

Direct Binary Search 法によるマルチトニング

平野祐樹 中野浩嗣
広島大学大学院工学研究科

概要

本論文では、多値 DBS 法を連続階調画像に適用したときに最も明るい領域と暗い領域に加えて中間調の領域でドットが打たれないクリッピングと呼ばれる現象に対応するために、多値 DBS 法と多値組織的ディザ法のハイブリッドマルチトニング手法の提案を行う。また、ランプ画像に対して既存手法と提案手法を適用し、その有効性について検証を行った。その結果、提案手法を適用することによりクリッピングに対して効果があることが検証できた。

1. はじめに

近年、パーソナルコンピュータやデジタルカメラの普及により、低コストで高性能なプリンタの需要は高まっている。プリンタの種類にはインクジェットプリンタや昇華型プリンタやレーザープリンタなどがあり、インクジェットプリンタが家庭用として最も普及している。インクジェットプリンタが最も普及している理由として、文書画像から写真画像に至るまで様々な種類の画像を印刷できることに加えて、その他のプリンタに比べて価格が低いという点が挙げられる。印刷を行う際に、昇華型プリンタは打つドットの大きさを調節することにより連続階調を表現するのに対して、インクジェットプリンタやレーザープリンタはドットを打つか打たないかにより連続階調を表現する。近年のインクジェットプリンタには、完全な黒色のインクに加えて、それよりも色が薄いライトグレーなどのインクを使用することができるものも多い。このような少数階調出力が可能なプリンタは、白黒二値出力しかできないプリンタに比べて高品質な印刷を行うことができる。

我々が普段パーソナルコンピュータで利用する写真や絵などの画像データの多くは連続階調画像であるが、近代のプリンタの多くは少数階調表現しかできない。そのため、少数色のドットを打つか打たないかによる少数階調表現によって擬似的に連続階調を表現するマルチトニングの品質が、印刷品質の重要な要素となる。

2. マルチトニング

2.1. マルチトニングとは

マルチトニングとは、連続階調画像を白と黒と中間色の少数値からなる多値画像に変換することである。

次に代表的な既存のマルチトニング手法である単純多値化法と多値組織的ディザ法について紹介する。各手法を紹介するにあたり、画像データの取り扱いに

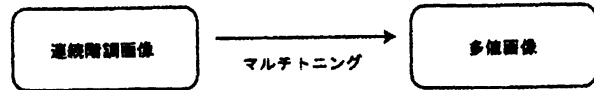


図 1: マルチトニング

ついて次のような条件を定める。

- 入力画像の輝度階調は $[0, 255]$ の 256 階調とする。このとき "0" は黒に対応し、"255" は白に対応している。
- 出力画像の階調数を L 階調とする。(この論文では $L = 3$ として出力している)
- 入力画像である連続階調画像を I とし、位置 (i, j) での入力画像のピクセル値を $I(i, j)$ とする。
- 出力画像である多値画像を O とし、位置 (i, j) での出力画像のピクセル値を $O(i, j)$ とする。出力画像 O は多値画像なので、各ピクセル値は $\frac{255}{L-1}$ の整数倍の整数値のみをとる。

2.2. 単純多値化法

この方法は入力画像 I のピクセル値 $I(i, j)$ と、出力画像 O の輝度値として出力可能な値を比較し、最も誤差が少なくなるものを出力画像 $O(i, j)$ として選択する手法である。例えば、出力階調が 3 階調のときには、以下の式で $O(i, j)$ を計算することができる。

$$O(i, j) = \begin{cases} 255 & \text{if } 191 < I(i, j) \\ 127 & \text{if } 63 < I(i, j) \leq 191 \\ 0 & \text{if } I(i, j) \leq 63 \end{cases}$$

2.3. 多値組織的ディザ法

多値組織的ディザ法は、ピクセル毎に異なる閾値を適用するマルチトニング手法である。このときに使用される閾値を格納した行列はディザ行列と呼ばれている。

今、 $M \times N$ のディザ行列を D とし、 D には 0 から $MN - 1$ までの異なる整数値が格納されているとすると、位置 (i, j) でのディザ行列の値は $D(i \bmod M, j \bmod N)$ に対応し、出力画像となる多値画像 $O(i, j)$ は次のような式で表される。

$$D' = \lfloor \frac{255}{MN(L-1)} (D + \frac{1}{2}) \rfloor$$

$$O(i, j) = \text{Quantizer}\{I(i, j) + D'(i \bmod M, j \bmod N)\}$$

ここで、 $\text{Quantizer}\{\}$ 演算は次のように定義する。

$$\text{Quantizer}\{F\} = \lfloor \frac{255}{L-1} \lfloor \frac{F(L-1)}{255} \rfloor \rfloor$$

3. void-and-cluster 法

この章では void-and-cluster 型ディザ行列の作成法について説明する。この章のデータの取り扱いには次のように定める。

- 輝度階調は [0,1] とする。ここで "0" は白に対応し, "1" は黒に対応している。

void-and-cluster 型ディザ行列は打たれるドットの等方性を高めるために, ガウスフィルタを用いて void 領域と cluster 領域を探すことによって生成される。これにより void-and-cluster 型ディザ行列はアーティファクトの少ない出力画像を得ることができる。ここで, void-and-cluster 法でよく使用されるいくつかの用語について説明する。

- 少数ピクセル
入力されたビットパターン中に含まれている "0" のピクセルの数と "1" のピクセルの数を比較したとき, 数が少ないほうのピクセル。
- 多数ピクセル
入力されたビットパターン中に含まれている "0" のピクセルの数と "1" のピクセルの数を比較したとき, 数が多いほうのピクセル。
- void
多数ピクセルに注目したとき, 少数ピクセルのドットが疎になっている様子を示す。
- cluster
少数ピクセルに注目したとき, 少数ピクセルのドットが密になっている様子を示す。

次の節では, void-and-cluster 法において鍵となる void 領域と cluster 領域の発見法について説明する。

3.1. void 領域と cluster 領域の発見法

void-and-cluster 法のメカニズムは, ガウスフィルタを用いて最も疎な void 領域のドットと最も密な cluster 領域のドットを探索して交換することにより, "0" と "1" の分布が均一になるようにすることである。

位置 (x, y) における $M \times N$ の void-and-cluster 型ディザ行列の原形となる二値パターンを $B(x, y)$, フィルタ係数を $f(x, y)$ とすると, ピクセルの疎密を判断するための $M \times N$ の行列 $V(x, y)$ は次の式で表すことができる。

$$V(x, y) = \sum_{p=-M/2}^{M/2} \sum_{q=-N/2}^{N/2} B(p', q') f(p, q)$$

$$p' = (M + x - p) \bmod M$$

$$q' = (N + y - q) \bmod N$$

$$f(p, q) = e^{-\frac{p^2+q^2}{2\sigma^2}}$$

上式はフィルタのサイズが $M \times N$ の時に利用できる式である。しかし, フィルタのサイズが大きすぎると処理に時間がかかるため, 実際は $\sigma=1.5$, 11×11 のガウスフィルタを使用する [1]。

3.2. 初期二値パターンの作成

図 2 は初期二値パターンの作成法のフローチャートである。void 領域と cluster 領域を調べる際には前節に記述した $V(x, y)$ を利用する。

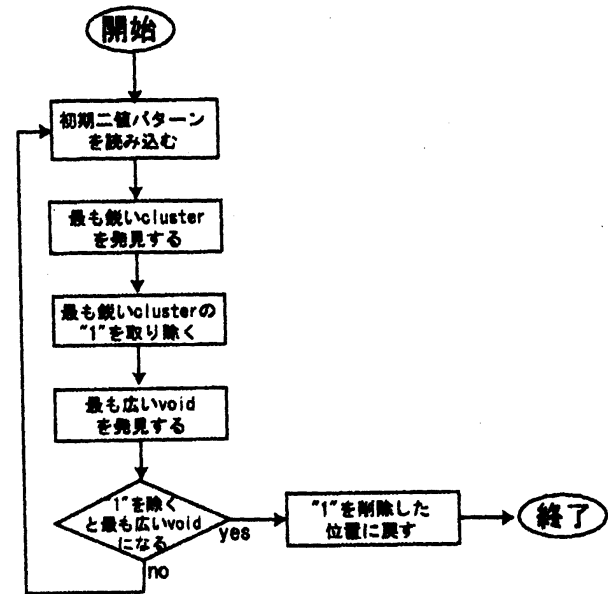


図 2: 初期二値パターンの作成法

3.3. ディザ行列の生成

次に, 初期二値パターンを利用してディザ行列を作成する方法を示す。このときディザ行列は $0 \sim MN-1$ までの Rank という数値を割り当てることにより, 3 段階構成で作成される。

● Phase I

初期二値パターンを読み込み, 初期二値パターン中に含まれている "1" の数の合計を *Ones* とおく。Rank に *Ones-1* を代入する。二値パターンの中から最も密な cluster となるドットを見つけ, 取り除くと同時に Rank をディザ行列に代入し, Rank をデクリメントする。これらの操作は Rank の値をデクリメントしながら, Rank の値が 0 になるまで繰り返す。Phase I が終了するときには, 二値パターンは全て "0" になっている。

● Phase II

初期二値パターンを読み込み, Rank に *Ones* を代入する。二値パターンの中から最も疎な void となるドットを見つけ, "1" を入れると同時に Rank

をディザ行列に代入し, $Rank$ をインクリメントする. これらの操作は $Rank$ の値をインクリメントしながら, $Rank$ の値が $\lfloor \frac{MN}{2} \rfloor$ になるまで繰り返す. Phase II が終了するときには, M と N の数が両方とも奇数だった場合を除いて二値パターン中の"1"の数と"0"の数は等しくなっている.

● Phase III

Phase III では少数ピクセルの意味を"1"から"0"に反転させる. 二値パターンの中から最も密な cluster となるドット (今は"0") を見つけ, "1"を入れると同時に $Rank$ をディザ行列に代入し, $Rank$ をインクリメントする. これらの操作は $Rank$ の値をインクリメントしながら, $Rank$ の値が $MN - 1$ になるまで繰り返す. Phase III が終了するときには, 二値パターンは全て"1"で埋められ, ディザ行列は完成する.

4. 多値 Direct Binary Search 法

この章では多値 Direct Binary Search (DBS) 法による多値化法について説明する. 多値 DBS 法は, 入力画像 I と出力画像 O の誤差 ϵ を最小にする二値化法である. ここで, 出力画像 O を得るために二値画像 O' を使用する. 二値画像 O' の条件は次のとおりである.

- 二値画像 O' は, "0"か"1"の二値のみをとり, "1"は"0"よりも一階調分明るいことを意味している.

次に, 多値 DBS 法の代表的なアルゴリズムを示す.



図 3: 多値組織的ディザ法 (void-and-cluster)

4.1. ガウスフィルタ

この節では, 多値 DBS 法を適用する際に使用するガウスフィルタ \tilde{p} を次のように定める.

$$\tilde{p}(a, b) = \frac{\exp(-\frac{a^2+b^2}{2\sigma^2})}{\sum_{i=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{j=-\frac{N-1}{2}}^{\frac{N-1}{2}} \exp(-\frac{i^2+j^2}{2\sigma^2})}$$



図 4: 多値 DBS 法

ここで, σ はガウスフィルタの標準偏差, N はガウスフィルタの一边の大きさ (奇数) と定める. a と b はそれぞれ $-\frac{N-1}{2}$ から $\frac{N-1}{2}$ までの整数値をとり, ガウスフィルタの中心の座標が $(0, 0)$ となる. 本実験では, $\sigma = 1.2$, $N = 11$ とした.

4.2. 多値 Direct Binary Search 法のアルゴリズム

1. 初期二値画像 O' を作成する
2. 入力画像 I と出力画像 O から誤差 ϵ を計算する
3. ピクセルの更新が出来なくなるまで, 以下の操作を繰り返す
 - (a) 次の操作の内どの操作が最も ϵ を減少させるかを調べる
 - ピクセル $O'(i, j)$ とその 8 近傍のピクセルの交換を行う
 - ピクセル $O'(i, j)$ の反転を行う
 - (b) 上記の内最も ϵ を減少させる操作を行う
 - (c) ϵ を更新する

誤差 ϵ は次のように定める.

$$\epsilon = \sum |\tilde{e}|^2$$

$$\tilde{e}(x, y) = \tilde{p} * (O - I)(x, y)$$

$$O(x, y) = \text{Quantizer}\{I(x, y)\} + \lfloor \frac{255}{L-1} O'(x, y) \rfloor$$

多値 DBS 法で使用される誤差 ϵ の関数は, 人間の視覚特性のモデルに基づいている. 関数中に使用されている \tilde{p} は人間の視覚特性に合ったローパスフィルタを意味し, $*$ は二次元の畳み込み演算を意味している. \tilde{p} は 4.1 節に表されるガウスフィルタが使用される. 初期二値画像 O' はランダムなホワイトノイズ画像が用いられる.

多値 DBS 法による出力画像は, 人間の視覚特性に基づいて作成されているため, 前章で紹介した void-

and-cluster法のディザ行列を使用した多値組織的ディザ法に比べて高品質のものになっている。(図3と図4参照)しかし、全てのピクセルについてフィルタ係数との二次元畳み込み演算を行わなければならないため、既存手法に比べて処理時間がかかり長いという欠点がある。

4.3. クリッピング

多値DBS法は非常に高品質な出力画像を得ることができる一方で、ランプ画像のように連続的に輝度値が変化する入力画像に対する出力画像を得るとき、入力画像の輝度値が $\frac{255}{L-1}$ の整数倍の周辺で、連続的にドットが打たれないクリッピングという現象が発生する。(図5参照)

クリッピングが発生する原因は、誤差 ϵ の計算方法にある。誤差 ϵ の計算方法は二乗誤差の総和なので、入力画像の輝度値が $\frac{255}{L-1}$ の整数倍の周辺でドットを打つと、その点における誤差が極端に増え、ドットを打たなかったときの誤差を超えてしまう。そのため、入力画像の輝度値が $\frac{255}{L-1}$ の整数倍の周辺ではドットが打たれることはない。



図5: クリッピング(左端, 中央, 右端に発生)

5. 提案手法

この章ではクリッピングを考慮した多値 Direct Binary Search(DBS)法による多値化法について提案手法を述べる。

提案手法における基本的な概念は、クリッピングの発生する領域に対して特別な操作を施すことである。

5.1. クリッピングの発生する領域

クリッピングの発生する領域を R 、閾値を T と定める。 R は入力画像の輝度値が $\frac{255}{L-1}$ の整数倍の周辺(距離 T 以内)の領域であるから、次のように定めることができる。

$$t = I(i, j) \bmod \lfloor \frac{255}{L-1} \rfloor$$

$$R = \{(i, j) : (t \leq T) \cup (t \geq \frac{255}{L-1} - T)\}$$

$$Z(a) = \sum_{i=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{j=-\frac{N-1}{2}}^{\frac{N-1}{2}} \left(\left(\frac{a}{255} - \frac{\tilde{p}(i, j)}{L-1} \right)^2 - \left(\frac{a}{255} \right)^2 \right)$$

$$T = \min\{a | Z(a) \leq 0\}$$

閾値 T は出力階調数 L とガウスフィルタの設定により増減する。今回の実験の場合、閾値 T の値は“4”となる。

次に、提案手法のアルゴリズムを示す。

5.2. 提案手法のアルゴリズム

1. $M \times N$ のvoid-and-cluster型ディザ行列 D を作成する。
2. 次の計算により、 D を標準化する。

$$D' = \lfloor \frac{255}{MN(L-1)} (D + \frac{1}{2}) \rfloor$$

3. $(i, j) \in R$ を満たす全てのピクセル (i, j) に対して多値組織的ディザ法を適用する。

$$O(i, j) = \text{Quantizer}\{I(i, j) + D'(i \bmod M, j \bmod N)\}$$

4. 多値組織的ディザ法を適用することにより R の領域に出現する残したいドットを確定ドットとし、確定ドットに対してはピクセルの変更を加えないように多値DBS法を適用し、出力画像 O を得る。

6. 実験結果

図6はランプ画像に対して提案手法を適用した結果である。通常の多値DBS法を適用した図5の結果と比べて、クリッピングを排除できていることがわかる。



図6: 提案手法

7. まとめ

本論文では、既存のマルチトニング手法について紹介し、クリッピングを考慮した多値DBS法を提案した。結果として、提案手法はクリッピングを排除した出力画像を得ることに成功した。

R の領域に適用するマルチトニング手法として、void-and-cluster型のディザ行列を利用した多値組織的ディザ法を使用することにより、多値DBS法と同様に視覚的に満足できるブルーノイズ特性を持った模様を表現可能となった。その模様を残しつつ、多値DBS法を適用して入力画像と出力画像の誤差を最小にするため、 R の領域と R ではない領域の境界が目立つことはない。

計算コスト面では、多値組織的ディザ法のコストは多値DBS法の計算コストに比べてかなり小さいため、多値DBS法と多値組織的ディザ法のハイブリッド

ド多値化法である提案手法の計算コストは、通常の多値 DBS 法と比べてあまり変わらないとすることができる。

参考文献

- [1] Robert Ulichney, *The void-and-cluster method for dither array generation*, Proceedings SPIE, Human Vision, Visual Processing, Digital Displays IV, vol.1913, pp.332-343, 1993.
- [2] Sagar Bhatt, John Harlim, Joel Lepak, Robert Ronkese, John Sabino, Chai Wah Wu, *Direct Binary Search with Adaptive Search and Swap*, pp.1-4, 2005.