

アクアビーズのためのデザインシステム

橋本怜実*

五十嵐悠紀*

概要: アクアビーズとは、ビーズを並べて霧吹きで水を吹き付けることによって、ビーズ同士をくっつけてデザインするアートである。本稿では、アクアビーズを対象としたデザインツールを提案する。通常は既存のデザインプレートの上にビーズをのせていくため、ビーズを配置できる箇所が事前に決定している。提案ツールではビーズ同士がくっついているという制約のもと、ビーズの配置をコンピュータ上でデザインすることができ、実際にビーズを制作するためのトレーの3次元モデルを出力する。3次元プリンタで出力することによって、より自由度の高い様々なデザインアクアビーズを作ることが可能となる。

A Design System for Aquabeads

SATOMI HASHIMOTO*

YUKI IGARASHI*

Abstract: Aquabeads are pieces of art designed by sticking beads together through spraying a mist of water on adjacent beads. In this paper, we propose a design tool for Aquabeads. Under the constraint that the beads be stuck together in the proposed tool, we can design the arrangement of the beads on a computer and output a 3D model of the tray that actually produces the beads. By outputting with a 3D printer, various designs of Aquabeads with higher degrees of freedom than previously possible can be produced.

1. はじめに

アクアビーズ[1]は、プレートにビーズを並べていき、ビーズが水で溶けるといった性質を使ってビーズ同士をくっつけるアートである。アイロンビーズ(パーラービーズ)[2]という、アイロンの熱でビーズを溶かし、ビーズ同士をくっつけるものも存在する。これらは子どもが楽しむおしゃれなアート作品として近年注目されている。通常、デザインは図1のような規則正しく並んだプレートを利用して、ビーズを並べていく。しかし、ビーズの制約としては「隣り合うビーズ同士が接している」という条件を満たせば良いため、この制約を満たすようなデザインを行うことで、デザインの幅を広げることができると考えた。

本稿では、ビーズ同士が「接する」という制約の下でデ

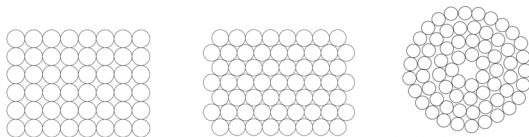


図1 既存のビーズプレートの種類。(左) 格子プレート、(中央) 充填プレート、(右) 回転プレート

Figure 1 Existing types of Aquabeads plates. (left) Grid plate, (center) filling plate, (right) rotating plate.

ザインできるエディタを開発し、オリジナルのビーズプレートの3次元モデルを出力する。これによって3次元プリンタで、デザインに合わせたプレートを作成することが可能になり、実際にユーザが制作することができる。

2. 関連研究

ドット絵に関する研究としては、ドット絵の輪郭線を再現する研究[3]や、ドット絵の制作工程を支援する研究[4]、入力画像をドット絵風に処理する手法[5]などが存在する。また、アクアビーズ対象としたシステム[6]やパーラービーズ用のアプリ[7]も発表されている。しかしこれらは図1に示したような既存のプレートの上でデザインをしていくのみである。我々はより自由度の高いデザインを作るための支援を行う。

五十嵐によるラインストーンを対象としたシステム[8]では、ユーザがストロークを描くと、そのストロークに同サイズのラインストーンを配置していくことで初心者がオリジナルなデザインで作成することを支援している。これはドット絵とは異なり自由度が高いが、ラインストーン同士が接している必要はなく、本提案システムとは異なる。

3. システム概要

提案システムの概要を述べる。ソフトウェアはProcessing-3.2.3で実装した。処理の内容は、デザインとプ

* 明治大学
Meiji University

プレート生成の2つに分けられる。以下にそれぞれの詳細を述べる。

3.1 プレートデザイン

図2にデザインエディタを示す。ビーズ同士が接するという制約の下で、次のビーズをデザインしていく。

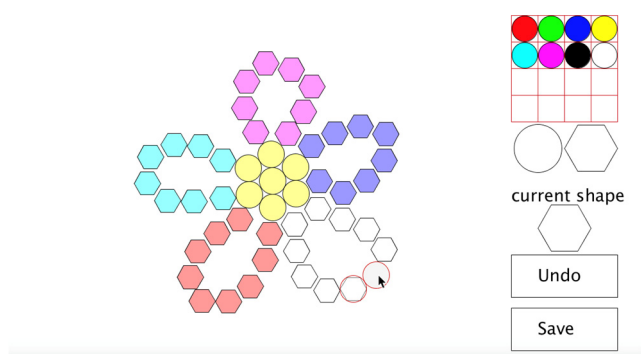


図2 デザインエディタ
Figure 2 Design editor of our system.

1つ目のビーズはプレート内のどこにでも自由に置くことができる。2つ目以降のビーズはすで置かれているビーズに対して接する場所にしか置くことができない。図3のように、まず、すでにデザインしたビーズ n 個の中で、マウス座標 $P(\text{mouse}X, \text{mouse}Y)$ から一番近い距離にあるビーズの中心座標 $P_k(x_k, y_k)$ を見つける。ビーズの半径を r とすると、ビーズとビーズの中心間の距離は $2r$ である必要があるため、 P_k から $2r$ 離れた直線 PP_k 上の点を候補座標としてユーザに提示する。もしこの点がすでに置かれている他のビーズと重なる場合には候補座標を更新しない。マウスをクリックすることで候補座標にビーズが置かれる。一番近いビーズをハイライトすることでユーザのデザインを助ける。

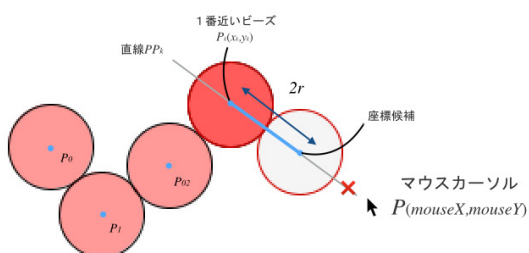


図3 アルゴリズム
Figure 3 Algorithm.

3.2 プレート生成

ビーズの中心座標 (x_k, y_k) ($k=0, \dots, n-1$) のリストを受け

取り、プレートを作成する。既存のビーズプレートは1つの穴の直径が4.5mm、深さが1.5mmであるため、これに対応した穴を点 (x_k, y_k) を中心として n 箇所作成していく。CSG (Constructive Solid Geometry)表現で表し、図4のように、OpenJSCAD [9]を用いて3次元メッシュを作成した後、STLファイルにして3次元プリンタで出力した。

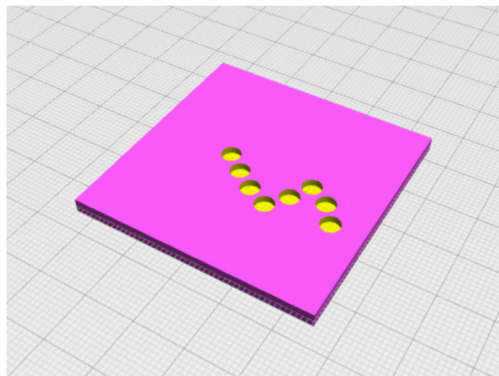


図4 プレートの穴の形状
Figure 4 Shape of a 3D plate.

4. 結果

本システムでユーザにデザインしてもらった結果を図5、図6に示す。既存のデザインとは異なり自由度の高いデザインが得られていることがわかった。

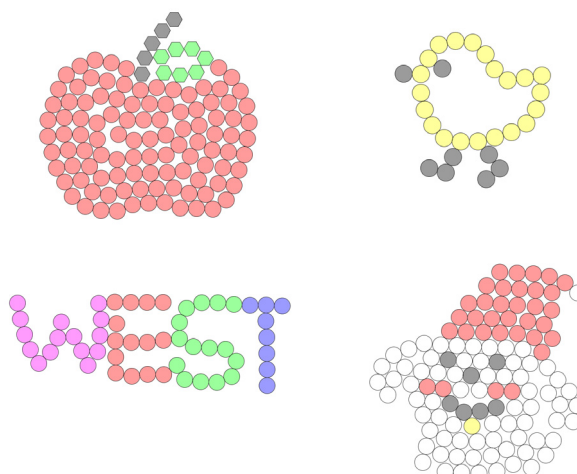


図5 デザイン結果
Figure 5 Results using our system.

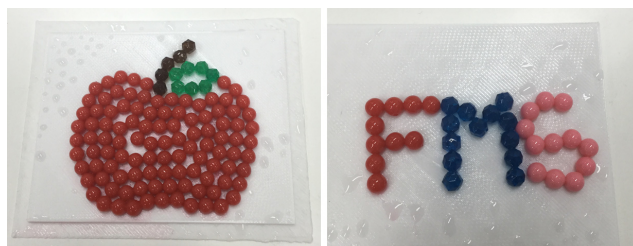


図6 実際に制作した結果
Figure 6 Actual aqua-beads using our system.

5. まとめと今後の課題

本稿ではオリジナルなアクアビーズデザインをするためのシステムを提案した。本システムを用いてビーズ同士が接続しているという制約を満たすようにデザインしていくことで既成のプレートの自由度を超えたデザインを作成可能となった。

今後は図1にあるようなプレートデザインと部分的に組み合わせることを行う。また、1点で固定されているなど、壊れやすい箇所を検出してユーザーに提示する機能を加えたり、壊れにくいようになる場所をシステムが提案したりするなど、より使いやすいシステムへと改善していきたい。

謝辞 本研究を進めるにあたってデザインインタフェースの開発に協力していただいた小林由佳氏に謹んで感謝の意を表する。

参考文献

- [1] エポック社. アクアビーズ. <http://epoch.jp/ty/aquabeads/>
- [2] Perler beads. <http://www.perler.com/>
- [3] Tiffany C. Inglis, Craig S. Kaplan. Pixelating vector line art. SIGGRAPH'12 Posters, 2012.
- [4] 松島立弥, 三上浩司. シームカービング手法を用いた人物ドット絵制作の効率化手法. 情報科学技術フォーラム講演論文集 Vol. 8, No. 3, pp. 293-294, 2009.
- [5] Johannes Kopf, Ariel Shamir, Pieter Peers. Content-adaptive image down-scaling. Transactions on Graphics, 32(6),173, 2013.
- [6] ウォータービーズデザイナー. <http://www.kids-beads.com/>
- [7] Perler. <https://itunes.apple.com/us/app/perler/id574503255?mt=8>
- [8] Yuki Igarashi. "DECO: A Designing Editor for Rhinestone Decoration." IEEE Computer Graphics and Applications, 31(5), 90-94, 2011.
- [9] OpenJSCAD. <http://openjscad.org/>