Rokuro：ジェスチャを用いたモデリングインタフェースの実装と検討

佐藤 慧太† 山本 敏雄† 松山 克胤‡ 今野 晃市‡

現在 3DCG モデリングは，主にマウスを用いたモデリングツールによって行われている．しかし，それらのモデリングツールには専門的知識や特殊な操作が要求される事が多く，一般的に敷居が高い．本研究では，身体情報について着目し，ジェスチャを用いたモデリングインタフェースについての提案および検討を行う．本稿では，モデリングの対象を回転体に限定し，現実に使用されている「ろくろ」をベースとしたモデリングインタフェースの提案および，実装方法の検討を行う．現在までに，手形状の取得，回転体モデルの変形操作などの，ろくろ体験の基礎的機能を実装した．センサの解像度による仮想空間と現実との差異や，力覚フィードバックが得られないなどの課題はあるが，実現性のあることを確認できた．

Rokuro : Implementation and Examination of Gestural Modeling Interface

SATOH KEITA† TOSHIO YAMAMOTO† KATSUTSUGU MATSUYAMA‡ KOUICHI KONNO‡

3D model is generally modeled with a standard mouse in many 3D CG/CAD tool. The modeling tools require to the user special knowledge and manipulation. This is the reason why a user feels the 3DCG modeling is difficult to use. In this study, it's focused on physical information, and proposed and examined gestural modeling interface. In this paper, it's set the jiggering object of modeling as the rotating object, propose potter’s wheel based modeling interface and examine feasibility. This system was implemented the parts of the basic function for experiencing potter’s wheel such as capturing hands shape, transforming a rotating object, etc. Although the current system has some subject such as the difference in the virtual space and the reality by the resolution of the sensor and no haptic feedback, it was checked the possibility of this study.

1. はじめに

現在 3DCG モデリングは，主にマウスを用いたモデリングツールによって行われている．しかし，それらのモデリングツールには専門的知識や特殊な操作が要求される事が多く，一般的に敷居が高い．専門知識の例として，ナイフツールは現実にあるナイフとは異なり，モデルの一部を切り離すためではなく，メッシュの分割を行うものとなっている．また，特殊な操作の例として，マウスの 2 次元上の動きに対して，仮想の 3 次元世界のオブジェクトを操作するため，カメラ視点から得られた平面上でしか操作ができないなどが挙げられる．このように，3DCG モデリングでは現実ではほとんど使わない操作や知識が必要となる事が多い．現実世界で実際にものを作る動作に類似した操作方法を提供することで，3DCG モデリングにかかる負荷の軽減が期待できる．

本研究では，身体情報について着目し，ジェスチャを用いたモデリングインタフェースについての提案および検討を行う．本稿では，モデリングの対象を回転体に限定し，陶芸の制作に使用されている「ろくろ」をベースとしたモデリングインタフェースの提案および，実装方法の検討を行う．モデリングは，現実のろくろをベースに，仮想世界の特徴を考慮した機能を追加するアプローチをとる．これにより，現実感のある操作に加え，以下の仮想世界の利点を活かすことができる．

- 現実のろくろでは行うことのできないマテリアル，例えば鉄のマテリアルに対するモデリングを行うことができる．
- 簡単にアンドゥを行えるため，試行錯誤によるモデリングが可能である．

2. 先行研究

2.1 ポインティングデバイス

現在の 3DCG モデリングには，マウスなどの 2 次

† 公立はこだて未来大学大学院 システム情報科学研究科
Future University Hakodate, Graduate School of System Information Science
‡ 岩手大学 工学部
Iwate University, Faculty of Engineering.

元的なインタフェースを用いて操作するものが多い．その中で Igarashi らによる Teddy[1]はスケッチ型のインタフェースを用いることにより，絵を描くようにして 3D モデルを作成することができる．この研究は，3DCG モデリングの敷居を下げることに成功し，多くの研究に影響を与えている．本研究は，2 次元的なポインティングデバイスを使用するものでは無いため，目的へのアプローチが異なる．

2.2 ジェスチャモーション

ジェスチャインタフェースを用い 3DCG モデリングを行う横川らの研究[2]がある．これは，ステレオカメラによりユーザの手振りに合わせたモデルの形状を作成するというものである．これはモデルの変更を行うことを考慮していないという点で本研究とは異なる．また，Kinect を使い，手の位置を取得し，更に手にセンサグローブを付けることにより，ジェスチャを用い 3DCG モデリングを行うという Pirch の研究がある[3]．この研究は，クリックやドラッグなどのマウス操作を直接ジェスチャに置き換えるものであり，ポインティングデバイス的なインタフェースであるといえる．Blackshow による Recompose[4]は，深度センサを用い手形状の取得を行ない，そのジェスチャの内容によりパネルの形状を変化させることができる．また，パネル自体を押すことで，接触によるモデリングが可能である．ただし，操作可能な変形が高さ方向のみという物理的な制約がある．

3. 提案インタフェースの概要

本システムは身体情報に着目し，ジェスチャを用いて 3DCG モデリングを行うものである．提案するモデリングインタフェースの概要を図 1 に示す．

まず，ユーザは，モデリングを行う回転体のマテリアルの設定を行う．画面上にはマテリアルが設定された円筒モデルが表示される．円筒モデルは中心軸で回転している．ユーザはカメラの 1 メートルほど前の位置から，手を前に伸ばし，実際のろくろで陶芸を行うようなジェスチャを行うことで，回転体の形状を変化させる事ができる．また，足元に設置したフットペダルを踏むことにより，回転体の回転速度を変化させることができ，回転体の変形に影響を与えることができる．

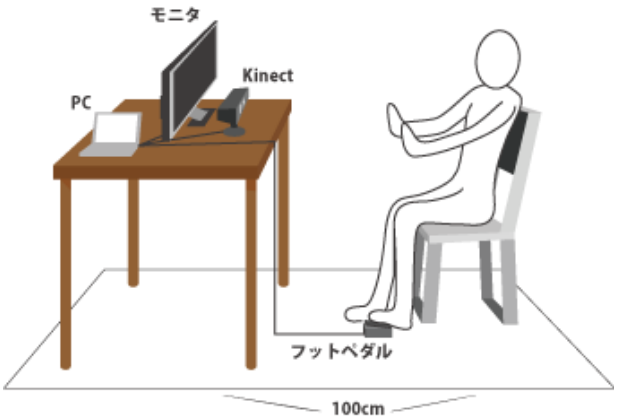

図 1 システムの概要図

このインタフェースを実現するために，ユーザから取得する情報として，リアルタイムでの手の位置，手形状，またそのジェスチャの認識などが必要になる．実現のための具体的な必要項目を以下に記述する．

1. **回転体のデータ表現**．変形可能かつリアルなレンダリングを行うことができる形状の表現形式を定義する．
2. **手形状のリアルタイム取得**．手形状の位置および形状の取得をリアルタイムに行う．
3. **身体情報に応じた回転体の変形**．取得した手形状および，手と回転体との位置関係に応じて，自然な形状変化を与える．
4. **3DCG モデルのレンダリング**．回転体モデルの描画を，操作性を損なわないように行う．
5. **フットペダルの作成**．踏み込む強さに応じて，ろくろの回転速度を操作できるようにする．

各項目の実現可能性，実装詳細については次章で述べる．

4. 提案システム

4.1 回転体のデータ表現

断面上の点列を用いて回転体の形状モデルを表現する．回転軸を通り，カメラ位置方向を向くビルボード平面 S 上に，図 2 のように一定間隔にサンプル点を配置する．隣接するサンプル点同士を接続したものを「ライン」と呼ぶ．ラインは，回転体の中心軸を境にして左右に 2 つ配置し，それぞれを，外部ラインと内部ラインのと定義する．外部ラインは回転体の外側の形状を，内部ラインは内側の形状を表すものとする．外部ラインと内部ラインを線形補間することにより回転体の断面形状を決定できる．また外部と内部ラインの差が厚さに相当する．

4.2 手形状のリアルタイム取得

手形状の位置および形状の取得をリアルタイムに行う．ここでは，Microsoft から発売されている Kinect を用いる．Kinect は深度情報やカラー画像，そして，

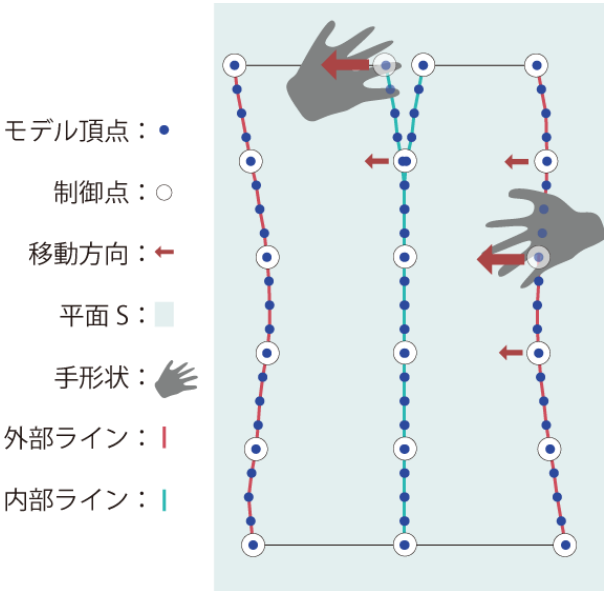

図 2 データ構造の例

ユーザの姿勢に相当するボーンを取得する事ができる．この深度情報を用いて手形状の計算を行う．通常の Kinect の深度センサの精度では，提案インタフェースでの手形状抽出は難しい．そのため，本研究では，NYKO から販売された Zoom という短距離認識用のレンズを Kinect に装着することで，より短い距離かつ広い画角で身体情報を取得することができる．

Microsoft Kinect for Windows SDK Beta では，深度センサから深度情報とユーザ情報が含まれた 16bit 間隔のテクスチャを取得することができる．ユーザ情報は，ユーザトラッキングに成功したユーザの ID である．手形状を認識するために，ユーザが手を前に伸ばしているかを判断する必要がある．これは，ユーザの手と肩による深度の差分が閾値以上になっていれば手を伸ばしているものとして判断する．今回この値は 100 とした．手形状の認識には，まず深度画像とユーザトラッキングにより得られた手の位置から，深度画像における手の X，Y 位置と深度，手形状の大きさ，そして，ユーザの ID を取得する．その後，深度画像上での手の位置を始点として、近傍のピクセルを探査していく。探査方法としては、下記の 1 から 5 の条件を満たせば手形状と判断し，探査対象のピクセルにおける上下左右の 4 ピクセルを探索し，条件に含まれなければ，破棄する．この探査を繰り返し行うことで手形状を取得する．

1. 手形状と認識したピクセル数が閾値以下であれば合格．そうでなければ不合格．
2. 現在調べているピクセルが手形状と判断されていなければ合格．そうでなければ不合格．
3. 現在調べているピクセル値のユーザ情報が，手のユーザ情報と同じならば合格．そうでなければ，不合格．
4. 現在調べているピクセル値の深度と手の深度との差分が閾値以下であれば合格．そうでなければ不合格．今回，この値は 50 とした．
5. 手の位置と現在調べているピクセルの位置によってできる距離が閾値以下であれば合格．そうでなければ不合格．今回のこの距離を 30 とした．

上記の手順により得られた手形状かれテクスチャを作成し，このテクスチャを手形状画像とよぶ(図 3)．手形状画像は Kinect から取得した深度画像を元に作成しているので，簡単に平面 S にマッピングすることができる．

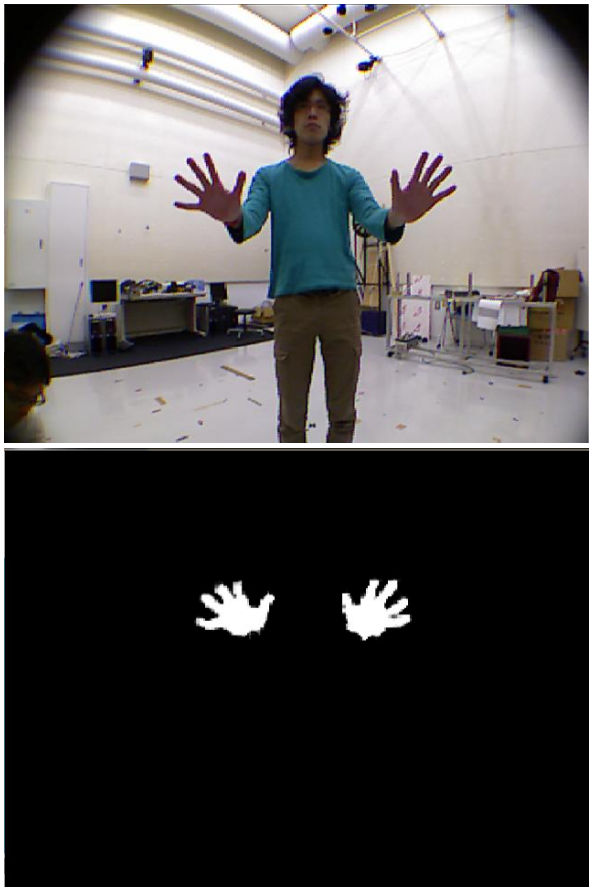

図 3 カラー画像（比較）と取得した手の画像

4.3　身体情報に応じた回転体の変形

手形状と制御点の当たり判定を行ない，当たっていれば，回転体の形状を変形させる．形状変形は，ラインのサンプル点ではなく，サンプル点の部分集合である「制御点」を新たに配置して，この制御点と手形状データとの位置関係により変形を行う(図 2).

左右において同じラインに存在する外部ラインの制御点であれば，同じ高さの内部ラインの制御点，また，同様に内部ラインであれば同じ高さの外部ラインの制御点を参照する．以後，この制御点の関係を『向かい合う』と定義する．

当たり判定は，各ラインの制御点を手形状画像へ射影し，射影した位置における手形状画像の画素値を参照することで行う．参照した画素値が一定以上であれば，手形状に当たったと判定し，制御点の移動を行う．

制御点の移動は，平面 S 上での水平方向，または，垂直方向のみ行い，ジェスチャと向かい合っている制御点との位置関係によってその方向を決定する．また，その移動の大きさは，ユーザが設定したマテリアルに合わせた定数をかけ合わせることで表現する．

制御点の移動には，2 つの制約を設定する．1 つ目は向かい合う制御点間の距離が閾値以上を保ち続けることである．もし，サンプル点の部分集合である制御点が交差してしまうと，ポリゴンの交差が起こってしまう．ポリゴンの交差は，現実には起こりえない現象であるため，3DCG モデルとしての体裁を保てなくなる．2 つ目は，内部ラインの制御点がモデルの回転軸上にあった場合，外部と内部ラインによる厚みを広げる際に，移動対象を向かい合う外部ラインの制御点のみに限定させることである．内部ラインは，左右に 2 つあるため，この条件を設定しなければ左右のラインに存在する内部の制御点が交差してしまい，結果的に，ポリゴンが交差してしまう．

回転体のモデリングを行うジェスチャは，「削り」，「厚みの変化」，「縦方向の引き延ばし」，「上部の削り」などが考えられるが，現時点では削りのみを実装している．これは，両手の手形状と内部ラインの制御点，または外部ラインの制御点と接触している場合，また片手の手形状のみが制御点に接触している場合，モデルの厚みを薄くする．具体的には，内部の制御点であれば外部の制御点へ近づけ，外部の制御点であれば，内部の制御点へ近づける．

この変形において，1 つの制御点での移動では，不連続な変形となってしまうことがある．そのため，なめらかな変化を与えるために，同じ面に存在する周囲の制御点についても影響を与える．具体的には，手形状の大きさに合わせ，同じラインの制御点を取得し，距離による係数をかけ合わせ移動させる．また，モデルの回転速度に合わせ，前フレームや次のフレームに用いるラインの制御点を取得し，時間を使った係数かけ合わせた移動を行う．また，計算し終わった制御点に対して，ライン上の制御点を通過する B-spline 曲線を作成し，B-spline 曲線上の点を回転体のモデル頂点とする．

次節で，実装中の他のジェスチャ要素についての記述を行う．

4.3.1　厚みの変化

左右において同じラインに存在する内部の制御点と外部の制御点が手形状に接触していた場合に，回転体の厚みを変化させる．例えば具体的には，両手の間隔を狭めると向かい合う制御点の間隔を近づけ，また，逆に間隔を広めると，制御点の間隔を広げる．

4.3.2　縦の大きさの変化

4.3.1 の厚みの変化のように，左右において同じラインの内部と外部の制御点が手形状が接触していた場合，両手を縦に動かすことでモデルの縦方向の大きさを変更することができる．制約として，設定した最大の大きさ以上にならないように，逆に，設定した最小の大きさ以下にならないようにする．

4.3.3　上からの削り

回転体の上部が開いていない状態で，内部を削ってしまうと，ユーザには認識できない位置を削っている事になる．そのため，この場合，閾値以上に広がっていない内部ラインの上部制御点を削るようにする．そうすることにより，ユーザが上から視認できる回転体部分を削ることになる．

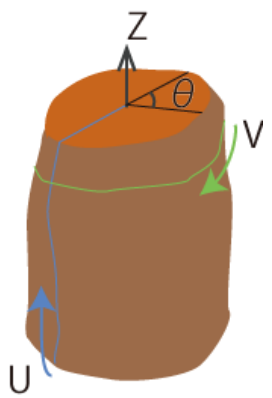

図 4 テクスチャのマッピング方法

4.4　3DCG モデルのレンダリング

視覚情報に対するフィードバックを与えるために，現実感のある描画や快適な操作に伴う描画を行う．描画には，「テクスチャマッピング」，「手の描画」，「回転体内部の表示」などが考えられるが，現時点では，テクスチャマッピング，および回転体内部の表示を実装している．

4.4.1　テクスチャマッピング

実際のろくろでは粘土のみを扱うのに対し，本手法では，任意のマテリアルを指定することができる．この時の，質感表現はテクスチャを用いて視覚的に行われるために，モデルを描画するときには，テクスチャマッピング可能であることが重要である．テクスチャマッピングのためには，回転体モデルにテクスチャ座標を与える．本研究では，テクスチャ座標(u, v)を，円筒座標系(r, θ, z)を用いて，θ と u，z と v をそれぞれ対応させた(図 4).

4.4.2　回転体内部の表示

3DCG モデルが描画されていると，回転体の内部を削る時に，モデルの手前部分によって手が隠れてしまい、ユーザに内部の様子を提示することができなくなる．これを回避するために，モデルの中心よりも手前に存在するモデル頂点の透明度を上げることで，アルファブレンディングを用い，内部を表示させる．図 5 は回転体内部の表示を行った結果である．

4.4.3　腕の姿勢の描画

現時点で実装中であるが，手形状の描画を行うことで回転体の変形の様子を視覚的に提示することができると考える．これは，まずユーザトラッキングにより 2 次元座標における手の位置，肘の位置を取得し，この二点を平面 S に射影することで姿勢の予測が可能となる．上記の 2 点から，肘と手を繋ぐボーンを作成し，

図 5 内部の描画

描画を行うことで実現できる．

4.5　フットペダル

現時点で実装中であるが，実際のろくろで用いられているフットペダルのような，回転速度の調整機能を提供すると，よりユーザの意図した制御が可能となると考えられる．これは，圧力センサを用いることで実現可能である．

5.　プロトタイプにおける実行

本提案システムについてのプロトタイプを実装し，3DCG モデルを作成した．図 6 はモデリングの様子とその結果できた 3DCG モデルである．動作環境は以下の通りである．

表 1 動作環境

OS	Windows7 Ultimate 64bit
CPU	Intel(R) Core(TM) i7 M620 2.67GHz
RAM	8.00GB
GPU	NVIDIA GForce GT 330M
開発言語	C++
開発環境	Visual Studio 2008

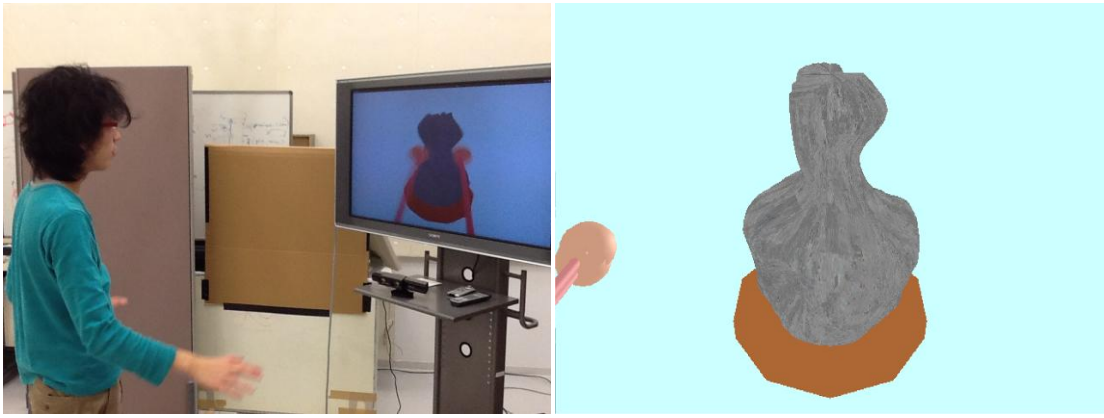

図 6 モデリングの様子と結果作成されたモデルの図

グラフィックラ ブラリ	DirectX 9.c
Kinect ドライバ	Microsoft Kinect for Windows SDK

Microsoft Kinect for Windows SDK を用いた理由は，キャリブレーションの動作が必要なくユーザトラッキングが行えること，また DirectX との親和性の高いということである．

実際に 3DCG モデルを作成していった結果として幾つかの，課題点が挙げられた．

1. 手形状や位置が安定せずにブレてしまうため，モデリングがうまく行えない．
2. 現実のろくろとは違い，かなり大きな動作をしなければモデルを作成することができない．
3. 現在，視覚によるフィードバックしかないため，回転体に触っている感覚が薄い．

6. まとめと展望

モデリングの対象を回転体に限定し，現実に使用されている「ろくろ」をベースとしたモデリングインタフェースの提案および，実装方法の検討を行った．試作としてプロトタイプにおけるモデリングを行った結果幾つかの課題点が挙げられた．この課題点を解消する方法について検討する．

6.1 センシング精度の向上

現在，手形状画像は Kinect のライブラリ上，精度が高くなく，安定性が低い．そのため，今後キャリブレーションとして，複数フレーム分の手形状画像を用意し，時間での重み付け平均化した手形状画像を作成する処理を実装していく．

また，より画素数の高い，かつ深度の精度の高い深度センサを用いることで，手形状の高精度な取得を行うことができるので，安定したモデリングが可能となり，更に，ユーザの動作範囲の問題も解決できる．

6.2 フィードバックの考慮

現在，力学的なフィードバックが与えられていないため，ユーザがモデルを触っているかどうかが分かりにくいという問題点がある．この点の改善としては，力覚デバイスの使用が考えられる．力覚デバイスには，SPIDAR[5]や PHANTOM [6]，超音波触覚ディスプレイ[7]などが挙げられる．SPIDAR は 8 本のワイヤとモータにより，仮想空間での衝突などを力覚で表現することができる．指にこのワイヤーを設置することで，ある程度の力覚を表現を行える．ファントムは，三次元上の位置の入力や仮想空間上での衝突に対して反力を出力することができる．これを手のひらに設置することにより，その反力などをフィードバックとして与えられる．また，超音波触覚ディスプレイは，超音波の振動によって離れた場所に対して力覚を与えることができる．これにより，回転体がある位置に対して力覚を与えることで，触覚に対するフィードバックを得ることが可能である．また，単純に手の指にモータを設置し，モデルと接触した際に振動を与えることでも，力覚に近いフィードバックを与えられるのではないかと考察している．

参 考 文 献

1) Takeo Igarashi， Satoshi Matsuoka and Hidehiko Tanaka: Teddy: A Sketching Interface for 3D Freeform Design，ACM SIGGRAPH 99 Los Angeles，August 1999
2) 横川 健， 石井 雅博， 唐 政， 山下 和也: ジェスチャインタフェースを用いた 3 次元モデリングシステムの構築，電子情報通信学会，pages 51-54，2007-6.
3) Sebastian Pirch: Kinect - 3D - Modeling，http://www.youtube.com/watch?v=LBpfYuVN0bI&feature=player_embedded，youtube，2011/11/08.
4) Matthew Blackshow， Anthony DeVlncenzi，David Lakatos，Daniel Leithinger and Hiroshi Ishii: Recompose: Direct and Gestural Interraction with an actuated Surface，CHI 2011 Working-in-Progress，pages 1237-1242，May 7-12 2011.
5) 株式会社アイネット 3D 事業部 : SPIDAR（スパイダー），http://www.ddd.co.jp/product/spidar/index.html，I-NET，2011/12/02.
6) 株式会社アイネット 3D 事業部 : 3 次元力覚デバイス PHANTOM，http://www.ddd.co.jp/product/spidar/index.html，I-NET，2011/12/02.
7) 名古屋工業大学 ：超音波触覚ディスプレイによる手書き文字コミュニケーション，http://www.youtube．com/watch?v=N8G7PSN9mQY，youtube，2011/12/02