

# 3D プリンタのための Paper User Interface

秋山 耀\* 宮下 芳明†

**概要.** 紙を用いたインタフェース (PUI: Paper User Interface) は、手軽さやわかりやすさの優位性から様々なシステムへの導入が試みられており、特に高齢者ユーザへの効果が期待されている。本稿では、3D プリンタでの出力を前提として、PUI に則った 3D モデル入力を提案し、システムとして実装した。専用紙にマーカで実寸の二面図を描き、角丸仕上げを望むならチェックボックスをチェックする。この紙をセットしてボタンを押すだけで、スキャン→形状推測→モデル生成→3D プリントまでが自動で完了する。本稿では小学生や 80 代の実験参加者でも有用性を確認するとともに、その制約について議論を行っている。

## 1 はじめに

紙をインタフェースとして用いる手法は、「paper user interface」(PUI) と呼ばれ、ディスプレイ上に表示されたインタフェース、「screen-based interface」に相対する概念である [1]。文献 [1] ではチェックボックスや文字認識を用いた PUI システムが提案されている。例えば、書類を送信する際には行いたい操作のチェックボックスにマークし、カバーシートに送信先などの情報を書いて一緒にスキャンするだけで、システムが自動認識して送信操作を行う。

「紙はとても軽く、フレキシブルで、電源が必要なく、そして書いたものがなくなる安心感がある。人びとは紙に慣れていて、しかもそれをどう使うかは誰でも知っている。」 [1] そのため、PUI の応用性は高く、高齢者を対象としたシステムにも使用できる可能性がある [2]。Zhao らは、紙を用いたインタフェースについて「コンピュータ操作に慣れていないユーザ、たとえば高齢者などにとってわかりやすく使いやすくと考えられる」と述べている [3]。最近でもアニメーション制作のワークフローの中に紙を用いる手法が提案されている [4] ほか、紙を用いたインタフェースはロボット制御 [3] や音楽制作システム [5] への導入が試みられている。また、植物の絵を紙に描くことで 3 次元的なモデルを手軽に生成できるようにした例もある [6]。本稿では、3D プリンタでの出力を前提として、PUI に則った 3 次元モデル入力を提案するが、上述の研究と同様に、手軽さやわかりやすさで大きな恩恵をもたらすと考えている。

紙でのモデリング作業や図面を描く作業は、小型のものなら実寸で行えるという利点がある。実世界のものに合わせて使いたい物はコネクタや入れ物など多く存在する。そして大きさ合わせについての研究も多い [7, 8, 9, 10]。サイズ感がわかりやすいとい

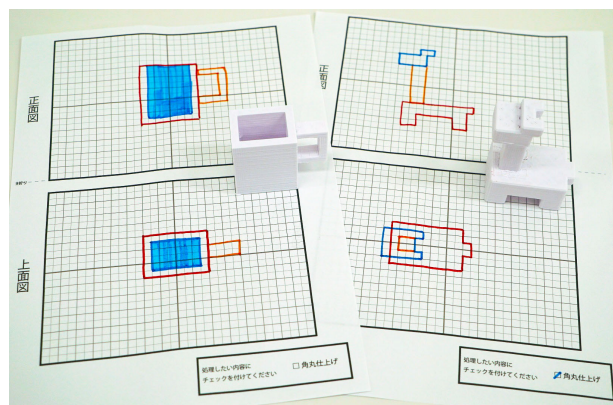


図 1. 手描きで作成した図面とそれによって得られた造形物

う点では、ピンチなどの拡縮操作ができない PUI の方がむしろ適している可能性がある。

我々が提案するのは PUI によってオブジェクトを出力できる 3D プリンタである。ユーザが行う操作はシステム専用紙にマーカで実寸の二面図を描き (図 1)、それをシステムにセットしてスキャンボタンを押すことのみである。システムは 3D 構造を解析し、3D モデルを作成、そのまま造形処理に移る。

本稿の寄与は、PUI で 3D プリントできるシステムの提案であり、PUI での 3D モデリングワークフローに新規性があると考えている。本稿ではまず第 2 章でシステムのデザインとそれに基づいたシステムを述べる。第 3 章では、PUI での 3D モデリングの有効性を検証する。そして第 4 章に作例、第 5 章に実装、第 6 章に議論を述べる。そして第 7 章で関連研究を挙げ、第 8 章で展望を述べる。

## 2 二面図モデリング

提案システムの対象ユーザは、高齢者や小学生などを含む PC を用いた 3D モデリングが困難な人、もしくは PC 自体を操作できない人である。そのた

Copyright is held by the author(s).

\* 明治大学大学院理工学研究科新領域創造専攻

† 明治大学総合数理学部先端メディアサイエンス学科

め、システムを実装する上でいくつかの設計方針を立てた。まず、ボタン操作のステップ数が多く必要な設計は、高齢者に適していない [11]。これより、3D プリンタや紙のスキャナなど電子機器を操作する回数や操作するボタンの数はできるだけ少なくする。更に、3D モデルのすべての情報は紙とペンによって与える。情報を A4 の紙 1 枚に集約させることでユーザの記憶の要求を減少させることができる。

本稿では誰でも行ったことがある「紙に図面を描く」という動作に注目した。そしてその中で一般的な方法である、上から見た図や横から見た図を描きとる方法を用いた。最低二面からの図面があれば 3 次元化が可能である。三面からの図面がなければ表現できない物体もあるが、三面を用いても表現しきれない物体が存在する。つまり面を増やせば増やすほど良いというわけではない。今回は描きとる手間も考え二面図を採用することとした。

システムで使用する専用紙には 5mm 間隔の方眼が書いてある枠が 2 つある。方眼は後述の画像処理の際に認識されないよう、なるべく薄い灰色で印刷する。上に正面図を描き、下に上面図を描く。描いた大きさと同じ大きさの造形物を得ることができる。そのため、図面に物を置いて大きさ合わせすることが可能である。方眼は縦 12cm 横 16cm であり、枠を除くと縦 11cm 横 15cm 高さ 15cm までの物を設計できる。二面図で作成できるのは CSG ソリッドモデルである。(ただし、木構造を持っているわけではないので厳密には違う。) 二面図では和集合と差集合のブーリアン演算を指定することができる。正面図と上面図の下に 1 つ枠があり、そこで作成するオブジェクトに対してオプションを設定することが可能である。今回はモデルの角丸仕上げを可能とした。

二面図を PC に読み込ませる必要があるため、図面作成に 3 つの条件を設けている (図 2)。まず、方眼に沿って水性マーカーで図面を描く必要がある。今回は画像処理を簡単に行うため、方眼上にもみ線が描かれるように制限した。具体的には、曲線や斜めの線を描くことを禁止した。システムはフリーハンドの曲がった線でも、3D モデル化する際に綺麗な直線となるような補正や角度を揃える補正をかける。なお、オプションとして角丸仕上げを用意することで擬似的に曲線的な仕上げを可能とした。また、マーカーはほかのペン類に比べ太く、色がはっきり出る。この特長により画像処理を行う際の誤認識が減ると考えたため、マーカーを用いることとした。使用できるのは赤、橙、黄緑、水色の 4 色である。

次に、違うパーツは違う色で描く必要がある。本稿のシステムでは、ユーザは 1 つのパーツごとに色分けを行う必要がある。また、同じ色は 2 回以上使用不可とした。これはシステム側に正面図と上面図のパーツの対応関係を示す必要があるためである。

最後に、くり抜く部分 (差集合をとる部分) は塗

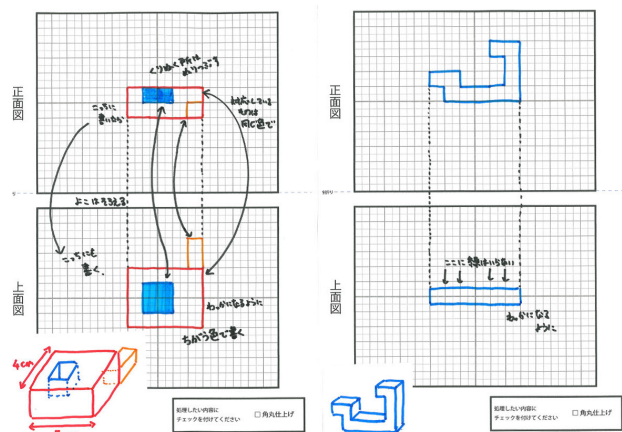


図 2. 図面の描き方の説明用資料

りつぶす必要がある。例えば、コップのように穴の空いた形をモデリングしたい場合は、外側の直方体から内側の直方体をくり抜くようにモデリングする。このとき、どの物体からどの物体をくり抜くか指定する必要がある。今回はくり抜く方の物体を塗りつぶすことで描き分けを可能とした。

### 3 PUI を用いたモデリングの有効性検証

本稿では、screen-based interface でモデリングできないユーザでも、PUI を用いることで可能になるという仮説を立てた。この章ではその仮説を検証する。CAD ソフトの 123D Design<sup>1</sup> を見せ、「使用できない、したくない」と回答した 49~81 歳の男性 1 名女性 2 名を対象に二面図を描く実験を行った。そのうち 1 名は PC の使用経験がないユーザである。

まず、システムについての簡単な説明と、サンプルを元に二面図の描き方の説明を行う。その後、厚紙方眼紙で作成したコップ (図 3 中央) を見せ、それを実寸で二面図に写しとらせた。専用紙と鉛筆、消しゴム、定規、マーカーを渡し、下書きを行ってもよいが必ず最後にはマーカーで清書するように指示した。また、定規の使用は任意だが、なるべく方眼からはみ出さないように描くように指示をした。ほかにも、大きさの写しとりで間違いがあった場合でもそのまま続けるように指示したが、描き方に致命的なミスがあった場合は描き直しをさせた。くり抜く部分を塗りつぶすか (空洞塗り)、それともくり抜かない部分を塗りつぶすか (本体塗り) の 2 通りを記述してもらった (図 3)。空洞塗りと本体塗りを 1 セットとして練習 2 セット、本番 1 セットを行った。参加者によって空洞塗りと本体塗りどちらから始めるかを変えて実験を行った。練習中は描き方にミスがあった場合は理由を添えて指摘を行ったが、本番中は指摘を行わない。実験後に所感を聞いた。

<sup>1</sup> <http://www.123dapp.com/design/>

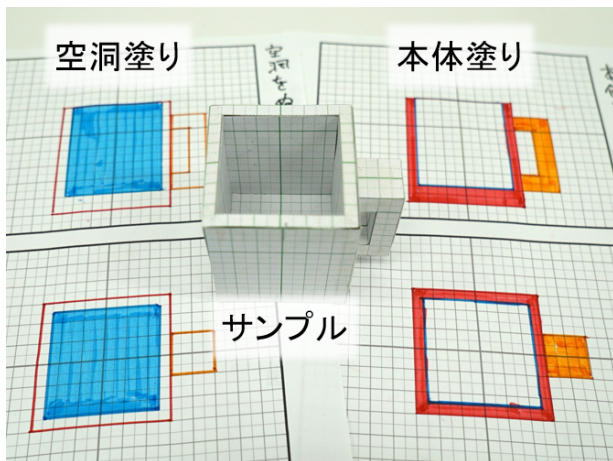


図 3. 空洞塗りと本体塗り

表 1. 空洞塗りと本体塗りの所要時間

参加者	空洞塗り (秒)	本体塗り (秒)
A (49 歳女性)	145	126
B (76 歳女性)	390	534
C (81 歳男性)	475	668

実験結果を表 1 に示す。参加者 A は空洞塗りの方が時間がかかったが、「空洞塗りの方が分かりやすかった」と回答している。参加者 B, C は「空洞塗りと本体塗り、どちらもわかりやすさは変わらない。塗ること自体が面倒臭い」と回答していたが、空洞塗りの方が速く描けた。よって今回は空洞塗りを採用することとした。

すべての参加者において、最初は図面を描く際の制約を守らず記述していた。例えばすべての線を同じ色で描くミスが多く見られた。しかし、図面を 3 回描いたあたりで描き方に慣れ、正確に記述することができた。大きさを間違えて描いた参加者もいたが、自分で間違いに気づき、修正を図る動作が見られた。最初は図面の描き間違いがあるとはいえ、その後正しい図面を描くことに成功している。これらより、前述した仮説は正しかったと考える。

#### 4 ワークショップと作例

9~81 歳の男女 7 名にワークショップを行い、いくつか図面の設計をしてもらった。説明には図 2 を紙に印刷したものを用いた。なお、造形に時間がかかるため、ワークショップ中は図面を描くことのみを行い、後日 3D プリントしたものを渡した。それらのうちのいくつかを写したものを図 4 に示す。筆者らがサポートを行いながら小学 3 年生と 5 年生の男女 2 人に図面を描いてもらったところ、うさぎのおもちゃ (i)、3 つの入れ物 (c)、ゲームキャラク

タを模した物、そしてコップを設計することができた。いくつか方眼から極端にはみ出して描いていた図面があったため、修正テープで適宜修正を行った。うさぎのおもちゃに関しては、顔と胴体をわけて図面を描き、2 つの造形物を組み合わせる形で実現していた。胴体は読みとりに失敗したが頭は造形に成功した。3 つの入れ物のうちの 1 つは造形に成功した。失敗した 2 つのうち 1 つは図面の枠にパーツが重なって描かれていた。また、もう 1 つは線が方眼からずれ過ぎていた。ゲームキャラクタを模したおもちゃは後述のカメのおもちゃを参考に設計していた。しかし、これとコップは正面図と上面図の対応が合っていなかったため読みとりに失敗した。

そのほか、20 代の参加者はトラックのおもちゃ (g) を設計した。別の 20 代の参加者は後述のキリンのおもちゃを参考にウマのおもちゃ (h) を設計した。これらは造形に成功している。また、40 代の参加者はリモコン入れやコースター (d)、まな板立て、ペットボトルを乾かすための道具 (e) の図面を設計していた。リモコン入れを設計した際は、専用紙にリモコンを当てて大きさを設定していた。まな板立ては、線が方眼からずれ過ぎていたため読みとりに失敗した。70 代、80 代の参加者はそれぞれ家の長押 (なげし) の大きさに合った S 字フック (a) とブックエンド (f) を設計した。それぞれ造形に成功している。また、著者らもいくつかのオブジェクトを設計した。3D プリンタのフィラメントを保持するアーム (b) や、キリン (図 1) やカメ、カエルのおもちゃ (g) を設計し、ほぼすべて造形に成功している。アームは掛けるラックの大きさに合った形で出力できた。カメはシステムがストロークを適切に読みとらず、甲羅が少し浮き上がったが、ほとんどは提示したとおりに造形することができた。

上記に述べた物以外も含め計 17 枚の図面を設計、そのうち 11 枚がシステムでの造形に成功した。失敗した 6 枚のうち 3 枚が図面の対応ミスである。

#### 5 実装

システムは大きく 2 つのプログラムによって形成されている。まずスキャンした画像からストロークを抽出するプログラムである。そして抽出したストロークから 3D モデルを生成・プリントするプログラムである。システムは大半が Processing を用いて記述されており、一部 Python を用いている。

##### 5.1 二面図のスキャン

スキャンにはフラットベッドスキャナである GT-S650 と、シートフィードスキャナである ScanSnap S1500 を用いた。これらはスキャンボタンを押すだけで特定のフォルダに jpg ファイルを書き出すことが可能である。システムはそのフォルダを監視している。jpg ファイルが追加されたことを確認すると、

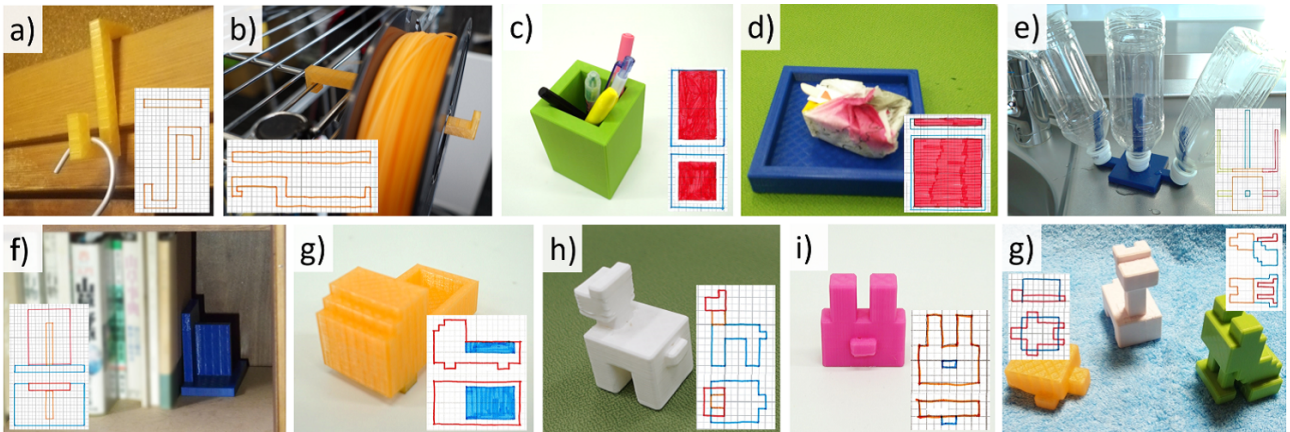


図 4. 作例: a) S 字フック; b) 3D プリント用アーム; c) 鉛筆たて; d) コースター; e) ペットボトル立て; f) ブックエンド; g) トラックのおもちゃ; h) ウマのおもちゃ; i) ウサギのおもちゃ; g) カメ (左) とカエル (右) のおもちゃ.

画像を二値化し、膨張・収縮処理によりノイズを減らす。画像処理には OpenCV を用いた。その後、輪郭抽出処理により正面図と上面図の枠、そしてオプションの枠を探し画像を切り出す。紙が逆さまだった場合でも、切り出した枠の位置から判別できる。その後、二値化画像を用いてマスク処理を行い、ストロークの鮮明化を行う (図 5 a)。次に、切りだされた画像からピクセルの色をサポートベクタマシン (SVM) によって分類する。使用する特徴量は HSV 色空間における 3 つのパラメータとした。現在は 4 色 + 枠の誤認識用に 1 色の計 5 色で学習モデルを作成しており、交差検証の結果 99.6 % の精度を得ている。入力図面は格子にそって描かれていると仮定し、格子点・格子中央・格子点中点の付近のピクセルの色を SVM に入力し、結果を多数決する (図 5 b)。なお、枠と認識された点は排除する。格子中央に色が付いている場合は、塗りつぶしと判定する。格子点が 4 方向塗りつぶしに囲まれている場合は、線がないと判定して除外する (図 5 c)。格子点中点が塗りつぶされている場合は、その格子同士が接続関係にあると考え、格子点同士を接続した重みなし無向グラフを生成する。

作成したグラフを元にストロークを識別していく (図 5 d)。ある頂点から幅優先探索を行い、その頂点に接続されている同じ色の格子点をたどっていく。辿れなくなった場合、格子点間に接続がなくても、同じ色の格子点が隣にあるならばその点を繋げて探索を続ける。画像処理を行っているため、接続関係の抽出が不完全となり、取得したストロークが途切れ途切れになる場合がある。この時、作成された各ストロークの端点同士の最短経路を比較し、短いものが繋がるようにまとめることで対応を行っている。この処理の一部にグラフ演算ライブラリであ

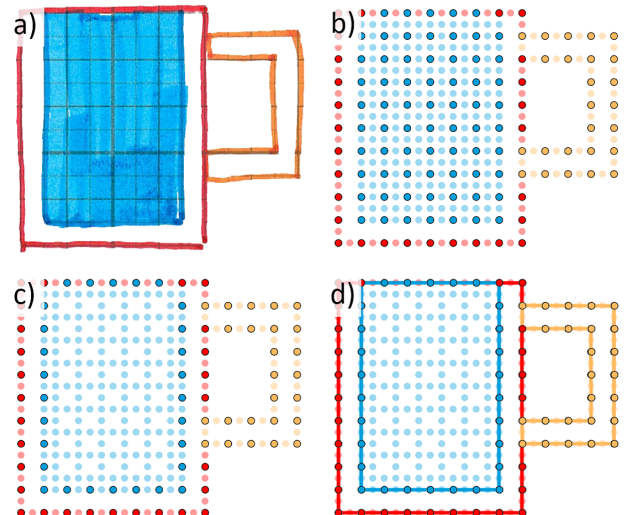


図 5. グラフ化の過程: a) 画像処理による画像の鮮明化; b) ストローク頂点候補の抽出; c) 塗りつぶし領域の除外; d) ストロークの抽出.

る Graphillion<sup>2</sup>を用いている。

## 5.2 図面からのモデルの作成

上記の処理でストローク化した後は、文献 [12] のように 3D モデルに変換する。今回は実装に OpenSCAD<sup>3</sup>を用いた。OpenSCAD はプログラムを書くことで 3D モデルを生成することができるソフトウェアであり、コマンドから呼び出すことが可能である。また、OpenSCAD は 3D モデルデータの一種である STL ファイルを出力することが可能である。正面図と上面図の各シルエットを linear-extrude 命令によってそれぞれ y 軸、z 軸に引き伸ばし、inter-

<sup>2</sup> <https://github.com/takemaru/graphillion/>

<sup>3</sup> <http://www.openscad.org/>

section 命令でそれらの論理積を計算する。こうして作成されたオブジェクトを、塗りつぶされたものと、塗りつぶされていないもので、それぞれ和集合を計算する。その後その2つの差集合を計算し目的の立体を得る。角丸仕上げは `minkowski` 命令を用いて、3D モデルの各頂点に半径 2mm の球を設置し、その凸包をとることで行っている。

ただし、screen-based interface から PUI に置き換えるうえで起きる問題に対処するため、文献 [12] のアルゴリズムを拡張している。具体的には、ユーザが正面図と上面図をずらして描く問題がある。その場合は、ずれを補正し正しい図面に直す。補正後でもモデルの整合性が保たれる場合はそのまま 3D モデル化する。正面図と上面図の横幅が異なる場合、伸縮することで対応できるが、実寸ではなくなるため造形不可とした。

OpenSCAD によって得られた STL ファイルはスライスソフト curaEngine<sup>4</sup>を用いて G-code に変換される。curaEngine は openSCAD と同じくコマンドから呼び出すことが可能である。今回は 3D プリンタとして熱溶融積層法を用いた RepRap 機である Prn3D<sup>5</sup>を用いている。RepRap 機に G-code をシリアル通信で送信することで 3D プリントが可能になる。システムの 3D プリント速度や精度は、使用している 3D プリンタやスライスソフトに依存する。今回は速度を上げるために若干精度を犠牲にする構成とした。造形に用いたのは歪みにくい PLA という樹脂であり、3D プリンタのエクストルーダの温度は 210 度とした。

## 6 議論と制約

### 6.1 設計不可能なもの

球など、投影図では表現できない曲面を持つ物体は本システムでは制作できない。これは文献 [12] にあるように 3D モデルにスムージングをかけることで対応可能である。使うシルエットの数は 2 つのみであるため、三面図に比べて作ることが可能な物体は限られる。例えば円錐や角錐などは二面図では表現が不可能である。また上面図や正面図を描いた際に、直線が多く重なる物体は制作が難しい。例えば、椅子の足などは二面図を描くと重なるため、制作が困難となる。しかし、二面図に比べ三面図を描く労力はおおよそ 1.5 倍となる。ユーザが記述する情報量が増えることで負担が増えすぎないようにするため、本稿ではモデリング手法として二面図を用いた。

### 6.2 画像処理精度

描かれた線の色はシステムが自動識別しているが、今回識別できる色は 4 色となっている。これは画像

処理の性能を上げる事で色を増やすことは可能である。マーカの種類による色の微妙な差なども機械学習によって対応が可能になるだろう。しかし、作例で実証した通り、4 色でも様々な物が設計できるため、本稿ではそれ以上の色への対応は行っていない。

また、形状の自動認識時に認識ミスが起こる場合がある。例えば、方眼の上に線が乗っておらず、塗りつぶしとして判定した例や 1 マスずれて認識した例があった。また、塗りつぶしがはみ出ている、システムが適切に差集合をとらなかった例もある。画像処理でのストローク抽出が不正確だった場合、モデル化の作業で立体が生成できない場合がある。描いた図面のうち、パーツが抜けて造形されることがいくつかあった。また、同じ図面でも、紙の傾きなどの影響でスキャンごとに解析結果が変わる場合もある。図面の枠に重なるように描かれた図面は読み込むことはできない。現在のシステムではスキャン後から造形処理までボタン 1 つで進むため、間違っても造形処理まで進む。そのため、スキャン後に 3D モデルのプレビューを表示し、その後にプリントボタンを押すように変更する必要がある。本稿は、3D プリンタでの出力を前提として、PUI に則った 3D モデル入力の実証が目的であり、本稿の実装の要件は達成されていると考えている。

### 6.3 計算可能性

画像処理を用いてグラフを生成しているため、不完全なグラフが入力される可能性がある。この時、グラフ化する際の条件をゆるくすることでストローク化できる確率が上がる。現在生成されるグラフは辺に重みを持たないが、条件によって重みを持たせることで、精度を犠牲にせずにストローク化を更に正確に行える可能性がある。

画像からストローク化を行う処理には長くて 2 分程度時間を有する場合がある。また、今回は 3D モデル生成に OpenSCAD を用いたため、文献 [12] のように二面図などに計算方法を最適化していない。よって複雑な図形になればなるほど更に計算時間が伸びていく。さらに、角丸仕上げをするために用いている `minkowski` 命令も計算時間が多くかかる処理である。しかし、今回の図面の大きさと複雑さを考えると、各処理は長くて 5 分で終わる。印刷自体に 1 時間やそれ以上の時間がかかることを考えると、この時間は全く無視できるだろう。

## 7 関連研究

紙を用いたモデリング手法には ModelCraft がある [13]。これは、紙でできた物体に対して専用ペンで書いたことが CAD ソフトにも反映されるものである。ModelCraft は紙を使ってはいるが、組み立てなどの作業が必要となるため本提案手法と異なる。また、ブロックを用いて 3D モデリングする研究 [14]

<sup>4</sup> <https://github.com/Ultimaker/CuraEngine/>

<sup>5</sup> <https://micro-factory.net/>

もあるが、それとは違い本提案手法は2次元の図面によってモデリングする手法である。

提案手法では実寸でのモデリングを行える。そのような設計・モデリングを可能にした研究はいくつかされている。Igarashiらは実物体の形状に合わせた布製のカバーを自動生成するアルゴリズムを提案している[8]。AutoConnectは2つのモデルを連結されるためのコネクタに注目し、コネクタを実物体に合った適切な大きさで自動生成するためのシステムである[9]。DrawFormingはvacuum forming methodにより動的に変化する平面を操作するシステムである[10]。等高線をsurfaceの上を書くことで、実物体と合った形の地形的な3Dモデルを制作することが可能である。MiragePrinterは3Dプリンタのステージの上に空中浮遊映像を出せるプリンタであり、実物体との大きさ合わせに使用することもできる[15]。MARCutではマーカベースのレーザカッティング手法を用いることで実物体に合わせたレーザカッティングを行うことが可能である[16]。

Riversらは二面図、または三面図を用いて簡単にモデリングするためのシステムを提案している[12]。提案システムはこのモデリング手法を参考に作成を行ったが、そのほかにもスケッチベースのモデリングシステムはいくつも提案されている[17]。また、3Dモデリングを簡単に行えるようにするため、スマートフォンでそれを行うアプリがいくつか公開されている。特にMakerBot社のスマートフォンアプリであるPrintShop[18]は手書きで描いた絵を3Dモデルに変換する機能を持っている。スケッチから3Dモデルを1から作成するのではなく、描いたスケッチから似た3Dモデルを検索・表示するシステムも存在する[19, 20]。

## 8 展望

今回、画像処理の簡単のためいくつか図面を描く際に制約を設けていた。しかし図面を描く際に制約を設ければ設けるほど作成難易度が上がっていく。画像処理の精度を上げ、制約も減らして自由度を上げることが今後の課題である。本稿では三面図と二面図の比較を行っておらず、ワークショップの参加者数も少ない。今後は更に深く調査をしていくことを考えている。

## 謝辞

本研究は、JST, COIの支援を受けたものである。

## 参考文献

- [1] W. Johnson, et al. Bridging the paper and electronic worlds: the paper user interface. In *Proc. of CHI '93*, pp.507–512, 1993.
- [2] S. Davidoff, et al. The book as user interface: lowering the entry cost to email for elders. In

*Proc. of CHI EA '05*, pp.1331–1334, 2005.

- [3] Shengdong Zhao 他. Magic Cards: 紙カードを利用したロボットとのインタラクション. *インタラクション* 2011, pp.131-138, 2011.
- [4] 中嶋誠 他. アナログ画材を用いたアニメーションの作成を支援するシステム. *情報処理学会論文誌, Volume 56, No. 4*, pp. 1317–1327, 2015.
- [5] J. Garcia and T. Tsandilas. Interactive paper substrates to support musical creation. In *Proc. of CHI '12*, pp.1825–1828, 2012.
- [6] C. Amati and G. J. Brostow. Modeling 2.5d plants from ink paintings. In *Proc. of SBIM '10*, pp.41–48, 2010.
- [7] Y. Akiyama and H. Miyashita. Fitter: A System for Easily Printing Objects that Fit Real Objects. In *Proc. of UIST'16 Adjunct*, 2016.
- [8] Y. Igarashi and H. Suzuki. Cover Geometry Design Using Multiple Convex Hulls. *Computer-Aided Design*, Volume 43, Issue 9, pp.1154–1162, 2011.
- [9] Y. Koyama, et al. AutoConnect: Computational Design of 3D-printable Connectors. *ACM Transactions on Graphics*, Volume 34, Issue 6, Article No. 231, 2015.
- [10] J. Yamaoka and Y. Kakehi. DrawForming: An Interactive Fabrication Method for Vacuum Forming. In *Proc. of TEI '16*, pp.615–620, 2016.
- [11] アーサー・D・フィスク 他. 高齢者のためのデザイン—人に優しいモノづくりと環境設計へのガイドライン. pp.82–84, 慶應義塾大学出版会, 2013.
- [12] A. Rivers, et al. 3D modeling with silhouettes. *ACM Transactions on Graphics*, Volume 29 Issue 4, Article No. 109, 2010.
- [13] H. Song, et al. ModelCraft: capturing freehand annotations and edits on physical 3D models. In *Proc. of UIST '06*, pp.13–22, 2006.
- [14] A. Miller, et al. Interactive 3D Model Acquisition and Tracking of Building Block Structures. *IEEE Transactions on Visualization and Computer Graphics*, Volume 18, Issue 4, pp.651–659, 2012.
- [15] J. Yamaoka and Y. Kakehi. MiragePrinter: interactive fabrication on a 3D printer with a mid-air display. In *Proc. of SIGGRAPH '16*, 2016.
- [16] T. Kikuchi, et al. MARCut: Marker-based Laser Cutting for Personal Fabrication on Existing Objects. In *Proc. of TEI '16*, pp.468–474, 2016.
- [17] L. Olsen, et al. Sketch-based modeling: A survey. *Computers & Graphics*, Volume 33, Issue 1, pp.85–103, 2009.
- [18] PrintShop. <http://www.makerbot.com/apps>. (2016/9/4 確認)
- [19] 辛孝宗 他. 手書きスケッチに基づく3次元シーンのデザインシステム. *WISS2006*, pp.11–14, 2006.
- [20] X. Guo, et al. CustomCut: On-demand Extraction of Customized 3D Parts with 2D Sketches. *Computers & Graphics*, Volume 35, Issue 5, pp.89–100, 2016.

## WISS2016 採録判定時のコメント

---

**採録区分：ショート採録**

**判断理由：**

PC 操作に不慣れな高齢者等でも 3D オブジェクトを製作できるよう、紙に描いた 2 面図から造形物を制作する手法を提案しています。興味深い研究ですが、製作可能な形状に制約が多いことに加え、そもそも研究の意義が不明瞭（＝本当に初心者が使うのか？）という点に課題があることから、ショート採録となりました。

---

**レビューサマリ：**

全体の構成：

本論文では、PC 操作に不慣れな高齢者や子供を対象とし、紙に描いた 2 面図から 3D モデルを生成し、3D プリンタで実際に造形物を出力する手法を提案しています。きちんと動作するプロトタイプを実装し、対象ユーザーを集めてワークショップを行うことで、提案手法の有効性を確認しています。

改良に向けたコメント等：

本研究の意義が明確ではありません。本当に提案システムを初心者が使うのか？という点が疑問です。そもそも「初心者」は立体を 2 面図で表現することはできませんし、提案手法では製作可能な造形物にかなりの制約が生じます。むしろ、レゴブロックを使って制作した立体物をスキャンして 3D データを得る方が、より「初心者向き」と言えるのではないのでしょうか？

より広範な従来研究との比較や深い考察を通じて、本研究の意義について述べてもらいたい。例えば、PC 操作に不慣れなユーザーが、設計にかなりの制約がある造形物を製作することにどのような意味や価値があるのか、またそのようなユーザーが造形物を製作する際に、他にどのような手法が考えられ、これまでどのような取り組みがなされてきたのか、などについての記述があると、論文の主張が明確になり、より説得力を持つようになると思います。

---

本論文に対する各査読者の詳しいコメントは以下のページを参照のこと：

<http://www.wiss.org/WISS2016Proceedings/oral/25.html>

---

**\*本ページは論文本体ではありません**

**【3D プリンタのための Paper User Interface】**