



ピクトの独り言

フーリエ変換の話し_その5



株式会社 アイネット



周波数補正



- 少し長いコーヒータイムでしたね(笑)。
- いよいよラストスパート、周波数領域での画像補正です。
- 次の5つの「周波数補正」を取り上げましょう。
- ① 直流波（直流成分）カットフィルタ
- ② 低周波カットフィルタ
- ③ 高周波カットフィルタ
- ④ 中周波（中間波）カットフィルタ
- ⑤ 畳み込み計算（畳み込み演算）
- 前章の「凹凸のある画像」を計算対象としてみましょう。
- それと、 $\text{Sin} \times \text{Cos}$ と $\text{Cos} \times \text{Sin}$ は、計算を省略しますね。

周波数の区分 1



- 周波数を考えるにあたって、一つ注意してください。
- 高周波・低周波等は、「相対的な概念」です。
- 絶対的・正確な区分基準は存在しません。

- 8×1 画素の場合、周波数は、例えば次の通りです。
- ① 4 行目 最高周波（高周波の一部）
- ② 3 行目と 5 行目 高周波 （高周波の一部）
- ③ 2 行目と 6 行目 中周波
- ④ 1 行目と 7 行目 低周波 （低周波の一部）
- ⑤ 0 行目 直流波 （低周波の一部）

- 中周波（中間波）は、高・低のどちらに入るのでしょうか？

画像17



凹凸のある画像_周波数の区分

色No	色値	フーリエ計算		x の計算 低周波		x の計算 高周波		x の計算 中周波
		x	y	直流波	除 直流	含 最高	除 最高	
4	140.0	40.0	△0.0	40.0	40.0	40.0	40.0	40.0
5	150.0	7.1	△85.4	7.1	7.1	7.1	7.1	7.1
6	100.0	30.0	50.0	30.0	30.0	30.0	30.0	30.0
7	120.0	△7.1	14.6	△7.1	△7.1	△7.1	△7.1	△7.1
0	140.0	1,020.0	0.0	1,020.0	1,020.0	1,020.0	1,020.0	1,020.0
1	120.0	△7.1	△14.6	△7.1	△7.1	△7.1	△7.1	△7.1
2	150.0	30.0	△50.0	30.0	30.0	30.0	30.0	30.0
3	100.0	7.1	85.4	7.1	7.1	7.1	7.1	7.1
合計	1,020.0	1,120.0	0.0	1,120.0	1,120.0	1,120.0	1,120.0	1,120.0

周波数の区分 2



- 2行目と6行目の中周波をどう考えるか。
- A 中周波を独立波と考える方法 = 中周波法。
- B 中周波を高周波に含める方法 = 高周波法。
- C 中周波を低周波に含める方法 = 低周波法。

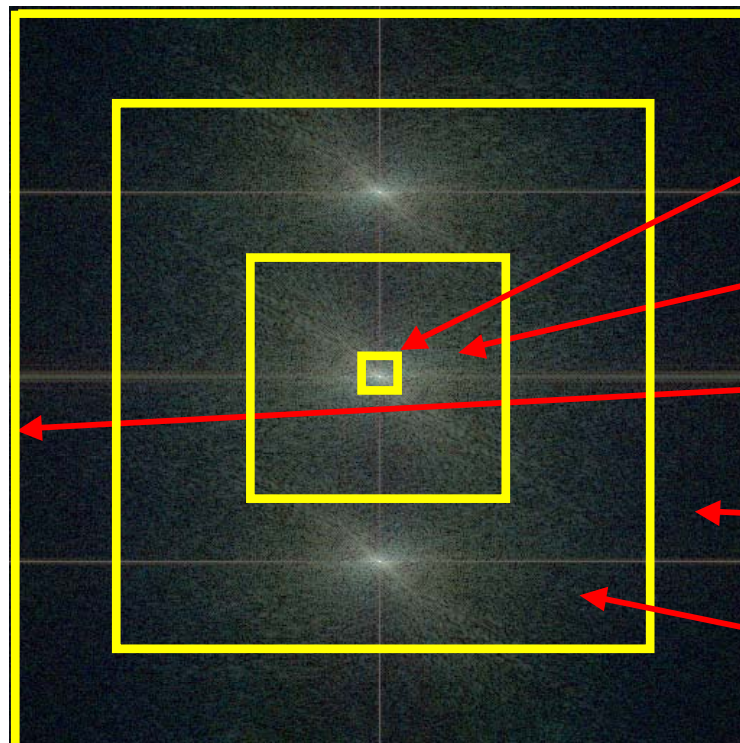
- BとCでは、カットフィルタとパスフィルタが一致します。
- ここでは、Aの「中周波法」の考えで処理を行います。
- また、直流波は、低周波から除いて処理します。
- ① 直流波カット（ハイパスフィルタの一部）
- ② 低周波カット（ハイパスフィルタの一部）
- ③ 高周波カット（ローパスフィルタの一部）
- ④ 中間波カット（ハイパスかローパス）

画像18



周波数の区分（低周波・高周波・中周波）

周波数の区分 ⇒ 「★★相対的★★」



直流波 (低周波) ⇒ 中央の十字部分

低周波 ⇒ 直流波以外の内側近辺

最高周波 (高周波) ⇒ 最外縁

高周波 ⇒ 外側近辺

中周波 (中間波) ⇒ 低周波・高周波以外の波

直流波カット



- 直流波のカット、すなわち色値合計のカットです。
- 直流波カットは、本来は、低周波カットの一領域です。
- 0行目の直流波をカットしてみましょう。
- カットするということは、データを使用しないことです。
- フーリエ計算した x と y の直流波色値をゼロにしましょう。
- 直流波以外の周波は、そのままデータを使用します。
- 色値合計はゼロとなるため、負の色値が発生します。
- 負の色値はないのでゼロになり、色差はそのまま残ります。
- ですから、輪郭だけが抽出されるのです(★重要★)。

画像19



直流波カットフィルタ

色No	x	x	y	y	x	y	x-y	
色値	cos	sin	cos	sin	cos	sin	Σ	÷ 数
4	100.0	0.0	0.0	0.0	100.0	0.0	100.0	12.5
5	$\Delta 20.0$	$\Delta 0.0$	0.0	$\Delta 200.0$	$\Delta 20.0$	$\Delta 200.0$	180.0	22.5
6	$\Delta 20.0$	0.0	$\Delta 0.0$	200.0	$\Delta 20.0$	200.0	$\Delta 220.0$	$\Delta 27.5$
7	$\Delta 60.0$	0.0	0.0	0.0	$\Delta 60.0$	0.0	$\Delta 60.0$	$\Delta 7.5$
0	100.0	0.0	0.0	0.0	100.0	0.0	100.0	12.5
1	$\Delta 60.0$	$\Delta 0.0$	0.0	0.0	$\Delta 60.0$	0.0	$\Delta 60.0$	$\Delta 7.5$
2	$\Delta 20.0$	$\Delta 0.0$	$\Delta 0.0$	$\Delta 200.0$	$\Delta 20.0$	$\Delta 200.0$	180.0	22.5
3	$\Delta 20.0$	0.0	0.0	200.0	$\Delta 20.0$	200.0	$\Delta 220.0$	$\Delta 27.5$
合計	0.0	$\Delta 0.0$	0.0	0.0	0.0	0.0	0.0	0.0

画像20



直流波カットフィルタ (フーリエ計算した $x * \cos$)

色No	4	5	6	7	0	1	2	3	合計
色値	40	7	30	△ 7	0	△ 7	30	7	100
4	40.0	△7.1	30.0	7.1	0.0	7.1	30.0	△7.1	100.0
5	△40.0	5.0	△0.0	5.0	0.0	5.0	0.0	5.0	△20.0
6	40.0	△0.0	△30.0	△0.0	0.0	0.0	△30.0	0.0	△20.0
7	△40.0	△5.0	0.0	△5.0	0.0	△5.0	△0.0	△5.0	△60.0
0	40.0	7.1	30.0	△7.1	0.0	△7.1	30.0	7.1	100.0
1	△40.0	△5.0	△0.0	△5.0	0.0	△5.0	0.0	△5.0	△60.0
2	40.0	0.0	△30.0	0.0	0.0	△0.0	△30.0	△0.0	△20.0
3	△40.0	5.0	0.0	5.0	0.0	5.0	△0.0	5.0	△20.0
合計	0.0	0.0	△0.0	0.0	0.0	0.0	△0.0	0.0	0.0

画像21



直流波カットフィルタ (フーリエ計算した $y * \sin$)

色No	4	5	6	7	0	1	2	3	合計
色値	$\Delta 0$	$\Delta 85$	50	15	0	$\Delta 15$	$\Delta 50$	85	0
4	0.0	$\Delta 0.0$	0.0	0.0	0.0	$\Delta 0.0$	0.0	0.0	0.0
5	$\Delta 0.0$	$\Delta 60.4$	$\Delta 50.0$	10.4	0.0	10.4	$\Delta 50.0$	$\Delta 60.4$	$\Delta 200.0$
6	0.0	85.4	0.0	14.6	0.0	14.6	$\Delta 0.0$	85.4	200.0
7	$\Delta 0.0$	$\Delta 60.4$	50.0	10.4	0.0	10.4	50.0	$\Delta 60.4$	0.0
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	$\Delta 0.0$	60.4	$\Delta 50.0$	$\Delta 10.4$	0.0	$\Delta 10.4$	$\Delta 50.0$	60.4	0.0
2	0.0	$\Delta 85.4$	0.0	$\Delta 14.6$	0.0	$\Delta 14.6$	$\Delta 0.0$	$\Delta 85.4$	$\Delta 200.0$
3	$\Delta 0.0$	60.4	50.0	$\Delta 10.4$	0.0	$\Delta 10.4$	50.0	60.4	200.0
合計	$\Delta 0.0$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

低周波カット



- ローカットフィルタは、低周波をカットするフィルタです。
- 直流波を除く低周波だけ、 1 行目と 7 行目をカットします。
- 低周波以外の周波は、すべて残します。
- 色値合計を残して、低周波領域の色差を削除しました。
- 色値を残したため、輪郭が抽出されることはありません。
- 低周波領域は、大きな波（滑らかな部分）でした。
- そこをカットしたため、画像がより一層滑らかになります。
- ボケた画像を作成したいときに利用します(★重要★) 。

画像22



低周波カットフィルタ

色No	x	x	y	y	x	y	x-y	
色値	cos	sin	cos	sin	cos	sin	Σ	÷ 数
4	1,105.9	$\Delta 0.0$	0.0	0.0	1,105.9	0.0	1,105.9	138.2
5	990.0	$\Delta 0.0$	0.0	$\Delta 220.7$	990.0	$\Delta 220.7$	1,210.7	151.3
6	1,000.0	0.0	$\Delta 0.0$	170.7	1,000.0	170.7	829.3	103.7
7	970.0	0.0	0.0	$\Delta 20.7$	970.0	$\Delta 20.7$	990.7	123.8
0	1,134.1	0.0	0.0	0.0	1,134.1	0.0	1,134.1	141.8
1	970.0	$\Delta 0.0$	0.0	20.7	970.0	20.7	949.3	118.7
2	1,000.0	$\Delta 0.0$	$\Delta 0.0$	$\Delta 170.7$	1,000.0	$\Delta 170.7$	1,170.7	146.3
3	990.0	0.0	0.0	220.7	990.0	220.7	769.3	96.2
合計	8,160.0	$\Delta 0.0$	$\Delta 0.0$	0.0	8,160.0	0.0	8,160.0	1,020.0

画像23



低周波カットフィルタ (フーリエ計算した $x * \text{Cos}$)

色No	4	5	6	7	0	1	2	3	合計
色値	40	7	30	0	1,020	0	30	7	1,134
4	40.0	$\Delta 7.1$	30.0	0.0	1,020.0	0.0	30.0	$\Delta 7.1$	1,105.9
5	$\Delta 40.0$	5.0	$\Delta 0.0$	0.0	1,020.0	0.0	0.0	5.0	990.0
6	40.0	$\Delta 0.0$	$\Delta 30.0$	0.0	1,020.0	0.0	$\Delta 30.0$	0.0	1,000.0
7	$\Delta 40.0$	$\Delta 5.0$	0.0	0.0	1,020.0	0.0	$\Delta 0.0$	$\Delta 5.0$	970.0
0	40.0	7.1	30.0	0.0	1,020.0	0.0	30.0	7.1	1,134.1
1	$\Delta 40.0$	$\Delta 5.0$	$\Delta 0.0$	0.0	1,020.0	0.0	0.0	$\Delta 5.0$	970.0
2	40.0	0.0	$\Delta 30.0$	0.0	1,020.0	0.0	$\Delta 30.0$	$\Delta 0.0$	1,000.0
3	$\Delta 40.0$	5.0	0.0	0.0	1,020.0	0.0	$\Delta 0.0$	5.0	990.0
合計	0.0	0.0	$\Delta 0.0$	0.0	8,160.0	0.0	$\Delta 0.0$	0.0	8,160.0

高周波カット



- ハイカットフィルタは、高周波をカットするフィルタです。
 - 高周波、すなわち $4 \cdot 3 \cdot 5$ 行目をカットします。
 - フーリエ計算した x と y の色値をゼロにしましょう。
 - 高周波以外の周波は、そのデータを使用します。
-
- 色値合計を残して、高周波領域の色差を削除しました。
 - 高周波領域は、小刻みな波（小さなノイズ）です。
-
- 「小さなノイズをカットしても、全体的に影響は少ない。」
 - そう考えることも可能です。
 - ですから、ハイカットの特徴を利用すると・・・。
 - 画像圧縮が可能になるのです(★重要★)。

画像25



高周波カットフィルタ（4・3・5行目をカット）

色No	x	x	y	y	x	y	x-y	
色値	cos	sin	cos	sin	cos	sin	Σ	÷ 数
4	1,094.1	0.0	$\Delta 0.0$	$\Delta 0.0$	1,094.1	$\Delta 0.0$	1,094.1	136.8
5	1,030.0	$\Delta 0.0$	0.0	$\Delta 79.3$	1,030.0	$\Delta 79.3$	1,109.3	138.7
6	960.0	0.0	$\Delta 0.0$	29.3	960.0	29.3	930.7	116.3
7	1,010.0	0.0	0.0	120.7	1,010.0	120.7	889.3	111.2
0	1,065.9	0.0	$\Delta 0.0$	0.0	1,065.9	0.0	1,065.9	133.2
1	1,010.0	$\Delta 0.0$	0.0	$\Delta 120.7$	1,010.0	$\Delta 120.7$	1,130.7	141.3
2	960.0	$\Delta 0.0$	$\Delta 0.0$	$\Delta 29.3$	960.0	$\Delta 29.3$	989.3	123.7
3	1,030.0	0.0	0.0	79.3	1,030.0	79.3	950.7	118.8
合計	8,160.0	$\Delta 0.0$	0.0	0.0	8,160.0	0.0	8,160.0	1,020.0

画像26



高周波カットフィルタ（3・5行目をカット）

色No	x	x	y	y	x	y	x-y	
色値	cos	sin	cos	sin	cos	sin	Σ	÷数
4	1,134.1	0.0	$\Delta 0.0$	$\Delta 0.0$	1,134.1	$\Delta 0.0$	1,134.1	141.8
5	990.0	$\Delta 0.0$	0.0	$\Delta 79.3$	990.0	$\Delta 79.3$	1,069.3	133.7
6	1,000.0	0.0	$\Delta 0.0$	29.3	1,000.0	29.3	970.7	121.3
7	970.0	0.0	0.0	120.7	970.0	120.7	849.3	106.2
0	1,105.9	0.0	$\Delta 0.0$	0.0	1,105.9	0.0	1,105.9	138.2
1	970.0	$\Delta 0.0$	0.0	$\Delta 120.7$	970.0	$\Delta 120.7$	1,090.7	136.3
2	1,000.0	$\Delta 0.0$	$\Delta 0.0$	$\Delta 29.3$	1,000.0	$\Delta 29.3$	1,029.3	128.7
3	990.0	0.0	0.0	79.3	990.0	79.3	910.7	113.8
合計	8,160.0	$\Delta 0.0$	0.0	0.0	8,160.0	0.0	8,160.0	1,020.0

中周波カット



- 中周波の概念は、高周波と低周波以上に曖昧です(笑)。
 - 「2種類の矩形にしましたが、どちらでも結構です。」
 - という文章が、第1章にありましたね。
 - あれは中周波の曖昧さを物語っています。
-
- 2・6行目のxとyの色値をカット・ゼロにしましょう。
 - 中周波以外の周波は、すべて残します。
-
- 同じ色値合計で、やはり色差が小さくなっています。
 - 大きな輪郭補正や小さなノイズ補正でもありません。
 - ミドルカットフィルタには重要な役割があります。
 - それは、「パターンノイズ補正」です(★重要★)。

画像27



中周波カットフィルタ

色No	x	x	y	y	x	y	x-y	
色値	cos	sin	cos	sin	cos	sin	Σ	÷ 数
4	1,060.0	0.0	0.0	0.0	1,060.0	0.0	1,060.0	132.5
5	1,000.0	0.0	0.0	$\Delta 100.0$	1,000.0	$\Delta 100.0$	1,100.0	137.5
6	1,060.0	0.0	$\Delta 0.0$	200.0	1,060.0	200.0	860.0	107.5
7	960.0	0.0	0.0	$\Delta 100.0$	960.0	$\Delta 100.0$	1,060.0	132.5
0	1,060.0	0.0	0.0	0.0	1,060.0	0.0	1,060.0	132.5
1	960.0	$\Delta 0.0$	0.0	100.0	960.0	100.0	860.0	107.5
2	1,060.0	$\Delta 0.0$	$\Delta 0.0$	$\Delta 200.0$	1,060.0	$\Delta 200.0$	1,260.0	157.5
3	1,000.0	$\Delta 0.0$	0.0	100.0	1,000.0	100.0	900.0	112.5
合計	8,160.0	$\Delta 0.0$	0.0	0.0	8,160.0	0.0	8,160.0	1,020.0

カットフィルタ 1



- カットフィルタには、二つの特徴があります。
 - ① パワースペクトルの直接補正ではない。
 - ② 補正後の画像は、元の画像より色差が小さくなる。
-
- 周波数領域の補正は、「スペクトルを直接補正」している？
 - そんな印象を受けますが、これはパフォーマンスです(笑)。
 - 確かに、スペクトルを直接補正する方法もあります。
 - でも、ここでは塗りつぶした部分を特定しているだけです。
-
- フーリエ計算した x と y は、色差を表します。
 - カットフィルタは、特定の色差をゼロにするものです。
 - ゼロにするから、本来の色差より小さくなるのです。

カットフィルタ 2



- カットフィルタの特徴は、次のとおりでした。
- ① 直流カット … 輪郭抽出
- ② ローカット … 画像暈し（大ノイズ暈し）
- ③ ハイカット … 小ノイズ暈し、画像圧縮
- ④ ミドルカット … パターンノイズ削減、画像圧縮

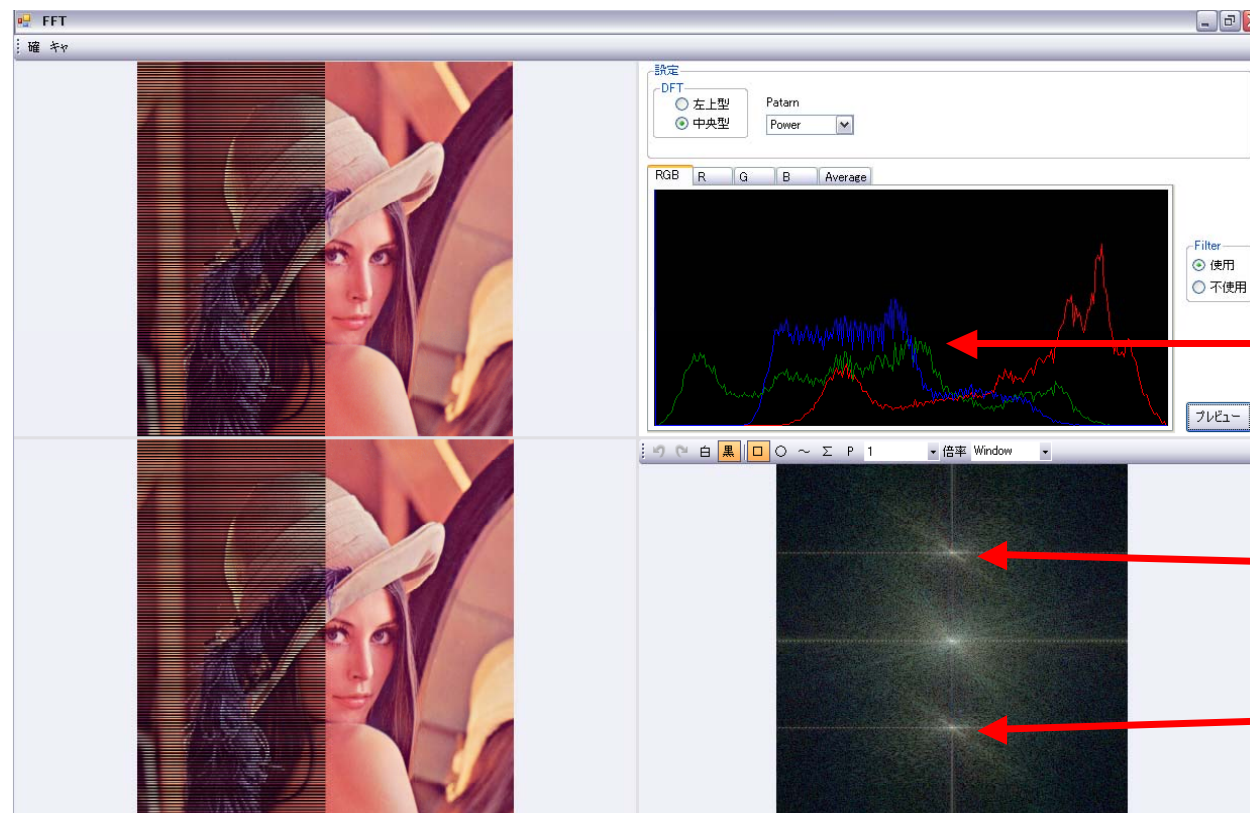
- これを実際の画像で確認してみましょう。
- 資料は、私「ピクト」のFFT画面です。

- まだ、開発中ですので、絵面が悪いですね(苦笑)。
- 使い勝手も、納得の行くものではありません。
- 皆さんの前に登場するのは、まだまだ先のようですね(笑)。

画像28

net

補正前の画像



小さなノイズが多い。

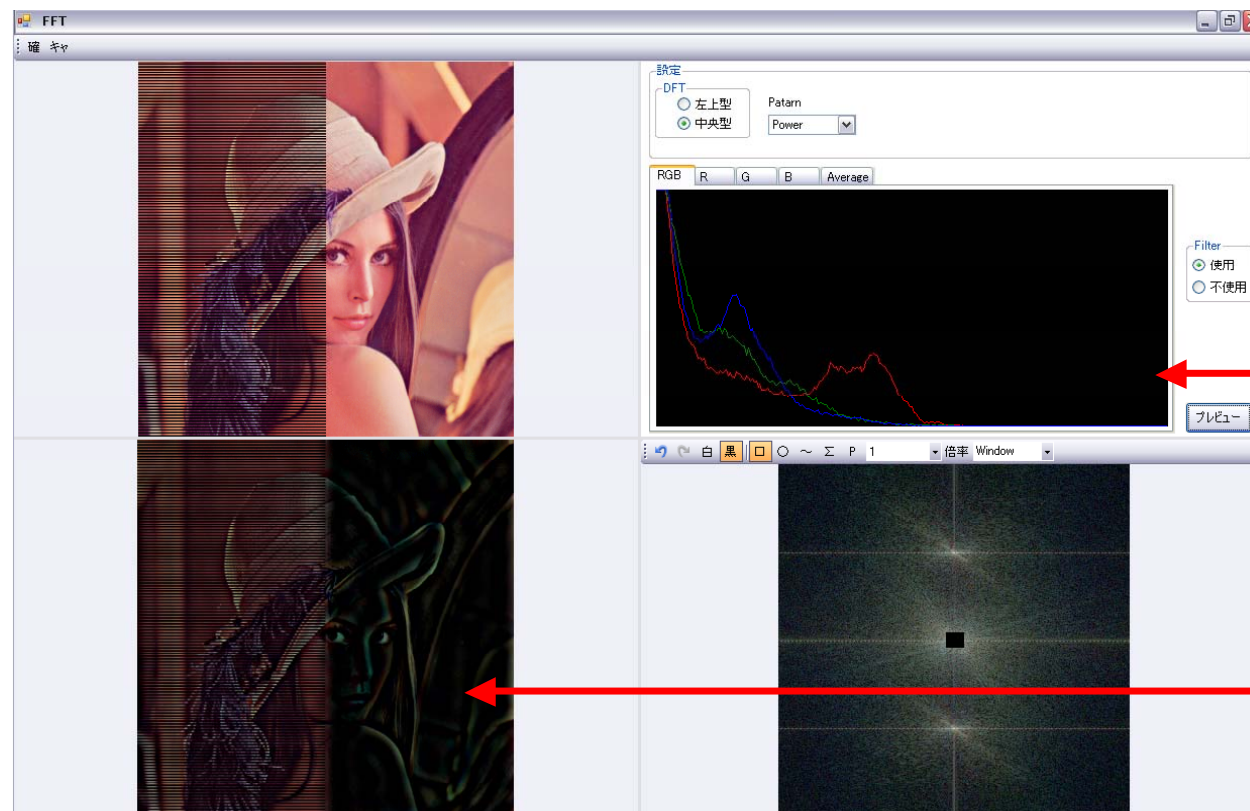
中間波のノイズ。

中間波のノイズ。

画像29

net

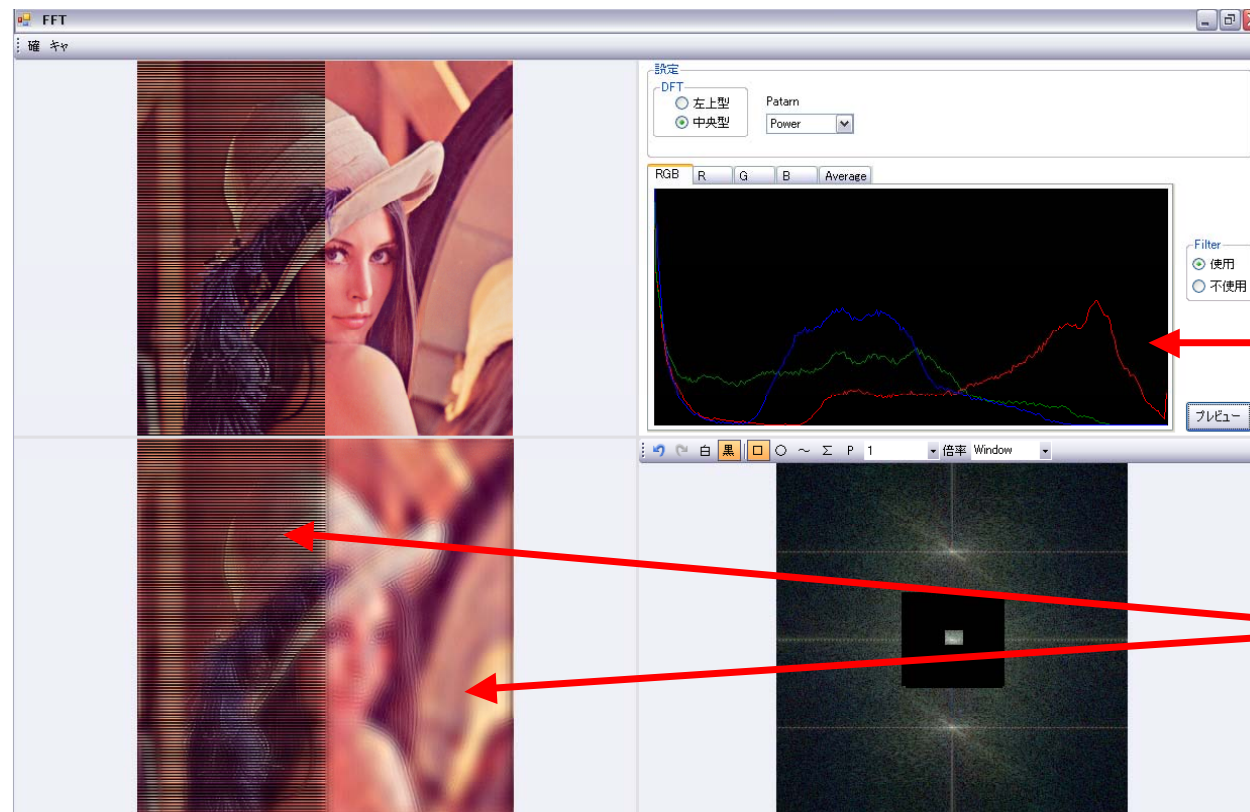
直流波カットフィルタ



画像30

net

直流波以外の低周波（ロー）カットフィルタ



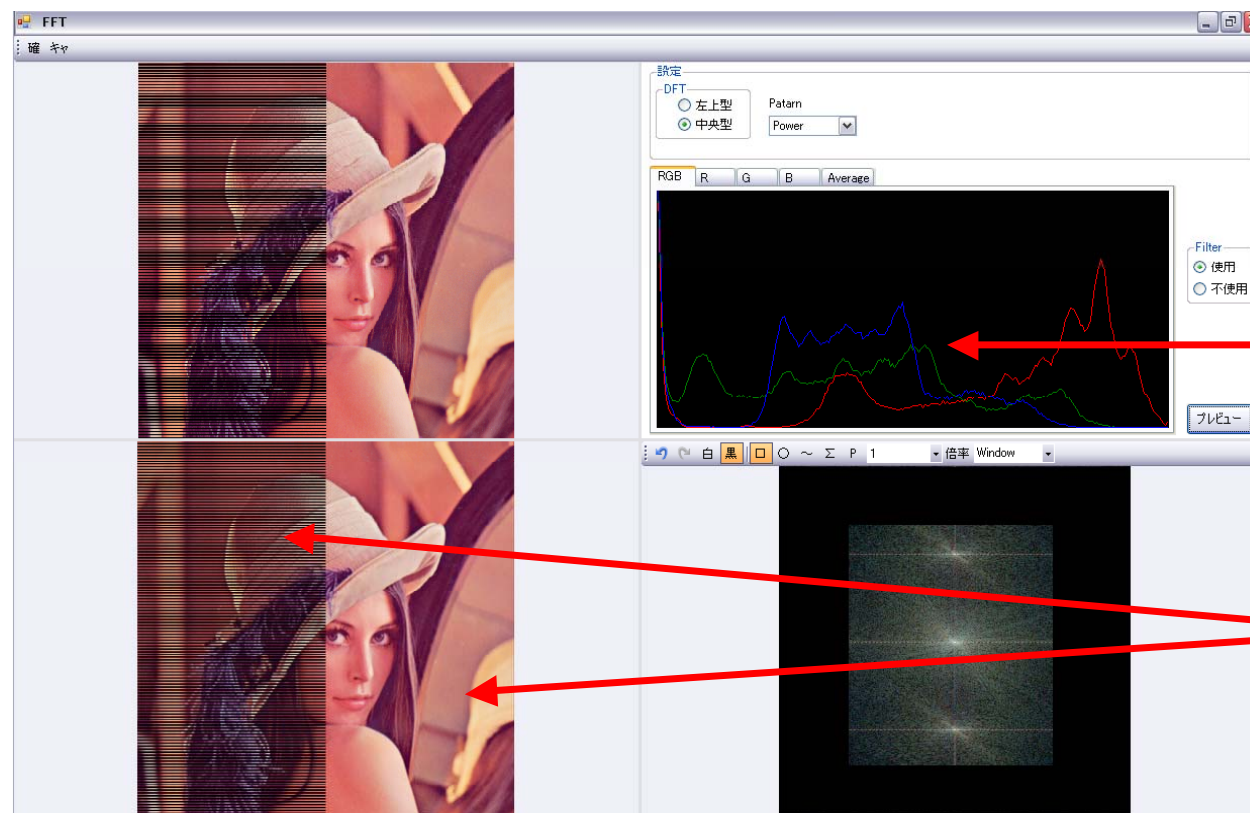
全体的に、滑らかになっている。

全体的に、ボケてしまっている。

画像31



最高周波を含む高周波（ハイ）カットフィルタ



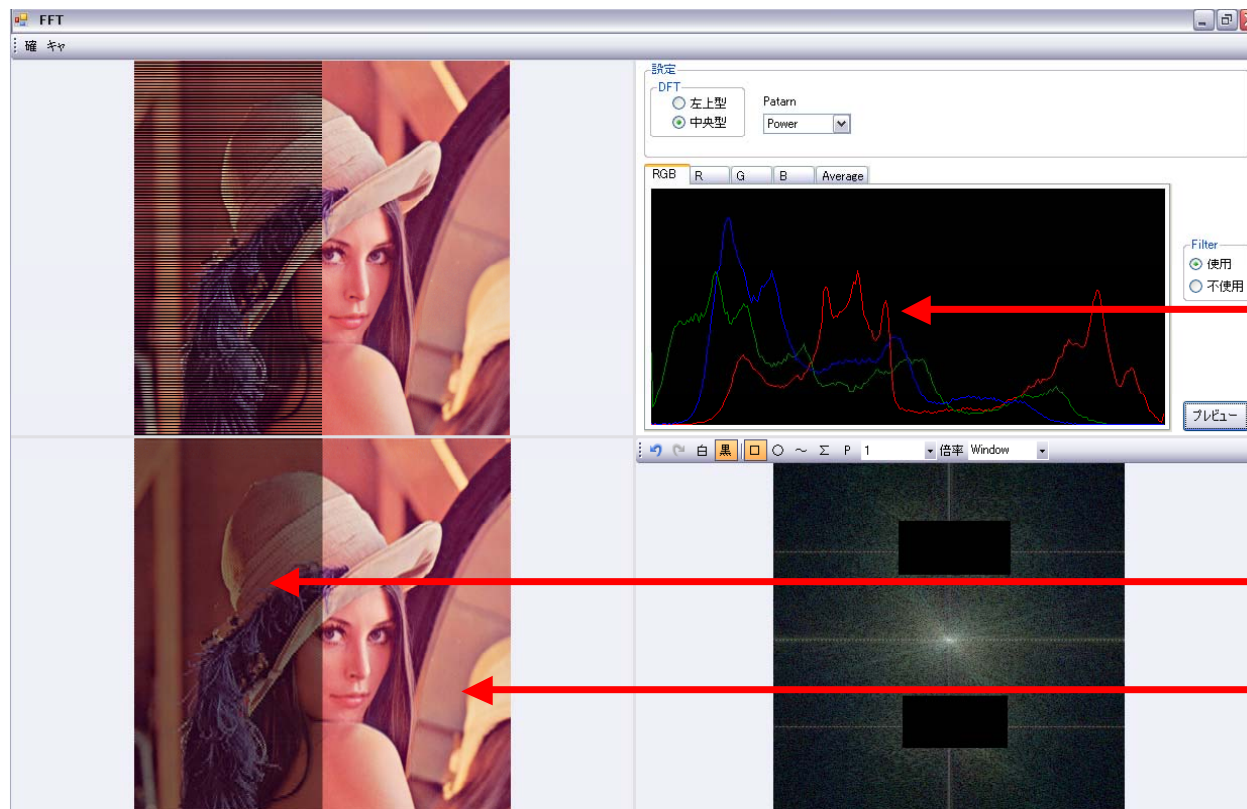
ノイズがまだ残っている。

全体的に、元の画像と大差ない。

画像32



中間波（ミドル）カットフィルタ



ノイズが消滅している

ノイズが消滅している

全体的に、元の画像と大差ない。

畳み込み 1



- では、最後に「畳み込み計算」です。
- 畳み込み計算がいかに高速になるかをご堪能ください。

- まず、通常の画像での畳み込み計算です。
- 以下は、フィルタを「F」として表示しましょうね。
- Fは、合計=1.0 になるように正規化してください。
- ① FNo 3, 2, 1, 0, 7, 6, 5 (4は無し)
- ② F値 1.0, 2.0, 3.0, 5.0, 3.0, 2.0, 1.0 ⇒ 計 17.0
- ③ 正規化 1/17, 2/17, 3/17, 5/17, 3/17, 2/17, 1/17

- 画像の色値にF値を掛け合わせます。
- Fは、正規化合計=1.0ですので、色値合計は近似します。

畳み込み 2

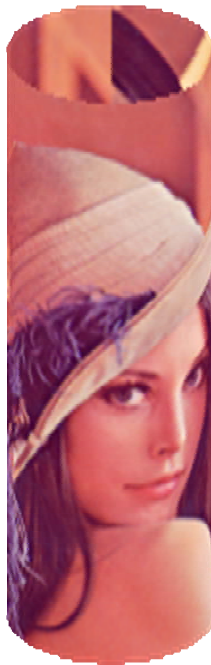


- 問題は、端の処理です。通常は、次の4つがあります。
 - ① 端全体を無視する方法
 - ② 一方向のみを対象とする方法
 - ③ 二方向を対象とする方法
 - ④ 反対側の端を対象とする方法
-
- どの方法によるかで、画像の品質と計算速度が異なります。
 - 計算速度は最速ですが、品質が最悪であるのも、①です。
 - ②と③は、それほどの差異はありません。
-
- フーリエ変換は、④の「友達の輪」です。
 - 頭とお尻が繋がっていたことを思い出してね(位相)。

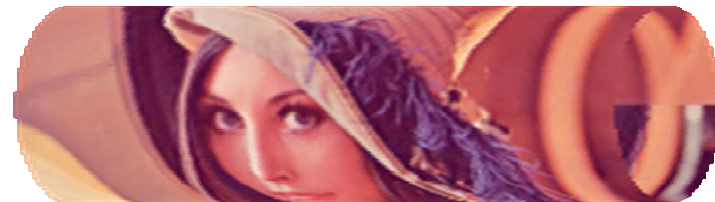
画像33



フーリエ変換の端の考え方
(レナさん_左右の端)



(レナさん_上下の端)



畳み込み 3



- ここでは、③の方法を採用してみましょう。
- まず、中央の色 0 の計算を考えます。
- 自分の色を基準にして、左右 3 つまでを加算します。
- 色と F を掛け合わせて行き、最後に合計するのです。
- $\{\text{色 } 0 \times F 0\} + \{(\text{色 } 1 + \text{色 } 7) \times F 1\} + \dots$ 。
- F 4 はありませんので、パス（ゼロ）になります。
- 合計すると、色 0 が補正されているのが分かります。
- 色 1 の計算では、F 3 に適用する色 No が存在しません。
- そのため、折返し運転となり、色 2 を適用します。
- その結果、色 2 $\times$ F 3 として計算します。

画像34



通常の畳み込み計算

色No		7	6	5	4	5	6	7	0	1	2	3	2	1	0	合計
色値		120	100	150	140	150	100	120	140	120	150	100	150	120	140	1,800
4	F	0.06	0.12	0.18	0.29	0.18	0.12	0.06								
	値	7.1	11.8	26.5	41.2	26.5	11.8	7.1								131.8
5	F		0.06	0.12	0.18	0.29	0.18	0.12	0.06							
	値		5.9	17.6	24.7	44.1	17.6	14.1	8.2							132.4
6	F			0.06	0.12	0.18	0.29	0.18	0.12	0.06						
	値			8.8	16.5	26.5	29.4	21.2	16.5	7.1						125.9
7	F				0.06	0.12	0.18	0.29	0.18	0.12	0.06					
	値				8.2	17.6	17.6	35.3	24.7	14.1	8.8					126.5
0	F					0.06	0.12	0.18	0.29	0.18	0.12	0.06				
	値					8.8	11.8	21.2	41.2	21.2	17.6	5.9				127.6
1	F						0.06	0.12	0.18	0.29	0.18	0.12	0.06			
	値						5.9	14.1	24.7	35.3	26.5	11.8	8.8			127.1
2	F							0.06	0.12	0.18	0.29	0.18	0.12	0.06		
	値							7.1	16.5	21.2	44.1	17.6	17.6	7.1		131.2
3	F								0.06	0.12	0.18	0.29	0.18	0.12	0.06	
	値								8.2	14.1	26.5	29.4	26.5	14.1	8.2	127.1
合計	F															
	値	7.1	17.6	52.9	90.6	123.5	94.1	120.0	140.0	112.9	123.5	64.7	52.9	21.2	8.2	1,029.4

周波数畳み込み 1



- 畳み込み計算を周波数領域で行いたいと思います。
- ① フィルタが存在しないF 4は、0.0と仮定する。
- ② 色0に適用するF値を1.0とする（ここでは「5/5」）。
- ③ その他も、「●/5」を適用して計算する。
- ④ FNo 3, 2, 1, 0, 7, 6, 5, 4
- ⑤ F 1/5, 2/5, 3/5, 5/5, 3/5, 2/5, 1/5, 0/5

- このF値をフーリエ計算したxとyに掛け算して・・・。
- ただ逆変換するだけ、それですべて完了です。
- わっ、凄い凄い。速い速い。パチパチパチ(笑)。

- でも、正式な計算は加法定理を使うんですけどね(謝罪)。

画像35



周波数領域での畳み込み計算_ 1

色No	フーリエ計算		Filter	cos	sin	Filter の注意点
色値	x	y		x	y	
4	40.0	△0.0	0.0	0.0	0.0	← Filter=0/5 にする。
5	7.1	△85.4	0.2	1.4	△17.1	
6	30.0	50.0	0.4	12.0	20.0	
7	△7.1	14.6	0.6	△4.2	8.8	
0	1,020.0	0.0	1.0	1,020.0	0.0	← Filter=5/5 にする。
1	△7.1	△14.6	0.6	△4.2	△8.8	
2	30.0	△50.0	0.4	12.0	△20.0	
3	7.1	85.4	0.2	1.4	17.1	
合計	1,120.0	0.0	3.4	3,808.0	0.0	

画像36



周波数領域での畳み込み計算_2

色No	x	x	y	y	x	y	x-y	
色値	cos	sin	cos	sin	cos	sin	Σ	÷ 数
4	1,049.7	0.0	0.0	0.0	1,049.7	0.0	1,049.7	131.2
5	1,028.0	0.0	0.0	$\Delta 51.7$	1,028.0	$\Delta 51.7$	1,079.7	135.0
6	996.0	0.0	0.0	51.7	996.0	51.7	944.3	118.0
7	1,012.0	0.0	0.0	28.3	1,012.0	28.3	983.7	123.0
0	1,038.3	0.0	0.0	0.0	1,038.3	0.0	1,038.3	129.8
1	1,012.0	$\Delta 0.0$	$\Delta 0.0$	$\Delta 28.3$	1,012.0	$\Delta 28.3$	1,040.3	130.0
2	996.0	$\Delta 0.0$	$\Delta 0.0$	$\Delta 51.7$	996.0	$\Delta 51.7$	1,047.7	131.0
3	1,028.0	$\Delta 0.0$	0.0	51.7	1,028.0	51.7	976.3	122.0
合計	8,160.0	$\Delta 0.0$	0.0	0.0	8,160.0	0.0	8,160.0	1,020.0

周波数畳み込み 2



- 周波数領域の畳み込み演算って、計算量が少ないでしょう。
 - でも、フーリエ変換自体の計算量が多いような・・・。
 - 心配後無用、高速化する方法がちゃんと用意されています。
 - 「高速フーリエ変換（FFT）」と言われる方法ですね。
-
- 周波数領域における畳み込み計算のポイントは、
 - ① 色値合計を元の値にするため、 $F0=1.0$ を適用する。
 - ② 色4に0.0を適用するため、隣との色差計=0.0になる。
 - ③ 実際の画像補正とは、計算結果が若干異なる。
 - ④ ただし、両方とも色値が滑らかになる。
-
- 以上が、低周波を中心にした畳み込み計算です。

数式によるフーリエ変換¹

- フーリエ変換では、次のような数式が散見されます。
- 連続フーリエ変換 $F(\omega) = \int f(t) \exp(j\omega t) dt$
- 連続フーリエ逆変換 $f(t) = \int F(\omega) \exp(j\omega t) d\omega$
- 離散フーリエ変換 $F(\omega) = \sum f(n) \exp(j\omega n)$
- 離散フーリエ逆変換 $f(n) = \sum F(\omega) \exp(j\omega n)$
- この数式を見て、フーリエ変換を諦めた方も多いでしょう。
- しかし、この式は見事な公式です。
- これまでのお話しを簡潔明瞭に言い表しているのです。

数式によるフーリエ変換2

- 連続フーリエ変換式は、次のように解説できます。
- ① 音や光は一次元の世界なので、時間の関数 $f(t)$ となる。
- ② 音や光は、連続信号なので、 $\int$ （インテグラル）を使う。
- ③ 回転しているということを $\exp(j\omega)$ と記載する。
($\exp$ =エクスポネンシャル=指数、 ω =オメガ=時間)
- ($j\omega$ の「j」は、虚数が入っているという意味)
- ④ 音や光は、時間の関数なので、時計回り（負）とする。
- ⑤ ③と④の結果、回転係数を $\exp(-j\omega t)$ と記載する。
- ⑥ 音や光を微量時間単位に分解して、 dt と記載する。
- ⑦ 微量単位の音や光に回転係数を乗じて周波数を計算する。
- ⑧ 細分した周波数の値をすべて合計（ $\int$ ）する。
- ⑨ 逆回転は反時計回りであるため、正の $\exp(j\omega t)$ とする。

数式によるフーリエ変換

フーリエ変換の数学的表現

(変換)

$$F(\omega) = \int f(t) \exp(j\omega t) dt$$

(逆変換)

$$f(t) = \int F(\omega) \exp(j\omega t) d\omega$$

数学はすばらしい！



画像のフーリエ変換



- 離散フーリエの説明は、割愛いたします(頑張っね)。
- これまで、画像のフーリエ変換と称してお話ししました。
- しかし実際は、1行8列の画像の説明でした。
- これは、一次元の説明であり、音と大差ありません。
- 実際の画像、二次元フーリエ変換はもっと複雑です。
- 「★★横に変換し、その結果を縦に変換する★★」
- という手続きになり、計算量も膨大となります。
- そこで、「高速フーリエ変換 (FFT)」が登場します。
- FFTのご説明は・・・。それは、またの機会に(笑)。

後半のまとめ



- それでは、この章のまとめを行っていきましょう。
- ① 高周波・中周波・低周波は、「相対的な概念」である。
- ② カットフィルタには、利用目的がある。
- ③ 輪郭抽出、ノイズ補正、画像圧縮等利用される。
- ④ 直流カット以外は、色値合計は維持される。
- ⑤ カットフィルタは、色差合計が小さくなる。
- ⑥ 周波数補正は、スペクトルの直接補正ではない。
- ⑦ 周波数補正は、フーリエ計算した x と y を補正する。
- ⑧ フーリエ変換で、畳み込み計算が高速になる。
- ⑨ フーリエ変換の端処理は、反対側の端を対象とする。
- ⑩ 色合計を元値と同じにするため、 $F0=1.0$ を適用する。
- ⑪ 数式は、回りくどい説明を簡潔明瞭に表現できる。