

MonoGoto : モノに機能を吹き込む情報操作環境の提案

濱田 雅大^{†1} 外村 佳伸^{†2}

概要 : 本報告では、モノに様々な情報や機能を仮想的に持たせて操作することを可能にするシステム MonoGoto を提案する。たとえば、あたかもモノに音声を吹き込み、置いた場所に応じてその音声を再生したり、モノの位置を記録し、その位置への配置を鍵としたりするなど、モノと場の関係を定義することで様々な機能を果たすことができる。本システムの特徴は、利用者がディスプレイを一切見ることなく、モノを動かすことだけで操作を実行できることにある。実装したプロトタイプシステムでは、操作面上方に設置したカメラでモノの位置と角度を取得すると共に、音声の入出力機能を備えている。MonoGoto を用いることで、日頃、人が普通に行っているモノを介した思考やコミュニケーションを支援、拡張することができると考えている。

MonoGoto : Virtual Function Embedding into Objects

Masahiro HAMADA^{†1} Yoshinobu TONOMURA^{†2}

Abstract: This report proposes a system called “MonoGoto” which allows us to virtually embed functions into objects and manipulate them. For example, you can record your voice message into an object and replay it as if the object were a voice recorder. Another example is to use the system as a kind of lock which only opens if all conditions defined by object positions and each orientation are exactly matched. Like these examples, it is possible to define various functions and associated manipulations based on attributes of objects and the field. Most important feature of MonoGoto is that you can use it just by handling daily objects without looking at a computer display. The prototype system has a web camera above the desk to capture entire working field on the desk and a microphone and speaker for voice recording and replaying respectively. We hope that MonoGoto could be a useful tool to expand our ways of daily thinking and communication.

1. はじめに

本報告では、モノに様々な情報や機能を仮想的に持たせ、操作することを可能にするシステム MonoGoto を提案する。あたかもモノに音声を吹き込み、置いた場所に応じてその音声を再生したり、モノの位置を記録し、その位置への配置を鍵としたりするなど、モノと場の関係を定義することで様々な機能を果たすことができる。このシステムの重要な特徴は、利用者がディスプレイを一切見ることなく、モノを動かすことだけで操作を実行できることにある。

モノと人のインタラクションを発展させる研究のルーツに、石井らによる Tangible Bits[1]がある。この研究では、実体の無いデジタル情報に、知覚し直接手で触り操作することのできるユーザインタフェースを与えている。従来、キーボードやマウスを利用することで間接的にしか操作できなかった情報を、モノを介して直接的、直感的に手で操作できる様にしており、人にとって実際のモノが、扱うインタフェースとして重要であることを示したものである。

同様にモノを扱ったインタラクションの研究は多数行われており、たとえば福地らの Ficon[2]、安藤らの StackBlock[3]、矢敷らの電力重畳通信機能を持つブロック型デバイスとその実装[4]、籾内らのバーチャルスタジオにおける現実物体を介した仮想物体操作[5]などがあり、それぞれモノとその操作に特有の意味を持たせている。

本報告では、身のまわりの任意のモノとの関わりに重点を置き、モノに込める情報とその操作、さらに、置く場との関係を規定するとともに、具体的な事例をプロトタイプシステムとして実現した結果を示す。

2. MonoGoto

MonoGoto は、モノを置いて何らかのコトを起こすシステムである。具体的には、あらかじめ机の上にモノとの関係を定義した場をつくり、その場でモノを登録、操作することで、操作の解釈に合ったコトが起こる。MonoGoto 上での主なモノと場の定義として、今回は、あたかもモノに音声を吹き込んだごとく、その音声を再生する MonoGotoRECO (以下 RECO)、置かれたモノの位置や角度を記録し、同じ条件の場合のみ反応する MonoGotoLOCK (以下 LOCK) を実現した。

^{†1} 龍谷大学大学院理工学研究科情報メディア学専攻
Ryukoku University Graduate School of Science and Technology

^{†2} 龍谷大学理工学部情報メディア学科
Ryukoku University Faculty of Science and Technology

2.1 モノの属性と登録

モノと場の定義においては、位置や角度、色や大きさといったモノを判別する情報を取得する必要がある。そのため、必要な情報を操作面上方のカメラで撮影した画面から取得している。

MonoGoto の実現には、まず場に置かれた複数のモノをそれぞれ判別し、トラッキングできることが必要である。このためには、専用の特殊なモノをあらかじめ用意する方法も考えられるが、我々は日常の任意のモノを扱えるようにしたいと考え、利用前にカメラで撮影した画像からモノの形や色を取得・登録し、その情報を用いてモノを判別することを考えた。また、モノに込める情報として、音声も扱えるようにした。

2.2 情報操作体系

MonoGoto の処理は、モノと場を用いた情報操作の体系に基づく。具体的には、モノ単体、モノと場の関係、複数のモノと場の3つの関係から成る(図1)。モノ単体では、モノだけで検出できる要素(向き)を用いて情報を操る。モノと場では、モノだけでわかる要素に加えて、場を用意することでわかる位置、角度、場の定義などの要素を加えて情報を操る。複数のモノを扱えば、それらに加えてモノ同士の相対位置、角度、距離なども扱えるようになる。また、今回はモノに込める情報として音声を用いたが、別のタイプの情報をモノに込めて扱うことも考えられる。こうしたモノと場の定義や関係を見ることによって、それらに様々な役割を与える情報操作体系を考えた。

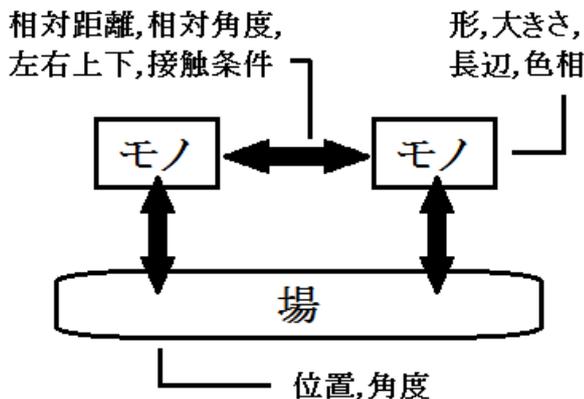


図1 MonoGoto で扱う情報

3. MonoGoto のプロトタイプシステム

3.1 システム構成

本システムは、机の真上にカメラを設置し、そのカメラ画面内に机上の中心とモノが映るようにした構成(図2)である。カメラの角度によってモノの大きさに大きな変化が無いように、操作面真上から撮影する。カメラは背後のPCに接続する。今回検討した構成は、机の大きさは横

140.0[cm]×縦70.0[cm]、高さ70.0[cm]であり、カメラの高さは91.0[cm]、撮影範囲は横92.0[cm]×縦70.0[cm]である。これは机の短辺とカメラ画像の短辺が一致し、画面から机がはみ出ない設定としたものである。

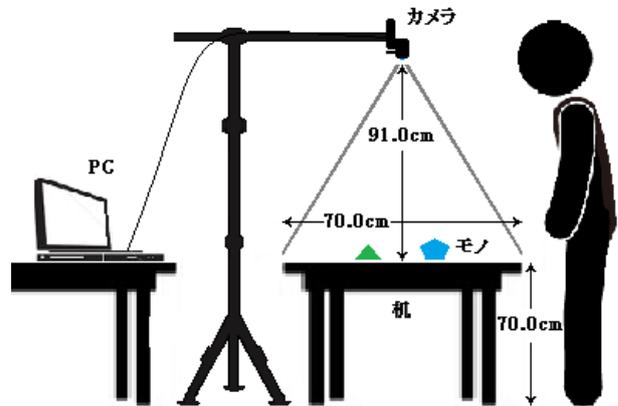


図2 システム構成

3.2 操作と処理の流れ

まず、モノの検出は、操作する場が固定として扱えるため、背景差分を用いて行うこととした。キャプチャボタンを押す事で背景画像を登録でき、この画像を常にカメラで撮影しているリアルタイム画像と比較することで、差分が発生した箇所のみを抽出した差分画像を作成する。

モノと場の定義は必要に応じて切り替えられるようにし、モノを登録する際には、利用者は机上の場に存在する登録ゾーンにモノを置くことで、置いたモノを登録することができる。また、このとき同時にモノへの音声録音を行うこともできる。登録ゾーンからモノを取り除くことによって、録音が終了すると同時にモノの登録が完了する。

操作・利用時には、登録したモノを置いて動かすだけでよい。定義によって様々な操作に機能をつけることができるが、あまり複雑にすると返って使いにくくなるため、できるだけシンプルにする方がよい。たとえば、RECOではモノの角度によって音量を、モノの位置で再生するタイミングを変えている。LOCKではモノの位置と角度を記録することができ、記録した条件と同じ条件の場合にのみあらかじめ定義された処理が実行される。

3.3 モノ属性抽出処理

モノを判別するためには、登録するモノの属性を抽出しなければならない。できるだけ共通する情報として、モノの面積、色相、長辺を利用することとした。本システムではOpenCVを用いて、背景差分画像からモノの輪郭線(図3)を抽出する。この輪郭線はいくつかの点から構成されているが、その点を内包するような楕円形を描写する。次にこの楕円形を内包する四角形を抽出する。この四角形を構成する4点の座標の内の1点を用いてモノの角度を取得す

る。また、この四角形からモノの面積と長辺も取得することができる。

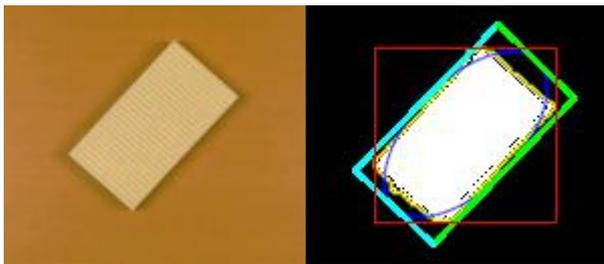


図 3 モノの輪郭線

このような手段を用いることで、カメラに映るモノならば、ほとんどのモノの面積、色相、長辺を検出することができる。今回モノのテクスチャーは考慮していないため、角度は対象性を考え、 180° までを検出範囲とするが、正方形では 90° までしか意味を持たず、円では全く意味を持たない。

3.4 モノ登録処理

場の中の登録ゾーン内の差分ピクセルが一定数以上になった場合、モノ登録処理が行われる。モノ登録処理はまず、登録ゾーン内の全差分ピクセルの色相を検出し、登録ゾーン内で最も多い色相と2番目に多い色相を決定する。次に、モノの面積と長辺を記録する。最後に、モノが登録ゾーンに存在する間だけ音声録音処理を行う。音声は登録処理が始まると同時に録音し始め、登録ゾーン内の差分ピクセル数が一定数以下になった場合、録音処理を終了する。録音の開始時と終了時にそれぞれピープ音を鳴らすことで、録音の区切りを利用者にわかりやすくしている。

3.5 モノ検出処理

モノ検出処理ではまず、登録された面積と長辺が許容閾値以内で一致するような四角形を探索する。一致する四角形を見つけた場合、その四角形内の全ピクセルの色相を調べる。登録した数値と許容閾値以内で一致する色相が一定数以上存在している場合、その四角形を登録したモノと判定し、その中心座標と角度を新たな情報として登録する。

輪郭抽出はリアルタイム処理中に生ずる細かな変動に左右されないため、閾値を設けて、あくまで似たようなモノを探索するようにしている。そのため、面積、色相、長辺全てが似たモノなら違うモノでも同じモノと見なしてしまうが、意図的にそうしたい場合以外では、あまり似たモノは扱わなければならない。

3.6 RECO/LOCK

RECO は、モノの角度や複数のモノの位置などで音声の再生方法を変化させるモノと場の定義例である。RECO の場では左から右に仮想の再生ラインがスキャンしており、

このラインと登録されたモノが重なることで、そのモノに登録された音が再生される。モノの角度によって再生する際の音量を変化させることもでき、複数のモノを置いた場合では、置く位置によって音声を順番に、または同時に再生することなどが可能となる。

LOCK は、モノの位置と角度を登録することで、モノ自体を処理のトリガー（鍵）として扱うモノと場の定義例である。モノ単体の場合だけでなく、複数のモノでも可能であり、登録した全てのモノの位置と角度を一致させることで処理が行われる。また、この方法以外に、複数のモノを利用した場合の相対距離と相対角度を用いる方法がある。この場合では、絶対位置や角度は関係なく、複数のモノの相対関係のみを見ることとなる。

4. 実験

4.1 閾値設定実験

4.1.1 閾値の必要性

MonoGoto では、モノ登録処理やモノ検出処理を行う際などに、多くの閾値を設定して検出精度にゆとりを持たせている。しかし、閾値の値によっては検出が厳しくなる事や、甘くなりすぎることが考えられる。そこで、適切な閾値を求める実験を行った。

4.1.2 実験方法

本実験はシステム構成で説明した構成と同じもので行う。登録ゾーン内に様々なモノを置いてピクセル数を調べ、その数値から本実験条件で最低限検出に必要なピクセル数を設定する。その後、そのモノ自体を登録し、場の様々な位置に様々な角度で動かしてみる。この時、モノ検出処理の面積と色相の閾値を変更し、本実験条件のどの場所・角度でも検出することのできる最小の閾値を探す。今回は後述する図4に含まれるペン・カード・時計・プラスチック・ぬいぐるみの5つのモノで実験を行った。検出する際の閾値は、面積については100/200/300/400/500/600[ピクセル]の6段階で行った。0~359の数値で表される色相は、1/3/5/7の4段階の数値で行った。

4.1.3 実験結果

カメラ直下でモノを回転させた場合のそれぞれの検出ピクセル数は、ペン:約850~1000ピクセル、カード:約2500~2700ピクセル、時計:約1150~1400ピクセル、プラスチック:約3550~3700ピクセル、ぬいぐるみ:7700~8100ピクセルとなった。どのような大きさのモノでも、検出されたピクセル数の幅は400ピクセル以内に収まる程度である。

面積の閾値は、100/200/300/400では移動させると検出不可、500では角度によっては検出できない、600ではほぼ完全に検出可能だが、四隅は若干不安定という結果となった。

色相の閾値は、数値 1 では移動させると検出不可、数値 3/5/7 では検出は可能だが、光を反射するようなモノは厳しく、数値が大きいくほど検出に余裕ができる結果となった。

これらの結果から、検出に必要なピクセル数は、本実験条件で最低限ペンを検出できる数値と考えると 600 とした。面積と色相の閾値は検出精度を考慮し、面積の閾値は 600、色相の閾値は 5 とした。

4.2 検出率評価実験

4.2.1 モノの検出率

MonoGoto はモノを利用するシステムのため、モノを可能な限り正確に検出しなければならない。そこで、閾値設定実験で設定した閾値を用いて、モノの検出率を測定する実験を行った。

4.2.2 実験方法

形状や形の異なる 15 個のモノを用意し、その中からひとつだけモノを登録する。その後、他の 14 個の登録していないモノを場に置いていき、それらを検出するかどうかを調べる。これを登録するモノ自体を含めて 15×15 の 225 通り調べ、登録あるモノ自身が検出できない場合と、検出しないはずのモノをどれだけ誤検出してしまうかを調べた。実験に使用したモノを以下の図 4 に示す。



図 4 実験に使用したモノ

4.2.3 実験結果

実験の結果、⑬を登録した場合に③を誤検出してしまうことがあったが、それ以外の 224 通りに関して誤検出は無く、登録したモノだけを検出できた。適合率は $254/255 = 0.996$ となった。

5. 考察

5.1 閾値設定実験

カメラ直下ではモノを回転させたとしても、検出される

ピクセル数は 400 ピクセル以内に収まる。一方で、場全体としては閾値 600[ピクセル]必要である。その理由は、カメラ画面におけるモノの映り方の違いにあると考えられる。カメラ直下から離れ、場の隅の方に行くほど、映り方に若干の違いが生まれ、その映り方の違いによって検出されるピクセル数が変化していると考えられる。

5.2 検出率評価実験

⑬を誤検出した原因として、背景の色相を取ってしまったことが考えられる。⑬は高さがあるビンであり、実験環境では左右天井に照明があったことも合わさって、ビンの左右に影ができてしまっている。その影の部分の色相、すなわち背景の色相をモノの色相として登録してしまったため、このような誤検出が起きたと考えられる。また、⑬と③の長辺と面積が似ていたのも原因だろう。対処法として、背景を黒一色にして影を映らないようにすることが考えられるが、それでは黒一色のモノを登録しづらくなる。実験をする場に応じて、モノか背景か、どちらかを工夫する必要があるだろう。

6. おわりに

本報告では、モノに様々な情報や機能を仮想的に持たせ、操作することを可能にするシステム MonoGoto を提案し、プロトタイプシステムとして実装した。

今回使用した様々なモノについては、ほぼ間違いなく検出でき、それらの位置や角度も十分に検出可能である。モノを利用することは、日頃人が自然に行っているモノを介したコミュニケーションに PC の処理を加えて、より豊富なインタラクションが行えるようになるのではないかと考える。今後はモノの検出率を上げるだけでなく、場の定義も増やすことで、利用できる状況をふやしていきたい。

参考文献

- 1) Hiroshi Ishii, Brygg Ullmer, “Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms,” in Proceeding of CHI '97, pp.234-241(1997).
- 2) 福地 健太郎, 高田 悠太, 中林 隆介, “Ficon: 立体映像表示が可能なタンジブルデバイスの提案,” インタラクション 2014 論文集 pp.100-107.
- 3) 安藤 正宏, 細井 俊輝, 中島 康祐, 高嶋 和毅, 伊藤 雄一, 足立 智昭, 尾上 孝雄, 北村 喜文, “StackBlock: 積み重ね形状構築を可能とするブロック型デバイス,” インタラクション 2014 論文集 pp.135-142.
- 4) 矢敷 貴之, 秋田 純一, “電力重畳通信機能を持つブロック型デバイスとその実装,” インタラクション 2014 論文集 pp.412-415.
- 5) 藪内 智浩, 船富 卓哉, 飯山 将晃, 角所 考, 美濃 導彦, “バーチャルスタジオにおける現実物体を介した仮想物体操作,” マルチメディア・仮想環境基礎(MVE)研究会, Vol.109 No.148, PP.1-6 (2009.7).