九州大学学術情報リポジトリ
Kyushu University Institutional Repository

画像パッチの模様への置き換えによるチャック・クローズ的ピクセルアートの生成

井上，光平
九州大学大学院芸術工学研究院 ： 准教授

崔，文一
九州大学大学院芸術工学府

原，健二
九州大学大学院芸術工学研究院 ： 教授

小野，直樹
九州大学大学院芸術工学研究院 ： 准教授

他

https://hdl.handle.net/2324/4123612

出版情報：画像の認識・理解シンポジウム．(IS3-2-7), pp.1-4, 2020-08-05. 画像の認識・理解シンポジウム (MIRU)
バージョン：
権利関係：

KYUSHU UNIVERSITY

CONFIDENTIAL EXTENDED ABSTRACT.
DO NOT DISTRIBUTE ANYWHERE.

画像パッチの模様への置き換えによるチャック・クローズ的ピクセルアートの生成

井上 光平[1,a)]　崔 文一[1]　原 健二[1]　小野 直樹[1]　平岡 透[2]

概要

画像を正方形のパッチに分割し、各パッチを様々な模様に置き換えて表現する新たなピクセルアート生成法を提案する。この方法は、画面を格子分割して写実的な肖像画を描くチャック・クローズの作品にヒントを得た。その具体例として、明るい画素をパッチの中心部に集中させてパッチが輝いているように見えるブリリアントピクセルアートと、パッチ内に 2 つの同心円を描く丸模様ピクセルアートの生成例を示す。

1. はじめに

ピクセルアートはドット絵とも呼ばれ、『スクリーン上の画像の最小単位を指す「ピクセル」（画素）によって描かれた平面作品』[1] とされ、ピクセルの存在を意識しているという点で、高解像度のデジタル画像と区別される。ピクセルアートは、解像度や色数が制限された初期のコンピュータゲームの世界で独自の発展を遂げた表現であるといえる。コンピュータの性能が向上した今日では当時のような技術的制約はなくなったが、『レトロゲームへのノスタルジーやゲーム周辺文脈から一歩離れた独立したアート形式へと発展し、SNS をメディアとして、より自由でより多様な作品が生み出されるようにな』[1] り、芸術表現の一形態として認知されてきている。

ピクセルアートを連想させる絵画を描く画家のひとりにチャック・クローズ [2] がいる。彼は、画面を格子状に分割し、各四角の中に曲線的な模様を描きながら、遠目に見ると人物の顔が写実的に見える絵を描いた。本研究では、彼の作品にヒントを得て、ピクセルアートの各画素の中に模様を描くチャック・クローズ的ピクセルアートを提案する。ピクセルアートの各画素は、元の高解像度画像上では、正方形のパッチに対応する。パッチ内のすべての画素の色を同じにするのが通常のピクセルアートであるが、ここでは、パッチ内の表現を工夫することによって新しいピクセルアートの表現を試みる。具体的には、パッチ内の色を強調して、画素を明るさの順にらせん状に再配置するブリリアントピクセルアートと、パッチ内に 2 つの同心円を描く丸模様ピクセルアートの 2 種類を実装する。

以下、2 節でブリリアントピクセルアートと丸模様ピクセルアートの生成手順を説明し、3 節でそれらの生成例を示す。

2. ピクセルアートの生成

本節では、入力されたカラー画像からチャック・クローズ的ピクセルアートを生成する方法を提案する。

入力カラー画像を F とする。F 上の座標 (i, j) における画素の RGB 値を r_{ij}, g_{ij}, b_{ij} とし、それらを要素とするベクトル（色ベクトル）を $\boldsymbol{f}_{ij} = [r_{ij}, g_{ij}, b_{ij}]$ とする。F の縦、横方向の画素数をそれぞれ m, n とし、この $m \times n$ 画素の画面に、左上詰めで、$h \times h$ 画素の正方形パッチをすき間なく敷き詰める。縦、横方向のパッチの枚数をそれぞれ M, N とし、m, n との関係を $m = Mh + r_M$, $n = Nh + r_N$ とする。ここで r_M, r_N はパッチからはみ出す F の領域の幅を表しており、その大きさは $0 \leq r_M < h$, $0 \leq r_N < h$ である。はみ出した領域は切り捨て、$M \times N$ 枚のパッチを別の模様に置き換えることによって、チャック・クローズ的なピクセルアートを生成する。以下では、ブリリアントピクセルアートと丸模様ピクセルアートの 2 つを詳しく説明する。

2.1 ブリリアントピクセルアート

上では、画像 F を $M \times N$ 枚のパッチに分割した。ここでは、各パッチについて同じ処理を行う。そこで、1 つのパッチに着目して説明する。着目したパッチに含まれる画素の色ベクトルを改めて $\boldsymbol{f}_k = [r_k, g_k, b_k]$ $(k = 1, 2, \ldots, h^2)$ とする。$\{r_k, g_k, b_k\}$ の中の任意の 1 つを $x_k \in \{r_k, g_k, b_k\}$ とすると、その値は $x_k \in \{0, 1, \ldots, 255\}$ を満たすとする。

まず、平均色ベクトルを $\bar{\boldsymbol{f}} = \sum_{k=1}^{h^2} \boldsymbol{f}_k / h^2$ によって求める。次に、RGB 色空間において、平均色ベクトル $\bar{\boldsymbol{f}}$ を中心とする色ベクトル $\{\boldsymbol{f}_k\}_{k=1}^{h^2}$ の分布を考え、その分布が RGB 色立方体に接するように分布を広げる。この操作は、

1 九州大学大学院芸術工学研究院
2 長崎県立大学情報システム学部情報システム学科
a) k-inoue@design.kyushu-u.ac.jp

パッチ内の色の彩度やコントラストを向上させることを目的としている。まず、各色ベクトルを

$$\boldsymbol{f}'_k = \alpha\left(\boldsymbol{f}_k - \bar{\boldsymbol{f}}\right) + \bar{\boldsymbol{f}} \tag{1}$$

によって移動するとし、どの $\boldsymbol{f}'_k$ も RGB 色立方体の外に出ないという制約の下で、α を最大化する。式 (1) を各ベクトルの要素を用いて表せば、

$$x'_k = \alpha\left(x_k - \bar{x}\right) + \bar{x} \tag{2}$$

となる。$\boldsymbol{f}'_k$ が RGB 色立方体の境界にあるとき、その要素の中に $x'_k = 0$ または $x'_k = 255$ を満たすものが存在する。このことを考慮して式 (2) を α について解くと、

$$\alpha_{x,k} = \begin{cases} \frac{-\bar{x}}{x_k - \bar{x}} & \text{if} \quad x_k - \bar{x} < 0 \\ \frac{255-\bar{x}}{x_k - \bar{x}} & \text{if} \quad x_k - \bar{x} > 0 \\ 0 & \text{if} \quad x_k - \bar{x} = 0 \end{cases} \tag{3}$$

となる。すべての $\boldsymbol{f}'_k$ を RGB 立方体内に収めるには、式 (2) の α を

$$\alpha = \min_{x,k}\left\{\alpha_{x,k}\right\} \tag{4}$$

とすればよい。

以上のようにして求めた $\boldsymbol{f}'_k$ の RGB 値を r'_k, g'_k, b'_k とすると、その平均値 $\bar{x}'_k = (r'_k + g'_k + b'_k)/3$ を求め、$\bar{x}'_k$ の小さい順に $\boldsymbol{f}'_k$ を並べ替え、その順番でパッチ内にらせん状に色を再配置する。後述の生成例では、パッチの左上の画素を始点として、時計回りに色を再配置した。そのようにすると、パッチの中心部に近づくほど明るい色が配置され、パッチが輝いているように見える。そこで、この方法で得られる画像をブリリアントピクセルアートと呼ぶ。

2.2 丸模様ピクセルアート

次の方法は、パッチ内に大小 2 つの同心円を描くものである。図 1 に示すように、パッチ内は 2 つの円によって 3 つの領域に分けられる。その 3 領域の色を次のようにして求める。

まず、いちばん外側の領域は平均色 $\bar{\boldsymbol{f}}$ で塗る。各色 $\boldsymbol{f}_k$ について、RGB 値の平均値 $\bar{x}_k = (r_k + g_k + b_k)/3$ を求め、その最小値と最大値をそれぞれ $\bar{x}_{\min}$, $\bar{x}_{\max}$ とし、それらに対応する色を $\boldsymbol{f}_{\min}$, $\boldsymbol{f}_{\max}$ とする。$\bar{x} - x_{\min} < x_{\max} - \bar{x}$ ならば、大きいほうの円の色を $\boldsymbol{f}_{\max}$ に、小さいほうの円の色を $\boldsymbol{f}_{\min}$ にし、そうでなければ両者の色を入れ替える。このようにすると、円のパターンが目立ちやすくなる。

次に、円の半径を決める。ここでは、大きいほうの円の半径は、$r_{\max} = \beta h/2$ に固定することにし、後述の生成例では $\beta = 0.9$ とした。小さいほうの円の半径 $r_{\min}$ を、次のようにして求める。今、$\bar{x}$, $x_{\min}$, $x_{\max}$ の位置関係が図 2 のようになっているとする。ここで $0 \le \gamma \le 1$ である。

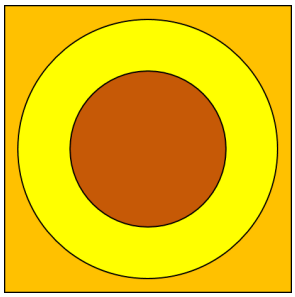


図 1　パッチ内の同心円

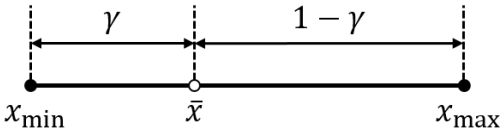


図 2　3 つの明度の関係

図 3　入力画像：(左) Lena、(右) Peppers

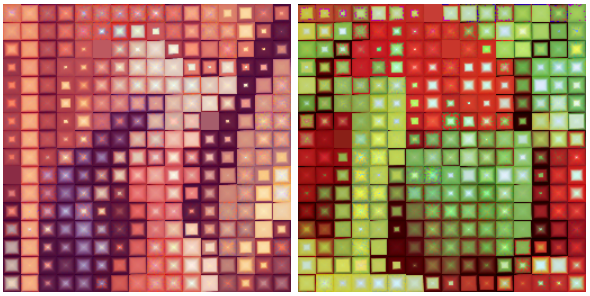


図 4　図 3 のブリリアントピクセルアート

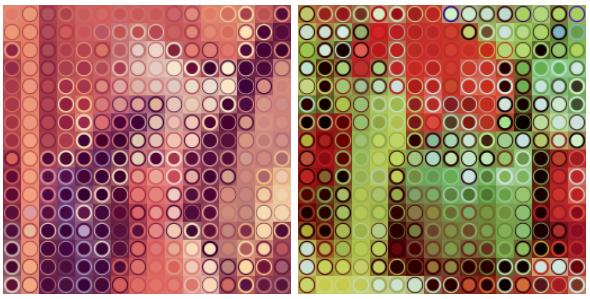


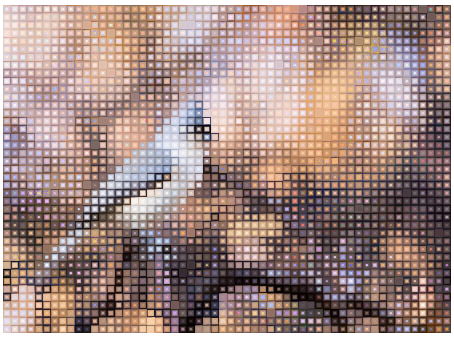
図 5　図 3 の丸模様ピクセルアート

すなわち、$(\bar{x} - x_{\min}) : (x_{\max} - \bar{x}) = \gamma : (1 - \gamma)$ とする。このとき、図 1 のドーナツ状の黄色い領域と丸い茶色の領域との面積比が図 2 の内分比に等しくなるように $r_{\min}$ を定める。$\gamma < \frac{1}{2}$ ならば $r_{\min} = \sqrt{1-\gamma}\, r_{\max}$ であり、$\gamma \ge \frac{1}{2}$

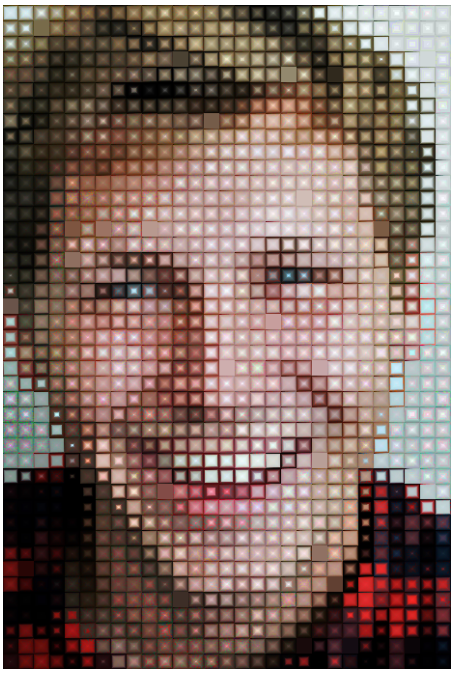
(a) 663 × 900 画素

(b) 700 × 467 画素

図 6 画素数の多い入力画像: (a) Bird、(b) Young man



(a)



(b)

図 7 図 6 のブリリアントピクセルアート

ならば $r_{\min} = \sqrt{\gamma}\, r_{\max}$ である。

以上のようにして 2 つの円の色と半径が定まり、それらをパッチ上に描くと、丸い模様が出来上がる。この方法で得られる画像を丸模様ピクセルアートと呼ぶ。

3. 生成例

上記の方法でピクセルアートを生成した例を示す。画像処理研究用標準画像データベース SIDBA [3] から選んだ 2 枚のカラー画像を図 3 に示す。どちらも画素数は 256×256 である。これらから生成したブリリアントピクセルアートを図 4 に示す。パッチの一辺の画素数は $h = 16$ とした。明度の高い画素を含むパッチが輝いて見える。

図 3 の画像から生成した丸模様ピクセルアートを図 5 に示す。異なる色を含むパッチでは丸い模様がよく目立ち、パッチ内の色変化が小さいと丸模様は薄くなる。

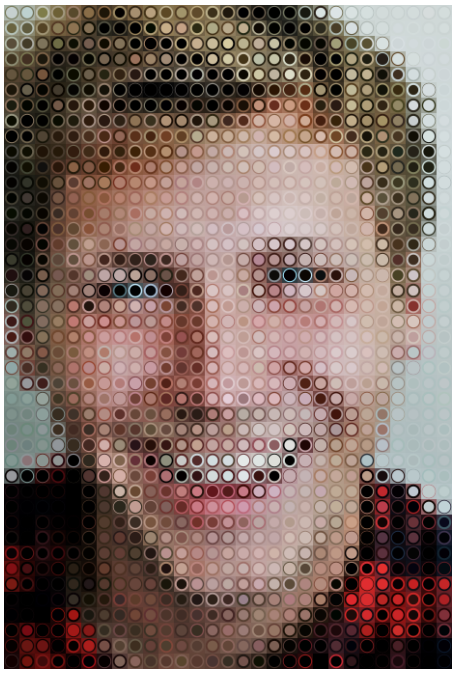
図 6 に画素数の多い画像の例を 2 つ示す。これらから生成したブリリアントピクセルアートを図 7 に、丸模様ピクセルアートを図 8 に示す。画素数が増えても同様の結果が得られた。

図 9(a),(b) に、パッチの大きさをそれぞれ $h = 32,\ 64$ にした結果を示す。図 9(a) では、$r_M = 28,\ r_N = 19$ の幅が、同図 (b) では、$r_M = 60,\ r_N = 19$ の幅がパッチからはみ出したためカットされている。

4. おわりに

ピクセルアートの新しい表現方法として、チャック・クローズの作品にヒントを得たチャック・クローズ的ピクセルアートを提案した。その具体例として、ブリリアントピクセルアートと丸模様ピクセルアートを実装し、いくつかの画像で生成例を示した。提案法は、ピクセルアートのピクセルをパッチに置き換えて、ピクセルアートの表現の幅を広げるものであり、パッチの表現を工夫することにより、多様な表現が可能になると考えられる。また、丸模様ピクセルアートは、円を使って画面を構成する Digital circlism [4] とも関連があると考えられることから、今後、それについても検討していきたい。

(a) $M = 41,\ N = 56$



(b) $M = 43,\ N = 29$

図 8 図 6 の丸模様ピクセルアート（$h = 16$）

謝辞 本研究は JSPS 科研費 JP19K12664 の助成を受けたものです。

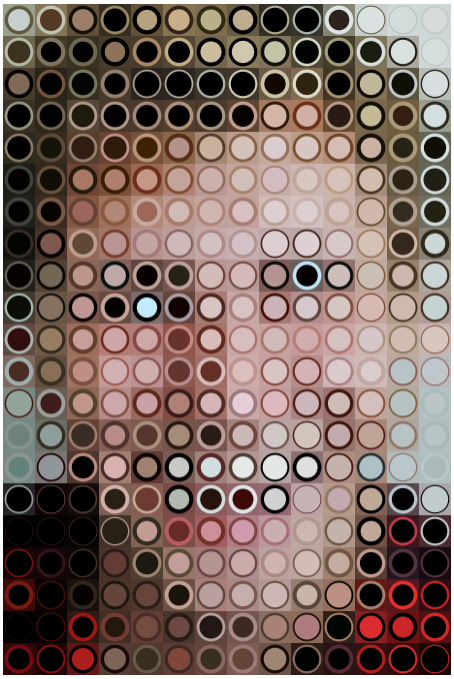
参考文献

[1] グラフィック社編集部（編集）, ピクセル百景—現代ピクセルアートの世界, グラフィック社, Jun. 2019.

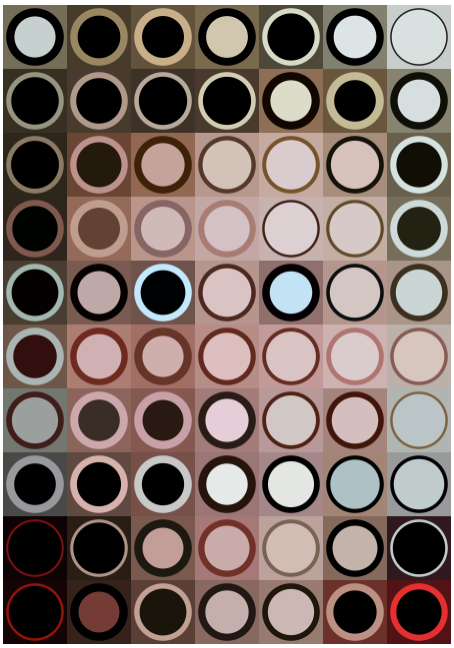
[2] Wikipedia. Chuck Close. `http://en.wikipedia.org/w/index.php?title=Chuck%20Close&oldid=960591386`, 2020. [Online; accessed 22-June-2020]

[3] 坂内正夫, 大沢裕, 曽根光男, 尾上守夫, “画像処理研究用標準画像データベース SIDBA の運用について,” テレビ学技報, vol. 8, no. 38, pp. 7–12, 1984.

[4] De S., Bhowmick P. (2013) Digital Circlism as Algorithmic Art. In: Bebis G. et al. (eds) Advances in Visual Computing. ISVC 2013. Lecture Notes in Computer Science, vol 8033. Springer, Berlin, Heidelberg



(a) $M = 21,\ N = 14$



(b) $M = 10,\ N = 7$

図 9 図 6(b) の丸模様ピクセルアートで h の値を変えた例: (a) $h = 32$、(b) $h = 64$