{\log(3)}$=2。
2. 科赫曲線(Koch-Curve)：科赫曲線的每一部分都由 4 個跟它自身比例為 1:3 的形狀相同的小曲線組成，那麼它的豪斯多夫維數為，是一個無理數為 $d=\dfrac{\log(4)}{\log(3)}=1.261859...$，為一個無理數。

直觀上來說，一個集合的維數是描述這個集合中一點所需的獨立參數的個數，例如：要描述一個平面裡的一點，需要兩個坐標 X 和 Y，那麼平面的維數便是 2，最接近這個想法的數學模型是拓撲維度，因此可見拓撲維度必是自然數。但拓撲維度在描述某些不規則的集合，例如：碎形的時候遭遇到了困難，而豪斯多夫維則可重新描述該種集合的工具。設萬花筒為一單位圓筒內進行不斷的旋轉，而每一次生成的六芒星乃是以一無理旋轉且等距離及等角度模式在生成。

豪斯多夫外測度：令 (X,d) 為一個度量空間，E 為 X 的一個子集（如同圖 4-5 中的 a_i 生成的六芒星），定義 $H^s_\delta(E)=\inf\left\{\sum_{i=1}^{\infty} diam(A_j)^s\right\}$，且 E 能被六芒星 $(A_j)_k$ 覆蓋，則 E 的豪斯多夫外測度被定義為：$H^s(E)=\lim\limits_{\delta\to\infty} H^s_\delta(E)$。此點剛好滿足三稜鏡所不停反射生成的無限大的平面，而豪斯多夫維的嚴格的定義：$\dim_H E=\inf\{s:H^s(E)=0\}$

$=\sup\left\{s:H^{s}(E)=\infty\right\}$ 正巧滿足 a_0 至 a_∞ 的無窮子圖，如圖 4-15、圖 4-16[1][8]。

遍歷定理 3：對於機率空間的保測變換 φ，從一個正測度集合中出發的幾乎所有軌道都要無窮多次地返回這一集合。

本文參考交通大學陳明璋教授所提的方法[4]，加以改良來分析萬花筒中若干幾何圖案方式，並以此直接當作萬花筒基本圖碎形產生的規則；再以 Visual Basic 模擬的方式進行點定位與線複製，且探討了不同邊長時對映生成的圖形關係（如圖 4-7 至 4-11），並使其進行不斷的迭代而產生碎形的美麗圖案變化（如圖 4-12 至 4-16）。

圖 4-12：三邊形

圖 4-13：四邊形

圖 4-14：五邊形

圖 4-15：六邊形 1

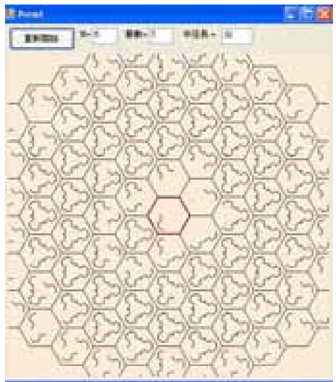


圖 4-16：六邊形 2

點定位複製及迭代

一個物件安置於特定的位置必須知道幾個資訊：(1) 物件的中心點(2) 物件的寬度及高度(3) 物件的方位角。只要能計算出這三個資料就可以了。因此只要知道迭代過程所產生的自相似物件的資料，就可以將自相似物件調整安置於預定位置。

點定位複製

中心點複製是以模型為基礎，預計在目標物件上建立一相同關係的複本。其中必

須掌握的資訊包含模型的寬度、長度及方位角，目標物件的中心點、寬度、長度及方位。只要以目標物件的中心點、寬度、長度及方位角設為複本的的中心點、長度及寬度以及方位角位即可。

1. 模型：是圖案組合，一般都使用同一種對稱性較高的物件來構圖。模型的中心點當作定位座標的基準點，其長及寬為大小的基準，其基準方向為零度。

2. 目標結構：由許多物件構成，每一物件自成系統，其長、寬及方向就是複本的長、寬及方向。

3. 複本：是模型的相似圖案，目標結構中各物件的長、寬及方向就是複本的長、寬及複本中的各組成物件，可被當作新的標的物件。

構圖步驟：（以 Sierpinski 三角形為例，進行三次迭代）

1. 首先建立模型及與標的結構，標的結構由許多物件構成，如圖 4-17；

2. 依據標的物中各物件的長、寬、方向，分別將圖案模型複製到物件的中心點上，如圖 4-18。

3. 重複前一步驟，就可以達到迭代的效果，如圖 4-19 及圖 4-20。

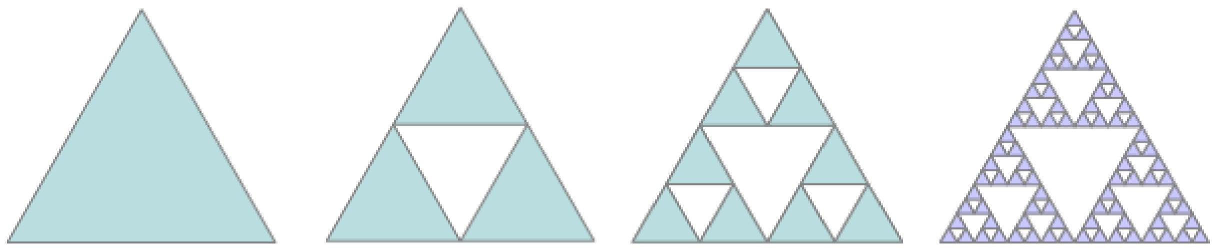


圖 4-17：初始值　　圖 4-18：一次迭代　　圖 4-19：二次迭代　　圖 4-20：三次迭代

Sierpinski 三角形點複製構圖

線定位複製及迭代

線定位複製：線定位複製是以基準線與模型之間的相對關係為基礎，預計在目標的線段上建立一相同關係的複本，已知的資訊包含以基準線與模型之間的相對關係，目標線段的長度及方位，可以計算得到複本的中心點、長度、寬度以及方位角。

因此本研究中修正了陳教授[4]若干線複製的概念，利用面蓋射的映射關係，以基準線與模型之間的相對關係進行蓋射對映，以符合萬花筒在多面鏡組中，進行重複反射的圖案生成複製原理，修正內容如下：

1. 模型：是圖案及線段的組合，圖案一旦被複製之後就不會改變，因此沒有迭代線要被精緻化，即壓縮比不會介於 0 到 1 之間而是固定為 1。

2. 基準線：基準線是一線段，它與模型共存，基準線與模型位置的相對關係，充當在複製時的基準。

3. 目標結構：標的結構由線段構成，每一線段自成一定位系統，基準線與模型之間的**相對**關係，就是初始線段與複本的關係，而基準線與模型之間的**位置**關係就是標的結構中線段與複本的位置關係。

4. 複本：是模型的全等圖案，其長與寬的大小相同，是基準線與對應初始值的比；其位置也是以基準線與模型之相對關係為基準。它們之間的關係用類比關係來呈現最為恰當－基準線：模型=目標物中的線段：複本。

構圖步驟：

1. 首先設計基準線、模型及以標的結構，如圖 4-21。

2. 以基準線與模型具備相對的關係為基準，依據標的結構中各線段的長度及方位，複製一份複本，放置於該線段的相對位置，如圖 4-22 及，如圖 4-23。

圖 4-21：初始值

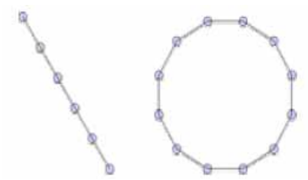


圖 4-22：一次迭代

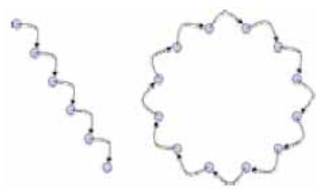


圖 4-23：二次迭代

以線複製齊一的弧

伍、研究過程及結果

標準制式萬花筒是由三片鏡子圍成封閉式的正三角形，再利用一些事先準備好的色紙充當底圖，供鏡子進行折射與反射產生的美麗重複的圖形。由已完成的萬花筒成品，可以發現它美麗多樣化的圖形來自筒內三片鏡子，即正三角形頂點與邊所構成的**點定位及線複製**兩種變化出多端的結果。而由圖 5-1 至圖 5-7 可以看出透過不同長度的邊長線段所呈現出的鏡射結果。

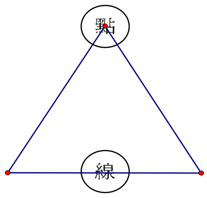


圖 5-1：制式萬花筒內部分析

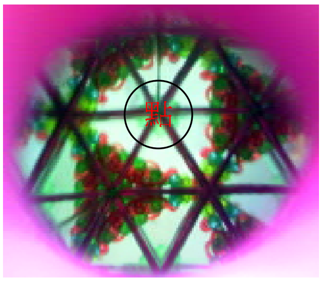


圖 5-2：萬花筒內部點定位分析

圖 5-3：萬花筒內部線複製分析

圖 5-4：全等自相似點定位及線複製圖

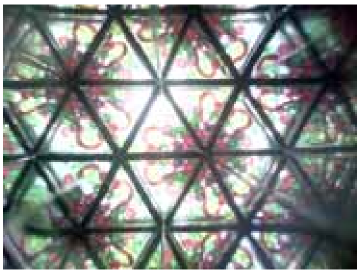


圖 5-5：全等自相似點定位及線複製圖

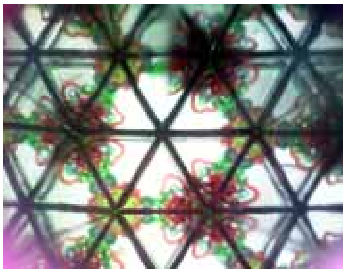


圖 5-6：全等自相似點定位及線複製圖

圖 5-7：全等自相似點定位及線複製圖

經我們利用 GSP 繪圖分析上列結果，假設下圖 5-8 中為初始值基礎圖 a_0，紅色部分為

一標準的萬花筒的三片鏡子。若鏡子內部有一「∧」符號當作萬花筒內部複製的參考基礎圖a_0=1∧，經由多次的點定位及線複製迭代鏡射，可得出多個如圖 5-11 的基礎圖

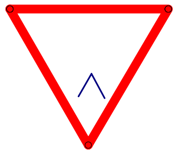


圖 5-8：初始值a_0=1∧

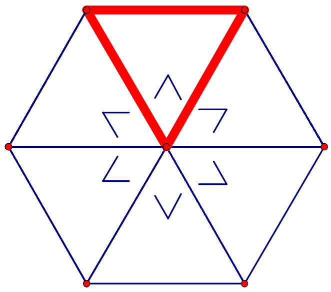


圖 5-9：完整 3 次迭代鏡射結果

a_n、$n \geq 3$的結果。由於萬花筒內鏡子實物本身鏡射的限制，所以 GSP 中無法一次對基礎圖直接進行點定位及線複製，即不可能第一次鏡射就產生圖 5-10 的結果（事實上圖 5-10 最少要三次鏡射複製，如圖 4-5 及圖 4-6）。

因此透過修正複製方法，每次同時對正三角形的原始三邊線段進行線複製，再利用 GSP 程式的模擬，最後再以 Visual Basic 程式來開發實際模擬的結果；發現基礎圖將分別由圖 5-11 轉變到 5-14 的結果，也就是說每次迭代都會產生新的基礎圖a_n，$n \in N$。

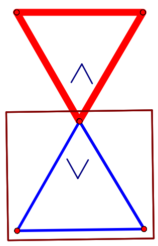


圖 5-10：三次以上鏡射成像

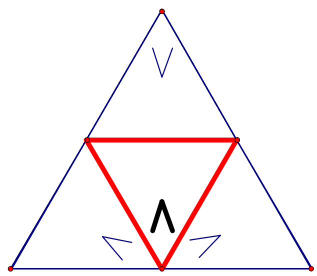


圖 5-11：一次迭代基礎圖

a_1=1∧+3∧

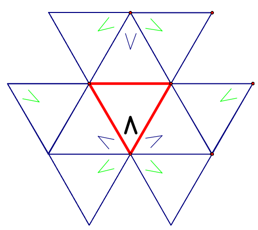


圖 5-12：二次迭代基礎圖

a_2=1∧+3∧+6∧

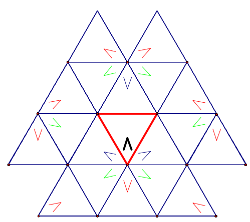


圖 5-13：三次迭代基礎圖

a_3=1∧+3∧+6∧+9∧

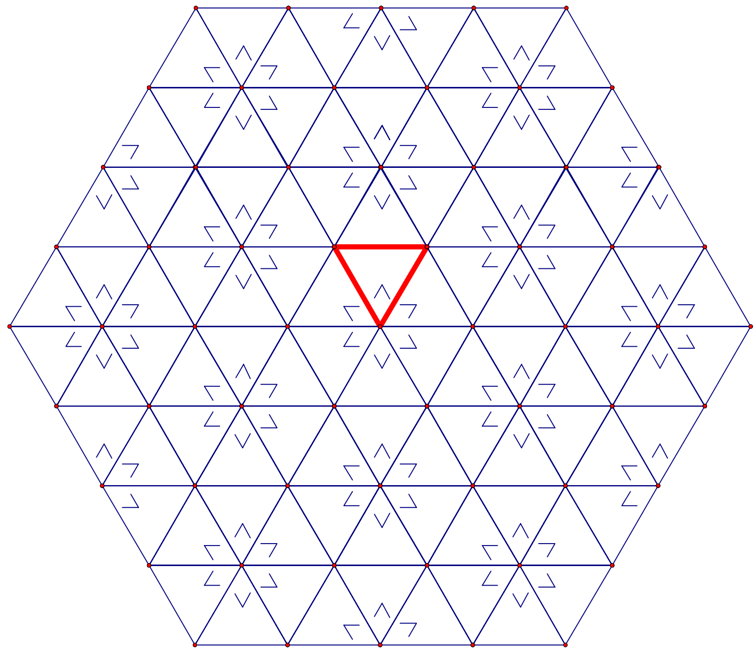


圖 5-14：多次迭代基礎圖 $a_n = 1 + \frac{3}{2}n(1+n)$ ∧

其後，我們逐步以圖 4-5 及 4-6 及 5-10 至 5-14 的分析探討，並利用點定位及線複製的概念，成功開發出程式本身的設計流程（如圖 5-15），相關程式碼列入附件(二)參考。

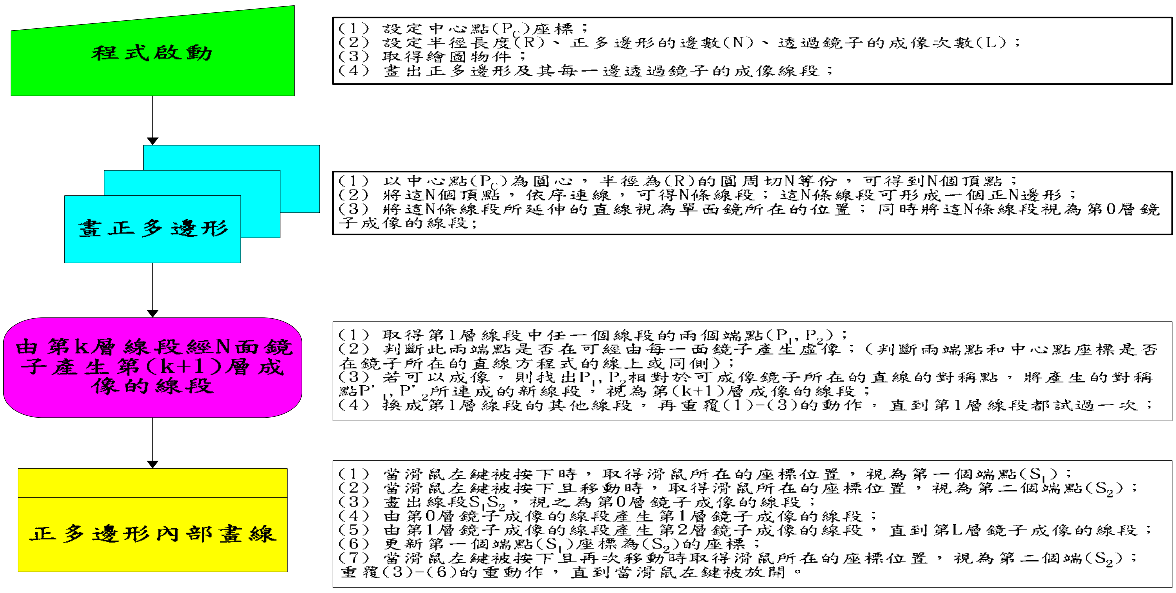


圖 5-15：萬花筒 VB 程式流程圖

陸、討論

在自然科學及數學上都對渾沌有著不同的定義。Devancy 為渾沌現象下了一個數學定義如下[1]：

設 $f:A\to A$ 是集合 A 到自身的一個連續映射，其中 A 是 m 維歐氏空間中的一個區域。又假設 $E\subset A$ 是 f 的一個不變集合。若映射 $f:A\to A$ 在集合 E 上滿足下列性質：

（1）f 的周期點在 E 上是稠密的；（六芒星的中心反覆進行無理旋轉）

（2）f 的迭代軌道在 E 中具有拓樸傳遞性；（保測變換）

（3）f 的迭代軌道具有對初始值的敏感依賴性，則我們稱 f 在 E 上的迭代軌道的行為是渾沌的；（所有圖形的演變皆是受其初始位置所影響、如圖 4-7 至 4-11）

在我們所研究的對象中，發現萬花筒可經歷無數次的轉動複製（即：轉的圖形變換次數可細分之無限次），並在反覆的鏡射中生成新的六芒星且其中心位置延著鏡射平面進行等角度及等距離的無理旋轉，而每次角度的改變亦會產生相對位置二維歐氏空間的基本圖變換，恰滿足保測變換的遍歷轉換程序。最後當基本圖「∧」回到初始值位置時，此點正好表示所有美麗圖形的演變皆是受其初始位置所影響，剛好滿足 Devancy 為渾沌現象下的定義。

由圖 5-15 開發的程式，如圖 4-12 至圖 4-16，發現經多次迭代之後的美麗圖形，恰好是周期為三的渾沌圖形。因此利用碎形理論加以分析得出圖 5-14 的豪斯多夫維數等同其拓樸維數，故維數 $d=\dfrac{\log 9}{\log 3}$=2。本文將前文所呈現的一系列迭代圖形分析結果，參考翁義聰教授編號處理碎形方法[5]若予以設計出本研究的賦值標號，如圖 4-5 及圖 4-6，可知若給定萬花筒中的一個初始符號，經由萬花筒內部三面鏡子的反覆複製結果，並計算重複映射生成的鏡射圖，個數序列為一公比為 3 的等比序列；另 n 邊多面鏡組（$n>3$）時公比為 n。此點恰好滿足李天岩及 Li-Yorke 兩位教授的所提出的結果[2]，在一個標準制式萬花筒中內部三面鏡或多面鏡組中所產生的自映射迭代，恰好也是一個周期為三的渾沌現象，如圖 5-9。

本研究也嘗試利用 GSP 模擬若筒內的鏡子是 4 片、5 片、6 片時，亦滿足三次迭代之後，即可以生出一個美麗的對稱圖形，即周期為三在萬花筒內皆有同樣結果，VB 程式模擬亦然。因此如果把圖 5-8 中的 $a_0=1\wedge$，看成對三面鏡子的同時反射則存在一關係式為 $a_n=1+\frac{3}{2}n(n+1)\wedge$。而在上一節中我們發現每一個萬花筒，都是由一個基礎圖如圖 5-8 多次迭代生成完全自相似的基礎圖，正好滿足基礎圖對自己的複合映射 $f:A\mapsto A$，這裡 f 可看成基礎圖對三面鏡子同時反射一次的過程，A 為每次迭代後新產生成的三邊鏡射的基礎圖集合，再經過三次複合映射迭代之後，即 $p\mapsto f(f(f(p)))$，$\forall p\in A$ 可得出如圖 5-9 的以端點為中心的完全對稱六芒星，生成關係如圖 4-6 中的{a_0}→{a_1、a_2、a_3}→{a_9、a_6、a_8、a_5、a_4、a_7}→{a_{10}、a_{15}、a_{13}、a_{17}、a_{11}、a_{16}、a_{18}、a_{13}、a_{14}、a_{10}、a_{16}、a_{12}}，其中三層箭頭代表三次迭代作用。由此可知**<u>萬花筒其實是一個具有渾沌現象的數學迭代序列</u>**$\left\{f^n:A\to A\,|\,n=0,1,\ldots\right\}$，因每次函數過程皆為基礎圖對三面鏡子同時反射一次的過程，即集合到自身的恒同映射：$p\mapsto p$ 記作 $f^0:A\to A$，且在映射時又滿足 Devancy 所定義的渾沌數學原理，對每一次迭代過程中「∧」可看成函數內的自變數 $f(\wedge)$，因此對於∧∈A 而言$\left\{f^n(\wedge)\,|\,n=0,1,2,\ldots\right\}$為「∧」的軌道，若 $f(E)\subset E$ 而 $f:A\to A$ 是集合 A 的一個自映射，稱 A 的子集 E 是不變集，而若 $E\subset A$ 是映射 $f:A\to A$ 的一個不變集合，則 E 中任意一點 a 的軌道均落在 E 之中，經多次複映射迭代可得圖 5-14 的結果，恰巧表明映射軌道系統的拓樸傳遞性，即任意一點 $\wedge\in E$ 中經多次迭代之後可散落於各邊並呈現美麗又對稱的圖形。透過高中數學中的平移、鏡射矩陣分析理論，將迭代三次所的完整對稱圖形，如圖 5-9 的圖予以分析成美麗六芒星的相對位置，如圖 6-1 至圖 6-7。

圖形座標數學分析：

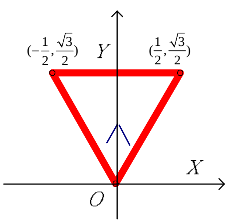


圖 6-1：基本座標點標示

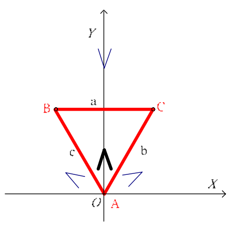


圖 6-2：一次迭代座標圖

將圖 5-8 予以座標化得圖 6-1 且三頂點座標分別為$(0,0)$、$(\frac{1}{2},\frac{\sqrt{3}}{2})$、$(-\frac{1}{2},\frac{\sqrt{3}}{2})$，即設每一邊長為 1 單位之正三角形。因此每次迭代即是將基礎圖對 a、b、c 三邊做一次鏡射複製，如圖 6-2，故對應的鏡射矩陣為 $\alpha=\begin{bmatrix}\cos\pi & \sin\pi\\ -\sin\pi & \cos\pi\end{bmatrix}$、$\beta=\begin{bmatrix}\cos(-\pi/6) & \sin(-\pi/6)\\ -\sin(-\pi/6) & \cos(-\pi/6)\end{bmatrix}$、$\gamma=\begin{bmatrix}\cos\pi/6 & \sin\pi/6\\ -\sin\pi/6 & \cos\pi/6\end{bmatrix}$，分別代表對 a、b、c 三邊的

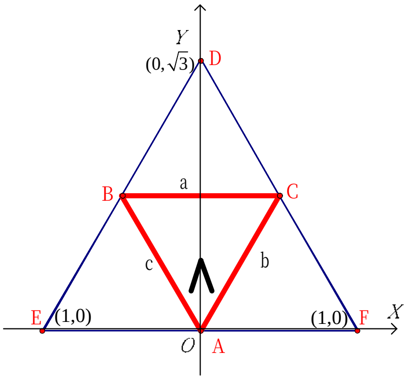


圖 6-3：邊長複製關係

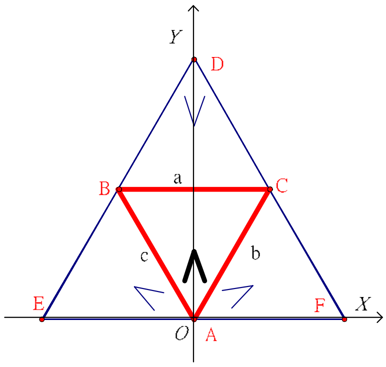


圖 6-4：一次迭代$a_0\to a_1$

鏡射矩陣，另每次鏡射迭代時也同時以原三邊長平移複製ΔABC至圖 6-3 ΔDEF，再複製基礎圖至新生成的另外三個三角形、如圖 6-4，此為一完整的一次迭代過程、即$a_0\to a_1$。

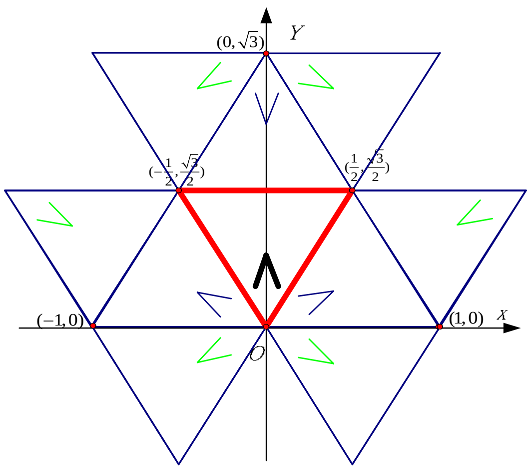


圖 6-5：二次迭代$a_1\to a_2$座標圖

同理可知，我們可以利用圖形的平移、鏡射來完成每三次複製過程，而每一個「∧」基本圖經三次迭代之後，便可生成一個六芒星圖，如圖 6-6。

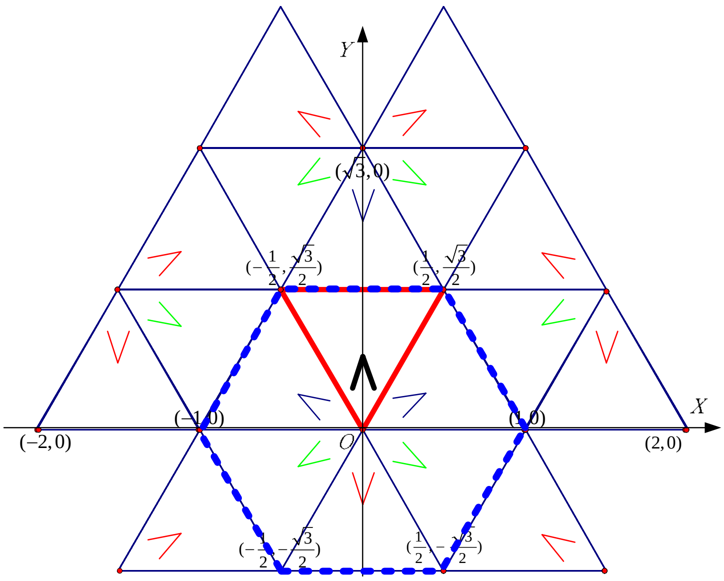


圖 6-6：三次迭代$a_2 \rightarrow a_3$座標圖複製出的六芒星圖

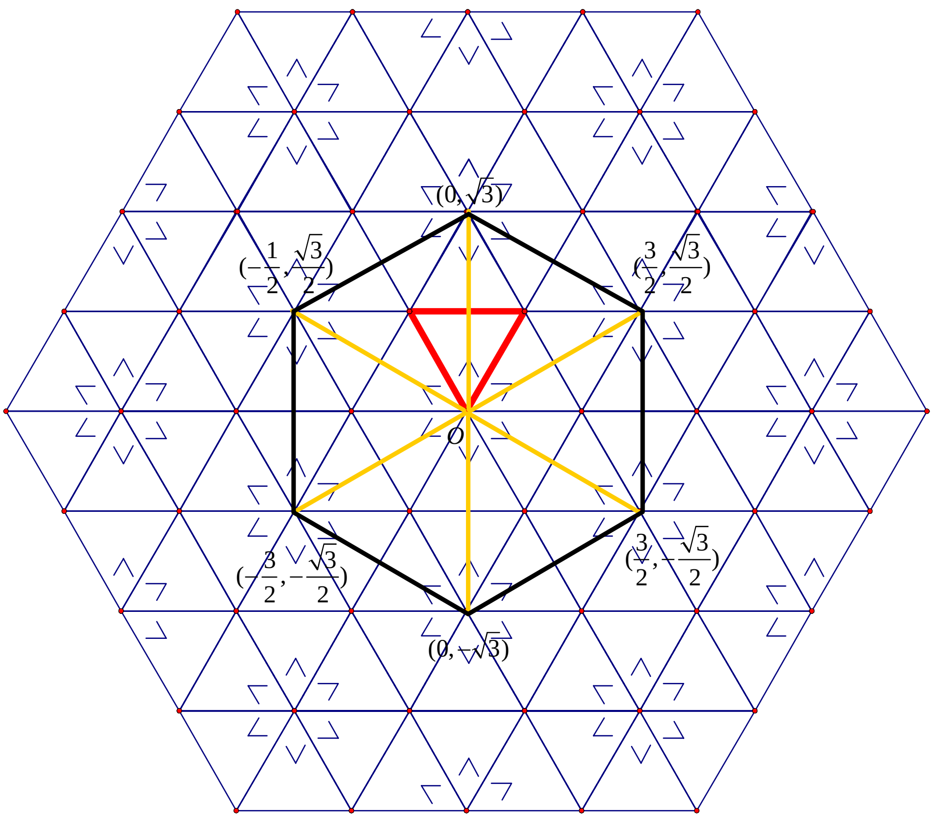


圖 6-7：多次迭代複製出的六芒星圖

而利用座標分析後，發現萬花筒中內部美麗圖形，是以三稜鏡六芒星頂點為圓心，$\sqrt{3}$鏡寬邊長半徑單位、60 度等分角，滿足遍歷分布於鏡射平面上，進一步再討論去對映關係，整個萬花筒的分析正好同時滿足 Devancy 及 Li-Yorke 兩位學者的渾沌數學

定義，例如：在具有「∧」的軌道中，萬花筒內的圖形最後都會回到原來的位置，可是卻無法計數到底要轉幾次，此現象剛好是基本圖產生的六芒星圖形中心經歷所有的軌道，最後基本圖回到原來起始的位置，但六芒星圖形卻無限地在鏡射平面複製生成，如同在實驗中一個保測變換「∧」回到原來位置，回到其起始狀態，而無理旋轉軌道、正稠密的佈滿整個圓周。當轉動萬花筒所產生的美麗圖形，正是基本圖在做圓筒內不停地進行複製基本圖映射的遍歷運動，此現象正好滿足機率空間上的一個保測變換「∧」稱為遍歷變換。

柒、結論

此次研究成功地分析萬花筒的數學關係，即為一渾沌現象和遍歷轉換，並分析出其圖形生成現象為一渾沌行為現象。雖然本研究主要以三面鏡子的標準萬花筒為主體，但亦以電腦模擬四、五、六面鏡子的迭代模式關係：

1. 萬花筒圖形生成關係滿足 Li-Yorke 教授所提的週期為三則渾沌的定義，因此未來萬花筒的玩具可朝向”正多邊形”鏡子的全反射來延伸實作出實物模型，以探討更多有趣的變化，是本研究可以進一步分析的方向。
2. 研究發現每次對不同階段的基礎圖進行複製的程序時，就如同在一軌道中進行拓撲傳遞。
3. 基礎圖則在一單位圓筒中反覆維持保測變換，在進行點定位及線複製運動時不斷透過六芒星圖形複製生成遍歷轉換及無理旋轉。
4. 總結：原來轉動萬花筒所產生的美麗圖形，就是利用基本圖在做圓筒內進行遍歷的映射而生成的多重反射效果。
5. 未來展望：設計出一新型萬花筒教學玩具，模型架構如圖 7-2 至圖 7-4（已申請專利中、如圖 7-1）。


地址：
收件人：[redacted]

經濟部智慧財產局
自行收納款項收據 收文日期：098年05月27日
NO. 2938083 開立日期：中華民國098年05月27日 098TP089511

繳款人	收入科目	金額	事由	備註
[redacted]	審查費	$3,000	收文文號：0982029889-0 案號：098209306 專利名稱：可調式多面鏡萬花筒	
合計 新台幣 參仟元整				

第一聯 收據（交繳款人收執）

機關長官 主辦會計 主辦出納 收件人員 柴語婷
第01號櫃台
經手人

圖 7-1：新型專利收據影本

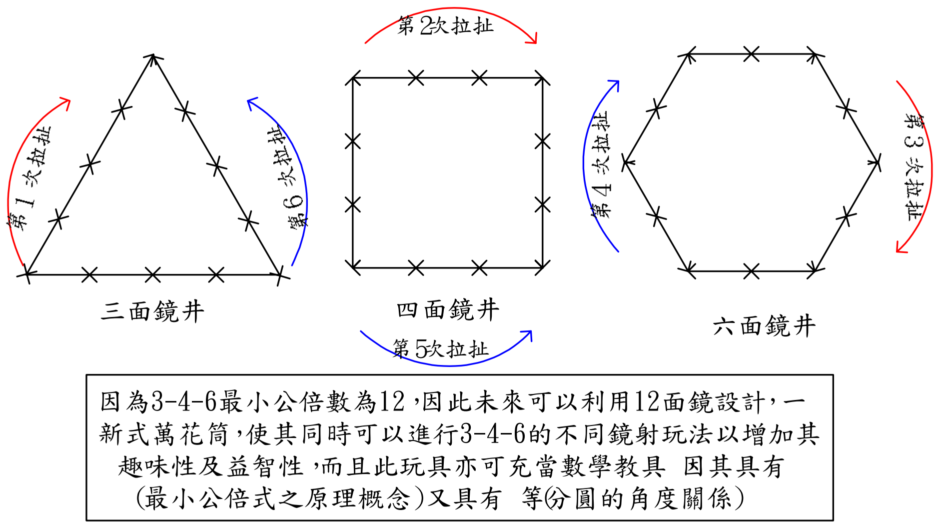


圖 7-2：可調式新型萬花筒稜鏡變換原理

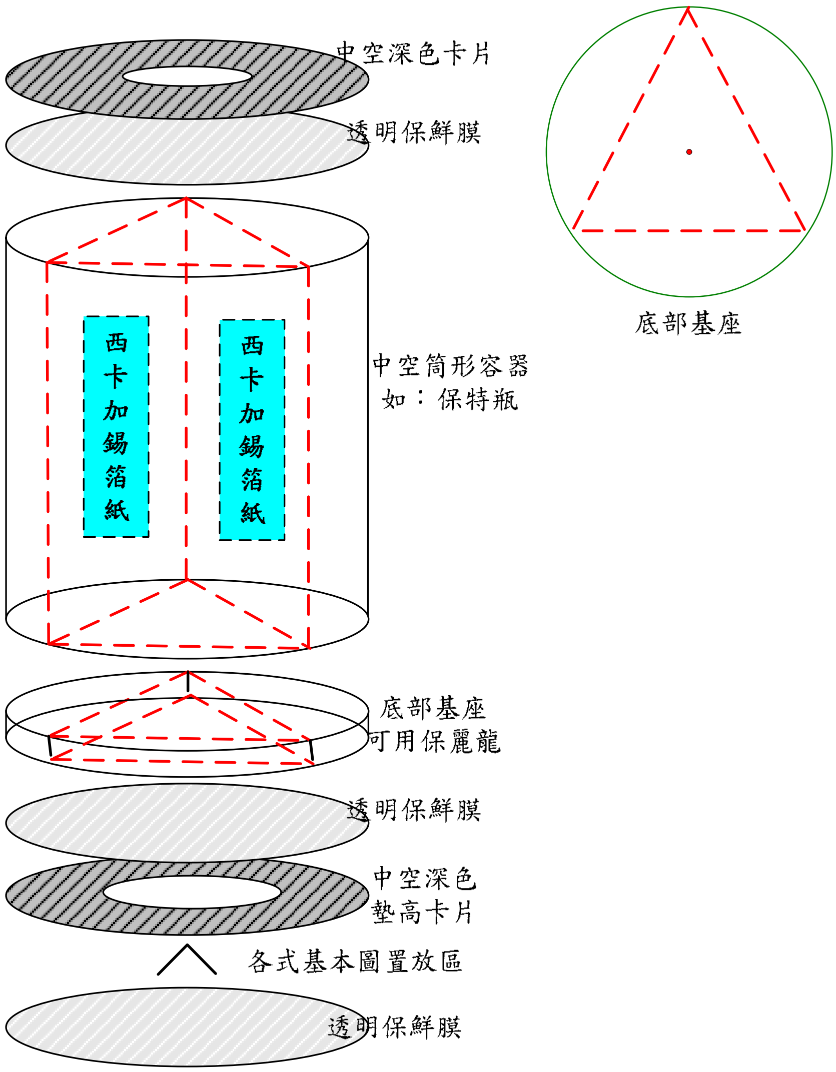


圖 7-3：傳統萬花筒機構圖

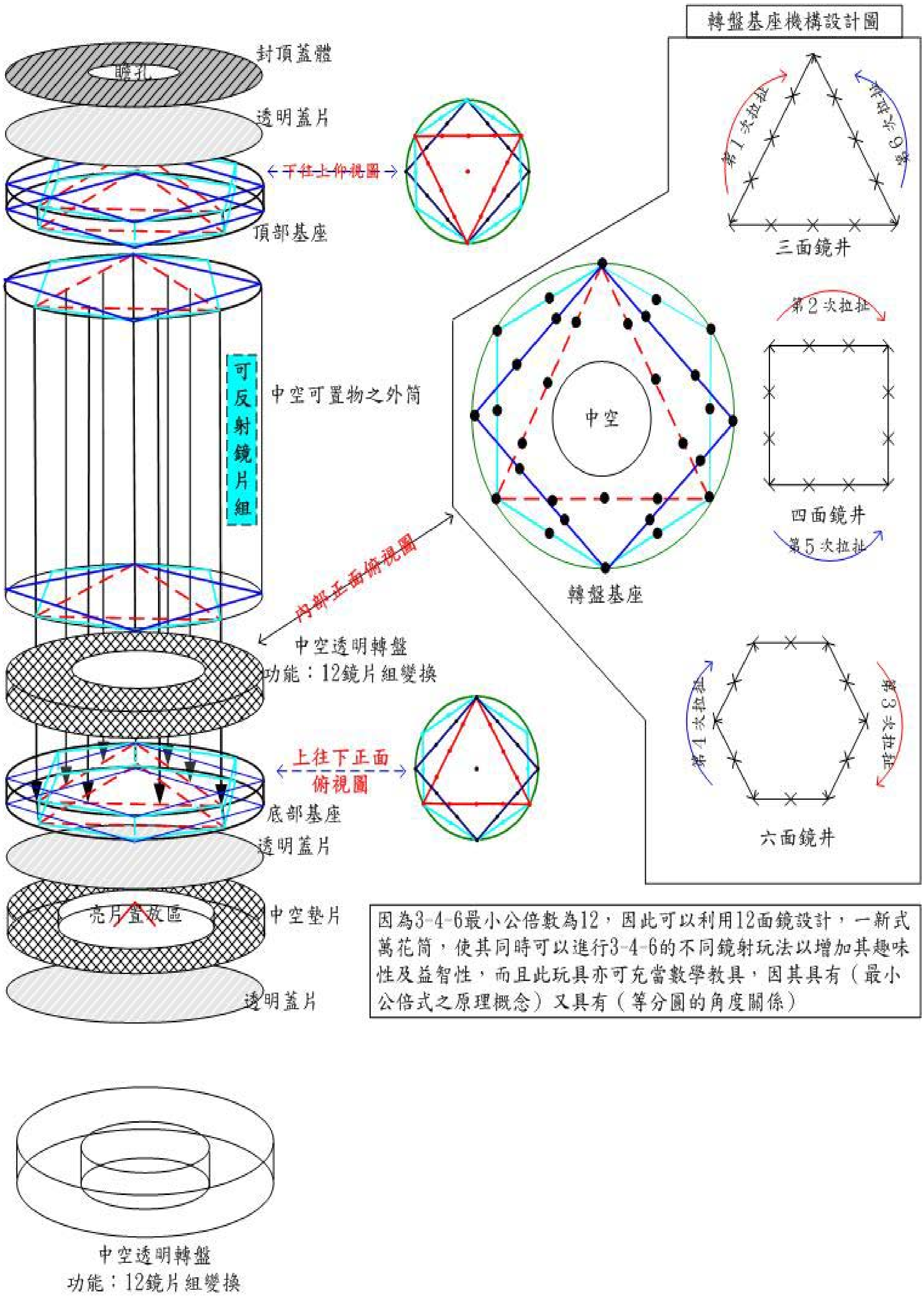


圖 7-4：可調式新型萬花筒機構設計圖

捌、參考資料及其他

[1] 李忠著，姜伯駒主編，迭代 渾沌 分形，科學出版社，2007。

[2] 李天岩，關於”Li-Yorke 渾沌”的故事，數學傳播第 12 卷第三期，1988。

[3] 謝南瑞，多重碎形，數學傳播第 25 卷第一期，2001。

[4] 陳明璋，創意碎形(一個新的碎形及圖像設計方法-視覺法)，2004。

[5] 翁義聰，利用編號處理碎形的演算法，數學傳播第 15 卷第三期，1991。

[6] 遍歷定理，百度百科 from http://baike.baidu.com/view/692121.html。

[7] Iterated function system，from http:// http://en.wikipedia.org/wiki/ Iterated_function_system [英文]。

[8] Hausdorff dimension，from http:// http://en.wikipedia.org/wiki/ Hausdorff_ dimension [英文]。

玖、附件（一）EXCEL 數據分析

邊數	內角	可否整除	邊數	內角	可否整除	邊數	內角	可否整除	邊數	內角	可否整除	邊數	內角	可否整除	邊數	內角	可否整除
3	60	6.000	63	174.286	2.066	123	177.073	2.033	183	178.033	2.022	243	178.519	2.017	303	178.812	2.013
4	90	4.000	64	174.375	2.065	124	177.097	2.033	184	178.043	2.022	244	178.525	2.017	304	178.816	2.013
5	108	3.333	65	174.462	2.063	125	177.120	2.033	185	178.054	2.022	245	178.531	2.016	305	178.820	2.013
6	120	3.000	66	174.545	2.063	126	177.143	2.032	186	178.065	2.022	246	178.537	2.016	306	178.824	2.013
7	129	2.800	67	174.627	2.062	127	177.165	2.032	187	178.075	2.022	247	178.543	2.016	307	178.827	2.013
8	135	2.667	68	174.706	2.061	128	177.188	2.032	188	178.085	2.022	248	178.548	2.016	308	178.831	2.013
9	140	2.571	69	174.783	2.060	129	177.209	2.031	189	178.095	2.021	249	178.554	2.016	309	178.835	2.013
10	144	2.500	70	174.857	2.059	130	177.231	2.031	190	178.105	2.021	250	178.560	2.016	310	178.839	2.013
11	147	2.444	71	174.930	2.058	131	177.252	2.031	191	178.115	2.021	251	178.566	2.016	311	178.842	2.013
12	150	2.400	72	175.000	2.057	132	177.273	2.031	192	178.125	2.021	252	178.571	2.016	312	178.846	2.013
13	152	2.364	73	175.068	2.056	133	177.293	2.031	193	178.135	2.021	253	178.577	2.016	313	178.850	2.013
14	154	2.333	74	175.135	2.056	134	177.313	2.030	194	178.144	2.021	254	178.583	2.016	314	178.854	2.013
15	156	2.308	75	175.200	2.055	135	177.333	2.030	195	178.154	2.021	255	178.588	2.016	315	178.857	2.013
16	158	2.286	76	175.263	2.054	136	177.353	2.030	196	178.163	2.021	256	178.594	2.016	316	178.861	2.013
17	159	2.267	77	175.325	2.053	137	177.372	2.030	197	178.173	2.021	257	178.599	2.016	317	178.864	2.013
18	160	2.250	78	175.385	2.053	138	177.391	2.029	198	178.182	2.020	258	178.605	2.016	318	178.868	2.013
19	161	2.235	79	175.443	2.052	139	177.410	2.029	199	178.191	2.020	259	178.610	2.016	319	178.871	2.013
20	162	2.222	80	175.500	2.051	140	177.429	2.029	200	178.200	2.020	260	178.615	2.016	320	178.875	2.013
21	163	2.211	81	175.556	2.051	141	177.447	2.029	201	178.209	2.020	261	178.621	2.015	321	178.879	2.013
22	164	2.200	82	175.610	2.050	142	177.465	2.029	202	178.218	2.020	262	178.626	2.015	322	178.882	2.013
23	164	2.190	83	175.663	2.049	143	177.483	2.028	203	178.227	2.020	263	178.631	2.015	323	178.885	2.012
24	165	2.182	84	175.714	2.049	144	177.500	2.028	204	178.235	2.020	264	178.636	2.015	324	178.889	2.012
25	166	2.174	85	175.765	2.048	145	177.517	2.028	205	178.244	2.020	265	178.642	2.015	325	178.892	2.012
26	166	2.167	86	175.814	2.048	146	177.534	2.028	206	178.252	2.020	266	178.647	2.015	326	178.896	2.012
27	167	2.160	87	175.862	2.047	147	177.551	2.028	207	178.261	2.020	267	178.652	2.015	327	178.899	2.012
28	167	2.154	88	175.909	2.047	148	177.568	2.027	208	178.269	2.019	268	178.657	2.015	328	178.902	2.012
29	168	2.148	89	175.955	2.046	149	177.584	2.027	209	178.278	2.019	269	178.662	2.015	329	178.906	2.012
30	168	2.143	90	176.000	2.045	150	177.600	2.027	210	178.286	2.019	270	178.667	2.015	330	178.909	2.012
31	168	2.138	91	176.044	2.045	151	177.616	2.027	211	178.294	2.019	271	178.672	2.015	331	178.912	2.012
32	169	2.133	92	176.087	2.044	152	177.632	2.027	212	178.302	2.019	272	178.676	2.015	332	178.916	2.012
33	169	2.129	93	176.129	2.044	153	177.647	2.026	213	178.310	2.019	273	178.681	2.015	333	178.919	2.012
34	169	2.125	94	176.170	2.043	154	177.662	2.026	214	178.318	2.019	274	178.686	2.015	334	178.922	2.012
35	170	2.121	95	176.211	2.043	155	177.677	2.026	215	178.326	2.019	275	178.691	2.015	335	178.925	2.012
36	170	2.118	96	176.250	2.043	156	177.692	2.026	216	178.333	2.019	276	178.696	2.015	336	178.929	2.012
37	170	2.114	97	176.289	2.042	157	177.707	2.026	217	178.341	2.019	277	178.700	2.015	337	178.932	2.012
38	171	2.111	98	176.327	2.042	158	177.722	2.026	218	178.349	2.019	278	178.705	2.014	338	178.935	2.012
39	171	2.108	99	176.364	2.041	159	177.736	2.025	219	178.356	2.018	279	178.710	2.014	339	178.938	2.012
40	171	2.105	100	176.400	2.041	160	177.750	2.025	220	178.364	2.018	280	178.714	2.014	340	178.941	2.012
41	171	2.103	101	176.436	2.040	161	177.764	2.025	221	178.371	2.018	281	178.719	2.014	341	178.944	2.012
42	171	2.100	102	176.471	2.040	162	177.778	2.025	222	178.378	2.018	282	178.723	2.014	342	178.947	2.012
43	172	2.098	103	176.505	2.040	163	177.791	2.025	223	178.386	2.018	283	178.728	2.014	343	178.950	2.012
44	172	2.095	104	176.538	2.039	164	177.805	2.025	224	178.393	2.018	284	178.732	2.014	344	178.953	2.012
45	172	2.093	105	176.571	2.039	165	177.818	2.025	225	178.400	2.018	285	178.737	2.014	345	178.957	2.012
46	172	2.091	106	176.604	2.038	166	177.831	2.024	226	178.407	2.018	286	178.741	2.014	346	178.960	2.012
47	172	2.089	107	176.636	2.038	167	177.844	2.024	227	178.414	2.018	287	178.746	2.014	347	178.963	2.012
48	173	2.087	108	176.667	2.038	168	177.857	2.024	228	178.421	2.018	288	178.750	2.014	348	178.966	2.012
49	173	2.085	109	176.697	2.037	169	177.870	2.024	229	178.428	2.018	289	178.754	2.014	349	178.968	2.012
50	173	2.083	110	176.727	2.037	170	177.882	2.024	230	178.435	2.018	290	178.759	2.014	350	178.971	2.011
51	173	2.082	111	176.757	2.037	171	177.895	2.024	231	178.442	2.017	291	178.763	2.014	351	178.974	2.011
52	173	2.080	112	176.786	2.036	172	177.907	2.024	232	178.448	2.017	292	178.767	2.014	352	178.977	2.011
53	173	2.078	113	176.814	2.036	173	177.919	2.023	233	178.455	2.017	293	178.771	2.014	353	178.980	2.011
54	173	2.077	114	176.842	2.036	174	177.931	2.023	234	178.462	2.017	294	178.776	2.014	354	178.983	2.011
55	173	2.075	115	176.870	2.035	175	177.943	2.023	235	178.468	2.017	295	178.780	2.014	355	178.986	2.011
56	174	2.074	116	176.897	2.035	176	177.955	2.023	236	178.475	2.017	296	178.784	2.014	356	178.989	2.011
57	174	2.073	117	176.923	2.035	177	177.966	2.023	237	178.481	2.017	297	178.788	2.014	357	178.992	2.011
58	174	2.071	118	176.949	2.034	178	177.978	2.023	238	178.487	2.017	298	178.792	2.014	358	178.994	2.011
59	174	2.070	119	176.975	2.034	179	177.989	2.023	239	178.494	2.017	299	178.796	2.013	359	178.997	2.011
60	174	2.069	120	177.000	2.034	180	178.000	2.022	240	178.500	2.017	300	178.800	2.013	360	179.000	2.011
61	174	2.068	121	177.025	2.034	181	178.011	2.022	241	178.506	2.017	301	178.804	2.013	361	179.003	2.011
62	174	2.067	122	177.049	2.033	182	178.022	2.022	242	178.512	2.017	302	178.808	2.013	362	179.006	2.011

附件（二）VB2008 EXPRESS 程式碼

```
        LayerColor(0) = Color.Black
        LayerColor(1) = Color.Blue
        LayerColor(2) = Color.Brown
        LayerColor(3) = Color.BurlyWood
        LayerColor(4) = Color.Chocolate
        LayerColor(5) = Color.Cyan


    End Sub

    Private Sub ClearResource()
        '重置底色
        g.Clear(Color.AntiqueWhite)
        '清除正多邊形之頂點集合
        Polygon_Points.Clear()
        '清除正多邊形之邊線方程式集合
        LineEqu.Clear()
        '清除待處理之線段集合
        LineQueue.Clear()
    End Sub

    Private Sub btnPolygon_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPoly-
gon.Click
        Dim angle As Double
        Dim angle_off As Double
        Dim pointarray As PointF()


        '取得半徑長度
        RADIUD = txtRadius.Text

        '取得層數
        LAYER = txtL.Text

        '取得多邊形邊數
        POLYGON_N = txtN.Text
        '每個弧的弧度
        angle = 360 / POLYGON_N * Math.PI / 180
        '第一個正 N 邊形的頂點的弧度值
        angle_off = Math.PI / 2 - angle / 2

        '重繪底色及重置變數值
        ClearResource()

        For i = 1 To POLYGON_N
            '產生每個頂點的座標
            Dim p As New PointF
            p.X = CENTER_X + RADIUD * Math.Cos(angle_off)
            p.Y = CENTER_Y - RADIUD * Math.Sin(angle_off)
            '加入到一個 List 內
            Polygon_Points.Add(p)

            '下一個頂數的弧度值
            angle_off += angle
        Next
        pointarray = Polygon_Points.ToArray()

        '畫出多邊形
        g.DrawPolygon(New Pen(Color.Red, 2), pointarray)
```

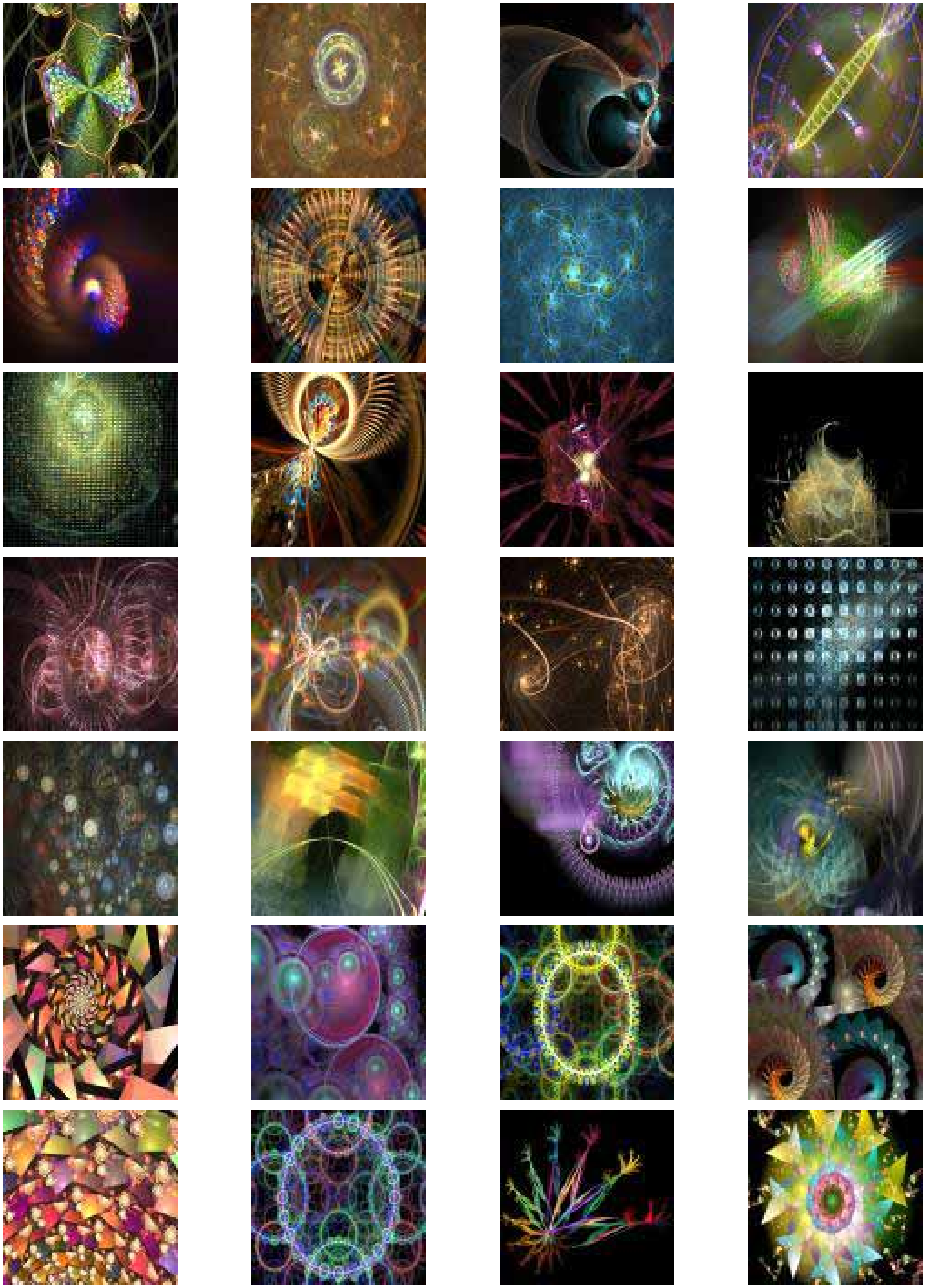
```
    '產生每個邊的方程式
    Dim P1 As PointF
    Dim P2 As PointF
    For i = 1 To POLYGON_N
        If (i = POLYGON_N) Then
            P1 = pointarray(i - 1)
            P2 = pointarray(0)
        Else
            P1 = pointarray(i - 1)
            P2 = pointarray(i)
        End If

        Dim equ(4) As Double
        '算出經過 P1, P2 的直線方程式: ax+by+c=0 中的 a, b, c
        equ(0) = (P1.Y - P2.Y)                                'a
        equ(1) = (P1.X - P2.X) * (-1)                         'b
        equ(2) = (equ(0) * P1.X + equ(1) * P1.Y) * (-1)       'c
        equ(3) = 2 * equ(0) / (equ(0) * equ(0) + equ(1) * equ(1))    '2a/(a^2+b^2), 用於計算對稱點的座標
        equ(4) = 2 * equ(1) / (equ(0) * equ(0) + equ(1) * equ(1))    '2b/(a^2+b^2), 用於計算對稱點的座標

        LineEqu.Add(equ)
    Next

    '產生邊框
    pointarray = Polygon_Points.ToArray()
    For i = 1 To POLYGON_N
        If (i = POLYGON_N) Then
            P1 = pointarray(i - 1)
            P2 = pointarray(0)
        Else
            P1 = pointarray(i - 1)
            P2 = pointarray(i)
        End If

        '繪出 P1,P2 所構成線段經過正 N 邊形反射後的結果
        DrawReflectionLine(P1, P2)
    Next

    '畫出多邊形
    g.DrawPolygon(New Pen(Color.Red, 2), pointarray)

    '顯示繪畫後的結果於視窗
    PictureBox1.Image = MainBMP
End Sub

'para: 直線方程式
'p1, p2: 線段的兩個端數
'判斷此線段和中心點是否在直線方程式的同一邊
Private Function InMirror(ByVal para As Double(), ByVal p1 As PointF, ByVal p2 As PointF) As Boolean
    Dim equvalue As Double

    equvalue = EvalEqu(para, p1) * EvalEqu(para, New PointF(CENTER_X, CENTER_Y))
    If (equvalue < -1) Then    '因為有小數點的問題，所以有些值會是-0.xxxxx
        Return False
    End If
    equvalue = EvalEqu(para, p2) * EvalEqu(para, New PointF(CENTER_X, CENTER_Y))
    If (equvalue < -1) Then    '因為有小數點的問題，所以有些值會是-0.xxxxx
        Return False
    End If

    Return True
End Function
```

```
'pX, pY 為線段的兩個端點座標
'在紅色正 N 邊形內繪出一個線段，並繪出此線段在每一邊的反射之其後續的反射
Private Sub DrawReflectionLine(ByVal pX As PointF, ByVal pY As PointF)
    Dim first_line As New LineInMirror
    Dim proceeding_line As LineInMirror
    Dim equs()() As Double
    Dim c1 As Color
    Dim c2 As Color

    '取得所有邊的直線方程式
    equs = LineEqu.ToArray()

    '設定紅色正 N 邊形的線段，兩個端點座標和第幾次反射
    first_line.p1 = pX
    first_line.p2 = pY
    first_line.l = 0              '使用者所設定，沒有經過任何反射；所以為 0
    LineQueue.Add(first_line)     '加到待處理的線段

    Do
        '取得第一個未處理的線段
        proceeding_line = LineQueue.First()

        '開始處理每個線段
        '先判斷是否超過設定的反射次數
        If (proceeding_line.l <= LAYER) Then
            '針對正 N 邊形的每個邊，檢查是否要做反射的處理
            For i = 1 To POLYGON_N
                '檢查是否此線段是否和中心點在同一邊
                If (Not InMirror(equs(i - 1), proceeding_line.p1, proceeding_line.p2)) Then
                    '若不是的話，表示是在鏡子後面，不用處理反射的動作
                    Continue For
                End If

                '繪出此線段
                'g.DrawLine(New Pen(Color.Black, 1), proceeding_line.p1, proceeding_line.p2)
                '   If (BgColor = MainBMP.GetPixel(proceeding_line.p1.X, proceeding_line.p1.Y) Or BgColor =
MainBMP.GetPixel(proceeding_line.p2.X, proceeding_line.p2.Y)) Then
                c1 = MainBMP.GetPixel(proceeding_line.p1.X, proceeding_line.p1.Y)
                c2 = MainBMP.GetPixel(proceeding_line.p2.X, proceeding_line.p2.Y)
                g.DrawLine(New Pen(LayerColor(proceeding_line.l), 1), proceeding_line.p1, proceeding_line.p2)
                'End If

                '開始處理反射
                Dim new_line As New LineInMirror
                '此線段的兩個端點，經正 N 邊形的某一邊，反射後後會得到兩個點
                new_line.p1 = Reflection(equs(i - 1), proceeding_line.p1)
                new_line.p2 = Reflection(equs(i - 1), proceeding_line.p2)
                '得到兩點座標後，再將反射的次數加一，產生新的未處理線段
                new_line.l = proceeding_line.l + 1
                '加到未處理線段的集合
                LineQueue.Add(new_line)
            Next
        End If
        '移除第一個的線段，因為已處理完畢
        LineQueue.RemoveAt(0)

    Loop While LineQueue.Count
End Sub
```

```
'para: 直線方程式
'p: 座標上的某一點
'將 p 點帶入直線方程式，得到的數值來判斷位於此直線的方位
Private Function EvalEqu(ByVal para As Double(), ByVal p As PointF) As Double
    Dim rtn As Double
    rtn = p.X * para(0) + p.Y * para(1) + para(2)
    Return rtn
End Function


'para: 直線方程式, aX+bY+c=0
'p: 座標上的某一點, (x, y)
'算出 p 點經直線反射的座標，即 p 點對直線的對稱點
Private Function Reflection(ByVal para As Double(), ByVal p As PointF) As PointF
    Dim np As New PointF
    Dim tmp As Double

    tmp = EvalEqu(para, p)          ' ax+by+c

    np.X = p.X - para(3) * tmp      ' x - (ab+by+c)*2a/(a^2+b^2)
    np.Y = p.Y - para(4) * tmp      ' y - (ab+by+c)*2b/(a^2+b^2)

    Return np
End Function

'滑鼠左鍵被按下的處理程序
Private Sub PictureBox1_MouseDown(ByVal sender As System.Object, ByVal e As Sys-
tem.Windows.Forms.MouseEventArgs) Handles PictureBox1.MouseDown
    '開始要繪線段
    StartDraw = True
    '取得起始點座標
    OldNode = New PointF(e.X, e.Y)
End Sub

'滑鼠左鍵被放開的處理程序
Private Sub PictureBox1_MouseUp(ByVal sender As System.Object, ByVal e As Sys-
tem.Windows.Forms.MouseEventArgs) Handles PictureBox1.MouseUp
    '不再繪線段
    StartDraw = False
End Sub

'滑鼠移動的處理程序
Private Sub PictureBox1_MouseMove(ByVal sender As System.Object, ByVal e As Sys-
tem.Windows.Forms.MouseEventArgs) Handles PictureBox1.MouseMove
    '是否開始要繪製線段
    If Not StartDraw Then
        '不是話就離開
        Return
    End If

    Dim equs()() As Double

    '取得線段的結束點座標
    NewNode = New PointF(e.X, e.Y)

    '取得正 N 邊形的每一個邊的直線方程式
    equs = LineEqu.ToArray()

    '檢查是否在正 N 邊形裡面
    For i = 1 To POLYGON_N
```

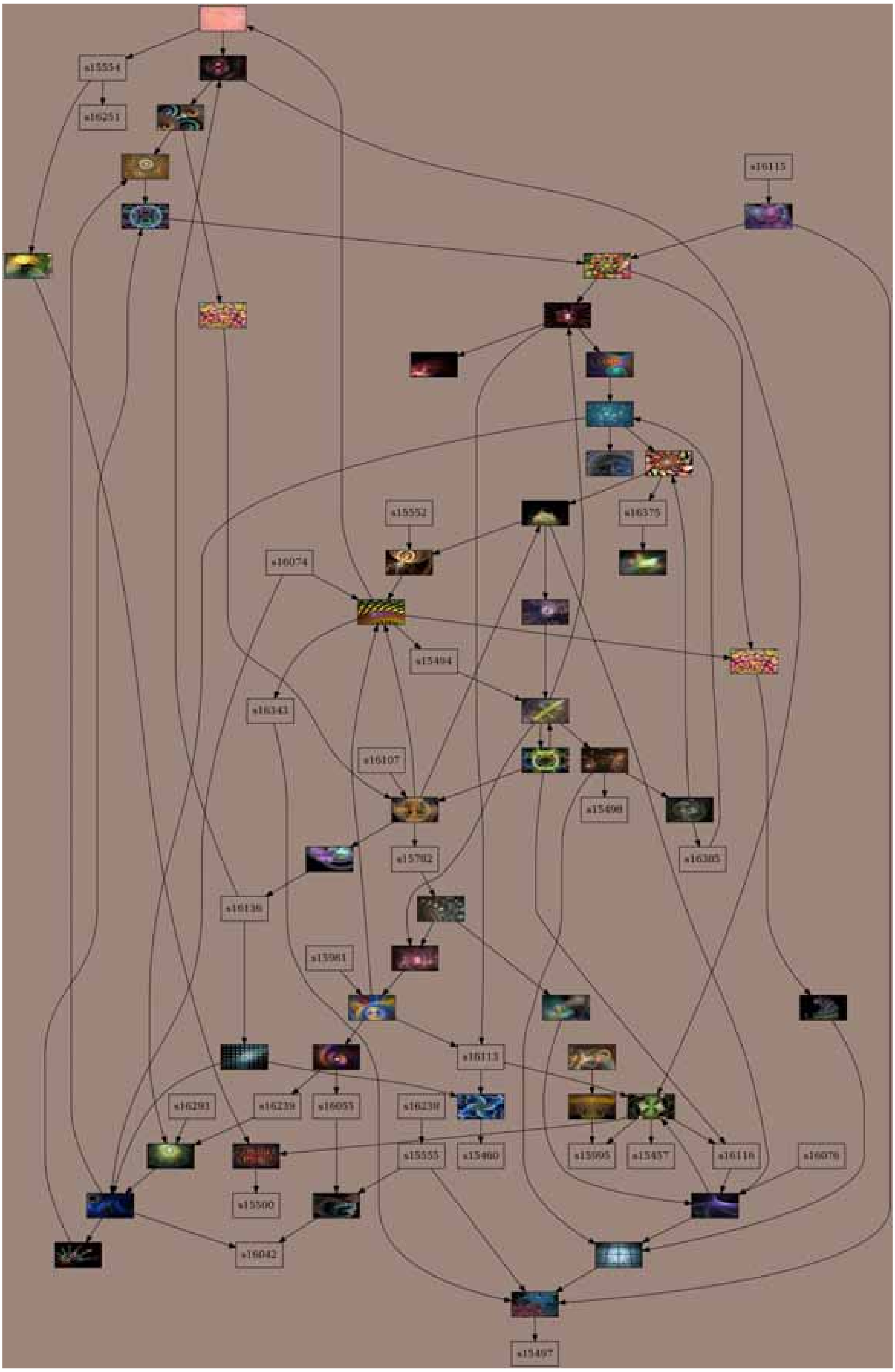
```
            '此線段是否和中心點是在同一邊
            If (Not InMirror(equs(i - 1), OldNode, NewNode)) Then
                '若不是的話，就不用繪製線段了
                '更新新的起始點座標
                OldNode = NewNode
                Return
            End If
        Next

        '開始處理繪製線段及其反射線段的程序
        DrawReflectionLine(OldNode, NewNode)
        '更新起始點座標
        OldNode = NewNode
        '將繪製的結果顯示於視窗
        PictureBox1.Image = MainBMP
    End Sub
End Class

'自訂的資料形態
'線段經鏡子反射時，除了要記錄線段的兩個端點外，還要記錄反射過幾次
Class LineInMirror
    Public p1 As PointF
    Public p2 As PointF
    Public l As Integer
End Class
```

附件（三）美麗的電子模擬碎形-綿羊分析 1



資料來源:http://tw.babelfish.yahoo.com/translate_url?doit=done&tt=url&trurl=http%3A%2F%2Felectricsheep.org%2F&lp=en_zt&. intl=tw&fr=yfp

附件（三）美麗的電子模擬碎形-綿羊分析 2



【評語】040412

1、 作者群設計可調式萬花筒，利用最多 12 面鏡的反射複製圖案產生碎形。

2、 大量使用數學專業名詞，正如萬花筒般。未能以實作方式說明重要的數學性質是可惜的地方。