

# StackBlock: 積み重ね形状認識可能なブロック型 UI

安藤正宏<sup>†1</sup> 伊藤雄一<sup>†1</sup> 細井俊輝<sup>†2</sup> 高嶋和毅<sup>†2</sup> 北村喜文<sup>†2</sup>

本研究では、柔軟なブロックの積み重ねを認識する新たなブロック型 UI “StackBlock” を提案する。ユーザは積み木のように任意の位置や角度でブロックを積み重ねることにより、3次元形状を構築することができる。StackBlockの各表面には赤外線 LED とフォトトランジスタをマトリクス状に敷き詰めており、ブロックを積み重ねると、向かい合う赤外線 LED とフォトトランジスタから接触領域を認識し、また、同様の赤外線 LED とフォトトランジスタを用いて、ブロック間で接触情報を赤外線通信する。このブロック間での赤外線通信のリレーにより全ての接触情報を一番底に配置されたブロックに集約し、ホスト PC に送信する。ホスト PC は接触情報から構築形状を認識し画面上に再現する。我々は StackBlock のプロトタイプを実装し、精度や形状認識の速度の評価を行い、柔軟な積み重ねを用いた 3次元形状認識が実現できていることを確認した。

## StackBlock: A Block-shaped Interface to Recognize Stacked Blocks

MASAHIRO ANDO<sup>†1</sup> YUICHI ITOH<sup>†1</sup> TOSHIKI HOSOI<sup>†2</sup>  
KAZUKI TAKASHIMA<sup>†2</sup> YOSHIFUMI KITAMURA<sup>†2</sup>

We propose a novel building-block interface called StackBlock that allows users to precisely construct 3D shapes by stacking blocks at arbitrary positions and angles. Infrared LEDs and phototransistors are laid in a matrix on each surface of a block to detect the areas contacted by other blocks. Contact-area information is transmitted to the bottom block by the relay of infrared communication between the stacked blocks, and then the bottom block sends all information to the host computer for recognizing the 3D shape. We implemented a prototype of StackBlock with several blocks and evaluated the accuracy and latency of 3D shape recognition. As a result, StackBlock could sufficiently perform 3D shape recognition for users' flexible stacking.

### 1. はじめに

コンピュータとより直感的にインタラクションする方法として、タンジブルユーザインタフェース (TUI) と呼ばれる実物体を操作対象とするインタフェースが提案されている[1-5]。TUI の中でもブロックを組み上げる操作により 3D モデルを構築できるブロック型 UI に注目が集まっている。ブロックなどの実物体を用いて形状を構築する行為は、専門的な知識を必要とせず、3次元空間での空間認知を容易にするため、建築設計や工業デザインなどの分野において、ブロック型 UI を用いてコンピュータに形状を入力し 3D モデリングを行う研究が行われている。

従来のブロック型 UI は構築形状の入力のために、ブロック同士をホックや磁石で固定し電氣的に認識する手法[6-9]や、可視光により認識する手法[10]、構築形状をカメラで認識する手法[11,12]などを用いてきた。しかし、複数のユーザが操作する場合、カメラによる認識手法ではオクルージョンの問題により正確な形状認識が困難である。また、電氣的に認識する手法ではホックや磁石といったブロック間のコネクタを必要とするため、ブロック同士の接続の自由度が低く、ブロックの接続や分離には力が必要となる。このように、従来のブロック型 UI では複数のユーザがブロックを任意の位置や角度で接続し、それを正確に認識することができないため、ユーザのより自由で柔軟な形状構築、そ

してその認識は不可能であった。

そこで我々は、積み木に着目し、自由な積み重ね形状を正確に認識するブロック型 UI “StackBlock” を提案する。ブロックの積み重ねを正確に認識することが可能となれば、様々な応用が期待できる。例えば、従来、積み木は慢性的な心的ストレスを被った子供に対する心理的ケアに、遊戯療法として用いられてきた[13]。この療法では、子供の積み木遊びをビデオでコーディングしたり、行動や操作を専門家が解釈したりすることが主な測定方法であったが、不安定な心理状態にある子供の遊び方の微妙な変化を正確に計測することは難しかった。そこで、子供の積み木遊びを正確に認識できれば、子供の心理状態を定量的にかつリアルタイムに推定することにつながり、心的ストレスに対する積み木遊戯療法にも新たな展開が期待できる。

本システムでは、ユーザは、積み木のように任意の位置や角度でブロックを積み重ねることによって 3次元形状を構築することができ、またホスト PC 上でリアルタイムに構築形状を認識する。ブロックの各表面には赤外線 LED と

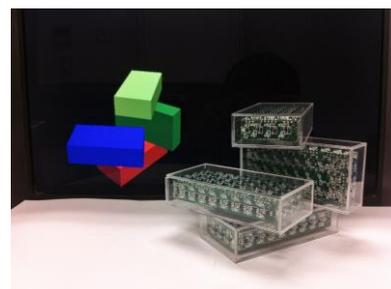


図 1 実装した StackBlock

Figure 1 Implementation of StackBlock.

<sup>†1</sup> 大阪大学大学院情報科学研究科  
Graduate School of Information Science and Technology, Osaka University

<sup>†2</sup> 東北大学電気通信研究所  
Research Institute of Electrical Communication, Tohoku University

フォトトランジスタをマトリクス状に敷き詰めており、ブロックを積み重ねた時の赤外線 LED の発光とフォトトランジスタの受光状況から積み重ね領域を認識する。そして、対面する赤外線 LED とフォトトランジスタ間での赤外線通信により、積み重ね情報を送受信する。積み重ね情報はブロック間の赤外線通信のリレーにより、最終的にコンピュータに送られコンピュータ上で構築形状が再現される。また赤外線を用いているため、ブロックと手の接触を認識することも可能である。

実装した StackBlock のプロトタイプの外観を図 1 に示す。図に示すように、StackBlock は、ユーザの自由で柔軟な積み重ねを正確に認識するブロック型 UI である。また、オクルージョンによる問題が発生しないため、複数のユーザが同時に形状構築する場合もその形状認識も可能である。

## 2. 関連研究

### 2.1 ブロック型 UI

ブロック型 UI は、実世界のブロックを用いることにより専門的な知識がないユーザの 3 次元モデリングを可能とする。こうしたブロックでの構築形状の認識方法は様々なものが提案されてきた。

その代表的なものとして、ブロック内部にセンサを埋め込んだ手法がある。Anderson らは、LEGO 型のブロックを用いて複雑な形状をコンピュータに入力できるシステムを提案している[6,7]。このシステムはユーザが自由に形状を組み立てた後にコンピュータに接続し、電源を投入すると、ブロックの 3 次元形状を認識し、その形状のモデリングを行うシステムである。各ブロックには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により構築形状を認識する。また、Watanabe らが提案した ActiveCube は各ブロックにマイコンを搭載したブロック型 UI で、Anderson らの手法と同様、コンピュータと各ブロック間の電気的な通信により構築形状を認識する[8,9]。ActiveCube は 3 次元形状の認識だけでなく、ユーザの操作意図を入力するためのセンサや、コンピュータの内部情報やシミュレーション結果を提示するための出力デバイスを備えたブロックを装備することにより、リアルタイムな双方向のインタラクションも実現している。これらのブロック間で電気的な通信を行うブロック型 UI では、安定したブロック間通信のために、ブロック同士を密着して接続する必要があり、LEGO のホックや磁石が用いられている。その結果、接続の向きの自由度が低く、ブロックの接続や分離に力が必要となる。

また、李らが提案した Bloxels はブロック間で光通信を行うブロック型 UI であり、ブロックが積み上げられたとき、隣接ブロックからの色のデータを受信しフルカラーで発光することで 1 つのピクセルの機能をなし、全体で 1 つの形状ディスプレイとして機能する[10]。Bloxels は光通信を行

うため、接続や分離に強い力を必要とせず、接続の向きの自由度も高い。しかし、形状認識のためのセンサを備えておらず、積み重なったブロックの ID などの情報を取得できないため、構築形状の認識は不可能である。

一方で、外部のセンサによるブロック形状認識手法も多く提案されている[11,12]。Miller らは Kinect を用いてブロックの構築形状を認識し、形状を PC 上に再現する手法を提案している[11]。これらのカメラベースの形状認識手法はブロック同士を密着して結合する必要がないため、接続・分離に強い力を必要とせず、接続の向きの自由度が高いという利点がある。しかし、これらの手法では、カメラから取得したデータをもとに形状認識を行うため、特にユーザが複数の場合、ユーザの手でブロックが隠れたり、構築形状の内部に空洞が生じたりした場合、正確な形状の認識が難しい。このオクルージョンによる問題はユーザが複数の場合には特に顕著に表れる。StackBlock では、接続の向きの自由度も高く、またオクルージョンによる問題の発生しない形状認識手法を赤外線により実現する。

### 2.2 赤外線による接触認識

Stackblock では積み重ねを認識するために、ブロック間で重なっている領域を検出する必要がある。こうした接触を認識する方法として、赤外線を用いた接触認識手法が主にタッチインタラクションの実現のために検討されてきた。

赤外線による接触認識手法として Han らは、FTIR と呼ばれる手法を提案している[14]。アクリルパネルのエッジから赤外線を照射し、全反射した状態で指などを当てると、触れた部分の反射率が変化し全反射が崩れ、赤外線が漏れ出す。FTIR はこの漏れ出した赤外線をディスプレイの裏側に取り付けられた赤外線カメラなどで撮影しタッチを認識する手法である。また、Echtler らが提案した Inverted FTIR はディスプレイの裏側に液晶ディスプレイを取り付け、赤外線カメラをディスプレイ上部に設置し、液晶ディスプレイで反射した赤外線を撮影しタッチを認識する[15]。Inverted FTIR では、通常の FTIR では不可能である液晶ディスプレイによる認識結果の提示が可能となる。

一方で、松下らは DI と呼ばれる手法を提案している[16]。ディフューザーを取り付けたディスプレイの裏側に赤外線を照射した状態で指などを当てると、触れた部分では赤外線は分散されない。DI はこの分散されない赤外線を赤外線カメラなどで撮影しタッチを認識する手法である。また、Benko らの提案した Sphere は DI を球状に応用し、球状のディスプレイでのタッチ認識を可能としている[17]。

このように赤外線を用いた接触認識では赤外線の受光さえ確保されれば接触認識が可能であり、接触対象の物体を自由に動かすことが可能である。StackBlock ではこのような赤外線を用いた接触認識を用いた形状認識手法を検討する。

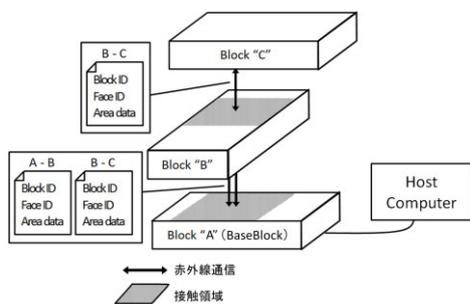


図 2 システム全体像  
Figure 2 System Overview.

### 3. StackBlock

#### 3.1 概要

図 1 に示すように、StackBlock はユーザの積み重ね形状構築を認識する。そこで、我々は赤外線 LED とフォトトランジスタをブロックの 6 面全てに敷き詰め、赤外線を用いた積み重ね認識手法を提案する。システム全体像を図 2 に示す。本手法では、赤外線を受光から積み重ねを検出し、そのブロック間の接触領域を推定し、赤外線通信を用いてデータ通信を行うことにより、赤外線のみで積み重ね認識を行うことを可能とする。特に積み重ねにおけるブロック間の接触領域推定の精度を上げるために、赤外線 LED とフォトトランジスタをブロック表面に敷き詰める。こうした積み重ね認識のためには、どのブロックが、どのブロックの、どの面に、どのように乗っているかを接触箇所毎に検出し、通常のブロックとは異なりホスト PC に接続された最も下に存在する BaseBlock に集約し、ホスト PC に送信する必要がある。そのためには、接触箇所ごとの、お互いのブロックの ID と、それぞれの接触面の ID、それぞれの接触領域情報（以下ではこれらを接触情報と呼ぶ）が必要である。また、接触領域情報とは、ブロック表面において、どのフォトトランジスタが接触領域内に存在するかを示す。

そこで、ブロック表面に赤外線 LED とフォトトランジスタをマトリクス上に敷き詰める方法を用いる。ユーザがブロックを積み重ねると、ブロック表面に配置されたフォトトランジスタの内、対面するブロックと接触している領域内のフォトトランジスタは、対面するブロック表面に配置された赤外線 LED からの赤外線を受光するため、ブロック表面のフォトトランジスタの受光状況に基づいて接触領域の認識が可能となる。

また、このように赤外線 LED とフォトトランジスタを敷き詰めたブロックでは、対面するブロック間での赤外線通信が可能となる。そこで、対面するブロック間で接触情報を赤外線通信で交換することによって、各接触箇所における接触情報を、一方のブロックに集約できる。

これを利用して、自分より下のブロックへ自身の接触情報を伝搬する。これを繰り返し、全接触箇所の接触情報を、BaseBlock に集約する。BaseBlock は、集約した接触情報を

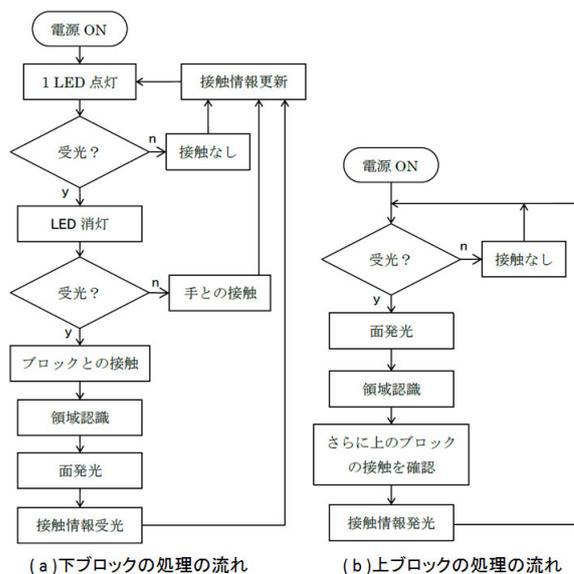


図 3 各ブロックのフローチャート  
Figure 3 Flowchart.

ホスト PC に定期的送信する。ホスト PC は BaseBlock から受信した全接触箇所の接触情報に基づき構築形状を復元する。

このようにして、提案手法では、接触領域の認識に用いる赤外線 LED とフォトトランジスタを、情報伝搬のための通信にも用いることで、ホスト PC との通信のための追加的なハードウェアを各ブロックに搭載せずに、ホスト PC への情報伝搬を実現する。

さらに、ブロック表面に赤外線 LED とフォトトランジスタを敷き詰めているため、ブロックを持つと、手とブロックの接触面において赤外線が反射する。この赤外線の反射を認識することにより、手との接触領域を認識することも可能である。

#### 3.2 各ブロックの処理

積み重ね認識のための各ブロックの処理フローについて、下に置かれたブロックのものを図 3 (a)に、上に載せられたブロックのものを図 3 (b)に示す。BaseBlock を含む、既に構築され認識されたブロック（以下、下ブロック）の上面は、上面にブロックや手などの物体が存在するかどうかをセンシングするために順番に 1 つずつ赤外線 LED を点灯し、受光を待つ。いずれかの下ブロックに接触した新たなブロック（以下、上ブロック）は、この下ブロックからの赤外線を受光すると、下面の全赤外線 LED を一定時間点灯する。また、下ブロックと手が接触する場合には、下ブロックからの赤外線が手で反射する。下ブロックは、受光が確認されない場合、点灯した LED の上には接触がないと判断し、次の LED を点灯させ上記処理に戻る。もし下ブロックが受光を確認すると、上面にブロックか手の接触があると判断し、ブロックと手の判別を行うために点灯していた赤外線 LED を消灯する。このときブロックとの接触の場合には上ブロックからの赤外線により受光が確認されるが、手との

接触の場合には赤外線が反射が起これないため受光は確認されない。下ブロックはブロックとの接触が確認されると、受光した上面のフォトトランジスタの受光状況に基づき接触領域を認識し、その後上面の全赤外線 LED を一定時間点灯する。上ブロックは下面のフォトトランジスタの受光状況に基づき接触領域を検出する。次に、上ブロックはさらに上にブロックがあるかどうかを確認するために、上ブロックの上面に対して図 3(a)の処理を実行する。上ブロックの確認が終わると、最後に、上ブロックは、自身の接触情報、及び、自身より上のブロックから受け取って保持している全ての接触情報を下に送信する。これを繰り返すことにより、最も下に敷かれた BaseBlock に全ての接触面での接触情報を集約し、BaseBlock がホスト PC にこれらの情報を送信する。

### 3.3 形状認識手法

本節ではホスト PC が、ブロックから受信した接触情報から構築形状を認識する手法について述べる。

接触情報は、すべての接触箇所における互いのブロック ID、接触する面 ID、そして接触領域の情報により構成されている。得られたブロック ID、面 ID からブロック同士の対応付けを行い、接触領域情報からブロック同士の接触の位置、角度を決定する。ただし、接触部分の領域情報は、ブロック表面において、どのフォトトランジスタが接触領域内に存在するかを示す。

接触領域情報から接触の位置、角度を算出する前提として、「接触する 2 個のブロックが作る接触領域の図形は、両ブロックにおいて合同」である。

ブロックの接触位置は、接触領域に含まれる点の座標の平均化により算出する。計算には  $x$  軸、 $y$  軸をもつ 2 次元座標系を用いる。ただし、使用する座標系はブロックの位置や角度に依存する相対的な座標系である。ここで、 $Cont(k)$  を以下のように定義する。

$$Cont(k) = \begin{cases} 1 & (k \text{ 番のフォトトランジスタが接触領域内}) \\ 0 & (k \text{ 番のフォトトランジスタが接触領域外}) \end{cases}$$

接触位置は、接触領域に含まれるフォトトランジスタの座標の相加平均をとることにより算出する。つまり、 $k$  番目のフォトトランジスタの座標を  $(x_k, y_k)$  とすると、接触位置の座標  $(x_T, y_T)$  は

$$(x_T, y_T) = \sum_{k=1}^{num} Cont(k) * (x_k, y_k)$$

で与えられる。ただし、 $num$  はブロック表面上の全フォトトランジスタ数である。

ブロックの接触角度は、接触領域の図形が、接触面に対してどの程度傾いているかを計算することにより決定する。算出した接触位置  $(x_T, y_T)$  を通り、接触領域内にあるフォトトランジスタとの距離の二乗和が最小となるような直線を

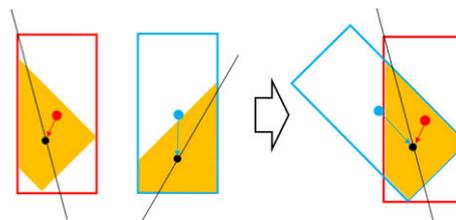


図 4 接触の位置、角度の決定

Figure 4 Determination of the Position and the Angle.



図 5 素子の配置

Figure 5 Arrangement of Elements.

算出し、その傾きを  $\tan\theta$  とする。この  $\theta$  を接触領域の角度とする。

図 4 に示すように、接触する 2 個のブロックにおける接触位置と直線が空間内で一致するように整合することにより、2 個のブロックの構築形状が認識される。この処理を全接触箇所について行い、ブロック同士の対応付けを行うことにより、構築された全体の形状認識を行う。

### 3.4 StackBlock の構成・実装

StackBlock の実装のための要件は次の通りである。

- ブロックの制御
  - マイクロコントローラによる制御
  - 2 つの 3 面ブロックによる 6 面ブロックの構成
  - マイクロコントローラの I/O ポート拡張
- 接触領域の認識
  - 赤外線 LED による発光
  - フォトトランジスタによる受光強度認識
- 赤外線通信
  - 赤外線 LED によるパターン発光
  - フォトトランジスタによる受光パターン認識

表 1 実装に用いた素子

Table 1 Elements of Use.

素子名	型番	製造元
赤外線 LED	TAN1111C	Stanley
フォトトランジスタ	TEMT1020	Vishay
リチウムポリマー電池	ID-1578	Adafruit
マイクロコントローラ	R5F2123CJFP	ルネサスエレクトロニクス
シフトレジスタ	TC74VHC595FT	東芝
デコーダ	CD74HCT238	Texas Instruments
マルチプレクサ	TC74LVX4051FT	東芝

実装に用いた素子の一覧を表 1 に示す。StackBlock は一般的な木製の積み木をモデルとして、縦 50 mm、横 100 mm、高さ 25 mm とし、重さは 100 g 以下を実現している。また、図 5 に示すように、ブロックの 6 面全てに赤外線 LED とフォトトランジスタを 1cm 四方あたりに 1 つずつ、マトリクス状に配置する。これは、使用する赤外線 LED の半値角は 145 度であり、1cm<sup>2</sup> の領域に 1 つずつ赤外線 LED を配置すれば、図 6、図 7 に示すようにブロック表面に少なくとも半値以上の強度の赤外線を隙間なく放射できるためである。

図 8 に示すように、ブロックの各面は 2 つの 3 面ユニットの組み合わせによって構成する。2 つの 3 面ユニットはそれぞれ 1 つずつのマイクロコントローラによって制御し、電源供給とユニット内通信のために有線で接続する。また、電源供給は 1 つのリチウムポリマーバッテリーにより実施する(3.7V, 500mAh)。2 つのユニットによる構成上、ブロックを積み重ねた際の下面と上面は別の 3 面ユニット上にあるため、積み重ね認識の際にはユニット間通信が必要となる。

ユニットの制御には、マイクロコントローラの I/O ポートでは不足するため、シフトレジスタやデコーダ、マルチプレクサといった素子を用いて I/O ポートを拡張する。また、接触領域の認識にはマイクロコントローラの A/D コンバータ機能を用いて、各フォトトランジスタの受光強度を認識し、しきい値による判定を行う。赤外線通信はマイクロコントローラのシリアル通信機能を用いて実装した。

この 2 つのユニットを組み合わせたものを、2 mm 厚の亚克力板のケース内に収納し、一つのブロックとする。

#### 4. 柔軟な積み重ね認識の評価

提案した赤外線 LED とフォトトランジスタによる積み重ね認識では、形状認識精度や、積み重ね認識のリアルタイム性が重要となる。そこで、実装したプロトタイプを用いて、1) 形状認識の精度、及び、2) 積み重ね認識のリアルタイム性をそれぞれ評価する。

##### 4.1 形状認識の精度に関する評価

StackBlock は各接触箇所において接触するブロック ID や面 ID だけでなく、接触領域情報も認識する。そのため、ブロック表面に敷き詰められた赤外線 LED の発光とフォトトランジスタの受光強度から導かれる接触領域情報は、形状認識の精度に直接的に影響を及ぼす。本節の評価では 1) 積み重ね時のずれに対するフォトトランジスタの受光強度、及び、2) 積み重ね時のブロック間距離に対するフォトトランジスタの受光強度をそれぞれ評価する。

##### 4.1.1 積み重ね時のずれに対するフォトトランジスタの受光強度の評価

StackBlock はユーザの自由な積み重ねによる形状構築およびその認識を可能とするブロック型 UI であるため、ブロックをずらして積み重ねた場合にも正確にその形状を認識する必要がある。そこで、ブロックをずらして積み重ねた

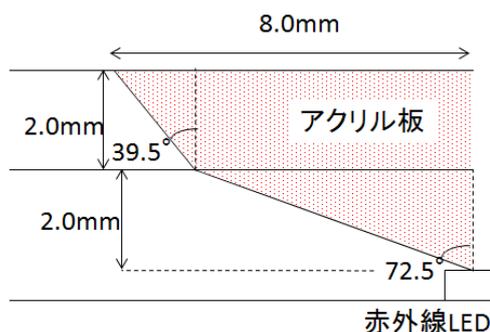


図 6 赤外線の強度分布 (側面図)

Figure 6 Intensity Distribution of Infrared(side view).

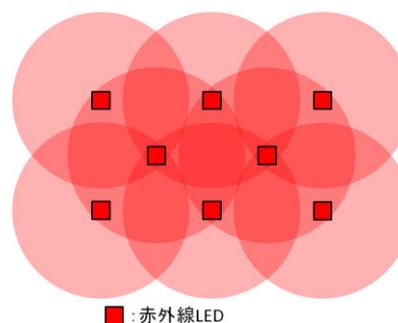


図 7 赤外線の強度分布 (平面図)

Figure 7 Intensity Distribution of Infrared(plane view).

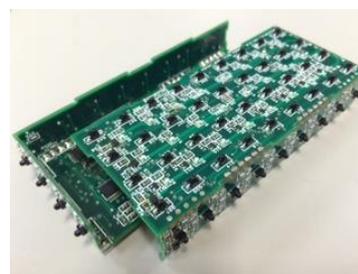
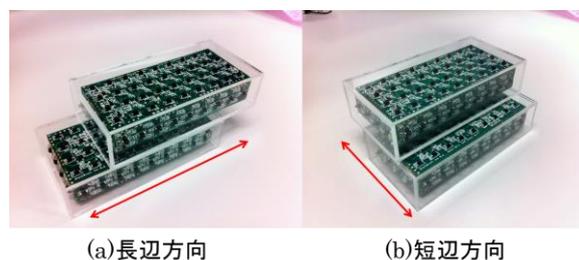


図 8 6面ブロックの構成

Figure 8 Configuration of the Six-Sided Block.



(a)長辺方向

(b)短辺方向

図 9 ブロックの積み重ね

Figure 9 Stack of Blocks.

際に、フォトトランジスタの受光強度がどのような分布を示すのか評価する。

本評価では、2 つのブロックの大きい面同士を積み重ね、上のブロックの赤外線 LED を 1 個点灯させ、下のブロックのフォトトランジスタを 1 個受光強度認識させる。そこで、

点灯する赤外線 LED と受光強度認識するフォトトランジスタをずれのないよう対面する位置を初期状態とし、以下に示す各場合において、フォトトランジスタの受光強度を評価する。

- 下のブロックの長辺方向に、上のブロックを 1mm ずつ、-20mm ~ 20mm の区間ずらした場合 (図 9(a))
- 下のブロックの短辺方向に、上のブロックを 1mm ずつ、-20mm ~ 20mm の区間ずらした場合 (図 9(b))

また、点灯する赤外線 LED の位置による影響も考慮し、赤外線 LED がブロック表面の頂点 (角) にある場合、辺にある場合、それ以外の内側の 3 種類で評価を行った。

実験の結果を図 10 に示す。なお、受光強度は A/D コンバータで出力された値を 0~1 で正規化してある。図より、フォトトランジスタの受光強度の分布は、長辺方向にずらした場合、短辺方向にずらした場合ともに、点灯する赤外線 LED がブロック表面の頂点にある場合、辺にある場合、それ以外の内側にある場合とでほとんど違いは見られなかった。また、長辺方向にずらした場合、短辺方向にずらした場合ともに、ずらした距離が -5mm ~ 5mm では受光強度が一樣に約 0.9 と非常に大きなものとなっているが、それ以外の区間では、ずらした距離が大きくなれば受光強度も小さくなるという特徴がみられた。

3.3 節で述べた形状認識においては、接触領域の推定のためにブロック表面の各フォトトランジスタの受光強度から、しきい値を用いて接触、非接触を判別し接触情報としている。このしきい値は図 10 内の赤い横線で示されるもので、経験上 0.48 としている。そのため、図 10 の赤網領域で示された、点灯する赤外線 LED から長辺方向には約 7mm 以内、短辺方向には約 5mm 以内に存在するフォトトランジスタのみが接触と認識される。

#### 4.1.2 積み重ね時の鉛直方向のブロック間距離に対するフォトトランジスタの受光強度の評価

StackBlock の積み重ね認識には赤外線を用いているため、ブロック間が鉛直方向に離れていても積み重なったとして形状認識が行われる可能性がある。

そこで、積み重ね時のブロック間距離に対するフォトトランジスタの受光強度を評価する。2つのブロックの大きい面同士を積み重ね、上のブロックの赤外線 LED を 1 個点灯させ、下のブロックのフォトトランジスタを 1 個受光強度認識させる。点灯する赤外線 LED と受光強度認識するフォトトランジスタをずれのないよう対面する位置を初期状態とし、ブロックの位置を 1mm 刻みで鉛直方向に離し、それぞれの位置でのフォトトランジスタの受光強度を調査する。

実験の結果を図 11 に示す。図中青線で示したように、ブロック間距離が大きくなればなるほど受光強度は小さくなった。このようにブロック間が離れていても赤外線を受光することが確認された。接触領域判定のための受光強度の

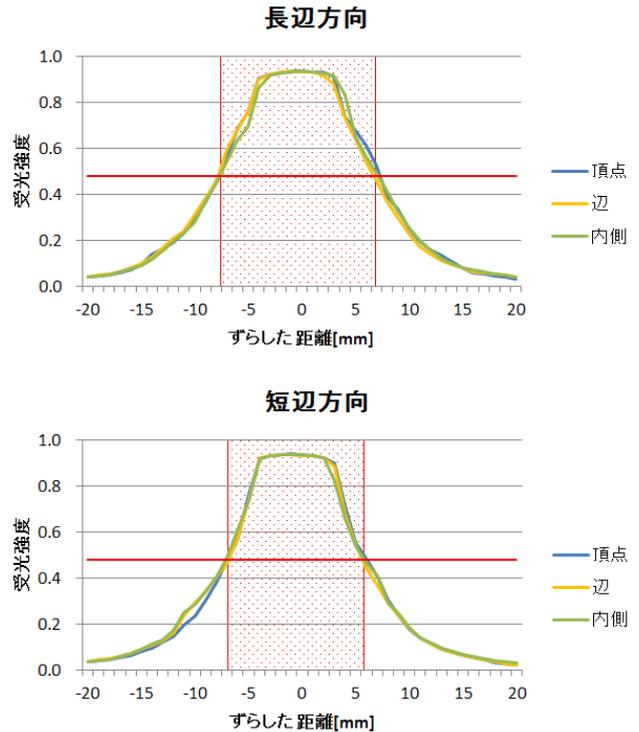


図 10 ブロックのずれに対する受光強度  
Figure 10 The Received Light Intensity for the Displacement of the Block.

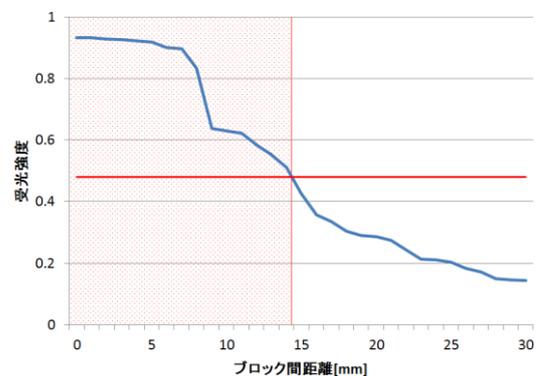


図 11 ブロック間距離と受光強度  
Figure 11 Received Light Intensity for the Distance

しきい値が図 11 中の赤横線で示される 0.48 (0~1 に正規化) であるため、図 11 中赤網領域で示されるブロック間が約 15mm までの場合に接触していると誤認識される可能性がある。

#### 4.2 積み重ね認識のリアルタイム性の評価

積み重ね認識のリアルタイム性の評価として、ブロック同士の接触がホスト PC に伝達され、形状認識が完了するまでの時間を測定する。ここでは、ユーザインタフェースとしての一般的なリアルタイム性を指標 [17] にして議論する。

実際の操作と同様に電源の入ったブロックを重ねると、4.1.2 節で示したようにブロックが離れた状態でも認識が行われるため、積み重ね認識に要する時間を正確に測定でき

ない。そこで今回の実験では予め BaseBlock が積み重ね認識を開始すると同時にブロック表面に設置された緑色 LED を発光し、Host PC に接触情報を送信し終わると同時に緑色 LED を消灯するようプログラムした。

積み重ね認識に要する時間の計測では、ブロックを積み重ね、BaseBlock の緑色 LED が点灯してから消灯するまでの時間を計測した。ブロック内のマイコンと Host PC の間でクロックの同期を取れないため、積み重ね認識に要する時間はオシロスコープ (Tektroix 製 MSO4054B) のプローブを緑色 LED 点灯用の端子に取り付けて測定する。積み重ねに用いるブロックの数は 1~3 個とし、測定はそれぞれに対して 10 回ずつ行う。

実験の結果、ブロックを 1 つ積み重ねた場合の積み重ね認識に要する時間は平均 403.5msec、2 つ積み重ねた場合には平均 881.5msec、3 つ積み重ねた場合には平均 1547.0msec となった。図 12 に 2 つのブロックを積み重ねた際の積み重ね認識に要する時間の内訳を示す。図より、

- ① 接触認識 (上にブロックがある場合) : 110msec
- ② 接触認識 (上にブロックがない場合) : 35msec
- ③ ブロック内通信 : データ量に比例
- ④ ブロック間での赤外線通信 : データ量に比例
- ⑤ BaseBlock から Host PC への接触情報送信 : データ量に比例

である。

また、積み重ねるブロックの数を  $n$  とした場合の積み重ね認識に要する時間  $T$  は以下の数式で示される。

$$T = 320n + 95 + 165 \sum_{k=1}^{n-1} k \quad (\text{msec})$$

図 12 に示したように、StackBlock では、積み重ね情報はブロック間の赤外線通信のリレーにより最終的に Host PC に送信されるため、積み重ねるブロックの数が大きくなれば、積み重ね情報の増大や赤外線通信の増加により積み重ね認識に要する時間は大きくなる。一般に、ユーザはタスクの種類に応じた適切な時間内にユーザインタフェースからレスポンスがあることを期待するが、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[18]。今回の実験結果では 2 個のブロックの積み重ね認識まではリアルタイム性を有しているといえるが、3 個以上のブロックにおいても更なる積み重ね認識に要する時間の削減が必要とされる。

## 5. 考察

### 5.1 性能評価実験について

4.1.1 節で述べたように、現在のしきい値を用いた実装では、接触情報が接触か非接触かという離散的なものであるため、図 13(a)に示すように実際の形状と認識される形状との誤差が大きくなる可能性がある。そこで、図 13 (b) に示

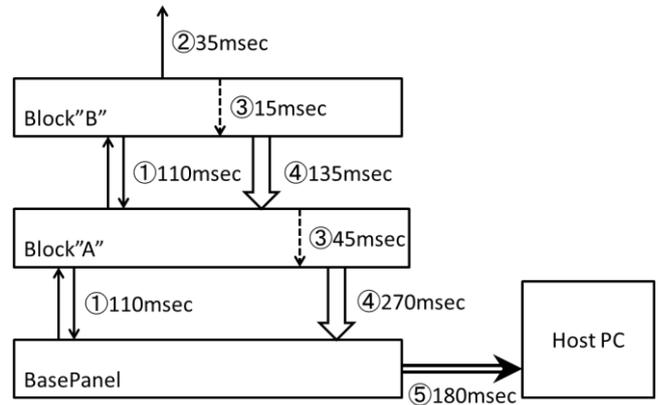


図 12 積み重ね認識に要する時間の内訳

Figure 12 Breakdown of the Time Required for Recognition.

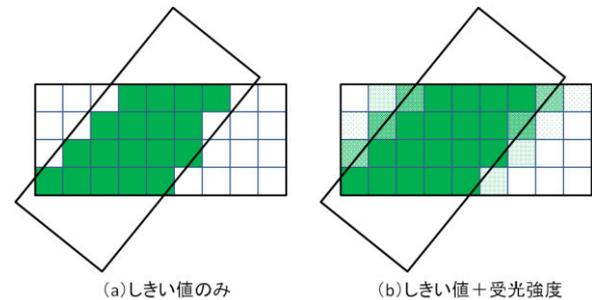


図 13 形状認識手法

Figure 13 Shape Recognition Method.

すように接触情報に受光強度データを含めることにより、現在の形状認識では用いられなかった、しきい値以下の多くのデータを用いて形状認識を行うことができる。これにより、実際の形状と認識される形状との誤差を、現在の実装よりもさらに小さくできると考えられる。

4.1.2 節で述べたように、StackBlock では赤外線を用いるため、ブロック間が離れていても赤外線を受光する。しかし、この特徴は、見方を変えればブロック同士が接続される直前の状態も認識できるため、StackBlock はユーザがブロックを積み重ねる前に構築形状を予測したり、ユーザがブロックをどこに置くか迷うといった動作を認識したりすることができると考えられる。

4.2 節で述べたように、StackBlock は積み重ね認識に要する時間をさらに削減する必要がある。そのための主な方法として、データ量の削減があげられる。データ量の削減が可能となれば、図 12 に示したように積み重ね認識に要する時間も大幅に削減可能である。今回の実装では、各ブロックの接触情報は、ブロック 1 個につき 200Byte と非常に大きなものとなっている。そこで、データの格納方法などを検討することにより接触情報のデータ量を大きく減らすことが可能である。

### 5.2 手との接触認識

StackBlock は本物の積み木をモデルとして設計しているため、ユーザは簡単に、直感的にブロックを積み上げるこ

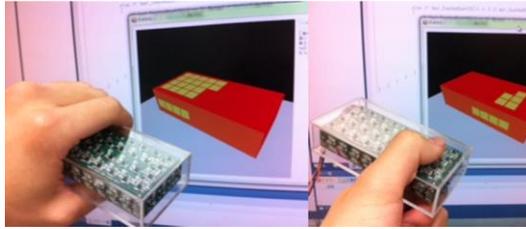


図 14 手との接触認識

Figure 14 Recognition of the Hand.

とで 3 次元形状を構築することができる。また、3 章で述べたように、各ブロックはブロックとの接触だけでなく、手との接触も検出しその接触領域を認識することができる。

(図 14) これにより、StackBlock は現在ユーザが握っているかどうか、ユーザがどのように握っているか、といった従来のブロック型 UI では認識することができなかった状態遷移も認識することが可能である。

### 5.3 応用

従来のブロック型 UI とは異なり、StackBlock はユーザの自由な積み重ねによる構築形状を正確に認識する。また、StackBlock は外観だけではなく、積み上げたり、崩したりというような積み木と同様に扱うことが可能である。この特徴から StackBlock は、教育ツールや医療ツールなど実際のブロックを用いた既存のアプリケーションを発展させる可能性を秘めている。例えば、積み木はプレイセラピーや教育ツールとして用いられてきたため、StackBlock はプレイセラピーや教育ツールの効果の解析や測定に応用することが可能である[19]。

また、StackBlock はユーザは簡単に、直感的にブロックを積み上げることで 3 次元形状を構築することができるため、言語、文化、世代などに依存しない応用が可能である。例えば、ブロックと手の接触を検出することにより、構築形状の認識だけでなく、形状構築プロセスにおけるユーザとブロックのインタラクションを認識することも可能である。そのため、ユーザのブロックとのインタラクションに対して、ブロックが特定の音や光を発したり、PC 上で再現される構築形状が変化したりといったフィードバックを与えることが可能である。

## 6. おわりに

本研究では、ユーザの自由な積み重ねによる構築形状を正確に認識できるブロック型 UI である StackBlock を提案し、プロトタイプを実装した。評価実験により、形状認識の精度、形状認識のリアルタイム性を確認した。今後は、より正確な認識手法やより短時間で認識可能なアルゴリズムの検討を進めていく。

**謝辞** StackBlock の実装にあたりご支援いただいた株式会社システムワットの菊池日出男氏に深い感謝の意を表す。

また、本研究は JSPS 科研費若手研究(A)24680013 の助成を受けたものである。

## 参考文献

- 1) H. Ishii, and B. Ullmer, "Tangible bits: towards seamless interfaces between people, bits and atoms," *Proc. of CHI '97*, pp.234-241, 1997.
- 2) G.W. Fitzmaurice, "Bricks: laying the foundations for graspable user interfaces," pp.442-449, *ACM Press*, 1995.
- 3) B. Ullmer, H. Ishii, and R.J.K. Jacob, "Token+constraint systems for tangible interaction with digital information," *ACM Trans. Comput. - Hum. Interact.*, vol.12, no.1, pp.81-118, March 2005.
- 4) S. Brave, H. Ishii, and A. Dahley, "Tangible interfaces for remote collaboration and communication," *Proceedings of the 1998 ACM conference on Computer supported cooperative work*, pp.169-178, New York, NY, USA, 1998, ACM.
- 5) H. Ishii, A. Mazalek, and J. Lee, "Bottles as a minimal interface to access digital information," *CHI '01 Extended Abstracts on Human Factors in Computing Systems*, pp.187-188, New York, NY, USA, 2001, ACM.
- 6) D. Anderson, J.L. Frankel, J. Marks, D. Leigh, E. Sullivan, J. Yedidia, and K. Ryall, "Building virtual structures with physical blocks," *Proc. of UIST '99*, pp.71-72, 1999.
- 7) D. Anderson, J.L. Frankel, J. Marks, A. Agarwala, P. Beardsley, J. Hodgins, D. Leigh, K. Ryall, E. Sullivan, and J.S. Yedidia, "Tangible interaction + graphical interpretation: a new approach to 3d modeling," *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp.393-402, New York, NY, USA, 2000, ACM Press/Addison-Wesley Publishing Co.
- 8) R. Watanabe, Y. Itoh, M. Asai, Y. Kitamura, F. Kishino, and H. Kikuchi, "The soul of activecube - implementing a flexible, multimodal, three dimensional spatial tangible," *Proc. of ACE '04*, pp.178-180, 2004.
- 9) Y. Itoh, S. Akinobu, H. Ichida, R. Watanabe, Y. Kitamura, and F. Kishino, "Tsu.mi.ki: stimulating children's creativity and imagination with interactive blocks," *Proceedings of International Conference on Creating, Connecting and Collaborating through Computing (C5 '04)*, pp.62-70, 2004.
- 10) 李鎮河, 箕康明, 苗村健, "Bloxels:発光ブロック群による実体ボクセルディスプレイ : 積層によるブロック間の情報伝達(人工現実感,及び一般)," 電子情報通信学会技術研究報告. MVE, マルチメディア・仮想環境基礎, Vol.109, No.75, pp.31-36, 2009.
- 11) A. Miller et al., "Interactive 3D model acquisition and tracking of building block," *IEEE Transactions on Visualization and Computer Graphics*, Vol.18, No.4, pp.651-659, April 2012.
- 12) A. Gupta, D. Fox, B. Curless, and M. Cohen, "DuploTrack: a realtime system for authoring and guiding duplo block assembly," *Proc. of UIST '12*, pp.389-402, 2012.
- 13) Reddy, L. A. et al., "Empirically based play interventions for children," American Psychological Conference, 2005.
- 14) J.Y. Han, "Low-cost multi-touch sensing through frustrated total internal reflection," *Proceedings of UIST '05*, pp.115-118, New York, NY, USA, 2005, ACM.
- 15) F. Echtler, A. Dippon, M. T'onnis, and G. Klinker, "Inverted fir: easy multitouch sensing for flatscreens," *Proceedings of ITS '09*, pp.29-32, New York, NY, USA, 2009, ACM.
- 16) N. Matsushita, and J. Rekimoto, "Holowall: designing a finger, hand, body, and object sensitive wall," *Proceedings of UIST '97*, pp.209-210, New York, NY, USA, 1997, ACM.
- 17) H. Benko, A.D. Wilson, and R. Balakrishnan, "Sphere: multi-touch interactions on a spherical display," *Proceedings of UIST '08*, pp.77-86, 2008.
- 18) T.W. Butler, "Computer response time and user performance.," *Proceedings of the CHI '83*, pp.58-62, New York, NY, USA, 1983, ACM.
- 19) H. G. Kaduson and C.E. Schefer, 101 more favorite play therapy techniques. Aronson: Maryland, 2001.