モデルファイル入力に基づく 3D モデルジェネレータ生成システム

木崎駿也† 宮下芳明†

概要：本稿で実現したことは，3Dモデル制作者がそれ以上の新しい知識や技能なしに，パラメトリック 3D モデル（エンドユーザがパラメータ操作によってカスタマイズ可能な 3D モデル）を作れるようにする仕組みである．提案システムは GUI をもたないソフトウェアであり，そこに 2 つの 3D モデルファイル（STL 形式）をドラッグ&ドロップすると，3D モデルジェネレータなる実行ファイルを出力するものとなっている．書き出された 3D モデルジェネレータを受け取ったエンドユーザは，それを起動するだけで，3D モデリングソフトを用いることなくスライダ操作のみで 2 つの 3D モデルの任意の中間体を得て，3D ファイルとして書き出すことができる．

1. システム概要

本稿では最初に，提案システムの概要を記す．提案システムは GUI をもたないソフトウェアであり，そこに 2 つの 3D モデルファイル（STL 形式）をドラッグ&ドロップすると，3D モデルジェネレータなる実行ファイルが得られる．このジェネレータにはプレビュー機能やスライダの GUI が搭載されており，つまみを右端・左端にドラッグすれば，提案システムに入力した 3D モデルそのものを表示し，その間でつまみを移動させれば「2 つのモデルの中間体」が表示される．

3D モデル制作者は 3D モデルではなくこのジェネレータを配信する．そしてジェネレータを受け取ったエンドユーザは，それを起動するだけで，モデリングソフトを用いることなく，スライダ操作で 3D モデルの多少の変形やカスタマイズを行える．

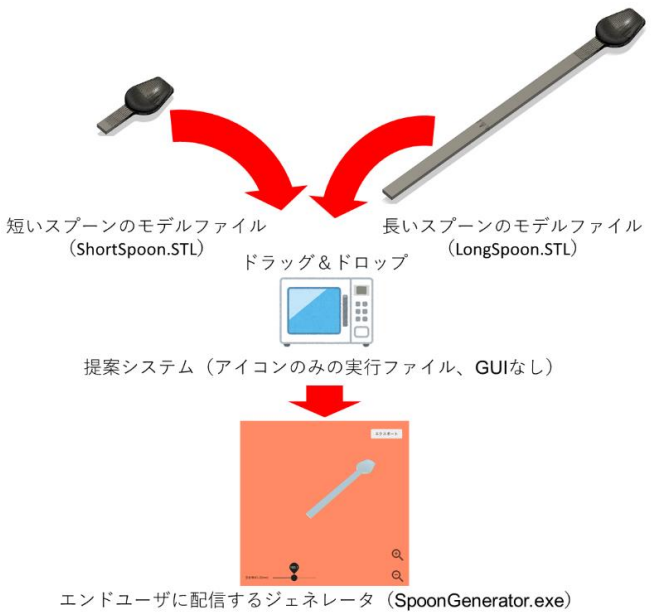

図 1　3D モデル制作者が提案システム（アイコンのみ）に 2 つの 3D モデルファイル（ShortSpoon.STL，LongSpoon.STL）をドラッグ&ドロップし，実行ファイルのジェネレータ（SpoonGenerator.exe）を得る．このジェネレータをエンドユーザ向けに配信する．

本稿における提案は，こうした「ジェネレータを生成する」ためのシステムであり，前述の通り GUI すらもたない．3D モデルを制作・配信したいユーザは，その 3D モデルのバリエーションを 2 つ作ってドラッグ&ドロップするだけで，「モデリングソフトなしにモデルをカスタマイズできるジェネレータ」を得ることができる．3D モデル制作者にとって，新たに獲得すべき知識や技能は不要である．

具体的に，スプーンについての事例を示す．図 1 の上部にあるのは，3D モデル制作者が配信しようとしているスプーンにおいて，その持ち手を短くしたもの（ShortSpoon.STL）と長くしたもの（LongSpoon.STL）である．この 2 つのファイルを提案システム（図 1 中部）にドラッグ&ドロップすると，図 1 下部にあるようなジェネレータ（SpoonGenerator.exe）が書き出される．

図 2 にこのジェネレータの挙動を示す．ジェネレータは.exe 形式の実行ファイルであり，対象ユーザは，若干のカスタマイズを行って 3D プリントしたいと考えるエンドユーザである．このジェネレータでは，画面ドラッグでモデルを回転させて様々な方向から閲覧したり拡大縮小で細部を確認できるだけでなく，画面左下にあるスライダの操作によって変形することができる．その最大・最小は，入力した 2 つのファイルに基づいている．図 2 左はつまみが左端位置のとき，図 2 右はつまみが右端位置のときであるが，それぞれ，図 1 で提案システムに入力したファイルと同一となっている．つまみをドラッグすれば間の長さを得ることができ，図 2 中央のようにほぼ中点にある場合は，中間のものが得られる．

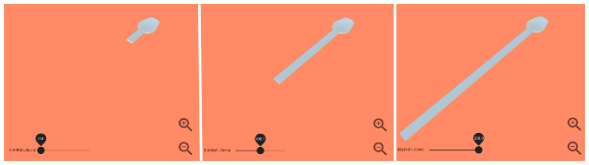

図 2　ジェネレータでのスライダの挙動（左：つまみが左端位置，中央：つまみが中央位置，右：つまみが右端位置）

† 明治大学

© 2019 Information Processing Society of Japan

これをエクスポートすると，ジェネレータで設定したとおりの 3D モデルファイル（MySpoon.STL）が書き出されるので，これを 3D プリントして使用する（図 3）．

図 3　ジェネレータを用いてエンドユーザが自分用に書き出したスプーンのモデルファイル（MySpoon.STL）

もうひとつ事例を示す．図 4 上部にあるのは，3D モデル制作者が配信しようとしている電池ボックスにおいて，その個数を 2×2 で作ったもの（BatteryBox2X2.STL）と 5 × 5 で作ったもの（BatteryBox5X5.STL）である．この 2 つのファイルを，提案システム（図 4 中部）にドラッグ＆ドロップすると，図 4 下部にあるようなジェネレータ（BatteryBoxGenerator.exe）が書き出される．

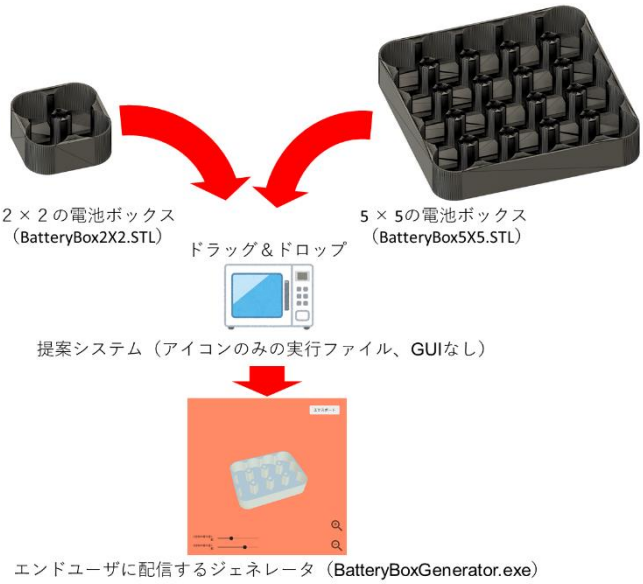

図 4　3D モデル制作者が提案システム（アイコンのみ）に 2 つモデルファイル（BatterBox2X2.STL，BatteryBox5X5.STL）をドラッグ＆ドロップし，実行ファイルとしてジェネレータ（BatteryBoxGenerator.exe）を得る．3D モデル制作者はこれをエンドユーザ向けに配信する．

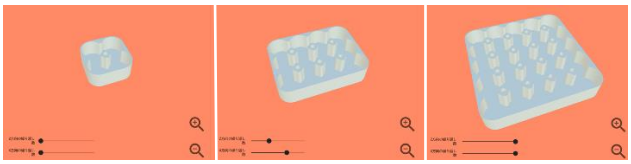

図 5　ジェネレータにおけるスライダの挙動（左：つまみが左端位置，中央：つまみが中央付近の位置，右：つまみが右端位置）

図 5 に書き出されたジェネレータの挙動を示す．今回は，先ほどのスプーンと異なり，スライダが 2 つ生成される．片方のつまみを操作すると縦方向の電池数が増減し，もう片方では横方向の電池数が増減する．下限は 2×2（図 5 左），上限は 5×5（図 5 右）で，たとえば 3×4 の電池ボックスを生成することが可能である（図 5 中央）．

このスライダの本数は，入力したファイルに基づいて生成されている．たとえば，2×2 と 2×5 の電池ボックスのファイルをドラッグ＆ドロップした場合は，縦の個数が同じため，図 6 のように 1 本のスライダしか生成されない．こうしてスライダ操作によって縦横の個数を調整したうえでエクスポートすると，設定したとおりの 3D モデルファイル（MyBatteryBox.STL）が得られる．エンドユーザはこれを 3D プリントして使用する（図 7）．

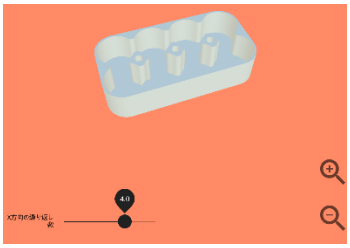

図 6　提案システムに 2×2 と 2×5 の電池ボックスのファイルをドラッグ＆ドロップしたときに生成されるジェネレータの画面．スライダの本数が 1 本となっている．

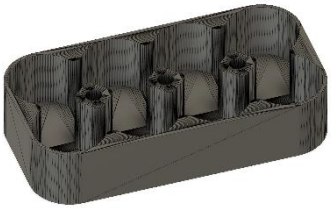

図 7　ジェネレータを用いてエンドユーザが自分用に書き出した電池ボックスのモデルファイル（MyBatteryBox.STL）

2.　背景

今日，パラメトリック 3D モデルなるモデル形式が提案されている．これは，モデリングソフトに習熟していないエンドユーザでも，パラメータ操作だけでカスタマイズを行える 3D モデルである（図 8）．

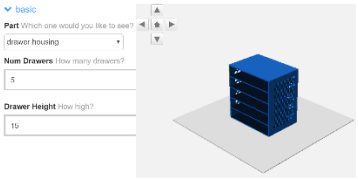

図 8　Thingiverse 上で提供されるパラメトリック 3D モデル

エンドユーザにとっては敷居が低く利便性の高いパラメトリック 3D モデルであるが，3D モデル制作者がこれを簡単に配信できるかというと，実はそうではない．モデリングの技術に加えて，プログラミングの知識やパラメトリックモデリングに関する技術が要求されるためである．

© 2019 Information Processing Society of Japan

たとえば，3D モデリングソフト OpenSCAD や CraftML では，3D モデリングした後に CUI 操作によって変数を設定し，3D モデルのどの箇所を変数で変形できるかを細かく指定しなければならない[2][3]．また，GUI による 3D モデリングソフトである Fusion360 には，3D モデルをインポートした後に変数を設定し，3D モデルのどの箇所をパラメータで変形するのか指定する機能がたしかに備わっている[4]．しかし，変数の設定を行い，面に適用するといった操作が必要になり，煩雑である．こうしたことが原因してか，エンドユーザがカスタマイズしやすいパラメトリック 3D モデルがインターネット上で提供されることは極めて稀で，3D モデル共有サイト Thingiverse で 99%の 3D モデルは非パラメトリックな 3D モデルであり，エンドユーザがカスタマイズするのは難しいままとなっている[1]．

本稿における筆者らの着眼は，3Dモデル制作者が「パラメトリック 3D モデルを作る」ことをいかに容易にするかにある．3Dモデル制作者が「それ以上の新しい知識や技能なしに」パラメトリック 3D モデルを作れるようにするにはどうしたらよいか思考した結果，1 章で紹介したような，「2 つのファイルをドラッグ&ドロップするだけの，GUI を持たないシステム」に行き着いた．以下，このシステムの実装，評価，考察について述べる．

3. 実装

3.1 3D モデルジェネレータ生成システム概要

提案システムは Unity，OpenSCAD，MeshLab を使用し実装した．システムに最小値を持つモデルと最大値を持つモデルの 2 つを入力すると，システムは交差する部分の 3D モデルを生成する．ここでは，入力された最小値のパラメータを持つ 3D モデルを A，最大値のパラメータを持つ 3D モデルを B，システムが生成した 3D モデルを C とする（図 9）．これらの 3D モデルを元に，3D モデルの変形，自動選択が行われ，最小値から最大値の範囲をパラメータ操作できる 3D モデルジェネレータが書き出される（図 9）．

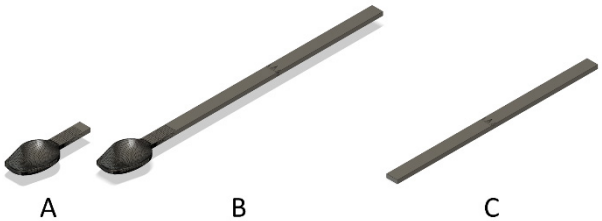

図 9 持ち手が短いスプーンと長いスプーンを入力したときに生成される 3D モデルの例

3.2 引き伸ばしの変形

3D モデルの一部分を最小値から最大値の範囲で引き伸ばす変形が可能となる．例として，スプーンを取り上げる．持ち手が短いスプーンと持ち手が長いスプーンを入力することで，短いスプーンから長いスプーンの長さの範囲で 3D モデルを生成できる 3Dモデルジェネレータが生成される．システムは C の尺度の基準点を A に最も近い面の中心に設定する．引き伸ばしが可能な方向は，XYZ 座標において，A と B の大きさの内，差分がある方向である．パラメータの単位は mm で，設定された C の尺度の基準点を元に尺度を変更する．パラメータの最小値は 0 に設定され，最大値は A と B を直方体で覆った際の大きさの差分値が設定される．パラメータを最小に設定した場合，生成される 3D モデルはA と同一になる．パラメータを最大に設定した場合，生成される 3D モデルは B と同一になる（図 10）．

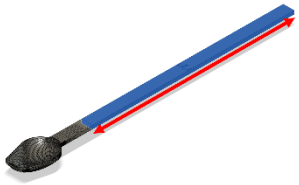

図 10 パラメータ操作でスプーンの持ち手が引き伸ばされる例

3.3 繰り返しの変形

提案システムでは 3D モデルを最小値から最大値の回数で繰り返す変形が可能である．前述の電池ボックスの事例の場合を図 11 に示す．システムは A の 3D モデルを 4 分割する．この 3D モデルを A’とする．分割した 3D モデル 1 つ 1 つを覆う直方体の大きさを計算する．その直方体を引き伸ばしてB の外周全てを覆う大きさに変更する．そして，その直方体で B の 3D モデルの外周を切り取る．外周を切り取った B の 3D モデルの表面の高さを検査し，同じ形状の繰り返しが何回行われているか計算する．

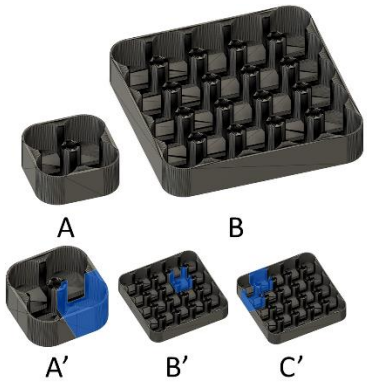

図 11 2×2 の乾電池ボックスと 5×5 の乾電池ボックスを入力したときに生成される 3D モデル（A’，B’，C’は青色の 3D モデルを生成）

繰り返しが行われている数を元に，外周を切り取った B のモデルを分割する．この中の 1 つの 3D モデルを B’とする．また，繰り返し数を元に切り取った外周を分割する．この中の端以外の 3D モデルを C’とする．パラメータは 2 つ表示され，最小値は 2 に，最大値は繰り返し数+2 に設定される．2 つのパラメータの数値をかけた数だけ 3D モデルが配置される．端は対応する A’，外周は C’，中央は B’の 3D モデルを配置する（図 11，12）．

© 2019 Information Processing Society of Japan

図 12　パラメータ操作で乾電池を入れる個数を設定可能

3.4 変形方法の自動選択

3D モデルの変形方法は 3D モデルを覆った直方体の大きさで選択される．B が A と比較して一方向に大きく，繰り返しのパターンが無い場合，引き伸ばしの変形方法が選択される．繰り返しパターンがある場合や，A が B よりも 2 方向に十分に大きい場合，繰り返しの変形方法が選択される．

3.5 3D モデルジェネレータ

システムが変形処理を終了した後，GUI 上でパラメータを操作して 3D モデルを生成できる 3D モデルジェネレータ画面が表示される．パラメータ名は変形方法によってシステムが自動的に決定する．パラメータ名には変形方法の名前と，変形の際に使われる単位が設定される．3D モデル使用者はこの画面でパラメータを操作し，エクスポートボタンをクリックすることで希望の 3D モデルを手に入れることができる（図 13）．また，3D モデルジェネレータ画面では，ドラッグで視点変更，拡大縮小ボタンをクリックするとズームイン，ズームアウトが行える．

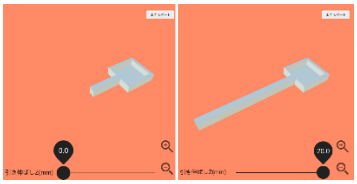

図 13　パラメータ操作で長さを設定（mm 単位）

4. 評価

4.1 目的

3D モデルジェネレータ生成システムを使用することで，パラメータの設定を行うことができない 3D モデル制作者が想定したパラメータ範囲を持つ 3D モデルジェネレータを生成できるのか調べる．

4.2 参加者

ノンパラメトリック 3D モデル制作経験，3D プリンタ使用経験がある大学生 3 人（22～24 歳）に実験を行った．実験参加者は全員違うモデリングソフトを使用していて，参加者 A は 3DCAD ソフト 123D Design，B は 3DCAD ソフト Fusion 360，C は 3DCG ソフト Blender を使用して 3D モデルを制作した．いずれの参加者もソフトの使用年数は 3 年である．

4.3 実験手順

最初に，実験参加者にシステムの使用方法を説明し，参加者が 3D プリンタを使い制作した 3D モデルを出力する際に適切な最小値のパラメータを持つ 3D モデルと最大値のパラメータを持つ 3D モデルを入力するように伝えた．次に，システムの各変形の成功例と失敗例を見せ，どのような 3D モデルを制作する必要があるのか説明した．その後に 3D モデル共有サイト Thingiverse でパラメータが設定されている 3D モデルの例を紹介した．以上でチュートリアルは終了である．チュートリアルが終了した後，参加者に自由に 3D モデルを制作させ，制作した 3D モデルを元に 3D モデルジェネレータを生成するタスクを行わせた．3D モデルジェネレータ生成後にアンケートを行った．その後，3D モデルジェネレータのアルゴリズムと同様に著者が 3D モデリングを行い，出力した 3D モデルを実験参加者に見せた．

4.4 実験環境

参加者のパソコン上で 3D モデルの制作を行わせた．制作が完了した後，USB メモリに 3D モデルデータを入力し，筆者のパソコンの指定したファルダに入力し，バッチファイルを実行することで，3D モデルジェネレータ生成システムを実行する環境を準備した．

4.5 実験結果

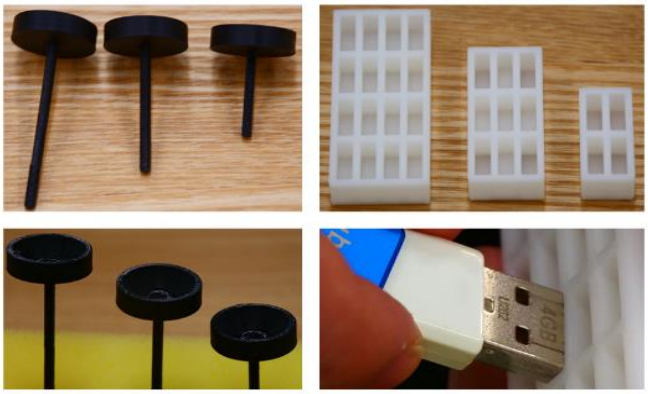

図 14　参加者 A が制作したゴルフのティー（左）と参加者 B が制作した USB メモリ立て（右）

参加者 A は，3D モデルジェネレータの生成に成功した．彼はゴルフのティーの 3D モデルジェネレータを生成した．参加者 B は，USB 差しの 3D モデルジェネレータの生成に成功した．参加者 C は 3D モデルジェネレータの生成に失敗した．また，3D モデルジェネレータの生成に成功した参加者は想定したパラメータの最小値と最大値の設定に成功していた．加えて，3D モデルジェネレータでパラメータを中間に合わせて書き出した 3D モデルを参加者に見せたところ，予想通りの 3D モデルが生成されたと答えた（図 14）．参加者 B が制作した USB メモリ立ては，穴のサイズが USB メモリと合わなかったため使用することはできなかったが，再度システムを使用することで正しいサイズに設定可能であった．

© 2019 Information Processing Society of Japan

4.6 アンケート

参加者 A と B はパラメータを意識して 3D モデル制作を行った．参加者 C はシステム 3D モデルを入力する際に，変形方法が正しく選択されるように 3D モデルの制作を行った．参加者 A は最初に制作した 3D モデルをパラメータで変形できることを望んだ．参加者 C は回転角度やそれぞれの変形方法を組み合わせることを望んだ．

5. 考察

参加者 B と参加者 C はシステムに 3D モデルを入力し，3D モデルジェネレータの生成に失敗した際に，システム側に原因があると考え，著者が助言を行うまで，3D モデル制作とシステムの使用を行おうとしなかった．参加者 B に 3D モデルの一部の大きさが異なるため，変形処理が有効に実行されないことを伝えると，参加者 B は 3D モデルを修正し，3D モデルジェネレータの生成に成功した．参加者 C は出力される 3D モデルデータに問題があり，3D モデルジェネレータが生成できなかった．システムが 3D モデルジェネレータ生成に失敗した場合，有効なエラーメッセージを提示する必要があると考えられる． 3D モデルジェネレータの生成に成功した参加者 A と参加者 B はパラメータの微調整を行うことなく，1 回の生成で，想定する範囲のパラメータの設定に成功していた．また，参加者が設定したパラメータの中間値の 3D モデルを 3D プリンタで出力し，見せたところ，予想通りの 3D モデルが出力されたと答えた．これは，制作者がシステムにパラメータの最小値の 3D モデルと最大値の 3D モデルを入力しているため，その間の 3D モデルの形状を予想でき，複数回パラメータの調整を行う必要が無かったと考えられる．参加者全員はチュートリアルで説明した 3D モデルの特徴と同一になるように 3D モデリングを行っていた．どの参加者も 3D モデル制作中にシステムに関する説明を求めた．また，参加者 B と C は，3D モデル制作において，システムが正しく 3D モデルジェネレータを生成できるようにするため，3D モデルを一定の形状と位置に配置するように注意していた．そのため，制作者が希望する 3D モデルの形状から離れてしまった可能性がある．

6. 制約と展望

システムは 2 つの 3D モデルの位置，角度が正確に一致する場合でなければ，3D モデルジェネレータ生成に失敗する．この仕様は制作者が独自の形状を制作することを妨げる要因になると考えられる．変形方法の自動認識アルゴリズムに 3D モデルの位置や角度がずれていた場合，修正するといった処理を行うことで解決できるだろう． また，曲面が含まれている 3D モデルや分割数が多い 3D モデルを入力した場合に誤差が発生し，システムが 3D モデルの自動認識や変形処理を失敗してしまうことが実験中何度か見られた．自動認識と変形処理のアルゴリズムの変更や，誤差が発生しないような処理方法を選択することで解決できると考えている．

現在，システムの 3D モデルの変形方法は 2 種類のみで，引き伸ばしは 1 方向のみ，繰り返しは最大 2 方向のみ可能である．また，制約として変形方法は併用できない．そのため，様々な種類の 3D モデルを制作可能とは言えない．3D モデル制作者がパラメータ化したい箇所を調査し，変形方法のバリエーションを増やす．変形方法の併用を可能とすることでより多くの種類の 3D モデルをパラメータで生成できるように改良を継続したい．

7. 関連研究

MetaMorphe ではバイナリファイルである STL フォーマットによって 3D モデル制作の自由度が制約されてしまうことを問題とし，STL ファイルに対してパラメータや天気情報などを使い，ユーザが新たな 3D モデルを制作することが可能なシステムを提案している[5]．Grafter では 3D モデルの汎用的な機構を使って，新たな 3D モデルを制作する際に組み合わせることが困難であることを問題とし，3D モデルを変形することで解決している[6]．ModelCraft では印刷した 3D モデルにメモを書くのは簡単であるが，そのメモを PC 上に反映させることが困難であることを問題とし，光学パターンを印刷した紙をユーザが組み立て，組み立てた 3D モデルに対してメモを書くことで PC 上に反映できる手法を提案した[7]．f3.js ではソフトウェア開発者が行う IoT アプリケーションのプロトタイピングには CAD システムの学習コストが高く，センサやアクチュエータの取り付けが困難であり，組み立ての際に機材が多く，手間も大きいといった問題点を述べている．これらの問題を解決する統合開発環境を提案している[8]．

8. まとめ

本研究では最小値のパラメータを持つ 3D モデルと最大値のパラメータを持つ 3D モデルを入力することで，その間にある 3D モデルを生成可能とする．3D モデルジェネレータを生成できるシステムを提案した．システムの評価では 3D プリンタを使用した経験がある 3D モデル制作者 3 人にシステムを使用させ，1 人は 3D モデルジェネレータの生成に成功し，1 人は著者が助言を行うことで 3D モデルジェネレータの生成に成功した． 3D モデルジェネレータの生成に成功した 2 人はパラメータの調整を試行錯誤することなく想定のパラメータ範囲を設定でき，提案システムはパラメータを持つ 3D モデルを制作するときに有用であることが分かった．

参考文献

[1] Lora Oehlberg, Wesley Willett, and Wendy E. Mackay. 2015. Patterns of Physical Design Remixing in Online Maker

© 2019 Information Processing Society of Japan

Communities. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15). ACM, New York, NY, USA, 639-648. DOI: https://doi.org/10.1145/2702123.2702175

[2] OpenSCAD - The Programmers Solid 3D CAD Modeller, http://www.openscad.org/

[3] Tom Yeh and Jeeeun Kim. 2018. CraftML: 3D Modeling is Web Programming. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18). ACM, New York, NY, USA, Paper 527, 12 pages. DOI: https://doi.org/10.1145/3173574.3174101

[4] Fusion360 https://www.autodesk.co.jp/products/fusion-360/overview

[5] Cesar Torres and Eric Paulos. 2015. MetaMorphe: Designing Expressive 3D Models for Digital Fabrication. In Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition (C&C '15). ACM, New York, NY, USA, 73-82. DOI: https://doi.org/10.1145/2757226.2757235

[6] Tianyi Zhang and Miryung Kim. 2018. Grafter: transplantation and differential testing for clones. In Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings (ICSE '18). ACM, New York, NY, USA, 422-423. DOI: https://doi.org/10.1145/3183440.3195038

[7] Hyunyoung Song, François Guimbretière, Chang Hu, and Hod Lipson. 2006. ModelCraft: capturing freehand annotations and edits on physical 3D models. In Proceedings of the 19th annual ACM symposium on User interface software and technology (UIST '06). ACM, New York, NY, USA, 13-22. DOI: https://doi.org/10.1145/1166253.1166258

[8] Jun Kato and Masataka Goto. 2017. f3.js: A Parametric Design Tool for Physical Computing Devices for Both Interaction Designers and End-users. In Proceedings of the 2017 Conference on Designing Interactive Systems (DIS '17). ACM, New York, NY, USA, 1099-1110. DOI: https://doi.org/10.1145/3064663.3064681

© 2019 Information Processing Society of Japan