第三章、 方塊效應的偵測、消除和改良

因為過完備小波並沒有做次取樣的動作,所以過完備小波在頻域上的發生點會和原來時域上的發生點一樣,也就是如果有高頻訊號出現的點會等於在時域上有個大的像素值的變化,有了這個特性,我們便很方便可以去偵測每個方塊邊界的位置是否有所謂的方塊效應發生.再加上Lang 等人證明[28]小波去噪在過完備小波比在傳統的小波轉換上有好的表現.所以本文便想利用過完備小波在這方面的特性來對方塊效應做偵測和去方塊效應.

在本章中,將會說明如何利用過完備小波轉換(overcomplete wavelete transform,OCWT)來對方塊效應的偵測和消除.

3.1. 方塊效應的偵測技術

在[7]中,作者提出了一個方法,利用採用1維(1-D)過完備小波的方式來偵測方塊效應. 首先,在每一個8x8的方塊中,並不是每個方塊與方塊之間的不連續現場皆為方塊效應,有下列三種可能:

1.此不連續現象是圖像的邊緣(edge).

2.此不連續現象是圖像的灰階緩慢的變化(smooth region).

3.此不連續現象是方塊效應(blocking effect).

針對這三種不同的情況,在過完備小波轉換的高頻係數中,如果我們可以設計一套判斷機制,來判斷出以上三種情況的話,那麼就可以針對方塊效應來做適

當的處理.

以圖14來看,在圖14(a)的原來訊號(f(n))上有邊緣,突波和平滑的區域.而f(n)經過二級的過完備小波之後,分別得到第一級的高頻訊號 $w_1^d f(n)$ (圖14(b))和第二級的高頻訊號 $w_2^d f(n)$ (圖14(c)).

而因為過完備小波轉換的頻域和時域的位置是一樣的,所以在原來f(n)上的邊緣,突波和平滑的區域可以在 $w_1^d f(n)$ 和 $w_2^d f(n)$ 一樣的位置看到不同的變化.

由圖14(b)可以看到,在邊緣之處(n=50),其相對的 $w_1^d f(n)$ 和 $w_2^d f(n)$ 的值都十分大,而在突波(n=150)之處,其 $w_1^d f(n)$ 和 $w_2^d f(n)$ 的值不是極大,但其與在右的 $w_1^d f(n-1)$ 和 $w_1^d f(n+1)$ 的比例來看,是遠大於 $w_1^d f(n-1)$ 和 $w_1^d f(n+1)$ 的值,而且在小波的細節空間係數上來看,其波形都像是一個突波(pulse).

而在影像的平滑之處(n=300),其相對的 $w_1^d f(n)$ 和 $w_2^d f(n)$ 的波形都是比較像一個高起來的平原,而其與週圍的小波細節空間係數來比,似乎並沒有很大的差異.

由這簡單的圖形(圖 14)可以看出,如果對於第一級的小波細節空間係數找出一個適當的閥值(threshold,T),來判斷是否為邊緣(edge),再將不是邊緣(edge) 的點來再做一次判斷,來正確偵測出突波(也就是方塊效應)的發生點,也就是我們想要的.

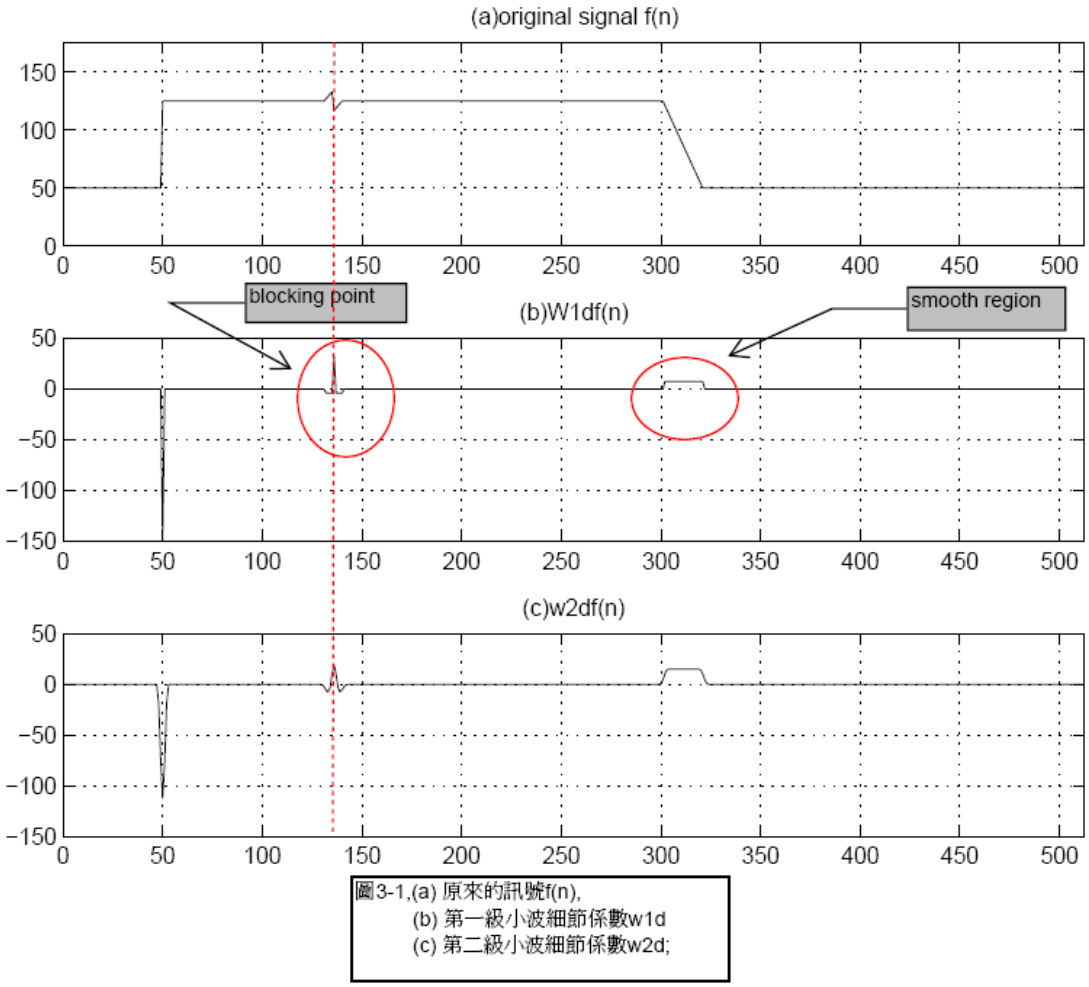


圖 14:過完備小波分析

如何計算一個閥值一直是解決方塊效應的第一條件,在[16][17]中,針對每一級的小波細節係數找到每一級的閥值,公式如下:

$$\delta_{h,j}(e) = f_a \sqrt{\frac{1}{N-1}\sum (w_j^d - m)^2} \qquad (3\text{-}1)$$

其中 N:是影像資料的大小,m :每一級小波中的平均值(mean).

w_j^d :第j級的小波細節係數, f_a :一個固定的倍數(factor)

並且[16]提出一個判斷影像好壞的量化公式,稱之為邊緣變異數(edge

variance, EV).

$$EV = \sum (i_1 - i_2)^2 \qquad (3\text{-}2)$$

i_1, i_2 :代表不同方塊相鄰的兩點.

在[18][7][19]中提出一個想法, 其中[18][19]是以二維觀點來看, 而[7]以一維的觀點來看, 但其觀念皆為相同, 其想法為每8點就是可能方塊效應發生之處, 而發生方塊效應時在小波細節係數上可以看做是原來的小波細節係數加上雜訊所引起的.

即:

$$w_1^d f_b(mM) = w_1^d f_r(mM) + w_1^d e(mM) \qquad (3\text{-}3)$$

其中: $w_1^d f_b(mM)$ 表無方塊效應時的小波細節係數.

其中M是方塊切割單位, 在本文中統一為8

$1 \le m \le N/M - 1$ N:訊號長度

再假設在每8點方塊週圍的小波細節係數的值, 通常是大於每8點內部的值, 因為大部份的8x8方塊的內部是可以看做變化比較不激烈的, 所以如果利用小波的觀點, 在小波細節係數上, 更會突顯這一點. 所以我們可以假設原來想要的小波細節係數和每個方塊中的中心值相去不遠, 即

$$w_1^d f_r(mM) \cong w_1^d f_b(mM - M/2) \qquad (3\text{-}4)$$

所以如果利用每個方塊的邊界(boundary)的灰階值的變異數(variance)減去每個方塊的中心點的變異數, 二者之差可以看做是造成方塊效應雜訊(noise)

的變異數,所以判斷邊緣(edge)與否的閥值(th)將是一個常數乘上造成方塊效應雜訊的標準差.

即:

$$\sigma^2_{w_1^d} = \mathrm{var}[w_1^d f_b(mM)] - \mathrm{var}[w_1^d f_b(mM - M/2)]$$
$$Th = 2Th1 = c\sigma_{w_1^d}; c = 5.5 \qquad (3\text{-}5)$$

所以上式(3-5)中的Th是用來判斷是否為邊緣處的閥值,

所以可以得到[7]:

$$\begin{cases} |w_1^d f_b(mM)| \geq Th & \text{edge} \\ \text{otherwise,} & \text{blocking or smooth} \end{cases} \qquad (3\text{-}6)$$

接下來要得到判斷出方塊效應或是平滑處的閥值(令為Th1)

$$Th1 = Th/2 \qquad (3\text{-}7)$$

在圖14中,可以看出當在突波發生處,在小波細節係數上突波最大點和左右兩點間是有滿大的差距,所以如果突波中心點和左右兩點的比例皆大於Th1時,可以斷定其就是個方塊效應發生之處.

$$\begin{cases} \left|w_1^d f_b(mM)\right| / \left|w_1^d f_b(mM-1)\right| \geq Th1 \text{ and } \left|w_1^d f_b(mM)\right| / \left|w_1^d f_b(mM+1)\right| \geq Th1, \\ \text{pulse region} \\ \left|w_1^d f_b(mM)\right| / \left|w_1^d f_b(mM-1)\right| < Th1 \text{ or } \left|w_1^d f_b(mM)\right| / \left|w_1^d f_b(mM+1)\right| < Th1, \\ \textit{smooth} \text{ region} \end{cases} \qquad (3\text{-}8)$$

綜合(3-8)和(3-6)可以得到判斷三種不同形式的波形.籍此來判斷出我們想要的邊緣,突波和平滑區域.

if $(|w_1^d f_b(mM)| < Th$ and $|w_1^d f_b(mM)|/|w_1^d f_b(mM-1)| \geq Th1$
and $|w_1^d f_b(mM)|/|w_1^d f_b(mM+1)| \geq Th1)$
it is pulse region
else
if $(|w_1^d f_b(mM)| < Th$ and $|w_1^d f_b(mM)|/|w_1^d f_b(mM-1)| < Th1$
or $|w_1^d f_b(mM)|/|w_1^d f_b(mM+1)| < Th1)$ **(3-9)**
it is smooth region
else
if $(|w_1^d f_b(mM)| \geq Th)$
it is edge region.
end if

3.2. 方塊效應的消除

偵測出方塊效應之後,如何去正確的消除方塊效應是個重要的課題. 在[7]中,認為小波近似係數是和方塊效應無關的,所以對小波近似係數不用做任何處理,反而是在小波細節係數上在方塊效應會有明顯的影響[16][19].所以對小波細節係數處理會有很好的去方塊效應的效果.

如果是邊緣區域(edge),常理上是不去處理的,因為要保留邊緣區域的銳利感(sharp),而如果是平滑區域(smooth),則是將第一級小波細節係數移除.即:

$$w_1^d \hat{f}_r(mM) = \begin{cases} 0, if\ f_b(mM) \text{ is smooth} \\ f_b(mM), otherwise \end{cases} \quad (3\text{-}10)$$

$$w_1^d \hat{f}_r(mM) = \underset{l \in [-1,1]}{median}[w_1^d f_b(mM-l)] \quad (3\text{-}11)$$

而在突波點,則除了第一級小波細節係數以(3-11)來執行外,

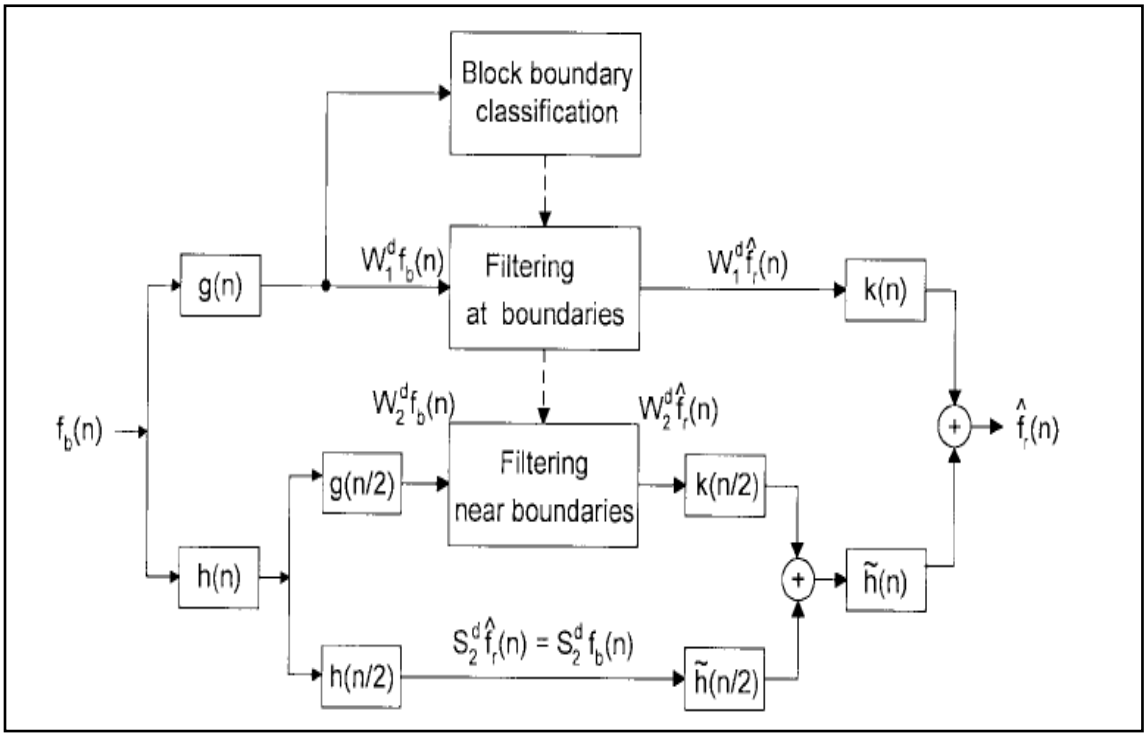
針對第二級小波細節係數也要處理,且不僅在8x8的方塊邊界,連邊界的週圍點

都要處理

$$w_2^d \hat{f}_r(n) = w_2^d f_b(n) - \hat{\alpha}(m) \cdot \gamma(n - mM)$$
$$= w_2^d f_b(n) - (w_1^d f_b(mM) - w_1^d \hat{f}_r(mM)) \cdot \gamma(n - mM)$$
$$= w_2^d f_b(n) - (w_1^d f_b(mM) - w_1^d \hat{f}_r(mM)) \cdot [0.125\delta(n - mM + 1)$$
$$+0.5\delta(n - mM) + 0.75\delta(n - mM - 1) + 0.5\delta(n - mM - 2)$$
$$+0.125\delta(n - mM - 3)]$$

(3-12)

所以[7]去方塊效應的流程圖和方塊圖如(圖15):



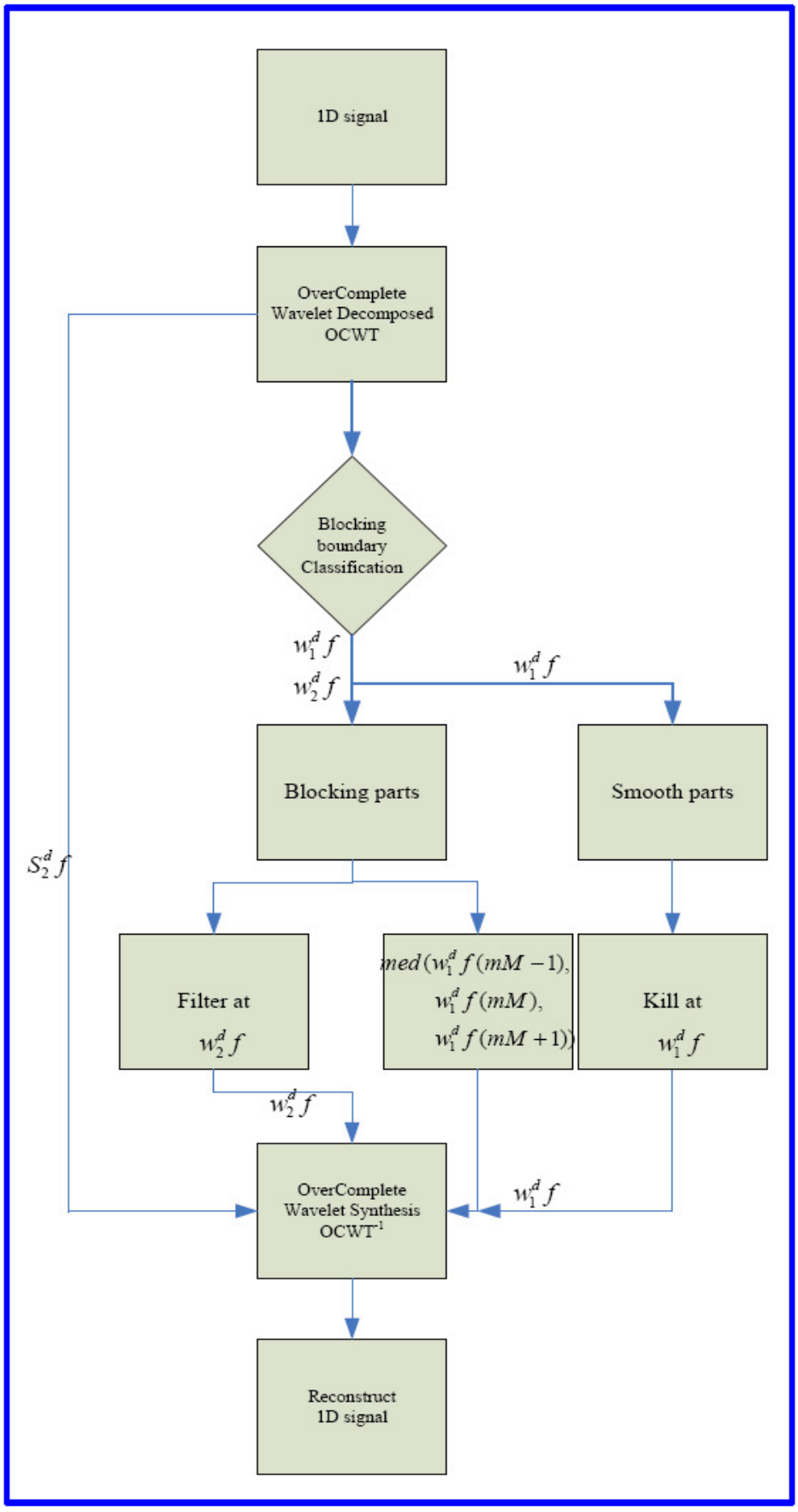


圖 15:[7] 的流程圖和方塊圖

3.3.進一步的改進Modify the Algorimth

經Matlab結果,[7]的去方塊效應效果不錯, 但是並非完全有解決所有方塊效應,見圖16.

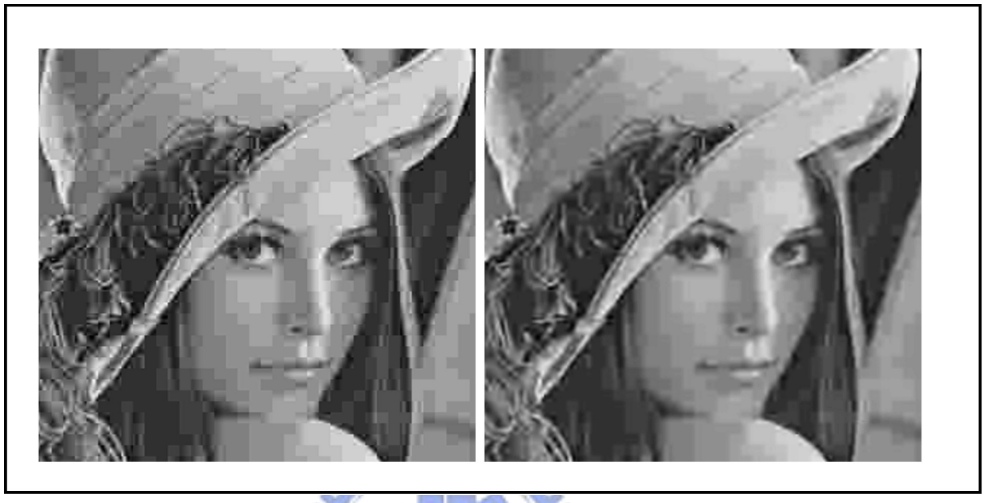
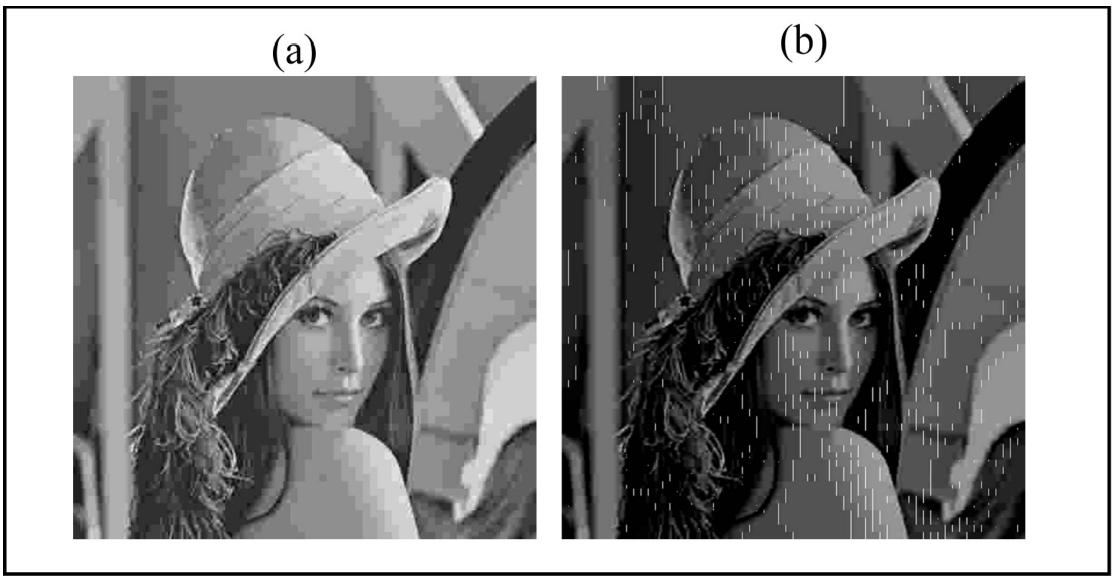


圖 16:[7] 的實驗結果

但也實驗出[7]在判斷邊緣,突波和平滑區域(圖17)有不錯的表現,所以覺得可以改良[7]的去方塊效應的方法,來加強視覺上的感覺 .

首先,方塊效應確實可以假設其是一種雜訊,不應出現在小波的細節係數上,但如果如[20][18]所提出,將判斷出來非邊緣的小波細節係數都設為0,會使去方塊之後的影像有太過模糊的結果. 小波去雜訊法是基於多重解析分析的原理。小波轉換將一個有限長度的時域信號，完美的在不同的頻寬中分解出不同的解析結果，分別是粗略的低頻部份和細緻的高頻部份。



(a)壓縮完的圖形.　　(b)方塊偵測的結果

圖 17:[7]在方塊效應上的判斷

藉由噪訊在不同頻率出現的特性,可以適度的去噪而不破壞原本的訊號.最近，小波轉換已經成為雜訊消減的強力工具之一

事實上,方塊效應可以看成是訊號加上雜訊[7],其抽像的模型如下:

$$f_b = f + e_q$$

$$e_q : quantization\ noise$$

所以如果我們可以把造成方塊效應的雜訊去掉,就可以重構成完整的訊號.

$$\hat{f}_r = D.N.(f_b)$$

其中D.N.是指去噪處理(de-noising).

因為在第二章中,在小波去噪中,Donoho提出了很有效率的方法,即對小波細節係數做去噪的處理,以達到重構成原來的訊號.所以本文中會將[7]中來做改進主要的改進為:

i.處理方塊效應的方法改成應用Donoho去噪的演算法來做改進.

ii. 改進Donoho去噪在硬體上的復雜度.

iii. 對平滑區域的處理.

以下將針對i, ii, iii做詳細的說明.

3.3.1 改進 Donoho 去噪演算法

在Donoho去噪演算法中, 最重要的就是定出要過濾掉的雜訊的閥值大小的選取。太大的閥值可能會造成信號的失真，而太小的閥值又無法有效的去除雜訊, 依據Donoho and Johnstone 所提出的小波細節係數中閥值的選取是由下式所決定[8]:

$$|Thr| = \sigma\sqrt{2\log N} \qquad (3\text{-}13)$$

其中, Thr 為閥值, σ 為雜訊的標準差;N為輸入信號的長度. 由於雜訊隨機性的包含於小波轉換後的信號中，因此無法預先知道隨機雜訊之標準差, 所以σ被取代如(3-14)所示:

$$\sigma = MAV / 0.6745 \qquad (3\text{-}14)$$

其中MAV 為某一特定尺度內訊號小波轉換中間絕對值(Median Absolute Values, MAV).

此σ為經驗值，原先被試驗於白色高斯雜訊的雜訊消除中。

而在小波轉換後的係數中, 在[21][22][7][19]中, 皆認為不只第一級小波細節係數須要使用去噪, 第二級也是須要被使用去噪的方式. 因為方塊效應的雜訊不應只有存在第一級小波細節係數, 也會在第二, 第三級的高頻地方出現, 所以我們也

會對第一, 第二級的小波細節係數分別做去噪的動作.

在第一級小波細節係數中, 可以看出來只要針對每個8x8方塊邊界的點做去噪即可, 因為附近的相鄰點係數是十分小的, 保留附近的係數, 可以避免過度模糊且保有本來圖像的銳利度, 而在第二級中可以由圖14看出, 因為小波的多解析度, 會使愈來愈多的高頻訊號出現, 而會有由中心點往兩旁擴展的現象, 所以如果只有針對8x8方塊的邊界來做第二級的去噪的話, 效果一定不好, 所以我們在第二級的小波細節係數去噪的點除了8x8方塊的邊界之外, 左右點也是我們去噪的範圍.

綜合上述的改進點, 我們可以改進[7]的方法, 經實際在Matlab上實驗的結果, 對去方塊效應有不錯的效果, 將matlab程式流程簡示如下:

do **(pulse region)**

{

$$Thr_p = \underset{m\in\{\text{pulse region}\}}{\textbf{median}}\left(|\mathbf{w}_1^{\mathbf{d}}(mM)|\right)\bullet\sqrt{2\bullet\log(length(m\in\{pulse\ \textbf{region}\})}/0.6745;$$

$$\hat{w}_1^d(mM) = \textbf{wthresh}(w_1^d(mM),'s',Thr_p);$$

$$\hat{w}_2^d(mM-1) = \textbf{wthresh}(w_2^d(mM-1),'s',Thr_p);$$

$$\hat{w}_2^d(mM) = \textbf{wthresh}(w_2^d(mM),'s',Thr_p);$$

$$\hat{w}_2^d(mM+1) = \textbf{wthresh}(w_2^d(mM+1),'s',Thr_p);$$

}

其中wthresh是做軟閥值去噪. 而最重要的Thr_p主要是由判斷是方塊點的第一級小波細節係數的點來做中位絕對值(MAD)的運算來求出.而第二級的小波細節係數也是採用Thr_p來做軟閥值去噪.

我們把[7]中單純只處理方塊效應的結果和本文單純只處理方塊效應的結果做一個比較.

(a).[7]只去方塊的結果

(b).本文只去方塊的結果

圖 18:本文去方塊效應和[7]的比較

由上圖(圖18)可以知道,經過Donoho去噪後的效果比原來[7]的去方塊的效果有很大的改進.雖然Donoho 去噪會有過度平滑的問題,但因為我們是先做方塊效應的分類,只去被偵測屬於方塊的地方做Donoho去噪的處理,所以對於不是方塊的其他地方,我們並沒有改變,也就是說整個影像是十分完整的,只有被偵測屬於方塊的小波細節係數會被處理,而不會破壞整個影像,這也是去方塊效應所應該要做到的事.

綜合所述,本文在去方塊的方法上主要是採取Donoho去噪,但是跟第二章的有以下不同:

A. 對第二級的去噪不針對發生方塊效應的點,左右點也會同時去噪.

B. 只對方塊效應發生的點來估算去噪的閥值和去噪的處理.

3.3.2 改進 Donoho 去噪演算法的硬體複雜度

要做Donoho去噪要先計算出要消除方塊效應的閥值, 消除方塊效應的閥值

的求的主要是由下列演算法:

$$
\begin{aligned}
&|Thr| = \sigma\sqrt{2\log N} \\
&\sigma = MAV/0.6745 \\
&N : signal\ Length
\end{aligned}
\qquad (3\text{-}15)
$$

下圖(圖19)來表示Donoho去噪的示意圖和流程圖(圖20).

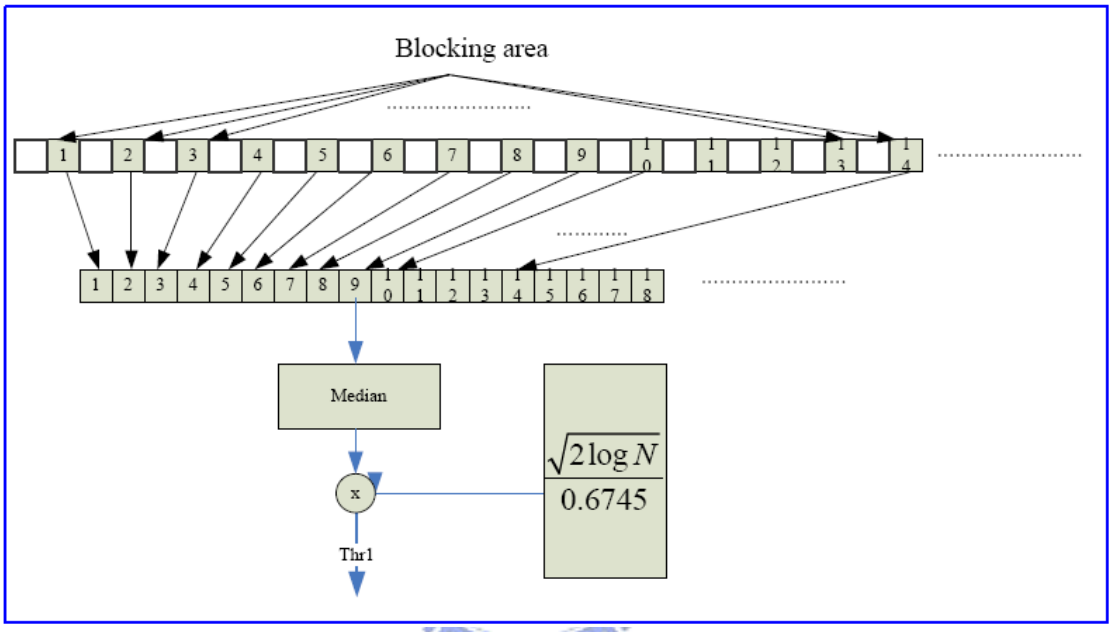


圖19: Donoho閥值求法示意圖

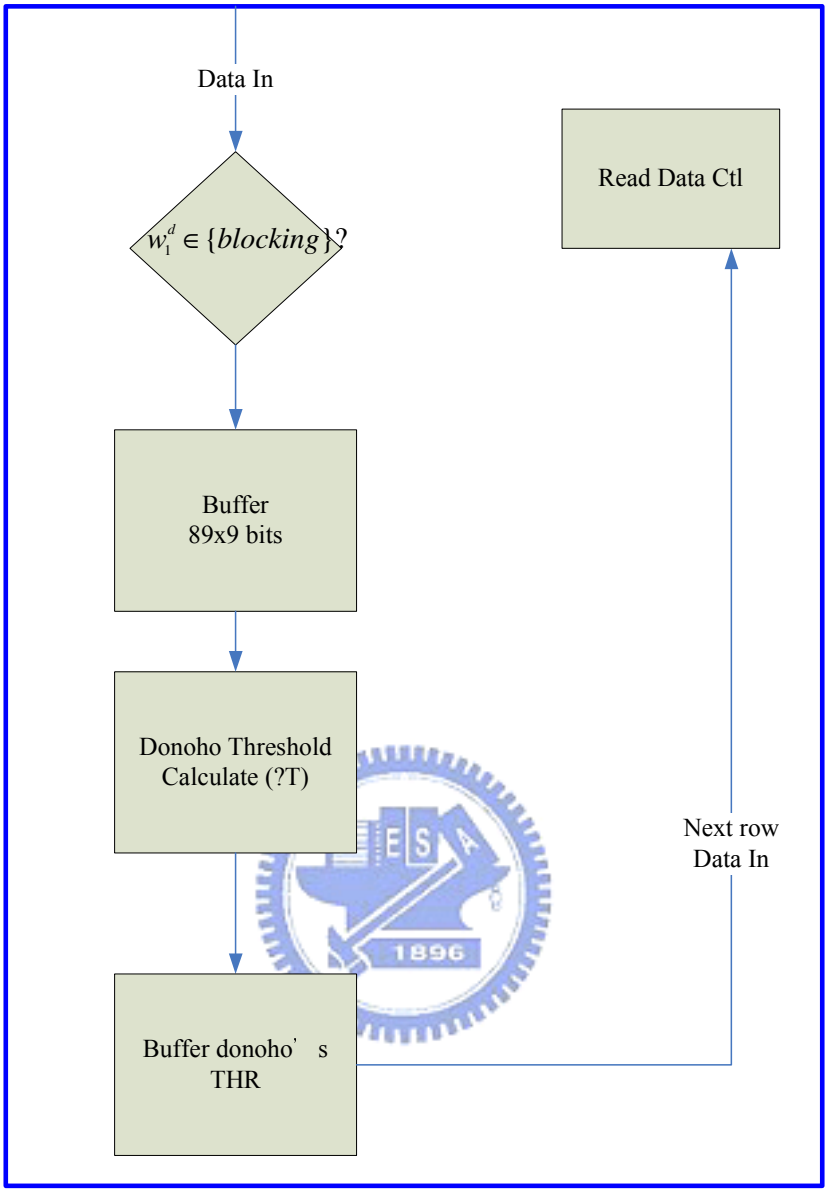


圖20：Donoho閥值求法流程圖

如果由上述求法來看，以硬體的觀點其複雜度是十分大的，主要是

(i). N值是一個不定數，其值可能是介於0~89(以D1(720x480) 的大小來看)，且要求logN的結果，這是硬體很大的負擔.

(ii). 要先求出其中間值. 要由排序來做，資料量如果是最多89的話，將會花費很大的硬體資源.

由(i),(ii)來看,donoho閥值似乎對硬體而言是不可能的任務,但是其在我們的演算法中卻是不可或缺的重要因素,所以有下列兩種做法.

(a). 以軟體搭配進行.

(b). 強行以硬體來做.

以(a)而言,其軟硬體搭配來做的話,因為我們須要滿足每秒30張幀率(frame rate),如果以處理一張D1大小的影像而言,因為我們是採一維處理,所以如果外部時脈為100MHz,則我們可以算出處理一列所須最大的時鐘數為:

$$(\frac{100\times10^6}{30}\div(720+480))\times0.9=2499(T) \qquad (3\text{-}16)$$

上式可以看出,我們在硬體上只處理一列的資料,最多只可以用到2499個clock,上式會乘上0.9主要是因為外部的時鐘源一般會有10%的偏差值.

所以如果以軟硬體來處理求出的Donoho閥值的話,硬體要先儲存最大89筆資料,再交由軟體來做中位數和閥值的產生,這對硬體和軟體都是十分大的負擔,且在有限的資源(金錢,chip大小..)都是不可行的.所以除非我們不做每秒30張幀率(frame rate),不然不可能採取(a)的作法.

以(b)而言,其成本又太大,所以我們要找出其最費硬體資源的關鍵點,由(i),(ii)來看,關鍵點都是資料量十分大,最大會有89個資料要處理,再加上要做中位數值的運算,還要乘上一個變數項.($\sqrt{2\log N}/0.6745$,其中N:0~89),這以現在的VLSI技術而言,都是一個十分大的負擔,而且以去方塊效應在整個影響壓縮中(以MPEG4 為例)只是佔了一小塊的地位而言,其他重要的如移動估測

(motion estimation), 移動補償(motion compensation)…等都會比去方塊重要, 所以不可能將去方塊在chip上做的太大, 也就是要能盡量小.

所以我們決定採取各個擊破的方式, 即將每S個資料當做一個群集, 對這S個資料來做Donoho的閥值運算, 當然這個S數目須滿足下列2點.

(i) 不能太小, 因為donoho是一個統計估算值, 如果S太少, 會影響其結果, 因為會使不是噪音的係數也會被當成噪音而被過份去除(over-delete). 會造成影像的過度模糊.

(ii) 也不能太大, 超過硬體的負擔,

綜合(i),(ii)兩點, 我們以S=15, 13, 9, 7來和原來的演算法來做比較(圖21)希望能在其中找出平衡的數字.

圖21是原來針對lena, baboon和barb這三個不同的圖像, 以不同的S(S=15, 13, 9, 7)所得到donoho閥值, 和原來演算法所求出的donoho閥值所產生的平均誤差率. 而這平均誤差率的算法為(以512x512的影像為例) :

$$AvgErrRate = \frac{1}{512}\sum_{i=1}^{512}\sum_{j=1}\frac{(R_i th - R_i ths_j)}{R_i th} \qquad (3\text{-}17)$$

上式中的$R_i th$是指在第i列沒有分段(segment)的時候, 由Donoho演算法求出的閥值, 而$R_i ths_j$則是表在第i列時, 利用分段所求出的第j個閥值, 由於對於每一個列中會分成幾段是無法預先得知的, 所以上式中對j並沒有上限.

而第i列的誤差率的算法為:

$$\sum_{j=1}\frac{(R_i th - R_i ths_j)}{R_i th} \qquad (3\text{-}18)$$

最後把每一列(在此例中為512列)的誤差率做一個平均,這也就是我們本文中所謂的平均誤差率.

基本上,我們要選的分段數S所造成的平均誤差率不希望大於0.5,因為大於0.5表示有一半以上的列其平均誤差率差了一倍,這就會造成跟原來不分段的donoho閥值差太多了. 我們用三個不同的圖像(barb,baboon,lena)來做統計,希望能找到一個可以滿足我們想要的分段值S.

由圖21可以看出,誤差率會隨S的縮小而變大,為了維持我們演算法的精準度,我們要選的S大小不可小於9.再加上實際上在Matlab上模擬的結果,和硬體複雜度做選擇之下,在我們的硬體中,我們採取S=9的方法.

所以重改我們的求閥值演算法如下:

$$\begin{aligned}
&w_1^d \tilde{f} \in \{blocking\ area\}\\
&for(m=1; m<blocking\ length; m=m+9)\\
&\sigma_m = med(|w_1^d \tilde{f}((m-1)\times 9)|, |w_1^d \tilde{f}((m-1)\times 9+1)|, ...|w_1^d \tilde{f}((m-1)\times 9+8)|)/0.6745\\
&Ths_m = \sigma_m\sqrt{2\log N_m} = MAV_m * \sqrt{2\log N_m}/0.6745\\
&end
\end{aligned} \qquad (3\text{-}19)$$

在式(3-19)中, $\sqrt{2\log N_m}/0.6745$ 因為 N_m 是介於0至9之間,數值不大,所以可以用查表的方式求出不同的 N_m 之下所得到的值,而且我們把得到的中位數值(MAV)和查表的值之間的乘法器用位移器(Shifter)來取代.

又因為我們的資料是一筆一筆讀進來的,所以我們只要做一組分段式的

donoho去噪演算法的硬體即可. 而求出的每段的閥值只會應用到那一段的小波細節係數上.所以在本小節中,我們為了節省硬體的大小和在不過於影響原來Donoho的性能找出一個平衡點,所以我們將用硬體實踐**分段式Donoho去噪演算法**.

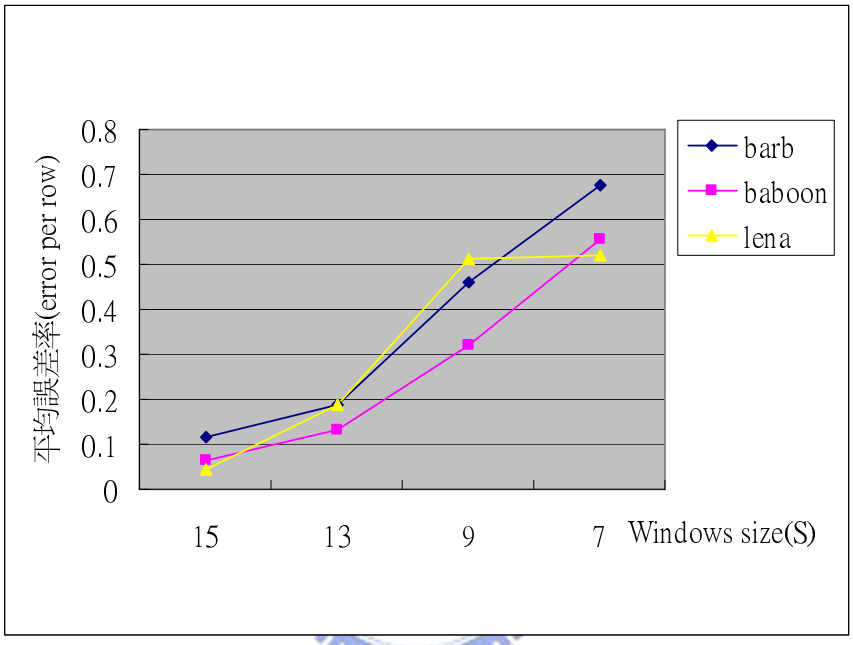


圖 21:比較不同 S 所得之平均誤差率

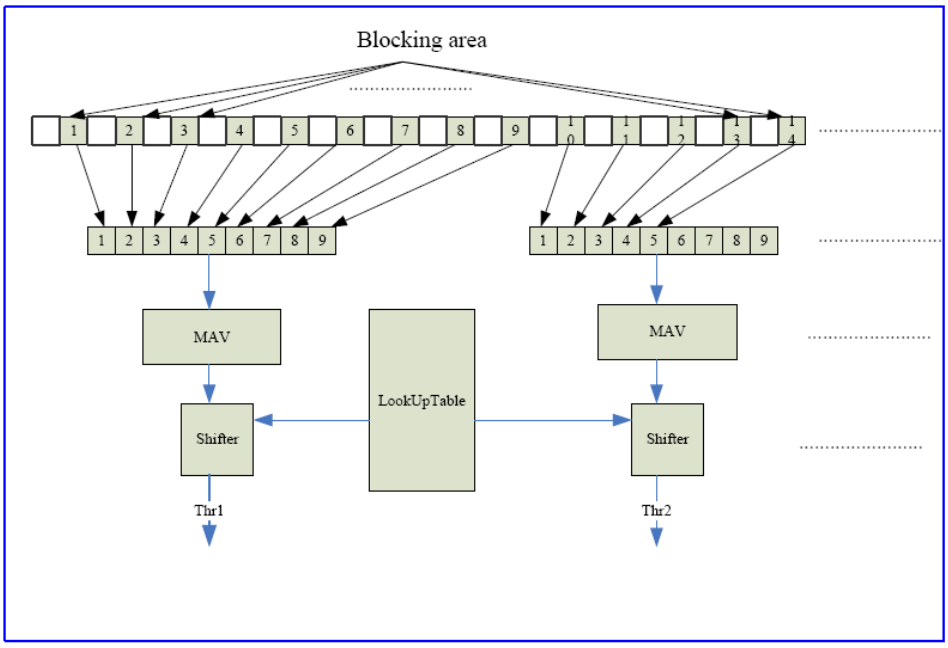


圖 22:改良式 Donoho 閥值演算法示意圖

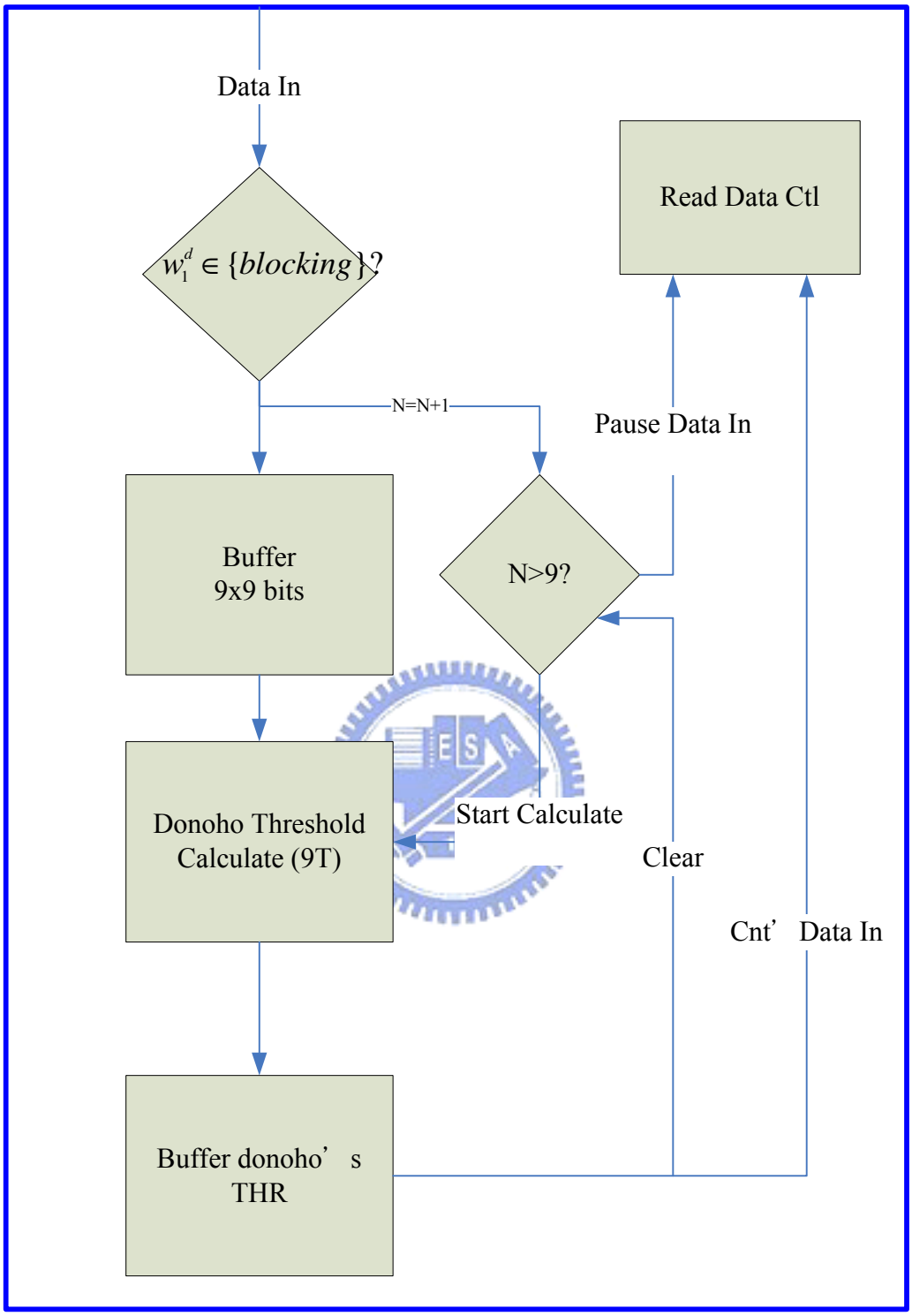


圖 23:改良式 Donoho 閥值演算法流程圖

圖 23 秀出了本文在修改 Donoho 閥值演算法的流程圖,可以看出,由於資料是一筆一筆進入處理的,所以我們在做 Donoho 處理時,每 9 筆屬於 blocking region 的第一層小波細節係數被收集之後,則會先計算這 9 筆資料的中位數,同時

也暫時停止資料的流入,等計算完了之後,就會再度啟動讀入資料的動作,雖然會因為計算 Donoho 閥值而中斷資料的讀入,但因為以 720x1 的資料量為例,最中只會有 10 個 Donoho 閥值,而每計算一次 Donoho 閥值也只要 9 個週期時間,所以花在計算 Donoho 閥值的時間上,每一組資料(720x1)最多也只有 90 個週期,所以在硬體上而言是值得的且可行的.下表 2 列出原本計算 Donoho 閥值和修改後分段式的 Donoho 閥值的比較.

	Median filter buffer size	Time to get Median	$\left[\frac{\sqrt{2\log N}}{0.6745}\right]$	PSNR (lena)
Donoho scheme	89x9 (for D1 size)	>9x2T	ROM	29.5682
Modify donoho scheme	9x9	9T	4x1 LUT	29.5198

表格 2:Donoho 閥值修正後的比較

由表 2 可以看出分段後的硬體暫存大小比原來的小了約 10 倍,而在每次計算閥值的時間上原本的中位數計算是很費時的,但修改後的是每次計算都是 9 個週期,而在得到中位數之後,要再乘上 $\frac{\sqrt{2\log N}}{0.6745}$ 才是最後的閥值,如果是原來的方法的話,因為 N 是介於(0~89)的數,所以須要一個 ROM 來做查表,而修改後的方法剛因為 N 只介於(0~9),再經過我們的四捨五入之後,只會有 4 種值,而且不用乘法器,只要位移器和加法器即可.最後可以看到,對量化的 PSNR 上,兩個方法並沒有什麼差別.

3.3.3 對平滑區域的處理

在[7]中, 我們可以看到, 除了方塊效應的地方, 在被判斷為平滑處的地方是將該處的第一級小波細節係數移走, 也就是變為0, 但是其實在平滑區域, 其小波細節係數會是呈現均勻起伏, 由圖24可以看出在平滑區域應該是會有類型台地的現象, 而因為噪音是會影響小波細節係數, 所以如果在平滑區域被噪音影響, 會破壞台地的現象, 所以我們就為了平滑這個現象, 可以用一個平滑濾波器在這平滑區域, 讓其可以盡量回復成原來平滑的樣子. 下面是我們對平滑區域所做的處理:

$w_1^d \tilde{f} \in \{smooth\ area\}$

do **(smooth region)**

{

$$\hat{w}_1^d f_r(mM) = \underset{m\in\{\text{smooth region}\}}{\textbf{median}} (\mathbf{w}_1^{\mathbf{d}} \tilde{f}(mM-1), \mathbf{w}_1^{\mathbf{d}} \tilde{f}(mM), \mathbf{w}_1^{\mathbf{d}} \tilde{f}(mM+1));$$

}

由上式可以看出, 我們對判斷成平滑區域的點做3點的中值濾波器.

所以這一小節說明了我們在平滑區域做的改進是**用一個3點的中值濾波器來對第一級的小波細節係數做平滑處理.**

綜合這一節的描述, 我們對[7]做的改進有下列幾點:

1. 在方塊的區域:

I. ***不只對第一級小波細節係數做去噪, 也對第二級的去噪不針對發生方塊效應的點, 左右點也會同時去噪.***

II. ***只對方塊效應發生的點來估算去噪的閥值和去噪的處理.***

2. 在Donoho閥值方面:

III. *實踐分段式Donoho去噪演算法*

3. 平滑區域:

IV. *用一個3點的中值濾波器來對第一級的小波細節係數做平滑處理.*

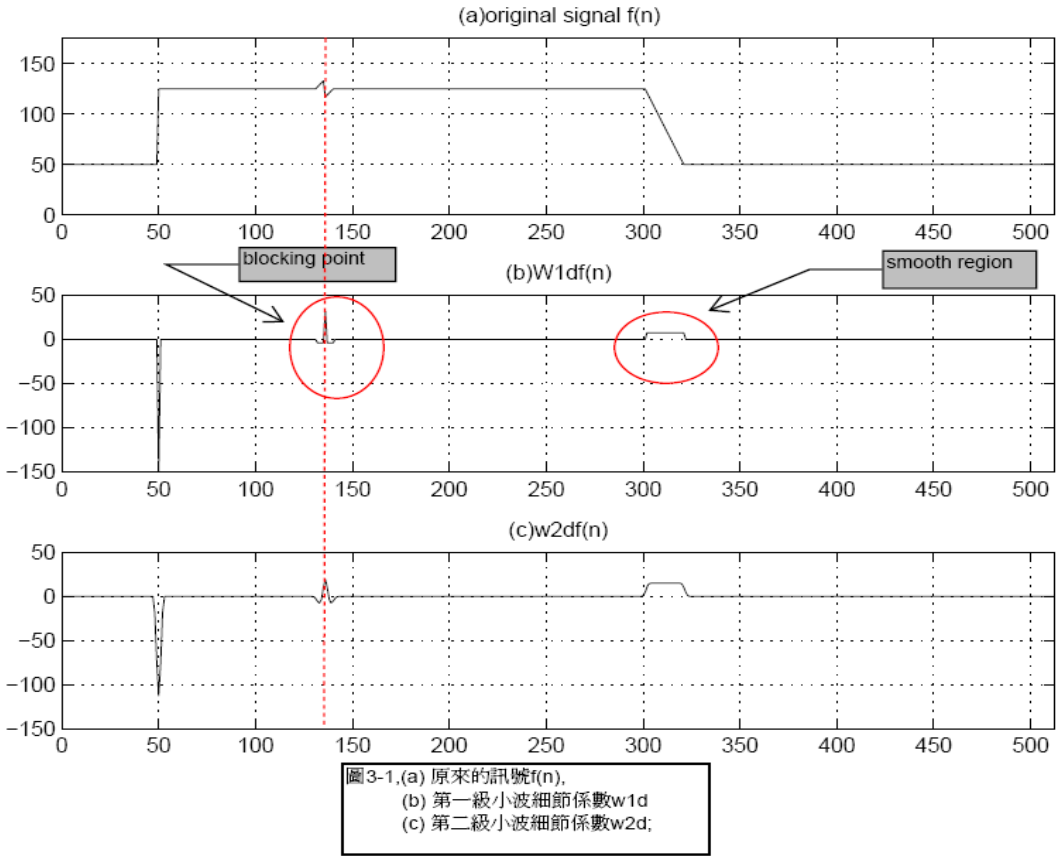


圖24:過完備小波特性分析