

# 卒業論文

How to make almost any Cosplay.

慶應義塾大学 環境情報学部

宇田周平

2013 年 1 月 22 日

指導教員 田中浩也准教授

# 目次

|       |                             |    |
|-------|-----------------------------|----|
| 第 1 章 | はじめに .....                  | 6  |
| 1.1   | 本研究のきっかけ .....              | 6  |
| 1.2   | 本論文の構成 .....                | 7  |
| 第 2 章 | 研究背景 .....                  | 8  |
| 2.1   | コスプレ .....                  | 8  |
| 2.1.1 | コスプレの種類 .....               | 8  |
| 2.1.2 | コスプレを取り巻く環境 .....           | 8  |
| 2.1.3 | コスプレとパーソナルファブリケーション .....   | 9  |
| 2.2   | ニコニコ動画 .....                | 10 |
| 2.3   | 実際に製作してきたもの .....           | 11 |
| 2.3.1 | 夜天の魔導書・蒼天の魔導書 .....         | 11 |
| 2.3.2 | 魔法陣 .....                   | 12 |
| 2.3.3 | バルディッシュアサルトライオットブレードⅡ ..... | 13 |
| 2.3.4 | 製作を通じて .....                | 14 |
| 第 3 章 | 研究背景 .....                  | 15 |
| 3.1   | オープンソース .....               | 15 |
| 3.2   | デジタルファブリケーション .....         | 15 |
| 3.2.1 | CAD/CAM .....               | 15 |
| 3.2.2 | アパレル CAD .....              | 17 |
| 3.2.3 | Grasshopper .....           | 17 |
| 第 4 章 | 設計 .....                    | 18 |
| 4.1   | 問題発見 .....                  | 18 |
| 4.2   | 目的 .....                    | 18 |
| 4.3   | 機能要件 .....                  | 19 |
| 4.4   | 関連研究 .....                  | 19 |
| 第 5 章 | 実装 .....                    | 20 |
| 5.1   | 実装環境 .....                  | 20 |

|         |                              |    |
|---------|------------------------------|----|
| 5.2     | プロトタイプの実装.....               | 20 |
| 5.2.1   | 概要.....                      | 20 |
| 5.2.2   | パラメータの算出アルゴリズム.....          | 21 |
| 5.2.2.1 | 身長.....                      | 21 |
| 5.2.2.2 | ウエスト.....                    | 22 |
| 5.2.2.3 | 腕の長さ.....                    | 24 |
| 5.2.2.4 | 体型スキャン時の姿勢.....              | 24 |
| 5.3     | プロトタイプの展示.....               | 25 |
| 5.4     | 改良版の実装.....                  | 25 |
| 5.4.1   | DXF ファイルの読み込みとエンティティの扱い..... | 25 |
| 5.4.1.1 | LINE.....                    | 25 |
| 5.4.1.2 | ARC.....                     | 26 |
| 5.4.1.3 | CIRCLE.....                  | 26 |
| 5.4.2   | ズーム・ドラッグ.....                | 27 |
| 5.4.3   | エンティティの選択.....               | 27 |
| 5.4.4   | 線分の伸縮時に移動する頂点の選択.....        | 28 |
| 5.4.5   | エンティティの伸縮.....               | 29 |
| 5.4.6   | 隣接するエンティティの自動修正.....         | 32 |
| 5.4.7   | Kinect でスキャンし、算出した値の適応.....  | 34 |
| 第 6 章   | 評価.....                      | 36 |
| 6.1     | 評価方針.....                    | 36 |
| 6.1.1   | 定量評価.....                    | 36 |
| 6.1.2   | 定性評価.....                    | 36 |
| 6.2     | 評価結果.....                    | 36 |
| 6.2.1   | 所要時間.....                    | 37 |
| 6.2.2   | 編集の容易さ.....                  | 37 |
| 6.2.3   | 再利用性の向上.....                 | 38 |
| 6.2.4   | Kinect によるスキャンの有効性.....      | 38 |
| 6.2.5   | まとめ.....                     | 38 |
| 第 7 章   | 結論.....                      | 40 |
| 7.1     | 本研究のまとめ.....                 | 40 |
|         | 参考文献.....                    | 41 |
|         | 謝辞.....                      | 43 |

# 図表目次

|                                              |    |
|----------------------------------------------|----|
| 図 1 デバイス設計図 1 .....                          | 6  |
| 図 2 Device 3D Image. Ver2nd.....             | 6  |
| 図 3 コスプレイヤーズアーカイブ .....                      | 9  |
| 図 4 Cure .....                               | 9  |
| 図 5 ニコニコ動画.....                              | 11 |
| 図 6 夜天の魔導書・蒼天の魔導書.....                       | 11 |
| 図 7 UV インクジェット印刷による魔法陣.....                  | 12 |
| 図 8 バルディッシュアサルト ライオットブレードⅡ（SolidWorks） ..... | 13 |
| 図 9 バルディッシュアサルト ライオットブレードⅡ（製作物） .....        | 13 |
| 図 10 デジタルファブリケーションの流れ.....                   | 16 |
| 図 11 身長 of 算出方法.....                         | 22 |
| 図 12 ウエスト of 算出方法.....                       | 23 |
| 図 13 腕の長さ of 算出方法.....                       | 24 |
| 図 14 円弧伸縮時の隣接エンティティの自動補正 .....               | 30 |
| 図 15 所要時間.....                               | 37 |
|                                              |    |
| 表 1 製作に使用したもの .....                          | 14 |
| 表 2 隣接するエンティティと自動修正を行うエンティティの組み合わせ....       | 33 |
| 表 3 moveType 毎の座標の修正 .....                   | 34 |

# ソースコード目次

|                          |    |
|--------------------------|----|
| ソースコード 1 円弧の取り扱い .....   | 26 |
| ソースコード 2 マウス座標の絶対値 ..... | 27 |
| ソースコード 3 円弧の選択判定 .....   | 27 |
| ソースコード 4 移動する頂点の選択 ..... | 28 |
| ソースコード 5 線分の伸縮 .....     | 29 |
| ソースコード 6 円弧の伸縮 .....     | 31 |
| ソースコード 7 数式処理 .....      | 34 |

3 杖や斧，魔導書などの武装.

多数公開されているし、素材は公開していないにせよ、製作記のようなものを公開している人は多い。実際、ブログに製作物の写真を公開していたプラモデル好きの高校生が実は私の卒業した高校の後輩だった事や、イベントには参加していないが、アルミ加工をして1/1 デバイスを製作していた方に図面を譲ってもらったという経験もある。そしてもちろんニコニコ動画を探せば、ニコニコ技術部ならぬニコニコ兵器開発局や時空管理局技術部といったタグが付けられた多数の作品を見つけることができる。

二次創作に加えて、コスプレの人気の高い事もリリカルなのはシリーズの特徴である。コスプレ衣装を販売するコスパティオ [4]では10万円前後と極めて高価なコスチュームが各シリーズ受注生産限定で発売され、それらが中古市場に流れてくると未だに数日で売り切れるといった状態である。また、小道具や大道具についても需要が大きく、同人イベントでは単価300円～500円の同人誌が多い中、20000円前後の杖などを販売するサークルも存在している。

## 1.2 本論文の構成

本論分は、リリカルなのはの同人活動に強く影響を受けた筆者が、デジタルファブリケーションやソーシャルファブリケーションといった新しいモノづくりの手法を経験して得た知見について述べたのち、それらの普及に必要なデータの再利用性向上という観点から作成したソフトウェアの実装について述べる。

まず初めに第2章・第3章では本研究にいたるまでの著者の体験や研究背景について整理し、それを受けて今回作成したプログラムの設計についてを第4章で、第5章および第6章で実装ならびに評価を述べ、第7章で結論について述べる。

## 第2章 研究背景

第2章と第3章では、近年のサブカルチャーおよびソーシャルファブリケーションの現状について、関連研究および実際の製作物を例に挙げながら述べる。

### 2.1 コスプレ

#### 2.1.1 コスプレの種類

コスプレと言っても様々な種類のものが存在する。アニメやゲームのキャラクターといった実在しないキャラを演じるもの、耳や尻尾など動物を連想させるもの、ヴィジュアル系、異性装、制服や和装といった普段身に付けないものを着ることでさえも現代においてはコスプレと呼ばれることがある。

#### 2.1.2 コスプレを取り巻く環境

前章で述べたリリカルなのはシリーズに限らず、コスプレ市場自体が近年拡大の一途をたどっている。2011年度のコスプレ市場は415億円という調査もあり、それによれば「「コスプレ」が一般的に認知されつつあることや、参加人口の増加により、市場規模拡大傾向である。」とされている。[5]

コスプレといえば以前はコミックマーケットやワンダーフェスティバルのような同人イベントや、TFT（東京ファッションタウンビル）や東京国際交流館、としまえんなどで開催されるコスプレイベントのように限られた場所でのみ行われていたが、現在においては学園祭でのコスプレ喫茶や、日常のイベント（ハロウィンやクリスマスなど）にまで広く浸透し、衣装自体もコスプレショップのみならずドンキホーテなどで手軽に購入できる状況となっている。

また、コスプレ専門雑誌や、コスプレ専門のSNSといったものも存在し、自分のコスプレ写真を投稿することや、コスプレイヤー同士でのコミュニケーション、必要がなくなったコスプレ衣装やウィッグの売買なども可能となっている。



なお、コスプレ文化については、「コスプレ女子の時代」 [6]や「コスプレする社会—サブカルチャーの身体文化」 [7]が詳しいが、それによれば「お手製であること」、「自分自身の手作業によるものであること」に価値を置くという傾向が見られる。とされ、「DIY 精神」ともいえるような価値観が、主に衣装政策の面に関して強く見られるのである。そこにはクリエイティブな作業への執着が垣間見られる。「生産行為」を絶対に他者に譲り渡そうとはしない精神がコスプレ文化を貫いている。と述べられている。



図 3 コスプレイヤーズアーカイブ [8]



図 4 Cure [9]

## 2.1.3 コスプレとパーソナルファブリケーション

コスプレ産業の拡大により、現在では多くのアニメやゲームキャラクターのコスプレ衣装が専門店やオークションサイトで販売され、入手できるようになった。とはいえ、増え続けるアニメやゲーム、ライトノベルのキャラクターの全てを網羅できるわけではない。また、市販されている衣装では細部の作りが甘く納得できないといった事も頻繁に起こりうる。そのため、前述のように自分自身で細部にまでこだわって作ることが重要視されてきたのである。

また、靴や小物、長物と呼ばれる武器類はよほど有名な作品で入手が可能な場合か、汎用品を使いまわせる場合以外は自分自身で作るか妥協するほかない。大量生産される市販品だけでは膨大なニーズをまかないきれないのである。2.3 において

著者自身が作成した造形物についても言及するが、これらと同じものを作っているのは日本中探しても数えるほどしかないだろうし、それぞれに作者のこだわりが詰まっていて、ひとつとして同じものは存在しない。

手作りが重視されるとはいえ、コスプレをする全員が造形技術を持ち合わせているわけではない。そのため、杖などを作る造形師と呼ばれる人々は重宝される事が多く、著者自身も複数の知人のために何度か依頼を受けて製作した経験を持っている。ただ、作る側が当該作品や依頼されたものの細部までを知っているとは限らず、Web上の画像や市販されている設定資料集などを参考にするとはいえ、細部の作りこみは甘くなってしまう。やはり、ほしい人が自分自身の手で作るということが最終的に求められる。そういった意味で、コスプレとパーソナルファブリケーションの相性は極めてよいと言う事ができる。

## 2.2 ニコニコ動画

ニコニコ動画 [10]は 2006 年よりサービスが開始された株式会社ニワンゴの提供する動画投稿サイトである。ニコニコ動画の特徴としては、動画にコメントを付与でき、そのコメントが動画上に表示されるという点が上げられる。また、タグと呼ばれるキーワードを最大 10 件まで登録することが可能となっている。中でも「ニコニコ技術部」や「作ってみた」といったタグの付けられた作品はユーザーが自ら製作した様々なソフトウェア・ハードウェア、造形物等の製作過程やデモムービーが投稿されており、中にはプロ顔負けの（もしくはプロが趣味で作成した）ハイクオリティな作品も数多く存在している。また、音声合成ソフト VOCALOID のキャラクターである初音ミクに関連した動画などにおいては、N 次創作とよばれる派生現象も生じている。



図 5 ニコニコ動画

## 2.3 実際に製作してきたもの

本節ではリリカルなのはに関連した著者の製作物と、その過程ならびに利用した工作機械について言及する。

### 2.3.1 夜天の魔導書・蒼天の魔導書



図 6 夜天の魔導書・蒼天の魔導書

高校時代から複数回にわたって改良を加え、同人イベントで販売もした魔導書である。表紙が紙製のものと、ポリウレタン合皮の 2 種類を製作した。3D モデルからワイヤーフレームを抜き出し、そのデータを元に CraftROBO でカットをし、表紙の模様を作っている。また、本文には著作権フリーの聖書や、Web で配布されて

いるミッドチルダフォント [11]など、いずれも他人の作成したデータを素材として利用している。製本作業もすべて Web 上に公開されていた手順を参考にし、無線綴じや糸かがり綴じを手作業で行った。また、製作過程を撮影した動画は現在ニコニコ動画において 2 作品<sup>4</sup>で合計 2 万再生程である。イベントにおいては 3000-5000 円で販売したが、後に中古のコスプレ衣装等を扱うショップ [12]に持ち込んだ際には 7000 円の値がついた事もある。

### 2.3.2 魔法陣

ニコニコ動画に投稿されていた AR 作品 [13]に影響され、印刷業者 [14]に UV インクジェット印刷で外注したものと、レーザーカッターによって彫刻したアクリルの側面からフルカラーテープ LED を使って光らせるものの 2 種類を製作した。前者は魔導書同様イベントで販売も行った。後者については ArduinoFIO および XBee を用いて LED の色を変更するリモコンも作成したが、本体の LED を光らせるために最大 4A 程度必要なため、電源の小型化が現在も課題として残っている。なお、元となったデータは Web 上で公開されていたパスデータ付きの PSD を利用したが、現在では当該サイトは閉鎖され残っていない。



図 7 UV インクジェット印刷による魔法陣

---

<sup>4</sup> <http://www.nicovideo.jp/watch/sm6906965>  
<http://www.nicovideo.jp/watch/sm8228696>

### 2.3.3 バルディッシュアサルトライオットブレードⅡ

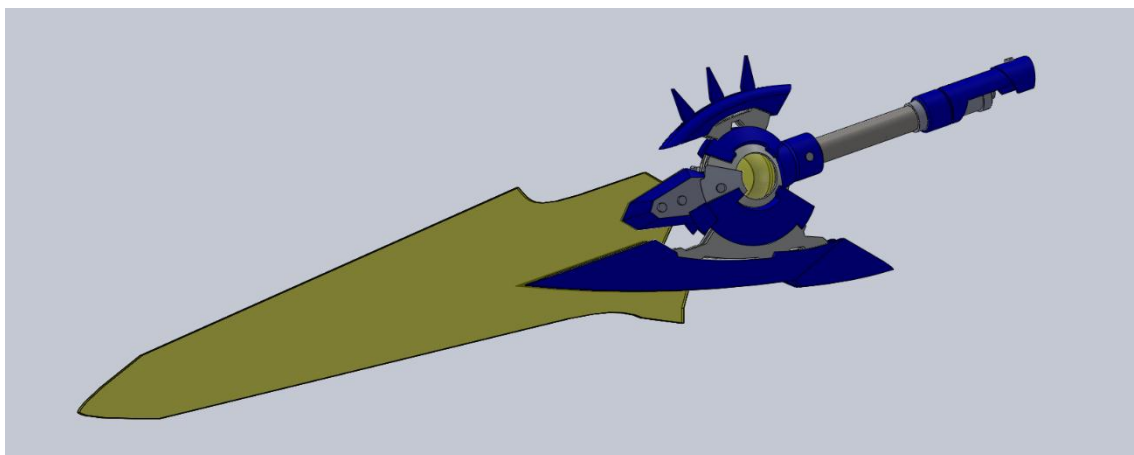


図 8 バルディッシュアサルト ライオットブレードⅡ (SolidWorks)

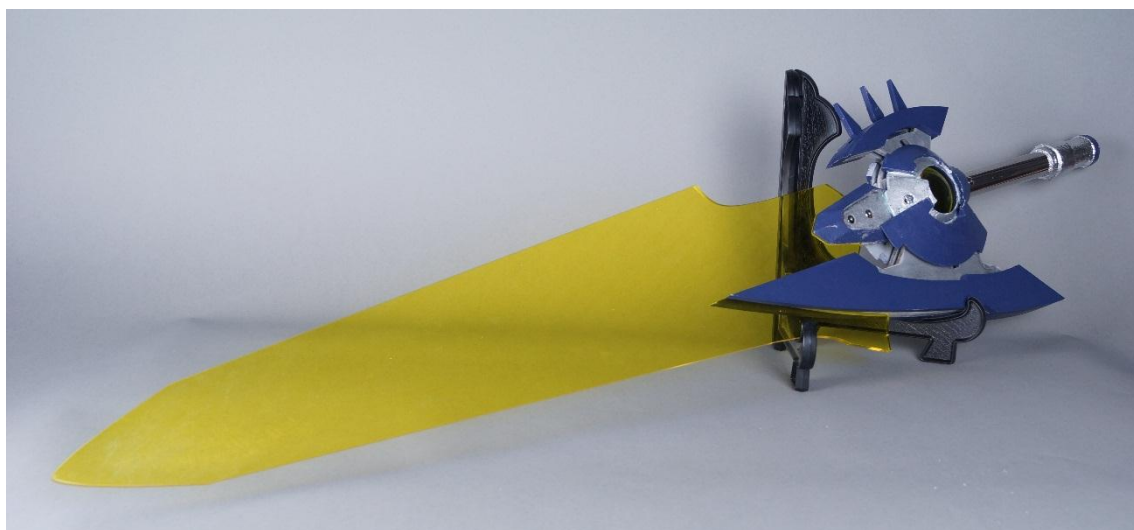


図 9 バルディッシュアサルト ライオットブレードⅡ (製作物)

SolidWorks を用いてゼロからモデリングをし (図 9), 3D プリンタや CNC フライスを利用して部品を製作した. モデルの作成に当たっては「緋賀ゆかり 魔法戦記 リリカルなのは Force 画集」 [15]等を参考にしている. なお, サイズやマテリアルの関係で, アクリル部分は購入時にレーザーカットを, フレーム部分は矢上キャンパスの実習室 [16]に図面を送り, アルミのワイヤーカット加工での製作を依頼した. 部品点数は組み立てに利用した組みネジを含め 50 点であるが, SolidWorks を用いたことで, 部品ごとのモデリングやアセンブリを容易に行うことができた.

## 2.3.4 製作を通じて

改めて，3 点の製作の際に利用したデータや工作機械について表 1 に整理する．

|       | 素材                 | データ・ソフト                | 工作機械                | 電子回路                |
|-------|--------------------|------------------------|---------------------|---------------------|
| 2.3.1 | 紙<br>合皮            | 3D / 2D CAD<br>フリーフォント | カッティングプロッタ          |                     |
| 2.3.2 | アクリル<br>テープ LED    | 2D CAD                 | レーザーカッター            | Arduino FIO<br>XBee |
| 2.3.3 | ABS<br>アクリル<br>アルミ | 3D CAD                 | 3D プリンタ<br>CNC フライス |                     |

表 1 製作に使用したもの

表に挙げ切れていないものもあるが，2D/3D の CAD，Arduino や電子回路，各種工作機械のノウハウといったものを習得したことに加え，その過程において，他人の公開したデータを利用したり，他者の作品に影響を受けたり，製作過程などを Web に公開したりといった，オープンソースやデジタルファブリケーションの一連の流れを自然と体験していたといえることができる．

## 第3章 研究背景

### 3.1 オープンソース

コンピュータの分野においては，Linux のような低レイヤーの分野から，WordPress のようなアプリケーションにいたるまで，数多くのソフトウェアが OSS(Open Source Software)として以前から公開されており，中でも Web テクノロジーの多くは OSS によって支えられている．

また，現在ではソフトウェアに限らず，電子機器の設計図や回路図，ファームウェア，ソフトウェアなどが公開され OSHW(Open Source HardWare)とも呼ばれている．代表的なのが AVR マイコンの Arduino [17]であり，Lilypad [18]はじめ数多くの派生系(クローン)が存在している．

### 3.2 デジタルファブリケーション

デジタルデータとコンピュータ制御の工作機械を用いることによって，手作業では困難な様々な加工を行うことが可能となった．3D プリンタや CNC フライス，レーザーカッターといった工作機械がその代表例であるが，これらの登場によって，必要なときに必要なだけ（オンデマンドで）生産することが可能となった．なおデジタルファブリケーションについては「FabLife —デジタルファブリケーションから生まれる「つくりかたの未来」」 [19]や，「「ものづくり」のベンディング」 [20]などが詳しい．

#### 3.2.1 CAD/CAM

デジタルファブリケーションにおけるハードルとして CAD と CAM の 2 つが挙げられる．以下にデジタル工作機械を使う際の大まかな流れを示す．



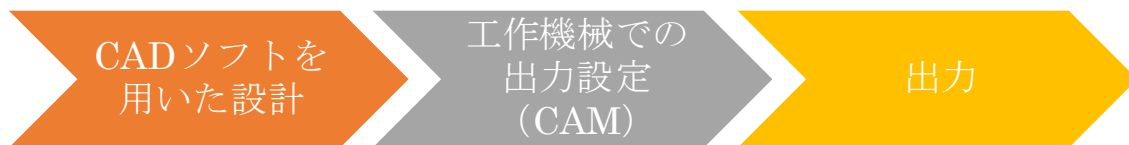


図 10 デジタルファブリケーションの流れ

まず初めに大きなハードルとなるのが、工作機械を使って加工・出力するもののデータを作成する工程である。2D/3D、有償/無償を問わず多くの CAD ソフトウェアが存在するが、それらを用いて自身の作りたいもののデータを作成するリテラシーを持った人々はまだまだ少なく、直感的に利用できる CAD ソフトも現在開発途上である。

次に、出来上がったデータを実際に機械で出力するために各種設定を行い、G コードや NC コードと呼ばれる工作機械を制御するデータを作成する CAM と呼ばれる工程がある。CAD で作成したデータの拡大縮小や、加工対象物における位置合わせといった操作は比較的容易に行うことが可能であるが、加工する材料にあわせてパラメータを変更したり、複雑な加工を行う場合にはさらに詳細な設定が必要となり、CAD ソフト同様に習熟には時間がかかる。

また、CAD/CAM 共にさまざまな種類のソフトウェアが存在するため、利用する工作機械ごとに別々のソフトウェアの使い方を習得しなければならないといった問題点も存在している。そのため、さまざまな工作機械を同一のソフトウェアで利用可能にすべく、FabModule [21]と呼ばれるソフトウェアの研究・開発が MIT によってなされている。

一方で、廉価な 3D プリンタ等の普及により Thingiverse [22]やブログにおいて個人の製作物およびデータが公開され、それらをダウンロードしてそのまま出力するといったことも行われ始めている。Web 上に公開された様々なデータをインターネット上から取得することにより、CAD の工程を踏まずとも自分のほしいと思っ



た様々なものを工作機械で出力することが可能となるのである。

### 3.2.2 アパレル CAD

CAD の中でも服飾に特化したソフトウェアとしてアパレル CAD と呼ばれるものが存在する。型紙をデザインするパターンメイキング，標準寸法の型紙を元にサイズを修正するグレーディング，布の無駄がないように面付けするマーキングなど服飾のデザインのための機能がそろっている。ただ，実際に講習会に参加して利用したところ，CAD としての機能は数世代前の 2D CAD と差がないようにも見受けられた。また，データが独自フォーマットで，ソフト自体が高価なこともあり，ネット上に個人がアパレル CAD を用いて作成したと思われる型紙のデータを公開しているといったケースは見当たらず，個人が服を作る際に CAD が利用されるといったことは現状ではほぼ行われていないようである。個人で利用するには高価であり，習得には他の CAD と同様に時間がかかり，アパレル CAD のオペレーターが専門職として成り立っているのが現状である。

### 3.2.3 Grasshopper

3D CAD ソフトの Rhinoceros には，Grasshopper というプラグインが存在する。ブロック状のオブジェクト同士を線で結ぶことでモデリングが行え，オブジェクトのパラメータに変数を設定しておくことで，辺の長さなどを後から変更することが可能なパラメトリックモデリングを行えるのが特徴である。ただ，習得に時間がかかることや，データ作成時のコストが増加すること，バージョン間の互換性が保障されていないことなどが課題である。

## 第4章 設計

### 4.1 問題発見

第2章で述べたように，他者の公開していた素材や，自身で作成したデータを元に，デジタル工作機械を用いることで，コスプレに必要な様々な造形物を製作することができた．一方でコスプレに必要な衣装の製作は容易ではないことが明らかとなった．Web上の，無償で型紙データを公開している個人サイト，有償で販売しているサイト共に，型紙のデータに以下のような問題点が存在していたためである．

- 家庭用のプリンタで出力する前提で A4 サイズに分割されている．
- ベクターデータではないため編集が行えない．
- サイズのバリエーションが少ない．

これらは，そもそもデジタル工作機械で型紙データを利用する事を想定していなかったために起こるものである．元のデータを作成する際には何らかの CAD ソフトを用いているため，配布するフォーマットを編集可能なベクター形式のデータにすることは極めて容易なはずだ．しかしながら，仮に型紙が編集可能なベクターデータとして公開されていたとしても，自分の体型にあうように型紙データにサイズの修正を加える必要性が生じることが考えられる．他人がネット上に公開していた型紙データをダウンロードしてきて，そのままそれを利用して衣装を作ったとしても自分の体型には当然合わないはずである．有償サイトであれば S/M/L 程度のバリエーションは揃っているが，それでも自分自身の体型に合う型紙として編集なしで利用するには十分ではないだろう．

### 4.2 目的

4.1 節で述べたような問題点を解決し，個人のブログや Thingiverse のような投稿サイトからダウンロードしてきた型紙データを利用して，衣服や靴などを，デジタル工作機械を用いて製作することを可能とする事が本論分の目的である．

## 4.3 機能要件

本研究における問題点を解決するための機能要件は以下の通りである．

1. ベクターデータの読み書きが可能であること．
2. CAD ソフトのように複雑な操作を必要としないこと．
3. 自分の体系にあうよう型紙データを容易に修正(グレーディング)できること．

## 4.4 関連研究

3D スキャンデータをもとに，自分にぴったり合ったジーンズを探し出してくれる「Bodymetrics」 [23]という技術がすでに開発され，実際にデパートなどで利用されている．この技術を用いれば，店内の商品から自分の体系にもっとも合ったものを簡単に見つけ出したり，ネットショップで買った商品のサイズが合わずに返品するといったことが避けられるとされているが，いずれの場合においても既にある商品から選ぶことに変わりはなく，気に入ったデザインのものを見つけたのに自分に合ったサイズがないなどといった場合の解決策とはならない．

## 第5章 実装

前章で述べた目的や要件を実現するためのアプローチとして、Microsoft が XBOX のコントローラーとして販売している Kinect を用いる。Kinect の 3D スキャンデータを元に型紙の修正に必要な身長やウエストなどの体型データを取得し、それらの値を利用して自動的に型紙データのリサイズを行うことを可能とするソフトウェアの実装を行った。

### 5.1 実装環境

実装環境は以下のとおりである

- Windows 7 64bit
- Processing 2.0 Beta 6 [24]

ライブラリ

- SimpleOpenNI [25]
- ControlP5 [26]

なお、Windows 環境下において SimpleOpenNI を利用するための環境構築については「日経ソフトウェア 2011 年 8 月号：初めての Kinect プログラミング 第 1 回」 [27]を参考にした。

### 5.2 プロトタイプの実装

#### 5.2.1 概要

前章で述べたような機能を有するソフトウェアを作成するに当たり、まず初めに着物の型紙データを体型に合わせて変更するプロトタイプを作成し、以下の機能を実装した。

1. SimpleOpenNI で得られる骨格データを利用して身長・ウエスト・腕の長さの各パラメータを算出するアルゴリズムの実装.
2. 型紙の各線分の長さに算出したパラメータを関連付け, リアルタイムで反映させる機能の実装.

なお, 型紙データに着物を利用したのは直線が多いことや, 寸法が規定されている事が理由である. また, 各パラメータは ControlP5 を用いてマニュアルでの修正も可能となるよう実装した.

## 5.2.2 パラメータの算出アルゴリズム

SimpleOpenNI 0.27 では一部の関節は 3 次元座標が取得できず, 原点座標を返す仕様が確認された. また, SimpleOpenNI で取得できるのは 3 次元座標のみであるため, 身長・ウエスト・腕の長さはそれぞれ以下で述べるのアルゴリズムで計算している. なお, 以下で「SKEL\_~」と表記しているのは SimpleOpenNI における関節名, それ以外は 3 次元座標を格納した変数名である.

### 5.2.2.1 身長

SKEL\_HEAD では頭の中央の座標しか取得できないため, userMap[] で人として検地したエリア (図 10 において緑色となっている部分) の一番上の点(top とした) から左右の足の中央 (footC) の座標までの距離を身長とした.

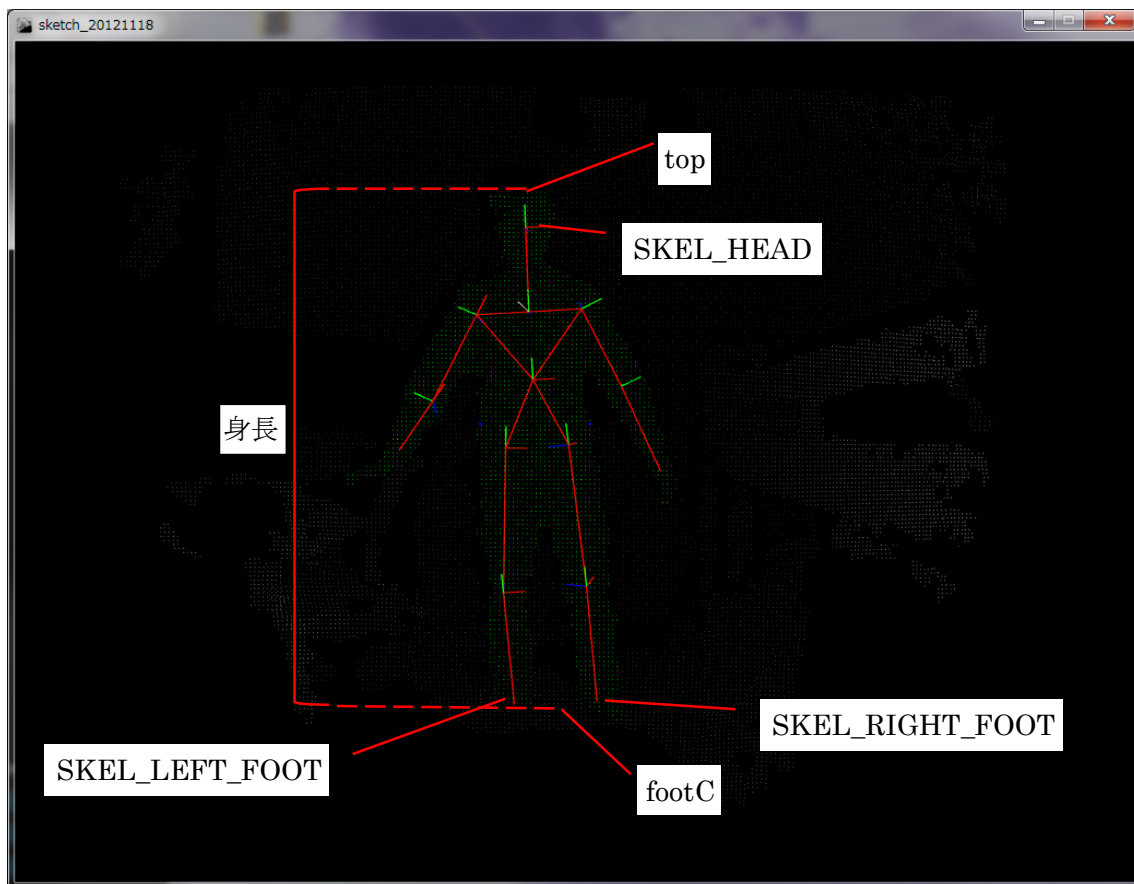
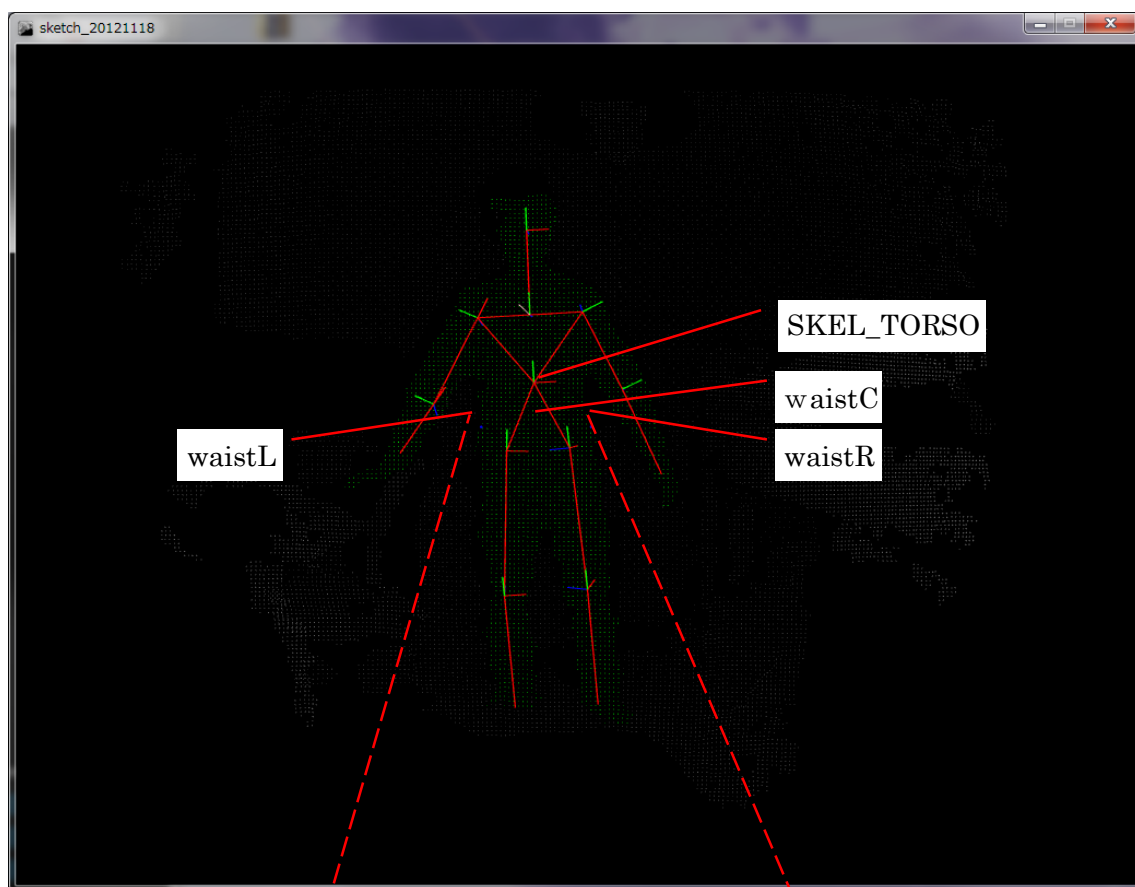


図 11 身長の算出方法

### 5.2.2.2 ウエスト

まず、ウエストの高さを左右のヒップの中心と胴の中間（waistC）とし、そこから、userMap で人として検地された左右の端をそれぞれ waistR, waistL とした。waistR-waistL 間を 10 ドット間隔で距離を取り、その和の 2 倍をウエストとした。なお、同様のアルゴリズムを用いることで、バストやウエストについても計測することが可能である。また、より厳密に測定する場合は、前後両側からスキャンすることが望ましい。



断面図

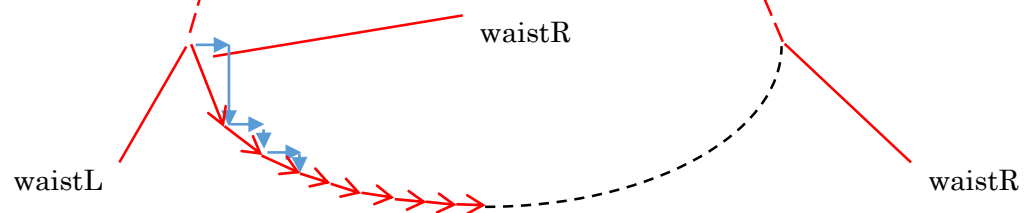


図 12 ウエストの算出方法

### 5.2.2.3 腕の長さ

左右の肩の中心(shoulderC), 右肩, 右ひじ, 右手の距離の和を腕の長さとした. 腕と同様に, 関節の距離を用いることで足の長さなども測定可能である.

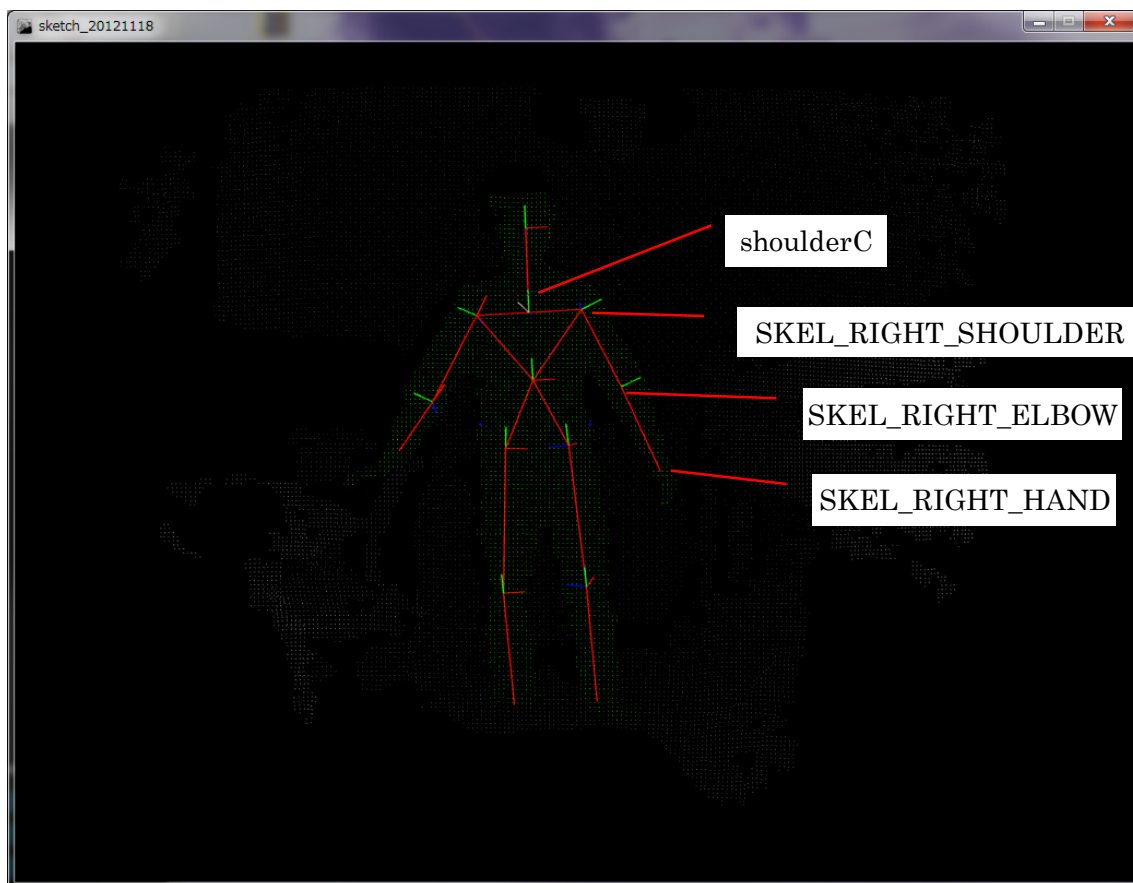


図 13 腕の長さの算出方法

### 5.2.2.4 体型スキャン時の姿勢

以上のアルゴリズムによって体型のスキャンを行うため, スキャン時には頭から足までの全身が映るようにまっすぐ立ち, 手を左右に大きく広げた姿勢でスキャンを行うことが求められる.



## 5.3 プロトタイプの展示

Open Research Forum 2012 [28]において作成したプログラムを実際に体験できる形で展示し、質問やコメントを頂いた。その内容は主に以下のとおりである。

- 任意の型紙でも同じことができるのか。
- Kinect を利用する必然性があるのか。
- Kinect の精度ではスーツの仕立てなどは困難ではないか。
- 既存のアパレル CAD のグレーディング機能ではいけないのか。
- 立体縫製なども可能になると素晴らしい。

## 5.4 改良版の実装

ORF2012 でのフィードバックを元に、プロトタイプにおいて実装した体型の各パラメータを算出するアルゴリズムと、線分にパラメータを関連付ける機能を、任意の DXF ファイル形式の型紙データにおいても利用できるよう改良を加え、CAD ソフトとしての基本機能についても実装を行った。なお、DXF ファイルの読み込みモジュールに関しては指導教員である田中浩也准教授から頂いたプログラムを元に一部を変更して用いており、DXF R12 形式 (Rhino 3D から DXF R12 直線&円弧にて出力したものを推奨) のデータを読み込むことが可能となっている。

### 5.4.1 DXF ファイルの読み込みとエンティティの扱い

DXF R12 形式のファイルから、LINE, ARC, CIRCLE の各エンティティを読み込む度に、lineType[], x1[], y1[], x2[], y2[], arcR[], arcSA[], arcEA[] の各配列を 1 つ拡張し、以下のように値を格納した。

#### 5.4.1.1 LINE

lineType[] に 1 を、始点の座標を x1[], y1[] に、終点の座標を x2[], y2[] に格納し、

未使用の `arcR[]`, `arcSA[]`, `arcEA[]`については 0 とした.

### 5.4.1.2 ARC

`lineType[]`に 2 を, 中心の座標を `x1[]`, `y1[]`に格納し, 直径の値を `arcR[]`に, 始点の角度を `arcSA[]`, 終点の角度を `arcEA[]`にラジアンで格納し, 未使用の `x2[]`, `y2[]`は 0 とした.

### 5.4.1.3 CIRCLE

`lineType[]`に 3 を, 中心の座標を `x1[]`, `y1[]`に, 直径を `arcR[]`に格納し, 未使用の `x2[]`, `y2[]`, `arcSA[]`, `arcEA[]`は 0 とした.

#### ソースコード 1 円弧の取り扱い

```
if (ent[i][1].contains("ARC")) {  
    float SA, EA;  
    SA = 360 - float(ent[i][17]);  
    EA = 360 - float(ent[i][15]);  
    if (SA > EA) {  
        EA += 360;  
    }  
    lineType=append(lineType, 2);  
    x1=append(x1, float(ent[i][7]));  
    y1=append(y1, -float(ent[i][9]));  
    x2=append(x2, 0);  
    y2=append(y2, 0);  
    arcR=append(arcR, 2*float(ent[i][13]));  
    arcSA=append(arcSA, radians(SA));  
    arcEA=append(arcEA, radians(EA));  
    moveType=append(moveType, 0);  
}
```

## 5.4.2 ズーム・ドラッグ

Processing の `translate` および `scale` を用いて実装した。ドラッグについては、累積の変異量を `dragX`, `Y` に、ドラッグ中の変異量を `dragBufX`, `Y` に格納している。また、マウス座標から視点変更前の座標に変換したマウス座標を `mouseTranslateX`, `Y` とし、以下のように計算した。

ソースコード 2 マウス座標の絶対値

```
mouseTranslateX=int((mouseX-width/2-dragX-dragBufX)/zoom+viewcenterX);  
mouseTranslateY=int((mouseY-height/2-dragY-dragBufY)/zoom-viewcenterY);
```

## 5.4.3 エンティティの選択

すべての線分・円弧・円について、マウスがそれらの上にある場合には太く描画するように実装した。クリックされた場合には選択したエンティティの `index` を `selectNum` に記録させると同時に、線分であれば長さを、円弧や円であれば半径を `Length` のスライダーの値に反映させている。

ソースコード 3 円弧の選択判定

```
if (lineType[i]==2) { //arc  
    if (i==selectNum || sq(arcR[i]/2-10)<sq(mouseTranslateX-x1[i])+sq(mouseTranslateY-y1[i])&&sq(mouseTranslateX-x1[i])+sq(mouseTranslateY-y1[i])<sq(arcR[i]/2+10)) {  
        strokeWeight(5);  
        if (mousePressed==true) {  
            selectNum=i;  
            selectFlag=true;  
            selectNow=true;  
            if (clickFlag==false) {  
                clickFlag=true;  
                originalLength=arcR[i];  
            }  
        }  
    }  
}
```

```

        for ( int n = 0; n < x1.length; n++) {
            moveType[n]=0;
        }
        controlP5.controller("Length").setValue(arcR[i]/2);
    }
}
else {
    clickFlag=false;
}
}
else {
    strokeWeight(1);
}
arc(x1[i], y1[i], arcR[i], arcR[i], arcSA[i], arcEA[i]);
}

```

#### 5.4.4 線分の伸縮時に移動する頂点の選択

線分の伸縮時に、始点・終点のいずれを移動させるかを選択できるようにし、datum には始点を移動させる場合には 0 を、終点を移動させる場合には 1 を設定した。なお、円弧や円の場合にはこの操作は発生しない。

##### ソースコード 4 移動する頂点の選択

```

if (lineType[selectNum]==1) { //line
    if (Length/2>sqrt(sq(mouseTranslateX-x1[selectNum])+sq(mouseTranslateY-
y1[selectNum]))) {
        strokeWeight(10);
        point(x1[selectNum], y1[selectNum]);
        if (mousePressed==true) {
            datum=0;
        }
    }
    if (Length/2>sqrt(sq(mouseTranslateX-x2[selectNum])+sq(mouseTranslateY-
y2[selectNum]))) {

```

```

strokeWeight(10);
point(x2[selectNum], y2[selectNum]);
if (mousePressed==true) {
    datum=1;
}
}
}

```

## 5.4.5 エンティティの伸縮

ControlP5 の Slider や Textfield でエンティティに加えた操作を，選択しているエンティティに以下のようなアルゴリズムで反映させる．

線分を伸縮させる場合は，始点－終点ベクトルの x 成分，y 成分それぞれを（新しい長さ／元の長さ）－1＝(newLength-originalLength)/originalLength 倍にした deltaX, deltaY を算出し，始点を移動する場合にはその-1 倍を，終点を移動する場合はそのまま始点ないし終点の x, y 座標に加える．

### ソースコード 5 線分の伸縮

```

if (lineType[selectNum]==1) { //line
    newLength=v;
    originalLength=sqrt(sq(x2[selectNum]-x1[selectNum])+sq(y2[selectNum]-y1[selectNum]));
    if (newLength<1.0) {
        newLength=1.0;
    }
    deltaX=(x2[selectNum]-x1[selectNum])*(newLength-originalLength)/originalLength;
    deltaY=(y2[selectNum]-y1[selectNum])*(newLength-originalLength)/originalLength;
    if (datum==0) {
        if (datum==0) {
            deltaX*=-1;
            deltaY*=-1;
        }
    }
}
}

```

```

    }
    x1[selectNum]+=deltaX;
    y1[selectNum]+=deltaY;
}
if (datum==1) {
    x2[selectNum]+=deltaX;
    y2[selectNum]+=deltaY;
}
Length=v;
}

```

また，円弧を伸縮させる場合には，直径  $\text{arcR}[]$  のみを修正してしまうと隣接する線分との位置関係に崩れが生じてしまうため，円弧の始点・終点それぞれにおける接線上に伸縮後の円弧の始点・終点がくるよう，中心の座標  $x1[]$ ， $y1[]$  についても修正を行うことで，下図のような動作を実現する．



図 14 円弧伸縮時の隣接エンティティの自動補正

まず，伸縮前の円弧の始点・終点および伸縮後の始点・終点の座標はそれぞれ

- $(x1+R*\cos SA, y1+R*\sin SA)$  .....①
- $(x1+R*\cos EA, y1+R*\sin EA)$  .....②
- $(x1'+R*\cos SA, y1'+R*\sin SA)$  .....③
- $(x1'+R*\cos EA, y1'+R*\sin EA)$  .....④

と表すことができる．（配列の  $[]$  ならびに  $\text{arcR}$ ， $\text{arcSA}$ ， $\text{arcEA}$  を  $R$ ， $SA$ ， $EA$  とそれぞれ省略している）

伸縮前の円弧の始点・終点における接線は①, ②より

$$(x-x_1)*(x_1+R*\cos SA-x_1)+(y-y_1)*(y_1+R*\sin SA-y_1)=R^2$$

$$\Leftrightarrow (x-x_1)*(R*\cos SA)+(y-y_1)*(R*\sin SA)-R^2=0$$

$$\Leftrightarrow (x-x_1)*\cos SA+(y-y_1)*\sin SA-R=0$$

$$\Leftrightarrow \cos SA*x+\sin SA*y-x_1*\cos SA-y_1*\sin SA-R=0 \quad \cdots \cdots \textcircled{5}$$

$$\text{および } \cos EA*x+\sin EA*y-R-x_1*\cos EA-y_1*\sin EA-R=0 \quad \cdots \cdots \textcircled{6}$$

という2式で表される.

⑤に③を, ⑥に④をそれぞれ代入して, 整理すると,

- $\cos SA*x_1'+\sin SA*y_1'+R'-x_1*\cos SA-y_1*\sin SA-R=0$
- $\cos EA*x_1'+\sin EA*y_1'+R'-x_1*\cos EA-y_1*\sin EA-R=0$

となる.

この二式を  $a*x+b*y+c=0$ ,  $d*x+e*y+f=0$  とおくと,  $x=(b*f-c*e)/(a*e-b*d)$ ,  $y=(c*d-a*f)/(a*e-b*d)$  となり, これを解くことで,  $x_1'$ ,  $y_1'$ を求めることができ, これが円弧伸縮後の中心点の座標である.

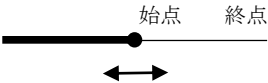
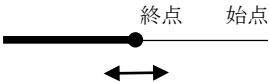
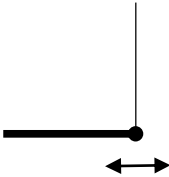
## ソースコード 6 円弧の伸縮

```
if (lineType[selectNum]==2) {//arc
    newLength=v;
    strokeWidth(2);
    float a=cos(arcSA[selectNum]);
    float b=sin(arcSA[selectNum]);
    float c=newLength-x1[selectNum]*cos(arcSA[selectNum])-
y1[selectNum]*sin(arcSA[selectNum])-arcR[selectNum]/2;
    float d=cos(arcEA[selectNum]);
    float e=sin(arcEA[selectNum]);
    float f=newLength-x1[selectNum]*cos(arcEA[selectNum])-
```

```
y1[selectNum]*sin(arcEA[selectNum])-arcR[selectNum]/2;
    xTemp=(b*f-c*e)/(a*e-b*d);
    yTemp=(c*d-a*f)/(a*e-b*d);
    x1[selectNum]=xTemp;
    y1[selectNum]=yTemp;
    arcR[selectNum]=v*2;
    Length=v*2;
}
```

5.4.6 隣接するエンティティの自動修正

型紙のデータはエンティティ同士が接している場合や、水平な線分が存在している場合が多く、複数の線分に似た処理を行うといったことが頻繁に起こりうる。また、ある線分の長さを変更すると、隣接する他の線分との位置関係を修正する手間なども発生する。エンティティを伸縮した際に、隣接するエンティティの頂点や、水平・垂直位置に存在するエンティティに対しても頂点の座標を自動で修正することで、型紙データを編集する際のコストの削減が図れる。選択したエンティティと、自動修正を行うエンティティの位置関係の組み合わせは以下の表に示すとおりである。

|                                                                                                                                                        |                                                                                                                                                        |                                                                                                                                                 |
|--------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <div>1</div> <div></div> <div>線分の伸縮時に、移動する頂点と隣接する線分の始点が一致する場合</div> | <div>2</div> <div></div> <div>線分の伸縮時に、移動する頂点と隣接する線分の終点が一致する場合</div> | <div>3</div> <div></div> <div>伸縮する線分と、隣接する線分が垂直である場合</div> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|



|                                                |                                                |                                                |
|------------------------------------------------|------------------------------------------------|------------------------------------------------|
| <p>4</p> <p>円弧の伸縮時に、円弧の始点と隣接する線分の始点が一致する場合</p> | <p>5</p> <p>円弧の伸縮時に、円弧の始点と隣接する線分の終点が一致する場合</p> | <p>6</p> <p>円弧の伸縮時に、円弧の終点と隣接する線分の始点が一致する場合</p> |
| <p>7</p> <p>円弧の伸縮時に、円弧の終点と隣接する線分の終点が一致する場合</p> | <p>8</p> <p>円弧の接線上に、伸縮する線分の移動する頂点が存在する場合</p>   | <p>9</p> <p>伸縮する線分の頂点と円弧の始点または終点が一一致する場合.</p>  |

表 2 隣接するエンティティと自動修正を行うエンティティの組み合わせ

線分を選択するたびに、選択された線分を除く全てのエンティティをチェックし、表 2 の各セル左上に付した番号を `moveType[]` に設定している。 `moveType[]` が 0 以外の場合に選択中のエンティティの伸縮が行われた場合には、以下の表に示すとおりエンティティの始点、終点、中心のいずれかに対して座標の修正を施す。

| <code>moveType[]</code> | 座標の修正 (i はエンティティのインデックス,)                                                                                        |
|-------------------------|------------------------------------------------------------------------------------------------------------------|
| 1                       | $x1[i] += \text{deltaX};$<br>$y1[i] += \text{deltaY};$                                                           |
| 2                       | $x2[i] += \text{deltaX};$<br>$y2[i] += \text{deltaY};$                                                           |
| 3                       | $x1[i] += \text{deltaX};$<br>$y1[i] += \text{deltaY};$<br>$x2[i] += \text{deltaX};$<br>$y2[i] += \text{deltaY};$ |
| 4                       | $x1[i] = x\text{Temp} + \text{newLength} * \cos(\text{arcSA}[\text{selectNum}]);$                                |

|   |                                                                                              |
|---|----------------------------------------------------------------------------------------------|
|   | y1[i]=yTemp+newLength*sin(arcSA[selectNum]);                                                 |
| 5 | x2[i]=xTemp+newLength*cos(arcSA[selectNum]);<br>y2[i]=yTemp+newLength*sin(arcSA[selectNum]); |
| 6 | x1[i]=xTemp+newLength*cos(arcEA[selectNum]);<br>y1[i]=yTemp+newLength*sin(arcEA[selectNum]); |
| 7 | x2[i]=xTemp+newLength*cos(arcEA[selectNum]);<br>y2[i]=yTemp+newLength*sin(arcEA[selectNum]); |
| 8 | x1[i]+=deltaX;<br>y1[i]+=deltaY;                                                             |
| 9 | x1[i]+=deltaX;<br>y1[i]+=deltaY;                                                             |

表 3 moveType 毎の座標の修正

### 5.4.7 Kinect でスキャンし，算出した値の適応

ControlP5 の Textfield に入力された文字列を数式として処理する際に，Kinect のスキャンを元に算出した height, waist, arm の値を変数として利用できるような実装を行う．なお，今回は入力された文字列を+\*/の記号で分割し，左から順に計算を行うだけの実装のため，括弧や計算の優先順位などは考慮していない．

ソースコード 7 数式処理

```
public void Expression(String v) {
    v=v.toLowerCase();
    String[] separate=splitTokens(v, "+*/");
    String[] operation=splitTokens(v, "0123456789abcdefghijklmnopqrstuvwxyz.");
    float[] num= {
    };
    for (int i=0;i<separate.length;i++) {
        if (separate[i].equals("height")) {
            num=append(num, Height);
        }
        else if (separate[i].equals("waist")) {
            num=append(num, Waist);
        }
    }
}
```

```

    }
    else if (separate[i].equals("arm")) {
        num=append(num, Arm);
    }
    else {
        num=append(num, float(separate[i]));
    }
}
newLength=num[0];
for (int i=0;i<operation.length;i++) {
    if (operation[i].equals("+")) {
        newLength+=num[i+1];
    }
    else if (operation[i].equals("-")) {
        newLength-=num[i+1];
    }
    else if (operation[i].equals("*")) {
        newLength*=num[i+1];
    }
    else if (operation[i].equals("/")) {
        newLength/=num[i+1];
    }
}
controlP5.controller("Length").setValue(newLength);
}

```

## 第6章 評価

### 6.1 評価方針

実装したソフトウェアを用いることで、他人の作成した型紙のデータを元に自分の体型にあった型紙へ容易に編集することが可能となり、型紙データの再利用性が向上したかを評価する。

#### 6.1.1 定量評価

以下の二通りの方法でデータを修正するのに要した時間を計測し、比較する。

1. メジャーで採寸をし、Adobe Illustrator で編集する方法
2. 今回作成したシステムを使い、Kinect で採寸をし、Processing で編集する方法

#### 6.1.2 定性評価

評価実験を行ったユーザーに対して、アンケートを行い、

- 実装したソフトウェアによって、型紙データの再利用性が向上したとを感じるか
- Adobe Illustrator と比較してどちらのほうがデータの修正が容易であったか
- Kinect による体型のスキャンは有効だと感じたか。

といった点について評価を行う。

### 6.2 評価結果

SFC の学部生 16 名（男性 10 名，女性 6 名）に対して評価実験を行い、以下のような結果を得た。

### 6.2.1 所要時間

二通りそれぞれの方法で型紙データを編集するのに要した時間を図 15 に示す．平均では Adobe Illustrator が 8:13 秒，今回 Processing で実装したソフトウェアが 3:03 秒であり，型紙を編集するのにかかる時間は 63%短縮されている．なお，作業内容の理解度が二度目に行った方法の場合に高くなり，作業時間も短縮される事が予想されたため，Adobe Illustrator から始めるグループ，Processing 側から始めるグループそれぞれ 8 名ずつとして評価実験を行った．Adobe Illustrator では，線の長さを修正するごとに他の線についても位置を調整する必要があるため，このように大きな差が生じる結果となったと考えられる．

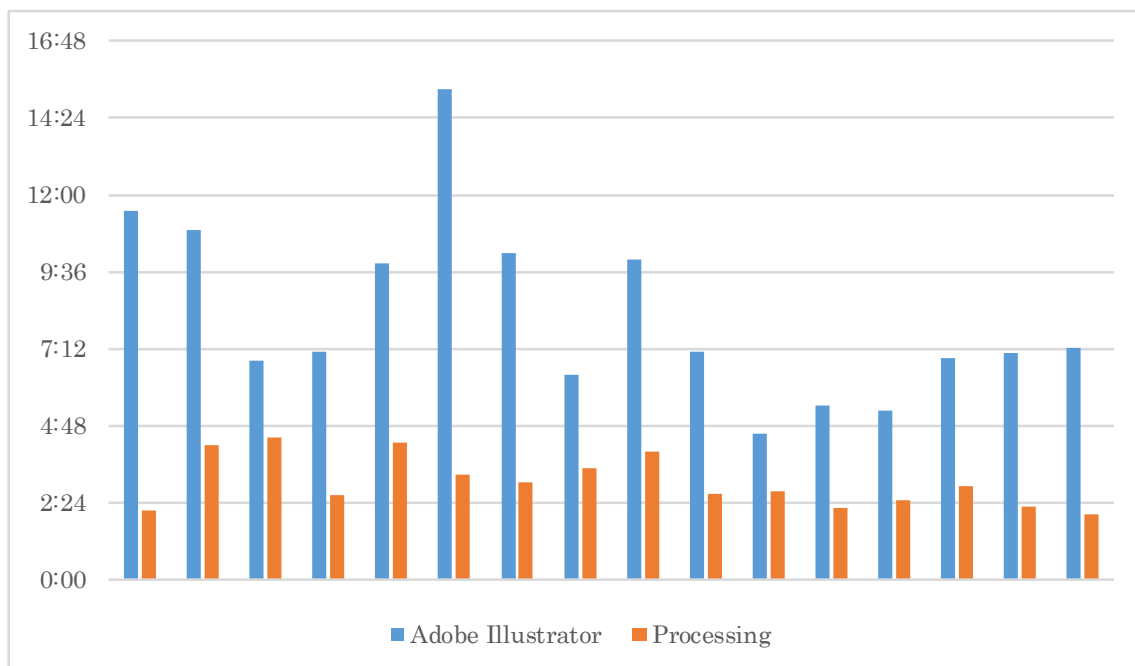


図 15 所要時間

### 6.2.2 編集の容易さ

編集が容易だったのはどちらかを「Illustrator」，「どちらかといえば Illustrator」，「どちらかといえば Processing」，「Processing」の 4 段階でたずねた所，16 名全員が「Processing」を選ぶ結果となった．なお，16 名の被験者のうち 10 名は Adobe Illustrator の仕様経験があるとしており，日ごろからよく Illustrator を利用して

いる被験者からは慣れた環境でないため多少戸惑ったとの声も見受けられた。また、Ctrl+Z での「元に戻す」機能や、Ctrl ないし Shift を押したままクリックすることによる複数選択の機能の有無を 3 名ほどから訊ねられた。

### 6.2.3 再利用性の向上

Processing で実装したソフトウェアを用いることで、型紙データの再利用性が向上したとを感じるか（口頭で「今回実装したようなソフトウェアがあれば、他人の作成したデータを自分で修正して利用できると思うか」と補足）を「はい」、「どちらかといえばはい」、「どちらかといえばいいえ」、「いいえ」の 4 段階でたずねた結果、11 名が「はい」、5 名が「どちらかといえばはい」と回答した。型紙データが蓄積された web サービスがあればよりデータの再利用性があがるとの声があった一方、他人が作成したデータをダウンロードしてきて利用するという行為のイメージがつかめないという被験者も一名存在した。

### 6.2.4 Kinect によるスキヤンの有効性

Kinect を用いて体系をスキヤンするというアプローチについて有効だと感じたかについても同様に 4 段階でたずねた結果、「はい」が 10 名、「どちらかといえばはい」が 6 名であったが、Kinect によるスキヤンの精度に関しては疑問視する声もあり、ライブラリや算出アルゴリズムの改良が必要である。

### 6.2.5 まとめ

今回実装したソフトウェアを用いることで、誰もが容易に型紙データのサイズを修正することが可能となり、型紙データの再利用性を向上させることができた。ただ、今回の評価実験において用いた着物の場合には、各辺の長さが予め決められているのに対し、洋服の型紙などの場合ではどの辺をどの程度修正すればよいかなど、服飾の知識が必要とされることが考えらる。

また，評価実験の際に複数回見られた間違いとしては，5.5.4 の頂点の選択について意味が理解されていない場合や，頂点をクリックしようとして他のエンティティが選択されてしまうといった事が見受けられた．この点についてはインターフェースや実装を改良する余地が残っていると考えられる．

## 第7章 結論

### 7.1 本研究のまとめ

本論文は，コスプレに関するほぼ全てのものをデジタルファブ리케이션によって作り出そうとした著者の高校時代から続く創作活動と，2年間の研究の記録である．今後より一層デジタル工作機械が普及し，誰もが簡単に使うことのできるCADソフトウェアの開発が進めば，私が行ってきたようなデジタルファブ리케이션のコスプレや同人活動における活用は一層進むことは間違いないだろう．

アニメやゲームといった日本の誇る優れたコンテンツをただ消費するだけだった多くの人々が，ニコニコ動画や初音ミクの登場によってクリエイターとなった．MikuMikuDance [29]がキラアアプリケーションとなって多くの優れた作品が生み出されたように，メイカームーブメントにおいても今後たくさんのソフトウェア/ハードウェアが登場してくるだろう．そして，多くの人々が形あるものを生み出すメイカーとなって現れるようになった時，その流れの中心にあるのは若い世代であり，オタクやギークと呼ばれる人々の情熱であるに違いない．



# 参考文献

1. 魔法少女リリカルなのは A's. 2004 年. <http://www.nanoha.com/>
2. デバイス設計図 1. 有害毒電波, 2006 年
3. 塗夢. Device 3D Image. Ver2nd. 1 Art 空間, 2007 年
4. コスパティオ. <http://cospatio.com/>
5. 矢野経済研究所. 「オタク」市場に関する調査結果 2012. 2012 年  
<http://www.yano.co.jp/press/pdf/1002.pdf>
6. 杉浦由美子. コスプレ女子の時代. ベストセラーズ, 2008
7. 成実弘至. コスプレする社会—サブカルチャーの身体文化. せりか書房, 2009
8. コスプレイヤーズアーカイブ. <http://www.cosp.jp/>
9. Cure. <http://ja.curecos.com/>
10. ニコニコ動画. <http://www.nicovideo.jp/>
11. ミッドチルダフォント置き場. <http://midfont.refy.net/>
12. コスプレショップみつば堂. <http://www.mitubado.com/>
13. 防御魔法「シールド」を発動してみた  
<http://www.nicovideo.jp/watch/sm4390881>
14. 株式会社ハナミ. [www.sign-company.jp/](http://www.sign-company.jp/)
15. 緋賀ゆかり. 緋賀ゆかり 魔法戦記リリカルなのは Force 画集.: 角川書店, 2012
16. 機械系共通実験室. <http://www.mex.mech.keio.ac.jp/>
17. Arduino. <http://www.arduino.cc/>
18. LilyPad Arduino. <http://arduino.cc/en/Main/ArduinoBoardLilyPad>
19. 田中浩也. FabLife —デジタルファブリケーションから生まれる「つくrikata」の未来—. オライリージャパン, 2012

20. 廣瀬悠一. 「ものづくり」のベンディング. 慶應義塾大学環境情報学部 卒業論文, 2012
21. Fab Modules. <http://kokompe.cba.mit.edu/>
22. Thingiverse. <http://www.thingiverse.com/>
23. Bodymetrics. <http://www.bodymetrics.com/>
24. Processing.org. <http://www.processing.org/>
25. simple-openni. <http://code.google.com/p/simple-openni/>
26. ControlP5. <http://www.sojamo.de/libraries/controlP5/>
27. 日経ソフトウェア. 笹尾和宏. 8月号, 日経 BP 社, 2011 年
28. Open Research Forum 2012. <http://orf.sfc.keio.ac.jp/>
29. Vocaloid Promotion Video Project. <http://www.geocities.jp/higuchuu4/>

# 謝辞

本卒業論文を執筆するにあたり、多くの方にご指導やご助力を賜りました。

慶應義塾大学環境情報学部准教授 田中 浩也博士には二年間に渡って多大なご指導頂き、本卒業論文の執筆にあたっても多くの助言や、プログラムの面でもサポートをして頂きました。そして、自由に研究をさせて頂いたことを本当に感謝しております。また、環境情報学部専任講師 水野 大二郎博士にも様々な面でご指導を賜りました。ありがとうございました。

SFC での四年間の生活においては多くの先生や先輩方にお世話になりました。なかでも講義や研究会の場において多くのご指導をいただきました環境情報学部教授の徳田 英幸博士，同学部教授 村井 純博士，同学部教授中村 修博士，同学部教授 武藤 佳恭博士，同学部准教授 高汐 一紀博士，同学部准教授 筧 康明博士，同学部准教授 三次 仁博士ならびに，徳田英幸研究室においてお世話になった小川 正幹氏，唐津 豊氏，天野 雅哉氏には特に感謝いたします。

田中浩也研究室の全てのメンバーに感謝します。プログラム面での相談にのって頂いた大嶋 泰介氏，藤吉 賢氏や，同期の升森 敦士氏，富中 裕介氏，中村 優介氏，また Fablab Kamakura の早稻田 治慶氏，渡辺 ゆうか氏，守屋 英義氏をはじめ，多くの方に刺激や助言を頂き，研究面やその他の面でも大変お世話になりました。ありがとうございました。

各種加工の際にお世話になった，ものづくり工房の柿崎 勇晃氏や理工学部実験教育支援センター機械系共通実験室の斉田 尚彦氏，東急ハンズのスタッフの方々やお世話になった多くの業者の皆様に感謝いたします。

UNREAL（旧称：SOS 団@SFC）の岡村 美奈氏，上田 智翔氏や，学内外の多くの先輩や友人たちに感謝します．

インストールマニアックスで出会った全ての参加者・関係者の皆様に感謝します．

日本マイクロソフト株式会社デベロッパー・プラットフォーム統括本部の平野和順氏，砂金 信一郎氏をはじめお世話になった皆様に感謝いたします．

インターンシップでお世話になったニフティ株式会社の鈴木 隆一氏，田代 光輝氏，佐古裕氏ならびに，関西学院大学文学部教授 三浦 麻子博士，関西大学総合情報学部教授 森尾 博昭博士，慶應義塾大学政策・メディア研究科特任講師 折田 明子博士に感謝いたします．

生活その他の面で支えてくれた両親 宇田 豊，恵子，愛犬 ミュウに心から感謝します．

浅野中学高等学校の同期に感謝します．特に宮川 朋大氏，松井 駿弘氏をはじめ，同人活動を共にした仲間感謝します．あの時の経験がなかったら私はこの論文を書いていなかったことでしょう．

同人活動を通じて出会った多くの素晴らしいサークルさん，コスプレイヤーさん，造形師さんに感謝します．

最後に，リリカルなのはを愛する全てのファンと関係者の皆様に感謝いたします．

# 付録

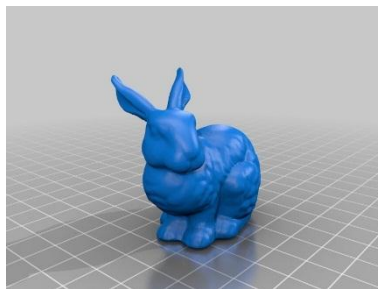
|               |     |
|---------------|-----|
| A) 評価手順書      | 2 枚 |
| B) 簡易説明書      | 1 枚 |
| C) 評価実験アンケート  | 1 枚 |
| D) 魔導書用カットデータ | 1 枚 |
| E) 製本用寸法      | 1 枚 |
| F) 魔法陣データ     | 1 枚 |
| G) アルミ加工図面    | 4 枚 |
| H) アクリルカット図面  | 1 枚 |

## 評価手順書(1/2)

はじめに：

3Dプリンタやレーザーカッターなどのデジタル工作機械の普及に伴い、デジタルファブリケーションと呼ばれる新しいモノづくりの手法が近年話題となっています。また、そういった工作機械で利用する2D／3Dのデータはインターネット上にオープンソースとして公開されており、専用の投稿サイトも存在しています。

たとえば3Dプリンタであれば下図のようなウサギの3Dデータをダウンロードしてきて、拡大縮小や出力位置を少し調整して、そのまま3Dプリンタで出力することが可能です。



<http://www.thingiverse.com/thing:11622>

ところが、服や靴などを作る際に使う型紙のデータでは、Webからダウンロードしてきたとしてもサイズを自分の体型に合わせてAdobe IllustratorやCADソフトなどを用いて修正する必要があります。このデータの再利用性が低い問題を解決しようというのが今回の評価実験のテーマです。

Adobe IllustratorとProcessingで作成したソフトウェアの2つの方法で簡単な型紙データのサイズを修正する工程を行ってもらい、その後簡単なアンケートへの協力をお願いします。

→ 裏へ

## 評価手順書(2/2)

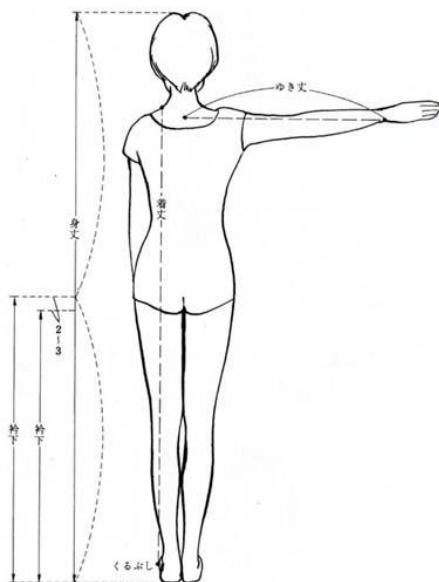
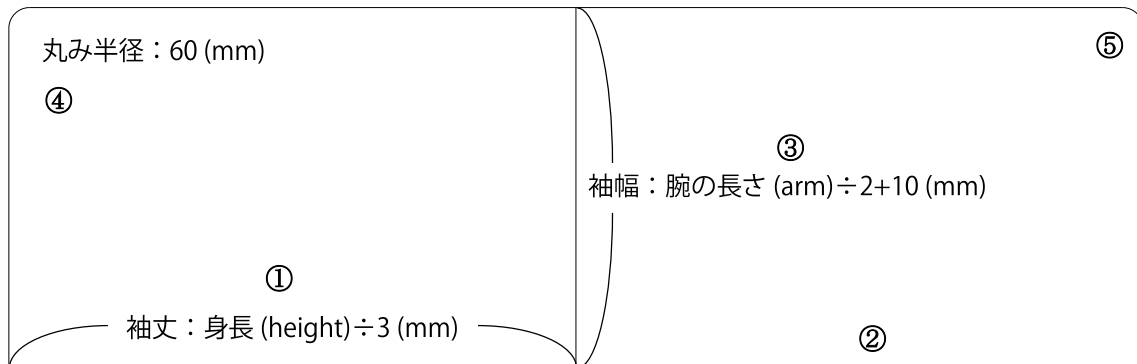
以下 2 通りの方法で着物の袖の部分の型紙データを編集してください。

1. Adobe Illustrator を開いて、下図のように型紙の寸法を修正してください。

なお、腕の長さは図のゆき丈の通りに採寸してください。

2. Processing 側においても同様に型紙の修正を行ってください。

こちらでは採寸の代わりに Kinect でスキャンしてください。

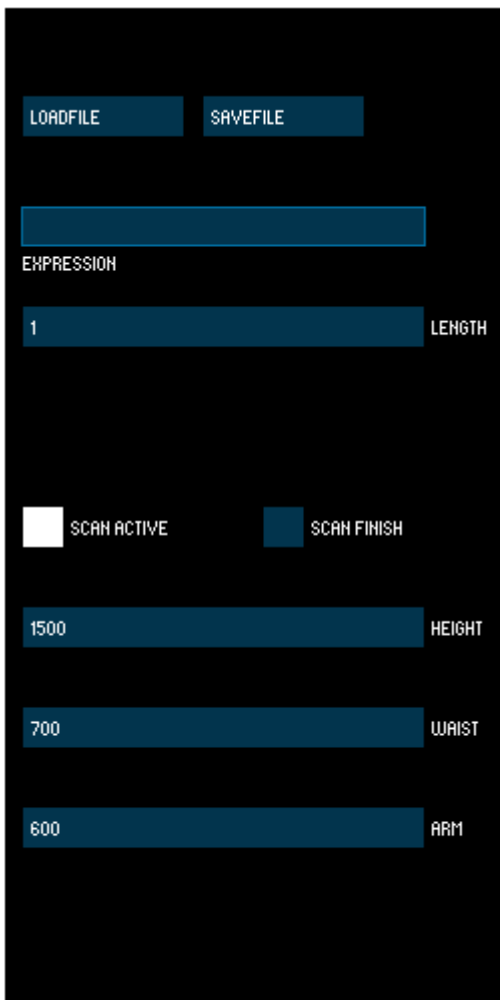


<http://www.kimono-bito.com/saisun.jpg>

## 簡易説明書

1. はじめに, Kinect の前に腕を広げて立ち, 体型をスキャンします.  
頭から脚までが画面に映り, 赤い骨格線が表示されたら **SCAN FINISH** をクリックします.
2. 長さを調整する線分をクリックします.
3. 選択した線分から伸縮する側の頂点を選択します.
4. ツールバーの **EXPRESSION** か **LENGTH** を使って選択した線分の長さを調整します.  
**EXPRESSION** には「 $\text{HEIGHT}/3$ 」, 「 $\text{ARM}/2+10$ 」, 「60」の様に数式を入力します.
5. 2～4 の操作を繰り返し, 必要な修正を加えます.

### ツールバーの機能



#### EXPRESSION :

選択した線分の長さを数式で入力します.

#### LENGTH :

線分の長さをスライダーで調整します.

#### SCAN :

体型スキャンが完了したら右のボタンを選択します

#### HEIGHT / WAIST / ARM :

身長, ウエスト, 腕の長さの値が表示されます.



## アンケート

氏名： \_\_\_\_\_

- Adobe Illustraotr を使用したことはありますか.

はい いいえ

|  |
|--|
|  |
|--|

- データの修正が容易であったのはどちらですか.

Illustrator どちらかといえば Illustrator どちらかといえば Processing Processing

|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|

- 本システムによって、型紙データの再利用性があがったと感じますか.

はい どちらかといえばはいどちらかといえばいいえ いいえ

|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|

- Kinect による体型のスキャンは有効だと感じましたか.

はい どちらかといえばはいどちらかといえばいいえ いいえ

|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|

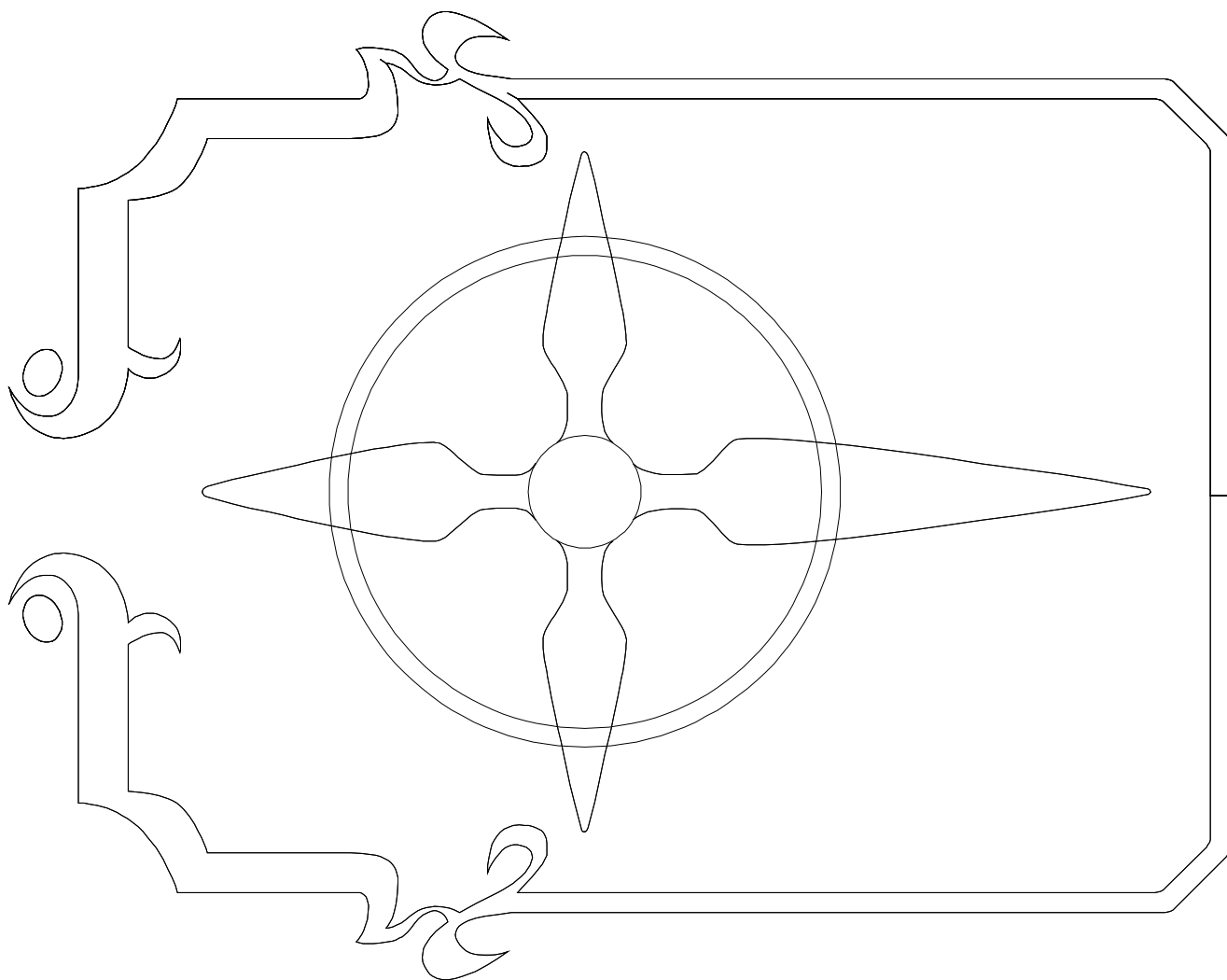
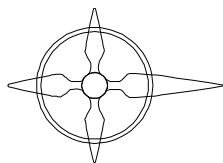
- 本システムにおける問題点・不足している機能等，気になった点を教えてください.

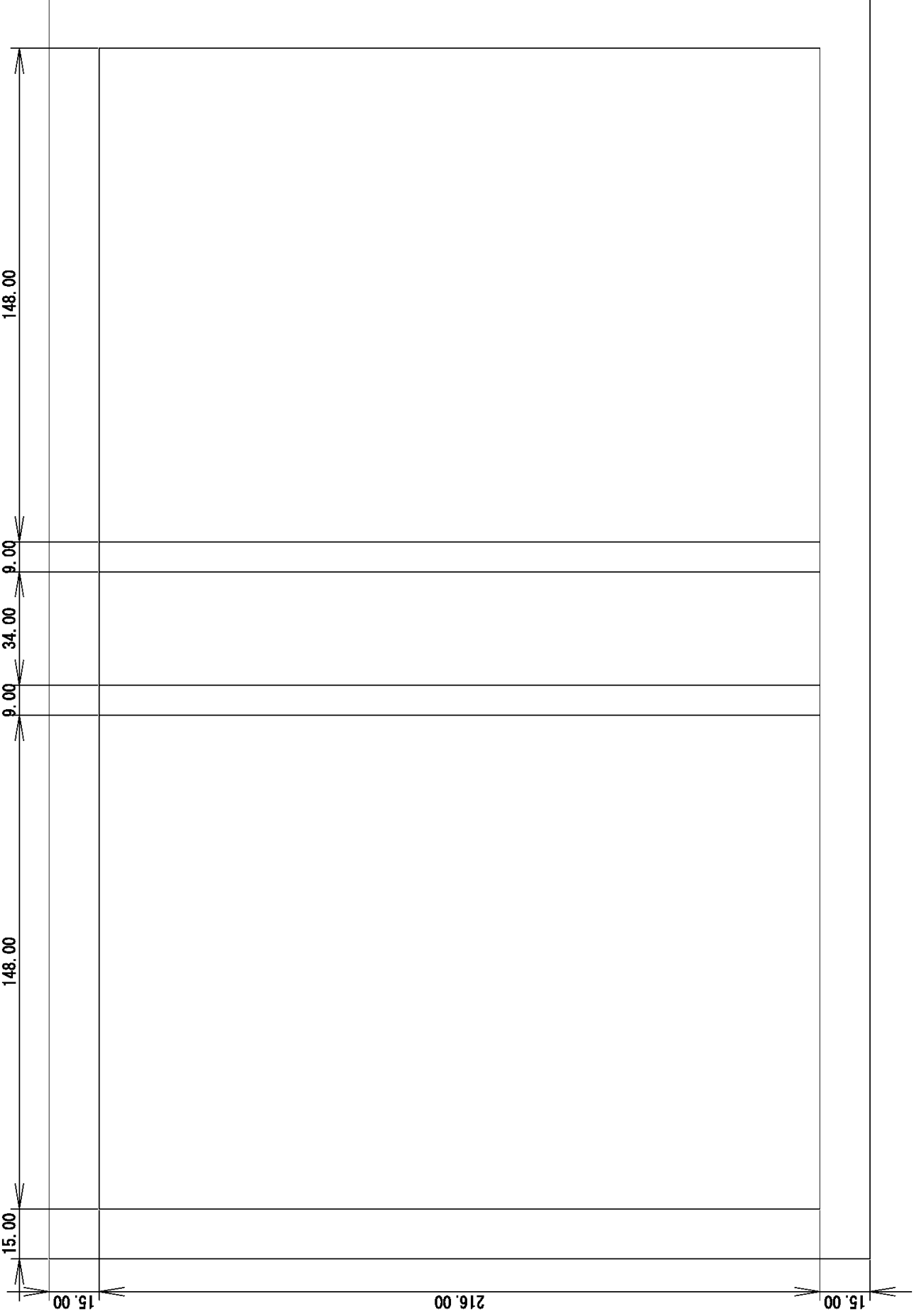
ご協力ありがとうございました.

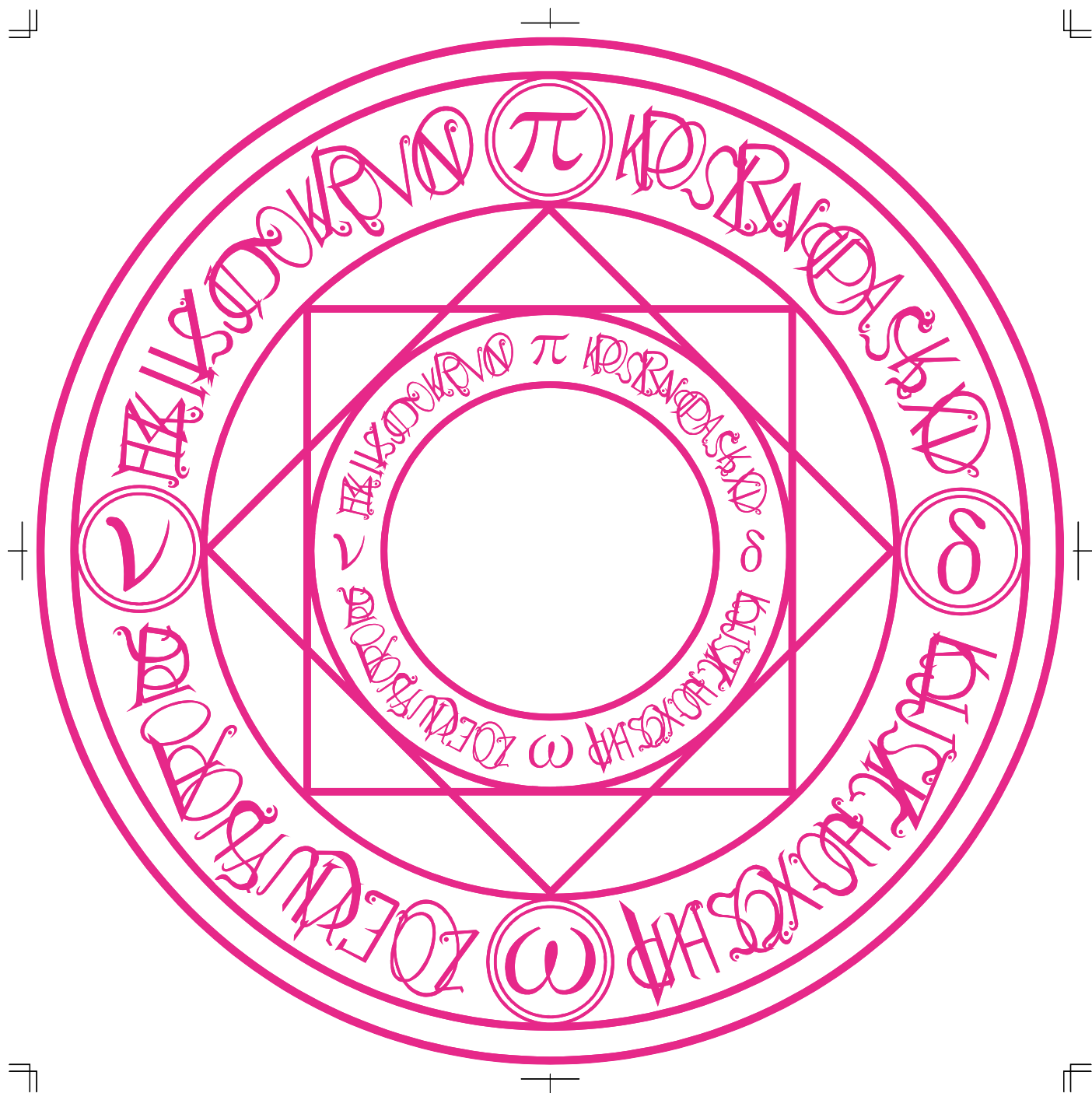
所要時間記入欄

AI : AI2

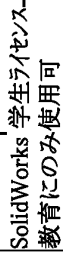
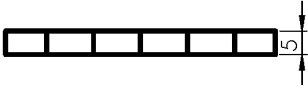
PS : PS2 :





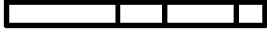
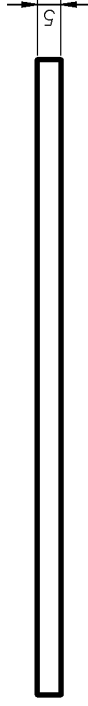
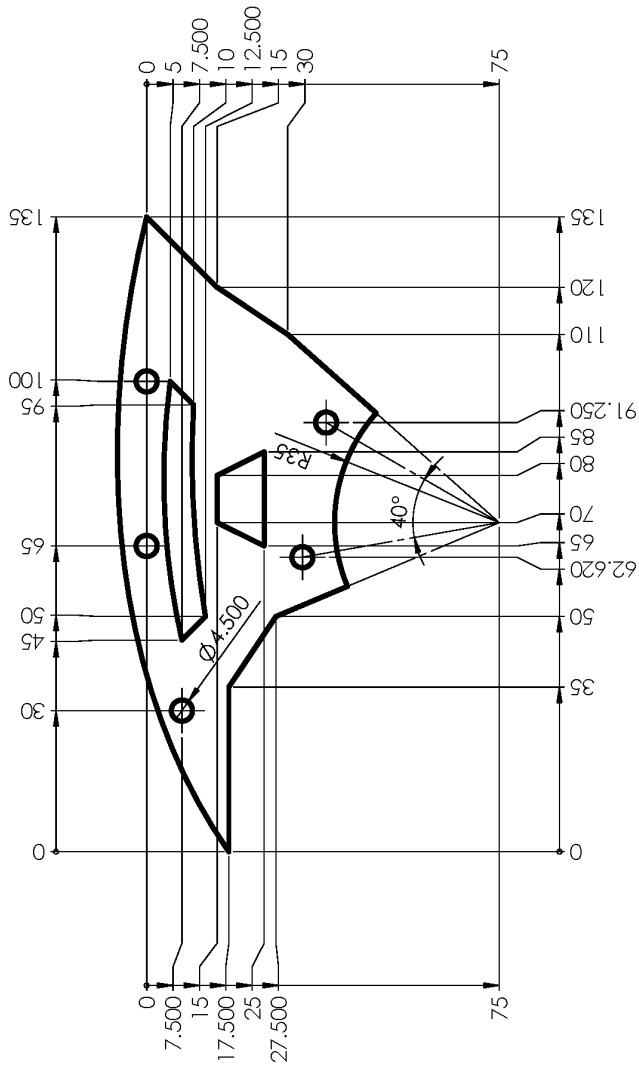






|                                                    |     |     |            |         |                                                                                                                            |
|----------------------------------------------------|-----|-----|------------|---------|----------------------------------------------------------------------------------------------------------------------------|
| 仕様を記載する<br>図面に記入し、欠ける<br>事項を明記して<br>公差<br>寸法<br>角度 | は上げ |     | 入り切り、鋸割の除去 |         | 図面決定不可<br><br>改訂                                                                                                           |
|                                                    |     |     |            |         |                                                                                                                            |
|                                                    |     |     |            |         |                                                                                                                            |
|                                                    |     |     |            |         |                                                                                                                            |
|                                                    |     |     |            |         |                                                                                                                            |
|                                                    |     |     |            |         |                                                                                                                            |
| 名前                                                 | 番 名 | 目 付 |            |         | 名称                                                                                                                         |
| 製造                                                 |     |     |            |         | <div style="text-align: center; font-size: 2em;">ブレード固定部</div> <div style="text-align: right; margin-top: -20px;">A3</div> |
| 検 査                                                |     |     |            |         |                                                                                                                            |
| 保 養                                                |     |     |            |         |                                                                                                                            |
| 設 定                                                |     |     |            |         |                                                                                                                            |
| 試 験                                                |     |     |            |         |                                                                                                                            |
| 品 名                                                |     |     |            |         |                                                                                                                            |
|                                                    |     |     |            | 材料:     |                                                                                                                            |
|                                                    |     |     |            | 数量:     |                                                                                                                            |
|                                                    |     |     |            | 尺 寸: 21 | 集数 1/1                                                                                                                     |



[illegible]



