

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 双方向ユーザインタフェース ActiveCube

伊藤 雄一[†] 北村 喜文[†]
河合 道広[†] 岸野 文郎[†]

我々はリアルタイム 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インタフェースの機能を持つ ActiveCube を新たに提案する。双方向ユーザインタフェースである ActiveCube を使うことによりユーザはブロックを組み立てることで 3次元環境を構築し、インタラクションを行うことができる。コンピュータはブロックとリアルタイムに通信することにより、組み立てつつある形状をリアルタイムで認識できる。また ActiveCube の各ブロックは、入出力デバイスを持つので、ユーザの操作意図とコンピュータ内でのシミュレーション結果の間の因果関係を明確に表現できる。さらに実物体と仮想物体間のコンシステンシはつねに保たれる。

ActiveCube: A Bi-directional Interface for Real-time Constructing and Interacting with a 3D Environment

YUICHI ITOH,[†] YOSHIFUMI KITAMURA,[†] MICHIMIRO KAWAI[†]
and FUMIO KISHINO[†]

We propose the ActiveCube system, which allows a user to construct and interact with a 3D environment by using cubes with a bi-directional user interface. A computer recognizes the 3D structure of connected cubes in real time by utilizing the real-time communication network among cubes. Also, ActiveCube is equipped with both input and output devices, at where the user expects to be, and this makes the interface intuitive and helps to clarify the causal relationship between the input of the user's operational intention and the output of simulated results. Consistency is always maintained between the real object and its corresponding representation in the computer in terms of object shape and functionalities.

1. はじめに

近年、3次元物体の形状モデリングや3次元世界に対するインタラクティブな操作について、直接操作の考え方¹⁾に基づいた、直感的なグラフィカルユーザインタフェースが多数提供されるようになってきた。しかし、モデラの操作にはある程度の専門的な知識を必要とし、子供や初心者には扱いにくく、操作対象である3次元オブジェクトに実体がないので、これを操作する際に実際に操作している感覚になりにくいことや、ユーザの操作意図などの入力装置とその結果の出力装置(ディスプレイ)が分離しているために、両者の間の因果関係が不明確になりやすいなどという問題もあった。

そこで、物理的に実体を持つ物体を操作対象物に利

用することにより、操作の直感性を向上しようとする試みも、いくつかなされるようになってきた^{2)~6)}。3次元形状モデリングを目的としたものとして、LEGO型ブロックを組み立てた後に、コンピュータに接続して電源を入れると、その形状がコンピュータに入力されるものがある⁷⁾。しかし、組み上げたブロック形状の認識はリアルタイムではなく、またブロック自体が利用者の操作意図などを入力したり計算機内のシミュレーション結果を表示する出力デバイスを持たないため、そのブロックを介したインタラクションをユーザが行うことはできない。ほかにも、正三角形平板のオブジェクトを組み合わせて形状を作成することにより、コンピュータ上の特定のプログラムを起動しようとする装置⁸⁾や、ブロックを組み合わせてある種のプログラムを作成しようとするもの⁹⁾などがあるが、これは3次元物体の形状モデリングや3次元世界に対するリアルタイムインタラクションを目指したものではない。我々は、実物体としてのブロックを組み立てること

[†] 大阪大学大学院工学研究科

Graduate School of Engineering, Osaka University

で容易にリアルタイムで 3 次元形状をコンピュータ内にモデリングすることができ、しかもブロック自体に各種のセンサやアクチュエータなどを実装することによって、利用者の操作意図を入力し、計算機内でのシミュレーション結果を表示することによるリアルタイムインタラクションの機能を有する双方向ユーザインタフェースとして、ActiveCube を提案する。ActiveCube を用いることで、本来は難しい 3 次元形状モデリングの作業が直感的に誰にでも簡単に素早くできる。さらに、目の前に物理的に存在する物体に対して自然にインタラクションを行うことにより、3 次元世界での因果関係を正しく理解することができる。また、実世界のブロックの形状と機能は、仮想世界内の表現と、いつでも対応がとれている。このような特徴を生かして、直感的な工業デザイン・シミュレーションツール、アートや教育の分野で人の直感や感性を支援する 3 次元インタラクションツールや、障害を持つ人のためのトレーニングやリハビリテーションツールなどへの応用が考えられる。

本論文では、まずこれらリアルタイム 3 次元形状モデリングとリアルタイムインタラクションを実現するための要素技術であるハードウェア、リアルタイム 3 次元形状認識手法、ソフトウェア、様々な入出力機能の詳細について述べ、ActiveCube のリアルタイム性を検証した実験について述べる。さらに ActiveCube の使用例について述べる。

2. リアルタイム 3 次元形状モデリングとリアルタイムインタラクション

コンピュータを利用した 3 次元形状モデリング作業の入力インタフェースとして、ブロックなど物理的に実体のある物体を用いれば、3 次元世界での人の空間認識の助けともなり、ユーザにとって扱いやすく直感的になる。この考えに基づいて、実体のあるブロックを用いて 3 次元モデリングを行う研究は、建築設計^{10)~12)} や、工業デザイン¹³⁾ の分野において、古くより行われてきた。近年では、LEGO 型のブロックをユーザが自由に組み立てた後に、コンピュータに接続して電源を入れるとその形状がコンピュータに入力されるものがある⁷⁾。この研究においては、各ブロックには PIC と呼ばれるプロセッサが搭載されており、コンピュータと各ブロック間の通信を行うことによって、組み上げられたブロック形状を認識する。しかしコンピュータと各ブロックはシリアル接続するため、各ブロックが保持している接続情報データをホストに送信する際、パケットが伝送経路上のブロック

の各 PIC を通るため、通信のオーバーヘッドが大きくなり遅延が生じる。また、ホストに近い側のブロックから順次そのデータを送信し、送信が終了したブロックから Message-Forwarding-Mode となり、下流にあるブロックからのデータをホストに順次転送する。もしその間にパケットロスなどがあると、各ブロック間で再送作業が行われることとなり、さらなる通信遅延を招く原因となる。さらに、あるブロックからホストにデータを伝送する経路が複数存在する場合は、すべての経路にデータが伝送されるため、ネットワークリソースを大幅に消費してしまい、これもまた通信遅延を招く原因となる。このように、ブロックの接続方式がシリアル接続であるため、ブロックを組み立てている最中のリアルタイム形状認識を行うことは不可能である。そのため、覚えやすく使いやすいといわれている直接操作ユーザインタフェースの必須な条件である即時応答性などの条件を満たしているとはいえない¹⁾。

さらに、これらのブロックを用いたモデリングシステムは、その形状のモデリングに特化したシステムであり、ブロック自体にコンピュータ内のシミュレーション結果を表示する出力デバイスを持たない。たとえば形状認識をさせた後に、ある種の計算結果を表示しようとしても、ディスプレイモニタやスピーカなど、通常コンピュータに接続されている一般的な機器を利用するしかない。しかし、3 次元の実体を持つ物体が目前にある以上、この物体に関わる計算結果は、物体上のしかるべき場所に表示されるのが自然であると思われる。同様に、上記のシステムでは、ユーザの操作意図や環境情報を入力するためのデバイスをブロック自体が持たないため、キーボードやマウスといった一般的な機器を利用するしかなかった。しかし、実体を持つ物体が目前にある以上、この物体に関わる操作の入力も、物体上のしかるべき場所に対して入力されるのが自然であると思われる。つまり、従来のシステムはいずれも、形状のみをオフラインで入力するユーザインタフェースであり、そのブロックを介して直接インタラクションを行うことはできなかった。

組み立てつつあるブロックの形状をリアルタイムにコンピュータに入力できれば、即時応答性のある直接操作ユーザインタフェースを構築することができる。また、センサやアクチュエータなどの入出力デバイスを装備したブロックを、組み立てた形状の任意の箇所に自由に接続することができれば、ユーザの操作意図や環境情報を物体上のしかるべき箇所からコンピュータ内に入力し、逆にコンピュータ内のシミュレーション結果を物体上のしかるべき箇所からユーザに提示す

るといった、直感的なインタラクションが可能となる。

ActiveCube の各ブロックとホスト PC はパラレル接続するため、すべてのブロックはホスト PC と直接通信できる。そのため接続・分離情報はリアルタイムにホスト PC に送信され、ホスト PC はその形状をリアルタイムに認識できる。さらに各ブロックはセンサやアクチュエータなどの入出力デバイスを装備し、ユーザの操作意図や環境情報をセンサを用いて取得し、ホスト PC にそのデータを直接送信し、ホスト PC 内のシミュレート結果はアクチュエータなどを用いて提示できる。このように ActiveCube はその形状のモデリングに特化したユーザインタフェースではなく、その形状情報を用いた様々なアプリケーションを構築できる。

次章以降では、リアルタイム 3 次元形状モデリングと、ブロックを介した双方向リアルタイムインタラクションを実現する ActiveCube の詳細について述べる。

3. ActiveCube のハードウェア構成

ActiveCube を実現するために必要なハードウェアとリアルタイム 3 次元形状認識手法について述べ、現在の ActiveCube が持つ制約について述べる。

3.1 ハードウェア

リアルタイム 3 次元形状モデリングとリアルタイムインタラクションを実現するためには、接続・分離された各ブロックから、接続状況やブロックの種類などに関する情報をリアルタイムで取得し、これらを制御しなければならない。すなわち、接続された各ブロックをノードとしてネットワーク化し、それぞれをリアルタイムに分散制御することが必要である。先述の非リアルタイムのシステム⁷⁾ などでは、各ブロック内に PIC と呼ばれるチップを実装して制御している。しかし、これによってネットワークをリアルタイムで管理することは、内部のリソースのパフォーマンスから考えて困難である。

そこで ActiveCube では、個々のブロックをネットワーク化し、各々を管理・制御する機能を米 Echeron 社の LON (Local Operating Network) と呼ばれる知的分散制御を可能とする技術を利用し、拡張することにより実現する¹⁴⁾。ActiveCube の各ブロックを LON におけるノードとし、それを制御するために NeuronChip と呼ばれるチップ (東芝製 TMPN3120FE3M) を搭載する。このチップは内部に 8 bit の CPU を 3 個搭載している。内部の RAM は 2 kbyte, ROM は 16 kbyte, EEPROM は 2 kbyte である。さらに、16 bit のタイマカウンタを 2 チャンネル持っており、動作周波数

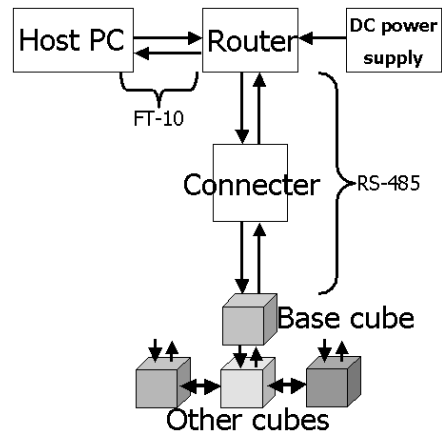


図 1 ActiveCube システムの構成図
Fig. 1 ActiveCube system configuration.

は 20 MHz である。またその形状は 32 ピンである。3 つのプロセッサのうち、1 つがネットワークの通信管理、もう 1 つがアプリケーションプログラムの実行と管理、残り 1 つがこれら 2 つの CPU との橋渡しの役割を果たしている。各ブロック制御用のプログラムは NeuronC でコーディングされ、コンパイル後、EEPROM に書き込むことで実行される。

各ブロックにはブロック ID を一意に割り振り、個々のブロックを識別可能とする。また、各面には接続面を識別するための面 ID を割り当てる。接続されたブロックはネットワークを構成し、RS-485 方式を用いて相互通信を行う。通信速度は 39 kbps である。ブロックとホスト PC との接続は、図 1 に示すようにベースブロックが担当し、ルータを介して RS-485 方式を FT-10 方式に変換して通信する。FT-10 方式の通信速度は 10 Mbps である。

このように、各ブロックの各面には通信線 2 本と電源線 2 本の計 4 本の通信線が必要となる。これらを実装するために、システムワット社製の RoboCube¹⁵⁾ のブロックのシェルを使用する。図 2 のように、ブロック各面の中心から同心円上に接触端子を 3 個配置し、中心側から通信 1, 通信 2, GND (-電源) を割り当てる。これは接続しているブロックの接続面が完全に接続されていない場合の通信線などの断絶を防ぎ、接続面内の回転に対しても対応するためである。また、Vcc (+電源) は各面の頂点近くに配置された接続ホックを用いることで供給する。さらに、接続面を認識するために Vcc 供給ホックのうち 1 つを接続認識用ホックとする。

3.2 リアルタイム 3 次元形状認識

組み上げつつあるブロック全体の形状をリアルタイ

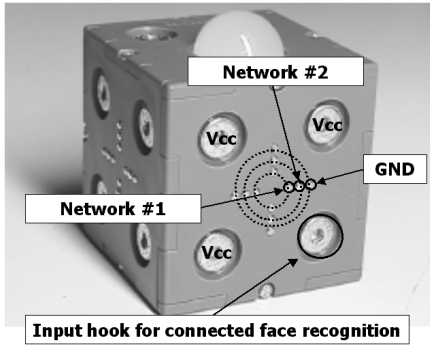


図 2 ActiveCube の外観
Fig.2 ActiveCube.

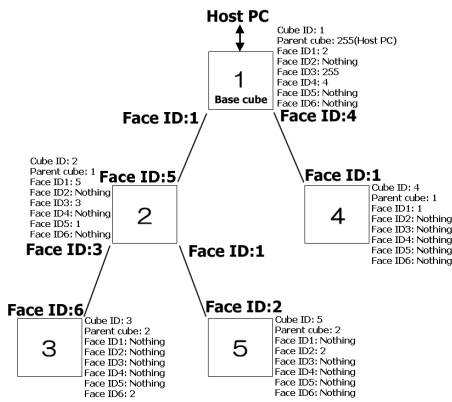


図 3 ブロック接続状況マップの例
Fig.3 Example of connection-status tree.

ムで認識する方法について述べる．すでにネットワークに接続されているブロック(親ブロック)に新しいブロック(子ブロック)が接続されると、親ブロックは自分のブロック ID と子ブロックが接続された面 ID を Host PC に送信する．この接続面は Vcc 供給ホックのうち接続認識用に割り当てられたホックに子ブロックの Vcc 供給ホックから Vcc が供給されることにより認識する

子ブロックは、接続されると電源が供給され、同時にブロードキャストにより、自分のブロック ID と接続面 ID を送信する．Host PC は、これらの情報を用いて図 3 のようなツリー状のブロック接続状況マップを更新してブロックの接続状況を管理し、ベースブロックから個々のブロックまで、ブロック ID とお互いに接続されている面 ID を各ノードに保持する．これを新たなブロックの接続と同時に更新することにより、ブロックの接続状況をリアルタイムで認識することができる．

続いて、ブロック接続状況マップを利用した形状認

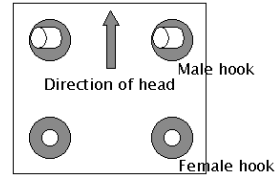


図 4 オス・メスホックの配置とヘッド方向の定義
Fig.4 Position of male-female hook and definition of direction of head.

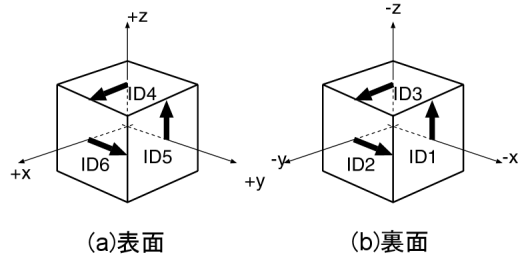


図 5 各面とヘッド方向の関係
Fig.5 Relation between each faces and direction of head.

識の方法について述べる．ブロック接続状況マップを更新するためには接続ブロックの接続方向の認識が必要となる．この認識における座標系は、すべて、左手系 3 次元直交座標系を用いる．またブロックの各面に割り当てる面 ID はサイコロのように面 ID1 の裏が面 ID6、面 ID2 の裏が面 ID5、面 ID3 の裏が面 ID4 となるように定義する．また、面の法線方向を各ブロック重心を原点とする座標軸の方向として定義する．ActiveCube の各ブロックは立方体であり、基本的に各面は等価である．接続時に面内で回転が生じることがあるが、現状では NeuronChip の信号ピンの制約から、これを認識することができない．そこで現在は、図 4 のようにオス・メスのホックを固定して用いることで面内回転を抑制することとする．ここで、ヘッド方向という概念を導入し、面の接続回転方向をヘッド方向によって一意に決定することにより、2 つのブロック間の接続・被接続はある一定の向きの面でのみ接続可能となる．また、ヘッド方向は展開図として管理し、各面の方向の演算を容易にするため、図 5 のように 3 次元的に対称となるように定義する．図のように、+X 軸面のヘッド方向を +Y 方向、+Y 軸面のヘッド方向を +Z 方向と定義すると、軸の負方向にも同様にサイクリックな関係が現れる．また、ヘッド方向が指す面 ID が 6 5 4 6 ...あるいは 3 2 1 3 ...と変化を繰り返すように面 ID を割り当てる．こうすることで、1 つの面の方向とその面のヘッド方向を決定すると、他の面の方向を一意に決定

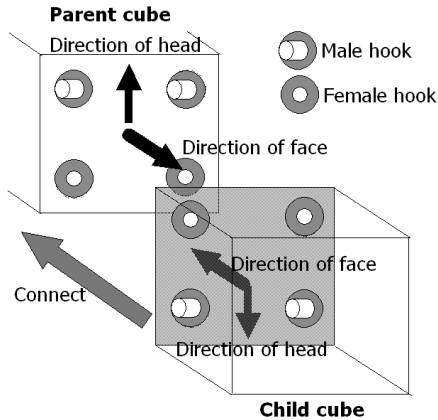


図 6 接続方向の認識

Fig. 6 Recognition of direction of new connection.

することが可能となる。ブロックの接続は各面のオスホックとメスホックが向かい合うように接続させなければならないので、接続面のヘッド方向は被接続面のヘッド方向と逆方向を向くことになる。さらに 1 つの面の面方向およびヘッド方向が決定すれば、すべての面方向とヘッド方向を求めることができる。

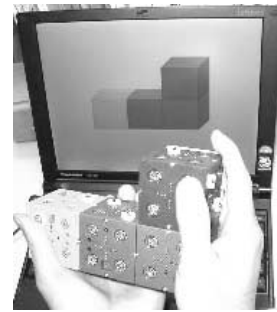
以上の接続ブロックの接続方向の認識手順をまとめると、次のようになる(図 6 参照)。

- (1) 被接続面(親ブロック)の面方向とヘッド方向をブロック接続状況マップより求める。
- (2) 接続面(子ブロック)のヘッド方向を被接続面のヘッド方向と逆であることより導出。
- (3) 接続面の面方向を被接続面の面方向と逆であることより導出。
- (4) 接続面のヘッド方向と面方向を決定。
- (5) 接続ブロックの+Z 軸面のヘッド方向および面の方向を接続状況マップに保存。

実際には、データ量を減らすために、接続されているブロックの+Z 方向の面 ID とその面のヘッド方向のみを接続状況マップに管理する。また、ベースブロックの+Z 方向の面 ID とその面のヘッド方向は、システム起動時に定義しておく。これにより接続状況マップを用いたリアルタイム 3 次元形状認識が可能となる。図 7 に、4 個と 39 個のブロックを用いて、リアルタイム 3 次元形状認識を行った様子を示す。

3.3 現在の制約

ActiveCube は、基本形状が一辺 5 cm の立方体であり、これを自由に組み合わせる様々な形状を作成することができる。多数のブロックを組み立てればさらに複雑な形状をも構築可能である。しかし、現状では LON のアドレッシングの制約から、同時に使用でき



(a) Four cubes.



(b) 39 cubes.

図 7 リアルタイム 3 次元形状認識の様子

Fig. 7 Snapshot of real-time recognition of 3D structure of connected cubes.

るブロックの数は 128 個以下である。しかしルータなどを使用すると、原理的には 39,000 個の接続が可能である。一方では、ブロックの重さは 1 個あたり約 100 g 程度あり、図 7 (a) のようにユーザが手に持って扱う際には、せいぜい 20 個程度が限界であると思われる。

3.2 節で述べたとおり、ブロック接続時に面内で回転が生じることがあるが、現状では NeuronChip の信号ピンの制約から、これを認識することができない。またブロックに供給する電源、および通信線はすべて平行であるので、すでに接続された複数個のブロックを同時に親ブロックに接続すると、接続したブロックすべてにほぼ同時に電源が供給され、それらがブロック ID を一度にブロードキャストするため、ホスト PC はブロックがどの順番で接続されたかを認識することができない。そのため、構築された形状を誤認識する可能性がある。また、ブロック接続状況は、図 3 のようなツリーをもとにして管理するため、ブロックの 2 面を同時に接続したり、複数のブロックで閉ループを作成したりすると、誤認識する可能性がある。これらの問題は、ActiveCube をフレキシブルなモデリングツールとして使用する際には、今後、解決しなければならない問題である。

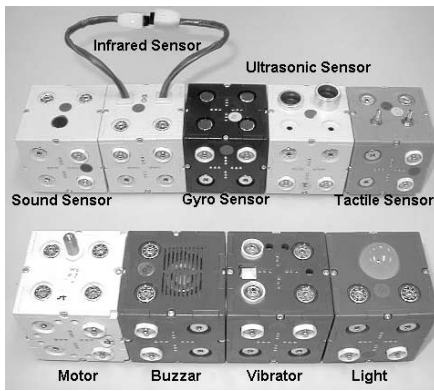


図 8 ActiveCube の入出力ブロックの例

Fig. 8 Examples of ActiveCube with input/output devices.

4. ActiveCube の入出力機能

双方向リアルタイムインタラクションの機能を実現する入出力機能の ActiveCube への実装について述べる。ActiveCube の各ブロックには NeuronChip が搭載されるので、ユーザの操作意図や実世界環境の情報を入力するブロック上のセンサを制御することが可能である。また、計算機内でのシミュレーション結果をブロックに実装されたアクチュエータやディスプレイで提示することも可能である。これらの入力・出力ブロックを使用することで、ユーザは様々なインタラクションを行うことができる。本章では、入出力機能を持つブロックと、これらを駆動するソフトウェアについて述べる。

4.1 入出力機能ブロック

現在、試作を進めている様々な入出力機能ブロックについて説明する。これらの例を図 8 に示す。これら以外にも、形状入力機能のみを持ち、入出力機能を有しないプレーンブロックも用意する。また一部のブロックは、システムワット社製の RoboCube¹⁵⁾ を拡張することにより、実現する。

超音波センサ 超音波センサは口径 16 mm、指向性 55 度 (−6 dB)、音圧 121 dB、感度 55 dB、周波数 40 KHz のものを使用する。これを使用して 3 m までの距離を測定可能であり、モードを切り替えることで測定可能距離は短い解像度を上げることできる。実際に ActiveCube では、測定可能距離 30 cm、解像度 0.1 cm で使用している。手のひらや他の物体からの距離を用いることにより、ユーザの操作意図を入力することが可能である。

タッチセンサ ユーザは 8 方向の入力が可能であり、仮想物体の操作や、マウスのクリックのような操

作ができる。

赤外線センサ 赤外線センサは発行部と受光部に分かれており、発行部の赤外線の波長は 950 nm、受光部が検知できる波長は 780–1000 nm である。センサから 20 mm 以内に物体があるかどうかを認識できる。

ジャイロセンサ ブロックの接続時の位置を初期位置とし、その位置から X 軸、Y 軸、Z 軸まわりの相対回転角度をそれぞれ 255 段階で測定できる。角速度は、使用しているジャイロセンサの仕様上、測定することができない。このブロックを用いることで、仮想空間内オブジェクトをユーザが傾けた方向に移動させたり、XYZ 軸の傾きにライトの RGB 値それぞれを対応させてリアルタイムに色を変化させるといったインタラクションが可能である。

音圧センサ ある閾値以上の音を感知すると True を、その閾値以下の音だと False を返す。この閾値はホスト PC で設定でき、仮想空間内のキャラクタをユーザが大声を出してびっくりさせるといったインタラクションが可能である。

ライト 緑・赤・青色を独立して 255 段階の明るさで点灯できる。すなわち 65535 色の表示が可能である。また、常時点灯モード、速い点滅モード、遅い点滅モードの 3 つのモードが選択でき、計算機内でのシミュレーション結果の提示用に使用する。

ブザー 単音だけではなく様々な音階の音を発音できる。さらに、連続音、断続音、サイレンなどといったモードが 4 つ用意されており、ライトブロックと同様に計算機内でのシミュレーション結果の提示用に使用する。

モータ 回転速度、回転方向、回転量を独立して制御できる。エンコーダ分解能は 1800 パルス/1 回転、ギア比は 225:1、モータトルクは最大 44g・cm、出力は 4 W、速度は 4–60 回転/分である。またブロックから軸が出ており、その軸にプロペラやタイヤなどを装着できる。モータブロックを使用して ActiveCube を自律移動させることも可能である。

振動素子 振動のオン・オフおよび振動間隔の制御が可能である。このブロックを使用することで、仮想空間内のオブジェクトが壁に当たったなどのフォースフィードバックを用いたインタラクションが可能である。

4.2 ソフトウェア

入出力機能を持つブロックを駆動するソフトウェア

表 1 入力および出力デバイスの因果関係の例
Table 1 Examples of causal relationships between input and output devices.

		measured physical values	output devices			
			light	buzzar	motor	vibrator
input sensors	ultrasonic	distance	brightness	loudness	rotational speed	frequency
	gyroscopic	3D orientation	color	tone/pitch	rotational speed/direction	frequency
	tactile	touch direction	color	sound mode	direction	frequency
	sound	loudness	brightness	loudness	rotational speed	frequency
	infrared	on/off	on/off	on/off	on/off	on/off

について述べる．各ブロックを制御するプログラムは NeuronC によってコーディング・コンパイルされ、各ブロック内の EEPROM に書き込まれる．このプログラムは 2 つのパートから構成される．1 つは形状認識のための部分であり、もう 1 つは、センサやアクチュエータの制御など、ブロック独自の動作を行うための部分である．ホスト PC からの命令は、ネットワーク変数と呼ばれる変数を用いて受信される．ActiveCube を用いてブロック独自の動作をさせるアプリケーションを開発する場合は、そのネットワーク変数に書き込む値だけを管理すればよい．また、ホスト PC 上のプログラムはすべて Microsoft 社の VisualBasic を用いてコーディングされる．このプログラムは、組み上げられたブロックの 3 次元形状を認識し、接続されているブロックの種類と数、およびその形状によって決められる動作を、個々のブロックのネットワーク変数に書き込む．ここで、ActiveCube を用いたアプリケーションの開発を容易にするため、3 次元形状認識とブロックの制御を行うライブラリは、ActiveX 技術を用いて実装する．

どのブロックが接続されているときにどのような動作をさせるかなど、入力および出力デバイスの因果関係は、ホスト PC 上のプログラムによって定義する．これらの関係は、様々な組合せやパターンが考えられるが、現在、我々が実装している一例を表 1 に記す．たとえば、ライトブロックと超音波センサを接続した場合、ライトブロックの明るさは、超音波センサによって測定される距離によって変更することができる．さらにブザーブロックを接続すると、超音波センサによって測定される距離によって、ライトの明るさだけでなく、ブザーの音量も変更できる．またライトブロックに超音波ブロックとジャイロブロックを接続すると、ライトの明るさは超音波ブロックで変更し、同時にライトの色はジャイロブロックによって計測される傾きで変更することもできる．このように ActiveCube では、複数入力に対する複数出力を組み合わせることも可能である．

各ブロックは固有の機能を持っているが、その制御

はホスト PC 上のプログラムが行っているため、組み立てたブロック全体の形状や入出力機能ブロックの接続位置などによって、これに応じた機能を持つ 1 つの物体として働かせることもできる．たとえば、ライトブロックを横に 3 個並べて接続すると信号機の動作をさせるといったことが可能である．これにより、ユーザはリアルタイム 3 次元形状モデリングだけではなく、モデリングした物体とリアルタイムインタラクションを行うことができる．

5. ActiveCube の即時応答性の評価

ユーザが ActiveCube を用いて何らかの操作を行う場合、その操作に対する応答が返るまでに、各ブロックとホスト PC の間のネットワーク遅延や、ホスト PC の処理遅延などがある．実際にユーザが ActiveCube に対して行うことのできる操作には、ブロックの接続・分離、入力ブロックに対するインタラクションがあり、これらをホスト PC が認識するまでの時間を検討する必要がある．またホスト PC が出力ブロックに対して命令を送信し、実際に出力ブロックが動作するまでの時間も検討しなければならない．そこで本章ではこれらの応答時間を測定し検討する．

5.1 実験

プレーンブロック、ライトブロック、ベースブロックの 3 種類に加え、ホスト PC からの命令を受信すると、ホスト PC に対して即座に応答を返す Ping ブロックを使用する．ホスト PC としてノートパソコン（松下電器産業 CF-L1, PentiumIII 600 MHz, 192 MB, XGA 13.3 inch）を用いる．また実験用ソフトウェアは Microsoft 社の VisualBasic6.0 を用いて制作する．
実験 1 ホスト PC と末端ブロック間の往復通信時間
ホスト PC と末端ブロックとの往復通信の時間を Ping ブロックを用いて測定する．ActiveCube では各種入力ブロックからのデータ受信を、ホスト PC が 0.1 秒ごとに定期的にポーリングを行うことで実現しているが、実際の所要時間を測定する．ホスト PC からの信号出力はルータ、ベースブロック、いくつかのプレーンブロックを経て Ping ブロックに到達し、そ

れを受けて Ping ブロックは信号をホスト PC に返す．その間の時間をホスト PC のタイマを用いて測定する．ここで，Ping ブロックとベースブロック間に接続するプレーンブロックの数が 0 個のときを距離 1，1 個増えるごとに距離を 1 増やし，本実験では距離 6 (プレーンブロック 5 個) までを測定する．それぞれの距離に対し 500 回試行する．

実験 2 ホスト PC による出力ブロックへの命令送信からブロックの動作完了までの時間

ホスト PC からブロック群の末端に接続したライトブロックにライトを点灯させる信号を送信し，実際にライトが点灯するまでの時間を測定する．

まず，ベースブロックとライトブロックを接続する．次に，ホスト PC 画面上のボタンをクリックするとホスト PC はライトブロックに赤色のライトを点灯させる命令を送信する．同時にホスト PC 画面上のラベル色が変化する．実験 1 と同様にベースブロックとライトブロックの間にプレーンブロックを接続し，距離を 1, 2, 3 と変化させて，ホスト PC 画面上のラベル表示が変化した時点から，ライトブロックのライトが点灯するまでの時間を 10 回測定する．応答時間の測定機器としてデジタルビデオカメラ (Sony DCR-TRV900, NTSC, 29.97 fps=33.4 msec/frame) を用いる．

実験 3 ホスト PC のブロック接続・分離認識時間

ベースブロックとライトブロックとを接続・分離し，その接続・分離をホスト PC が認識するまでの時間を測定する．まず，電源が供給されるとライトが点灯するようにプログラムされたライトブロックを用意する．ライトブロックをベースブロックに接続し，それと同時にライトブロックが点灯する．ライトブロックが接続されたことをホスト PC が認識すると，ホスト PC 画面上のラベル色が変化する．分離についても同様に，分離した瞬間にライトが消灯し，ライトブロックが分離したことをホスト PC が認識するとホスト PC 画面上のラベル色が変化する．この，ライトブロックが接続・分離された時点からホスト PC が接続・分離を認識するまでの時間をデジタルビデオカメラを用いて 10 回測定する．

5.2 実験結果と考察

実験 1 の結果を図 9 に示す．グラフより，ブロックの数が増えても応答時間はほぼ一定であり，全体の平均は 248.2 msec であった．また実験 2 の結果 (図 10) より，ホスト PC からの命令を受けて末端の出力ブロックが動作するまでの平均時間は 447.56 msec であった．また，ブロック間距離の増加による遅延時間はほとんどなく，実験 1 の結果が裏付けられた．実験 3 の結果，

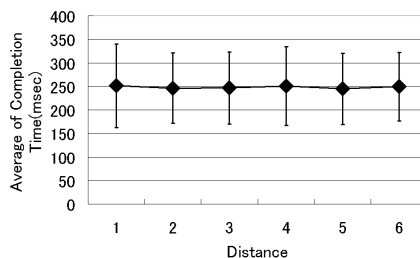


図 9 ホスト PC-ブロック間の往復通信時間

Fig. 9 Communication time between host PC and cubes.

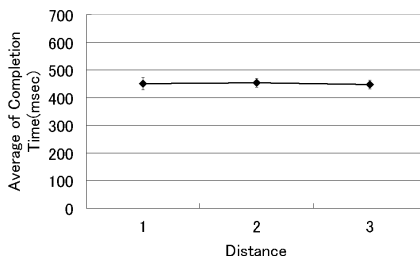


図 10 出力ブロックの動作完了時間

Fig. 10 Completion time of output cube.

ブロックの接続を認識する時間は平均で 951.9 msec (標準偏差 30.8 msec) であり，分離を認識する時間は平均で 965.26 msec (標準偏差 27.7 msec) であった．また，個々の結果よりその時間はほぼ一定で，ばらつきが小さいことが分かった．分離の認識時間の方が若干長くなっているが，これはホスト PC が分離処理をする際に，接続状況マップのノードの分離ブロック以降をすべて再呼び出しにより処理するためにかかる時間だと思われる．

一般にユーザインタフェースでは，タスクの種類に応じた適切な時間内に，レスポンスが返されることをユーザは期待するが，コマンドラインからの入力のような比較的単純なタスクでは，1 秒程度以下のシステム応答時間が好ましいとされている¹⁶⁾．ここで述べた実験結果は，ActiveCube に対するインタラクションの反応時間がいずれも 1 秒以内であり，この点からも ActiveCube は，自然なインタラクションを実現していると考えられる．

実際に ActiveCube を用いたアプリケーションでは，これら 3 種類の処理が並行して行われる可能性がある．しかし，ブロックの接続・分離と同時に入出力デバイスを用いてインタラクションを行うといった操作は考えにくい．また，ブロックの入力センサを用いて何らかの出力を得る操作をユーザが行った場合は，実験 1, 2 の結果の和と考えられるので，平均で 695.7 msec の応答時間となり，即時応答性を満たしているといえ

る。ただし、出力ブロックの制御と入力ブロックからのデータ受信は、並列処理ではなく逐次処理であるため、制御する出力ブロックの数が増えたり、入力ブロックの数が増えた場合、すべての処理が完了するまでの応答時間は長くなる。しかし、その間にも、接続しているいずれかのブロックに対する入出力は動作しているので、ユーザに対してまったく応答が返らないという時間は1秒以内であると考えられる。また、文献7)のシステムでは、構成するブロックの数が多くなればなるほど、形状認識に要する時間が増加する。たとえばブロックの数が98個の形状を認識するまで35秒要し、560個のときは53分要することが報告されている。ActiveCubeは距離6で150個程度のブロックの接続が可能であるが、実験の結果よりブロックの数が増えても形状の認識に必要な時間はほとんど増加しない。この差を生じる主な原因としては、ブロックの接続方式がシリアルかパラレルかの違いが考えられる。

6. おわりに

ActiveCubeを用いてリアルタイム3次元形状モデリングとリアルタイムインタラクションを実現するために必要なハードウェア、リアルタイム3次元形状認識の手法、ソフトウェア、様々な入・出力機能の詳細について述べ、ユーザインタフェースとしての有効性を検討する実験について述べた。ユーザはActiveCubeを組み立てることで直感的にリアルタイム3次元モデリングを行い、さらに入・出力機能を使用することでリアルタイムインタラクションが可能となる。今後は、入出力ブロックのさらなる試作・充実を図り、ActiveCubeのリアルタイム3次元形状モデリングとリアルタイムインタラクション機能を最大限に生かしたアプリケーションを実現していく予定である。

謝辞 本論文の執筆にあたり、日頃ご議論いただく大阪大学大学院工学研究科正城敏博講師に厚く御礼申し上げます。また、ActiveCube試作に関してお世話になった株式会社システムワット菊池日出男氏に深い感謝の意を表す。なお、本研究の一部は、科学研究費補助金基盤研究(B)(2)12480096、ならびに栢森情報科学振興財団の助成を受けた。

参 考 文 献

- 1) Shneiderman, B.: *Designing the user interface — strategies for effective human-computer interaction*, 3rd edition, Addison-Wesley (1998).
- 2) Murakami, T. and Nakajima, N.: Direct and intuitive input device for 3-D shape defor-

- 3) Hincley, K., Pausch, R., Goble, J.C. and Kassell, N.F.: Passive real world props for neurosurgical visualization, *Proc. Conference on Human Factors in Computing Systems (CHI '94)*, pp.465–470 (1994).
- 4) Esposito, C. and Paley, W.B.: Of mice and monkeys: a specialized input device for virtual body animation, *Proc. Symposium on Interactive 3D Graphics*, pp.109–114 (1995).
- 5) Johnson, M.P., Wilson, A., Kline, C., Blumberg, B. and Bobick, A.: Sympathetic interfaces: using a plush toy to direct synthetic characters, *Proc. Conference on Human Factors in Computing Systems (CHI '99)*, pp.152–158 (1999).
- 6) Strommen, E.: When the interface is a talking dinosaur: learning across media with ActiMates Barny, *Proc. Conference on Human Factors in Computing Systems (CHI '98)*, pp.288–295 (1998).
- 7) Anderson, D., Frankel, J., Marks, J., Agarwala, A., Beardsley, P., Hodgins, J., Leigh, D., Ryall, K., Sullivan, E. and Yedidia, J.: Tangible interaction + graphical interpretation: a new approach to 3D modeling, *Proc. SIGGRAPH 2000*, pp.393–402 (2000).
- 8) Gorbet, M.G., Orth, M. and Ishii, H.: Triangles: tangible interface for manipulation and exploration of digital information topography, *Proc. Conference on Human Factors in Computing Systems (CHI '98)*, pp.49–56 (1998).
- 9) 鈴木, 加藤: アルゴブロック: アルゴリズム教育のための物理言語, 第8回ヒューマンインタフェースシンポジウム論文集, pp.245–248 (1992).
- 10) Frazer, J.H., Frazer, J.M. and Frazer, P.A.: Three-dimensional data input devices, *Proc. Conference on Computers/Graphics in the Building Process* (1982).
- 11) Frazer, J.H.: *An evolutionary architecture*, Architectural Association, London (1995).
- 12) Aishi, R. and Noakes, P.: Architecture without numbers — CAAD based on a 3D modeling system, *Computer-Aided Design*, Vol.16, No.6, pp.321–328 (1984).
- 13) Anagnostou, G., Dewey, D. and Patera, A.T.: Geometry-defining processors for engineering design and analysis, *The Visual Computer*, 5, pp.304–315 (1989).
- 14) 根岸 哲: 装置間通信を容易にした知的分散制御ネットワーク「LON」, 計装, Vol.36, No.12, pp.54–62 (1993).

- 15) <http://www.watt.co.jp>
 16) Butler, T.W.: Computer response time and user performance, *Proc. Human Factors in Computer Systems (CHI '83)*, pp.56-62 (1983).

(平成 12 年 11 月 1 日受付)
 (平成 13 年 4 月 6 日採録)



伊藤 雄一

昭和 50 年生。平成 12 年大阪大学大学院工学研究科電子情報エネルギー工学専攻博士前期課程修了。同年同専攻博士後期課程入学。ヒューマンインタフェースの研究に従事。日本バーチャルリアリティ学会、電子情報通信学会各会員。



北村 喜文(正会員)

昭和 37 年生。昭和 62 年大阪大学大学院基礎工学研究科博士前期課程修了。同年、キヤノン株式会社情報システム研究所、平成 4 年 ATR 通信システム研究所、平成 9 年より大阪大学大学院工学研究科助教授、現在に至る。ヒューマンインタフェース等の研究に従事。博士(工学)。ACM, IEEE, ヒューマンインタフェース学会, 日本バーチャルリアリティ学会等会員。



河合 道広

昭和 50 年生。平成 11 年大阪大学工学部通信工学科卒業。同年大阪大学大学院工学研究科電子情報エネルギー工学専攻博士前期課程入学。ヒューマンインタフェースの研究に従事。ヒューマンインタフェース学会員。



岸野 文郎

昭和 21 年生。昭和 46 年名古屋工業大学大学院電子工学専攻修士課程修了。同年日本電信電話公社(現 NTT)電気通信研究所入所。同ヒューマンインタフェース研究所において、高速・広帯域通信システムの研究・実用化、画像処理の研究に従事。平成元年 ATR 通信システム研究所知能処理研究室室長。画像処理、知能処理、臨場感通信の研究に従事。平成 8 年より大阪大学大学院工学研究科教授、現在に至る。博士(工学)。電子情報通信学会、ヒューマンインタフェース学会、日本バーチャルリアリティ学会等会員。