

## 小波處理與分析 (Wavelet)

### 1. 什麼是 wavelet ?

一張圖片，是由一系列不同波長（或者頻率，圈子內說頻率說得多些，但我覺得波長更容易理解）的成分所組成的。

什麼意思？

我們先從一維的角度去想這個問題。我舉一個 PILE 說明書裏的例子。假設有這麼一個統計資料：500 年前到今天，每一個小時的溫度。這是一個及其龐大的資料組，大約由 450 萬個資料組成。然後我們把它描繪成一個函數圖像，x 軸是時間，y 軸是溫度。

有了這麼一個圖像，我們可以很直觀地觀察溫度的變化趨勢。但是，這裏的所謂觀察，存在一個尺度問題。我從「1 天」這個尺度去看，可以看見溫度大概呈一個正弦曲線，下午兩點左右達到最大值，凌晨兩點左右達到最小值。把尺度放大一點，從「1 年」去看。那溫度變化還是像一個正弦曲線，只不過放大了許多倍。夏天達到最大值，冬天達到最小值。

再將尺度放大，放大到半個世紀，那麼溫度的波動就是另一個模樣了。溫度的波動看起來沒有先前那麼規律了。但我們還是能把溫度的變化和一些東西聯繫起來，比如幾年一次的厄爾尼諾，以及太陽活動週期等。

再放大，放大到兩個世紀，那麼，週期性消失了，取而代之的是逐漸上升的曲線，那是進入工業時代以來的全球變暖。

試想一下，假如我們從「1 天」這個尺度上去看，能看出全球變暖嗎？從「2 個世紀」這個尺度上去看，能看出每天的溫度在下午兩點左右最高嗎？顯然是不可能的。

從上面的例子可以看出，溫度-時間函數如果看做一系列波的話，可以被拆分爲波長不同的好幾個部分。主體是一個波長極長的波，然後上面疊著波長小點的

波，如此大波疊小波，如漢諾塔。

Wavelet 做的工作，就是把這些不同波長的波拆解出來，分別進行處理，處理完以後又重新疊回去。

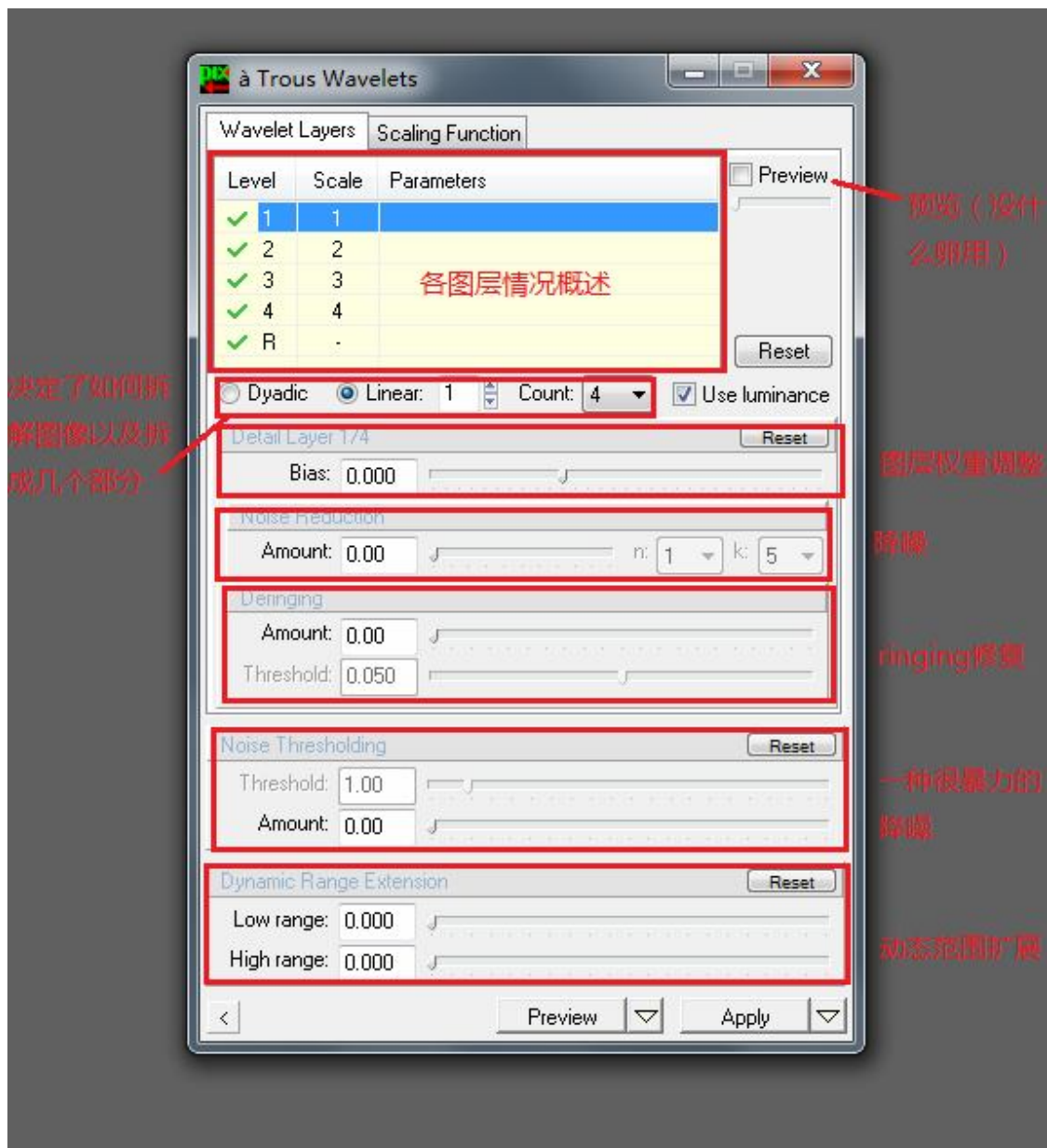
現在我們把思維擴展到二維圖像上。這裏所說的波長，差不多和圖像中物體的半徑有關。一張標準的深空片子，裏面包含的東西波長從小到大會有：噪點，暗星，亮星 與微妙細節，大尺度細節，星雲（或星系）的整體形狀，整個背景的明暗變化。Wavelet 工具可以把這些成分拆解出來，分別處理。比如我想降噪，同時不想 過多地影響細節，那根據雜訊的高頻性（也就是波長較小），我們可以在小波長圖層裏進行降噪，而更大波長的圖層不會受到影響。

## 2. 認識工具：a Trous Wavelets Transform

在 PixInsight LE 裏面，Wavelet 運算靠 a Trous Wavelets Transform（ATWT）工具實現。我們首先來認識一下這個工具是怎麼運作的。

ATWT 首先把一張圖片拆分成多個波長的圖層，然後對各圖層進行處理。處理完畢以後，ATWT 會把先前拆解出的所有圖層（這裏存在一個圖層是否有效的問題，下面會說）重新疊加起來，生成新圖像。

ATWT 工具只允許在拆分之前調節參數，所有的拆分與處理工作都在軟體內部進行。也就是說，所謂的拆分，不是真的拆成十幾個圖層然後顯示出來給你看。因此你在進行 ATWT 之前，要先想好自己所要獲得的效果，然後再設置參數。也就是說，想盲目地靠調節 ATWT 參數來試出一個最佳效果，是不科學的。但這種在腦子裏 預處理的能力，不是一朝一夕就能得到的，需要對 ATWT 與自己所處理的圖像有充分的瞭解才行。



好了，現在來看看 ATWT 的操作面板。首先我們來看第一個選項卡，Wavelet Layers。這是對拆分方式以及對每個圖層進行處理的控制處（另一個選項卡 Scaling Function 決定的是拆分時使用的演算法，或說濾鏡）。

先看到右邊那兩個可以打勾的東西，Preview 與 Use Luminance。

Preview：有什麼作用，我到現在也沒有搞懂，不過到目前為止的實踐上看，似乎沒什麼卵用，默認是不選中的，大家按照默認就行了，別去勾選（勾選了以後，應用在圖片上會有很神奇的事情發生，無聊者可以試試看）。

Use Luminance：決定了在哪個通道（明度與色彩）上拆解圖像，一般來說，都

會選中它。

今天下午沒多少時間了，我也就只講講我在圖中標注的*i* 決定了如何拆解圖像以及拆分成幾個部分*i* 的那個框框。

**Dyadic**：選中這個模式的話，圖像按照以 2 為公比的等比數列進行拆分，即波長為 1,2,4,8*i* 但要注意的是，裏面所謂的*i*8*i* 圖層，不是只包含了波長為 8 的成分，而是包含了波長 5-8 的所有成分。其他的諸如*i*16*i* 圖層，*i*32*i* 圖層也是一樣，不止包含了 16、32 的成分。

**Linear**：選中這個模式的話，圖像按照以後面那個框裏的數值為公差的等差數列進行拆分。比如，如果那個框裏的值為 1，那麼就拆分成波長為 1,2,3,4*i* 的圖層。如果那個框裏的值為 5，那麼就拆分為 1,6,11*i* 同樣，第二個例子裏的*i*6*i* 圖層包含的是波長 1-5 的所有成分。其他圖層也一樣，除非是*i*1*i* 圖層或者那個框裏的值為 1。

**Count** 的作用是顯示多少個圖層。需要注意的是，這個值不包含一定會生成的*i*R 圖層*i*（以後會說）。所以，你能看到的圖層應該是*i*count+1*i* 個圖層。

下面是一個 Wavelet 拆解圖像的實例，原圖是烤鴨拍攝的獵戶廣域，後期處理是我完成的。

原圖：



拆解：

波長 1 圖元：





波長 2 圖元：





波長 3-4 圖元：



波長 5-8 圖元：





波長 9-16 圖元：

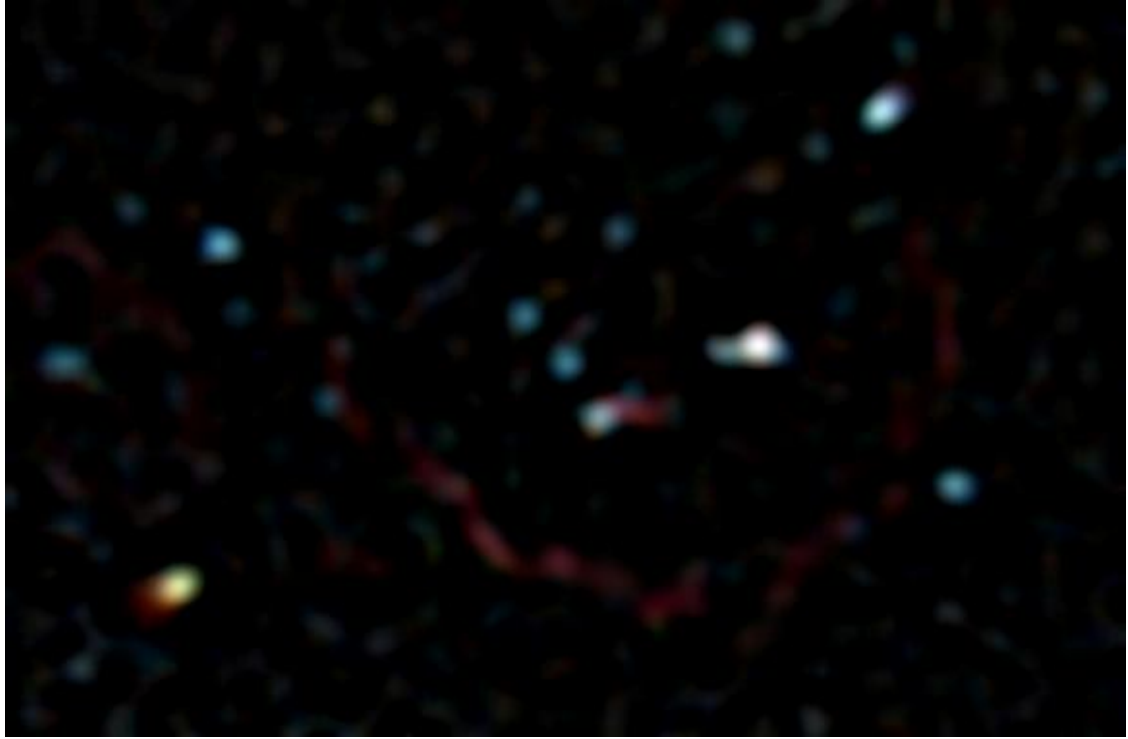


波長 17-32 圖元：

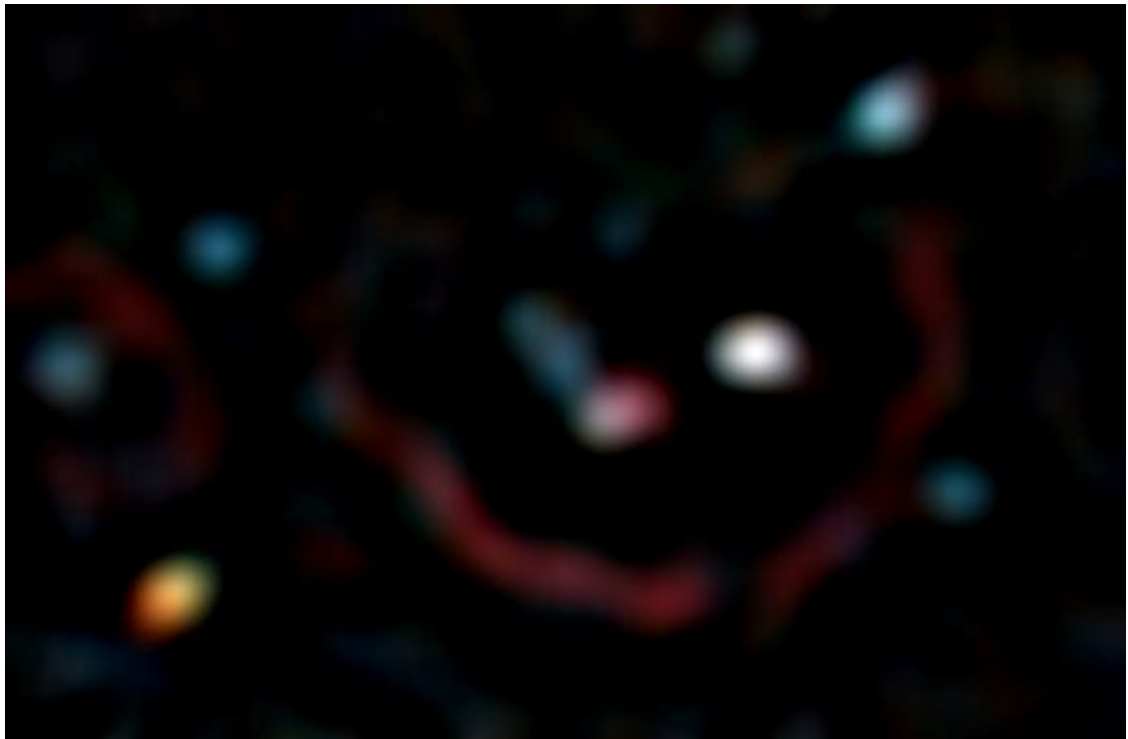


波長 33-64 圖元：





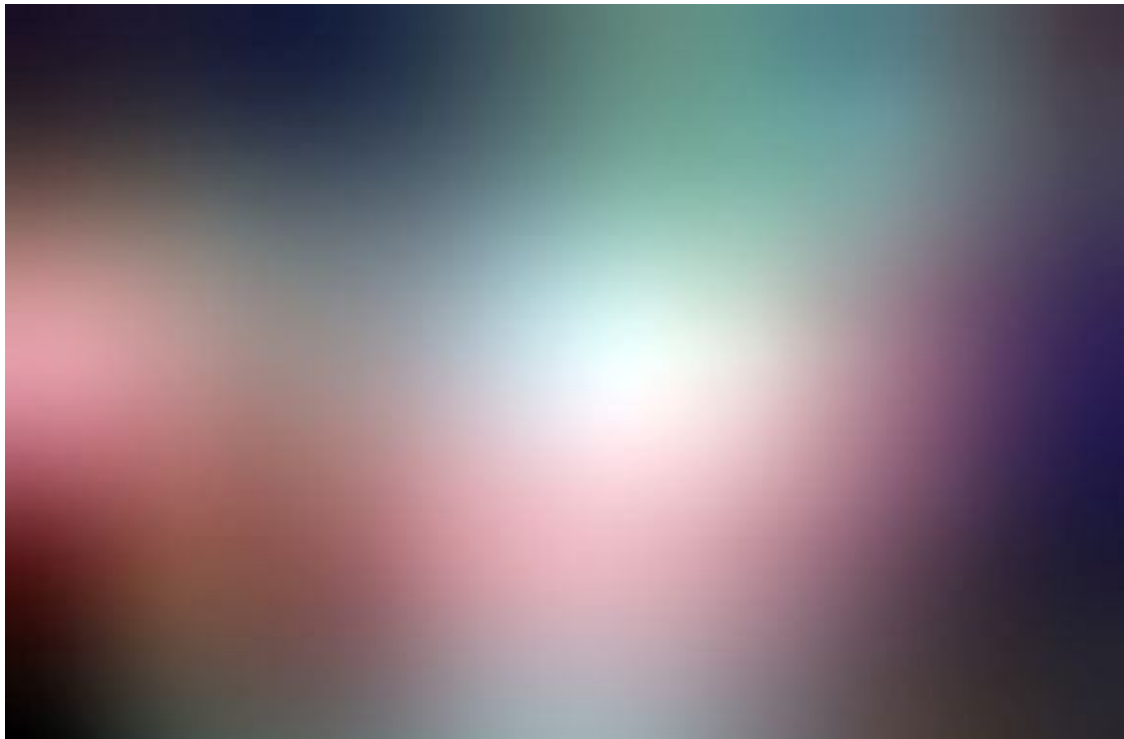
波長 65-128 圖元：



波長 129-256 圖元：



波長>256 圖元：



上面說的 **preview** 選項，我在翻譯 **wavelet** 模組時沒弄懂，但今天上午在翻譯【**wavelet + sgbnr**】模組時弄懂了它的作用。

操作 方法：先在 **detail layers** 裏面選中一個圖層，然後選中 **preview**，應用到圖像上。運算完成後出現的結果，是你選中的那個圖層包含的結構，經過了反色。**preview** 下面還有一個滑塊，用來調整結果的 **gamma**。這就是所謂預覽，讓你瞭解一下這個圖層裏有哪些結構。

對我個人來說，我不喜歡這種預覽方式，因為反色後的圖像會降低我的判斷力。爲了觀察每個圖層裏有什麼，我寧願麻煩點，用 **disable layer+histogram transform**

**Layer Parameters Summary** 各圖層情況概述：

在我之前給出的截圖中，我框出了一個區域，標注爲 **各圖層情況概述**，我利用微機課時間來給大家講講這個東西。

這個區域，由 **n** 行（取決於你拆解出了多少個圖層），3 列組成。



Level	Scale	Parameters
✗ 1	1	
✓ 2	4	+1.000, S:[0.20,2,3], D:[0.10,0.040]
✓ 3	7	
✓ 4	10	
✓ R	-	

☐ Preview

☐ Dyadic
 ☒ Linear: 3
 Count: 4
 ☒ Use luminance

行：按照從上到下波長遞增的規則，每一行為一個圖層。最後一行為 R 圖層。什麼是 R 圖層呢？R 是 *Residual*（剩餘的）的縮寫。由於諸多因素的限制（如記憶體大小），Wavelet 運算不能完全拆解圖像，只能拆解出圖像中相對高頻（小波長）的成分，而剩下沒有拆解出來的較低頻的成分，一起組成了 R 圖層。比如說我拆解出來四個圖層，使用 Linear 模式，公差為 1，則拆解出的圖層為波長 1,2,3,4 的四個圖層，剩下的所有波長大於 5 的成分，統一聚集在 R 圖層中。

列：三列分別是：Level，Scale，Parameters。

Level：Level 列裏又分處兩列，左列表示該圖層是否有效（具體之後說），有效為一綠勾，無效為一紅叉。右列裏的數字表示的是該圖層的編號。

Scale：表示該圖層的波長。注意，之前說過，一個圖層可能不只包含一個波長的成分。

Parameters：該圖層的 Detail Layer 參數設置情況。顯示格式是：a，S:(b，c，d)，D:(e，f)

其中，各字母的含義為：

a：該圖層的 Bias；b：該圖層 NR（Noise Reduction）的 Amount（強度）；c：該圖層反復進行 NR 的次數（實際上應該理解為遞迴次數或堆疊次數）；d：該圖層 NR 應用的 Filter Kernel Size；e：該圖層 Deringing 的 amount；f：該圖層 Deringing 的 threshold（閾值）。

在我給出的截圖中，從該區域可讀取的資訊為：總共拆解出來四個圖層，使用的是 Linear 模式，公差為 3，1 號圖層無效。其中，第二個圖層（波長 2-4）的 Detail Layer 參數進行過設置：Bias 為 +1；NR 強度為 0.2，執行兩次，Filter Kernel Size 為 3x3；Deringing 強度為 0.1，閾值為 0.04。

Individual Layer Parameter 對特定圖層進行調整：

1，Enabled&Disabled Layer：之前說過，ATWT 將圖像拆解開來，對各波長圖層進行處理之後再疊回去。在這裏，我們可以決定一個圖層是否有效。若某圖層無效，

則在重新疊合 時，該圖層將不會疊入。比如說我把波長為 3、4 的兩個圖層無效掉，那最終結果中將不包含波長為 3、4 的結構。具體操作是，雙擊特定圖層在 Level 列裏表示圖層是否有效的那個符號（綠勾或紅叉），即可改變該圖層的狀態。

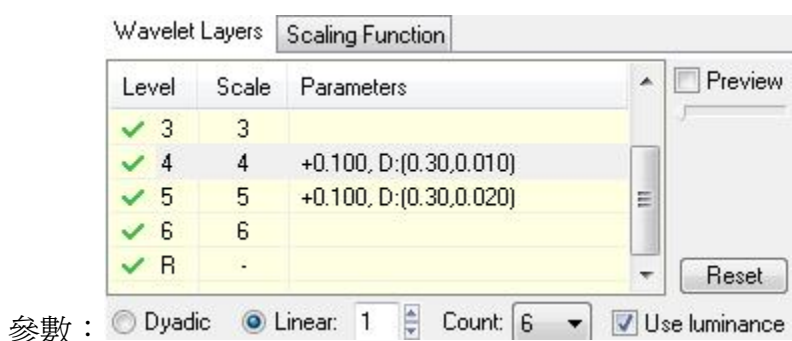
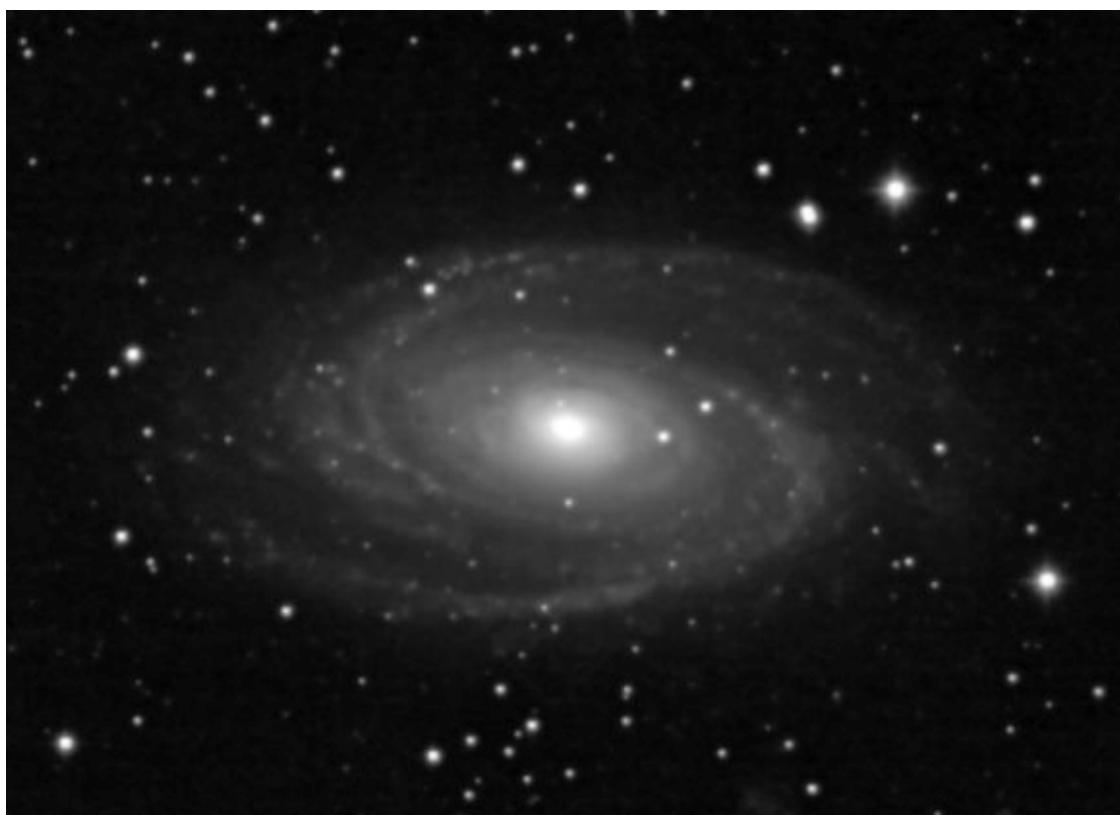
2，Bias：這個參數調整的是該圖層在最終結果中所占的比重，正值表示增加，負值表示減少。如果某圖層的比重增加了，那最終結果中那個波長的成分會顯得突出一些，也就是說，這些結構得到了「強化」。相反，如果一個圖層的比重減少了，那這個波長的結構會被壓制。

調整 bias 強化結構一例，原片是蟲子拍的 M81。

原片：



強化後：



其實，我還用了蒙版，但這不是重點。而且，這也不是真正正統的強化方法，不要模仿，我只是舉這個例子，說明一下 atwt 強化。

3, Noise Reduction：降噪。這裏的 NR 用的是[一種特殊的抹平機制]，具體演算法是啥，說明書裏沒說。

Amount：表示 NR 強度，為 0-1 的實數，該值越大，NR 力度越大。

Number of Iterations：NR 執行次數（事實上應該稱為遞迴次數或者堆疊次數），一般來說，用較低的 amount 配合多次執行，效果更好。

Filter Kernel Size：這裏所謂降噪演算法，可以理解為一種低通濾鏡，一個正方形的低通濾鏡，Filter Kernel Size 表示該正方形的邊長。照直覺，越大的低通濾鏡降



噪效果越強，但這裏不是這樣。這個 Fks 和雜訊尺度關係更大，和降噪力度的關係很微妙。一般來說是 使用 3 或 5 的 Fks，其他不推薦用。

4，Deringing。通過調整 bias 的方式進行細節強化，和使用高通濾鏡強化細節沒什麼本質上的區別。然而高通濾鏡有一個令人詬病已久的毛病，叫做 Gibbs 效應，也叫做 Ringing 效應。具體來說就是經過強化以後的圖片，在亮部周圍有一圈暗圈，暗部周圍一圈亮圈。在這裏我就不舉例子了，大家自己拿一張深空照放進 Photoshop 裏面做一個大半徑（比如，半徑 20 圖元）的 USM 銳化，就能很清楚的認識到 Ringing 效應。

Ringing 效應可以用 ATWT 裏的 Deringing 功能壓制，但壓制 Ringing 與強化細節很難兼顧，因為如果過度壓制 Ringing 可能會使之前做的強化工作效果不明顯。我們要做的應該是讓 Ringing 效應小到不足以引起注意，應該尋求效果足夠的細節強化與不明顯的 Ringing 之間的微妙平衡（事實上，使用蒙版配合 Deringing 可以獲得很好的效果，這裏不多說）。

Amount：強度，不解釋。

Threshold：閾值，決定了容許出現的最低 Ringing 效應，0 是最暴力的設置，完全消除 Ringing。越高的 Ringing 值，壓制 Ringing 的效果越弱。

下面來說說另外兩個不能用於特定圖冊的工具。

Noise Threshold:這個演算法只能用於前四個圖層(對前四個圖層都有效，不能指定用於特定圖層)。這個演算法的原理很簡單:設定一個閾值，然後拿每一個圖元的值和閾值比對，若低於閾值，則將該圖元乘上一個很小(甚至為零)的值，以此壓制雜訊。

Threshold:閾值，低於閾值的圖元將被視為雜訊被壓制(事實上，閾值並不是你輸入的這個值，而是基於這個值算出來的，演算法基於正態分佈，你輸入的事實上是 sigma)。該值越大，壓制雜訊效果越明顯。說明書裏建議值為 1~3。

Amount:可以理解為強度。低於閾值的圖元，將被乘以(1-amount)。因此，amount 越大，降噪力度越大。

Dynamic Range Extension 動態範圍擴展:用調整 bias 的方式來強化(弱化)細節之後，可能會使影像的某些部分飽和(或減為零)，出現黑白純色。這種問題可以通過 DRE 解決。

HDRE:通過把圖像壓暗來保護亮部細節，該值越大，強度越大。

LDRE:同 HDRE 相反。

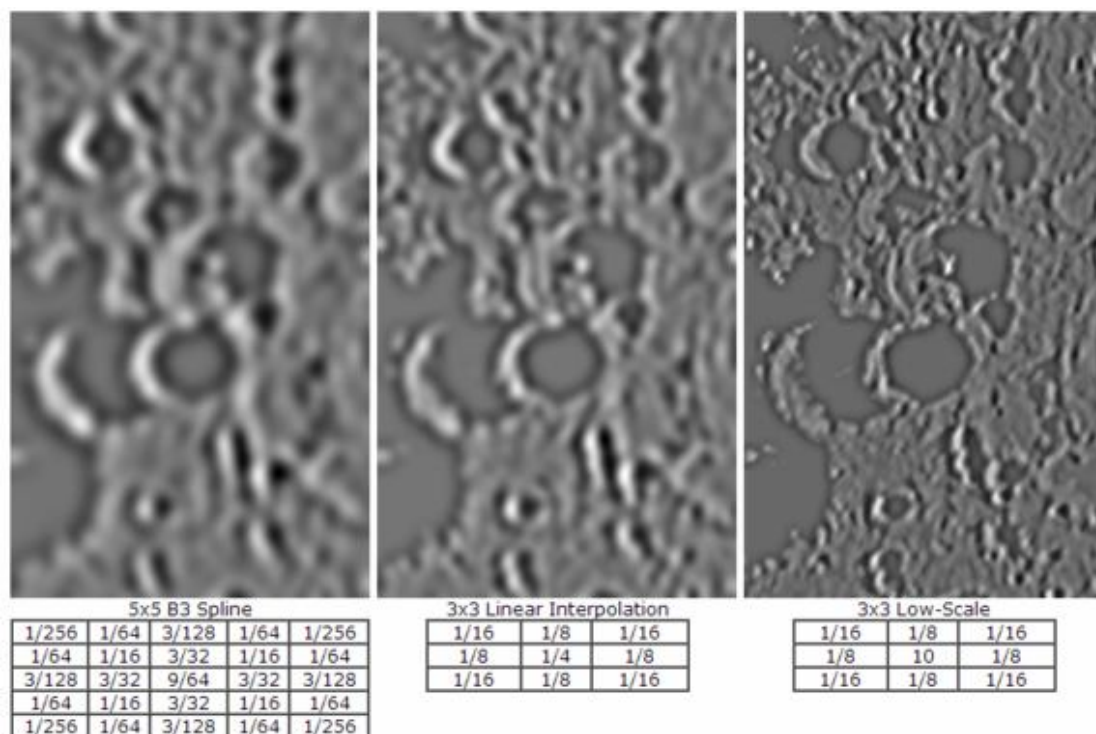
這兩個值可以分別獨立設置，共同發揮作用。

現在我們看看 ATWT 視窗的另一個選項卡：

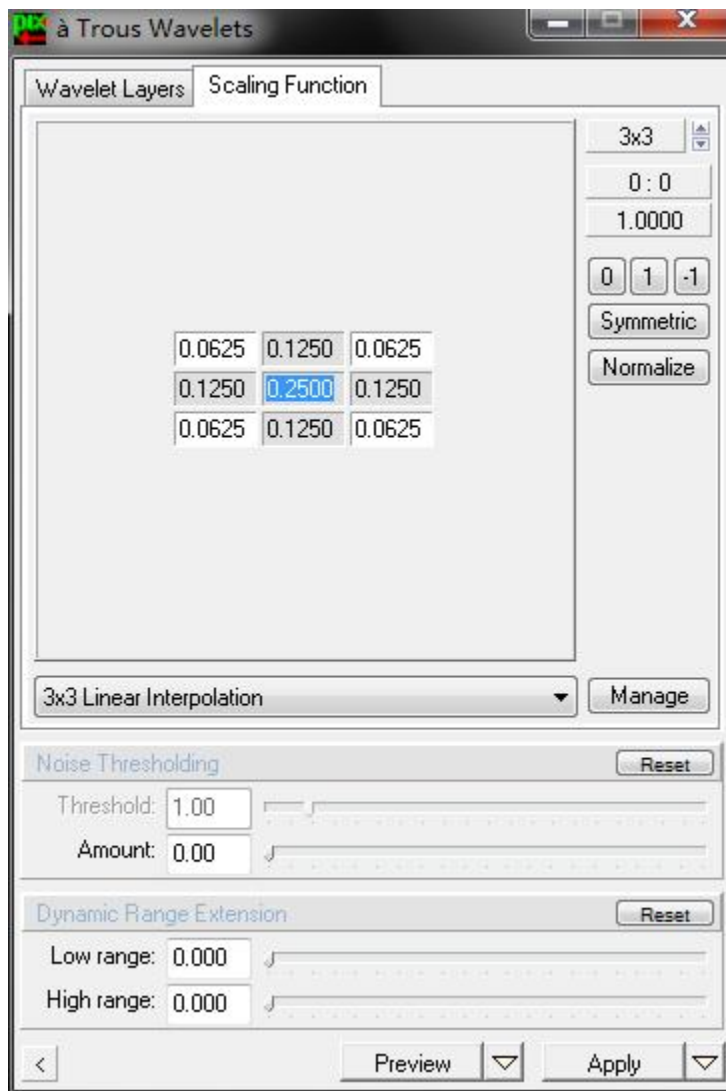
### Scaling Function

Wavelet 拆解圖像可以用不同的演算法（或者說濾鏡，在 ATWT 裏稱作 Scaling function），不同的演算法出來的結果也不同。比方說我想拆一棟房子，可以用推土機推倒，也可以用炸藥爆破。雖然最終結果都是房子成了一堆碎片，但推土機搞出來的碎片肯定要比炸藥炸出來的大。不同的演算法有各自的「擅長」，有的在分離小波長成分上效果好些，有些則在分離大波長時表現更好。我們可以針對不同需求來選擇不同的 scaling function。

一組例子，看看各種濾鏡間的不同，圖來自 PI 說明書。



接下來就是 Scaling Function 的具體操作了。它的介面長這樣：



最下面的 Noise Thresholding 和 Dynamic Range Extension 之前說過了。我們先看看 Noise Thresholding 上方那個下拉表單。裏面有五個選項，分別是 New Filter、5x5 B3 Spline、5x5 Linear Interpolation、3x3 Linear Interpolation 和 3x3 Small-Scale。New Filter 是自定義濾鏡，我們先不管他，先看看現成的四個濾鏡。

5x5 B3 Spline：最「平滑」的濾鏡，最擅長分離大波長成分。做低頻蒙版的時候一般就是用它。

5x5 Linear Interpolation：比 5x5 B3 Spline 更中性些，分離中等波長成分的效果比 5x5 B3 Spline 好些。

3x3 Linear Interpolation：最中庸的濾鏡，分離大波長與小波長的成分效果都不錯，是軟體的默認設置。

3x3 Small-Scale：最擅長分離小波長成分，對小波長成分分離地比較乾淨，較少混入大波長成分，一般用於月面與行星的銳化。



除了這四個現成的濾鏡，我們還可以自定義濾鏡。選擇 **New Filter**，我們就可以編輯上面那一大片單格。這裏的自定義濾鏡到底是什麼原理，單格裏的值如何決定演算法，說明書上沒有說。但直覺上看，它和 **Photoshop** 裏的「自定」濾鏡很相似，大家可以試著去參照一下 **Photoshop** 自定濾鏡是如何工作的。在這裏我就只講操作方法了（以下所說以我給出的截圖為例）。

看右上角有一個「3x3」，那是濾鏡的 **Kernel Size**。比如，選 **5x5**，那就有 **5x5** 的單格矩陣可以讓你填數位。

在 **Kernel Size** 下面，有一個「0:0」，表示你目前操作的單格的座標。前一個數值為橫軸，後一個為縱軸，原點為中央單格。

再往下，是一個數值，表示所有單格數值的和。

之後是三個按鈕，**0**，**1**，**-1**。點擊其中一個按鈕，所有單格的值就會變成按鈕上的值。

**Symmetric**：對稱按鈕。比如你在左上角填了一個值，按一下它，那剩下的三個角落的值就會變成你在左上角填的那個值。

**Normalize**：把所有單格的值全部歸為 **0**，**1** 或 **-1**（取決於你按了 **0**，**1**，**-1** 三個按鈕你按了哪一個）。比如你按了 **1**，**Kernel Size** 為 **5x5**，則所有單格的值歸為 **0.04**。