

Tag and Think: センサネットワークにおけるモノ自身を推定するフレームワーク

Tag and Think: Inferring Object Type in Sensor Network Environments

前川卓也*¹

Takuya Maekawa

柳沢豊*¹

Yutaka Yanagisawa

岡留剛*¹

Takeshi Okadome

*¹NTT コミュニケーション科学基礎研究所

NTT Communication Science Laboratories

The system framework, *Tag and Think*, proposed here enables us to introduce ubiquitous environments easily and simply by attaching sensor nodes to physical objects without any information about the objects. In the framework, we present a method that infers types of physical indoor objects and their states automatically from sensor data produced by the sensor nodes, by using real-world knowledge constructed from a person's common knowledge.

1. はじめに

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場合、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手動で与えることは困難である。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を自動的に推定するフレームワーク *Tag and Think* (TnT) を提案する。

TnT では、汎用的なセンサノードがさまざまなモノに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から、収集したデータを最も出力しうるモデルを決定することでモノのタイプの推定を行う。さらに、モデルを用いてモノにどのような状態変化が起こったかも同時に推定する。本研究における汎用的なセンサノードは、加速度センサや照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。本稿では、特に人の経験や実世界に関する知識を用いて屋内にあるモノのタイプとその状態を推定する手法を紹介する。

2. 背景と目的

近年、安価になってきたセンサノード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。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 [Beckmann 04]。特に本稿では、エンドユーザによるセンサ導入の際の、センサノードとそれが添付するモノのタイプの関連付けの問題に注目する。関連付けとは、センサノードがどのタイプのモノに添付されているかをコンピュータに入力する作業である。関連付けの作業は、センサノードの数が増えれば増えるほどエンドユーザにとっての負担となる。

そこで本研究では、エンドユーザがモノに汎用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を行うことを目指す。‘添付するだけ’を実現するために、世界中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をあらかじめ用意する。そして、そのモデルとセンサノードからある期間に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最もよいモデルを決定することでモノのタイプを推定する。さらに本研究では状況依存サービスを実現するために、モノに起こった状態変化も推定する。状態変化とは、例えば‘ドアが開く’や‘イスに人が座る’などである。

3. アプローチ

モノには、そのモノのタイプに固有な特徴的な使われ方がある。例えば、人が歯ブラシを使って歯を磨くときに、それに添付しているセンサノードから得られる加速度の周波数成分には特徴がある。また、人がドアを開けるときは、それに添付しているセンサノードから回転運動と照度変化が同時に得られるだろう。TnT では、モノに添付されたセンサノードから得られたデータから、その‘モノの使われ方の特徴’を検出し、用意したモデルデータと比較することで、モノのタイプとモノの状態変化を推定する。ただし、モノが使われているときに状態変化が起こっているとすると、われわれは、モノのタイプはその使われ方の特徴から下記の3つに分類されると考えている。(C1) ペンや歯ブラシなど、特徴的な繰り返し動作をもつモノ。(C2) ドアやイスといった、モノのタイプに特徴的な動作がセンサの特徴的な出力の組み合わせからなるモノ。(C3) 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 のカテゴリに属するモノの特徴的な動作を検出するには、ADL 推定の研究における‘歩行’や‘自転車での走行’などの繰り返し運動を推定する研究 [Bao 04] を利用できるだろう。つまり、人がペンや歯ブラシなどを用いる際の加速度信号のフーリエ成分を用いてそれらのモノをモデル化することで、C1 のカテゴリに属するモノのタイプと状態変化は推定できる。C2 に属するモノのタイプの特徴的な動作は特徴的な周波数成分などをもち、特徴的な運動や照度変化などからなり、それらは大まかに想像可能である。本稿では、特に C2 に属するモノのモデルデータの作成方法とモノのタイプの推定方法について以降で述べる。C3 のカテゴリに属するモノのタイプを推定することは極めて困難であるが、他のカテゴリに属するモノとの

連絡先: 前川卓也, NTT コミュニケーション科学基礎研究所, 京都府相楽郡精華町光台 2-4, 0774-93-5394, maekawa@cslab.kecl.ntt.co.jp

共起関係などを用いて推定することが可能と考える。例えば、イスとテーブルは同時に使われることが多いため、イスのセンサーデータと共起するセンサーデータを出力するセンサノードはテーブルに添付されている可能性が高いという推論ができる。

では、TnTにおけるC2に属するモノのタイプのモデル作成方法について説明を始めよう。前節で述べたように、TnTではセンサー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も行う。それを実現するためには、モノの状態と状態変化の特徴を含み、それらに対してセマンティクスが付加されているモノのタイプのモデルデータを用意しなければならない。簡単なアイデアとして教師信号を用いた学習によりモデルを作成することが考えられる。しかし、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ふし、同じタイプのモノにもサイズや重さは多様である。すなわち、学習によりモデルを作成するならば、エンドユーザの家で得られたセンサーデータから教師信号を作成しなければならない。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ーデータにラベリングすることで教師信号を作成することが多い。しかしADLにラベリングすることと異なり、エンドユーザがセンサノードを添付している膨大なモノごとにラベリングをすることは簡単ではない。

エンドユーザによるラベリングなしで物理現象をモデル化する研究も存在する。[Philipose 04]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。このようなモデルは世界中の環境において利用できるだろう。ここで、世界中の屋内環境で利用可能なC2に属するモノのモデルについて考えてみよう。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。このモデルは、誰もが容易に想像しうるモデルであり、世界中のほとんどのドアに共通する性質を反映しているであろう。もちろん実環境では、そのようなモデルに厳密に合致しない現象も起こるだろう。しかし、そのような現象に対してはロバストな推定アルゴリズムを利用することで対処できると考える。われわれは一般的な知識を用いて上記のようなモデルを容易に作成することができる。そのモデルは多くの環境で利用可能であるが、現実世界で起こる現象を全て厳密に記述できていない。一方、教師あり学習を用いて厳密なモデルを作成することができる。そのモデルは作成にコストがかかる上、特定の環境でしか利用できない。われわれは、容易に作成可能だが不完全なモデルとロバストなアルゴリズムを用いる手法が現実的であると考えられる。

本稿では、経験と実世界に関する知識に基づいて作成された上記のようなモノのモデルをモノのタイプや状態変化の推定に用いることを想定する。そして、実験により実験参加者に作成してもらったモデルがモノのタイプや状態変化の推定にどの程度有効かを検証する。そのために、次節では誰もが簡単に記述できるようなモデルの記述方式を提案する。さらに5.では、マニュアルで作成されたモデルとセンサーデータを用いてモノのタイプと状態変化を推定する手法を提案する。

4. モノのタイプのモデル

われわれは、C2に属するモノをモデル化するために状態遷移図を選んだ。状態遷移図を用いることで、モノの状態と状態

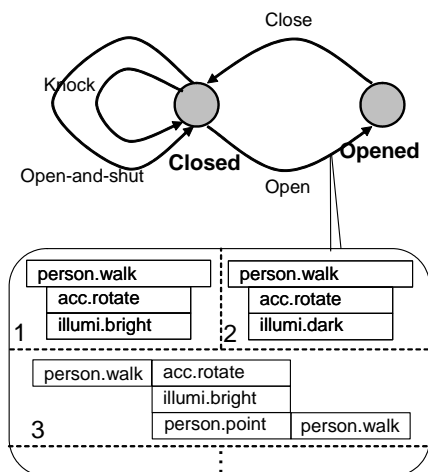


図 1: ドアの状態遷移図。

変化を直感的に記述できると考える。以降では、状態変化を状態遷移と呼ぶ。

4.1 モノのタイプの状態遷移図

図1は、後述する実験で得られたドアタイプの状態遷移図を図式化したものである。‘Open’の遷移から出ている吹き出しは、その遷移が起こる際にセンサノードによって検知されるであろうイベントの集合を時系列で表している(以降ではイベントの集合をイベント集合と呼ぶ)。遷移はイベント集合のセットからなっており、‘Open’の例では3つのイベント集合を示している。ただし、‘acc.rotate’, ‘illumi.bright’, ‘illumi.dark’は、加速度センサにより検知される回転運動のイベント、照度センサにより検知される照度上昇のイベント、照度下降のイベントをそれぞれ表す。また、‘person.walk’と‘person.point’は、人の歩行と、人が一箇所まで何らかの動作をしている現象を表し、モノに添付しているセンサノード以外のデバイスから得られるイベントである。イベント集合の詳細な記述方法については後述するが、この例では、‘Open’の遷移に含まれる3通りのケースを示しており、1つ目と2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、3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。

4.2 センサデータと状態遷移図

センサーデータと状態遷移図の関係について説明する。遷移図における遷移は、センサーデータにおける振幅が大きく変動している区間に対応する(以降ではその区間を変化と呼ぶ)。遷移図における状態は、センサーデータにおいて振幅がほとんど変動していない区間に対応する。図2を用いてセンサノードにおける用語について説明する。図は、ノードを添付したドアが開き、閉じ、開閉する遷移を2回繰り返した際に得られたX軸方向の加速度、Z軸方向の加速度、照度である。Y軸方向の加速度センサの波形は省略している。 ax_i および az_i ($i = 1, \dots, 6$) は、X および Z 方向の加速度センサから得られた変化であり、 ai_i は照度センサから得られた変化である。加速度センサの変化である aa_i ($i = 1, \dots, 6$) は、 ax_i, ay_i, az_i の和集合である。 ag_i ($i = 1, \dots, 6$) は、変化の集合であり、以降では変化集合と呼ぶ。例えば、 ag_1 は aa_1 と ai_1 からなる。変化集合は遷移(イベント集合)に対応し、例えば、 ag_1 と ag_4 は ‘Open’ に対応

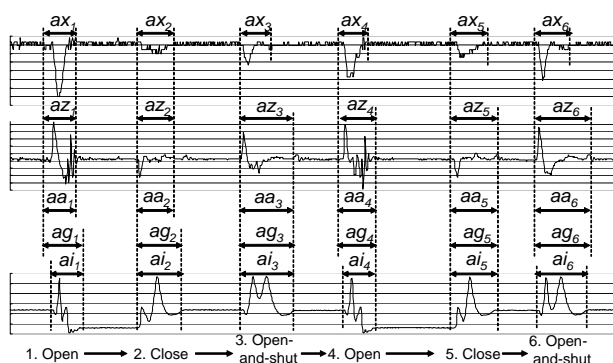


図 2: ドアに添付されたセンサノードから取得した信号。

している。タイプの推定を行う際、われわれは変化集合とイベント集合の比較を行う。イベント集合はシンボル(‘acc.rotate’や‘illumi.bright’など)で記述されているため、比較の際は変化を抽象化する(シンボルに変換する)。例えば、 aa_1 は‘acc.rotate’に抽象化される。さらに、変化集合を時系列順に並べることで、 ag_1, ag_2, ag_3, \dots といった変化集合系列を得ることができる。次節で説明する推定手法では、(1) センサデータから変化を抽出し、(2) 変化を抽象化し、(3) 変化をグループ化し、(4) グループ化した変化集合を時系列順に並べ、(5) 変化集合系列に最も合致する尤もらしい状態遷移図を決定する。

4.3 イベント集合

イベント集合の記述方法について説明する。まず、われわれが用意したイベントについて紹介する。‘illumi.bright’, ‘illumi.dark’, ‘illumi.bright_dark’, ‘illumi.dark_bright’を照度センサが取得できるイベントとして用意した。‘illumi.bright_dark’は、照度が上昇した後に元の照度に戻る現象を表す。また、‘acc.slide_horizontal’, ‘acc.rotate’, ‘acc.drop’, ‘acc.fall’, ‘acc.raise’, ‘acc.vibrate’, ‘acc.other’を加速度センサが取得できるイベントとして用意した。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」、「その他の動き」を表す。これらは、加速度センサデータの解析により高い精度で検知できるイベントである。ただし、累積誤差によりイベントの初期動作のみ高い精度で検知できる。例えば、図2の“Open-and-shut”では、ドアが回転運動をした直後に、反対方向に回転運動をする。しかし、上記の理由により反対方向の回転が起こったことを正確に検知することは困難であるため、これらの運動を表現するときは最初に起こった運動(‘acc.rotate’)のみを用いる。‘person.walk’と‘person.point’を他のデバイスから取得できる人の歩行に関するイベントとして用意した。

イベント集合のイベント同士には時間関係がある。例えば、図1の“Open”の1つ目のイベント集合では、‘person.walk’が起こっている間に、‘acc.rotate’と‘illumi.bright’が同時に起こっている。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[Allen 83]を用いて表現する。Allenはイベント間の時区間関係を13通り定義している。例えば、“X before Y”は、イベントXのあとにイベントYが起こっている関係を表す。

イベント集合はその生起確率をもつ。経験や実世界の知識により、頻度をよく起こるかそうでないかの2値で記述する。例えば後述する実験で作成された遷移図では、ドアの前で立ち話をした後にドアを開けるという行為に対応するイベント集合

は、生起頻度が低いとされていた。

5. 推定手法

5.1 変化区間の抽出

音声認識の研究において、音声信号から音声区間(voice activity)を抽出する研究は多く行われている。これらの研究では、発話区間とそれ以外の区間のモデルをあらかじめ学習によって作成しておき、テストデータがどちらのモデルに近いかどうかを判定する。われわれもこの方法と同様のアプローチを取る。しかし、エンドユーザが変化している区間のデータを用意することは困難であるため、何も起こっていないときのセンサデータから作成したモデルとテストデータとの比較からテストデータが変化を含むかどうかを判定する。モデルはセンサごとに、ある長さのセンサデータセグメントのフーリエ成分を用いてGaussian Mixture Model(GMM)で作成する。実際にセンサデータセグメントが変化を含むかどうかは、そのセグメントのフーリエ成分のモデルに対する対数尤度が閾値より小さいかどうかで判定する。

5.2 変化の抽象化

抽出された変化は、4.3節で述べたイベントに抽象化される。‘acc.rotate’と‘acc.slide_horizontal’の検出は、ルンゲクッタ法を用いて加速度を2回積分し、センサノードの軌跡を求めることで検出する。‘acc.drop’と‘acc.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.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.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。

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。変化の前後で照度が20以上上がった場合は‘illumi.bright’とし、20以下下がった場合は‘illumi.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‘illumi.bright_dark’とし、照度が20以上上がってから元の照度に戻った場合は‘illumi.dark_bright’とした。

5.3 モノのタイプの推定

変化集合系列とモデル(状態遷移図)を比較する手段として、高速でロバストであるViterbiアルゴリズムを用いる。Viterbiアルゴリズムは、Hidden Markov Model(HMM)が既知の場合に、与えられた観測系列から最尤推定になる状態系列を動的計画法で求める。本手法では、HMMがモノのタイプの状態遷移図に、観測系列が変化集合系列に、状態系列がイベント集合系列(状態遷移系列)に対応する。Viterbiアルゴリズムにより、変化集合系列($AG = ag_1, ag_2, \dots, ag_T$)の尤度 $P(D, AG)$ をそれぞれの状態遷移図 $D = (A, ST)$ に対して求め、尤度が最も大きい遷移図に対応するタイプを推定結果として出力する。すなわち、

$A = \{a_i\} : a_i = 1/N$ は初期に状態 $s_i (1 \leq i \leq N)$ に存在する確率、
 $ST = \{t_{n,m}\} : t_{n,m}$ は状態 s_n から s_m への遷移 ($1 \leq n \leq N; 1 \leq m \leq N$),
 $t_{n,m} = \{eg_i\} : eg_i$ is $t_{n,m}$ が含むイベント集合 ($i \geq 0$) としたとき、Viterbiアルゴリズムでは $P(D, AG)$ は以下のように求められる。

$$P(D, AG) = \max_{1 \leq i \leq N} (p_{i,T+1}),$$

$$p_{i,j} = \max_{1 \leq k \leq N} (p_{k,j-1} + Sim(t_{k,i}, ag_{j-1}, Th)),$$

$$(1 \leq i \leq N; 1 \leq j \leq T+1),$$

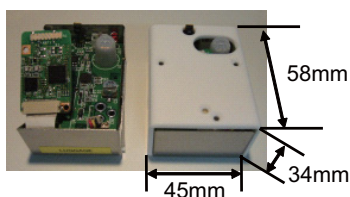


図 3: センサノード.

$$Sim(t_{n,m}, ag_i, Th) = \max_{j, e_j > Th} (sim(ag_i, eg_j) \cdot e_j),$$

ここで、 Th は閾値である。 $j = 1$ のとき $p_{i,0}$ は a_i である。 $sim(ag_i, eg_j)$ は、 ag_i と eg_j の類似度であり、 e_j は eg_j の生起頻度である。われわれは、 ag_i と eg_j の類似度を ag_i を eg_j に変換するためのコストの逆数としている。[Freksa 91] では時間関係間の距離が定義されており、類似映像検索などに用いられている。われわれはこの距離を変換のためのコストとして用いる。また、イベントを追加または削除するコストを 20、イベントを他のイベントに変換するコスト (ただし同じセンサのイベントにしか変換できない) を 5 とした。これらの値は事前実験でよい結果を得たものを採用した。

6. 評価

6.1 実験環境

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、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、通信モジュールを搭載している。図 3 にセンサノードを示す。また、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。部屋の床にはセンサフロアシステムを設置しており、人の歩行を検知できる。室内の C2 と C3 に属すると考えられるモノにそれぞれ 18 のセンサノードを添付した。C2 に属するモノとしては、2 脚のイスの背もたれ、車輪付き回転イスの背もたれ、カップ、3 枚のドア、スライドドア、3 台のロッカーの扉の外側、3 台のロッカーの内部、引き出しの外側、ふた付きごみ箱のふたの内側に添付した。C3 に属すると考えられるモノとしては、机と目覚まし時計に添付した。

6.2 モデルの作成

3 人の実験参加者にモデルの作成方法を記述した文書を読んでもらった後、ドローソフト (Microsoft PowerPoint) を使ってモデルを描いてもらった。参加者にはそれぞれ 11 種類のモノに加え、13 のモデルを作成してもらった。これらのモノは、家具オンラインショップ KAGOO (<http://www.kokugai.com/kagoo>) が扱う約 4000 点の商品から、ユーザがそれを利用したときに特徴的な動作が起こると考えられるモノから選んだ。ただし、異なるモノでも同じ動作や役割をもつと考えられるモノは区別しなかった。例えば、机の引き出しとロッカーの引き出しは同じものとする。遷移図の形状 (状態数や遷移) を決定する際の参考にしてもらうため、参加者に 1.4GByte のコーパス (ニューヨークタイムズ) から抽出した名詞 (モノ) に関連する上位 10 の頻出動詞を提示した。例えば、イスに対しては、'sit down', 'take a seat', 'get up' などを与えた。

6.3 実験

われわれは、連続する 5 日間に実験環境内のモノに添付した 18 のセンサノードからセンサデータを収集した。連続する 2 日間のデータを用いて、センサノードが添付しているであろうモノのタイプに対応するモデルを 6.2 節で作成した 24 のモデルから推定する試行を 4 回行った。実験の結果、回転イス、ドア、スライドドア、ロッカーの扉、ごみ箱に関しては、どの参加者が作成したモデルにおいてもほぼ推定が成功していた。ただし、ある被験者が作成したモデルにおいて、ロッカーの扉の推定が全て失敗していた (ロッカー扉がドアと間違えて推定されていた) これはモデルの書き分けに起因している。他の参加者は、モノのモデルがモノのタイプの推定に利用されることを知っていたため、ロッカー扉とドアの特徴を考慮してモデルに違いが出るように書いていた (ロッカーを開けるときは、ロッカーの前に立ち止まることが多い。一方、ドアを開けるときは歩きながら開けることが多い。) その参加者もその事実は知っていたが、書き分けに対してあまり注意を払っていなかった。モデルの書き分けは、タイプの推定精度に大きな影響を与える。背もたれ付きイスに関しては、人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ため、精度が低かったと推測される。また、2 人の参加者から引き出しの外側のモデル作成に苦労したという感想をもらった。なぜなら、人が引き出しを開けたり閉じたりするとき、その人が引き出しの周囲にもたらず照度変化はさまざまなパターンが考えられるからである。実際、引き出しのタイプの推定精度は低かった。明らかに C3 に属する机と目覚まし時計はタイプの推定ができなかった。これらは状態遷移図によるモデル化には適していないだろう。以上の結果から、ある程度シンプルで複雑でない動きをするモノのタイプが参加者にはモデル化しやすく、その推定精度も十分高いことが分かった。

7. おわりに

本稿では、モノにセンサノードを添付するだけで、そのモノのタイプとモノに起こった状態変化を推定するフレームワーク Tag and Think を提案した。特に本稿では人の経験や実世界に関する知識によって作成されたモデルによりモノのタイプを推定する方法を提案し、その簡単な評価を行った。

参考文献

- [Allen 83] J. F. Allen, "Maintaining knowledge about temporal intervals," *Commun. of the ACM*, 26(11):832-843, 1983.
- [Bao 04] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," *Proc. PERSASIVE 2004*, pp. 1-17, 2004.
- [Beckmann 04] 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, 2004.
- [Freksa 91] C. Freksa, "Conceptual neighborhood and its role in temporal and spatial reasoning," *Proc. IMACS Workshop on Decision Support Systems and Qualitative Reasoning*, pp. 181-187, 1991.
- [Philipose 04] M. Philipose, K.P. Fishkin, and M. Perkowski, "Inferring activities from interactions with objects," *IEEE Pervasive computing*, 3:50-57, 2004.