

Swellart : 制約付き膨張によるスケッチベースのデフォルメデザイン

湯浅 海貴 中山 雅紀 藤代 一成

慶應義塾大学

Swellart: Sketch-based deformed design using constrained swelling

Kaiki Yuasa Masanori Nakayama Issei Fujishiro

Keio University

{kaiki.yuasa, nakayama, fuji}@fj.ics.keio.ac.jp

概要

本研究では、初心者でも直感的にデフォルメされた2Dグラフィックデザインを生成できるシステム—Swellartを提案する。我々は、システムを開発するうえで、変形の方法のなかでも膨張に着目している。ユーザが描画した枠の形状に合わせて対象の画像が膨張する過程を単純に示すことで、すべてのユーザが容易に変形過程やその結果を想像させられる。膨張方法として、まずユーザが画像を指定して、デフォルメしたい範囲をユーザがくり抜き、その範囲内のメッシュ分割を行う。次に、分割された各セルに対し、さまざまな制約をユーザが指定して膨張を行うことで、多様なデフォルメデザインを生成できる。本論文では、Swellartの処理手順と結果を示し、今後の方向性を議論する。

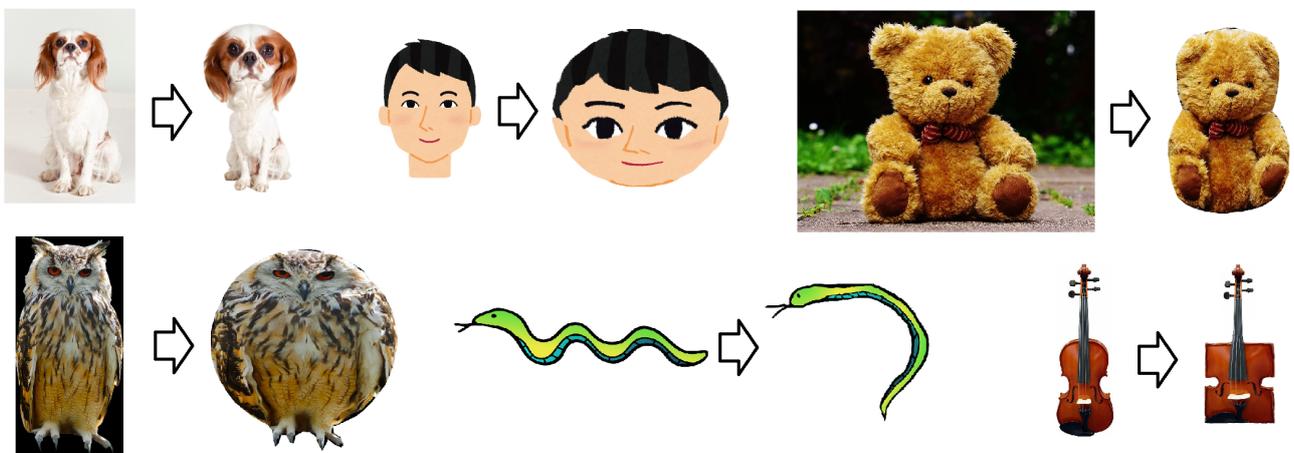


図 1: 提案システム Swellart で作成したデフォルメデザインの例

1 背景と目的

ロゴやアクセサリのデザインなどには、円形や四角形の枠に収まるようにデフォルメされたデザインが数多く存在する。しかしそれらのデフォルメデザインの多くはデザイナーによって描かれており、専門でない人がデザインすることは難しい。デザイナーでなくとも、ある程度絵をうまく描ける人であれば自身で描いた絵をデフォルメすることのできるツールは存在するが、全く絵を描けない人はそれらのツールを使うことも難しい。

そこで本研究では、ユーザが描画した枠内に収まるように、任意の形状のデザインを膨張、変形させることで、図1のようなデフォルメされた2Dグラフィックデザインを誰でも簡単に生成できるシステム—Swellartを提案する。デザイナーでない人や、そもそもデフォルメされていない絵を描くことも難しい人でもデフォルメデザインを生成することを可能にするため、入力画像はユーザが描いた絵に限らず、写真や既にあるイラストなど、好きな画像を指定してデフォルメしたい範囲をユーザがくり抜くことができる。Swellartでは枠による制約の他に、セル単位でより細かい制約を与えることも可能にする。

Swellartのインタフェースを図2に示す。



図2: 提案システム Swellart のインタフェース

本論文は次章以降、次のように構成されている。まず、次章で関連研究について概説する。3章では、提案手法の機能とアルゴリズムについて詳説する。4章では、Swellartを用いてデフォルメデザインを生成した結果を示す。5章ではユーザテストによって、Swellartと似たような機能をもつ既存のシステムであるPhotoshopのワープ機能との比較を行い、提案手法の有用性を実証する。最後に6章で、本論文の結論を述べ、今後の課題に言及する。

2 関連研究

既存のデフォルメシステムとして、まずZouらのシステム[2]が挙げられる。枠の制約の中で2Dグラフィックを変形させる研究で、ひとつの英単語を、読みやすさを保持した状態で配置、変形させることで、単語全体のシルエットが指定した2D形状になったデザインを生成するシステムを提案した。しかしこのシステムでは、外枠はあらかじめ用意されたもので、変形させる対象も文字に限られている。Xuらの研究[3]では、システムで用意されている3Dモデルを、ユーザの指定した画像上のオブジェクトの形状に合わせて変形させ、デフォルメデザインの3Dモデルを生成できる。しかし、使用する3Dモデルは他のインタフェースを用いて作成が必要で、モデリングに不慣れなユーザが扱うのは難しい。直感的な操作によって大幅なデフォルメを行う研究として八坂らのシステム[7]が挙げられる。八坂らはクロス&グラフトによって2つのオブジェクトから両者の特徴を併せもった新たな世代のオブジェクトを生成するシステムを提案した。このシステムはドラッグ&ドロップするだけの直感的な操作で大幅なデフォルメが可能である。

次に、スケッチベースのモデリングインタフェースとして、五十嵐らのシステム—Teddy[6]が挙げられる。絵を描くように3Dモデルを生成できるため、モデリングに不慣れなユーザでも簡単にモデリングを行うことができる。しかし、直感的に操作はできる反面、生成できるモデルも比較的シンプルなものに限られる。また、描いた後にデフォルメを行う機能は充実していない。他にも、ユーザが3D空間上に描いた線すべてを通るように表面を生成する研究としてGrimmandらのシステム—Just DrawIt[1]が挙げられる。Teddyに比べ、より複雑な形状のモデリングが可能であるが、描画方法が直感的でなく、絵を描くのに不慣れなユーザには扱いが難しい。

以上の関連研究をふまえて、本研究では、画像を読み込んで必要な範囲だけをくり抜くことで、絵を描くのが苦手なユーザでも直感的にシステムの利用を可能にする。また、ユーザが外枠を自由に描画し、変形させる対象も写真やイラストなど自由に指定する。さらに、その画像の任意領域のくり抜きができ、より自由度の高いデフォルメを可能にする。変形の方法は膨張を中核におき、直感的な操作による大幅なデフォルメの実現を目指す。

3 Swellart の手法

まず Swellart では、使用する画像 (図 3(a)) を読み込み、くり抜きたい領域を描画する (図 3(b))。指定された領域内を図 3(c) のように三角メッシュ分割し、図 3(d) のように不要な部分を消し、くり抜いた領域だけを表示する。その後、膨張範囲を制御するための外枠を図 3(e) のように描画し、膨張させることで図 3(f) のようなデフォルメされたデザインを生成する。

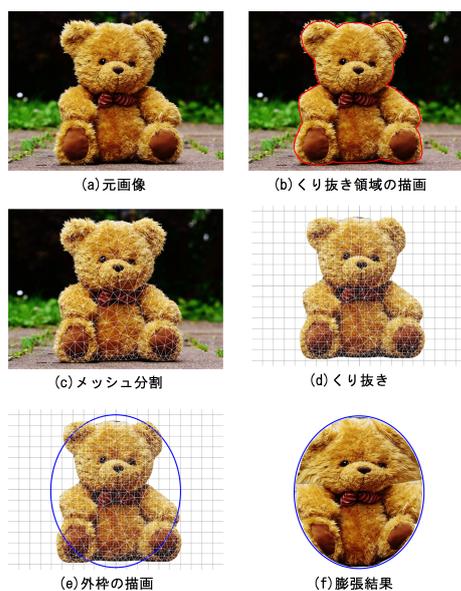


図 3: Swellart の処理の流れ

また、膨張させる際に、外枠以外に以下の制約を与えることができる。

- セル単位の膨張率指定
- セル単位の形状固定
- セルの格子点の位置固定・移動

まず膨張処理の流れを示したのち、くり抜きと膨張や外枠の描画、上記の制約機能について説明する。

3.1 膨張処理の流れ

膨張が実際にプログラムでどう処理されているかを以下に示す。

まず、セルの各頂点の座標 p_0 と、各セルの重心 g を用いて、式 (1) によりセルの各頂点の、重心に対する位置ベクトル v_0 を求める。

$$v_0 = p_0 - g \quad (1)$$

画像全体の膨張率 s_a と、部分指定した各セルの膨張率 s_x を用いて、式 (2) により膨張後の理想のセルの各頂点の、重心に対する位置ベクトル v_1 を求める。

$$v_1 = v_0 \cdot (1 + ((s_a - 1) \cdot s_x \cdot 0.99)) \quad (2)$$

形状マッチング法によって導かれる最適な回転行列 M を用いて、式 (3) により膨張後の理想のセルの各頂点の座標 p_1 を求める。

$$p_1 = M \cdot v_1 + g \quad (3)$$

外枠によって生じるベクトル場の力を b としたとき、式 (4) によりセルの各頂点に与える力 f_p を求める。

$$f_p = b + (p_1 - p_0) \quad (4)$$

位置固定されていれば 0、されていなければ 1 となる変数 a を用いて、式 (5) により p_0 を移動させる。

$$p_0 = p_0 + f_p \cdot a \quad (5)$$

最後に、食い込み防止の処理を行う。膨張率が変更された場合、また式 (1) から順に処理を繰り返す。以上の流れを図 4 に示す。

3.2 くり抜きと膨張

画像のくり抜いた領域を三角形メッシュに分割し、セルの各頂点の座標と各セルの重心から、3.1 節の式 (1) によってセルの各頂点のベクトルを求める。この際、変形した際に輪郭の形状を滑らかに保つため、輪郭上にあるセルの辺長を他の部分の半分に設定し、より細かく分割している。また、分割するセルサイズは図 5 のように 50% から 200% まで、ユーザが指定可能であり、対象のくり抜き画像の大きさや処理速度との兼ね合いでセルサイズを調整可能である。



図 5: セルサイズの変更例

メッシュ分割の方法としては、ある程度の品質の効果的なメッシュ生成を行える手法のなかでもっとも単純なアルゴリズムであり、それゆえに Swellart に合わせて扱

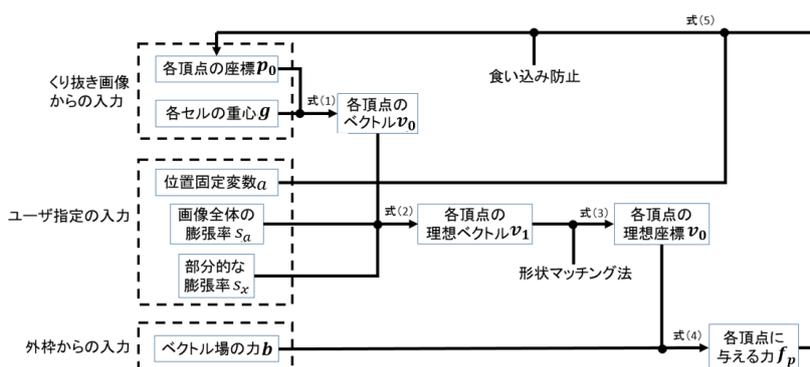


図 4: 膨張処理の流れ

張を行いやすい Chew の手法 [4] を採用している. Chew の手法をもとに, まずくり抜いた輪郭上にユーザ指定のセルサイズに応じた等間隔の点を発生させ, ドロネー図を作成する. その後, ユーザが指定したセルサイズの閾値以上に半径が大きい, ドロネー図を構成するすべての三角形の外接円を探し, その外心に点を追加してドロネー図を更新する (図 6). そして閾値以上の外接円がなくなるまでこれを繰り返すことでメッシュ分割を行う. 輪郭上の閾値だけ他の部分の半分にする事で, 輪郭上のセルを細かく分割している.

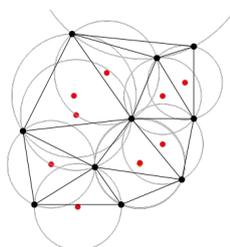


図 6: メッシュ分割

くり抜かれた画像を, 画像全体の膨張率と各セルの膨張率で 3.1 節の式 (2) によって拡大したものを膨張後の理想の画像とし, メッシュの各格子点に対して, 元の画像と理想の画像の座標の差分だけ力を加えて移動させる. 膨張させる倍率はユーザが指定することができ, 50% から 200% まで対話的に変更可能である.

外枠の制約などによって膨張の際にメッシュの形状が歪んだ場合, 形状マッチング法 [5] を用いて 3.1 節の式 (3) のように各セルを元の形状に近づくように最小の頂点移動で変形させることで, 元の画像のパーツの位置関係を保持したデフォルメが可能である.

また, 一つの画像から複数くり抜くこともでき (図 7(a)), 図 7(b) のように衝突判定の考慮された膨張を行うことができる. 複数のくり抜き画像同士, もしくは自身への食い込み防止については後述する.

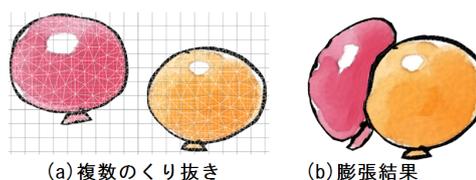


図 7: 複数のくり抜き画像の膨張結果

3.3 食い込み防止

画像が枠内で膨張する際に, 複数のくり抜き画像同士, もしくは自身への食い込み防止を行う (図 8). まず, 輪郭上は 0, 内部は 1 の値を各セルの頂点に与え, ポテンシャル分布をメッシュ内に生成する. そして, 内部に食い込んだ頂点の存在する三角面を特定し, 三角面内のポテンシャルの下り勾配方向に押し戻す. このとき, 押し戻す力は三角面に食い込んだ深さによって決定する.

食い込んだ三角面に輪郭上の頂点が 2 つある場合には, 輪郭上の 2 点を結ぶ線分の単位法線ベクトルを \mathbf{n}_a , 線分との距離を d_a として, 以下の式で処理を行う.

$$\mathbf{p}_0 = \mathbf{p}_0 + \mathbf{n}_a \cdot d_a$$

食い込んだ三角面に輪郭上の頂点が 1 つだけの場合には, 輪郭上でない 2 点を結ぶ線分の単位法線ベクトルを \mathbf{n}_b , 線分と食い込んだ点の距離を d_b , 線分と輪郭上の頂点の距離を d_x として, 以下の式で処理を行う.

$$\mathbf{p}_0 = \mathbf{p}_0 + \mathbf{n}_b \cdot (d_x - d_b)$$

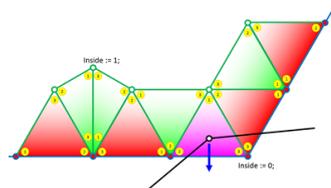


図 8: 食い込み防止

3.4 外枠の描画

外枠の描画方法は自由描画のほかに、円描画と四角描画も可能であり、自由度の高いデフォルメから基本形状のデフォルメまで、幅広いデザインを生成できる。

処理としては、画像をくり抜いた後、背景の格子を目安に、画像を膨張させたい範囲を描画する。外枠を描画した際に、画面上に正方形格子で細かく点を取り、それぞれの点が枠の内側ならば 1、外側ならば 0、内側と外側の境界部分ならば 0~1 のグラデーションになるようにポテンシャル分布を生成する。このとき、境界部分にグラデーションがかかるようにするために、各点の近傍 10 点をランダムにサンプリングし、外枠の内側と外側にそれぞれ存在する点の個数の割合でポテンシャルを決定する。このポテンシャル分布を使用して、値の大きな個所にくり抜かれた画像が引き寄せられるようなベクトル場を生成する。膨張させるために各格子点に対して力を加える際に、3.1 節の式 (4) のように理想の座標と現在の座標の差分を加えると同時にベクトル場の力を一緒に加えることで、画像を外枠の中に収まるように動きを制御し、変形させる。

Swellart における外枠は、くり抜いた画像を膨張させる際に外枠に衝突したら押し返すのではなく、くり抜いた画像を枠内に引き寄せるベクトル場のため、図 9(a) のように画像に重なった外枠や画像より小さい外枠を描画しても、図 9(b) のように変形させることができる。

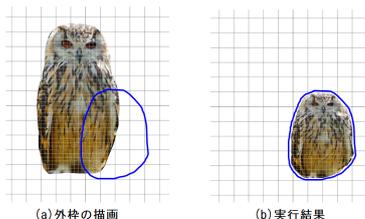


図 9: 外枠を利用した膨張結果

3.5 セル単位の膨張率指定

膨張率を部分的に変更したい部位があるとき、セル単位で変更できる。セルの各頂点に対し力を加える際に、そのセルの膨張率を乗ずることで頂点の移動を制限し、部位によって膨張率を指定できる。

セル単位の膨張率指定機能を利用した場合のデフォルメデザインの生成手順を図 10 に示す。まず、図 10(a) のような画像をくり抜き三角形メッシュ分割を行う。次に、図 10(b) のように、顔の中央に近づくにつれ膨張率を高く、輪郭に近づくにつれ膨張率を低く設定する。膨張率が 100% 以上のセルは赤く、膨張率が 100% 未満のセルは青く表示している。膨張率を部分指定したら図 10(c) のように外枠を描画し、膨張させる。セルの境界線を非表示にした生成結果が図 10(d) である。

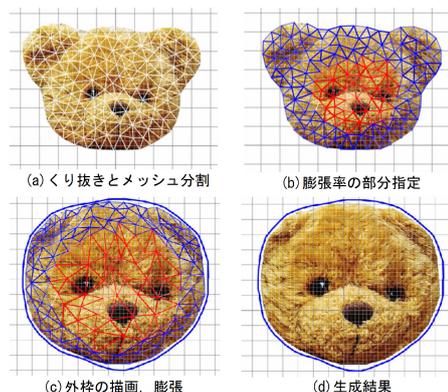


図 10: セル単位の膨張率指定を利用したデフォルメデザインの生成手順

セル単位の膨張率指定機能を利用することで、図 11(a) のようなくり抜き画像を魚眼レンズで写真を撮ったようなデザインにデフォルメが可能である。膨張率の部分指定を行ったとき図 11(c) のほうが、膨張率の部分指定を行っていない図 11(b) に比べて顔の中央部分が膨らみ、効果が表れていることが明白である。



図 11: セル単位の膨張率指定の有無による比較

3.6 セル単位の形状固定

画像のデフォルメを行う際に元の形状のまま保持したい部位があるとき、セル単位で指定でき、変形させずに形状を固定できる。

形状固定機能を利用した場合のデフォルメデザインの生成手順を図 12 に示す。まず、図 12(a) のような画像をくり抜き三角形メッシュ分割を行う。次に、図 12(b) のように形状固定したい領域をなぞって設定する。形状固定した部分のセルは黒く表示している。形状固定する部分を指定したら図 12(c) のように膨張させる。セルの境界線を非表示にした生成結果が図 12(d) である。

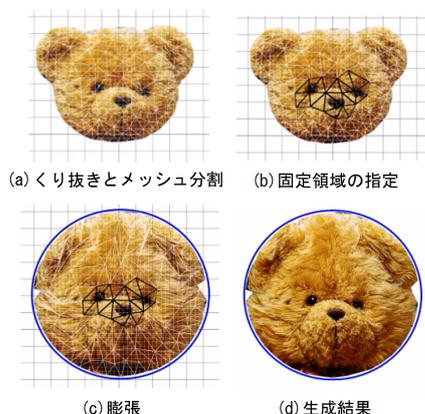


図 12: 形状固定機能を利用したデフォルメデザインの生成手順

形状固定機能を利用することで、図 13(a) のようなくり抜き画像の目と鼻の部分を形状固定して、他の部分だけ膨張させることが可能である。形状固定をしていないときは図 13(b) のように画像全体が膨張してしまうのに比べ、形状固定した場合は図 13(c) のように目と鼻以外だけが膨張しているのがわかる。



図 13: 形状固定の有無による比較

3.7 セルの格子点の位置固定・移動

位置固定機能を利用することで、セルの格子点単位で画面上に位置を固定することができる。上記の形状固

定では画面上での位置までは固定されないため、状況によってこの二つの機能の使い分けが重要となる。3.1 節の式 (5) の通り、固定されていると頂点を動かす力をすべて無効化するため、形状固定した部分は外枠の制約を受けるが、位置固定をした部分は外枠の制約に縛られない。これにより、図 14(a) のように耳の部分だけを固定し、他の部分を固定しなかった場合、図 14(b) のように固定していない部分だけを外枠の形状に合わせて変形できる。

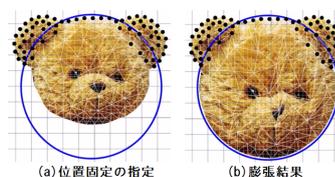


図 14: 位置固定機能と外枠描画の併用例

また、膨張を行った後に少し絵を修正したいとき、セルの格子点それぞれに対してドラッグ&ドロップによって位置移動をすることができる。図 14 で生成した画像を、図 15(b) のように位置移動機能を利用して修正した結果、図 15(c) の結果が得られる。

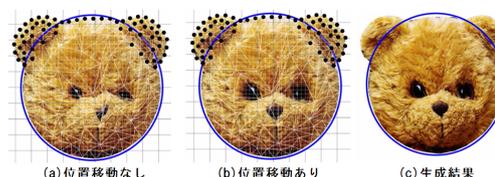


図 15: 位置移動機能の使用例

4 結果

4.1 開発環境

開発環境として CPU: Intel Core i5-3230M 2.60GHz, RAM: 4.00GB を、プログラミング言語として Delphi を使用した。

4.2 実行結果

本手法の実行例として、図 16(a) のくり抜き画像から様々なデフォルメデザインを生成した。図 16(b) では 3.5 節の機能により顔の中心部分の膨張率を高く指定し、図 16(c) では四角い外枠を描画した。また、図 16(d) では 3.7 節の固定機能を用いて耳以外の部分を固定し、図

16(e) ではさらに耳の部分だけに外枠を利用して変形を行った。画像のくり抜きや外枠の描画など、多くの部分がユーザによって自由に指定できるため、同一の画像からでも多様なデザインが生成できた。

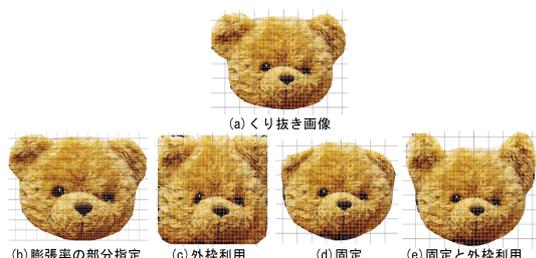


図 16: Swellart で生成したデフォルメデザインの例

この他にも、図 1 のように、写真やイラストなどのさまざまな画像から、多様なデフォルメデザインを生成できた。

また、世の中には図 17 のような、ある形状の枠にはめながら育てることでその枠の形状になる、変わった形状のスイカが存在する。このスイカを、Swellart を用いて再現した。元画像に 図 18(a) のような通常の丸いスイカの画像を用いて、図 18(b) のような丸形、四角形、ハート形の 3 種類のデザインを生成した。図 17 と比べると、模様の流れなどかなり再現率が高くなっていることがわかる。



図 17: 変わった形状のスイカの例

5 ユーザテスト

Swellart の有用性を実証するために、Swellart と似たような目的をもつ既存のシステムである Photoshop のワープ機能との比較を行った。方法としては、デザインを専門としない 20 人のユーザにそれぞれのシステムの操作方法を説明して理解してもらった後に、それぞれのシステムで図 3(a) の元画像から顔部分をくり抜き、図 13(b) のような円形のデフォルメデザインを生成してもらった。また、使用するツールの順番によって結果に偏

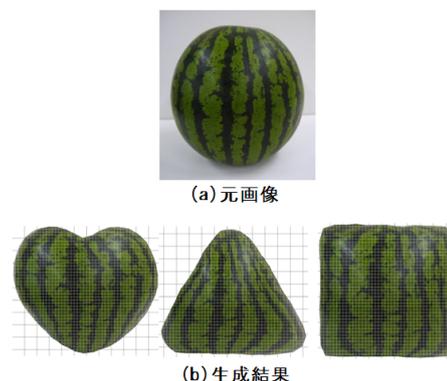


図 18: Swellart による変わった形状のスイカ画像の生成結果

りが出ることがないように、10 人ずつの 2 グループに分け、2 つのツールを利用する順序を変えてユーザテストを行った。

システムが直感的に利用可能かどうかを定量的に比較する方法として、脳波や視線の総移動量などから心理的負担を定量化する方法や、同じデザインを生成するまでの時間を比較する方法などが考えられるが、今回のユーザテストでは、簡便に定量比較できる、生成するまでの時間の比較によって評価を行った。また、Photoshop の熟練度も 3 段階で自己申告してもらった（1：使ったことがない、2：たまに使う、3：よく使う）。

その結果を表 1 に示す。Photoshop の熟練度に関わらず Swellart での平均生成時間は Photoshop に比べ半分以下となっており、全員の平均では約 4.4 倍の差が生じている。この結果より、Swellart は Photoshop よりも直感的に操作可能であり、有用性があるといえる。

表 1: 生成時間の比較 (分:秒)

	Swellart			Photoshop		
	最大	最小	平均	最大	最小	平均
Photoshop 熟練度 1	1:59	0:34	1:06	11:18	3:14	5:51
Photoshop 熟練度 2	4:08	0:37	1:51	11:10	1:38	5:32
Photoshop 熟練度 3	1:17	0:41	0:59	2:52	1:19	2:05
全員	4:08	0:34	1:14	11:18	1:19	5:24

また、実験参加者 20 人に対して、以下の 3 つの項目についての 5 段階のアンケート調査も行い、その結果を図 19 に示す。どの項目においても Photoshop のほうが良

いと答えている人はいないため、Swellart のほうが直感的に操作可能で、有用性があるといえる。

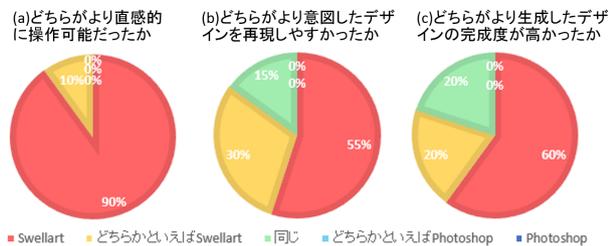


図 19: アンケート結果

6 結論と今後の課題

本論文では、直感的な操作により誰でも簡単にデフォルメされたデザインを生成できるシステムを提案した。Swellart により、写真やイラストなどの様々な画像から直感的な操作だけで大幅なデフォルメができることを確認した。

課題として以下の 4 つの事柄が挙げられる。

まず最初に、外枠の描画を手描きにすることで自由度の高いデフォルメが可能となっているが、実際に何かにプリントするためのデザインを生成しようとした場合、その対象の正確な形状を外枠として描画する必要がある。次に取り組むべき課題として、プリントする対象となる物体の写真を撮り、それを Swellart に読み込むことでその対象の形を認識し、外枠として設定することができれば、より実用的なデフォルメが可能になる。

2 つ目に、Swellart でデフォルメを行った場合、外枠の形状によっては元の画像の対称性が保持されなくなってしまう点が挙げられる。対称性を保持したい場合に選択できるようにしておくことで、よりユーザの意図に沿ったデザインが可能になる。

3 つ目に、外枠の摩擦力の指定が挙げられる。外枠の摩擦力を指定できれば、くり抜き画像が外枠に当たった瞬間動きを止め、元の形状をできる限り保ったデフォルメができる。また、逆に摩擦をなくし、画像が外枠上を滑りながら折れ曲がり、より細かい隙間まで埋めることもできる。

最後に、3D での実装も課題として残っている。2D ではロゴのデザインや、アクセサリや食器などにプリントするデザインの生成に使用できるが、3D で実装すること

ができればブローチやお菓子などの、立体的にデフォルメされたデザインやファブリケーションも可能になる。

参考文献

- [1] C. Grimmand and P. Joshi: “Just DrawIt: A 3D Sketching System,” *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, pp. 121-130, June 2012.
- [2] C. Zou, J. Cao, W. Ranaweera, I. Alhashim, P. Tan, A. Sheffer, and H. Zhang: “Legible Compact Calligrams,” *ACM Transactions on Graphics*, Volume 35, Issue 4, Number 122, July 2016.
- [3] K. Xu, H. Zheng, H. Zhang, D. Cohen-Or, L. Liu, and Y. Xiong: “Photo-Inspired Model-Driven 3D Object Modeling,” *ACM Transactions on Graphics*, Volume 30, Issue 4, pp. 80-80, August 2011.
- [4] L. P. Chew: “Guaranteed-Quality Triangular Meshes,” *Department of Computer Science Tech Report*, Cornell University, No. TR-89-983, April 1989.
- [5] M. Muller, B. Heidelberger, M. Teschner, and M. Gross: “Meshless Deformations Based on Shape Matching,” *ACM Transactions on Computer Graphics*, Volume 24, Issue 3, pp. 471-478, July 2005.
- [6] T. Igarashi, S. Matsuoka, and H. Tanaka: “Teddy: A Sketching Interface for 3D Freeform Design,” *SIGGRAPH '99 Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 409-416, July 1999.
- [7] 八坂 俊, 藤代 一成, “クロス&グラフトに基づく 2D モデリングインタフェース”, 情報処理学会研究報告, Volume 2012-CG, Issue 146, Article No. 30, 2012 年 2 月.