K-030

形状・圧力センシングを用いたテーブルトップインタフェースによるアート作品のシミュレーション

Simulation of Artworks based on Table-top Interface by using Shape and Pressure Sensing

浦 正広 † Masahiro Ura　山田 雅之‡ Masashi Yamada　遠藤 守‡ Mamoru Endo　宮崎 慎也‡ Shinya Miyazaki　安田 孝美‡ Takami Yasuda

1. はじめに

コンピュータを用いてアート作品やその制作技法をシミュレートする試みは様々あるが，それらは主に 2 つに大別される．1 つは作品としての出力結果に重点を置くもので[1]，もう一方は，結果のみでなく制作過程の再現にも重点を置くものである[2][3]．後者の目的には，制作環境の構築が難しいアート作品について，ユーザが手軽に実際と同じような制作工程をシミュレートできることや，制作過程そのものがパフォーマンスであるアート作品の練習環境を提供することなどがある．制作過程の再現のためには，実際の制作環境や技法のメタファに基づき個々の技法を表現できるとともに，アーティストの感性を反映できる直観的な入力インタフェースを提供することが重要である[4]．

直観的な入力を可能とするインタフェースとしてテーブルトップが挙げられる．テーブルトップは協調作業システムのインタフェースとして多く用いられているが，アート作品は平面的な制作環境であるものも多いことから，そのシミュレートのインタフェースとしても有効であると考えられる．しかしながら，テーブルトップをインタフェースとするアプリケーションの多くは，入力に指先やペンを用いたポインティングを想定した設計となっているため，制作に様々な道具を用い，また，入力の微妙な差異が作品に影響を及ぼすアート制作に適したものとはなっていない．このため，テーブルトップにおいてアート作品や制作技法のシミュレーションを行うためには，現実と同等なリアリティのある入力の実現が必要となる．

そこで本研究では，FTIR（漏れ全反射）を利用したテーブルトップインタフェースにおいて形状・圧力センシングを実現するためのフレームワークを構築する．入力形状をトラッキング可能なポインタとして扱えるようにし，また、入力のある／なしだけでなく，その入力の圧力（深度）を取得可能とすることで，様々なアート作品における制作技法への対応を実現する．フレームワークの応用例として，制作に手を用いて砂の配置を動的に変更するサンドアニメーション，一本のブラシを用いてカラフルな描画を可能とするレインボーアート，また，面に凹凸をつけることで立体的な形状を表現するレリーフアートといった，様々な描画・造形手法により制作されるアート作品を対象にシミュレータを開発し，その有効性と再現性を確認する．

†名古屋大学大学院情報科学研究科, Nagoya University

‡中京大学情報理工学部, Chukyo University

2. 描画・造形のための圧力センシングの現状

圧力センシングは，触覚を入力とするインタフェースにおける，入力を検知する仕組みのことである．代表例として，力覚フィートバックが得られる Phantom のようなデバイスが挙げられるが，テーブルトップを用いた圧力センシングの事例も報告されている[5]．このように，圧力センシングには一般的に専用のハードウェアが用いられるが，Smith らは，FTIR を用いたテーブルトップにシリコンを貼り付けることで，安価に圧力センシングを実現する方法を提案している[6]．また，このインタフェースのデモンストレーションとして，手や筆など様々な形状を入力とするペイントツールを開発している．しかしながら，入力形状をそのまま取得して描画に反映させる処理を行っているため，各入力が移動した際にトラッキングが行えない．このため，入力の移動量が大きい場合には補完処理が行なえず，意図した結果が得られない場合がある．アート作品は，微妙な入力の差異が表現力の違いとして現れるため，入力が適切に反映されないと作品の完成度に影響を及ぼす．そのため，様々なアート制作に対応するには，入力形状をトラッキング可能なポインタとして扱い，また，その際の深度も扱う必要がある．

3. FTIR テーブルを用いたインタフェース

提案するインタフェースは FTIR テーブルを用いて形状・圧力センシングを行い，複数の任意の形状をポインタとして扱いトラッキングする．また，入力される濃淡画像から圧力を推定し，それを深度等に対応する．

3.1 FTIR テーブルの作成

テーブルに接地した形状を入力として取得するために，FTIR を利用して接地センシングを行う．赤外線の全反射する性質を利用することで，テーブルに手が触れた箇所を白，触れていない箇所を黒とする濃淡画像が，カメラを介して取得可能となる．また，Smith の手法を用いてアクリル板に薄いシリコンの膜を貼ることにより，接地面における圧力の差異を取得することが可能となる．図 3-1 にハードウェア構成を示す．

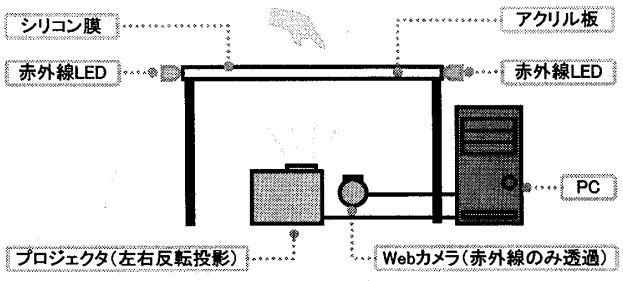

図 3-1　FTIR テーブルの構成

3.2 フレームワークの構築

FTIR テーブルを用いたマルチタッチ環境構築用ライブラリの 1 つに Touchlib があり[7]，これは複数の指によるポインティングを認識し，また，トラッキングする機能を有するが，入力形状をすべて点として扱う．このため，アート制作のように手やブラシなど様々な道具を用いて様々な形状の入力が想定される場合には，入力の反映に限りがある．また，Smith らの例では，入力形状をそのまま入力として扱えるが，先述のようにトラッキングが行われないために，動作の速度がカメラ入力のフレームレートを超える場合には，入力が適切に反映されないなどの問題がある．そこで，前節のインタフェースで取得可能な画像から，テーブルに接地した手の形状をポインタとして認識し，複数ポインティングおよびトラッキングを可能とする手法を提案する．まず，カメラからの入力画像を 2 値化，ラベリングし，領域画素およびその輪郭画素を抽出する．つぎに，各領域の輪郭画素座標の平均値を算出し，それをポインタの中心点とする．各ポインタの中心点の位置は前回フレームと比較し，距離の近いものを同一ポインタとして処理する．また，新しいポインタには識別 ID を付与する．これにより，ポインタのトラッキングを実現する[8]．

つぎに，入力の深度を取得する手法について提案する．先述のように，FTIR テーブルでは圧力がある箇所ほど白くなるため，カメラ画像の濃淡がその地点における圧力に対応する．そのため，この画像をそのまま用いることで圧力の取得は可能であるが，画像全体が処理の対象となるため計算効率が悪くなる．そこで，抽出した各領域画素のみを対象とし，対応する濃淡画像の画素の値をそのまま圧力として採用する．これにより，計算量が少なく，かつ，ポインタに深度情報を持たせることが可能となる．

これらポインタのパラメータは，図 3-2 で示す構造により保持する．図 3-3 にポインタの認識例を示す．左図が処理前の，右図が処理後の画像であり，白色が輪郭，その輪郭内部が深度情報を付加した接地面，数字がポインタの識別 ID である．

```
struct pointer{
  int id;                 // 識別 ID
  list<pixel> area;       // 領域のリスト
  list<pixel> outline;    // 輪郭のリスト
  list<int> depth;        // 領域の深度のリスト
  point center;           // 中心点
};
```

図 3-2　ポインタの構造

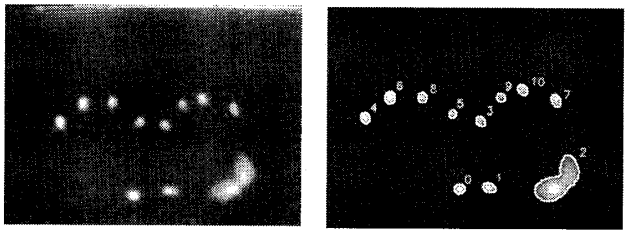

図 3-3　ポインタの認識例

FTIR テーブルにおいては，プロジェクタによる投影映像とカメラの取得映像の座標とを，機器の配置により完全に一致させることが困難であるため，ソフトウェア処理によりキャリブレーションを行い，両者が一致するように補正している．Touchlib のように入力を点として扱う場合，各点の中心座標に対しての射影変換処理のみで，補正処理を行うことが可能である．一方で，提案手法のように入力形状をそのままポインタの形状として用いる場合，Touchlib 同様に形状領域の中心座標のみ変換し，その中心にあわせて入力形状を配置すると，カメラとプロジェクタの歪みが大きいほど両者の映像の差異が大きくなってしまう．そのため，形状を一致させるには入力形状の画素も射影変換の対象とする必要があるが，このとき，点のポインタの場合と比較して計算量が幾倍にも大きくなることは明らかであり，処理に遅延を発生させる要因となりうる．そこで，3D 描画ライブラリに備わるテクスチャ機能を用いることで，計算量を削減する．図 3-5 はキャリブレーションの例である．左図の黒色の四角形はカメラでの取得領域，灰色の四角形はプロジェクタの投影領域を示す．これを右図のように投影領域の 4 隅の座標を，取得領域の 4 隅の座標と合致するようにテクスチャを貼り付ける．これにより，画像の精度が実際の解像度よりも低下するものの，安定した処理速度が期待できる．一方で、入力が限られた個数の点として処理できる場合もあり，このときは計算量に大差がなく，かつ，点で処理することにより解像度の低下も防げるため，従来同様に点を射影変換する機能も併せて提供する．

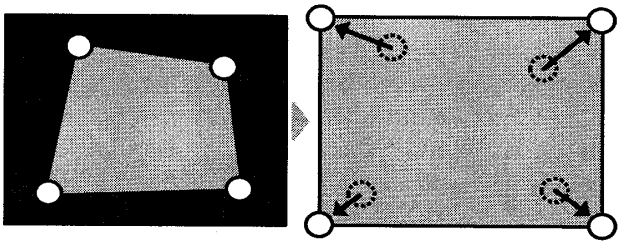

図 3-5　テクスチャを用いたキャリブレーション

4. アート制作のシミュレート

前章で示した環境を用いてアート制作のシミュレーションを行う．対象とするアートはサンドアニメーション，レインボーアート，レリーフアートとし，それぞれのアートの性質や処理速度を考慮し，それぞれ表 4-1 に示すフレームワークの機能を利用してシミュレータを制作する．また，シミュレーションで制作した作品と実際の環境で制作した作品とを比較し，その際の処理速度を計測することで，提案手法の再現性と実用可能性を確認する．実験において制作する作品はそれぞれ制作手順を定め，事前に作成した絵コンテに基づいて制作を行う．実験は表 4-2 に示す環境下で実施する．

表 4-1　利用するフレームワークの機能

	トラッキング	圧力	キャリブレーション
サンドアニメーション	あり	なし	テクスチャ
レインボーアート	あり	なし	点
レリーフアート	なし	あり	テクスチャ

表 4-2　実験環境

OS	Windows Vista Business
CPU	Intel Core2 Duo E6400 2.13GHz
Memory	4GB
Graphic Card	ATI Radeon HD 4800 (RV770)
Camera Resolution	640x480pixel

4.1　サンドアニメーション

サンドアニメーションの制作技法の多くは手をテーブル状のキャンバスに接し行われる．本節ではサンドアニメーションアーティストである Ferenc Cako 氏の作品映像[9]から制作環境と技法の調査を行い，その結果に基づいて砂のふるまいや制作技法のシミュレーションを行う．

4.1.1　サンドアニメーションとは

サンドアニメーションは，砂を用いてアニメーション作品を制作するアート技法，また，その制作過程そのものを披露するパフォーマンスアートの名称である．キャンバス上に砂を配置し，それをライトにより背面から照らすことでセピア調の温かみのある独特の表現が可能である．サンドアニメーションは，図 4-1 で示すような環境下で制作される．アーティストはテーブル上のキャンバスの上で砂を操作し作品を制作する．パフォーマンスアートとして制作過程を披露する場合には，キャンバスをカメラで撮影し，その映像をスクリーンに投影する．

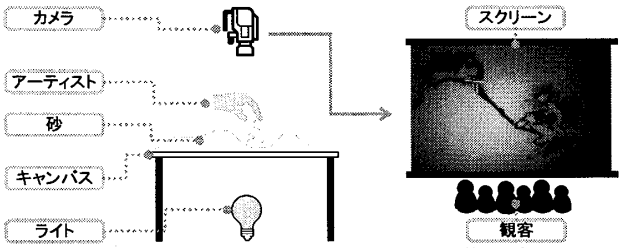

図 4-1　サンドアニメーションの制作環境

4.1.2　制作技法のシミュレート

サンドアニメーションには制作のための描画技法が存在する．そこで，調査結果に基づいて各技法における砂のふるまいを再現する．砂の配置や配置済みの砂の移動や除去など，アニメーションを構成するためのいくつかの制作技法があるが，それらを「置く・撒く・削る」の 3 つに分け，それぞれシミュレーションを行う[10][11].

置く：手に砂を持ち，キャンバス上にそれを置くようにして線や点を描く．

撒く：手とキャンバスを離して砂を撒き，一定の範囲を淡く塗りつぶす，また削るための下地を作る．砂を落とす際，慣性力が大きいと砂は広範囲に飛び，速さや勢いのある表現になる．

削る：指や手の平により配置済みの砂を削る．削った後にできる隙間から背面光が漏れ，コントラストによりはっきりとした線を表現する．

なお，実際のサンドアニメーションでは砂を「置く・撒く」際には，手はキャンバスには触れないが，FTIR テーブルでは接地が検出されないとポインタの認識が行えないため，現段階においては手をテーブルに接地させて入力の指示を行う．また，シミュレーションにおいて，砂はセルごとに高さを管理するハイトフィールドにより扱い，砂量に応じて描画を変更することで，濃淡や質感を表現する．砂が無い状態すなわち砂量 0 の場合の色を RGB(255,240,120)，砂量が T 以上で背面光が遮断されている状態の場合の色を RGB(40,0,0)とする．さらに，砂の質感を表現するため，ディザ法を用いる．これはあらかじめフィールドの各画素にランダムで閾値を設定しておき，ハイトフィールドにおける砂量が閾値に満たない場合は描画時の輝度を高め，ばらつきを表現する．

(1)　技法「置く」のシミュレート

ポインタの中心点 p を中心とする半径 r の円の内側に砂を式 $k_1 \cdot (r-l)+e$ で定める量だけ追加する．ここで，k_1 は制御定数，l は対象画素と p との距離，e は乱数値とする．これにより砂を置いた際に，砂が崩落し山状に配置される様子が再現できる．また，追加する砂量をランダムに増減させることで，一様でない自然な描画が得られる．

(2)　技法「撒く」のシミュレート

現在のフレームのポインタの中心点 p と，1 フレーム前のポインタの中心点 q との距離 d を求める．d に比例する慣性力を仮定し，線分 pq の延長線上の点 p' を定める．つぎに p' を中心とする円の内側に，「置く」より広範囲に薄く砂を配置し，距離 が大きければ補完処理を行う．

(3)　技法「削る」のシミュレート

この技法は，配置済みの砂を手の進行方向へ移動するため，ポインタ領域画素上の砂をポインタ輪郭画素に移動させるという処理により，シミュレートを実現する．まず，ポインタの領域リスト area に格納された各画素の砂量が v 以上ならその画素から砂量を v だけ減少させ，v より小さければ砂量を 0 とする．次に，減少させた砂量の総和 s を求め，そのポインタの各輪郭画素に s/n の量の砂を追加する．ここで n は輪郭画素数である．ポインタの移動距離が大きい場合には，処理ごとに進行方向と反対側の輪郭に砂が発生してしまうため，1 つ前の補完点におけるポインタ領域と重複する場合には，その輪郭画素に砂の分配を行わない．

4.1.3　実行例

絵コンテに基づいて制作したサンドアニメーションを図 4-2 に示す．左がシミュレーションによるもの，右が実際に砂を用いて制作したものである．ディスプレイのキャプチャと実物の撮影写真であるため，背景色に差異がみられるが，様々な技法を用いた作品が，ほぼ同一の描画物として描かれていることが確認できる．図 4-3 はシミュレーションにより「削る」を実行している様子である．

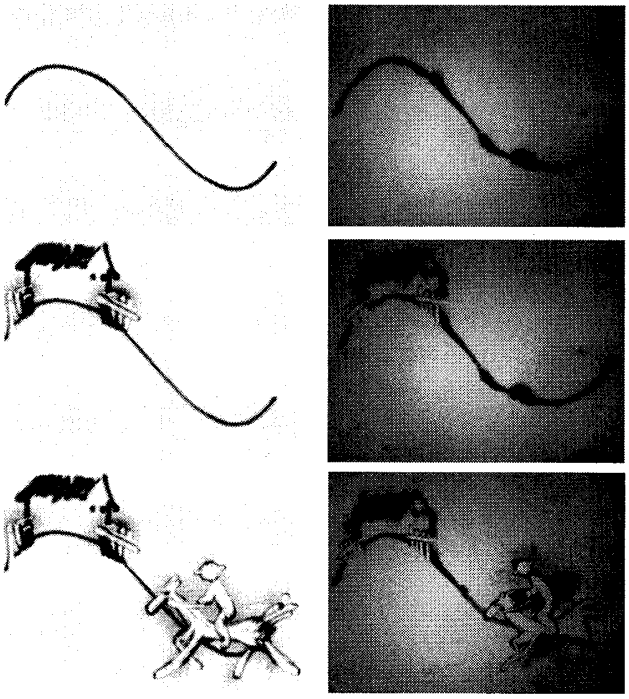

図 4-2　制作したサンドアニメーション

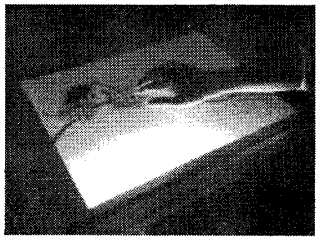

図 4-3　シミュレータによる「削る」の操作

4.1.4　処理時間

「置く・撒く・削る」の各操作について，手を接地させてからそれをポインタとして認識するまでの時間，また，1 フレーム分の処理が完了するまでの時間を計測する．結果，ポインタの認識までに要する時間は平均で 500ms であり，これは，Web カメラが取得画像を表示する際に発生する遅延と同等であったため，ポインタ認識の処理時間は十分に小さいといえる．また，表 4-3 が示すように，いずれの操作時も 30fps でのフレームの更新間隔である 1/30s より実行時間が小さくなったため，処理落ちのないシミュレーションが実現できているといえる．

表 4-3　1 フレーム分の処理の計測結果

条件	面積（px）	処理時間（ms）		
		置く	撒く	削る
指 1 本	300	1	1	3
指 5 本	1700	1	1	3
手側面	3500	2	2	5
片手	13000	3	4	9
両手	27000	5	5	10

4.2　レインボーアート

描画に道具を用いたアート作品のシミュレート実行例を示す．細部の再現性の確認のため，通常のブラシのように単色ではなく，虹色のカラフルな描画が可能なレインボーアートを対象としたシミュレートを行う．

4.2.1　レインボーアートとは

レインボーアートは，ラスゾロ・スゼキリー氏によって考案された低年齢向けの描画ツールである[12]．板状のスポンジブラシを用いて紙に色を塗ることで，作品制作を行う．スポンジブラシは横長になっており，複数の着色剤を同時に付着させることで，虹のような線を描くことが可能である．これによって，誰にでも容易に色彩豊かな絵を描くことができる．ブラシの特性上，描ける形がある程度決まっており，多くの作品は扇形，四角形，直線の組み合わせで制作される．図 4-4 にツールと作品制作例を示す．ブラシの大きさは様々あり，最大のもので 6 色まで付着が可能である．作品の制作は一人だけでなく，複数人が同じキャンバスに同時に描くような，レクリエーションとして行われることもある．

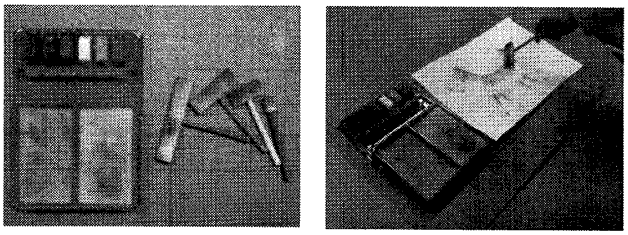

図 4-4　ツールとその使用例

4.2.2　制作技法のシミュレート

ブラシには，着色剤のついた先端面を塗装したい面に接地させたまま移動させることで，着色剤が引き延ばされ面が描画されるという特性がある．そのため，レインボーアートで用いられるような幅の広いブラシは，接地面がブラシ先端面の末端同士を結ぶ一本の線であるとみなしたとき，その移動ストロークが面になるという解釈が成り立つ．また，レインボーアートにおいては，その線が通常のブラシのように単色ではなく，複数の色により構成されていると考えることができる．

そこで，フレームワークにより取得される入力の中心点と輪郭の情報を用いて，入力を線として扱うことにより，ブラシによるテーブルトップ上での描画を実現する．まず，入力された領域の中心点 c から，最も離れている輪郭点 p をブラシの片端とする．つぎに，中心点 c に関して点 p と対称な点 q をブラシのもう一端とし，p と q を両端に持つ線分 l をブラシの接地線とみなす．また，トラッキングを可能とするため，線分 l の両端点 p と q をマーカーとし，ブラシが移動する際には，前フレームの座標から現在の座標までの距離が近い点を同一マーカーとして扱う．これにより，ブラシ接地面の位置や角度が変更されても，ブラシ方向の情報を保つことが可能となる．図 4-5 にマーカーの移動に伴う，ブラシストロークの描画について示す．直前のフレームのマーカー2 点と直後のフレームのマーカー2 点により四角形を構成し，この四角形の連続によりストロークを表現する．

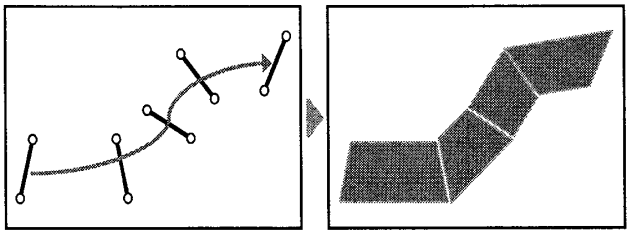

図 4-5　ブラシストロークの生成

レインボーアートに用いられる着色剤は水彩絵具に似ており，描画の際はブラシに付着した隣り合う色と混ざり合ってグラデーションとなることで，虹のような描画を可能としている．そのため，2 頂点の色を指定することで，グラデーションにより頂点間の色を補完する OpenGL の機能を用いて，レインボーアート独特の色味を再現する．図 4-6 に，3 色の色からなるブラシストロークの描画例を示す．線分 pq は前フレームのブラシ接地面，線分 p′q′ は現フレームのブラシ接地面を示す．このとき，3 色の色から接地面が構成されるため，色の変わり目は線分 pq の中点 m と p′q′ の中点 m′ からなる線分 mm′ となる．そこで，四角形 pmm′p′ と mqq′m′ の組み合わせでブラシストロークの描画を行う．これにより，滑らかでムラの無いグラデーションを表現する．

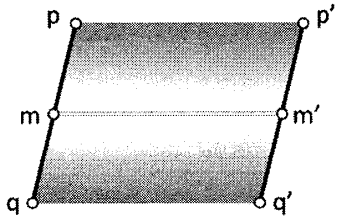

図 4-6　グラデーションの生成例

4.2.3　実行例

絵コンテに基づいて制作したレインボーアートを図 4-7 に示す．左がシミュレーション，右が実際の画材により制作したものである．また，シミュレーション実行時の様子を図 4-8 に示す．いずれの作品においても，色合いや概形が同様の描画結果となっていることが確認できる．また，サンドアニメーションのシミュレーションと同様に，処理速度はフレーム更新間隔以下となったことからも，その有効性を示しているといえる．

しかしながら，細部に着目するとシミュレーションでは描画ストロークにブレが生じている．これはインタフェースのシリコン膜と，スポンジブラシがどちらも圧力を吸収し干渉し合っているため，硬質のテーブル上の紙に描画する実際の制作と異なり，支持体に対して一定ではない圧力が加わるためと考えられる．また，実際の制作においてはブラシに色が付着しているため，新しく描画を行う際には目視での確認が可能である．対して，シミュレータではブラシをテーブルから離すとブラシの方向情報が失われるため，色の方向が意図したものと反転するという，意図しない描画が発生する場合があるといった相違点も挙げられる．

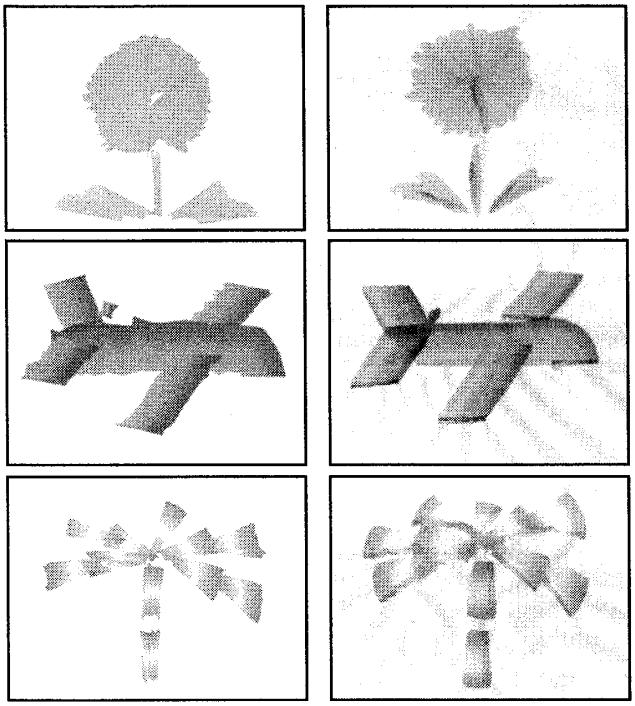

図 4-7　制作したレインボーアート

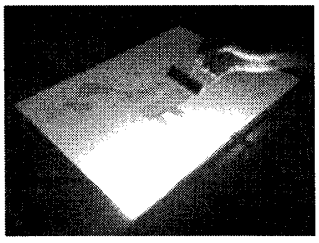

図 4-8　シミュレータによる制作中の様子

4.3　レリーフアート

テーブルにかかる圧力の深度を制作に用いた事例として，支持体に対する圧力により生成される凹凸が作品の構成要素となる，レリーフアートのシミュレーションを行う．

4.2.1　レリーフアートとは

レリーフとは浮き彫りのことをいい，ここではレリーフアートを，銅像などの完全な立体物ではなく，平面的なキャンバスをベースに圧力を加えて加工することにより，立体的な描画を行うアート作品の総称とする．代表的なアート作品として，薄い銅板に鉄筆やヘラなどで凹凸をつけて，絵に立体感を持たせる銅板レリーフが挙げられる．本研究では，著名人の手形にみられるような凹みのある形状による石膏レリーフと，ピンにより凹凸を表現するピンスクリーンという[13]，片側からのみの圧力により構成される作品をシミュレーションの対象とする．ピンスクリーンとは Alexandre Alexeïeff と Claire Parker によって提案された手法である．キャンバスが同一方向を向いた多数のピンにより構成されており，圧力のかかったピンがその強さに比例して隆起することにより，立体的な造形を可能とするアート作品である．

4.2.2 制作技法のシミュレート

レリーフアートでは，支持体となるキャンバスに対する圧力が作品の形状を生成する．このとき，キャンバスの厚みは有限であるため，ある地点に対して入力をし続けて深度を極端に深めるといった操作は行われない．そこで，キャンバスをハイトフィールドで扱い，入力があった場合にフィールドの対応する画素に深度を保持し，つぎに入力があった際に，現在の画素の深度と入力の深度とを比較し，入力の深度が大きい場合に画素に保持された深度と置き換える．これにより，厚みのあるキャンバス上での入力の特性を再現する．また，石膏レリーフにおいてはハイトフィールドの値に対応したメッシュを，ピンスクリーンにおいては対応した位置に円柱を移動させることにより，それぞれのアート作品の持つ描画における質感を再現する．

4.2.3 実行例

制作したレリーフアートを図 4-9 に示す．上段が入力画面であり，中・下段の左がシミュレーションにより制作した，右が実際の環境により制作したレリーフアートである．また，中断が手形，下段がピンスクリーンの生成結果である．なお，図では凹凸が識別しやすいようキャンバスに角度をつけてある．図が示すように，入力に対応した凹凸が再現されていることが分かる．また，シミュレーションの実行速度もフレームの更新間隔の範囲に収まり，その有効性を示しているといえる．一方で，シリコン膜の厚みの範囲でしか圧力が加わらないため，圧力が反映される最大の深度はその厚みとなり，これがシミュレーションにおけるキャンバスの厚みとなる．このため，その厚み以上の入力が想定されるアート作品については，ハードウェアの制約上，シミュレーションにおいて入力のすべてを取得することが困難となる．

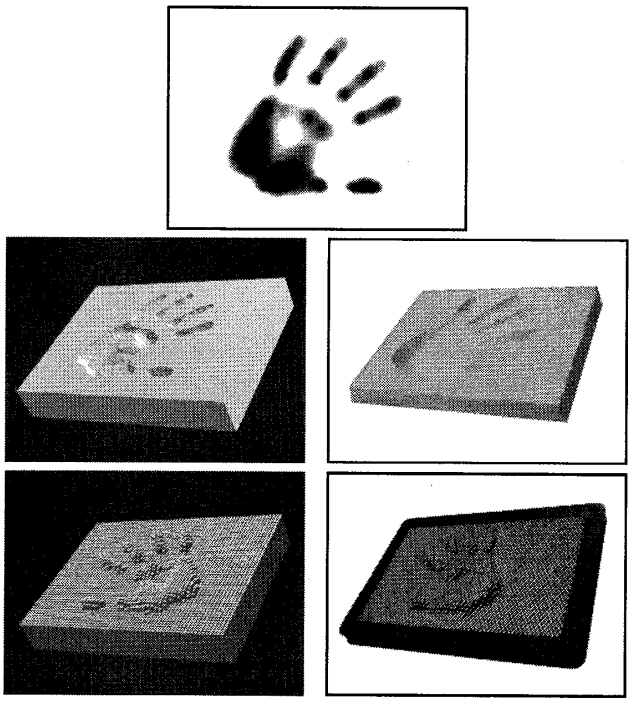

図 4-9 制作したレリーフアート

5. おわりに

本研究では，ポインタのトラッキング，入力の深度を取得可能とする，FTIR テーブルを用いた形状・圧力センシングのためのフレームワークを構築した．また，フレームワークを用いて実際のアート作品のシミュレートを行うことで，その再現性と有効性を確認した．

謝辞

本研究の一部は財団法人栢森情報科学振興財団研究助成，財団法人人工知能研究振興財団研究助成，財団法人堀情報科学振興財団研究助成による．

参考文献

[1] Hertzmann A, “A Survey of Stroke-Based Rendering,” IEEE Computer Graphics and Applications, vol.23, no.4, pp.70-81, Jul.2003.

[2] Kumiyo Nakakoji, Kazuhiro Jo, Yasuhiro Yamamoto, Yoshiyuki Nishinaka, Mitsuhiro Asada, “Reproducing and Re-experiencing the Writing Process in Japanese Calligraphy,” Proceedings of 2nd IEEE Tabletop Workshop, IEEE Xplore, pp.75-78, Oct.2007.

[3] 内藤良太, 中貴俊, 遠藤守, 山田雅之, 宮崎慎也, “メディアインスタレーション作品のための拡張可能な多軸モーションデバイスインタフェースの提案, ” 電子情報通信学会技術研究報告, vol.109, no.281, MVE2009-66, pp.29-30, Nov.2009.

[4] Kazuto Kamiyama, Kevin Vlack, Terukazu Mizota, Hiroyuki Kajimoto, Naoki Kawakami, Susumu Tachi, “Vision-Based Sensor for Real-Time Measuring of Surface Traction Fields,” IEEE Comput Graph Appl, Vol.25, No.1, pp.68-75, Jan.2005.

[5] Kazuto Kamiyama, Kevin Vlack, Terukazu Mizota, Hiroyuki Kajimoto, Naoki Kawakami, Susumu Tachi, “Vision-Based Sensor for Real-Time Measuring of Surface Traction Fields,” IEEE Comput Graph Appl, Vol.25, No.1, pp.68-75, Jan.2005.

[6] J.David Smith, T.C.Nicholas Graham, David Holman, Jan Borchers, “Low-Cost Malleable Surfaces with Multi-Touch Pressure Sensitivity,” Proceedings of 2nd IEEE Tabletop Workshop, IEEE Xplore pp.205-208, Oct.2007.

[7] David Wallin, Touchlib: an opensource multi-touch framework, http://www.whitenoiseaudio.com/touchlib/

[8] 原健輔, 浦正広, 山田雅之, 遠藤守, 宮崎慎也, 安田孝美, “FTIR テーブルによる圧力センシングのためのフレームワークの構築とアートシミュレーションへの応用,” 電子情報通信学会技術研究報告, vol.109, no.466, MVE2009-150, pp.135-140, Mar.2010.

[9] Ferenc Cako: Sand Animation, http://www.sandanimaiton.com/

[10] 浦正広, 山田雅之, 遠藤守, 宮崎慎也, 安田孝美, “サンドアニメーション風の画像生成のためのペイントツールの開発,” 電子情報通信学会技術研究報告, vol.109, no.75, pp.7-12, Jun.2009.

[11] 原健輔, 浦正広, 山田雅之, 遠藤守, 宮崎慎也, 安田孝美, “サンドアニメーションの制作技法とそのシミュレーション,” 第 25 回 NICOGRAPH 論文コンテスト論文集, I-3, Oct.2009.

[12] Laszlo Szekely, US Patent No. 5,318,171, Jun.1994.

[13] アレクサンドル・アレクセイエフ作品集, ジェネオンエンタテインメント, Jul.2006.