

Real から Virtual への移動システムの実現の試み

山之上 卓^{†1}, 筒井保博^{†2}, 筒井隆夫^{†3}

概要 : Real な「物」を, その内部構造も含めて Virtual な世界に移動 (正確にはコピー) するシステムの実現を試み, 一部成功したことについて述べる. 対象となる Real な「物」は, 規格化された知的部品を組み立ててできた「物」に限定する. この「物」を構成する部品は, お互いに通信が可能であり, また, お互いの相対的な位置や向き関係も知ることができるようになっている. 知的部品がお互いに情報交換することにより, この「物」自体が自分で自分の構造の情報を知ることができるようになっている. この情報を外部に吸い上げ, Virtual 空間上で Real な「物」と同じ物を構築することにより, この移動が実現する. ステレオカメラやレーザースキャナなどを利用した入力装置は表面上の形や色の情報しか入力できないのに対して, ここで試みた方法は, 内部の構造の情報も入手できる. CT スキャナや MRI は, 内部の構造の情報も入手できるが内部の機能の情報是一部しか入手できないのに対して, ここで試みた方法は, 内部の機能の情報も入手できる. また, CT スキャナや MRI は対象物より大きなものになってしまうが, ここで試みた方法は対象物そのものが, 自分の情報を取り出す. なお, Real な世界にあった「物」を破壊することにより, このコピーは実質的な「移動」になる.

1. はじめに

Real な「物」を, その内部構造も含めて Virtual な世界に移動 (正確にはコピー) するシステム実現の試みが一部成功したことについて述べる. 対象となる Real な「物」は, 規格化された知的部品を組み立ててできた「物」に限定する. この「物」を構成する部品は, お互いに通信が可能であり, また, お互いの相対的な位置や向き関係も知ることができるようになっている. 知的部品がお互いに情報交換することにより, この「物」自体が自分で自分の構造の情報を知ることができるようになっている. この情報を外部に吸い上げ, Virtual 空間上で Real な「物」と同じ物を構築することにより, この移動が実現する. ステレオカメラやレーザースキャナなどを利用した入力装置は表面上の形や色の情報しか入力できないのに対して, ここで試みた方法は, 内部の構造の情報も入手できる. CT スキャナや MRI は, 内部の構造の情報も入手できるが内部の機能の情報是一部しか入手できない

[3]. これに対して, ここで試みた方法は, 内部の機能の情報も入手できる. また, CT スキャナや MRI は対象物より大きなものになってしまうが, ここで試みた方法は対象物そのものが, 自分の情報を取り出す. なお, Real な世界にあった「物」を破壊することにより, このコピーは実質的な「移動」になる.

2. システム概要

図 1 にシステムの概要を示す. 相互に接続された面とその向きを認識可能な知的部品「サイコロ」を組み合わせて, 「物」を作る. それぞれの知的部品は, 隣接した知的部品と赤外線を使って情報交換を行い, どの面がどの向きでどのサイコロと接続されるか認識する. この情報をサイコロの中の 1 つが持っている通信機能を持った素子(M5Stack)を通じて外部にあるパソコンに取り出し, そのパソコンで, Python を使って, 「物」と同じものを, パソコン内部に再現する. 図 2 にこのシステムの実現例を示す.

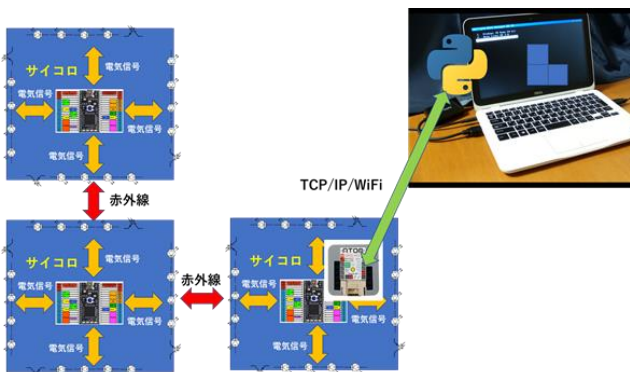


図 1. システム概要

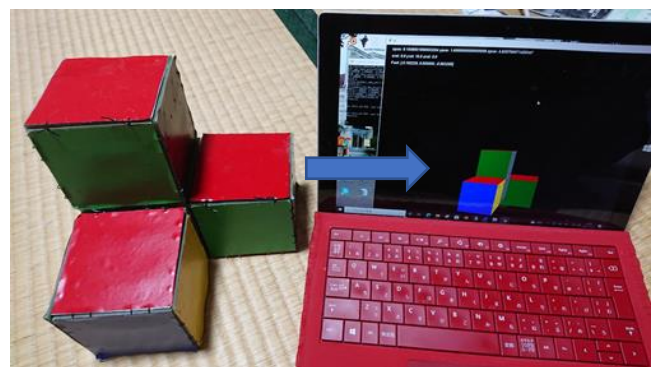


図 2. システムの実現例

†1 福山大学

†2 福岡産業保健総合支援センター

†3 産業医科大学

3. 「物」を構成する規格化された部品

この試みでは知的部品として大きさが同じ正6面体のブロック(サイコロ)を利用した。サイコロの各面は図3のような部品間の物理的な結合機構と結合向き認識のための素子を。対称軸に対して相補的に配置している。この素子は。サイコロ間通信にも利用する。このような配置を行うことにより。面が重なる向きであればどの向きでも。サイコロを結合することができ。また。どの向きで面が結合されたかを知ることができる[7][9]。

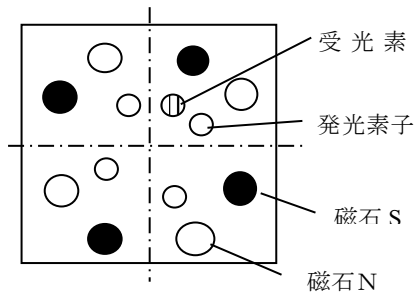


図3 サイコロの面

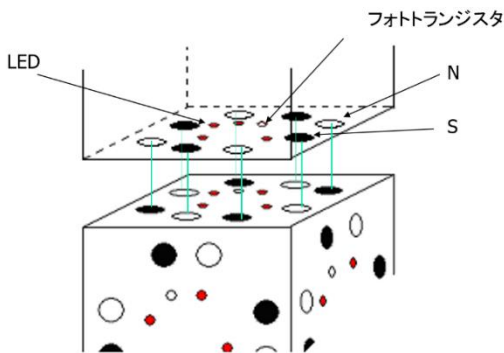


図4 サイコロ同士の物理的な結合

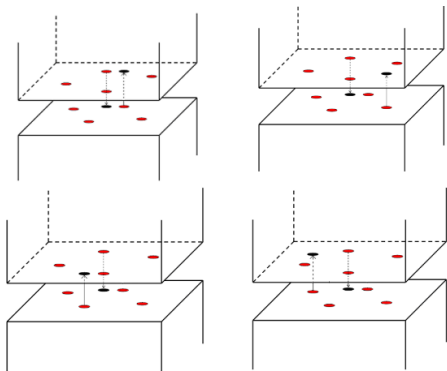


図5 結合した向きの認識とサイコロ間の通信機構

3.1 サイコロ同士の結合機構

図4にサイコロ同士の物理的な結合機構を示す。この図のように磁石を配置することにより、任意の2つのサイコロを、それぞれの面の一つが重なるように近づけると、磁石によって引き付け合い、結合する。このような磁石の配置により、面と面が重なる向きであれば、どの向きでも結合する。

3.2 結合の向きの認識と部品間の通信機構

図5にサイコロ同士が結合したときの、結合の向きの認識と部品間の通信機構を示す。2つのサイコロが物理的に結合したとき、それぞれのフォトトランジスタは、向きによって異なるLEDと向き合う。片方のフォトトランジスタが、相手方の、どのLEDと向かい合わせになったか？を、相手方が、結合面の4つのLEDをひとつずつ点灯し、相手方のどのLEDが点灯したとき、自分のフォトトランジスタがそれを検知したか？を知ることによって結合の向きを知ることができる。

また、自分のフォトトランジスタと相手方のフォトトランジスタはそれぞれLEDと向き合っており、お互いに全二重の通信を行うことができる。

4. 「物」の構造の文法と意味付け

4.1 「物」の文法

「物」の構造を自分自身で把握するために、「物」を、ある一定の(文法)規則の集合から生成された、階層構造を持った文であると解釈する[8][12]。

ここで、サイコロを生成規則として解釈する。サイコロ I の面 m を 非終端記号

$F(I,m)$ または $G(I,m)$ ($0 \leq I < \text{サイコロの個数}, 0 \leq m \leq 5$) で表すことにする。 $F(I,m)$ は面が結合したとき、構文木上で根の側にある面を表し。 $G(I,m)$ は葉の側にある面を表す。開始記号 S はプローブを表す。

$$G(I,m_0) \rightarrow F(I,m_1)F(I,m_2)F(I,m_3)F(I,m_4)F(I,m_5)$$

(生成規則1)

はサイコロ I がこの構造物の一部であることを表し、解析木上に高々1度出現することにする。 $F(I,m_2)$ はサイコロ I の $G(I,m_0)$ ではない面を表す。

$$G(I,m) \rightarrow \varepsilon \quad (\text{生成規則2})$$

は、ある面が、サイコロ I の面 m と結合しており、かつ、構文解析上の他の場所でサイコロ I に関する生成規則1が使用されていることを表す。

$$F(I,m) \rightarrow G(k,n) \quad (\text{生成規則3})$$

は、サイコロ I の面 m がサイコロ k の面 n と結合していることを表す。

$F(l,m) \rightarrow \varepsilon$ (生成規則 4)

はサイコロ I の面 m が他のサイコロと結合していないことを表す。

$S \rightarrow G(l,m)$ (生成規則 5)

はプローブ S が、サイコロ I の面 m と結合していることを表す。

4.2 「物」の意味

「物」の物理的な結合情報と、それぞれの部品の内部状態を外部に取り出すために、以上の生成規則に対して、属性文法[1]を使って意味付けを行う。生成規則 1 において、 $G(l,m)$ の合成属性値は、リスト

(“サイコロ情報”, l ,
 $F(l,m_1)$ の合成属性値, ..., $F(l,m_n)$ の合成属性値,
 . サイコロ I の内部状態)

とする。またサイコロ I の内部状態は、この合成属性値そのものを含まないこととする。生成規則 3 において、 $F(l,m)$ の合成属性値は、リスト

(“結合情報”, l, m, k, n ,
 面 $F(l,m)$ と面 $G(k,n)$ の結合の向き,
 $G(k,n)$ の合成属性値)

とする。生成規則 5 において S の合成属性値は $G(l,m)$ の合成属性値とする。これ以外の生成規則の、左辺の合成属性値は nil とする。

それぞれの部品がセンサを持っていた場合、そのセンサが計測した値を内部状態として外部に取り出すことにより、それぞれの部品のセンサの計測値を知ることができる。

相続属性を利用することにより、外部から「物」を構成する部品のそれぞれに対して情報を送ることができる。このことにより、アクチュエータを持っている部品があれば、そのアクチュエータを外部から操作できる。

5. 「物」の構文解析

「物」の 1 箇所にはプローブ S を取り付け、 S から生成規則 1 ~ 5 を下向きに適用していくことによって、構文木が構成され、これに意味規則を適用していくことによって、この構造物の構造(どこがどのように結合しているか?)と、この構造物を構成するすべての、サイコロの内部状態の情報が、プローブ S の合成属性値として得られる。各生成規則の適用や意味解析は、対応するサイコロ自身が行う。この情報を使って、コンピュータ上で構造物の表示を行ったり各構成部品の状態をモニターしたりすることができ

る。図 3 に本手法を使って得られた構造物の構文木の例を示す。

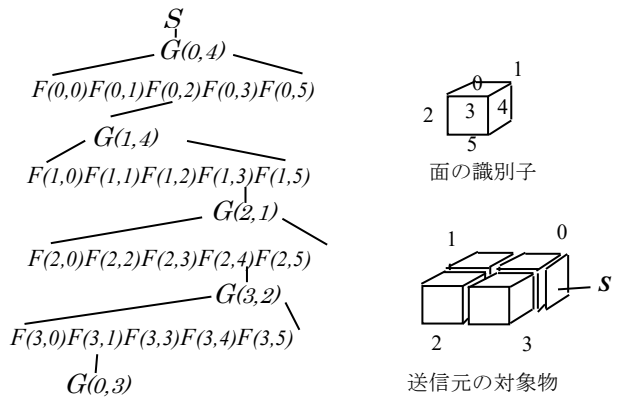


図 6 構文木の例

6. サイコロのハードウェア

サイコロのそれぞれは図 7 で示すような回路を内部に持っている。

- ① この中で、M5Stack Atom は 1 つのサイコロにだけ入っている。
- ② このサイコロの中の Mbed LPC1768(以後 1768)と M5Stack ATOM (以後 ATOM)の間が UART で接続されている。
- ③ パソコンと ATOM の間は WiFi を使って、TCP/IP でつながる。
- ④ ATOM で WiFi のサーバを動かすと同時に、TCP/IP ストリーム(リモートの PC 側)と UART(ローカルの 1768)の中継サーバを起動する。
これにより、パソコンと、この 1768 の間で、双方向に、コマンドとメッセージが流れる。
- ⑤ パソコンから Atom と繋がった 1768 に対してコマンドを送ると、その 1768 からパソコンにコマンドの実行結果がメッセージとして返る。このコマンドが構文解析を行うコマンドだった場合に、構文解析が行われ、構文木の情報が実行結果のメッセージとして、パソコンに送信される。

また、LED には赤外線 LED を利用している。赤外線はカラーシートを透過してフォトトランジスタに届くので、サイコロの面をカラーシートで被うことにより、接続・通信機構が見えないようにしている。但し、黒のカラーシートは赤外線を通さないので穴をあけている。

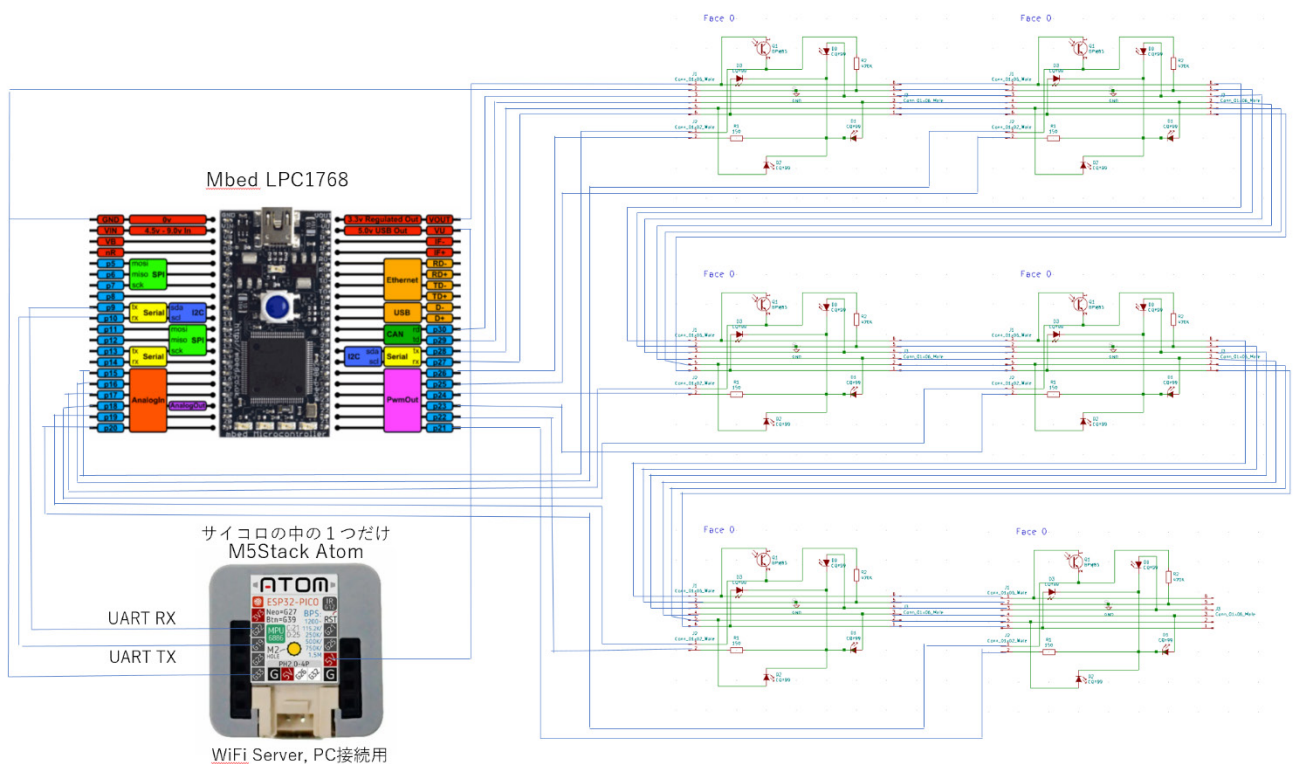


図 7. サイコロの回路

7. サイコロのソフトウェア

サイコロのソフトウェアは Mbed のクラウド開発環境とプログラミング言語 C++を使って開発した。このソフトウェアは主に、CommandBuffer のコマンドインタープリタと、InBuffer のコマンドインタープリタによって構成されている。

7.1 CommandBuffer のコマンドインタープリタ

デバッグのための USB-シリアルインターフェースと、Atom の UART を通じてパソコンからの入出力を行うインターフェースから入力されるコマンド、および、InBuffer 経由で入力されるコマンドを解釈実行するコマンドインタープリタである。このコマンドインタープリタが受け付ける主なコマンドは以下のとおりである。

(a) *cq face <n> dir <d>*.

そのサイコロの n 番目の面の<d>番目の LED から、その面に接続されているサイコロがあるか否かを調べる。もしその面に接続されている別のサイコロがあった場合、そのサイコロは、次の *ack1* コマンドを、その面の発光ダイオードを通じて、cq コマンドを送ったサイコロに返す。

(b) *ack1 face <n> dir <d1> uDir <d2>*.

そのサイコロがどれかの面から cq コマンドを受け取った時に、このコマンドを、その面の、

相手のフォトトランジスタと向かい合っている発光ダイオードを使って送信する。<n>は cq を受け取った面の番号、<d1>は接続の向き、<d2>は相手方の接続面の接続の向きを表す。<d1>と<d2>はいつも同じ値になる。

このコマンド *ack1* が受け取られたサイコロでは、コマンド *send up "str (this <myname> f<n1>d<d1> next <downname> f<n2>d<d2>)"* を、構文解析木上で根の側に位置するサイコロに送る。もしこのコマンドを受け取ったサイコロが構文解析木上で根にあたる場合、Atom を通じて、このコマンドを PC に送る。

このコマンドは、サイコロ <myname> と <downname> が、<myname> 側では <n1> の面で、<downname> 側では <n2> の面で接続されており、その接続の向きは、<d1> と <d2> であることを示している。<d1> と <d2> は同じ値になる。ここで <myname> は自分のサイコロの識別子であり、この名前は構文木上で根ノードの名前を "s*" とし、それ以外のサイコロの名前は、そのサイコロの構文木上で一つ根の側のノードの名前に、上のノードとそのノードが接続している、上のノードの面の番号を加えたものとしている。<downname> はこのサイコロの、構文木上で葉の側に接続されたサイコロの名前を表す。..

(c) *send up "<str>"*.

文字列<str>を構文木上で根の側に位置するサイコロに送る。

(d) *srs*.

このコマンドを実行するサイコロを構文解析木の根ノードとして、「物」の構文解析を行う。このコマンドは、それが接続されたサイコロで再帰的に *srs* を実行する。*srs* コマンドを受け取ったサイコロでは、もし、そのサイコロがはじめて *srs* コマンドを受け取った場合は、構文解析木上で根の側のサイコロと接続されている面(*srs* コマンドを受け取った面)を除いて、すべての面で *cq* コマンドを実行する。その結果、*ack1* が返ってきた面を通じて、隣接するサイコロに *srs* コマンドを送る。もし、このサイコロが既に *srs* コマンドを受け取ったことがある場合、*srs* コマンドの送信を行わない。サイコロの構文解析結果は根ノードが受け取る (b) の *send up* メッセージを並べることによって得ることができる。

(e) *srr*.

一度構成された構文解析木を、それを再帰的にたどって、初期化する。

7.2 InBuffer のコマンドインタープリタ

サイコロのそれぞれの面のフォトトランジスタから入力される bit 列をデコードするコマンドインタープリタである。この bit 列を文字にし、その文字がコマンドの一部と思われるものと、そうでないものに分け、コマンドコマンドの一部と思われる場合は、その文字をつないでコマンドを組み立て、そのコマンドを、7.1 のコマンドインタープリタにコマンドバッファを通じて送信する。

bit 列を入力するために、タイマー割込みを使っている。

8. パソコンのソフトウェア

サイコロを組み立てて「物」を作った後、パソコン側の Python のプログラムで“srs.”コマンドを、その「物」に WiFi を通じて送信すると、構文解析が行われ、サイコロの間の結合の情報が、コマンド *send up "str (this <myname> f<n1>d<d1> next <downname> f<n2>d<d2>)"*

の羅列として得られる。

この情報を元に、仮想空間上でサイコロを組み立てることにより、Real 空間で組み立てられた「物」が、仮想空間で再現される。図 8 の「物」が組み立てられた後、パソコンから *srs* コマンドを送信すると、以下のコマンドの列が「物」からパソコンに送られる。

send up "str (this s f2d0 next s*3 f3d0)".*

*send up "str (this s*3 f0d2 next s*35 f5d2)".*



図 8. 8 章のコマンド列が生成されたときの Real な「物」と Virtual 空間で再生された「物」

図 8 のサイコロは、面 0 に赤、面 1 に青、面 2 に黄、面 3 に緑、面 4 に黒、面 5 に白の色が着いている。*s** は図 8 の右側の Real な「物」において、右下のサイコロを表し、*s*3* は左下のサイコロを表し、*s*35* は左上のサイコロを表す。

(*this s* f2d0 next s*3 f3d0*) はサイコロ *s** の黄色の面(f2)とサイコロ *s*3* の緑(f3)が、向き 0(d0)で接続されていることを表す。

(*this s*3 f0d2 next s*35 f5d2*) はサイコロ *s*3* の赤の面(f0)と、サイコロ *s*35* の白の面(f5)が向き 2(d2)で接続されていることを表す。

この情報を使って図 8 の右側のように Virtual 空間で Real な「物」と同じものを組み立てることができる。

9. 応用例

ここで示した技法は、立体構造のコンピュータ入力のほか、各サイコロの各面にストレインゲージなどのセンサを取り付けることによって、構造物の応力解析に応用することができる。また構文解析の結果得られた情報に基づいて、遠隔地点で同じものを自動的に組み立てる受信システムなどを加えて、一種の teleportation システムを構成することができる [11][12][13][14]。また、この受信システムそのものが、このサイコロにセンサやアクチュエータなどを加えたもので構成されていれば、一種の self-reproduction(self-replication)システムを構成することができる。

10. 関連研究

10.1 Active Cube

伊藤らはリアルタイム 3 次元形状モデリングとインタラクションのための双方向ユーザーインターフェース機能を持つ Active Cube[10]を提案した。本システムは Active Cube は本システムと類似している

が, Active Cube は面の向きの認識は行わない代わりに, 面が結合する向きに制限をかけている. 本システムは面の結合する向きに制限はなく, 結合の向きを認識する機構が備わっている.

10.2 iCray

橋本らは, 微小な共振アンテナを粘土に大量にうめこむことで, 粘土で作られた造形物をコンピュータに取り込むシステム iCray に関する基礎的な研究を行っている[15]. また, iCray はこする, 叩く, つまむなどの操作を検知することに成功している. しかしながら iCray はアクチュエータを埋め込むには向いていない.

10.3 CEBOT

福田らは, ロボットを複数の小さなロボット (セル) で構成した CEBOT の研究を行っている[4]. 我々のシステムに CEBOT の研究成果を組み込むことで, 子供が設計図なしで組み立てたロボットを Virtual 空間上で再現したり, そのロボットを遠隔地に転送して遠隔地でうごかしたりすることが可能になると思われる.

10.4 Triangles and Narratives

Triangles and Narratives [5]は Real な世界で三角形のデバイス同士がその端の磁石で結合できるようになっており, そのデバイスの結合状況が Virtual 空間で再現されるものである. 一部立体的な物体も Real と Virtual な空間で実現できるが, 構造上, 沢山のデバイスを使った少し大きな立体構造を実現することは困難である.

10.5 相補的な構造

中野は, 相補的な構造による部品の結合や, 進化するロボットなどについて述べている[6]. 本システムは相補的な構造による部品の結合を行っている.

コンパイラ[2]の自己生成(ブートストラッピング)は, コンパイラ作成者は日常的に行っており, 本論文の応用例で述べた Self-reproduction はこれと同様のことを行う. これによって, 物を進化させることが可能であると考えている.

10.6 ぶろっくめーかー

ぶろっくめーかー[16]は, 遊び感覚で使える 3D モデル作成ソフトである. Virtual 空間中でマスをクリックしていくだけで簡単にカラフルな 3D モデルを作成できる. ぶろっくめーかーは Virtual な世界の中だけで 3D モデルを組み立てるのに対して, 本システムは Real の世界で組み立てたものを Virtual な世界に移動させている.

11. おわりに

Real な世界で組み立てられた「物」を Virtual な世界に移動することができたことについて述べた. 現在 PC の Virtual な世界は独自の 3D グラフィクスで作成しており, データの互換性がない. 現在, この Virtual な世界を Blender に置き換える作業を行っている. これができれば, 本移動システムを, Blender のモデリングツールの一つとして利用

できる. また, Virtual な世界の「物」を Real な世界の「物」に移動する機能を付け加えることにより, ある場所にある Real な物を別の場所に「移動」することが可能になる. これは一種の Teleportation System である.

謝辞 本研究は北九州市の産学官交流サロン. What How For 北九州で行われた議論が発端となって始まったものである. What How For 北九州の関係者の皆様に感謝します.

参考文献

- [1] D.E.Knuth, "Semantics of context-free languages", *Mathematical Systems Theory*, 2, 2, pp.127-146,1968.
- [2] 中田育夫, "コンパイラ", 産業図書,1981.
- [3] The 1st 3D fax, done in 1991. http://www.tacc.utexas.edu/~reyes/tacc_personal/self/3D_fax.html
- [4] T.Fukuda and T.Ueyama, "CELLULAR ROBOTICS AND MICRO ROBOTICS SYSTEMS", World Scientific,1994.
- [5] M. G. Gorbet 他, "Triangles: tangible interface for manipulation and exploration of digital information topography", *Proceedings of the SIGCHI conference on Human factors in computing systems*, Los Angeles, CA., US., pp.49-56.,Jan. 1998.
- [6] 中野馨, "脳を作る", 共立出版,1995.
- [7] 筒井他, "自己構造解析を行う機構", 情報処理学会第 61 回全国大会論文集, デ-10,2000.
- [8] 山之上他, "物理的な構造の構文解析とそれを実現する部品", 情報処理学会第 61 回全国大会論文集, 5Q-03,2000.
- [9] 山之上卓, 筒井保博, 筒井隆夫, "セル及びその結合状態認識方法", 特許:特開 2001-188029 (P2001-188029A)
- [10] 伊藤雄一, 北村喜文, 河合道広, 岸野文郎, "リアルタイム 3 次元形状モデリングとインタラクシヨンのための双方向ユーザインタフェース ActiveCube", *情報処理学会論文誌*, vol. 42, no. 6, pp. 1338-1347, 2001.
- [11] T. Yamanoue. Y. Tsutsui, T. Tsutsui, "Realizing a Practical Teleportation System Using the Intelligent Parts", *Proceedings of the 1st International Conference on Information Technology and Applications (ICITA-2002)*, 221-2, Bathurst, Australia, Nov. 25-29, 2002.
- [12] 山之上 卓, "テレポーテーションとコンパイラ", *情報処理学会夏のプログラミングシンポジウム 2004*, pp.99-102, 函館, 22-24 Aug. Japan, 2004.
- [13] 山之上 卓, 筒井 保博, 筒井 隆夫, "テスト計画の進捗状況 2006 -テレポーテーションは実現できるか?-", *情報処理学会夏のプログラミングシンポジウム 2006*, pp.9-15, 別府, 2-4 Sep. Japan, 2006.
- [14] 山之上 卓, 筒井 保博, 筒井 隆夫, "多数の知的部品で構成された機能物体をネットワークで遠隔地に転送するシステム", *SCI' 07 第 51 回システム制御情報学会研究発表講演会論文集*, 2W1-4, pp.99-100, 京都, 2007.5.
- [15] 橋本昌宜(研究代表者), "極小センサノードを用いたリアルタイム 3 次元モデリングインターフェース iClay", *科学研究費助成事業 研究成果報告書, 基盤研究(A), 研究期間 2011-2013 課題番号 23240015*, 平成 26 年 6 月 9 日現在, 2014.
- [16] ANDOR "ぶろっくめーかー", <https://www.andor.co.jp/products/blockmaker/index.html>