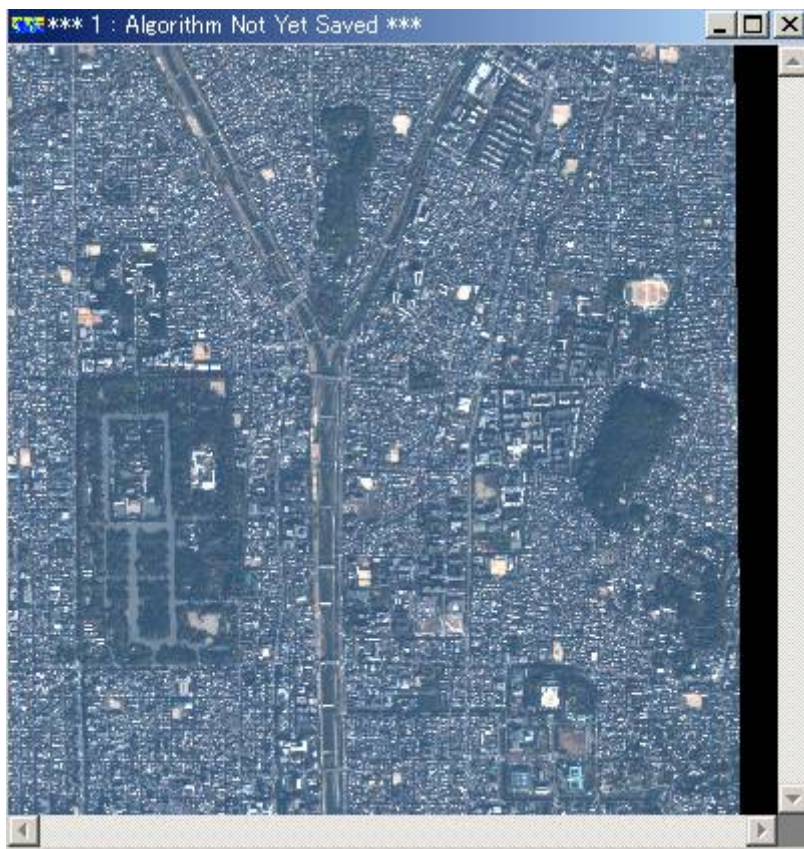
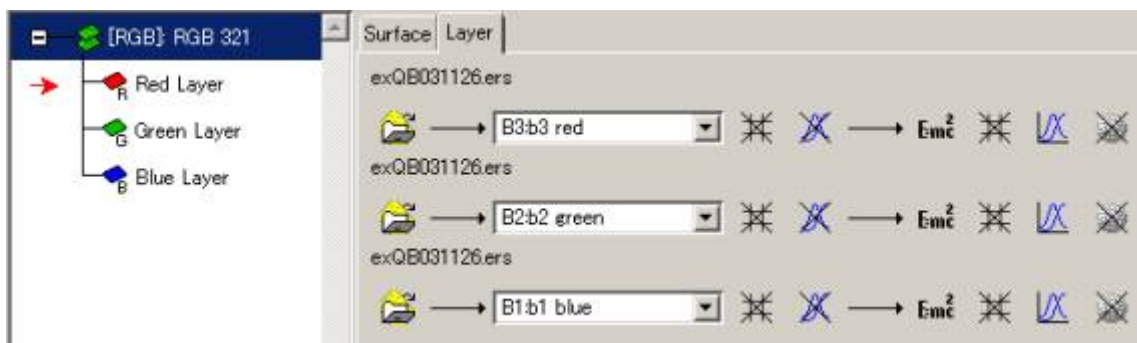



■実習課題「フィルター演算」

1. ローパスフィルターによる演算
 - 1.1. 自分の名前のフォルダーの中にex10 フォルダを新しく作ります。
 - 1.2. ex9 フォルダ内のexQB031126.ers を開きます。

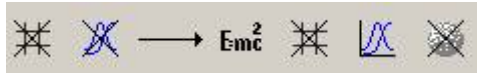



- 1.3. アルゴリズムウィンドウを開きます。





- 1.4. 赤色のレイヤの矢印の後ろのフィルター（あるいはカーネルと言います）のアイコン  をクリックします


メモ：



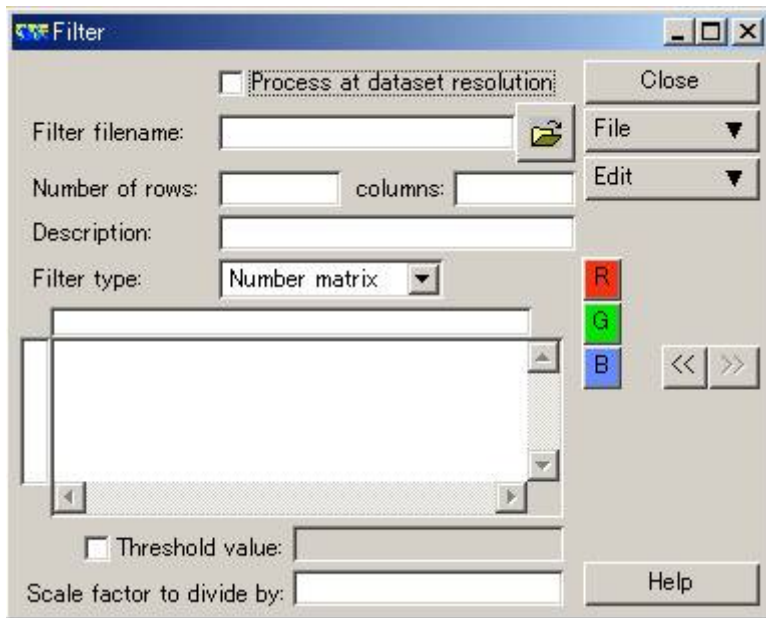
矢印の前と後ろに  がありますが、違いはフォーミュラ（数式）演算を実行する順序です。


矢印の前の  を利用すれば、フィルター演算の結果にたいしてフォーミュラ演算を実行することになります。

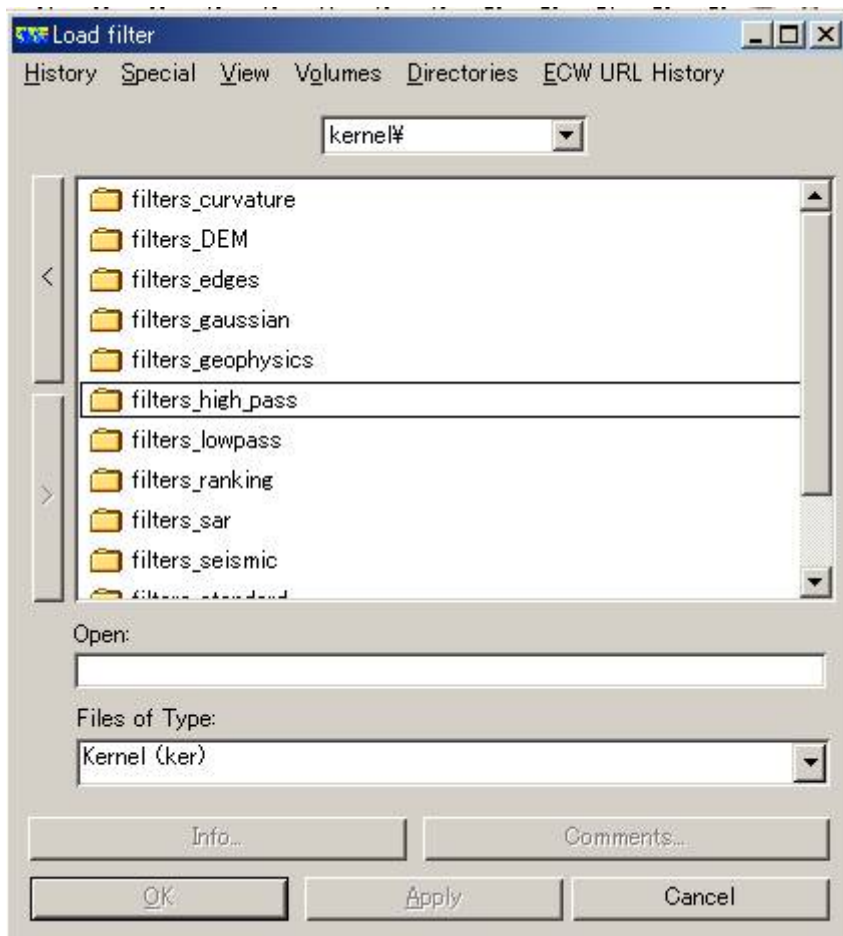
一方、矢印の後の  を利用すれば、フォーミュラ演算の結果にフィルター演算を実行することになります。

輝度変換のアイコン  も矢印の前後にありますが、使い方はフィルター演算の場合と同じです。

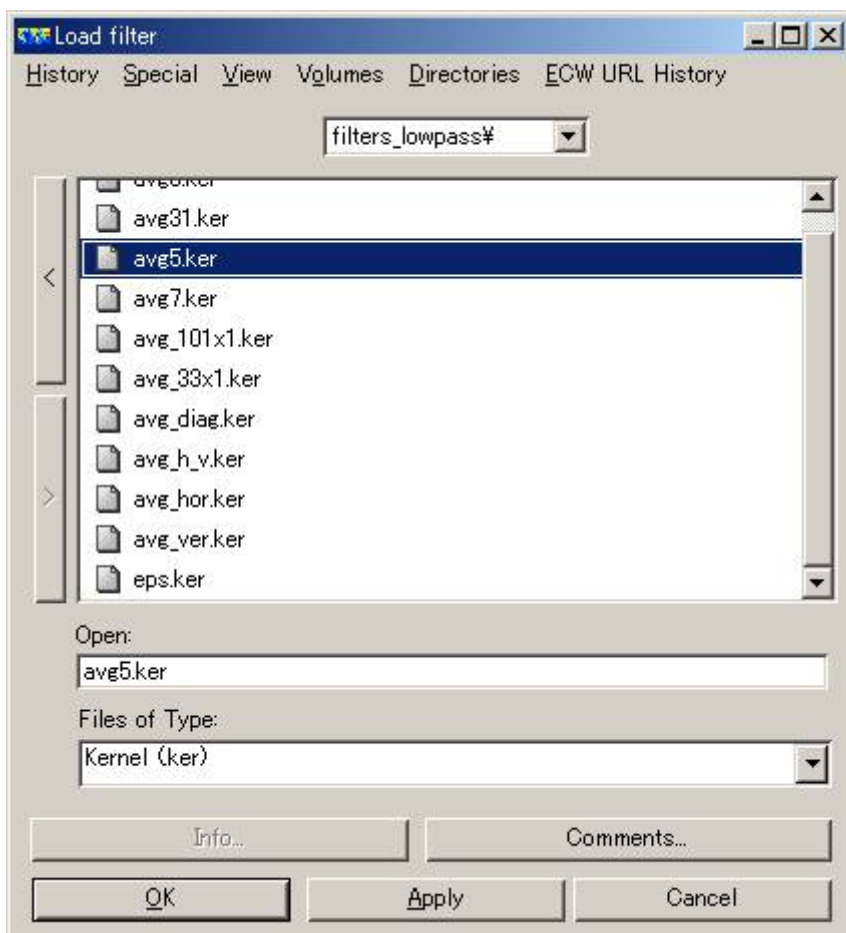
フィルターウィンドウが現れます。



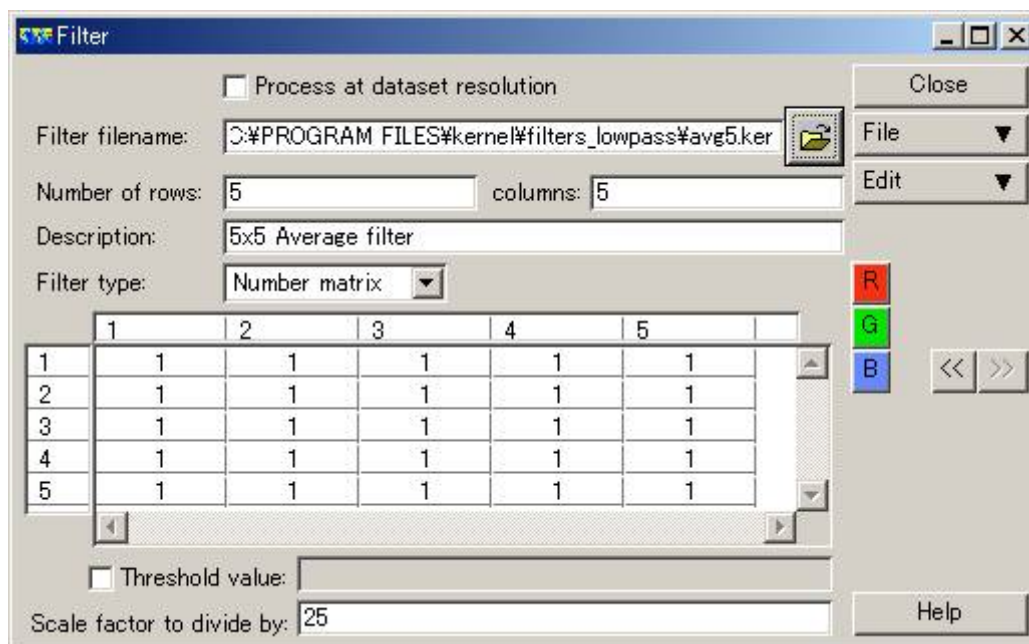
- 1.5. フォルダーのマーク  をクリックし、既存のフィルターを選択します
kernel フォルダーの中にたくさんのフィルターが用意されています。



filters_lowpass フォルダ内の avg5.ker を選択し、OK を押します。



フィルターウィンドウ内は次のように変わります。



フィルターの大きさが5 ×5 で、それぞれのピクセルに1ずつの均等な重みをかけていることがわかります。

Scale factor to divide by のところで、最後に25 で割って平均値を求めています。

このフィルターは、5 ×5 の算術平均フィルターです。

1.6. 緑色のレイヤや青色のレイヤにも同じフィルター演算を設定します。

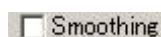
メモ：



を利用すれば、簡単にレイヤ間を移動できます。

1.7. フィルター適用の結果を確認しましょう

アルゴリズムウィンドウのsmoothing のチェックマークを外します



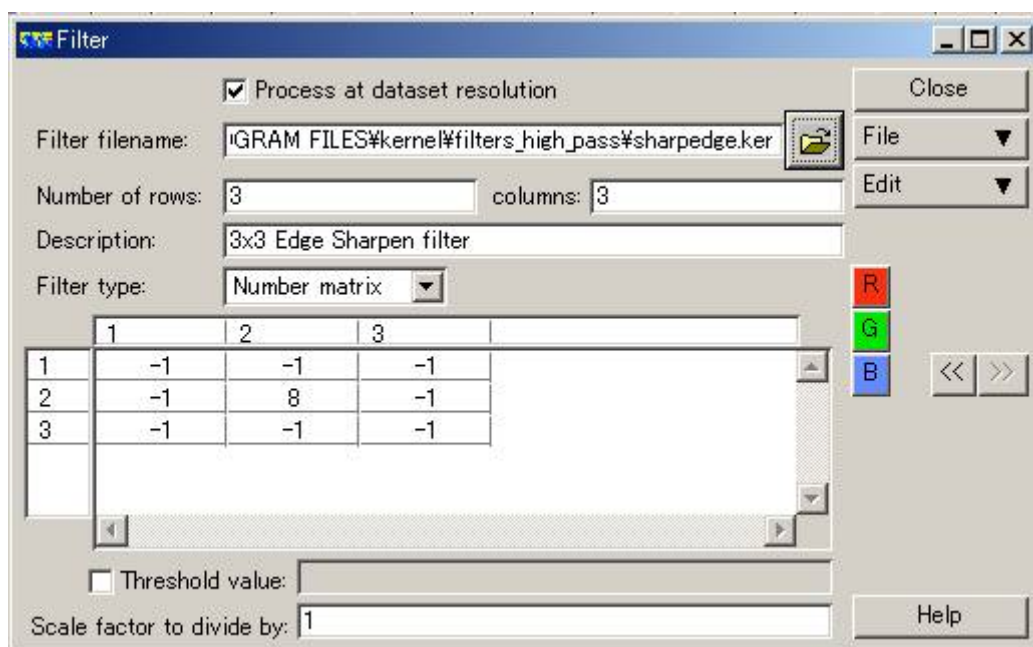


画像が平滑化されていることがわかります。

2. ハイパスフィルターによる演算

2.1. 同様の方法でハイパスフィルターを適用してみましょう

例えば、filters_high_pass フォルダ内のshapedge.ker をロードしてみましょう。

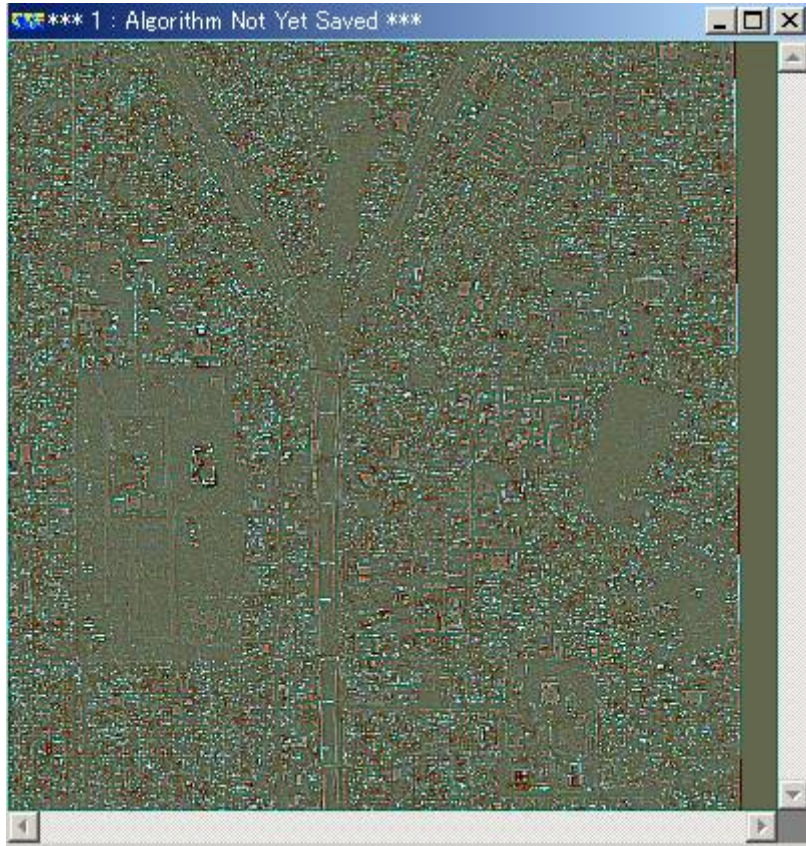


中心のピクセルには8 を、周囲のピクセル値には-1 をかけて、足し合わせていることがわかります。

これは3 ×3 のエッジ強調フィルターと呼ばれるフィルターです。

またはラプラシアンフィルターと呼ばれます。

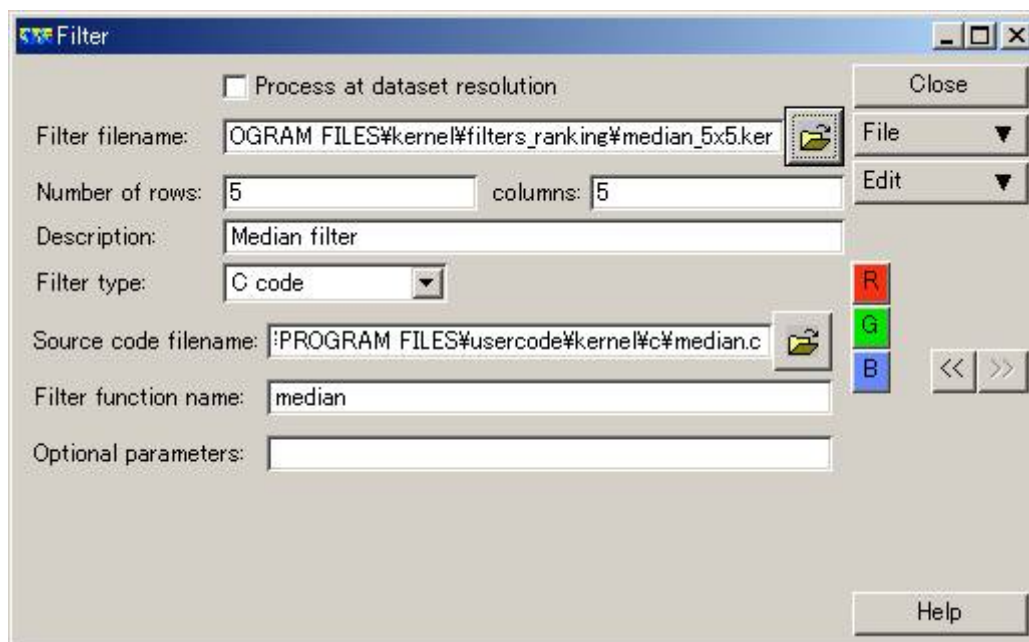
このフィルター演算を他のレイヤにも適用して画像を確認してみましょう。



画像のエッジだけが抽出されたことがわかります。

3. メディアンフィルターによる演算

3.1. 同様の方法で、filters_ranking フォルダ内のmedian_5x5.ker をロードします。



このフィルターはメディアン（中央値）フィルターと呼ばれ、5 × 5のカーネル内の中央値を

当該ピクセルの値として出力します。

メモ：

これまでのフィルターでは、マトリックスが表示されていましたが、Filter type がC code、

C 言語によるプログラムとなっていることに注目してください。

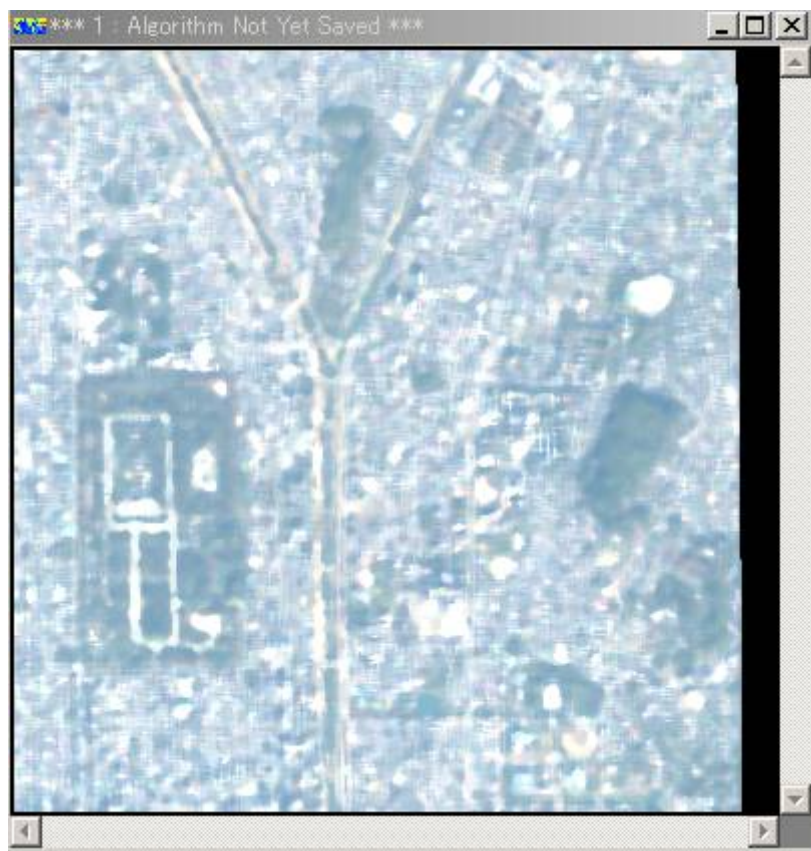
このプログラムのソースコード（人間の読める形のプログラム）は、

Source code filename に入っています。テキストエディタで開けば読めるので、興味のある人は覗いてみてください。

C 言語によるフィルターを自作することもできます。

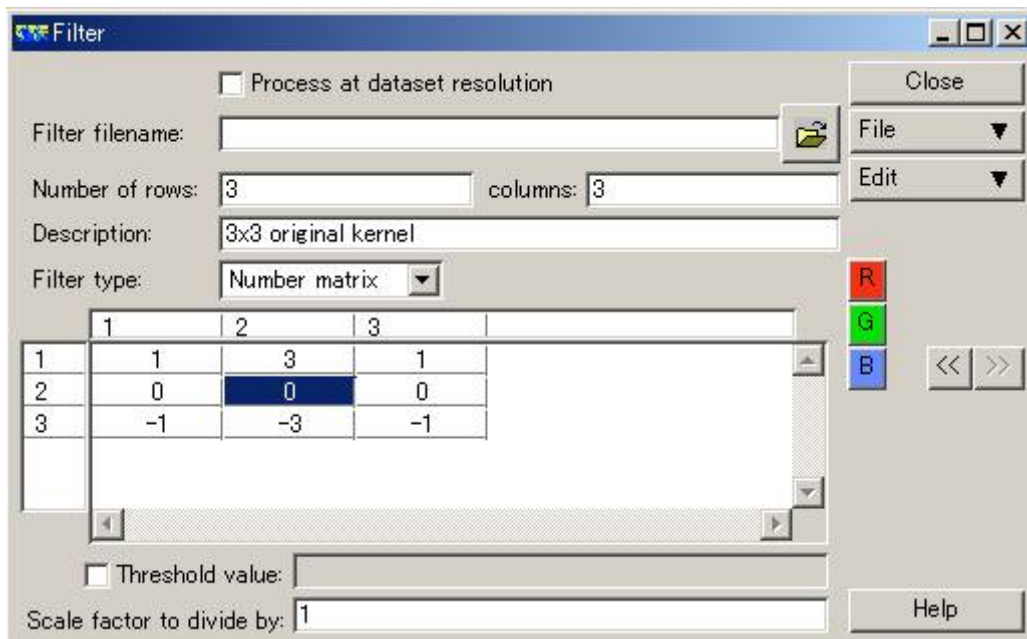
ただし、Visual C++ によりコンパイルする必要があるようです。

すべてのレイヤに同じメディアンフィルターを適用し、結果を確認してください。

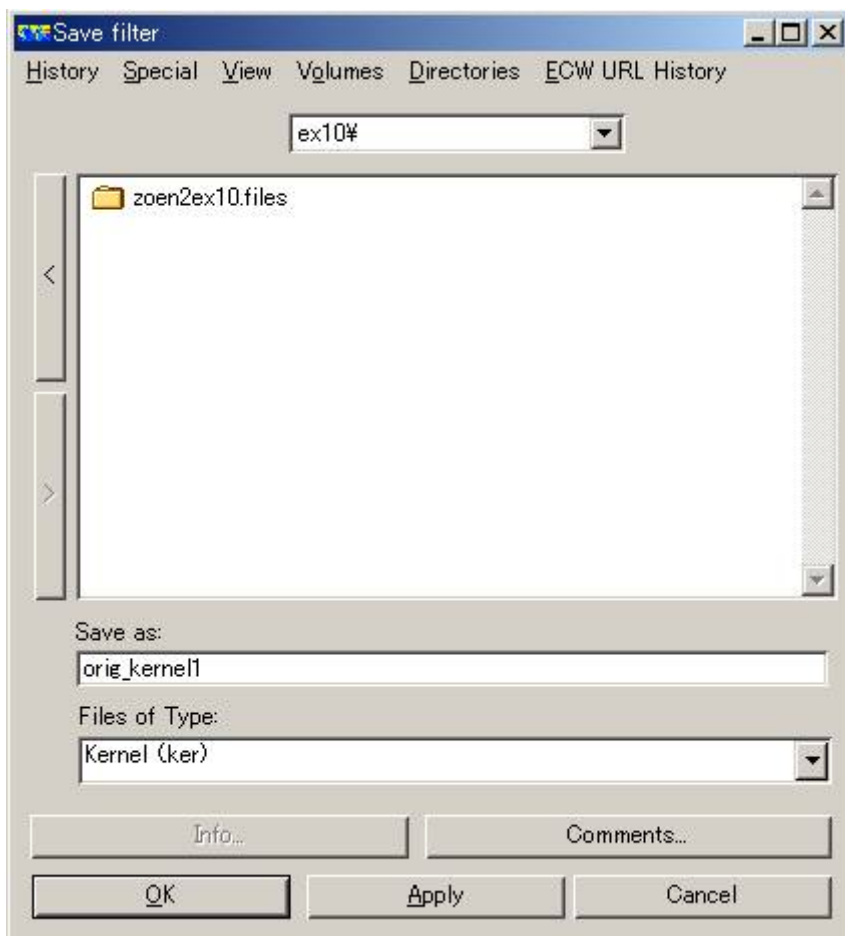


平滑化されていることがわかります。


4. 自作フィルターによる演算
 - 4.1. フィルターウィンドウのedit からdelete this filter を実行します。
 - 4.2. Filter type をNumber matrix に設定し、
Number of rows とcolumns にそれぞれ3 を指定し、
3 ×3 のカーネルにします。
マトリックスに好きな数字を入れ、
Scale factors で出力される数値の大きさを適宜調整します。



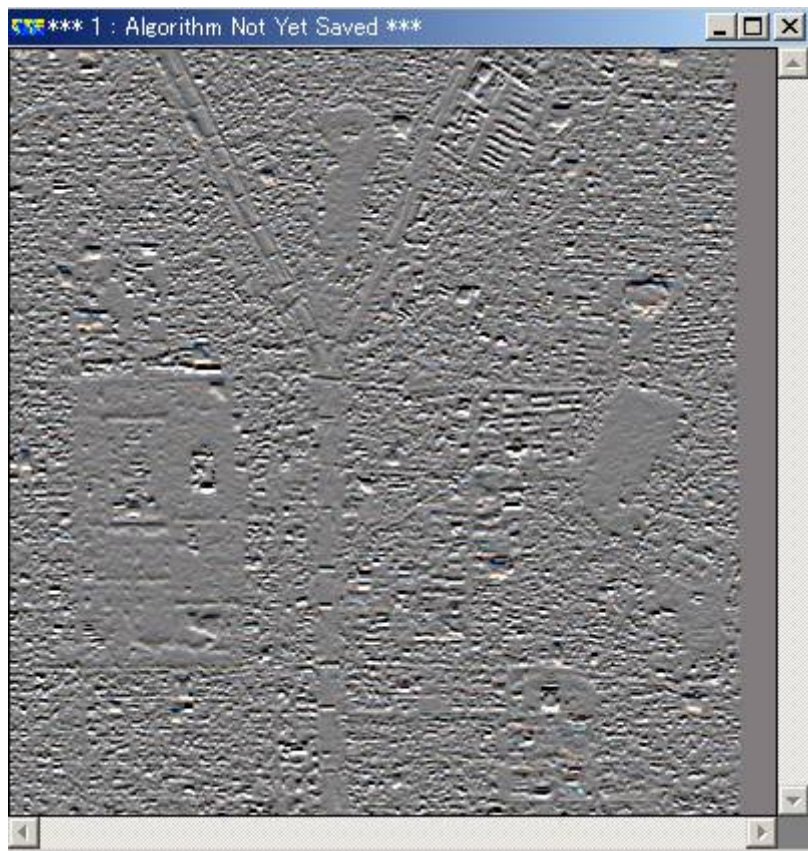
- 4.3. フィルターウィンドウのfile からsave as を実行し、
自分のex10 フォルダー内にorig_kernel1 という名前で自作フィルターを保存します。



- 4.4. 他のレイヤにも自作フィルターをロードして適用します。

フィルターウィンドウ上の  をクリックして、ex10 フォルダーまで移動し、自作フィルターを選択して、ロードします。

4.5. 自作フィルターの結果を確認します。



横方向（x 方向）のエッジが強調されていることがわかります。

■今日の提出課題

1. ERMapper にはさまざまなフィルターが用意されています。好きなフィルターを4つ取り上げて、

フィルター演算の結果とその特徴について簡単に考察しなさい。このとき、フィルターを適用するバンドにも注意を払いなさい。

2. Quick Bird 画像の空間解像度はマルチスペクトルセンサーで約2.5 m です。ERMapper に用意されているフィルターと

自作フィルターを組み合わせ、京都御苑の樹木の樹冠を抽出することを試みなさい。

試した結果、もっともよかった組み合わせをレポートにまとめてください。

メモ：

フィルターを重ねて、連続的に処理するには、

フィルターウィンドウのedit からinsert new filter を実行し、新しいフィルターを追加します。

3. リモートセンシングに関する賛成および反対の意見を調べ、列挙しなさい。

4. ランドスケーププランニングに関するさまざまな課題の中から1つを取り上げ、それを解決するためのツールとして

リモートセンシングが必要かどうかについて、前問の賛成および反対の立場から論述しなさい。

■課題の提出の仕方

winshot で画面上の画像を保存し、レポートにして、プリントアウトしたものを提出してください。

提出先は、5階環境デザイン事務室の今西のメールボックス（ポットの並びにあります）です。

A4 またはA3 用紙で1枚程度にまとめてください。名前を入れるのを忘れないように。