

画素値順ポアソンディスクサンプリングによる点描画

李 少龍¹ 王 濤¹ 井上 光平^{1,a)} 原 健二^{1,b)} 小野 直樹^{1,c)} 浦浜 喜一^{1,d)}

概要：点描画を生成するポアソンディスクサンプリング(PDS)では、通常ランダム順に画素をスキャンするが、物体輪郭や細かい線が不明瞭になる。そこで本稿では、入力濃淡画像の画素値順にスキャンするPDSを提案し、灰色の背景のなかに白点と黒点を打つ2値点描画にも拡張する。この2値点描画は、通常の白地に黒点や黒地に白点の点描画よりも入力画像の再現性が高いことを示し、カラー画像での実験例も示す。

1. はじめに

点描画(stippling)の生成法としてポアソンディスクサンプリング(PDS)[1]やLloyd法[2]などがよく使われる。Lloyd法はPDSよりも点配置の均一性が高いが、計算量が多く濃淡変化のコントラストが低い。コントラストを高める改善法も提案されている[3]が依然として計算量が多い。従って本稿では、計算が容易で濃淡コントラストが高いPDSを用いて点描画を生成する。通常のPDSでは、ランダムに画素をスキャンするので、入力画像のなかの線状の構造やテクスチャなどの模様の保存性が低い。誤差拡散法によるハーフトーニングでは、構造の保存性を高めた改善法が提案されている[4],[5],[6]。またPangら[7]は最適化問題による定式化を提案しているが、シミュレーテッドアニーリングで解を求めるので計算量が多い。

本稿では、手順が単純で計算量も少ないPDSについて、同様な構造保存点描画の生成法を提案する。通常のPDSではランダム順に画素をスキャンして点を打っていく。ランダム順スキャンはラスタースキャンの誤差拡散よりも人工的パターンが生じにくく、性質のよい点描画が得られる。しかし、入力画像の濃淡に関係なくランダムスキャンすると、入力画像の模様の保存性が低下する。そこで本稿では、入力画像の画素値順に画素をスキャンするPDSを提案し、ランダム順PDSよりも入力画像の再現性が高いことを示す。また、白地に黒点や黒地に白点を打つ通常の点描画よりも、灰色地に白点と黒点を打つ2値点描画のほうが汎用性が高いことを示し、カラー画像への応用例も示す。

2. PDSによる点描画

モノクロ濃淡画像を点描画に変換する手法として、通常のランダム順PDSは入力画像の保存性が低いことを示し、改善法として画素値順PDSを提案する。

2.1 ランダム順PDS

濃淡モノクロ画像(サイズ(総画素数)を n とする)を点描画に変換する通常のPDSの手順は以下である。

[ランダム順PDSの手順]：

1. 各画素 (i, j) での半径 r_{ij} を設定する。
2. 全ての画素にランダム順に番号 $k(=1, \dots, n)$ を付ける。
3. $k=1$ の画素に点を打つ。
4. k を1増やす。 $k=n$ なら終了。 $k < n$ ならステップ5へ。
5. 番号が k の画素 (i, j) を中心として半径 r_{ij} の円内に、まだ点が打たれてなければ、その画素に点を打ってステップ4へ。既に点が打たれていれば点を打たずにステップ4へ。

このPDSによる点描画でのエッジや細かい線の保存性を検証する。まず、エッジが分かりやすい図1(a)の入力画像から作った点描画を図1(b)に示す。半径 r_{ij} は、入力画像の画素値を $d_{ij} \in [0, 255]$ とすると $r_{ij} = a + bd_{ij}/255$ とした。図1(b)は $a=3, b=15$ での結果である(図1(b)では眼や口が黒く塗り潰されているが、これは点が見やすいように点を大きく描いたためであり、実際の点は癒着していない)。図1(b)では顔の輪郭が所々で乱れている。これは、図1(b)に図1(a)のエッジを重ね描きした図1(c)(左下部の拡大を図1(d)に示す)から分かるように、エッジのすぐ外にも点が打たれているので、それらの点が輪郭の内側か外側なのかが不明瞭なためである。このように、ランダム順に点を打つと、顔の外側に先に点が打たれて、その後

¹ 九州大学 芸術工学研究院, 福岡市
Kyushu University, Fukuoka 815-8540, Japan

a) k-inoue@design.kyushu-u.ac.jp

b) hara@design.kyushu-u.ac.jp

c) ono@design.kyushu-u.ac.jp

d) urahama@design.kyushu-u.ac.jp

で内側に点が打たれると、顔の外の点での半径以内にも、内側の点が打たれ、外側の点が内側に属すように見えてしまい、輪郭の形状が乱れる。

次に、線状の模様が含まれる図 2(a) の画像の点描画を図 2(b) に示す。図 2(b) では点は 1 画素のままである。点どうしが癒着しないように $a = 1, b = 4$ とした。大きな螺旋のための黒線が乱れており、それから出ている細い直線もほとんど視認できない。

このように、ランダム順の PDS は入力画像中の構造や模様の保存性が低い。

2.2 画素値順 PDS

そこで、画素のスキャンをランダム順でなく、半径が小さい暗い画素から順番にサンプリングするように、上記の手順を以下のように変える。

[画素値順 PDS の手順] :

1. 各画素 (i, j) での半径 r_{ij} を設定する。

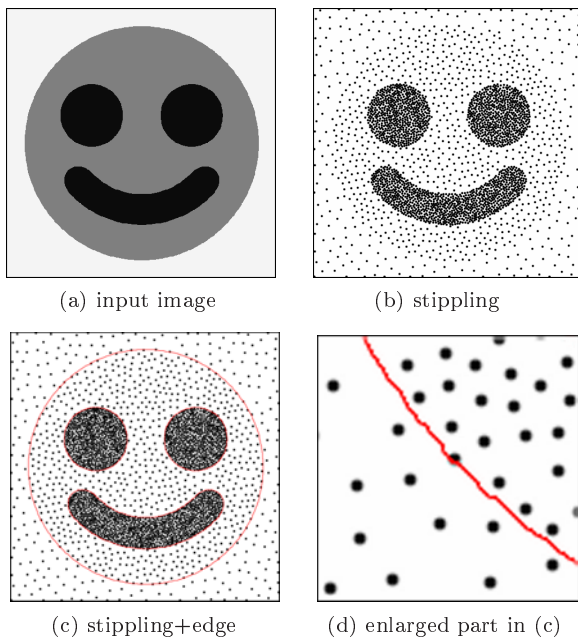


図 1 ランダム順 PDS による点描画 (エッジ保存性)

Fig. 1 Stippling by random-order PDS (edge preservation).

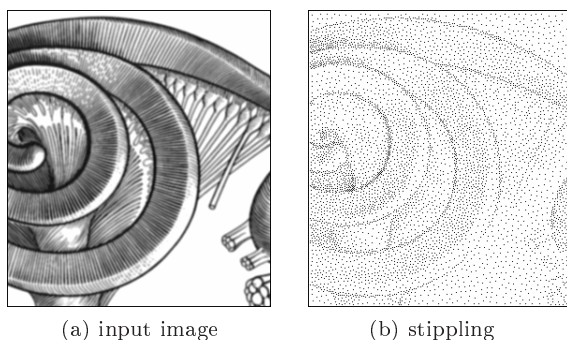


図 2 ランダム順 PDS による点描画 (線の再現性)

Fig. 2 Stippling by random-order PDS (clarity of lines).

2. 各画素の値 d_{ij} に値が 0 以上 1 以下の一様乱数 n_{ij} を加えて $f_{ij} = d_{ij} + n_{ij}$ とする。

2. f_{ij} が小さい順番に、全ての画素に番号 $k (= 1, \dots, n)$ を付ける。

3. $k = 1$ の画素に点を打つ。

4. k を 1 増やす。 $k = n$ なら終了。 $k < n$ ならステップ 5 へ。

5. 番号が k の画素 (i, j) を中心として半径 r_{ij} の円内に、まだ点が打たれてなければ、その画素に点を打ってステップ 4 へ。既に点が打たれていれば点を打たずにステップ 4 へ。

図 1(a) の画像での、この画素値順 PDS の結果を図 3(a) に示す。図 3(a) では、図 1(b) のようなエッジのすぐ外に打たれた点がなく、エッジが明瞭である (従ってエッジの重ね描きは省略する)。図 3(b) は図 1(d) と同じ部位の拡大である (これにはエッジを重ね描きしている)。エッジのすぐ外には点が打たれていないので、図 1(b) よりも顔の輪郭が明瞭である。また、図 2(a) の画像での画素値順 PDS の結果を図 4 に示す。螺旋の黒線も乱れておらず、細い直線も視認できる。

3. 2 値点描画

このように、通常のランダム順の PDS よりも画素値順の PDS のほうが、入力画像の構造の保存性が高いことが確認された。そこで次に、画素値順 PDS について、どの画像でも暗い順番でよいかを検証する。以上の点描画では、

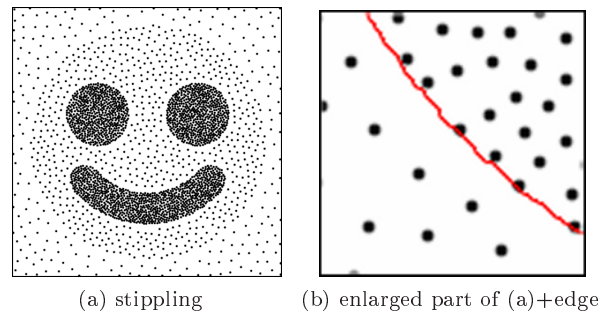


図 3 図 1(a) の画像での画素値順 PDS による点描画

Fig. 3 Stippling by grayscale-order PDS for Fig.1(a).

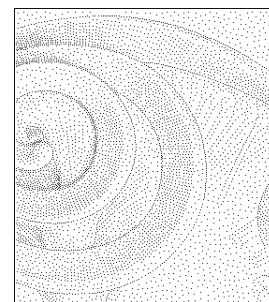


図 4 図 2(a) の画像での画素値順 PDS による点描画

Fig. 4 Stippling by grayscale-order PDS for Fig.2(a).

通常通り白い背景に黒い点を打った．図 1(a) や図 2(a) のように，入力画像の背景が白い場合にはこれでよいと思われるが，背景が黒い場合にはこの点描画は不適切な可能性がある．

そこで，図 2(a) を白黒反転した図 5(a) の画像では，図 5(b) のような点描画になった．点描画の背景は白で点が黒なので，図 5(a) のなかの白い線は図 5(b) では点がない空隙になっている．それでも線は視認できるが，図 5(a) の線よりも太くなっており，図 4 のような細い線としては認知され難い．

3.1 ネガ点描画

従って，図 5(a) のように黒い背景のなかに白い模様が描かれている入力画像では，点描画も黒い背景に白い点を打つ方がよい．そのためには，入力画像の画素値を d_{ij} とすると，半径を $r_{ij} = a + b(255 - d_{ij})/255$ として，番号を上記の手順の k から $n - k + 1$ に変えればよい．すなわち明るい画素から順番にスキャンする．図 5(a) の画像での点描画を図 6 に示す．これは図 4 を白黒反転させたものと同じである．

3.2 2 値点描画

以上のように，白い背景に黒い模様が描かれた入力画像では白背景に黒点の点描画がよく，黒い背景に白い模様が描かれた入力画像では黒背景に白点の点描画のほうがよい．しかし，部分的に黒い領域と白い領域とが混在した背

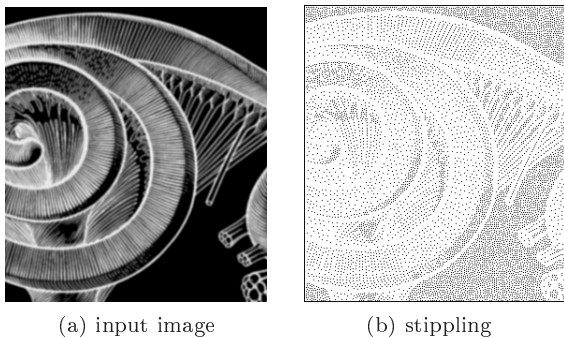


図 5 図 2(a) のネガ画像の点描画
 Fig. 5 Stippling for negative image of Fig.2(a).

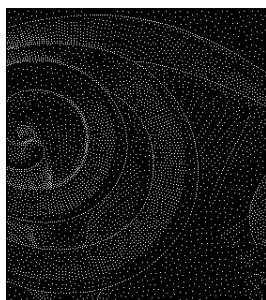


図 6 図 5(a) の画像での黒背景に白点の点描画
 Fig. 6 White-point stippling in black background for Fig.5(a).

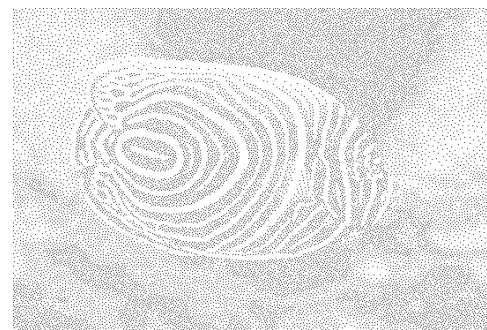
景や灰色の背景のなかに，白い線と黒い線の両方が描かれている場合も多い．従って一般には，点描画も背景を灰色にして，そのなかに白い点と黒い点の両方を打つほうがよいと考えられる．そのような白黒 2 値の点描画を生成する画素値順 PDS の手順は以下となる．

準備：

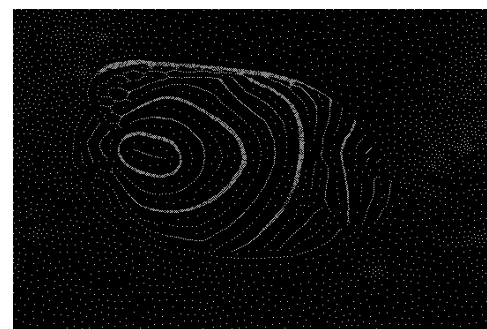
画素の数を n とする．入力画像の画素 (i, j) の値を d_{ij} とする． $f_{ij} = d_{ij} +$ 乱数とする．各画素 (i, j) に 2 つの



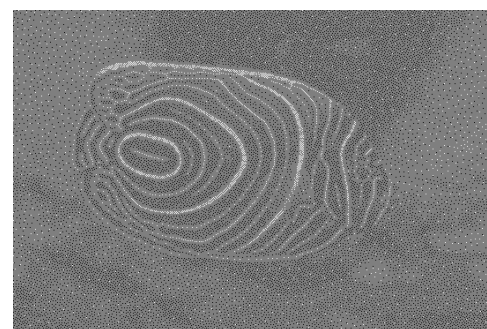
(a) input image



(b) usual stippling



(c) negative stippling



(d) binary stippling

図 7 グレイスケール画像例 1

Fig. 7 Example 1 of grayscale image.

番号 x_{ij} と y_{ij} を付ける． x_{ij} は暗い順番， y_{ij} は明るい順番である．つまり， f_{ij} が小さい順番に x_{ij} を付け， $y_{ij} = n - x_{ij} + 1$ とする．半径も各画素に 2 つ付ける．黒点の半径を $s_{ij} = a_b + b_b * d_{ij}/255$ ，白点の半径を $t_{ij} = a_w + b_w * (255 - d_{ij})/255$ とする．

[2 値点描画の画素値順 PDS の手順]：

1. $k = 1$ とする． $x_{ij} = k$ の画素に黒点を打つ． $y_{ij} = k$ の画素に白点を打つ．
2. $k = k + 1$ とする．もし $k = n$ なら終わり． $k < n$ ならステップ 3 へ．
3. $x_{ij} = k$ の画素 (i, j) に白点が打たれていたらステップ 5 へ．打たれてないならステップ 4 へ．
4. その画素を中心とする半径 s_{ij} の円のなかに黒点はまだ

打たれてないなら，その画素に黒点を打ってステップ 5 へ．打たれていたら黒点を打たずにステップ 5 へ．

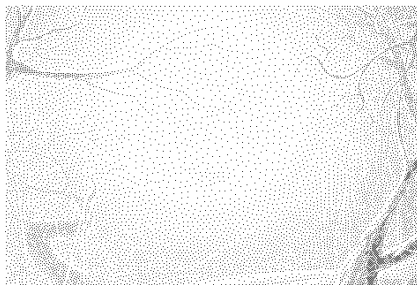
5. $y_{ij} = k$ の画素 (i, j) に黒点が打たれていたらステップ 2 へ．打たれてないならステップ 6 へ．

6. その画素を中心とする半径 t_{ij} の円のなかに白点はまだ打たれてないなら，その画素に白点を打ってステップ 2 へ．打たれていたら白点を打たずにステップ 2 へ．

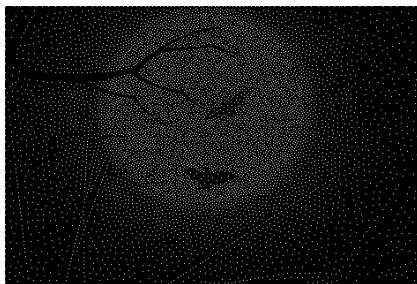
実験結果を図 7～図 9 に示す．まず，図 7(a) の入力画像は，黒い背景のなかに白い線がある．図 7(b) は白い背景に黒い点を打った点描画で，白い線が太くなっており，元々の線の幅が分かり難くなっている．図 7(c) は黒い背景に白



(a) input image



(b) usual stippling



(c) negative stippling



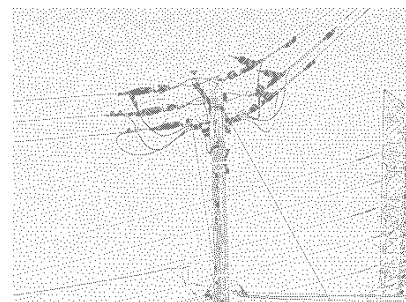
(d) binary stippling

図 8 グレイスケール画像例 2

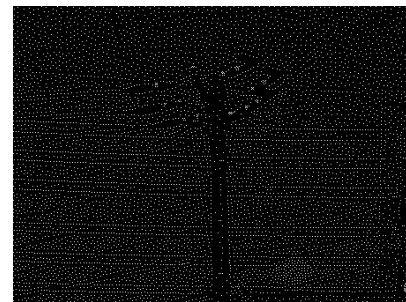
Fig. 8 Example 2 of grayscale image.



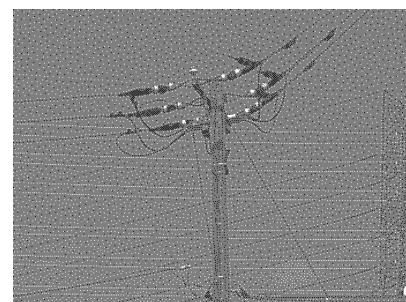
(a) input image



(b) usual stippling



(c) negative stippling



(d) binary stippling

図 9 グレイスケール画像例 3

Fig. 9 Example 3 of grayscale image.

い点を打った点描画で、白い線が分かりやすいが、背景の濃淡は均されてしまって分かり難い。図 7(d) は 2 値点描画で、線も分かりやすく、背景の濃淡変化も視認できる。

次の図 8(a) は、灰色の背景の中に白い月や白い直線と黒い木の幹や枝がある入力画像である。図 8(b) は白い背景に黒い点を打った点描画で、月や白線がほとんど分からない。図 8(c) は黒い背景に白い点を打った点描画で、白い線が分かりやすいが、背景の濃淡は均されてしまって分かり難い。図 8(d) は 2 値点描画で、木の枝や白い線も分かりやすい。

図 9 は、灰色の背景のなかに黒い線と白い線とが混在する画像である。図 9(b) は白い背景に黒い点を打った点描画で、白い線が分かり難い。図 9(c) は黒い背景に白い点を打った点描画で、黒い線がほとんど分からない。図 9(d) は 2 値点描画で、白い線も黒い線も分かりやすい。

以上のように、通常の白地に黒点や黒地に白点の点描画よりも、灰色地に白点と黒点の 2 値点描画のほうが入力画像の模様の再現性が高い。しかし、白地に黒点や黒地に白点の点描画では背景と点との明暗差が 255 であるのに対し、

2 値点描画ではその半分しかないので、コントラストが低い。そこで、図 7(d)、図 8(d)、図 9(d) の点のサイズを少し大きくしたのを図 10 に示す。点の間隔が狭い所では点同士が癒着して点描画よりもディザ画像に近づいてしまっているが、図 7(d)、図 8(d)、図 9(d) よりもコントラストが上がり、入力画像の模様の再現性が向上している。

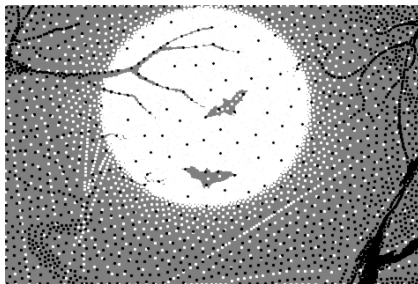
4. カラー画像の実験例

以上ではモノクロ画像について述べたが、カラー画像の場合は、入力画像を R,G,B 成分に分け、それら 3 枚のモノクロ画像をそれぞれ点描画に変換し、それらの点描画を結合してカラー点描画にする。各成分画像を 2 値点描画にすると、R,G,B それぞれが白、灰色、黒の 3 値になるので、結合したカラー点描画は $3^3 = 27$ 色となる。

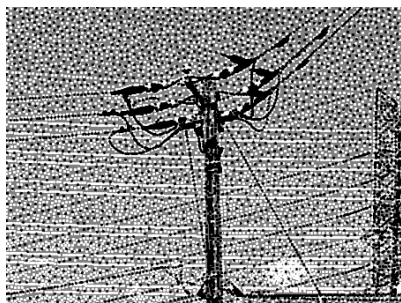
図 11(a) の入力画像について、R,G,B それぞれを 2 値点描画に変換してカラーに戻した結果を図 11(b) に示す。図 7~9 と同じく、背景が灰色で点が小さいのでコントラストが低い。図 10 と同様に、図 11(b) の点のサイズを少し大きくした結果を図 12(a) に示す。図 10 と同様に、部分的に点同士が癒着しているが、コントラストが上がっている。このカラー点描画においても入力画像の細かい模様が再現



(a) Fig.7(d) with large points



(b) Fig.8(d) with large points



(c) Fig.9(d) with large points

図 10 点を大きくした 2 値点描画

Fig. 10 Binary stippling with large-size points.



(a) input color image



(b) color stippling

図 11 カラー点描画

Fig. 11 Color stippling.



(a) mild enlargement



(b) more enlargement

図 12 点を大きくしたカラー点描画

Fig. 12 Color stippling with large-size points.

- [4] J. Chang, B. Alain, and V. Ostromoukhov: *Structure-aware error diffusion*, ACM Trans. Graph., 28, 5, (2009).
- [5] H. Li and D. Mould: *Contrast-aware Halftoning*, Comput. Graph. Forum, 29, 2, pp.273-280 (2010).
- [6] H. Li and D. Mould: *Structure-preserving stippling by priority-based error diffusion*, Proc. Graphics Interface, pp.127-134 (2011).
- [7] W.-M. Pang, Y. Qu, T.-T. Wong, D. Cohen-Or, and P.-A. Heng: *Structure-aware halftoning*, ACM Trans. Graph., 27, 3, (2008).

されているのは、画素値順 PDS の効用である。なお、図 12(a) では背景の灰色がまだ残っているが、点を更に拡大して図 12(b) にすると、背景がほとんど点で塗り潰され、鮮やかさが増すが、細部の模様も塗り潰されてしまう。

5. おわりに

入力濃淡画像の画素値順にスキャンする PDS を提案し、灰色の背景のなかに白点と黒点を打つ 2 値点描画にも拡張した。この 2 値点描画は、通常の白地に黒点や黒地に白点の点描画よりも入力画像の再現性が高いことを示した。本稿では、カラー画像は RGB 各成分をモノクロ点描画にしてから結合してカラー点描画としたが、2 値点描画を多値に拡張して、直接カラー点描画を生成する方法も考えられる。そのような多値点描画への拡張が今後の課題である。

参考文献

- [1] R. L. Cook: *Stochastic sampling in computer graphics*, ACM Trans. Graph., 5, 1, pp.51-72 (1986).
- [2] A. Secord: *Weighted Voronoi stippling*, Proc. NPAR, pp.37-43 (2002).
- [3] M. Balzer: *Capacity-Constrained Voronoi Diagrams in Continuous Spaces*, Proc.ISVD, pp.79-88 (2009).