

1. 担当PM

原田 康德 PM

(日本電信電話株式会社 NTT コミュニケーション科学基礎研究所 主任研究員)

2. 採択者氏名

チーフクリエイター:若園 祐作

(東京大学大学院 学際情報学府 学際情報学府学際情報学専攻)

3. 委託金支払額

1,792,000 円

4. テーマ名

可塑性のある物体に対するプロジェクションマッピングとその応用

5. 関連Webサイト

なし

6. テーマ概要

本提案では、3次元立体物のような言語では表現が困難な情報を適切に表現するための一手法として粘土での造形に着目し、プロジェクタによる粘土への画像投影と粘土の変形に対する追従機能によって、自由に粘土表面の色を変更したり擬似的に動きを表現したりすることが可能なシステムを構築する。このシステムによって、粘土での造形の簡単さを維持しながら、より素直かつ適切に人間の脳内イメージを実体化させることを可能にする。

本提案で開発を行うのは

- 1.粘土の形状認識とプロジェクションマッピング
- 2.粘土の形状変化に対応して投影画像を変形させる機能
- 3.システムへの入力インターフェース

である。

粘土での造形という単なる遊びと見られがちな行為を表現手法のひとつとして実用的に発展させようというアイデア、3D モデリングでしばしば用いられるようなコンピュータ内の粘土の再現ではなく現実世界の粘土に対してコンピュータによる拡張を加えようとする方向性、人間の理性だけではなく感性にも訴えかけるシステム入出力が本提案の特徴であり、アピールポイントである。また、本提案の成果は製品のプロトタイプリングや立体物のイメージ検索などに利用でき、様々な場面で応用が可能である。

7. 採択理由

粘土好きなら一度はやってみたいと思っていた提案。

kinect の登場で実現可能になったが、先駆者としてどこまで暴走するか。

もしくは現実的で実用的な解を目指すか。

どちらも可能性のある方向なので、楽しんで開発を進めてもらいたい。

8. 開発目標

タンジブル・ビットおよびタンジブル・ユーザ・インターフェースという概念が石井裕氏によって近年提唱され、注目を集めている(参考文献 1)。タンジブル・ビットは物理世界そのものをコンピュータのインターフェースに変えることを目的としており、これまでコンピュータの中だけの存在で実体のなかった情報を物理世界に存在する物体と結びつけることで、情報に実体や物理的表現を与える。それによりユーザは、情報に対して直接触れたり操作したりするような体験を得られる(図 1)。物体そのものが情報であり、その物体が入出力一体のインターフェースであるが故に、ユーザは直感的に操作しやすく、これまでのコンピュータに対して苦手意識を持っていた人々にも安心感を与えやすい。しかし、タンジブル・ビットの概念は各システムに特化したインターフェースを指向しており、汎用的なインターフェースを作りづらいという欠点がある。目的特化型の専用システムを開発する際にこの特性は障害とならないが、汎用的なコンピュータに接続する場合、マウスやタッチパネルディスプレイなどと同様に汎用性を持ったインターフェースであることが望ましい。本プロジェクトではタンジブル・ビットの概念を基にして、その特性を損なうことなく汎用性を高め、対象物を立体物全般に広げたタンジブル・ユーザ・インターフェースの創造を目的とする。そして、そのインターフェースを利用して新

たに実現できる数多くの体験を広めていくことで、コンピュータと物理世界のより緊密な融合、拡張を行いたい。



図 1 タンジブル・ユーザ・インタフェースの例 (Reactable)

タンジブル・ユーザ・インタフェースの汎用性を高めるためには、情報と結びつけられる物体として汎用的でかつそれ自体に意味づけのされていないものを選択することが効果的である。そのような例として、SandScape(図 2)や Illuminating Clay は物体として無色の砂や粘土を利用しており、プロジェクタによって色を投影することによってシステムから意味づけを行っている。これらの例では平面上の高さ情報だけを深度カメラによってリアルタイムで取得し、高さに応じた色の投影を行っているため、上部が膨らみ下部が凹んでいるような形状を対象とすることができない。本プロジェクトではこれらの手法を踏襲しながら、前述のような形状を含めて任意の立体形状を対象とすることができるようにする。

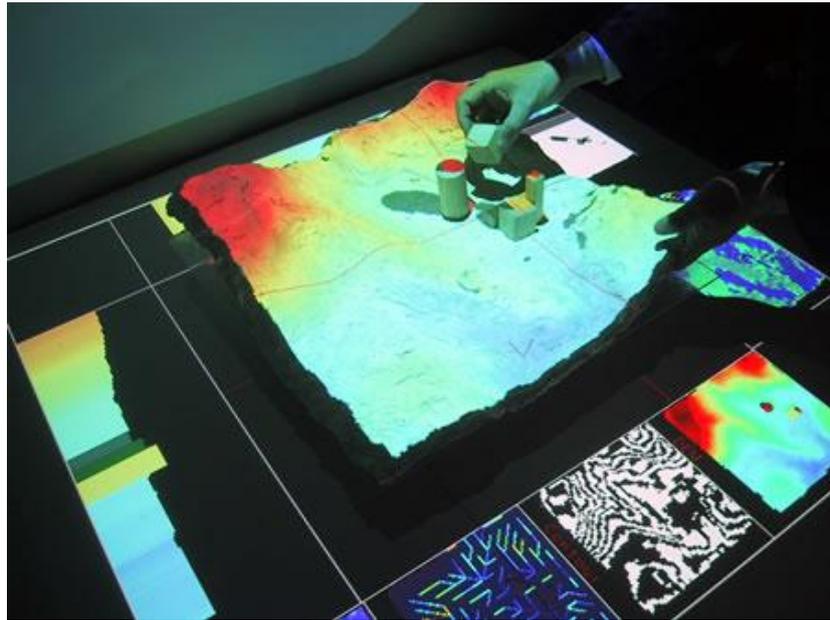


図 2 SandScape

誰にでも操作可能で任意の立体形状を表現可能な物体として、前述の Illuminating Clay でも利用されている粘土がある。粘土はその変形が保存される性質(可塑性)の扱いやすさから、現在でも幼児の知育玩具としてだけではなく、クレイアニメのような芸術・エンタテインメント作品(図 3)や、製品開発の際にプロトタイプを制作する目的にも利用されており、実用的な素材である。本プロジェクトでもモデリング用の白色粘土を利用し、その粘土に対して様々な投影を行う。



図 3 クレイアニメ制作風景

モデリング用粘土を利用するため、物体自身が変形しようとするのではなくユーザ自身が全ての変形を行う必要があるが、それでもこのインタフェースには様々な用途が考えられる。

- i. クレイアニメやプロトタイピングでは、粘土を固形化させてから表面に色を塗布したり、あるいははじめから色のついた粘土を組み合わせたりすることで粘土に色づけを行っているが、これらの方法では後から色を変更したり、完成後に再度変形を加えたりすることが難しい。ユーザが粘土を用いて任意の形状を制作し、そこに任意の色の投影を行うシステムが完成することで、表面の彩色を自由に試行錯誤することができ、作業工程の大幅な短縮に貢献できる。また、柄やアニメーションを投影することでこれまでにない外見の粘土を利用することが可能になり、創作の幅が広がることを期待できる。
- ii. 焼き物や置物、像を作成する際、既に存在する物体を手作業で忠実に再現したいという需要がプロ、アマチュアを問わず存在する。また、陶芸教室などで形状の修正を行う際、どの箇所をどれだけ修正すべきかが図示されるとわかりやすさが著しく向上する。現在の粘土の形状と正解モデルの差異を直接粘土に表示するシステムがあれば、これらの場面でユーザを強力に支援することができる。
- iii. コンピュータ上で3次元モデルを作成する際、現在は全ての工程をマウスやキーボードで行うことが一般的である。従来のこれらのインタフェースは精密な微調整には適しているが、概形を素早く作成することには不向きである。また、モデルのプレビューも画面のみで行う必要があり、実際にモデルを3次元で見ることはできない(図4)。モデル作成工程を実際の粘土と連携することができれば、モデル作成とプレビューの効率を大幅に向上させることができる。粘土自体に自立して変形する機能がなくても、過去に保存したモデルデータとの差分表示機能があれば、過去のモデルをもう一度手元の粘土で復元したいという需要にも応えることができる。

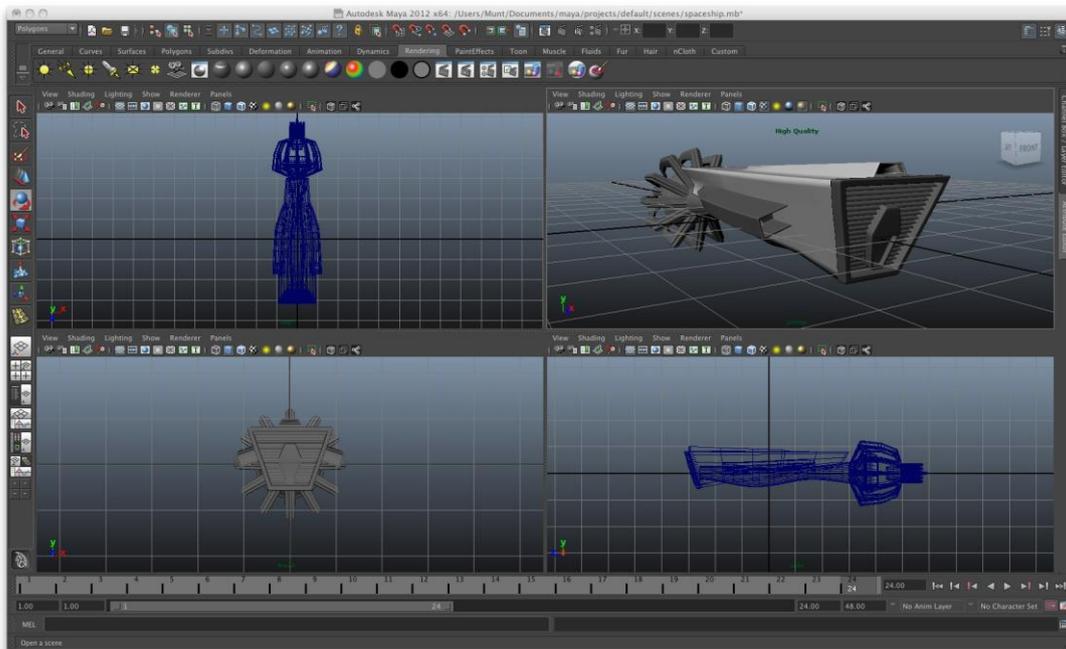


図 4 3D モデリングソフトの操作画面 (Autodesk Maya)

9. 進捗概要

開発内容

(1) “ケバブ型”試作機の開発

未踏期間の前半は、最終成果物とは全く外見の異なる試作機を開発した(図 5)。ソフトウェア処理は OpenGL 依存部を除いてノート PC 上の CPU で動作し、使用するプロジェクトも 1 台のみであった。

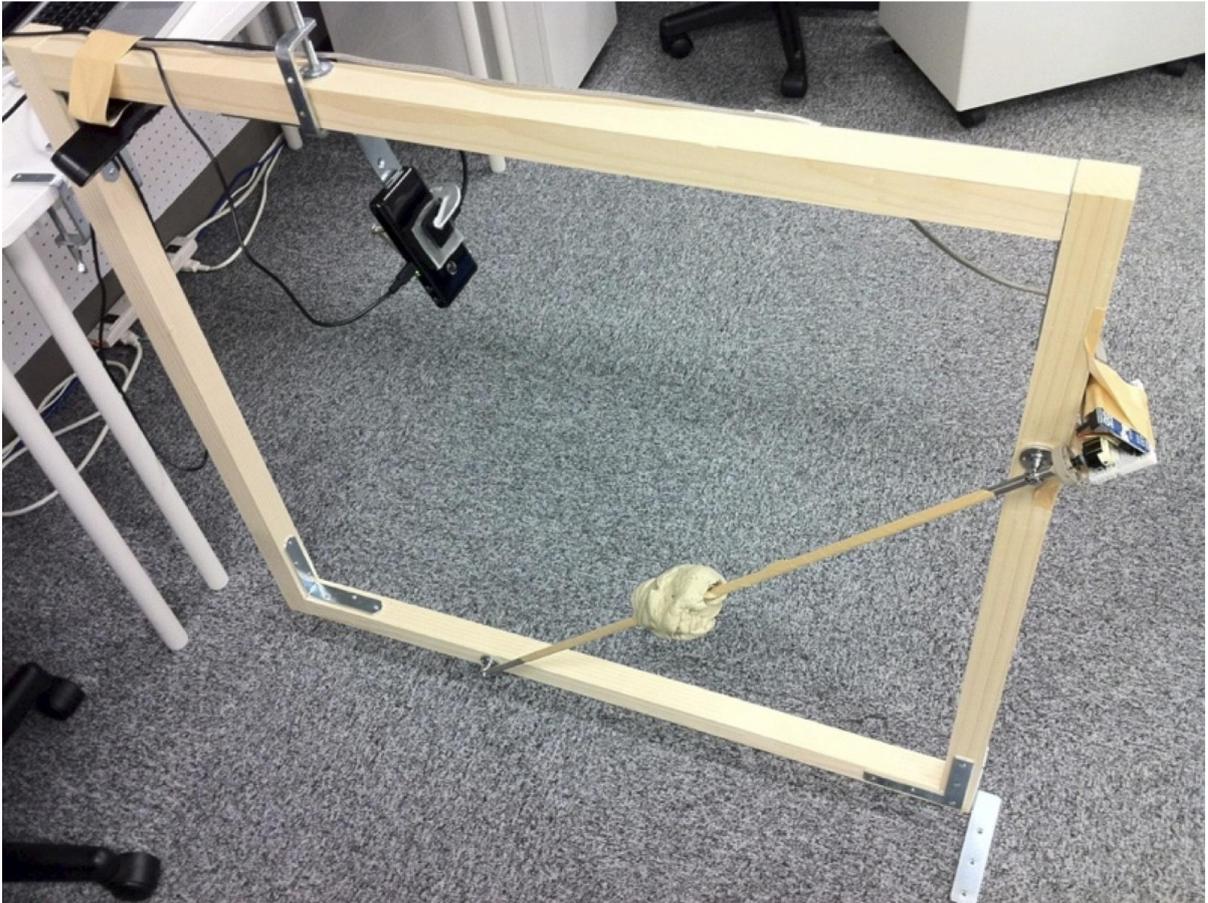


図 5. “ケバブ型”試作機

採用したプログラミング言語の C++は不慣れでメモリーリークが頻発、3D 処理ライブラリの Point Cloud Library にはバグが多いため自身の手でソースコードに手を入れ、開発は混迷を極めた。最終的に設計コンセプトと処理速度が問題となり、ほぼ全ての途中成果を不採用とした。しかし、この試作機の開発を通して C++の言語仕様や 3D 処理ライブラリの使い方を把握し、最終成果物の開発速度向上につながった。

① ハードウェア概要

形状読み取りおよび投影対象の粘土は図 5 の木枠右下、回転する鉄棒の中心に取り付けられている。粘土の横だけでなく上下の状態も確認しやすくなるよう、あえて鉄棒は木枠に対して斜め(45°)に取り付けられている。鉄棒の上部先端にはロータリーエンコーダおよび制御用 Arduino が取り付けられており、鉄棒および粘土が何度回転したかを USB 経由で出力し続けている。

粘土の形状を木枠右上に取り付けられた深度カメラ(図 6, 以降 Xtion と表記)で読み取る。そのデータは MacBook Pro 上の Windows プログラムへ送られ、投影する色を決定した後に木枠上部中央に取り付けられたレーザプロジェクタ(図 7)により粘土に

対して色投影を行う。



図 6. ASUS 製 Xtion Pro Live



図 7. Microvision 製 SHOWWX+ HDMI Laser Pico Projector

② ソフトウェア概要

ソフトウェアは Windows7 上の Visual Studio 2010 で C++を用いて開発し、ライブラリとして Xtion を利用するために必要な OpenNI、3D 処理ライブラリである Point Cloud Library、OpenGL を簡単に利用するためのライブラリである freeglut を使用した。なお、Point Cloud Library が依存する Boost ライブラリ、VTK ライブラリなども単体で使用した。

ソフトウェア処理の流れを図 8 に示す。Xtion からのデータは 3 次元空間上の点群データに変換されて読み込まれる。Xtion の解像度は 640×480 であるため点群の数

は $640 \times 480 = 307200$ だが、高速化のため 1/10 程度に点群を間引きする。そして、点群を小さな面の集合と見たときの各法線ベクトルを計算し、点群と法線ベクトルよりポリゴンデータを作成する。その後、粘土表面に貼り付けたい画像ファイルの情報を読み込み、テクスチャマッピングの一種である UV マッピングを行う。ここまでの処理は Point Cloud Library で用意されたメソッド群を用いることで行える。

1 度の読み込みで得られるデータは粘土の形状の一部であるため、粘土全体の形状を再現するためには複数角度から見たデータ群を統合する必要がある。本試作機では、加工前の点群データを Arduino から得た鉄棒の角度情報とセットにして保存し、ポリゴンデータ化処理の前に複数点群を統合したデータを入力と差し替える仕様とした。

Point Cloud Library には作成した 3 次元データを表示するためのビューアメソッドが付属しているが、テクスチャマッピング後のデータには対応していないため自作する必要があった。自作ビューアの土台として、OBJ ファイルを読み込み表示する freeglut のサンプルコードを参考にした。OBJ ファイルのかわりに Point Cloud Library での独自データ保持形式を読み込めるよう変換メソッドを記述し、独自データ保持形式を freeglut で扱いやすい形式に直接変換できるようにした。

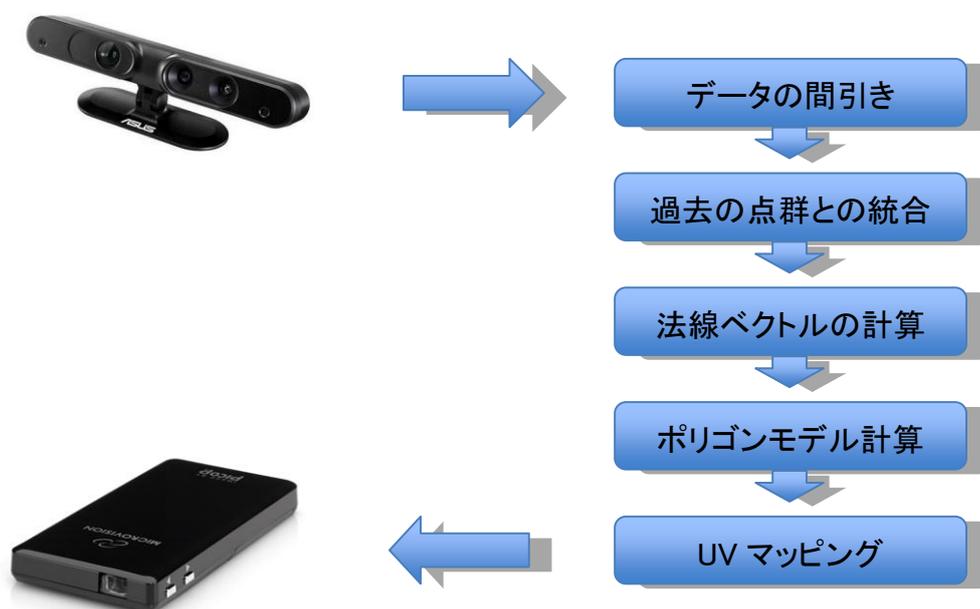


図 8. 試作機におけるソフトウェア処理の流れ

③ 研究室公開での展示と反応

3 月 19 日に所属研究室で研究室公開があったため、試作機の途中経過を展示(図

9)した上で来場者に対してコンセプトの説明を行った。好意的な反応をする来場者も若干名いたものの、特に斜めに取り付けた鉄棒に対して「邪魔で粘土を操作しづらい」「わざとわかりにくく斜めに付けているとしか思えない」といった声を多く得ることとなった。



図 9. 粘土に焼き物の模様を投影し、完成形との差分を赤色で指摘するデモ

④ 高速化のための最適化と限界

ある程度の機能が完成した時点で動作をさせたところ、画面の更新間隔が 0.1fps 程度しか得られなかった。その後、保存された点群データの回転角度の重複をなくすなどの対策を行ってある程度の改善がみられたものの、点群からポリゴンデータを作成する際に数秒の処理時間がかかることがわかり、このままの方向性ではリアルタイム性を保証できないと判断してこれまでのソースコードの破棄を決定した。

(2) 最終成果物の開発

5 月より、試作機の反省を生かして新たな成果物(図 10)を一から開発し直すことを決定した。試作機からの変更点は以下の通りである。

- 形状データのリアルタイム更新を実現するため、主な処理は GPU 上で並列処理することで大幅な高速化を実現した。それに伴い、新たにデスクトップ PC を導入し、NVIDIA 社の CUDA が利用できるビデオカードを装着した。

- 粘土の自由度を上げるため、鉄棒に粘土を固定するのではなく回転台(ろくろ)の上に乗せることとした。ろくろ上での作業に限定されるという制約はあるものの、元々ろくろは造形行為を行いやすくするための台であるので粘土の自由度は大幅に向上した。
- 粘土に対して投影面が狭かったため、投影面を広げるためにプロジェクタを 2 台構成として、深度カメラの左右に取り付けた。2 台にすることで投影の死角が減少し、ユーザの手によって投影が遮られる問題も軽減することができた。
- Xtion と同様の深度カメラである Microsoft 社製 Kinect を 3D スキャナとして用いて動きのない物体のキャプチャを行う KinectFusion という研究(参考文献 2)を参考とし、そのオープンソース実装である kinfu のソースコードを成果物に積極的に利用した。その結果開発速度がより向上し、成果報告会までにデモアプリケーションを開発することができた。

kinfu を積極的に用いることでソフトウェア処理の流れも試作機から変化し、図 11 のようになった。

デモアプリケーションとしては、特定の物体の形状(A)を記録し、記録後に置かれた粘土の形状(B)との差分を色によって提示するシステムを開発した(図 12)。これは 2. 背景の ii. に相当するシステムである。提示に用いる色はユーザが決定し、A と B を同じ位置で重ね合わせたときに A が B よりも突出している場合、ユーザの決定した色が投影される。B が A よりも突出している場合はユーザの決定した色の反対色(例えば赤を選択した場合は青緑、黄を選択した場合は青)が投影される。



図 10. 最終成果物として開発したシステム

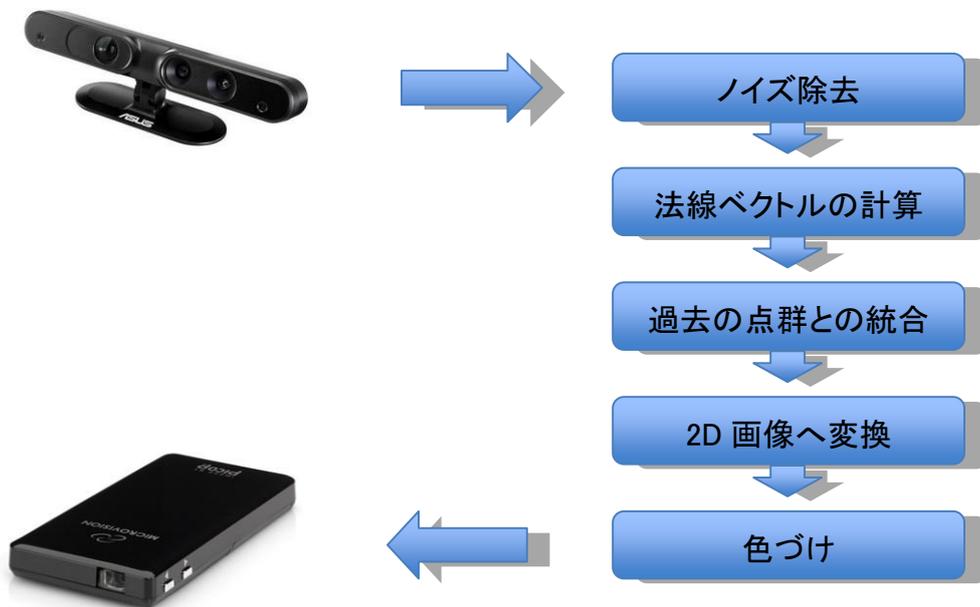


図 11. 最終成果におけるソフトウェア処理の流れ



図 12. 差分表示を行うデモアプリケーションの動作風景

あらかじめ人形の形状データを取り込んでおり、中心部に顔の輪郭が見える。黄は粘土を付け足すべき領域、青は粘土を取り去るべき領域を示す。

これ以後は、最終成果物の機能別に説明を行う。

② 粘土の形状をリアルタイム認識する機能

様々な投影を行う前に、投影対象である粘土が現在どのような形状であるかを認識する必要がある。また、投影対象はユーザによって変形させられるため、コンピュータ上の認識済みデータは絶えず更新し続けなければならない。本プロジェクトでは、粘土の形状を読み取り続けるために ASUS 社の Xtion Pro Live(以下 Xtion)を 1 台用いる。これは Microsoft 社の Kinect と同様に赤外線パターンを照射するプロジェクタ、赤外線パターンを読み取る深度カメラ、そして RGB カメラを搭載しており、解像度 640x480、フレームレート 30fps で深度データと色データを取り込むことができる。

赤外線パターンは 1 方向からしか照射されないため、1 度の読み取りで粘土の形状全体を読み込むことはできない。よって、粘土と Xtion の位置関係を変化させながら読み取りを継続し、時間的にデータを統合することで粘土全体の形状を認識することとする。この方法は、粘土の形状のうち Xtion から見て影になる部分に変形しても認識済みデータが更新されないという欠点があるが、粘土を回転させることで即座に変形部分を認識させることが可能であり、問題はないと判断した。

ソフトウェア処理の流れは以下の通りである。主要な部分は GPU 上で処理される。

(a) ノイズと不要な情報の除去

Xtionからの入力データにはノイズや不要な情報が含まれるため、前処理としてノイズと不要な情報の除去を行う。ノイズ除去・平滑化のためにはモルフォロジー演算とバイラテラルフィルタを適用した。不要な情報としては(i)映り込んだユーザの腕、(ii)背景の2つを考慮した。(i)に対してはRGBカメラからのデータを参照して色情報をHSVに変換後、肌色箇所の除去を行った。このとき粘土の表面色が肌色であると誤認識する可能性があるため肌色判定の閾値を厳しくとり、あえて肌色箇所の完全な除去は行わなかった。(ii)に対しては事前に背景を取り込み入力データに対して減算を行うことで背景除去を行った。背景除去は4.2.4の機能と処理内容を共通化している。

(b) 面の法線ベクトルの計算

Xtionからの入力データは各“点”の深度データである。各点を面の一部とみたとき、その点と周囲の点との位置関係より面の向きを計算することができる。この処理はkinfuのソースコードを流用した。

(c) 入力データと過去の認識済みデータとの統合

前述の通り、1度の読み取りで得られる入力データは粘土の一部分であるので、異なる角度から粘土を読み取った過去のデータとの統合を行う(図13)。kinfuではIterative Closest Pointアルゴリズム(参考文献3)を用いて入力データと過去データだけを用いてXtionの位置推定を行うが、これを本プロジェクトに適用したところ精度が悪くデータの統合がうまく行えなかったため、4.5で述べる方法で位置関係を取得し、その位置関係データをそのまま用いてデータの統合を行った。統合先はkinfuでも用いられているTSDF Volume(参考文献4)と呼ばれるもので、次々と新しいデータを統合することができ利便性が高いためそのまま用いた。

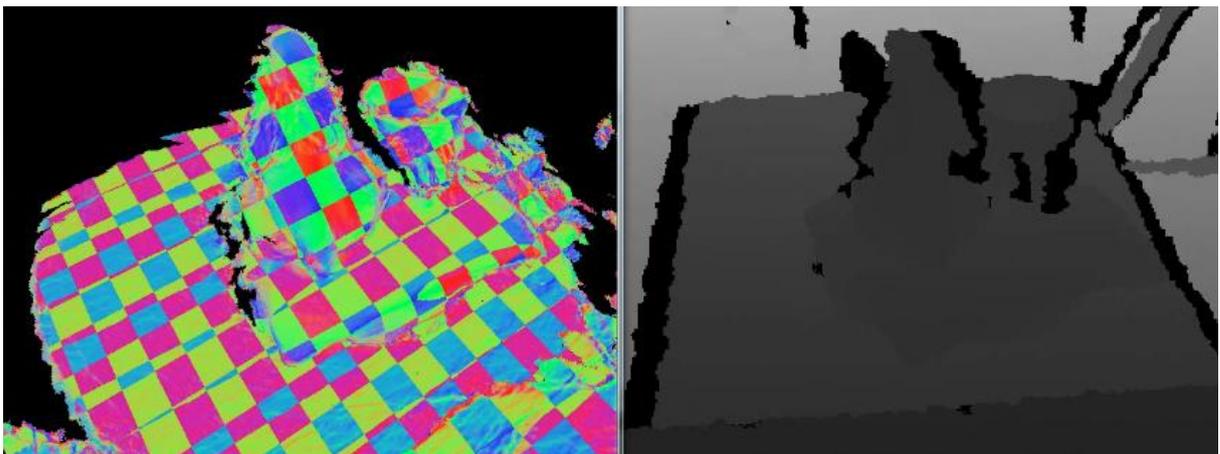


図 13. 過去のデータを含めた統合データ(左)と現在の Xtion からの入力データ(右)

② 粘土の形状を記録・保存する機能

前述の TSDF Volume データをそのまま保持し、新たなデータ入力を禁止することで粘土の形状を一時的に保存する仕様とした。本プロジェクト期間中に検証を行うことが難しいため凍結しているが、3D モデルの形式としては一般的な OBJ 形式ファイルの入出力機能も試験的に実装している。

③ 描画機能

データは独自の TSDF Volume として保持しているため、これを画面に描画するための 2D 画像に投影する処理も独自で持つ必要がある。この処理も TSDF Volume のソースコードと同様に kinfu のものを流用し、修正を行って利用している。2D 画像を生成する箇所は機能を拡張し、システムが任意の色づけを行えるようにした。

本プロジェクトでは粘土へのプロジェクションに 2 台のレーザプロジェクタを用いるため、異なる視点の 2 枚の 2D 画像を生成するよう記述した。粘土と Xtion との位置関係は 4. 2. 5 で述べる方法で取得できるがレーザプロジェクタの位置は取得できないので、Xtion とレーザプロジェクタの位置関係をシステムで固定することで粘土とレーザプロジェクタの位置関係を割り出すことができるようになった。

④ 保存された粘土の形状と現在の粘土の形状との差分を表示する機能

差分計算は 4. 2. 2 において保存したデータと現在読み込まれている認識済みデータを比較することで行う。本ソフトウェアでは 3D データを画面表示のための 2D 画像に投影する際に深度データも 2D 画像に付加しているため、簡単のため 2 つの 3D データから 2 枚の描画用 2D 画像を生成し、その 2D 画像同士で深度データの減算を行うことで差分計算を行った。

⑤ 回転台の角度を読み取る機能

kinfu では Iterative Closest Point アルゴリズムを用いて入力データと過去データだけを用いて Xtion の位置推定を行うが、これを本プロジェクトに適用したところ精度が悪く実用的ではなかった。そこで、粘土を認識させるための回転台(ろくろ)を設置してその回転角を読み取る方法を採用した。回転台の位置と回転軸はシステム起動時にキャリブレーションして固定し、回転角は回転台に接触させたロータリーエンコーダによって読み取る(図 14)。

回転軸のキャリブレーションは(i)回転軸に垂直な回転台平面の取得、(ii)回転軸に細長い物体を設置してそれを読み取ることによる回転軸決定の 2 段階とした。(i)は 4. 2 で述べた背景除去に必要な背景取得と同時にされる。取得データを点群データにした上で、平面状に分布した部分点群のうち一番数が多いものを抽出する。(ii)は専用のキー押下によって開始され、細長い物体(図 15)の中心点を通り平面と直交し

た線を回転軸とする。このように 2 段階に分けることで、回転軸が本来の角度とずれる可能性が低減された。

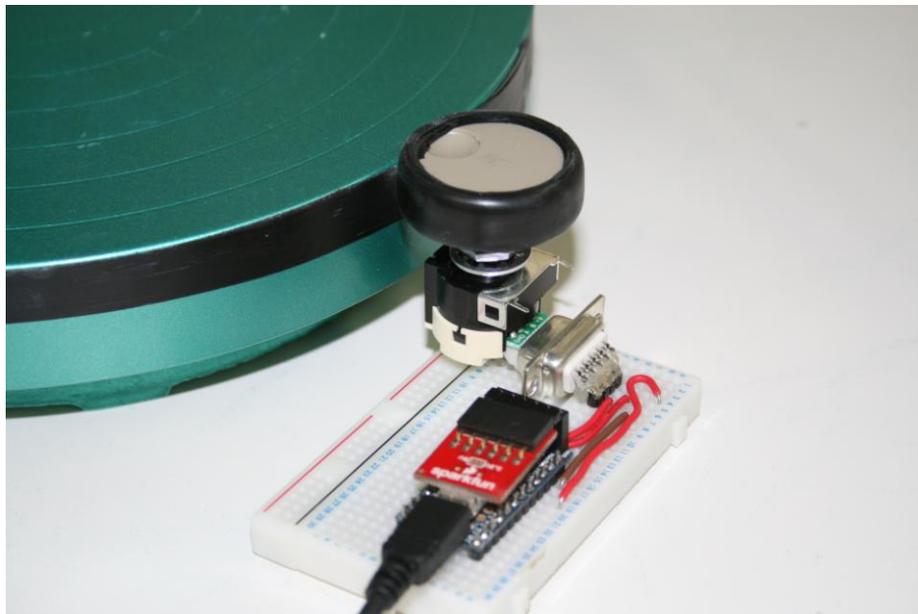


図 14. ろくろの回転角を読み取るロータリーエンコーダ



図 15. ろくろの中心に細長い物体を設置しキャリブレーションする様子

⑥ 回転台の回転角によって色相を選択する機能

差分表示プログラムにおいて好きな色を選択したいという要望が開発中に生まれたため、新たにデバイスを追加することなく色を選択する仕組みを考案した。一度粘土全体を読み込むと回転台を回転しても読み込みデータはほとんど変化しないことを利用し、回転台を色相環に見立てたインターフェースを実装した。回転台の角度が初期位置(0°)のとき赤とし、回転台を回転すると徐々に黄→緑→青→紫→赤と変化する(図 16)。内部的には色情報を HSV で保持し、そのうち H(色相)の値を回転角と同期することで実現している。色の決定は粘土情報の保存と同時にされる。



図 16. 回転台の角度変化によって色相が変化する様子(左:赤、右:紫)

⑦ ハードウェア制作

(a) ロータリーエンコーダの値送信機器

ロータリーエンコーダはインクリメンタル型を用いているため、出力される ON/OFF の値変化をカウントして角度を計算する必要がある。この読み取り・計算を Arduino で行い、USB に対して角度を出力し続ける仕様とした。PC 上のソフトウェアがロータリーエンコーダの角度を取得するためには、Arduino から USB へ出力される文字列を読み取ればよい。Arduino とロータリーエンコーダの配線は図 17 のように行った。

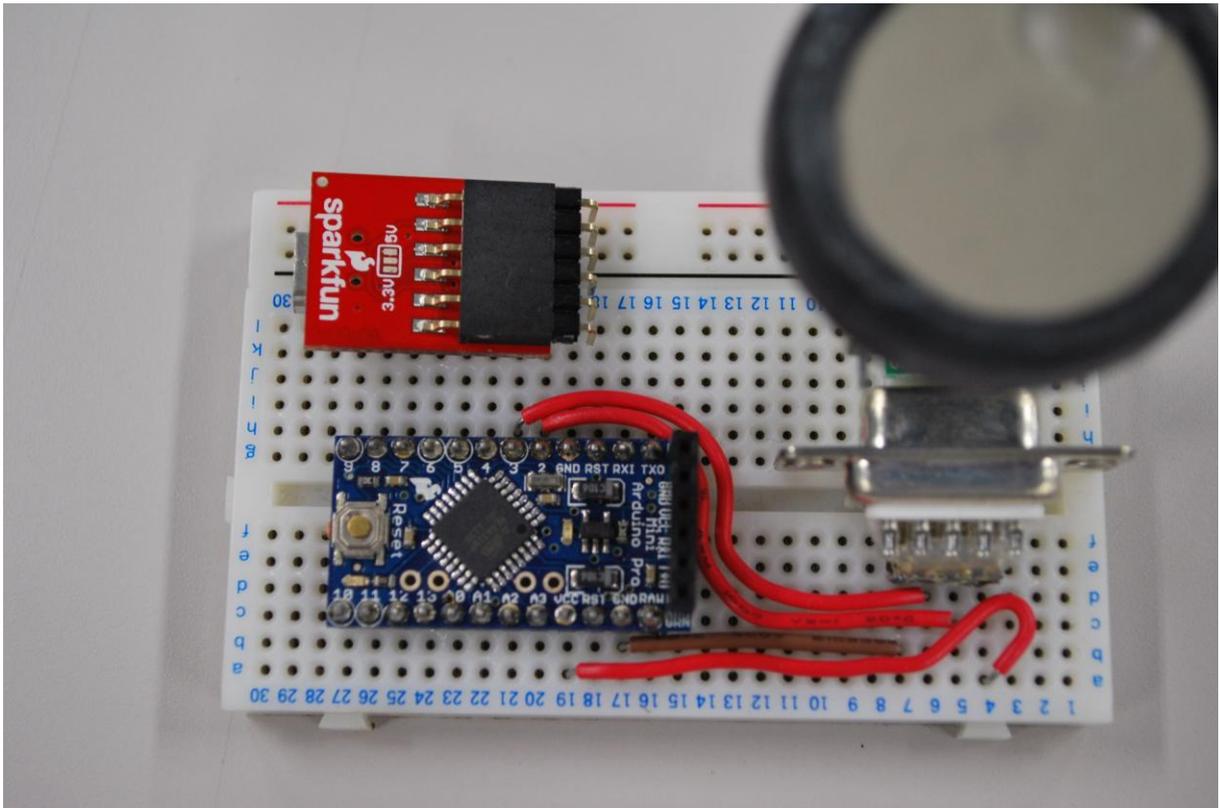


図 17. Arduino とロータリーエンコーダの回路

(b) 深度カメラ・プロジェクトの固定器

(3)で述べたように、Xtion とレーザープロジェクト 2 台の位置関係を固定する必要がある。そのため、厚さ 10mm のアクリル板を切り出し、そこに Xtion とレーザープロジェクトを固定することとした(図 18)。実験したところ、Xtion は本体より 60cm 以上離れた物体しか認識することができず、反対にレーザープロジェクトは 60cm 以上離すと鮮明な投影が行えないことがわかったため、粘土とレーザープロジェクトとの距離よりも粘土と Xtion との距離を大きくしている。また、簡単のため、粘土とレーザープロジェクトとを結んだ直線と粘土と Xtion とを結んだ直線のなす角は 45° 、レーザープロジェクト同士のなす角は 90° になるように固定した。レーザープロジェクトの投影面は横長よりも縦長のほうが投影面を広く取れたため、各レーザープロジェクトは向きを 90° 回転させて固定した。アクリル板と机はディスプレイ用アームを用いて向きを固定し、重さのバランスを取るために後方に重りを設置した。



図 18. アクリル板に取り付けられた各装置

10. プロジェクト評価

途中で開発方針の大幅な変更があったため、実質的には非常に短い時間での開発になってしまった。それでも、ある程度の形にまでもってこられた点は評価したい。このまま継続して開発を進めれば、世にインパクトを与えることができるだろう。

11. 今後の課題

本開発期間中に 1 つのデモアプリケーションを開発することができたが、システムの安定性に未だ問題があるため実証実験を行うに至っていない。そのため、まずは開発成果のソースコードを再点検してバグ修正やリファクタリングを行い、実証実験に耐えうるシステムの完成を急ぐ。そして、システムの使い勝手を向上させるため、粘土を回転台から離し再度置いたときに位置の再認識を行う機能などを実装する。その後は、開発成果の汎用性を生かし、さらに数種類のデモアプリケーションの開発を行う。現在のところ、コンピュータ上に蓄積された 3D 物体データベースから粘土の形状に類似したものを自動で選択し色を投影するデモや、粘土に対して筆で色を塗るよ

うに自由に着色を行えるデモを開発する予定である。

また、8月上旬の成果報告会において、より巨大な物体を制作する際の支援や食べ物に対するプロジェクションといった応用例が意見として出された。特に、巨大な物体を制作する際にはユーザが制作物全体を俯瞰することが困難であるため、制作物に対して直接情報を投影する開発成果の有用性はより一層増すと考えられる。食べ物に対するプロジェクションは、粘土以外にも開発成果は利用可能であることを示すために良い例であると考えられる。これらの意見を参考にしつつより多くのデモアプリケーションを開発し、開発成果の有用性について今後ともアピールし続けていく。

本開発の最終的に目指すのは、粘土に限らず 3次元入出力インタフェースの認知および普及である。これまでコンピュータにおいて主流であった、マウスやタッチパネルのような2次元入出力インタフェースの制限にとらわれない新たなコンピュータの使い方を提案する。