電子顕微鏡像のためのFFT [II]

$1 序論

本アプリケーションソフトウェアーは、下記の特徴を持っています:

1. 電子顕微鏡像(8ビットグレースケールのビットマップ)を処理するFFTフィルターの一つを選択可 (*).
   (*): ハイパス･フィルター, ローパス･フィルター, バンドパス･フィルター, バンドカット･フィルター.
2. 又は、好みの 「カスタム形状のFFTフィルター」 を作って適用することも可 (下記の $4 を参照).

****

注: 処理可能な像のサイズは N x M ピクセル (96 ≤ N, M ≤ 4096).

****

$2 ファンクション

 ビットマップ (8ビットグレースケール) のファイルを開きます.

 FFTパターンの表示の為のモードを選択します.

 FFTを実行します.

 逆FFTの為のモードを選択します.

 逆FFTを実行します.

 FFTパターン等をクリップボードへコピーします.

 FFTパターン等を名前を付けてファイルに保存します.

 FFTパターンのラインプロファイルデータを名前を付けてファイルに保存します.

 逆FFT後の像の明るさ･コントラストを調整します.

 バージョン情報を示します.

 (このファイル).

 本アプリケーションソフトウェアーを終了します.

$3 使い方

1.  で、電子顕微鏡像(8ビットグレースケールのビットマップ)を開きます.
2.  で、 FFTパターンの表示の為のモードを選択します.
3.  で、FFTを実行します.
4.  で、逆FFTの為のモードを選択します.
5.  で、逆FFTを実行します.
6. 必要に応じ、  で、逆FFT後の像の明るさ･コントラストを調整します.
7.  で、ファイルに保存します.
8. 必要に応じ、  で、クリップボードへコピーします.

****

注: 像やパターンは自動保存されません. 必要に応じて、“save button”で 保存を行ってください.

****

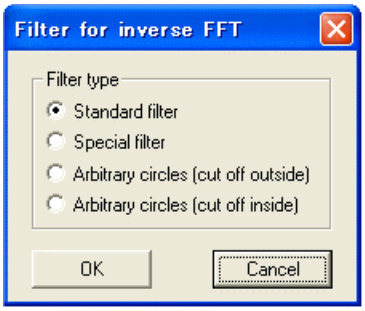
$4 カスタム形状のFFTフィルターの作成法

手順 「$3 使い方 － 4」 で得られる (下記の) ダイヤログで、逆FFTの為のフィルターのモードが選択できます.
“Standard filter” を選ぶと、ハイパス・フィルター, ローパス・フィルター, バンドパス・フィルター, バンドカット・フィルターの中から、1つのフィルターを選択できます. .
一方、“Special filter” を選ぶと、カスタム形状のFFTフィルターが作成できます. 作成法は、次の通りです.

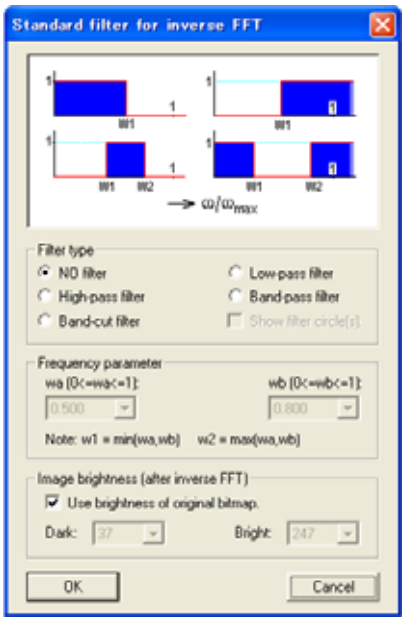
横軸のボタンで 「赤丸 (変更したいポイント)」 を選び、縦軸のボタンで、当該の 「赤丸 (変更したいポイント)」 の位置を上下させる. 得られる曲線は、スプライン近似で、出来るだけ滑らかとなるようにしてある.
なお、グラフの左下角は座標原点 (0, 0) です.

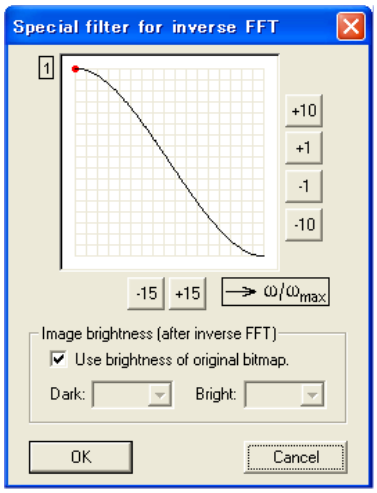
更に、“Arbitrary circles (cut off outside)”、“Arbitrary circles (cut of inside)” の2者の いずれかを選ぶと、マウスカーソルが 「円環状の形」 となるので、FFT画面上で、マウスカーソルの左ボタンを押し ～ マウスをドラッグすると、円が描かれます (必要に応じて、複数個の円を描きます). この (複数個の) 円の内側領域 or 外側領域が、逆FFTを実行する時に カットされる領域 となります (下図を参照して下さい).

ボタンを押すと



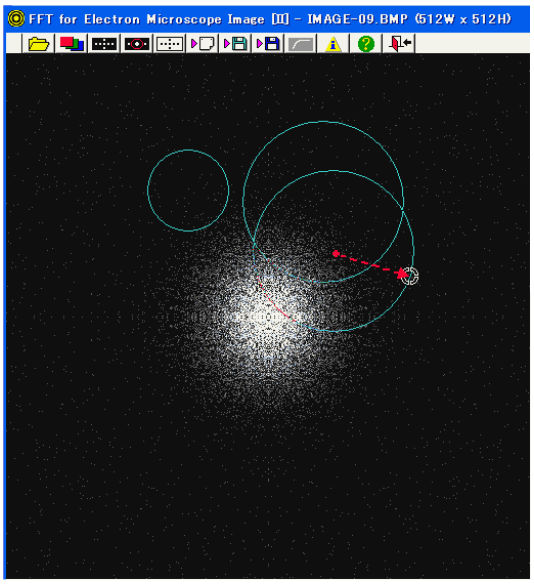
OK ボタンを押すと





カスタム形状のFFTフィルターの作成後に、「OK」 ボタンを押すと、曲線の数値は INI ファイルに保存されるので、次回に (自動的に) 再利用できます.

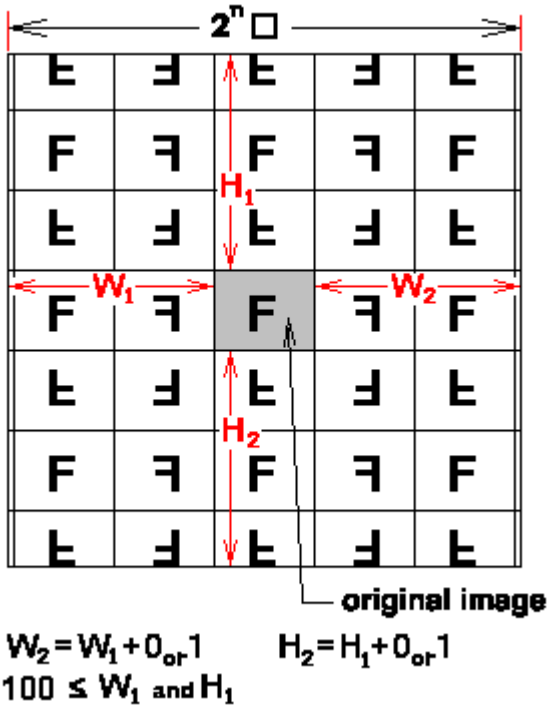
マウスカーソルが 「円環状の形」 となるので、FFT画面上で、

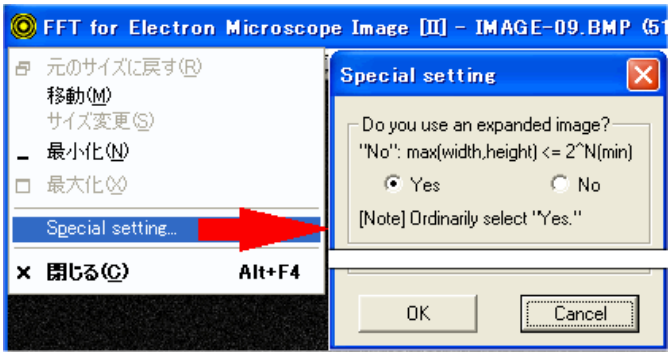


マウスカーソルの左ボタンを押し ～ マウスをドラッグすると、円が描かれます.
この (単数個 or 複数個の) 円の内側領域 or 外側領域が、逆FFTを実行する時にカットされる領域 となります.

$5 参考

FFTアプリケーションソフトウェアー開発のポイントは下記と考えます：





1. 鏡映像、反転像、鏡映反転像、を作る.
2. これらの像を、原画像に組み合わせる(上図を参照ください).
3. 上図のような拡大像を処理できるFFTアプリケーションソフトウェアーを開発する.
4. FFT実行後のFFTパターンの表示は、原画像に対応する領域のみ行う.
5. FFTパターンへの空間周波数フィルター適用は、拡大像全体で行う.
6. このような空間周波数フィルター適用で、逆FFTを実行する.
7. 逆FFT実行後の像の表示は、原画像に対応する領域のみ行う.

注：“100 ≤ W1 and H1”は、最小限の条件です. この値は大きいほど良い (例えば 200).

****

備考：

(1) ご存知のように、DFTと逆DFTでは、N x M (N ≠ 2^n, M ≠ 2^m, n, m =正整数) ピクセルの像の処理を行えます.

(2) しかし、DFTと逆DFTをその通りに行うと、非常に長時間を要します.

(3) もし、像のサイズが 2^n 正方形 (n = 正整数) であるなら、Cooley と Tukey による FFT･逆FFTのアルゴリズムが使用できます.

(4) これを使えば、 FFT･逆FFTの実行時間を大いに節約できます.

(5) これを使って、“FFT → ハイパスフィルター → 逆FFT”を行ってみると、逆FFT後の像の周辺部分が明るくなると云うアーティファクトが出現します.
注：“FFT → ローパスフィルター → 逆FFT”では、このようなアーティファクトを (目で)見つけるのは難しいです.

(6) これは、像が周期的境界条件を満たさない為に生じるものです. これを避ける手法として、像にハミング窓関数を荷重することも考えられますが、もし像が周辺部のみ微細構造を有している(中央部には微細構造は一切無い)場合には、どうしますか？
それで、上記のような拡大像手法が良いと思いますが、如何でしょうか.

(7) 上記の最小限の条件“100 ≤ W1 and H1”は、“FFT → ハイパスフィルター → 逆FFT”を行った場合、逆FFT後の像の周辺部分が明るくなると云うアーティファクトを、(見た目で)最小限の押さえる - 完璧ではないですが - 条件です.

****