

Tag and Think: センサネットワークを前提としたモノ自身とその状態の推定

前川 卓也† 柳沢 豊† 岡留 剛†

あらかじめモノに関する情報は一切与えずに、モノにセンサノードを添付するだけで、添付されたモノが何であるか、そしてそのモノが現在どのような状態にあるかを推定するためのフレームワーク Tag and Think を提案する。本研究では、モノのタイプごとに固有の有限状態オートマトンが定義されると仮定し、あらかじめ室内の様々なモノ（のタイプ）に固有の状態遷移図をあらかじめ用意しておく。そして、センサノードがある程度の期間内に収集したデータと状態遷移図を比較し、そのデータが観測され得る尤もらしい状態遷移図を決定することで、上記を実現する。ここでモノ固有の有限状態オートマトンとは、モノの状態とモノに起こり得る状態間の遷移の関係をグラフとして記述したものであり、例えばイスにはイスの状態遷移図を、ドアにはドアの状態遷移図を定義する。本研究の成果により、エンドユーザが家庭内にセンサノードを貼り付けるだけで容易にユビキタス環境の構築を構築できるようになる。

キーワード センサノード、センサネットワーク、モノの推定

Tag and Think: Estimating type and state of object in sensor network

TAKUYA MAEKAWA,[†] YUTAKA YANAGISAWA[†]
and TAKESHI OKADOME[†]

Tag and Think proposed here is a framework of a system in which, by simply attaching sensor nodes to physical objects with no information about the objects, the sensor nodes infers the type of the physical indoor objects and the states they are in. In the framework, assuming that an object has its own states that have transitions represented by a finite-state automaton, we prepare the state transition diagrams in advance for such indoor objects as a door, a drawer, a chair, and a locker. The system enables us to easily introduce pervasive environments by simply attaching sensor nodes to physical objects around us.

Keywords Sensor node, sensor network, inferring type of object

1. ま え が き

近年、家庭屋内環境においてセンサノードをモノに添付し、モノのタイプごとにモノの状態や状態変化に応じた状況依存サービスを提供するための研究が多く行われている。ドアというタイプのモノが深夜に長時間開け放たれていたなら、ユーザに通知を行うなどがそのサービスの一例である。しかし、数百のセンサノードが家庭内のモノに添付された状況では、それぞれのノードにそのノードが添付されているモノのタイプをマニュアルで与えることは非常に困難である。無線技術、センシング技術、省電力化技術の進展により、小

型で安価なセンサノードをばら撒くように使える時代がまもなくやってくる。われわれは、そのような時代を見越して、モノに汎用的なセンサノードを添付するだけで、そのノードがどのタイプのモノに添付されているか、そのモノがどのような状態にあるかを自動的に推定するフレームワークである *Tag and Think* (TnT) を提案する。

TnT では、センサノードが添付された環境において一定期間センサデータを収集したあと、あらかじめ用意したモノのタイプごとのモデルデータから、収集したセンサデータを最も出力しうるモデルデータを決定することでモノのタイプの推定を行う。さらに、モデルデータを用いてモノの状態や状態変化も同時に推定する。ここで、本研究における汎用的なセンサノ

† NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories

ドは、加速度センサや温度センサなどの一般的なセンサを含み、全て同一の機能を持つものであるとする。本論文では、特に教師なし学習により屋内にあるモノのタイプとその状態を推定する方法を紹介し、その評価を行う。屋内にあるモノには、教師つき学習による推定が適しているモノと教師なし学習による推定が適しているモノがある。前者は、そのモノを特徴付ける現象が主に繰り返し動作であり、それに対応するセンサデータの特徴が全世界のどのような環境（家庭）でも共通しているモノである。歯ブラシなどがその典型的な例である。ユーザが歯ブラシを使って歯を磨くときに得られる加速度の周波数成分はほとんどの世界で共通するからである。後者は、そのモノを特徴付ける現象が繰り返しでない動作であり、それに対応するセンサデータがセンサノードが添付されているモノが存在する環境によって全く異なるモノである。例えば、ドアが開けられるとき、ドアに添付されているセンサから得られる照度変化の波形は周囲の照明によって異なるし、その加速度の波形もノードが添付される向き、ドアを開ける人、ドアの大きさによって異なる。本報告では、後者に属するモノのタイプとその状態・状態変化の推定を、エンドユーザの知識によって作成された簡単なモデルを用いて行う。

2. 背景と目的

近年、安価になってきたセンサを家庭にある大量のモノに添付することで、モノの状態や状態変化を取得することが可能となってきている⁵⁾。それに従い、エンドユーザによる屋内設置型センサの導入を念頭に置いた調査研究なども行われている³⁾。我々もユビキタス環境はエンドユーザの生活に根ざしたものであり、エンドユーザが容易に導入できるものであるべきであるとする。エンドユーザがセンサの導入を行う際に発生する様々な問題が明らかにされている。例えば、3) ではノード導入の説明を容易にするため色々な種類の派手な色のセンサノードを用いていたが、室内の美観を損ねるためセンサノードを設置したくないという実験参加者がいた。また、3) ではセンサノードとそれが添付されるモノのタイプを関連付けるため、センサノードに添付されたバーコードとバーコードリーダを用いていたが、ノードとモノのタイプを関連付けるという行為自体の意味を理解できない被験者もいた。3) では上記のような問題を考慮して、エンドユーザによるセンサ環境の導入を実現するための指針を示している。しかし、ノードとモノのタイプを関連付けるという行為を軽減する方法については述べられていない。

バーコードを用いる方法は、直感的でありとてもスピーディーな方法であると考えられるが、3) で行われた実験では10個のセンサノードを設置するのに平均して84分もの時間を要している（ただしこれは関連付けのみに要した時間ではない）。しかも10個というセンサノードの数は少ないと言え、数が数十倍に増えた場合マニュアルの関連付けはエンドユーザに多大な苦痛を強いるだろう。

そこで本研究では、エンドユーザがモノにセンサノードを添付するだけで、センサノードとモノのタイプとの関連付けを自動的に推定する方法を考案する。“添付するだけ”を実現するためには、世界中の全てのエンドユーザの家庭において利用できるモノのタイプのモデルデータが必要となる。これは、モデルデータとエンドユーザが添付したセンサノードから得られたセンサデータを比較し、センサデータを出力しうる尤もらしいモデルデータに対応するモノのタイプを推定結果として出力するためである。モノのタイプを推定するだけなら、大量のトレーニングデータと機械学習を用いて実現できる可能性がある。しかし、コンテキストウェアなユビキタスアプリケーションを実現するためにはモノのタイプを推定するだけでは不十分である。われわれは、コンテキストウェアなユビキタスアプリケーションは下記の二つの方法から実現されると考えている。(1) センサノードが添付されたモノの状態や状態変化をトリガとしたサービスをユーザに提供する¹⁾⁶⁾。(2) 複数のセンサノードの出力と教師データを用いてADL (Activity of Daily Living) を推定し、ADLをトリガとしたサービスをユーザに提供する。後者を実現する際、センサノードが添付されているモノのタイプやモノの状態に対してそのセマンティクスを付加する必要はない場合が多い。例えば、9) ではモノの状態変化を取得できるシンプルなセンサを用いて機械学習のアプローチによりADLの推定を行っている。学習に用いる教師データは、ユーザがそのユーザの生活にラベリングすることで作成される。(例えば、15:45から16:00まではトイレに行っていたなど。) そのラベルとラベルが示す期間のセンサの出力を用いて機械学習を行うため、モノの状態や状態変化にセマンティクスは必要無い。前者を実現するためには、モノの状態や状態変化に対してセマンティクスを付加し、それらをトリガとするサービスを提供すればよい。(例えば、ユーザがイスに座れば照明をつけるなど。) 本研究では、上記の二つの方法を満たした環境を構築するために、モノに対してそのタイプを推定した上で、センサデータから得られたモノの状態や状態

変化に対してもセマンティクス与えることを目指す。

推定手法について詳細に説明する前に、本論文が焦点とするモノのタイプについて説明する。TnT は、モノに添付したセンサノードから取得したセンサデータを用いてノードが添付しているモノのタイプを推定する。すなわち、モノの使われ方の特徴を用いる。われわれは、モノのタイプはその使われ方（動作）の特徴から下記の3つにカテゴリ分けされると考えている。(C1) ペンや歯ブラシなど、ユーザによる特徴的な繰り返し動作がユーザによる移動などの動作の中に紛れているモノ。(C2) カップやイスといった、モノのタイプに特徴的な短時間の繰り返しでない動作がユーザによる移動などの動作の中に紛れているモノ。(C3) ドアや引き出しといった、モノのタイプに特徴的な短時間の繰り返しでない動作のみをもつモノ。(C4) テーブルや定規など、モノのタイプに特徴的な動作がない、またはセンサで特徴的な動作を検知できないモノ。C1 のカテゴリを推定するには、ADL 推定の研究における「歩行」などの繰り返し動作を推定する研究が利用できるだろう。C4 のカテゴリは、加速度や照度センサを含むセンサノードのみでは推定が困難である。本論文が主にフォーカスするのは C2 と C3 のカテゴリである。C2, C3 における特徴的な短時間の繰り返しでない動作とは例えばドアが開けられる際の回転運動などである。以降では、C2 や C3 のカテゴリに属するモノを推定する方法について述べる。

3. アプローチ

前述のように、TnT ではセンサデータ中のモノの状態や状態変化に対してセマンティクス与えた上で、モノのタイプの推定を行う。そこで、本研究では、モノの状態と状態変化の特徴を記述したモノのタイプのモデルデータを用意する。モデルデータ中の状態と状態変化にはセマンティクスが付加されている。用意したモデルデータと、ある期間内に得られたセンサデータを比較し、センサデータ中の信号の大きな変化（信号の振幅が大きいセグメント）が、モデル中のどの状態変化に最もマッチするかを計算する。そして、信号変化とそれに最もマッチするモデル中の状態変化とのマッチ度合いの合計から、センサデータとモデルとのマッチ度合いを求める。上記の手順を用意した複数のモデルデータに対して行い、最もマッチ度合いが高いモノのタイプを推定の結果とする。また、上記の手順において求めた信号変化とモデル中の状態変化とのマッチングにより、状態変化にセマンティクスを与える。

しかし、それぞれのエンドユーザの環境において共

通に利用可能な、モノの状態や状態変化に対してセマンティクスが付加されているモデルを作成することは非常に困難である。簡単なアイデアとして教師データを用いた機械学習によりモデルを作成することが考えられるが、エンドユーザの家の環境は様々であるという問題がある。例えば、家によって照明の具合は違うし、同じタイプのモノにもサイズや重さが異なる様なものがある。さらに、ノードを添付する向きや角度が異なれば得られるセンサデータも異なる。つまり、機械学習に用いる教師データはエンドユーザの家で作成されたものでなくてはならない。ADL（繰り返しでない動作）推定の研究では、エンドユーザがそのユーザの生活の間に得られたセンサデータにラベリングすることで教師データを作成することが多い。しかし、エンドユーザがセンサノードを添付しているモノごとにラベリングをすることは簡単ではない。なぜなら、屋内環境に存在するモノごとに何が起こったかを示すラベルを貼る必要があるためである。つまり、モノに添付するノードの数が増えればラベリングの労力は増大する。エンドユーザによるラベリングなしで教師データを作成する研究も存在する。7) は、Web から抽出した知識を用いてある作業をユーザが使うモノの系列としてモデル化している。例えば、ユーザがお茶を作る行為は、ケトルを使ったあとに砂糖、レモン、またはミルクを使うという系列としてモデル化されている。では、日常生活のモノはどのようにモデリングされるであろうか。例えば、ドアには開と閉の2状態があり、ドアが開から閉に状態が変化するときには、加速度センサが回転を検知すると同時に照度センサが照度の変化を検知する。このようなモデルは世界中のほとんどのドアに共通するモデルだろう。しかし、このような誰でも想像できるような知識が記述されるようなリソースは筆者らの知る限り存在しない。

そこで、われわれはエンドユーザが上記のようなモノのモデルの基礎を作成し、それを推定に利用することを想定する。しかし、IT 知識をほとんどもたないエンドユーザが様々なモノのモデルを作成することは不可能であろう。そこで TnT では、エンドユーザによるインターネットを介した集合知によって作成されたモノのモデルを利用することを想定する。すなわち、多数のエンドユーザが協力・分担しながら様々なモノのモデルを作成していくことを考えている。集合知による知識の形成は非常に強力であり、その典型的な例が Wikipedia であろう。Wikipedia では、エンドユーザが書き込むことにより何の利益を得られないにも関わらず、数多くの語彙に対する解説に溢れている。TnT

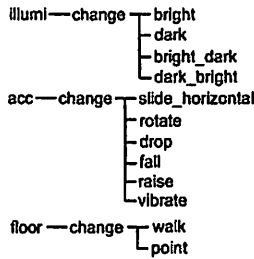


図1 センサが取得できるイベント。
Fig.1 Sensor events.

で用いるセンサ環境におけるモノの定義を形成する行為は、そのような集合知を利用するモデルによく合致すると考える。ユビキタス・センサネットワーク技術はIT知識をもつエンドユーザの興味を引きやすい。また、新しく作成されたモノのモデルはインフラさえ整えれば世界中のIT知識をもつエンドユーザの家庭で検証され、検証結果を考慮して様々な改良が加えられるだろう。

4. モノのモデル

本研究は以下の基本仮定から出発する。

[基本仮定] モノのタイプには、そのタイプに属するインスタンスの状態や状態遷移を表現するそのタイプ固有の有限状態オートマトンが付随する。各タイプの有限状態オートマトンの状態遷移に伴う出力は、時間関係をもつ複数の（センサで検出できる）イベントの集合である。

この仮定から見れば、モノのタイプの状態遷移図（有限状態オートマトン）は、センサ・ユビキタス環境におけるモノそのものの定義であると考えられることができる。本研究で提案するモノの状態遷移図は、モノに添付されているセンサが取得できるイベントを用いて記述したモノの定義であると思わせるため、実際に取得したセンシングデータを解析することで状態の推定などを行うことができる。例えば、ドアは「(1) 床圧センサが歩行を検知。(2) 加速度センサが回転を検知。照度センサが変化を検知。(3) 床圧センサが歩行を検知。」といったイベントが順番に起こる遷移などを含むモノとして定義される。以降では本研究における状態遷移図の表現方法を説明する。

4.1 遷移図の形状

まず、モノのタイプごとに状態遷移図を決定する。簡単に言えば、モノに一般的に起こると考えられる遷移とモノの状態を選定し、それらをつなげることで状態遷移図を決める。

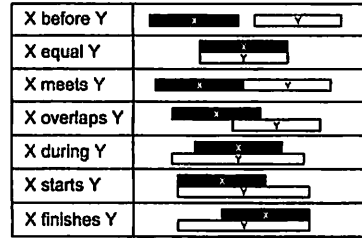


図2 Allenの時区間関係表現。
Fig.2 Allen's interval logic.

4.2 イベントの記述

われわれの提案する状態遷移図における状態遷移は値（有限状態オートマトンの出力）を持ち、その値は遷移の際に加速度センサ、照度センサ、床圧センサにより検知されるイベントを用いて記述される。

われわれは、図1のように、それぞれのセンサが検知するイベントを、ノードの深さが抽象度に対応する木構造で表現する。「illumi」を根とする木は照度センサが検知できるイベントを表しており、最も抽象的なイベントとして「change」がある。そして、その子に対応する具体的なイベントとして、「bright」、「dark」、「bright_dark」、「dark_bright」がある。「bright_dark」は、照度が上昇した後に元の照度に戻る現象を表す。例えば、「照度が変化する」を表現したいときは、「illumi.change」、「照度が上昇する」を表現したいときは「illumi.change.bright」とドットを用いた接続で記述する。加速度センサの木でも最も抽象的なイベントとして「change」がある。そして、その子に対応する具体的なイベントとして、「slide_horizontal」、「rotate」、「drop」、「fall」、「raise」、「vibrate」がある。これらはそれぞれ、「水平方向への移動」、「回転運動」、「落下」、「倒れること」、「鉛直上向きの移動」、「微小振動」を表す。これらのイベントは、3軸加速度センサの信号のみから導くことができると考えられるものである。床圧センサの木でも最も抽象的なイベントとして「change」がある。その子に対応する具体的なイベントとして、「walk」と「point」がある。これらはそれぞれ、床圧センサが検知する人の歩行と、一箇所で何らかの動作をしている現象を表す。

4.3 状態遷移の記述

イベントの集合である状態遷移が持つ値の記述について説明する。状態遷移図における状態遷移が持つ値は、それぞれのセンサによって得られるイベントの集合であり、それらのイベントの間に時間的な関係が存在する。例えばイスに座るという遷移では、床圧センサが人が歩くイベントを検知した後、加速度センサと

照度センサがそれぞれの信号が変化するというイベントをほぼ同時に検知することができる。このようなイベント間の時間的な関係は Allen の時区間関係表現²⁾を用いて表現する。Allen は図 2 に示す 13 通りの時区間関係を定義している (“equal” 以外にはそれぞれインバースの関係が存在する.)。この時区間関係表現を用いて、遷移の際に起こりうるイベントの時間的な関係を記述する。

4.4 状態遷移図の記述

以上を考慮して 11 のモノのタイプの状態遷移図を作成した (その内わけは図 5 に示す)。ただし、本報告では筆者らが遷移図を作成している。滅多に起こらないと考えられる例外的な遷移は遷移図に含めないようにした*。そして、遷移の値に含まれるイベントとその時間的な関係を記述する際、どのような環境においても共通してイベントを記述するようにした。図 3 にドアタイプの状態遷移図を図式化したものを示す。“Open” の遷移から出ている吹き出しは、その遷移が起こる際のイベントの集合を時系列で表しており、その遷移の値である。この例では、“Open” の遷移は 2 通りのケースが存在しており、1 つ目は人が歩きながらドアを開けて通り抜けていくケースを、2 つ目は人がドアの前まで歩いてきて立ち止まってからドアを開けドアから離れて歩いていくケースを表している。このように、1 つの遷移に対して複数のケースを記述できるため、さまざまな状況を考慮できる。

5. センサデータ解析

以下ではセンシングデータからの変化の抽出とイベントへの抽象化の手順について説明する。

5.1 変化の抽出

定常状態においてもセンシングデータの値は常に振動しているため、定常状態の振動より大きな振動を変化として抽出する。ただし、定常状態の振動にはセンサノードごとに個体差があり、本研究のように多数のセンサノードを設置する環境では定常状態の振動の大きさをあらかじめ設定しておくことは困難である。さらに、加速度センサにおいては添付の仕方や空調設備などの条件により、その振動の大きさが変化する現象も見られる。このような環境において変化を抽出するため、ある時点の定常状態の変化量を用いる。つまり、ある時点のサンプル s が変化しているか判定するために、 s の周辺のサンプルの変化量から s 周辺の定常状

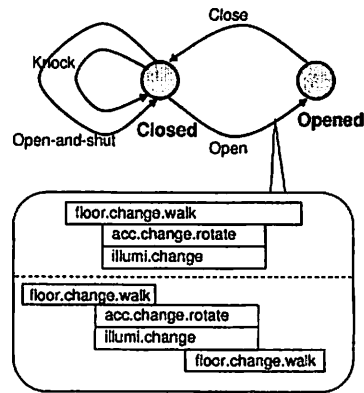


図 3 ドアの状態遷移図。
Fig. 3 State transition diagram of a door.

態の変化量を求める。ある時点 p の変化量とは、 p を中心とする幅が 15 サンプルのウィンドウ内の最大値と最小値の差とする。以下に定常状態の変化量の求め方について簡単に説明する。まず、 s の周辺から選んだサンプルの変化量を計算し、変化量の近さからサンプルをクラスタリングする。さらに、同じクラスタに属するサンプルの変化量の平均を求める。要素数が閾値以上のクラスタの変化量の平均のうち最大のものを s の周辺の定常状態の変化量とする。そして、変化量が定常状態の変化量を大きく上回る区間を変化とする。

5.2 変化の抽象化

抽出した加速度センサの変化を図 1 の “acc” を根とする木に示すイベントに抽象化する。“acc.change.rotate” と “acc.change.slide_horizontal” の検出は、加速度を 2 回積分し、センサノードの軌跡を求めることで行っている。“acc.change.drop” と “acc.change.fall” の検出は、センサノードの軌跡と鉛直方向の加速度の大きさを求めることで行っている。“acc.change.raise” の検出は、軌跡の方向が鉛直上向きかどうかを判定することで行っている。“acc.change.vibrate” の検出は、加速度がほとんど高周波成分かどうかを判定することで行っている。そして、以上のイベントに該当しないものは “acc.change” とした。

抽出した照度センサの変化も同様に図 1 の “illumi” を根とする木に示すイベントに抽象化する。本研究で用いた照度センサは、0 から 1024 までの値を照度として出力する。本実装では、変化の前後で照度が 20 以上上がった場合は “illumi.change.bright” とし、20 以上下がった場合は “illumi.change.dark” とした。照度が 20 以上上がってから元の照度に戻った場合は “bright_dark” とし、照度が 20 以上上がってから元の

* 例外的な遷移を含めると提案する推定方式の精度が下がることがあった。また、遷移図において例外的な遷移を明示して記述することもでき、そのような遷移は推定方式において無視した。

照度に戻った場合は“darl_bright”とした。それ以外は“illumi.change”とした。

5.3 床圧センサデータの解析

後述するが、本論文で行う実験ではセンサフロアシシステムを用いて人の歩行を検知する。センサフロアシシステムは、床面の圧力の有無を180mm四方の分解能で観測する。本研究では、圧力が無から有への変化に注目する。まずこれらの変化を位置的、時系列的な距離から関連のある変化同士にクラスタリングしたあと、クラスタリングされた変化のグループを図1の“floor”を根とする木に示すイベントに抽象化する。クラスタリングは、任意の2つの変化同士の、時間と位置の距離が閾値（実装では2秒と100cm）以内のものを同じクラスタとすることで行う。

イベントへの抽象化に関しては、同じクラスタ内における変化の位置の分散、クラスタ内の任意の二点間の最大距離、および、同じクラスタ内の最初に起こった変化と最後に起こった変化の時間差を用いる。ここで位置の分散とは、床圧センサシステムにおける変化のX座標とY座標の分散の平均としている。分散と任意の二点間の最大距離がそれぞれ閾値（実装では2と1m）より大きいときは“floor.change.walk”とする。分散、任意の二点間の最大距離、および、最初と最後の変化の時間差がそれぞれ閾値（実装では2と1mと2秒）より小さいときは“floor.change.point”とする。それ以外は“floor.change”とする。

6. モノとその状態の推定

センサノードから得られたイベント集合の系列が出力される尤もらしさ（スコア）をモノの状態遷移図ごとに計算し、スコアが最も大きい状態遷移図に対応するモノにそのノードが添付されているとする。例えばあるセンサノードのセンサデータから、イベント集合系列 g が得られたとき、ある状態遷移図 $T1$ に対するスコア $SC(T1, g)$ を計算する。そして、この手順を用意した全ての状態遷移図に対して行う。ただし、

$g = g_1 \rightarrow g_2 \rightarrow g_3 \rightarrow \dots \rightarrow g_n$ である。

ここで、まず2つのイベント集合系列

$g = g_1 \rightarrow g_2 \rightarrow g_3 \rightarrow \dots \rightarrow g_n$ と

$t = t_1 \rightarrow t_2 \rightarrow t_3 \rightarrow \dots \rightarrow t_n$ 。

との類似度 $Sc(g, t)$ を以下で定義する。

$$Sc(g, t) = \sum_{j=1}^n Sim(g_j, t_j, f_j),$$

ここで、 f_j はイベント群 g_j に含まれるそれぞれのイベントとイベント群 t_j のイベントとの対応付けである。ただし、同じセンサに關係するイベント同士を対応付ける。 $Sim(g, t, f)$ は2つのイベント群 g と t の類似度であり、下記で計算される。

$$Sim(g, t, f) = SimA(g, t, f) \cdot SimT(g, t, f),$$

ここで、 $SimA(g, t, f)$ は、 t のそれぞれのイベントと g のイベントとの類似度の総和である。イベント間の類似度は、図1(a)に示す木を用いて計算する。例えば、 t に含まれるイベントが“acc.change.rotate”(node n_1)で、写像によりそれに対応する g に含まれるイベントが“acc.change.drop”(node n_2)のとき、これらの類似度は下記の式で計算される。

$$SimN(n_1, n_2) = 1 - \frac{dist(n_1, n_2)}{\max(d(n_1), d(n_2))},$$

ここで、 $dist(n_1, n_2)$ は木におけるノード間の距離を、 $d(n_1)$ は木におけるノードの深さを示す。この場合は、“rotate”と“drop”のノード間距離は2であるため類似度は $\frac{1}{2}$ となる。ただし、遷移のイベントの方がイベント集合のイベントより抽象度が大きい場合は、これらの類似度は1となる。

また、 $SimT(g, t, f)$ は、状態遷移図の遷移においてAllenの時区間關係表現で表現されているそれぞれのイベント間の關係と、それに対応するイベント集合におけるイベント間の關係の距離の総和の逆数である。例えば、 t に“acc.change”と“illumi.change”が含まれ、 g に含まれる“acc.change”と“illumi.change.bright”にそれぞれ対応するとする。そして、 t のイベント同士の關係が“equal”で、 g のイベント同士の關係が“before”であるとすると、“before”が“equal”となるためには、 g に含まれるいずれかのイベントの開始時刻と終了時刻を何秒かずらす必要がある。このずらすのに必要な時間をイベント間の關係の距離として定義している。

ここで、床圧センサから取得したイベントについて述べる。遷移に床圧センサのイベントが含まれるとき、それに対応する床圧センサのイベントを選ぶ必要がある。しかし、それぞれのセンサノードは室内における位置の情報を持たないため、床圧センサのどのイベントが対応するか分からない。ここでは、 $Sim(g, t, f)$ を最も大きくするものを対応するイベントとして選んでいる。以上の説明では、イベント集合と遷移のイベント同士の写像について説明していなかった。例えば、床圧センサに関するイベントが遷移とイベント集合にそれぞれ2つずつ、照度と加速度センサに関するイベントがそれぞれ1つずつ含まれているとき、遷

移に含まれるイベントとイベント集合に含まれるイベントの写像は、2通り(2!×1!×1!)ある。ここでは、 $Sim(g, t, f)$ を最も大きくする写像を選んでいる。

g が得られたときのスコア $SC(T1, g)$ は、最大の類似度 $Sc(g, t)$ として定義する。つまり、 $Sc(g, t)$ を最も大きくする t を求めることで、 $SC(T1, g)$ が得られる。ただし t は、 $T1$ に含まれる状態遷移の系列であり、それぞれの要素が g の要素に対応する。つまり t の i 番目の要素が、 g の i 番目の要素(イベント集合)に対応する遷移であるとする。以下に $Sc(g, t)$ を最も大きくする t を求める手順を示す。

- (1) 全てのイベント集合ごとに、状態遷移図 $T1$ の全ての遷移との類似度 $Sim(g, t, f)$ を計算する。図3の“Open”のように複数のケースがある場合は、全てのケースにおいて計算する。
- (2) 系列に含まれるイベント集合と遷移の対応の組み合わせを求める。例えば時系列的に並んだ10個のイベント集合があるとき、それぞれのイベント集合ごとに対応し得る遷移の組み合わせの系列を求める。すなわち、イベント集合の系列 $g = g_1 \rightarrow g_2 \rightarrow \dots \rightarrow g_n$ からは、対の系列 $sq_1 : (g_1, t_1) \rightarrow (g_2, t_1) \rightarrow \dots \rightarrow (g_n, t_1)$, $sq_2 : (g_1, t_2) \rightarrow (g_2, t_2) \rightarrow \dots \rightarrow (g_n, t_2)$, $sq_3 : (g_1, t_3) \rightarrow (g_2, t_3) \rightarrow \dots \rightarrow (g_n, t_3)$, ... が得られる。ただし、 t_i の添え字は状態遷移図における遷移のIDを表すものとする。

この例では($T1$ の遷移の数)¹⁰通りもの系列を求める必要がある。そこで、下記の3つの制約を満たす対の系列をランダムにできるだけ多く求める*。ただし g_i に対応する遷移を選ぶとき、類似度 $Sim(g_i, t, f)$ が大きい t ができるだけ選ばれるような貪欲アルゴリズムを用いる。

- (1) 状態遷移図を考慮して矛盾する系列は無視する。ここで矛盾とは、例えばドアの状態遷移図において“Open”の遷移が起こった直後に“Open-and-shut”が起こるなどを示す(閉状態で“Open-and-shut”は起こらない)。(2) あるイベント集合 g_i に対応する遷移 t_n を決めるとき、遷移図においてある状態に遷移する遷移が複数あれば、それらの遷移のうち $Sim(g_i, t_n, f)$ が最大のもののみを選ぶ。(3) 一つの系列において、加速度と照度センサの波形がそれぞれ類似しているイベント集合同士は、同じ遷移に対応付けられる。波形の類似度はDPマッチング⁸⁾を用いて計算した。

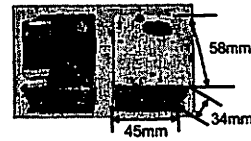


図4 センサノード。
Fig. 4 Prototype sensor node.

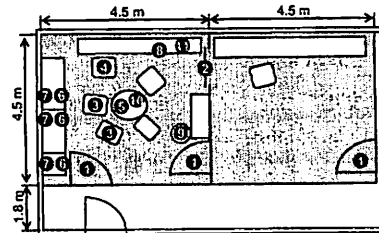


図5 センサ部屋。
Fig. 5 Sensing room.

- (3) 対応の系列の集合 $\{sq_1, sq_2, \dots\}$ に含まれる全ての sq_i から得られる t に対して $Sc(g, t)$ を計算し、最も大きい $Sc(g, t)$ を得る t を求める。

以上の手順を用意した全ての状態遷移図において行い、最もスコアの高い系列 t を出力する状態遷移図に対応するモノをモノタイプの推定結果とする。そして、最もスコアの高い系列 t が指定する遷移をデータ収集期間に起こった遷移として出力する。

7. 評価

まず、本研究で利用するセンサノードについて説明しよう。センサノードは、われわれの研究グループが開発したもので、一般的に普及している照度センサ、焦電センサ、三軸加速度センサを搭載する。これらのセンサから取得したデータをデータの保存用に設置したサーバシステムに送信するために、XPort(Lantronix)もしくは無線モジュール(NECエンジニアリング, ZB24FM)を搭載する。約60msecごとにその間にセンシングしたデータの平均値を送信する。図4にプロトタイプのセンサノードを示す。

以上のセンサノードを設置して観測を行うために、図5に示すようなセンサネットワークシステムを備えたオフィス環境実験室を構築した。実験室は、作業部屋としてレイアウトしており、イスや机、棚などの家具とPC類を配置して、日常的な作業が行える環境を整えている。この部屋では、午前9時から午後5時頃まで1人から4人程度の作業員(研究者ではない)が各々の作業や本センサネットワークシステムの整備などを行っている。また、図5の網掛け部分の床にはセ

* 実装では1分間の処理で得られる系列としている。

表1 モノのタイプ (o) と遷移 (t) の推定精度 (%)。

Table 1 Accuracies of object class (o) and transition (t) inferences(%).

o/t	日数	ドア	スライド ドア	背もたれ 付きイス	回転 イス	机	ロッカー の扉	ロッカー 内部	引き出し	ごみ箱 のふた	カップ	目覚まし 時計	平均
o	1	80	80	90	100	0	87	87	80	100	20	20	74
o	2	100	100	100	100	0	100	100	100	100	25	25	86
t	1	94	51	84	58	-	92	90	89	95	45	100	81
t	2	92	73	81	67	-	97	80	78	99	50	100	80

ンサフロアシステム (Vstone, VS-SF99) を設置しており、床面の圧力の有無を 180mm 四方の分解能で観測することができる。

図 5 はまた、センサノードを添付したモノの位置を示している。具体的には、ドア (1)、スライドドア (2)、背もたれのあるイス (3)、車輪付き回転イス (4)、机 (5)、ロッカーの扉 (6)、ロッカーの内部 (7)、引き出し (8)、ふた付きごみ箱 (9)、カップ (10)、目覚まし時計 (11) にセンサノードを添付した。

評価では、2006 年 10 月 11 日から 15 日までの間に図 4(b) に示した 18 個のセンサノードから取得したデータを用いてモノの推定と遷移の推定を行い、その精度について評価する。ここでモノの推定の精度は、あらかじめ用意した状態遷移図の数に大きく依存するため、前述した 11 種類のモノ以外に 9 種類の状態遷移図を用意し、計 20 種類の状態遷移図の中から推定を行った。このとき、推定に使用するセンサデータの日数を変更して手法を実行した (1 日分と 2 日分)。

表 1 にその結果を示す。まずモノのタイプの推定結果について述べる。机、カップ、目覚まし時計以外の推定精度は 1 日分のセンサデータを用いても十分高いことが分かる。以下では推定に失敗したモノとその原因を考察する。机は日常生活において特徴的な動きをしない。また、カップや目覚まし時計もユーザが自由に動かすことができるため、状態遷移図を用いてこれらのモノを正確に定義することは難しい。1 日分のセンサデータを用いた推定精度は 2 日分のそれより低かった。これは、日常生活においてあまり使われなかったオブジェクトがあったためである。例えば、あるドアは 1 日に 2 回ずつ開閉されたのみであったため、その推定に失敗していた。

モノのタイプの推定が成功したモノに対して遷移の推定も行った。その評価は室内を記録したビデオ映像を見ながら行った。スライドドア、回転イス、カップ以外は高い精度を達成していることが分かる。以下では、遷移推定の失敗事例について論考する。部屋の照明が消された状態でスライドドアが何度も開閉される日があった。このとき、スライドドアに添付したノードは照度の変化を検知することができなかったため、遷移

図において照度変化のイベントを含むスライドドアの開閉を正しく推定できなかったと考える。一方、状態遷移図において定義されていない遷移は正しく推定できなかった。例えば、作業員がごみ箱の中をあさっている現象は、遷移図において定義されている '捨てる' という遷移として推定されていた。また回転イスにおいては、作業員がイスを全く動かすことなく立ち上がる現象を検知することができなかった。

8. おわりに

本研究では、容易なユビキタス環境の導入を実現するためのモノの推定と状態の推定を行うフレームワークである Tag and Think の実現手法を構築した。本提案手法を、センサネットワーク環境で取得したデータに対して評価し、その結果モノのタイプと各モノの状態を十分高い精度で推定できることを示した。本提案では、提案した状態遷移図を用いてモノの定義を行った。状態遷移図を用いることで、ユビキタスセンサネットワーク環境におけるコンテキストウェアなサービスの提供に適したモノの定義を行うことができる。

参考文献

- 1) M. Addlessee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper, "Implementing a sentient computing system," *IEEE Computer*, vol. 34, pp. 50-56, 2001.
- 2) J. F. Allen, "Maintaining knowledge about temporal intervals," *Commun. of the ACM*, 26(11):832-843, 1983.
- 3) C. Beckmann, S. Consolvo, and A. LaMarca, "Some assembly required: supporting end-user sensor installation in domestic ubiquitous computing environments," *Proc. UbiComp 2004*, pp. 107-124.
- 4) J. Giles, "Internet encyclopaedias go head to head," *Nature*, vol. 438, pp. 900-901, 2005.
- 5) S.S. Intille, E. Munguia Tapia, J. Rondoni, J. Beaudin, C. Kukla, S. Agarwal, L. Bao, and K. Larson, "Tools for studying behavior and technology in natural settings," *Proc. UbiComp 2003*, pp. 157-174, 2003.
- 6) C. Kidd, et al., "The aware home: a living laboratory for ubiquitous computing research," *Proc. CoBuild99*, 1999.
- 7) M. Philipose, K. P. Fishkin, and M. Perkowitz, "Inferring activities from interactions with objects," *IEEE Pervasive computing*, 3:50-57, 2004.
- 8) H. Sakoe, S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, and Signal Process.*, Vol. ASSP-26, pp. 43-49, 1978.
- 9) E.M. Tapia, S.S. Intille, K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," *Proc. Pervasive 2004*, pp. 158-175, 2004.