紙コントロールパネル：紙を用いたタッチインタフェースの作成

金子 将大[1]　田中 二郎[2]

概要：我々は，紙コントロールパネルの提案と，その作成システムの実装を行った．紙コントロールパネルは，紙に描いた図形をボタンとして扱い，図形を指でタッチすることで計算機の操作を行う．紙に図形を描き，アプリケーションの機能のショートカットなどを割り当てることで，タッチ入力インタフェースが作成される．機能の割り当てはキーイベントやマウスイベントを用いることで簡単に行える．紙コントロールパネルによって，ユーザは使いやすい入力インタフェースを自作し利用できる．

Paper Control Panel: Making Paper-based Touch Interface

Kaneko Masahiro[1]　Tanaka Jiro[2]

Abstract: We present the Paper Control Panel, which is a paper-based touch interface for operating a computer. We have developed the prototype system to implement and use the interface. Figure 1 shows several examples. This interface is based on drawn closed shapes. When the user touches one of these shapes, the computer calls a function assigned to the shape. The user can design shapes and assign functions to them.

1. はじめに

計算機操作用の入力インタフェースには，マウスやキーボードをはじめ，さまざまなものが存在する．入力インタフェースを利用するユーザや操作対象のアプリケーションは無数に存在するため，それらの特徴に合わせた専用のものが存在すれば，作業効率やユーザエクスペリエンスの向上につながると考えられる．しかし，これらの多種多様な要求に対して，既製品の入力インタフェースだけでは対応が難しい．また，専用入力インタフェースの自作は，専門の知識が必要であり，コストもかかるため現実的ではない．

本研究では，ユーザ自身が自分専用あるいはアプリケーション専用の入力インタフェースを作成可能な環境を目指し，紙とペンを用いて簡単に作成可能な入力インタフェースである紙コントロールパネルを提案する．また，紙コントロールパネルの作成と使用を行うためのプロトタイプシステムを実装した．図 1 に紙コントロールパネルの例を示す．

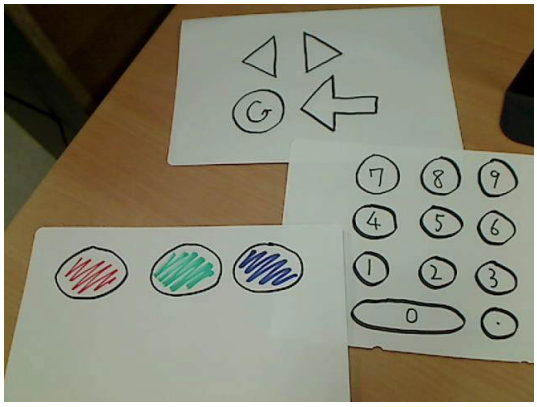

図 1　紙コントロールパネル

[1] 筑波大学大学院 システム情報工学研究科 コンピュータサイエンス専攻
Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba
[2] 筑波大学システム情報系
Faculty of Engineering, Information and Systems, University of Tsukuba

2. 関連研究

2.1 手書きによる GUI の設計

GUI を作成する際に通常はプログラミングを行う必要があるが，GUI のデザイナーはプログラミングの知識がないことが多い．そのため，GUI のデザインをスケッチすることで，自動的に GUI を作成する研究が数多く行われている [1] [2] [3].

Holzmann らは紙とデジタルペンによりモバイル端末用 GUI のプロトタイプを作成するシステムを開発した [4]. デジタルペンによって書かれたスケッチを認識し，インタラクティブなモバイル端末用 GUI を自動作成する．

Li らはペーパープロトタイピングの様子から自動的に Web ページを作成する FrameWire を開発した [5]．Web デザインのスケッチをタッチで操作する様子の動画からリンクの位置や対応するページ遷移を満たす HTML を自動的に作成する．

本研究は，スケッチすることでインタフェースを作成する点で関連するが，GUI を生成するのではなく，直接スケッチをインタフェースにする点で異なる．

2.2 紙を用いたインタフェース

Keys は紙に描いたウィジェットのスケッチを用いてシンセサイザを操作する SketchSynth を開発した [6]．ユーザ自身がボタンやスライダ，トグルスイッチなどを描くことで作成し，そのスケッチ自体を実際に指で操作することができる．

竹川らは紙上に描いた任意形状のオブジェクトに指で触れて音を鳴らす絵楽器のためのプロトタイピング支援システムを開発した [7]．導電性インクを用いて描いた図形に，様々な機能を持つピンを接続して絵楽器を構築する．

本研究は紙に描いた図に指で触れる入力インタフェースを作成する点で関連しているが，一般的なアプリケーションを対象としている点や，機能の割り当て方法も研究対象としている点で異なる．

3. 紙コントロールパネル

紙コントロールパネルは，本研究が提案する紙を用いた入力インタフェースである．紙にボタンとする図を描き，機能を割り当て，指でタッチすることで，割り当てた機能が実行され計算機を操作することができる．また，計算機を経由してネットワークで接続された家電などの操作も可能となる．

紙コントロールパネルは一般的な入力インタフェースにはない以下のような紙特有の特徴を持つ．

- 切り貼りや折り畳みによるサイズ変更が可能である
- 薄さと軽さによって高い携帯性を持つ
- 印刷・コピーによる複製が可能である
- 不要になった際の破棄が容易である

本研究では紙コントロールパネルを作成し使用するためのプロトタイプシステムを作成した．本システムは紙とペンの他に，紙を映すように設置された USB カメラを用いる．また，紙へのタッチの認識に手の影を用いるため，ライトスタンドなどの光源が紙の上部にあることが望ましい．図 2 にシステムの概観を示す．

次に紙コントロールパネル作成の流れとして，図形の描画と機能の割り当てについて述べる．

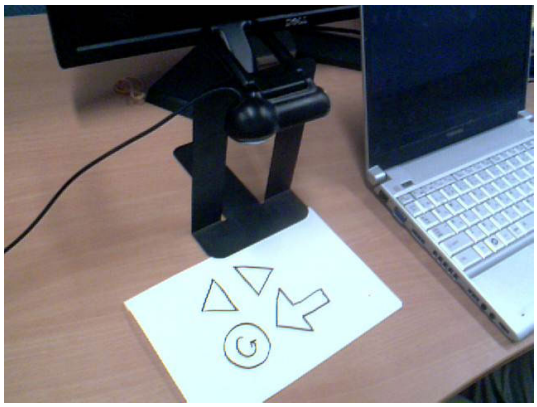

図 2 システムの概観

3.1 図形の描画

まず，紙にペンでボタンとして扱う閉じた図形を描く．図形を描く場所や，形，大きさは自由であり，図形の内部にはメモや書き込みことが可能である．これにより，図形の外見に意味を持たせて，割り当てられた機能が一目で分かるインタフェースを作成可能である．

3.2 図形の登録

図形への機能の割り当ては管理用のフォームアプリケーションを用いて行う．メニューバーからボタンに用いる色を選択し，カメラプレビュー上で図形を左クリックすると，選択された図形に色がつき，ボタンとして登録される．この時点では，ボタンにはシステムが定めたデフォルトの機能が割り当てられている．

図 3 に管理用フォームを示す．図右側のフォームにカメラ画像が表示されており，3 つの登録された図形に色がつけられている．

3.3 機能の割り当て

登録済みの図形を右クリックすると機能割り当て用のコンテキストメニューが表示される．メニューにはキー操作とマウス操作のほか，あらかじめ登録された機能が表示される．図 4 に右クリックメニューの例を示す．メニュー最上部には選択された図形に対応する色が表示され，以降

キー操作・マウス操作割り当て用の項目，その他のあらかじめ登録された機能が続く．

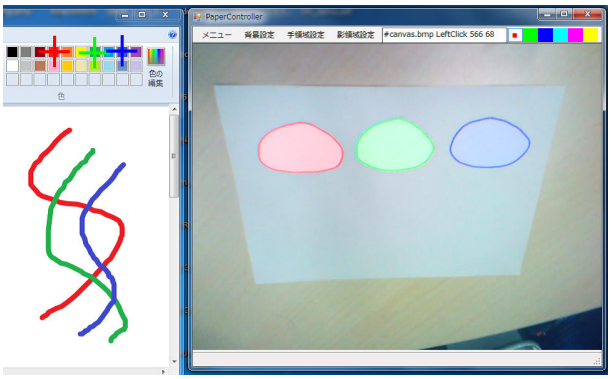

図 3 紙コントロールパネル管理用フォーム

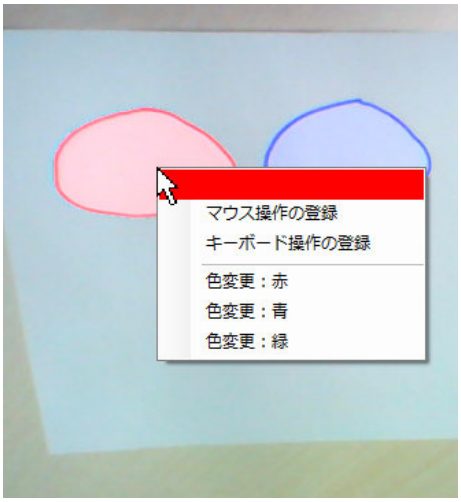

図 4 機能割り当て用メニュー

3.4 キー操作の割り当て

メニュー選択後に表示されるテキストボックスに実行したいキー操作を文字列で指定することで，図形へのタッチによりそのキー操作が実行されるようになる．これにより，テキスト入力やショートカットキーを介したアプリケーションの操作が可能になる．

指定する文字列のフォーマットには.Net Framework の SendKeys クラスのものを用いた．例えば，「A」なら A キーイベント，「+{TAB}」なら Shift を押しながらの Tab キーイベントが割り当てられる．

3.5 マウス操作の割り当て

メニュー選択後に操作の対象とするアプリショーンに対してクリックやドラッグを行うことで，図形にその操作が割り当てられる．対象となった座標には図形の色と対応したマーカが表示される．座標は操作対象のアプリケーションの位置を基準として記録されるため，ウィンドウの移動を行っても正しく動作する．これにより，GUI を介したアプリケーションの操作などが可能になる．

図 3 ではペイントソフトのカラーボックスに対する左クリックが割り当てられており，図左側のカラーボックス上に表示されている十字が左クリック座標のマーカである．

3.6 登録済み機能の利用

あらかじめアプリケーションごとの利用可能な機能について定義したリストを用意することで，コンテキストメニューから選択し，その機能を図形に割り当てることができる．リストは機能の名前・対象のアプリケーション・行われる処理を 1 セットとし，テキストファイルに列挙したものを読み込む．

機能リストを定義することで，連続的なマウス操作や別のアプリケーションの実行など，フォームを介して操作を指定する場合よりも，高度な操作を利用することが容易になる．例えば，対象のアプリケーションに熟練したユーザが作成した機能リストを不慣れなユーザが使うことで，途中の操作を意識せずにその機能を利用することができる．また，機能リストはインタフェースを作成した際に自動で生成されるため，1 度割り当てた機能を再度割り当てる場合にも利用できる．

また，図形登録直後に割り当てられるデフォルトの機能も，同様にリストを用意することで色ごとに指定することができる．例えば，各図形に持たせる機能はあらかじめ決まっており，デザインを試行錯誤したい場合は，デフォルトの機能としてあらかじめ記述しておくことで，図形の登録だけでインタフェースを作り直すことができる．

3.7 処理コマンドの記述

ボタン登録後または右クリック後に，対象のボタンに登録されている機能を示すコマンドがメニューバー中央のテキストボックスに表示される．これを編集することより詳細な機能の割り当ても行うことができる．このテキストボックスや機能リストの処理は，コマンドプロンプトで用いられる Windows コマンドによって記述する．

ここで使用するための実行ファイルをいくつか用意しており，例えば，「LeftClick 10 10」というコマンドを記述することで，(10, 10) 座標への左クリックを行うことができる．また，テキストボックスの先頭に#と記号とアプリケーションの名前を入力することで，操作対象のアプリケーションを指定することができる．例えば「#canvas」と記述した場合，アプリケーションのタイトルに canvas という文字列を含むアプリケーションを対象とする．

対象のアプリケーションを指定した場合は，そのアプリケーションにフォーカスが当たっている場合にのみ機能が実行されるようになる．また，マウス操作の割り当てと同様に，座標を用いる処理はそのアプリケーションの位置を

基準とした相対座標によって動作するようになる．

3.8 記録と復元

1 度作成した紙コントロールパネルは描いた図形と割り当てた機能が記録される．他の紙を使用するために入れ替えカメラから外れた後も，再度カメラ前にセットすることですぐに使用を再開できる．

また，紙コントロールパネルは作業ごとに使い分けられることを想定し，最後に使用していた際のウィンドウの状態を記録しておき，使用が再開された際にウィンドウの状態も復元するようにした．これにより紙コントロールパネルを入れ替えるだけで，作業の切り替えを行うことができる．例えば，ペイントソフト用の紙コントロールパネルを作成しておき，カメラ前にセットして使用を再開すると，ペイントソフトが起動し前回編集していたファイルが読み込まれる．既に起動していた場合は，ウィンドウを前面に移動しフォーカスする．

4. 紙コントロールパネルの実装

システムの実装は C#を用いて行った．各機能は Windows コマンドによって実行される．また，Windows API によってウィンドウの情報の取得やマウス操作の実行を行っている．

OpenCVSharp によって USB カメラの画像を処理することで，図形の認識やタッチの判定を行っている．本システムで用いられている画像処理として，各領域の抽出，図形のラベリング，指先の認識などがある．

システム利用時に行われる処理を順に説明していく．

4.1 各領域の抽出

カメラ画像から紙，手，影の領域を抽出する．

まず，カメラに手が映っていない状態でシステムを起動するか初期設定ボタンを押下する．このとき，背景差分取得のための背景初期化を行い，カメラ画像の中心部に位置する白い矩形部分を紙の領域として記憶する．

初期化後，フレームごとに背景差分を取得し，差分中に肌色の量を元に手が映っているかどうかを判断する．手が映っている場合は，手と手の影の領域を抽出する．肌色を手の領域，肌色でない部分を影の領域とする．例として図 5 に元のカメラ画像，図 6 に手領域の画像，図 7 に影領域の画像を示す．

4.2 図形のラベリング

手が映っていないときにフォームのプレビューが左クリックされた場合，紙に描かれた図形の認識を行いボタンとして登録する．カメラ画像から抽出した紙領域の内部にある図形の輪郭線を求め，クリック座標を含む輪郭の内部をボタンの有効領域として登録する．登録された図形の領域ごとに塗り分けた画像を保持することでラベリングを行う．図 8 に図形のラベリング画像を示す．

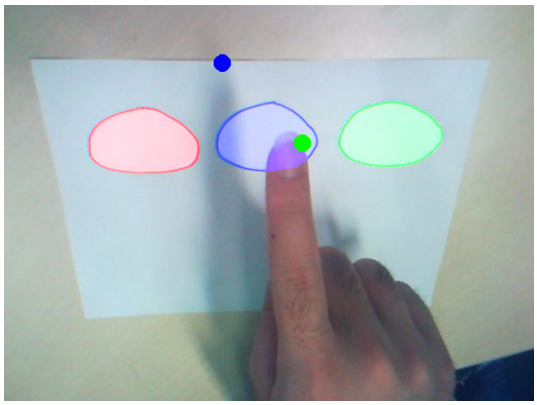

図 5 元のカメラ画像

図 6 手領域の画像

図 7 影領域の画像

4.3 機能の割り当て

手が映っていないときにフォームのプレビューが右クリックされた場合，登録された図形への機能の割り当てを行う．ラベル画像におけるクリック座標の色を取得し，対応する図形を対象として選ぶ．その後，右クリックメニューから選択されたアイテムやテキストボックスの記述など決定した Windows コマンドと，対象の図形を紐付けて記憶する．

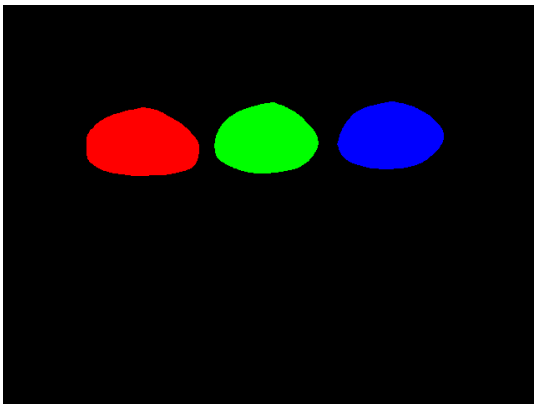

図 8 図形のラベリング画像

4.4 指先の認識

抽出した手領域から輪郭線を求め，その輪郭線上に位置する点を一定間隔ごとに取得する．それらの点の中から位置や角度を基に最もそれらしいものを選び指先の座標とする．同様に影領域における指先の影の座標も取得する．図 5 では指先の座標に緑の点，指先の影の座標に青の点が表示されている．

取得した指先の座標と指先の影の座標を用いてタッチ判定を行う．紙から手が離れている場合，2 点の距離は長くなるが，紙に近づくほど短くなる．これを利用し，2 点間の距離の変化から紙と指が接触しているかどうかを判定する．

4.5 機能の実行

図形がタッチされたと判定された場合は，ラベリング画像からタッチ座標の値を取得し，その値に対応した図形に紐付けられたコマンドを実行する準備をする．対象アプリケーションが指定されていない場合は，そのまま実行する．対象アプリケーションが指定されている場合は，Windows API によってアクティブウィンドウを調べ，それが対象アプリケーションならばコマンドを実行する．また，クリック操作などの座標を用いるコマンドの場合は，Windows API によってアクティブウィンドウの位置を調べ，操作対象の座標を再計算する．

4.6 紙の情報の保存

紙を移動しカメラに映らない状態にすると，システムはその紙の情報として，ラベル画像・ボタン登録時にクリックした座標・割り当てた機能・対象アプリケーションの状態を保存する．ラベル画像とクリック座標は紙領域の 4 隅の座標を用いて射影変換を行うことで正規化する．これらの情報は紙を再度映した際の識別に用いる．対象アプリケーションの状態として，表示位置やウィンドウサイズなどを保存しておく．情報を保存するとシステムは紙の識別待機状態に移行する．

4.7 紙の識別

同じ紙あるいは別の紙を再度カメラに映すと，システムはその紙が保存された情報と一致するかを確認する．まず，映された紙の領域を抽出し，4 隅の座標から保存したラベル画像とクリック座標の射影変換を行う．現在映っている紙に変換したクリック座標を用いた場合のラベル画像を生成し，保存しておいたものと比較する．比較した結果一致していると判定された場合は，登録された図形や機能，アプリケーションの状態を復元して同じように利用できる状態にする．

5. タッチパネルによる試作システム

紙のシステムの試作として，タッチパネル上で動作する同様のシステムを実装した．タッチパネルは紙のシステムと同じように，図を描いて触れるインタラクションを利用できるため採用した．機能を追加するにあたり，まずは実装が容易なタッチパネル上で実装した．図 9 にシステムのスクリーンショットを示す．本章ではタッチパネルで実装した機能のうち，今後紙のシステムにも実装予定のものについて説明する．

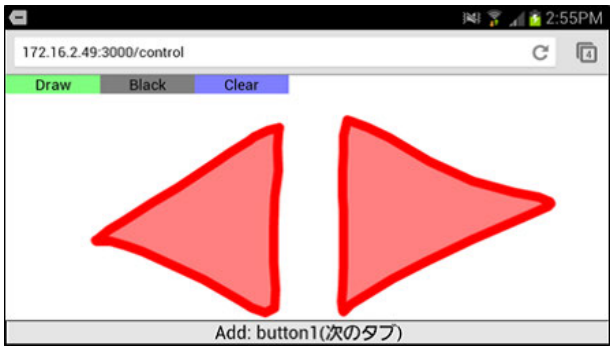

図 9 タッチパネル版画面

5.1 機能と対応する図の設定

機能リストに記述された機能に，対応する図を指定可能にした．対応する図形が設定されている場合，その図形を描くだけで対応する機能が自動的に割り当てられる．計算機を操作する必要があった機能の割り当てが省略され，図を描くだけでインタフェースの作成が可能になる．

例えば，ブラウザのショートカットキーを定義した機能リストが存在し，初心者ユーザがリストから作られた図 10 のような表を見ながらインタフェースの作成を行った場合，「戻る」機能を利用したい場合は，対応する図形である左向きの矢印を描けば，それがそのまま「戻る」ボタンとして動作するようになる．このとき，このユーザが「戻る」のショートカットキーが BackSpace キーであることを知らなくても利用できる．

インタフェースの作成の際に，行われる処理と対応する

図形の紐付けを行うことができるため，機能の名前を加えることで機能リストを作成することができる．熟練したユーザがインタフェースを作成した際に生成される機能リストを，不慣れなユーザが使うという利用方法を想定している．

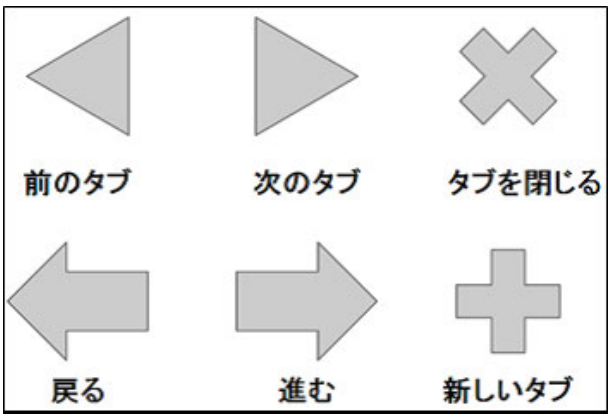

図 10 ブラウザの機能表

5.2 スライダ

ユーザが利用できるウィジェットとしてスライダを追加した．図 11 にタッチパネル版でスライダを設置した画面を示す．スライダは線を引いた後に，その線の端に閉じた図形を描くことで設置される．図で示しているように，スライダのノブや線の形状は問わない．

スライダ上で指を動かすと，移動量と位置に応じてスライダに割り当てられた機能が実行される．例えば，スライダにスピーカの音量を調節する機能を割り当てれば，スライダ上で指を動かすことで音量が変化するようになる．タッチパネルでは現在値に応じてノブが移動するように実装した．しかし，紙に描かれたノブを動かすことはできないので，ノブの代替として硬貨などを用いることも考えている．

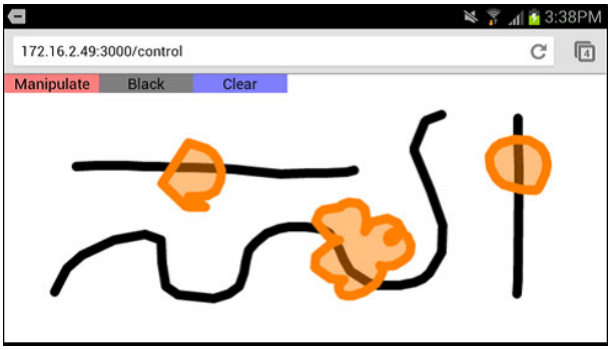

図 11 タッチパネル版スライダ

6. まとめ

我々は紙でできた入力インタフェースである紙コントロールパネルを提案し，その作成と使用を行うためのシステムを実装した．ユーザは紙に図形を描きマウス・キーボードマクロなどの機能を割り当てることで紙コントロールパネルを作成する．図形にタッチすることで割り当てた機能が呼び出され，計算機の操作を行うことができる．

今後はタッチパネルで実装した機能を紙でも利用可能にし，被験者実験によって評価を行う．

参考文献

[1] J. A. Landay, and B. A. Myers. Sketching Interfaces: Toward More Human Interface Design. IEEE Computer, Vol. 34, Issue 3, pp. 56-64, 2001.

[2] A. Coyette, S. Faulkner, M. Kolp, Q. Limbourg, and J. Vanderdonckt. SketchiXML: towards a multi-agent design tool for sketching user interfaces based on USIXML. TAMODIA '04 Proceedings of the 3rd annual conference on Task models and diagrams, pp. 75-82, 2004.

[3] V. C. V. B. Segura, S. D. J. Barbosa, and F. P. Simoes. UISKEI: a sketch-based prototyping tool for defining and evaluating user interface behavior AVI '12 Proceedings of the International Working Conference on Advanced Visual Interfaces, pp. 18-25, 2012.

[4] C. Holzmann, and M. Vogler. Building interactive prototypes of mobile user interfaces with a digital pen. APCHI '12 Proceedings of the 10th asia pacific conference on Computer human interaction, pp. 159-168, 2012

[5] Y. Li, X. Cao, K. Everitt, M. Dixon, and J. A. Landay. FrameWire: a tool for automatically extracting interaction logic from paper prototyping tests. CHI '10 Proceedings of the 28th international conference on Human factors in computing systems, pp. 503-512, 2010

[6] B. Keys. SketchSynth: A Drawable OSC Control Surface. Interactive Art and Computation Design 2012, Carnegie Mellon University

[7] 竹川佳成, 福司謙一郎, Machover Tod, 寺田 努, 塚本昌彦. 絵楽器の設計段階におけるプロトタイピング支援システムの設計と実装. インタラクティブシステムとソフトウェア XIX: 日本ソフトウェア科学会 WISS2011, pp. 60-65, 2011