

ブラインド画像回復: 修正 Lucy 法 (その 4)

Blind deconvolution: modified Lucy-algorithm

福留治隆, 片山泰男 (ウィンボンド・エレクトロニクス (株))

Harutaka Fukutome, Yasuo Katayama (Winbond Electronics Corp. Japan)

平成 19 年 7 月 18 日

1 デコンボリューション処理

デコンボリューションの基本的処理は、フィルタ処理、

$$\Phi^{r+1}(\chi) = \int \Psi^r(\xi) p^r(\chi - \xi) d\xi \quad (1)$$

推定画像の修正 (比エラーの転置 PSF フィルタ処理画像を推定画像に掛込む処理)、

$$\Psi^{r+1}(\xi) = \Psi^r(\xi) \int \frac{\Phi^0(\chi)}{\Phi^r(\chi)} p^r(\chi - \xi) d\chi \quad (2)$$

これは、次のように加算的に記述でき、差エラーの推定画素値と PSF との積による分配処理と等しい。

$$\Psi^{r+1}(\xi) = \Psi^r(\xi) + \int (\Phi^0(\chi) - \Phi^r(\chi)) \frac{\Psi^r(\xi)}{\Phi^r(\chi)} p^r(\chi - \xi) d\chi \quad (3)$$

PSF の修正、差のエラーが正の画素について差のエラー (又はその自乗) と推定画像との相互相関からなる。

$$p^{r+1}(\xi') = p^r(\xi') \int (\Phi^0(\chi) - \Phi^r(\chi) (> 0)) \Psi^r(\chi - \xi') d\chi \quad (4)$$

又は、

$$p^{r+1}(\xi') = p^r(\xi') \int (\Phi^0(\chi) - \Phi^r(\chi))^2 \Psi^r(\chi - \xi') d\chi (if \Phi^0(\chi) > \Phi^r(\chi)) \quad (5)$$

2 離散的処理にあるバリエーション

現実のデコンボリューションを数値解で求めるためには、離散的な画像と PSF を使う必要があるが、式 (1) のフィルタ処理は、推定画像 $X_{i-k,j-l}$ に $PSF_{k,l}$ を使って、観測画像に近いぼやけた画素 $Y_{i,j}$ を作成する通常の 2 次元 FIR フィルタである。

$$Y_{i,j} = \sum_{k,l} X_{i-k,j-l} PSF_{k,l} \quad (6)$$

PSF と推定画像値の積を使った差のエラーの分配処理は (1 周辺画素への処理を記述すれば)、次式となる。

$$X_{i-k,j-l} = X_{i-k,j-l} + \frac{(Y_{i,j}^0 - Y_{i,j}^r) X_{i-k,j-l} PSF_{k,l}}{Y_{i,j}^r} \quad (7)$$

これは、次のように (比エラー -1) の推定画像と PSF 積を使った分配処理と等しく、

$$X_{i-k,j-l} = X_{i-k,j-l} + \left(\frac{Y_{i,j}^0}{Y_{i,j}^r} - 1 \right) X_{i-k,j-l} PSF_{k,l} \quad (8)$$

$X_{i-k,j-l}$ を前に出し、 $i-k, j-l$ を i, j と置き換え、同じ画素への分配を集め、

$$X_{i,j} = X_{i,j} + X_{i,j} \sum_{k,l} \left(\frac{Y_{i+k,j+l}^0}{Y_{i+k,j+k}^r} - 1 \right) PSF_{k,l}$$

$\sum_{k,l} PSF_{k,l} = 1$ を使うと、次の掛込み型の式となる。これは、比エラー $Y_{i,j}^0/Y_{i,j}^r$ の転置 PSF によるフィルタ処理画像の推定画像への掛込みである。

$$X_{i,j} = X_{i,j} \sum_{k,l} \frac{Y_{i+k,j+l}^0}{Y_{i+k,j+k}^r} PSF_{k,l} \quad (9)$$

また、式 (8) は、比エラー毎に (比エラー -1)*PSF+1 を周辺の推定画像に (1 周辺画素毎に) 掛込む処理と記述することができる。

$$X_{i-k,j-l} = X_{i-k,j-l} \left(\left(\frac{Y_{i,j}^0}{Y_{i,j}^r} - 1 \right) PSF_{k,l} + 1 \right) \quad (10)$$

上式 (8), (9), 及び (10) のどれを実際の処理に使うかは、結果を変えない処理のバリエーションでは実はない。結果が違って来るのである。エラー画素の周辺推定画像にエラー分配する処理は、エラー画像よりも広い推定画像を想定し、式 (9) の掛込み型の処理は、推定画像より広いエラー画像を想定するから、画像の周辺処理によって、異なる処理となる。式 (8) と式 (10) の違いについては、後述する。

2.1 推定画像修正における周辺推定画像の修正前後の可能性

式 (7) 及び式 (8) において、エラー画素毎に分配比率、PSF と周辺推定画像との積を誤差分配比率として (比エラー -1) を分配するが、このとき、周辺推定画像に修正前の推定画像を使うか、修正中の推定画像を使うかの、処理のバリエーションがある。

転置 PSF フィルタ処理、式 (9) の方法は、比エラー画像を介在するため、そこで使われている推定画像は、修正前の推定画像である。それに対して、誤差分配を掛込みとする処理、式 (10) の方法ではエラーの周辺の推定画素値として、修正中の推定画像が使われている。修正前を使う方法が修正中を使うよりもよいとは、一概に言えることなく、デコンボリューション結果によって判断されるべきであると考え。

2.2 画像の周辺処理に関する問題

推定画像の周辺処理において、推定画像の範囲を超えた画素への読み書きアクセスは、画像の端の画素によって代用する方法が採られた。この画像の周辺処理を使う場合、差のエラーの分配処理 (式 (7) 又は式 (8)) は、ひとつのエラー画素に対して同じ周辺画素への複数回の書き込みアクセスが行われる。そのため、推定画像が負にならないようなチェック処理が必要である。(画像の範囲外への書き込みアクセスを無効にすれば不要になる。)

転置 PSF によるフィルタ処理 (式 (9)) では、推定画像より大きい範囲の比エラー画像を想定するから、エラー画像への周辺例外処理が必要である。これには、比エラー画像範囲を超えた読み出しアクセスには画像の端の画素によって代用するという方法を採用した。この方法では推定画像修正は、推定画像の範囲内であり、推定画素への修正は、単一回しかなされない。

さらに、次に述べるように、実際的に同様な効果をもたらすバリエーションがあり得る。

3 フィルタ処理と推定画像修正の画素処理ループの合体

フィルタ処理と推定画像修正との両者の画素毎の処理を同じ画面内の処理ループにに入れて合体する方法がある。推定画像を PSF を使ってフィルタ画素を作成する毎に、その画素位置の差のエラーを推定画像に分配して推定画像を修正する方法である。

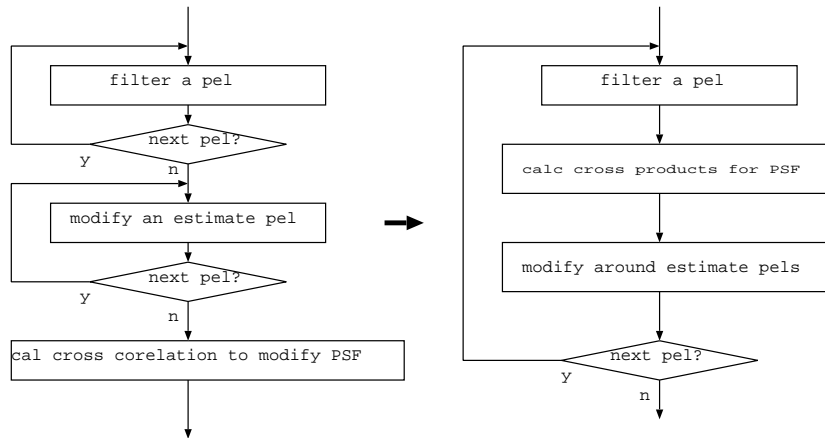


Fig.41 フィルタ処理、PSF 修正のための計算、推定修正を合体した処理

この処理順ではエラーの周辺推定画像の要素積 $X_{i-k,j-l}PSF_{k,l}$ は、フィルタ処理で行われた積計算であるので、新たに計算する必要がない。画像の画素あたり、PSF 範囲の処理ループが 2 回回られるが、フィルタ処理ループにおいて、PSF と周辺推定画像の要素積を保存しておき、後の誤差分配ループでの分配比率に使用する。その総和であるフィルタ画素値も同じ画素位置にあって、再利用でき、再計算又は保持する必要がない。通常処理では、フィルタ処理と推定画像の修正とを画像全体で分離するが、その場合、フィルタ処理相当の再計算が必要である。

このとき、フィルタ処理とエラー分配における周辺の推定画素として、修正前の推定画像を使うか、修正中の推定画像を使うかの 2 種類の方法が可能である。前者は、フィルタ処理の画面処理の前に推定画像のコピーを保存する必要があり、フィルタ処理との処理を合体した処理順変更の効果はない。後者の修正中の推定画像をフィルタ処理に使う場合にだけ、この処理順の意味がある。実際のデコンボリューションの結果において、後者がよい場合が多い。

4 フィルタ処理と PSF 修正計算の画素処理ループの合体

PSF 修正に使われる、相互相関の計算もフィルタ処理のときに計算する方法がある。フィルタ処理と、推定画像修正処理とが画面全体について終了した後、差のエラーが正の画素について、差のエラー (またはその自乗) と推定画像との相互相関を計算する方法ではなく、フィルタ処理のときに求められた差エラーが正の画素について、差のエラー (またはその自乗) と周辺の推定画像との積を累和する。

この方法は、差のエラーを画像として保持する必要がなく、必要メモリー量が小さく有利であり、性能的問題がなければこの方法が望ましい。そして、この方法は、前項の処理と組み合わせることができる。

その場合、フィルタ処理は、推定画像の修正処理と合体するだけでなく、PSF 修正のための相互相関計算も同じ画素ループの中にはいる。PSF 修正のための相互相関計算は、推定画像の修正処理より、先の方が結果が実際の結果ではよいため、フィルタ処理、PSF 修正のための相互相関のクロス積の部分積計算、そして推定画像の修正、という PSF サイズの 3 つのループが同じ画素処理ループの中に並ぶことになる。

デコンボリューション結果は、この最終的合体の方法は、PSF が非常に画素の処理順に影響をうけ、PSF の重心が下に落ちることが多いという欠点があるが、多くの画像のデコンボリューションにおいて、この最終的合体をしない場合よりも結果がよい。その考えられる理由としては、すでに修正中の周辺推定画像を使ってフィルタ計算と PSF を修正のための相互相関計算を行うことによって、処理を先取りしていることによる収束の速さであろうか、と推測する。