ContextDial ー 回転板を用いた新しいインタフェース

松本　明浩†　　増井　俊之†

本論文では回転円板を用いた新しいインタフェース手法を提案する．実世界またはパソコン画面上の円板に触れてから回転させることによって GUI のスライダやメニューのような操作を行なうが円板に触れる位置により異なる動作を指示することができるため，ひとつの円板だけを使ってテレビのチャンネルを切り換えたり音量を調整したり各種の制御を行なうことが可能である．またコンテクストに応じて動作を切り換えることによりアプリケーションや環境に応じて様々な操作を割り当てることが可能である．

ContextDial

AKIHIRO MATSUMOTO†　　TOSHIYUKI MASUI†

We propose a new wheel-based interface widget "ContextDial" with which users can easily control various parameters by touching and rotating a wheel. Since users can select different functions by touching different portions of the wheel, a single wheel can be used as a controller with multiple functions required for complicated modern consumer devices. Users can implement a ContextDial either on a GUI window or using a circle plate.

1. はじめに

これまで様々な電子機器が登場，普及し急速に進歩していった．だが，その高機能化に伴って設定や操作が PC のように複雑化し，ユーザの負担となりはじめている[1]特にテレビや DVD プレイヤーなどの情報家電はその傾向が顕著である．例えばテレビのリモコンは機能が増える度にボタンの数を増やす仕様が ARIB によって規格化されている．しかしその規格は菅原ら[2] のいうように機能がそれほど多くなかった時には有効であったが，多機能化した今では普段はあまり使われないような機能が重要な機能と混在しユーザを迷わせる原因となっている．さらに GoogleTV[3] などの Web と繋がった多用途かつ膨大なコンテンツをもったサービスも今後登場していく中で現状の単純にボタンを増やしていく方式ではリモコンが大きくなってしまう．このような点で現状のリモコンのインタフェースはこれからの情報家電やコンテンツ検索には向いていない．大規模なデータでもユーザが簡単に閲覧できる仕組みが必要である．

また様々な情報機器が多機能化する過程でリモコンが増えたことにより管理が面倒になっているという問題もある．機器とリモコンが 1 対 1 の関係でリモコン同士の互換性がないため，全ての機器のリモコンもなくさないように保管しておかねばならず，その対応付けもユーザが判別・管理しなければならない．どのリモコンもそれぞれの機器に最適化して開発されたものであるため，ユーザは新しく機器を導入するたびにその操作方法を学ぶ必要がある．

これらの冗長性も，もし統一されたインタフェースで操作することが出来れば楽になる可能性があった[1] はずである．

このようにリモコンのインタフェースは

- 大規模な情報を横断的に閲覧できない．
- 操作が統一されていない．

という問題を抱えている．これらの問題を解決するにはテレビなどの膨大な情報量をもったコンテンツを含め様々な電子機器をシンプルに操作できる統一されたインタフェースが必要である．

そこで本論文では回転板を用いた汎用的なインタフェース手法「ContextDial」を提案する． ContextDial は実世界またはパソコン画面上の円板に触れてから回転させることによって GUI のスライダやメニューのような操作を行なうインタフェースで，円板に触れる位置により異なる動作を指示することができるため，ひとつの円板だけを使ってテレビのチャンネルを切り換えたり音量を調整したり各種の制御を行なうことが可能である．スナッピングの動作を利用して大きく値を変更したり微調整を行うことができるため大規模な情報であってもユーザは素早く簡単に閲覧すること

† 慶応義塾大学環境情報学部
Faculty of Environment and Information Studies, Keio University

ができる．

またコンテクストに応じて動作が自動的に切り替わり必要最小限の機能だけが円板上に表示されるため，ユーザは項目を探す手間なく操作することができる．

2. システム

本章では，ContexDial のシステムとそのプロトタイプについて説明する．

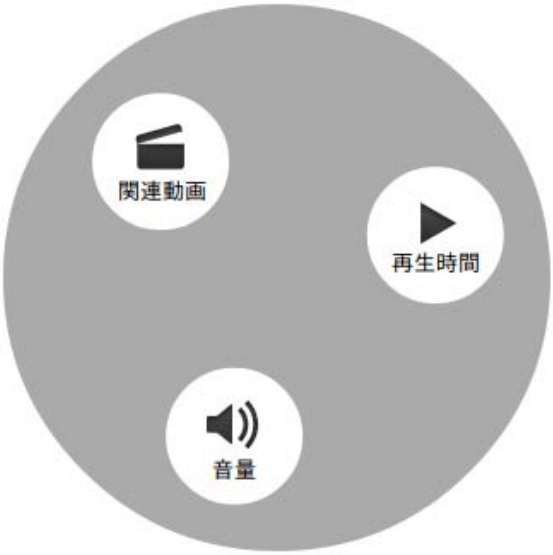

図1 ContextDial

2.1 回転による動作指示

ContextDial は円板を回転させることによって値を変更できる．図 1 のように円板の上に複数の円が等間隔に並んでおり，音量や再生時間などの機能がそれぞれ割り当てられている．その円に指で触れ円状に動かすことで円板全体が回り，対応する機能の設定値が変更される．構造は GUI のスライダやスクロールバーに近いが円の回転によって値を移動できるため，母数が多いコンテンツであってもスライダやスクロールバーのようにボタンが小さくならないという特徴がある．また触れる場所に応じて操作できる機能が変わるためひとつの円板だけでさまざまな操作を行うことが出来るという点でも違いがある．

値の調節には SmoothSnap[4] の機能を利用しており円板を大きく回すことで値が重要なポイントまでスナッピングされ，小さく回すことで細かい粒度で値が連続的に変化する．

2.2 コンテクストに応じた切り替え

ContextDial は操作対象の機器の状況に応じて機能の配置を動的に変更する．動画の再生やテレビ放送の視聴などのコンテクストに対応する操作項目を Web 上にストックしており，コンテクストの変更命令を受け取るたびに適切な操作セットを引き出し配置する．これによってそのコンテクストに必要な最小限の項目だけが配置され，ユーザは操作したい項目を簡単に選ぶことができる．

2.3 利用例

このインタフェースを利用したプロトタイプとその構成について述べる．このプロトタイプは Web ブラウジングを ContextDial によって行えるようにするもので，リモコン上の円板を動かすことでスクロールや現在見ている Web ページ内のリンクを辿るなどブラウザにおける主要な動作を行える．動画を再生するページにアクセスすると，それに応じてコントローラも操作項目が自動的に切り替わり，動画のシークや音量を制御可能になる．

今回のプロトタイプではスマートフォンをリモコンとして使い Web ブラウザを操作する．図 3 のようにブラウザとリモコンはどちらも Web サーバに接続しており，サーバを介してお互いの状況を同期する．リモコンはサーバが描画する Web ページ上にあり WebSocket 通信が可能なスマートフォンならば URL を入力するだけで利用することができる．またユーザ ID によるログインのシステムを実装しており，ユーザがリモコンを使う際には URL にアクセスした後に自身で登録したユーザ ID とパスワードを入力しログインする．これにより Web サーバはどの端末からのアクセスでもユーザを識別することが出来るようになり，リモコンの紛失によって機器が操作できなくなるというリスクがなくなる．

ログイン後に Web ページへアクセスし，リモコンの円板をタッチして動かすとスクロールやページ内のリンクにフォーカスを合わせるなどのリモコン操作をブラウザが反映する．ブラウザはタブが切り替わる，現在アクセスしているページの URL が切り替わるなど，状況が変わるたびにリモコンに情報を送っており，操作項目の変更が必要な場合にはリモコンに命令を送り配置を変更させるようになっている．コンテクスト毎の操作項目のセットはデータベースに保存されており Web サーバが管理をしている．

プロトタイプでは動画と Web ページの閲覧用の操作セットが用意されており図 2 のように Web ページの閲覧中にはスクロール/ リンク/ タブ/ 履歴のボタンが表示され，動画を再生中には再生時間/ 音量/ 関連動画が表示される．このプロトタイプは対応する操作コンテクストが増えた場合でも Web インターフェースを通じてデータベースに加えることでシステムを

拡張することなく操作セットを追加できるので，田中ら[5] の研究のような Web 経由で通信するホームネットワークシステムの用途などに使うことが出来る．

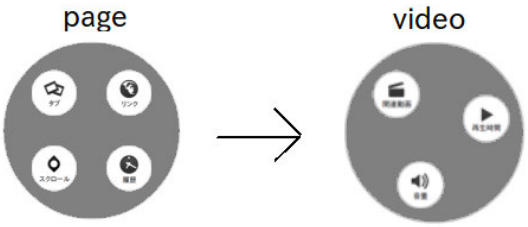

図2 コンテクストに応じた切り替え

3. 実装

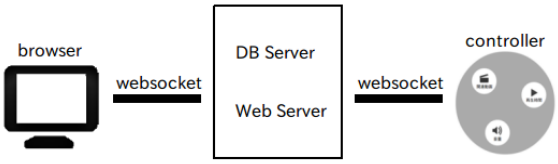

図3 プロトタイプのシステム構成

上記のプロトタイプの実装について述べる．

3.1 サーバ

node.js[6] フレームワークを使用し，Web サーバとして稼働させている． node.js はノンブロッキング IO を用いることで大量のアクセスを捌くことに長けたフレームワークであり WebSocket などを用いたリアルタイムアプリケーションに適している．リモコンからの値の送信などの頻繁に使われる通信には WebSocket を利用している．操作セットやユーザの情報を保存するデータベースには mongoDB[7] を利用した．

3.2 クライアント

リモコンのビューは javascript を使い HTML5 の canvas 上に描画している．jQuery[8]と自作のプラグインを利用し図形の配置や色などをオプションだけで変更可能になっている．ライブラリは MIT ライセンスで公開されている[9]．Web ブラウザには GoogleChrome を使用しており，javascript を使いリモコンとの通信やページの書き換えを行う chrome 拡張を作成した．

4. 議論

4.1 応用

今回のプロトタイプは PC 上の Web ブラウジングに特化したものだが ContextDial は様々な電子機器に汎用的に使えるインタフェースである．

テレビのリモコンとしてチャンネルの切り替えや音量の調節に行える他，スライドショーの閲覧やエアコンの操作などに使うことができる．これらの操作セットは前述の通り Web インターフェースを通じて登録することが出来るので様々な機器やコンテンツを統一したインタフェースで操作することが出来るプラットフォームとしても運用できる．

スマートフォン端末で動作するように実装されているので RFID タグとの対応付けによって家電製品にタッチすることで適切なリモコンインタフェースを起動するアプリケーションなどにも応用可能である．

また GUI 上のインタフェースとしてだけでなく実世界の円板を用いることでより直感的なインタフェースの実現が期待できる他，手が水に濡れていても操作できるようになれば台所で操作するなどの別の活用が考えられる．

4.2 問題点

ContextDial は動きの大きさに伴い移動の粒度が自動的に変わるので粒度を最初の状態に戻すためには一旦，円板から指を離さなければならない．そのため操作するユーザは粒度の変更の度に円板から指を離してまたタッチするという動作を行わなければならず，操作がやや冗長になってしまう．

また，スナッピングの機能を使って値を一度に大きく移動させると，その移動に伴い操作するユーザへのフィードバックも速く更新されてしまい目視が追いつかず，思うように調節できないという問題もある．

5. 関連研究

回転によって値を制御したり GUI を操作するものは様々なものが研究されているが[11,12,13]，操作する機能の切り替えが明示されておらず要求される操作が複雑な場合に対応できないという問題がある．

ThePieSlider[14]は大きな値の移動や微調整をする機能や動的にメニューを書き換える機能の実運用などはないが同じようなインタフェースを提唱している．円上の触れる位置と動きによって入力を指示するインタフェースは Frode ら[15]によって提唱されているがビジュアルによるフィードバックがないのでユーザは入力したい対応する動きを覚えなければならず学習コストが高い．

ContextDial は特定のフォーマットに則って操作項目を記述しておくことで同じインターフェースで様々な利用用途に対応するという特徴を持っているが，イ

ンタフェースを統一することで操作するユーザの負担を減らすという工夫は Jeffrey[16] や jan[17]の研究にも近いと言える．

6. 結論

ContextDial を用いれば円板上に等間隔に配置されたボタンを「触って回す」という統一したインタフェースの上で様々な電子機器を操作することが出来るようになる．また，スナッピングの動作を用いることで大規模なコンテンツでも簡単に閲覧することできる．電子機器の多機能化に伴い顕著になりはじめたリモコンの不便さを回転板のインタフェースを用いることで解消している．

これまで電子機器の操作インタフェースはハードウェア上のものが多くあったが電子機器が多機能化した現在，リモコンの UI もソフトウェアと融合することで柔軟な表現が可能になり，より便利なものになって行くと考える．今後は議論の章であげたものに加えコンテクストの切り替わりを知らせるフィードバックやユーザがボタンを触った際のフィードバックなどユーザが快適に操作するためのインタラクションを検討していく．

参 考 文 献

1) The Wall Street Journal Online. Gomes, L.Appliances Have Become Like PCs: Too Complex for Their Own Good, 2003.
2) 菅原 善博，矢島 美希，高木 真一，富永 英義. ユーザ情報を用いたテレビリモコンシステムの提案，映像情報メディア学会技術報告， 社団法人映像情報メディア学会， 2009.
3) Google TV. http://www.google.com/tv
4) 増井俊之. SmoothSnap: スナッピングにもとづく微調整可能な GUI 部品．ソフトウェア科学会インタラクティブシステムとソフトウェア(WISS2009):2009.
5) 田中章弘，中村匡秀，井垣宏，松本健一．Web サービスを用いた従来家電のホームネットワークへの適応， 電子情報通信学会技術研究報告，Vol.105, No.628, 067-072, 2006.
6) nodejs. http://www.nodejs.org/
7) mongoDB. http://www.mongodb.org/
8) jQuery. htttp://jquery.com/
9) en. htttps://github.com/myatsumoto/en
10) Gonzalo Ramos, Ravin Balakrishnan . Zliding: fluid zooming and sliding for high precision parameter manipulation. in Proceedings of the 18th annual ACM symposium on User interface software and technology, ACM Press, 143-152, 2005 .
11) Graham Smith, M.C. Schraefel, and Patrick Baudisch. Curve dial: eyes-free parameter entry for guis. In CHI ’05: CHI ’05 extended abstracts on Human factors in computing systems, ACM Press, 1146.1147, 2005.
12) Tomer, M., & John, F. Navigating Documents with the Virtual Scroll Ring. UIST, 57-60, 2004.
13) Alexander Kulik, Andre Kunert, Christopher Lux, Bernd Frohlich, ThePieSlider: Combining Advantages of the Real and the Virtual Space, Smart Graphics, 93-104, 2009.
14) Frode Eika Sandnes, Yo-Ping Huang. From Smart Light Dimmers to the IPOD: Text-Input with Circular Gestures on Wheel-Controlled Devices, 2007.
15) J. Nichols, B. A. Myers, B. Rothrock. "UNIFORM: Automatically Generating Consistent Remote Control User Interfaces," in Proc. CHI 2006, Montreal Quebec, Canada, ACM Press, 611-620, 2006.
16) Jan Hess, Guy Kustermann, Volkmar Pipek. Preemote: a user customizable remote control, CHI, 3279-3284, 2008.