

スマートサーフェスにおける木構造知識を用いた分散制御手法

SATO, Atsushi / 佐藤, 温

(出版者 / Publisher)

法政大学大学院情報科学研究科

(雑誌名 / Journal or Publication Title)

法政大学大学院紀要. 情報科学研究科編

(巻 / Volume)

8

(開始ページ / Start Page)

167

(終了ページ / End Page)

172

(発行年 / Year)

2013-03

(URL)

<https://doi.org/10.15002/00009860>

スマートサーフェスにおける木構造知識を用いた分散制御手法 Distributed Control Approach with Tree-structured Knowledge in the Smart Surface

佐藤 温

Atsushi Sato

法政大学大学院情報科学研究科情報科学専攻

E-mail: atsushi.sato.9q@stu.hosei.ac.jp

Abstract

This research proposes a distributed control approach with tree-structured knowledge for performing the part recognition in controlling the smart surface. In this approach, a tree-structured array is used to store knowledge about models of the objects at the offline stage and to accumulate and share knowledge among neighboring active cells about shapes of objects which must be reconstructed and differentiated on the smart surface at the online stage. The approach is analyzed and compared with the previous matrix-based approach, which contained redundant information in each cell and communication, and demanded excessively frequent comparison in the shape differentiation. Though the current approach shows better performance in terms of a smaller memory footprint and lower communication cost, it trades off the reduction of memory footprint against the probability of the differentiating. Moreover, a knowledge distribution technique for cutting down memory footprint in each cell is addressed and the relation between a distribution ratio and its efficiency of reducing the memory footprint is analyzed.

1. 序論

大量生産が行われている機械製品の組立工程では、それぞれ製造された部品を適切な作業場に運搬する必要がある。しかし多量の微小な部品を生産ラインで扱う際にはそれらが混在してしまい、適切な場に届かない部品が現れる。このような問題に対して、微小な部品を効率良く運搬するための分散操作デバイスが近年研究されている。分散操作 (distributed manipulation) とはある物体に対する一つの操作を複数のアクチュエータによって多点に同時に力を加えることで実現する手法を指す。構成する個々のアクチュエータは安価で単純なものを用いるが、それらが協力することで大きく重い物体に対する制御を可能とすることが第一の利点である。また分散操作による平面状のコンベアデバイスは 1990 年代から盛んに研究がされており、ローラーホイールによる操作[1]や電磁石による操作[2]、空気圧による操作[3]などの研究がされてきた。

スマートサーフェス[4]はセルと呼ばれるアクチュエー

タにセンサと知能を加えた構成要素がデバイス上に置かれた物体を感知し、種類を識別し、その物体の種類によってそれぞれ適切な操作を分散制御によって実行するデバイスである。スマートサーフェスのソフト面における研究として物体の種類別の分散制御における各セルが必要とされるメモリ量の削減やデバイスの拡張に対する柔軟性などの改善が必要とされている。

本論文ではスマートサーフェスの分散制御における物体の識別に木構造知識を用いることで従来の制御手法より拡張性と柔軟性が高まり、また構成するセルが必要とされるメモリ量が減少することを示し、さらに各々のセルに保存していた知識を分散させることによる必要とされるメモリ量のさらなる削減を検討する。

2. スマートサーフェス

スマートサーフェスとは MEMS (Micro-Electro-Mechanical Systems) にセンサやアクチュエータ、計算装置を集積し、これを格子状に複数敷き詰める (図 1) ことで物体の感知、識別、操作を実現する知的小型コンベアデバイスである。スマートサーフェスを構成する MEMS はセルと呼ばれ、各セル自身で個別に制御する。また各セルは自身と隣り合う 4 つのセルと P2P 通信によって互いの持つ情報を交換する。

2.1 機能

スマートサーフェスは混在した部品群に対して各部品を適切な作業場に運搬するため、物体の識別、運搬、配置の 3 つの機能を実現する必要がある。各セルは自身の上に物体があるかを感知し、状態から行動を判断し、局所的に物体に力を加える能力によりこれらを実現する。

1) 物体の識別

スマートサーフェス上に置かれた物体に適した処理をするために、予め決められたいくつかのモデルの内、その物体がどのモデルなのかを識別する。各セルはセンサによって自身の上の物体の有無を判断出来るため、その情報を各セルから収集することで物体の形を判断し、その形を元に物体がどのモデルなのかを識別する。

2) 物体の運搬

スマートサーフェスにおけるアクチュエータは弁の開閉による空気ジェットである。各セルは東西南北の 4 方向に向いた穴を持ち、下部から常に空気が送られている。セルは 4 つの弁を開閉することで物体を 4 方向に押す事が出来る。スマートサーフェスは各セルの弁の開閉を組み合わせる事により物体の運搬を実現している。

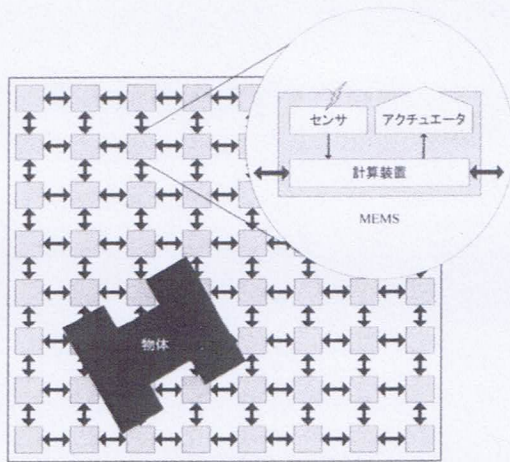


図1 スマートサーフェスの構成.

3) 物体の配置

任意の位置における物体の固定や物体の回転, 複雑な軌道による物体の運搬等が出来るよう, スマートサーフェス上に置かれた物体の位置を制御する.

2.2 分散制御構造

本研究ではスマートサーフェスの制御構造は Boutoustous らの提案[5]に従う. 図2に示すようにスマートサーフェスを構成する各セルは分散制御により感知, 判断, 操作を行う. 自身のセンサの値と P2P 通信により得た周囲のセルのセンサの値により物体の形状復元と識別を行う. 識別の結果としてその物体がどの部品なのかを座標と共に動作制御部に渡し, その値をもとにアクチュエータである弁の開閉を操作する.

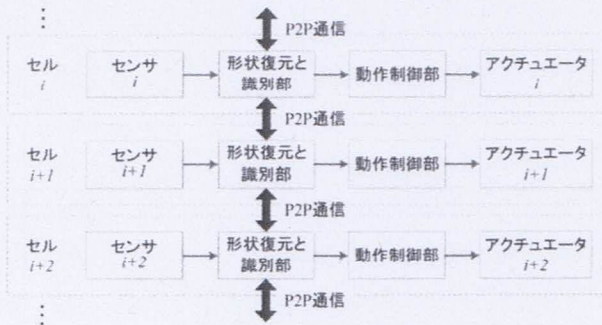


図2 分散制御構造.

3. Boutoustous らによる識別手法[5]

部品の識別は図2の“形状復元と識別部”で実行される. 形状の復元と識別部は2つの局面で実現される. 識別には物体の形を示す行列から計算した識別基準値[6]を用いる.

1) オフラインステージ

スマートサーフェスを動かす以前に実行する準備段階であり, 回転, 平行移動操作を組み合わせることで各部品モデルの形状データを出来る限り多く取得し, センサによって離散化することで行列を生成し, マスク化によってモ

デルの形情報と関係のない情報を削ると同時にいくつかの識別基準値をモデルに合わせて決定する. その基準値を計算して各部品の形状データベースを構築する. これはスマートサーフェスを構成するセル全てに組み込まれ, オンラインステージで用いられる. データベースには各部品の形一つ一つを表した行列と, その形を特徴付けるいくつかの基準値全てが保存される.

2) オンラインステージ

実際にスマートサーフェス上で物体を識別する. スマートサーフェス上に物体が置かれると, 物体の下に位置するセルはセンサによって物体を感知し, 隣接した4つのセルからの情報と合わせて物体の形状を復元する. 物体の形状は行列で表現し, この行列を元に基準値を計算し, データベース内の基準値と比較することで識別をする.

この手法はモデルの組み合わせによって識別に用いる基準値を事前に決める必要があるが, それによって精度の高い識別が実現している.

4. 木構造知識を用いた分散制御手法

Boutoustous らの手法においてはスマートサーフェス全体の状態を表す行列によって物体の形状を表現したため, サイズが大きくなるにつれてセル間の通信メッセージのサイズが大きくなる, 局所的な操作が出来ないといった問題がある. また, セルとなる MEMS が将来的により小さいサイズ (NEMS 等) になることを考慮した上で, 各セルに保存されるモデルの形状データベースが必要とするメモリ量の削減が必要とされている.

4.1 知識表現

知識とは物事の関係性や性質を示す構造化されたデータである. 知識の表現には様々な構造が考えられるが, 知識を利用する知的行動によって適切なものを選択する.

4.2 木構造知識[7]

木構造知識とは木の構造で表現された知識であり, 木とは閉路を持たず, 根と呼ばれる特別な頂点を1つ持ち, その他の全ての頂点からは根へ唯一通りに辿ることが出来るグラフである. 木はノードとリンクという2つの要素からなり, ノードはデータを表し, リnkは親子関係を持つ2つのノード間をつなぐことでその関係を表現する.

木構造知識の実装には様々な方法があり, システムが処理しやすい方法を選択する. 本研究では, 子ノードが4つと固定された四分木を1次元の木構造配列として扱う.

4.3 木構造知識を用いた部品の識別手法

木構造知識を用いた部品の識別は Boutoustous らの手法をベースとして, 図3に示すように, 行列や基準値の代わりに木構造知識を用いている. オフラインステージでは各部品モデルの形状データを木構造配列として回転や平行移動により出来る限り多く取得し, データベースを構築することでオンラインステージの比較に用いる. オンラインステージでは形状復元工程と識別工程があり, 形状復元工程で物体の形状を木構造配列として復元し, 識別工程ではデータベース内のモデルの形状を表す木構造配列と復元した形状を比較するため, 固有な配列へ木構造配列を変形する. 以下ではオンラインステージにおける2つの工程について説明する.

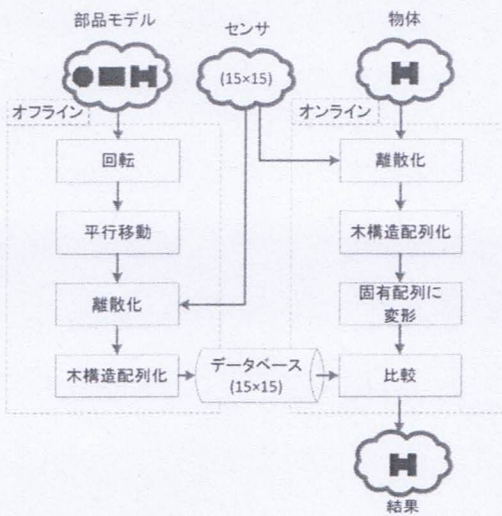


図3 木構造知識を用いた部品の識別手法の概要.

1) 形状復元工程

根がそのセル自身の状態であり、4つの隣接するセルの状態を4つの子ノードとする四分木で物体の形状を表現する。1次元の木構造配列による実装においては各ノードの値は親ノードの直ぐ右側に順に北、西、南、東の隣接したセルの値を子ノードとして持ち、それぞれの子ノードも同様に直ぐ右側に4つの子ノードを持つ。

通信段階：アクティブセルは通信段階で送信先の枝を切り落とすことで三枝の部分木をメッセージとして隣接する4つのセルにそれぞれ生成し(図4参照)、送る。また隣接する各アクティブセルから送られてきた最大4つのメッセージを自身の木構造配列に合成して視界を更新する。

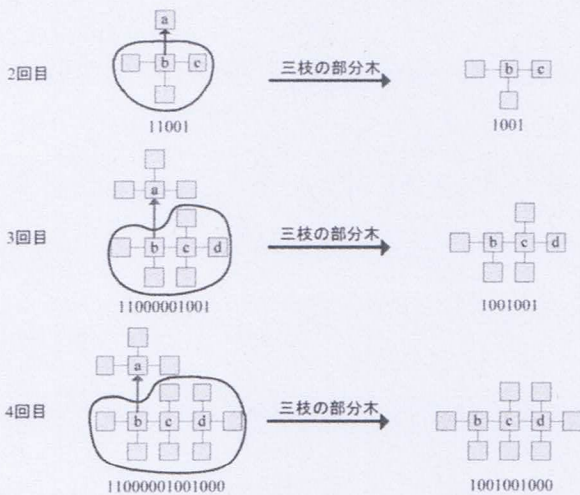


図4 メッセージとなる部分木の生成.

その結果、各アクティブセルは物体の全体を含む形状を表す木構造配列を獲得する。

形状復元においてセルは格子状に並んでいるため、循環構造が出来る。これを避けるため、1つの木に同じアク

ティブセルの状態が現れた場合、一方を削除する。各アクティブセルにおいてメッセージを受け取り、自身の木構造配列と合成した際に重複チェックを行う。配列中の値“1”にそれぞれ相対的な座標をIDとして与えることで検出する。重複したアクティブセルを値“0”と置き換えて削除することで重複のない新たな配列を獲得する。形状復元工程は以下のようにまとめられる。

- 1: 木構造配列の初期化
- 2: repeat
- 3: 4つの各隣接セルにメッセージを作成する
- 4: 各隣接セルにメッセージを送信する
- 5: 隣接するアクティブセルからメッセージを受け取る
- 6: 根の4つの子ノードを受け取ったメッセージと置き換える
- 7: 更新された木構造配列を重複チェックする
- 8: 重複した要素を“0”と置き換える
- 9: until 受け取った配列の葉ノードが全て“0”もしくは更新されない
- 10: [識別工程へ進む]

図5に形状復元の一例を示す。図中の四角形はそれぞれスマートサーフェスを構成するセルを示し、各セルは4つの方向にそれぞれ隣接するセルを持つ。これを図中右上の木で示すように“N”, “W”, “S”, “E”と方向に対応して固定された順番で保持する。根を除き、各ノードは1つの親ノードと3つの子ノード(親ノードに対応する子ノードは省かれる)を持つ。四角形が描かれていないリンクは親ノードと対応する子ノードの位置を示している。

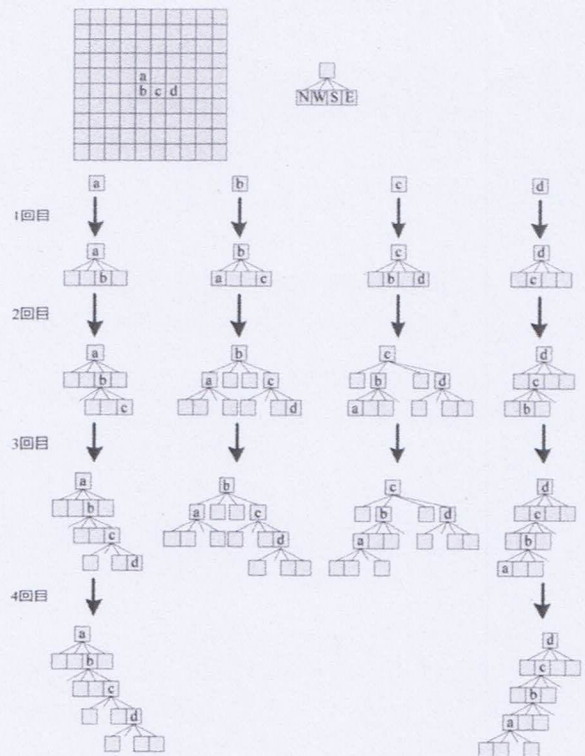


図5 形状復元工程.

2) 識別工程

物体の識別にはオフラインステージで生成する各モデルの形状データベースが必要である。第3章で述べた従来手法では行列から計算した基準値が比較されたが、提案手法では各形状に対応した木構造配列をデータベースに格納し、直接形状で比較する。ただし、同じ形状でも複数の表現が可能のため、最も北かつその中で最も西に位置するセルの状態が根となる固有の配列として比較する。

しかし、オンラインステージで形状復元工程により得られる木構造配列の根は必ずしもこの条件に当てはまらないので、比較を行う前に固有配列に変形する。変形は現在の根の北にアクティブセルがあればそのセルが根となるように、最も北のセルが根の時、その根の西にアクティブセルがあればそのセルが根となるように変形をする。まず、配列の値を左から順に見ていく。左から1ビット目は常に根であり、2ビット目は根の北のセルの状態を示すため、2ビット目の値が“1”であれば根が一つ北のセルとなるように変形する。図6は以下の手順で根bの木構造配列を根aの木構造配列に変形している。

- (1) 値“1”をセルaを根とする部分木を除く部分配列(001001000)に加え、
- (2) その部分配列(1001001000)を矢印で示されている4ビット目と5ビット目の間に挿入する。
- (3) 元の根である1ビット目の値“1”を削除して変形後の木構造配列である10010010010000を取得する。

もし2ビット目が“0”であれば、次に3ビット目を見る。3ビット目が“1”であれば根の西にアクティブセルがあるので、同様に根が西のセルとなるように変形する。

以上のチェックと変形を、2, 3ビット目が共に値“0”となり、固有配列になるまで繰り返す。変形後、データベース内の木構造配列と一つずつ比較していき、一致する配列があればその配列が示すモデルと識別される。識別結果のモデルとその物体の位置は図2の動作制御部に送られる。動作制御部はモデルによって運搬先を判断し、それぞれのアクチュエータを制御する。識別工程は以下のようにまとめられる。

- 1: {形状復元工程が完了}
- 2: repeat
- 3: if 根の北のセルがアクティブ then
- 4: 根が北のセルになるよう配列を変形
- 5: else if 根の西のセルがアクティブ then
- 6: 根が西のセルになるよう配列を変形
- 7: end if
- 8: until 根の北と西にアクティブセルが存在しない
- 9: if 根が最も北かつその中で最も西 then
- 10: repeat
- 11: データベース内の配列と比較
- 12: if 一致する配列が見つかる then
- 13: 動作制御部に識別結果を送信
- 14: end if
- 15: until 一致する配列が見つかるもしくはデータベース内の全ての配列と比較した
- 16: end if
- 17: 物体が移動し、形状復元工程が始まるのを待つ

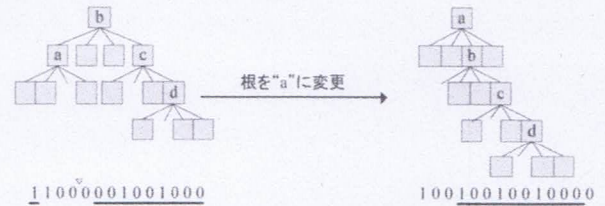


図6 正規化における木の変形。

5. 木構造知識を用いた識別手法の評価

P2P 通信回数、通信トラフィック量、処理時間、セルのメモリ量の4つの観点から提案手法を従来手法[5]と比較して評価する。処理時間とセルのメモリ量に関しては第3章で紹介した従来手法と同じ条件の下で比較する。

5.1 P2P 通信回数

形状復元工程における各セルの通信回数は物体に対するセルの位置に依存しており、外側であれば外側であるほど物体全体の視界を獲得するまでに多くの通信回数が必要となる。通信の最大回数は物体の覆っているセルの高さと幅の合計数より1小さい数、最小回数は高さと幅の合計数の半分より1大きい数となる。例えば図7において各セルが物体全体の視界を得るには、セル“a”は8回、セル“b”は5回と差が現れる。この観点からは従来手法と同等と言える。

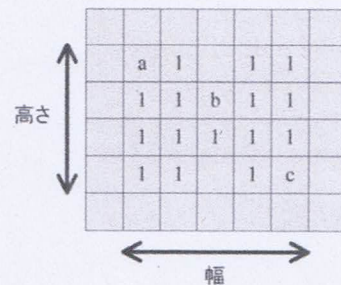


図7 H型の部品が置かれた際のセルの状態の例。

5.2 通信トラフィック量

提案手法においては通信におけるメッセージとなる木構造配列は可変サイズであり、通信を繰り返して視界を広げる毎に隣接するセルと交換するメッセージも大きくなる。例えば図5におけるトラフィック量は1ビット、4ビット、7ビット、10ビットと増加していく(図4参照)。このサイズのメッセージを4つのアクティブセルがそれぞれ一度に4方向に送信すると考えると $(1+4+7+10) \times 4 \times 4 = 352$ ビットと概算出来る。

従来手法においてはメッセージとなる行列は常にスマートサーフェスのサイズと同等であるため、図5の 10×10 のスマートサーフェスにおけるメッセージサイズは100ビットである。従って4つのアクティブセルがそれぞれ一度に4方向に送信し、それぞれ提案手法における計算と同様に4回の通信で物体の形状復元が出来るとすると、

$100 \times 4 \times 4 \times 4 = 6400$ ビットと計算出来る。これは提案手法のおよそ18倍である。

5.3 処理時間

提案手法において処理時間は大きく3つの要素：形状復元時間 (TR)、木構造配列の変形時間 (TT)、そして比較時間 (TC) に分けられる。これを元に図7において最も時間がかかる場合を考えると、セル“c”は固有配列の根であるセル“a”から最も遠く、通信回数と変形回数も8回、7回となっている。比較回数は多くともデータベース内の形状データの数である。

処理時間の比較に比較時間も参考するために、次の条件における形状データの個数を用いる。600×600ピクセルのスマートサーフェスを15×15のMEMSによって構成しているとし、各セルとMEMS、また識別に用いた3つのモデルの種類とサイズを図8に示す。比較のために従来手法で使う識別の基準値は以下の3つとする。

- A：直角を形作っている角の数
- P：0に面しているアクティブセルの数
- S：セルを覆っている物体の面積

これらの条件は Boutoustousら[5]の研究を参考にした。

これらの条件の下、オフラインステージのシミュレーションにより、形状データがそれぞれ円型48個、長方形型248個、H型428個というデータが得られた。従ってこの合計724個の形状データが各セルのデータベースに格納されると仮定して比較する。また最悪の場合として全ての形状データと比較した場合を考える。処理時間 T は：

$$T = 8TR + 7TT + 724TC \quad (1)$$

となる。

次に従来手法で図7のセル“c”が識別するまでにかかる処理時間を求める。通信回数は8回、比較回数は提案手法の時と同様とする。また、シミュレーションによって得た724個の形状データから基準値を計算した結果58個の基準値が得られた。すると処理時間 T' は：

$$\begin{aligned} T' &= 8(TR' + TA + 58TC') \\ &= 8TR' + 8TA + 464TC' \end{aligned} \quad (2)$$

となる。ただし、 TR' 、 TA 、 TC' はそれぞれ形状復元時間、基準値の計算時間、そして比較時間を指す。

T と T' の比較により、従来手法の方が比較回数が少なく済むという点以外ではあまり差が見られなかった。これは提案手法が形状データそのものを比較に使うのに対して、従来手法は事前に識別しやすい基準値を計算し、比較に利用するためである。

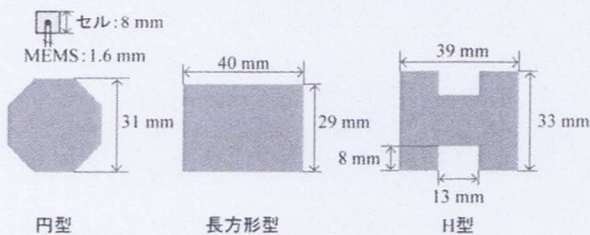


図8 セルと部品モデルのサイズ。

5.4 メモリ量

物体の識別に用いるモデルの形状データベースが必要とするメモリ量について比較する。

第5.3節と同様の条件の下、従来手法においては一つ当たりの形状データのメモリ量は29バイト($\cong 15 \times 15/8$)。従って724個の形状データが占めるメモリ量は 29×724 バイト。また基準値一つ当たりのメモリ量を4バイトとすると58個の基準値が占めるメモリ量は 4×58 バイト。これらを合計して21228バイト ($\cong 21$ キロバイト)が必要とされる。

次に提案手法において、一つ当たりの形状データ S_i が必要とするメモリ量 MS_i は、

$$MS_i = \begin{cases} 4 & \text{if } n = 1, \\ 3n + 2 & \text{if } n \geq 2. \end{cases} \quad (3)$$

ただし、 n はアクティブセルの数。

さらに N_{Model} をモデルの数、 N_{Shape} を形状データの数とすると一つ当たりのモデル M_k の必要とするメモリ量は、

$$MM_k = \sum_{S_i \in M_k} MS_i, \quad i = 1, 2, \dots, N_{Shape} \quad (4)$$

と計算出来る。従って全てのモデルが必要とするメモリ量は次のように求められる。

$$M = \sum_{i=1}^{N_{Model}} MM_i \quad (5)$$

ここで例として、約430個の形状データを持つH型のモデルが25個未満のアクティブセルを覆うサイズであるとき、数式(5)より一つ当たりの形状データの占めるメモリ量は77ビット ($\cong 10$ バイト)と計算される。

従ってH型モデルが必要とするメモリ量は4.3キロバイトと求められる。同様にして円型、長方形型のモデルはそれぞれ0.5キロバイト、2.5キロバイトのメモリを必要とする。合計7.3キロバイトという数は提案手法が従来手法と比較して3分の1程度までメモリ量を削減出来たことを示している。

さらに、全ての形状データが均等に価値を持っているわけではないため、識別に有効な形状や間違った識別結果を導く形状等その価値にはバラつきがある。識別に有効な形状データを選定することで効率の良い形状データのコレクションを生成し、利用することでメモリ量を減らしつつ識別成功率の低下を抑えられると考えられる。

6. 知識分散

セルが必要とされるメモリ量の削減を目指す上でモデルの形状データベースの分散という方法が考えられる。従来の識別手法は全てのセルが同様のデータベースを持ち、同様の振る舞いをしたが、識別に用いる形状データという知識を複数に分けて利用することで個々のセルが必要とされるメモリ量を減らす事が出来る。この章では第5.3節の条件の下、知識の分散方法とそれにより知識をいくつに分散させるべきかを検討する。

知識を分散させる方法として、モデル毎に分散させる方法と全てのモデルの形状データを複数のセルに均等に分散させる方法が考えられる。

モデル毎に分散させる方法では、円型、長方形型、H型の3つに分散させることになる。この場合各セルはいずれか1つのモデルの形状データを全て持つこととなるため、メモリ量は、第5.4節の通り各セルは724個の形状データを持っていたが、分散させることで円型、長方形型、H型の形状データは、48個、248個、428個となり、それぞれ約15分の1、3分の1、17分の10になる。この方法の問題は各モデルの持つ形状データの数の偏りが各セルが必要とされるメモリ量に影響し、最も多い形状データを持つモデルを基準にメモリ量を決定することである。

全てのモデルの形状データを複数のセルに均等に分散させる方法では、 n 個に分散させることでセル一つ当たりのメモリ量を n 分の1に減少させることが出来る。しかし、分散させることによって二つの問題が生じる。一つはスケールの問題である。スマートサーフェス上に置かれた物体を全ての形状データと比較するためには、データベースを n 個に分散させた場合、物体は n 種類全てのセルの上に存在しなければならない。従って、分散させる数を適切にしなければ識別は失敗しやすくなる。例えば図9(a)に示すように、データベースを“A”, “B”, “C”, “D”, “E”の5種類のデータベースに分け、セル毎にそのいずれかのデータベースを持たせた場合、モデルは全ての形状データと比較が出来る。またもう一つの問題として識別結果の矛盾が考えられる。つまり各セルの振る舞いはデータベースの種類により異なるため、セル毎の識別結果が異なってしまう。この問題に対して形状復元工程のように隣接するセル間で互いの識別結果を共有し、それら全てを考慮した上で最終的な結論を出すことで解決が図れる。この時、全てのデータベースのセルの識別結果を集める上で、P2P通信回数 N_c に基づいて効率を考えると、 $N_c = 1$ のとき、隣接する4つのセルと自身の識別結果を集められるため、5つに分散すると効率が良い(図9(a)参照)。 $N_c = 2$ のとき、同様に隣接する4つのセルと、それらに隣接する4つのセルと自身の識別結果を集められるため、13つに分散すると効率が良い。しかし、図9(b)を見てわかるようにこのスケールでは13つに分けられたデータベース全てを用いた識別結果を得るのは難しい。

以上より第5.3節の条件の下、知識を分散させ、セルが必要とされるメモリ量を減らす場合、5つが最適であり、その時のメモリ量は従来手法であれば第5.4節より約4.2キロバイト、提案手法であれば約1.5キロバイトとなる。

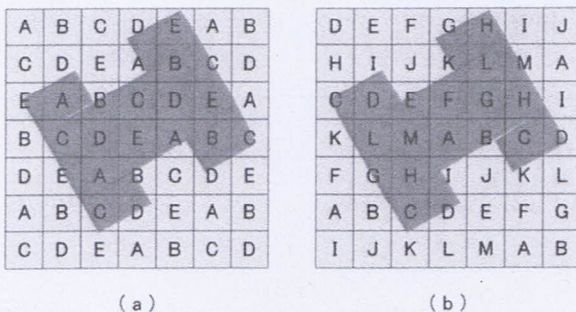


図9 分散された知識を持つスマートサーフェスの例。

7. 結論

本研究ではスマートサーフェスの分散制御手法に木構造知識を用いることで改善を目指した。モデルの形状を木構造で表すことでスマートサーフェスを構成するセルが必要とされるメモリ量を削減しつつ、デバイスのサイズに依存せず、局所的な操作が可能な拡張性を持った制御手法を提案した。その結果木構造知識を用いることで通信トラフィック量とセルが必要とされるメモリ量を減らせることが分かった。しかし減ったメモリ量でさえ実際にMEMSを用いて実装するには大きすぎるため、さらなる工夫が必要である。その方法の一つとして物体の識別に用いる形状データの選定が考えられる。各形状データと一致して導かれた識別結果が有効であるものに高い評価値を与え、複数のモデルに共通して複数のモデルを識別結果としてしまうものや試行の中で利用されないものには低い評価値を与えるとといった機械学習が考えられる。またもう一つの方法として、知識を分散させて扱うことによって一つ一つのセルに求めるメモリ量を減らしつつ利用する知識量は減らさないという方法について検討し、その結果適する分散数は部品とセルのスケールによって変化し、分析を行った条件においては知識を5つに分散させることで効率よく一つ当たりのセルのメモリ量が減らせるという結論に至った。しかし、その場合の各セルにおける識別結果の共有、統合といった具体的な物体の識別手法の提案と分析が今後の課題と言える。

参考文献

- [1] J. E. Luntz and W. Messner, "A distributed control system for flexible materials handling", IEEE Control Systems, vol. 17, no. 1, pp. 22-28, February 1997.
- [2] C. Liu, T. Tsao, Y.-C. Tai, W. Liu, P. Will, and C.-M. Ho, "A micromachined permalloy magnetic actuator array for micro robotics assembly systems", In Int. Conf. on Solid-State Sensors and Actuators Dig. Tech. Papers, Transducers '95, vol. 1, pp. 328-331, 1995.
- [3] S. Konishi and H. Fujita, "A conveyance system using air flow based on the concept of distributed micro motion systems", IEEE Journal of Microelectromechanical Systems, vol. 3, no. 2, pp. 54-58, June 1994.
- [4] Smart Surface Project, <http://www.smartsurface.cnrs.fr/index.html>
- [5] K. Boutoustous, G. J. Laurent, E. Dedu, L. Maignon, J. Bourgeois, and N. Le Fort-Piat, "Distributed control architecture for smart surfaces", In Int. Conf. on Intelligent Robots and Systems (IROS), pp. 2018-2024, Taipei, Taiwan, October 2010.
- [6] K. Boutoustous, E. Dedu, and J. Bourgeois, "An exhaustive comparison framework for distributed shape differentiation in a MEMS sensor actuator array", In Int. Symposium on Parallel and Distributed Computing (ISPDC), pp. 429-433, Krakow, Poland, July 2008.
- [7] R. Huang, J. Ma, and Q. Jin, "A tree-structured intelligence entity pool and its sharing among ubiquitous objects", In Proc. Int. Conf. on Computational Science and Engineering (CSE), vol. 2, pp. 318-325, Vancouver, Canada, August 2009.