

食材の重量変化を用いたユーザの手間を最小化する ストック食材管理支援システム

岡田 みなみ^{1,a)} 山本 景子^{1,b)} 倉本 到^{1,c)} 辻野 嘉宏^{1,d)}

概要：一般家庭の冷蔵庫には、牛乳や卵などいつも一定量以上保存しておきたいと思っている「ストック食材」がある。しかし、このストック食材を維持するために、在庫の量を全て覚えておくことは難しく、たとえ覚えていたとしても外出中に家族が使ったりして量変動することがある。冷蔵庫の在庫管理を行う研究には様々なものがあるが、一般家庭の冷蔵庫において自動で正確に在庫管理を行う手法は確立されていない。一方、この在庫管理をユーザがすべて手動で行うと手間が大きくなってしまう。そこで本稿では、冷蔵庫に出し入れされた物を重量変化を基に自動で推測し、その推測された候補の中からユーザが選択することで、ユーザにかかる負担をなるべく減らしたうえで正確な在庫管理を行うストック食材管理支援システムを提案する。提案システムのプロトタイプを作成し、実験を行った結果、提案システムはストック食材管理支援システムとして運用可能であることがわかった。

1. はじめに

1.1 問題意識

一般家庭の冷蔵庫にはさまざまな物が保存されている。その中には、牛乳や卵などのように、家庭で冷蔵庫を利用している人（ユーザ）がいつも一定量以上保存しておきたいと思っている食材がある。以降、このような食材のことを「ストック食材」と呼ぶ。このストック食材が少なくなったり無くなったりしたとき、ユーザは次に買い物に行ったときにそれを買い足す。しかし、ユーザが買い物に出かけたにもかかわらず、そのストック食材を買い損ねてしまうという問題が発生することがある。

この問題には、ユーザはストック食材の現在の残量がそもそもわかっていないことと、ストック食材の残量はわかっているが、「買い足さなければいけない」という事実を忘れていていることの2つの原因がある。本稿では、前者のストック食材の現在の残量がわからないという点に注目する。

ストック食材の現在の残量がわからないのは、そもそもユーザがストック食材の残量を常に覚えているわけではないということと、ユーザが一人でストック食材を利用して

おらず、知らないうちに量が変わってしまうことの2つの要因を含んでいる。前者は、一人でストック食材を利用しているため知らないうちにストック食材の残量が変動することはないにもかかわらず、ストック食材がどれ位あったかを忘れてしまった時に発生する。一方後者は、ストック食材を複数人で利用しており、母親の外出中に子どもがストック食材であるジュースを飲んでしまった場合などに発生する。

そこで本稿では、ユーザの代わりにストック食材の残量変化をリアルタイムに検出することで在庫管理を行う、ストック食材管理支援システムを提案する。

1.2 管理する食材の情報

ストック食材管理支援システムの実現のためには、冷蔵庫がリアルタイムでストック食材の残量変化を検出する必要がある。これを達成するためにシステムが管理しなければならない要素は次の2種である。

種類 ストック食材が何であるかを識別する情報（品名や写真など）

量 ストック食材の残量（重量など）

在庫管理を行うためにはどんなストック食材があるかという種類の情報が必要である。また、本稿の目的を達成するためには、ストック食材の有無だけでなく、それらが「あとどれくらい残っているか」という量の情報が重要となる。そして、図1に示すようにこの2要素を結びつけることで「何があとどれくらい残っているか」を管理するこ

¹ 京都工芸繊維大学
Kyoto Institute of Technology, Matsugasaki, Sakyo-ku,
Kyoto 606-8585, Japan

a) minami@hit.is.kit.ac.jp

b) kei@kit.ac.jp

c) kuramoto@hit.is.kit.ac.jp

d) tsujino@kit.ac.jp

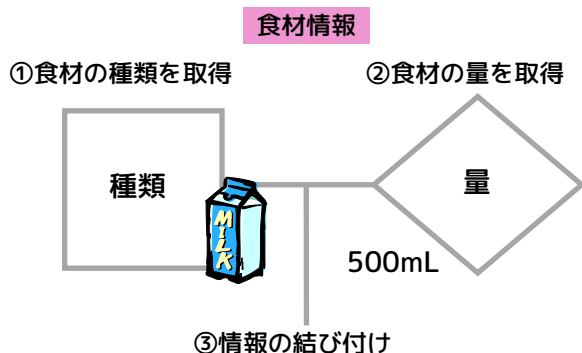


図 1 食材情報

とができる。本稿では、この2要素からなる組を結びつけて一つの食材情報とし、これを複数管理することを在庫管理と呼ぶ。

1.3 在庫管理における既存手法の問題点

ここでは、在庫管理を行う既存手法の問題点について述べる。

- 在庫管理を手動で行う手法

うちメモ [1] は在庫管理を行い、在庫が少なくなった時に通知する iPhone アプリケーションである。うちメモでは、ユーザが買ってきた食材のうち、管理したい任意の食材をアイコンの中から選択し、その食材の量を入力して在庫として登録する。また、その在庫を使うときは在庫量を手動で更新することで、在庫管理を行っている。つまり、食材の種類と量、そしてそれらの情報の結び付けをすべて手動で行う手法である。しかし、これらの情報の更新をストック食材を買ってきたり使う度にすべて手動で行うのは、ユーザにかかる負担が大きいと考えられる。さらに、うちメモで管理を行っている人以外が在庫を使った場合には在庫情報が更新されないで、うちメモの在庫情報と実際の庫内状況が異なる場合があるという問題点もある。

- 在庫管理を自動で行う手法

一方、食材の情報の取得から結び付けまでをすべて自動で行う既存手法もある。

加茂田ら [2] は、食材の持つ重さの特徴と、冷蔵庫内のどこにあるのかの位置情報を組み合わせた荷重特徴の変化を冷蔵庫内に取り付けた荷重センサで検出し、その変化から取り出された食材の種類を同定して在庫管理を行う手法を提案している。

しかし、この手法の精度は高いとはいえない。例えば、冷蔵庫内に食材が整然と並べられた実験環境下においても、隣り合った同じような重さの食材の種類の

同定に失敗している。一般家庭の冷蔵庫では、たくさんのおかずが積み重ねられていたりするなど、渾然と収納されていることが多いので、この手法を実運用するのは困難である。

他にも原田 [3] による RFID タグを食材に付加することで在庫管理を自動で行う手法もある。しかし、RFID タグを個々の食材に付与するのは、流通の時点で行うには業者に手間と金銭的なコストがかかり、個人で行うには手間がかかり過ぎるため、この手法は現実的ではない。

以上のように、在庫管理をすべて手動で行う手法では、正確に在庫管理ができる一方、ユーザにかかる負担が大きいという問題点と庫内情報が実際の庫内状況に同期されないという問題点がある。一方、在庫管理をすべて自動で行う手法では、情報の取得精度に依存してうまく在庫管理を行えない場合がある。また、RFID タグなどで食材一つ一つに食材情報を付与するのはコストの点から実現が難しい。

そこで本稿では、なるべくユーザに負担がかからないようにしたうえで正確に在庫管理を行うことを目指し、手動と自動とのバランスを取ることを考える。これを実現するために、情報の取得を自動で行い、結びつけのみを手動で行う手法を提案する。

2. 提案手法

提案手法では、図 2 に示すように、食材の種類を画像、量を重さとして自動で取得し、情報の結びつけの部分を手動で行うことで正確な在庫管理を行う。さらに、手動の結びつけを行う際、ユーザの負担が最小となるように、重さから取り出されたものを推測し、ユーザは提示された候補の画像からその食材を選択するだけで結びつけが行えるようにする。

このように、手動と自動をうまく組み合わせることで、ユーザの手間を最小化し、正確な在庫管理を行うことができる。

これらを実現するために必要な機能を次に示す。

- 画像によるストック食材の種類の自動取得
- 出し入れされる食材の重さの自動取得
- 出し入れされた食材候補の提示

以降では、これらの機能について詳述する。

2.1 画像によるストック食材の種類の自動取得

ストック食材が何であるかを識別する情報には、画像やテキスト情報などが考えられる。ユーザにかかる負担を最小化するためには、自動で取得でき、かつ普段どおり冷蔵庫にもものを出し入れする動作をできるだけ妨げずに取得できる情報が望ましい。

そこで提案手法では、ユーザが冷蔵庫にストック食材を入れる際、短時間冷蔵庫にストック食材をかざすことで、

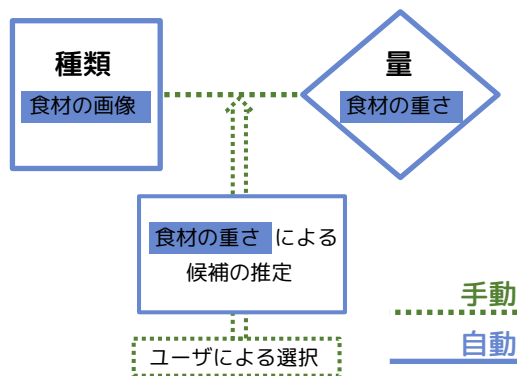


図 2 提案手法

それをシステムが検出してそのストック食材の画像を自動で取得することにより情報を取得する。このとき、かざす位置は冷蔵庫に食材を入れる自然な流れを妨げない位置にしておく。こうすることで、普段通り冷蔵庫を使う動作をほとんど妨げずに、食材の種類を自動で取得することができる。また、ストック食材でない食材の画像は必要ないため、ユーザーがストック食材でない食材を冷蔵庫に入れたときは冷蔵庫に食材はかざさなくてよい。このようにすることで、ユーザーにかかる負担を少なくしている。

2.2 出し入れされる食材の重さの自動取得

ストック食材の量情報として、提案手法では重さを用いる。システムが重さを自動取得する際に一つ一つの食材の重さが正確に取得できるように、本手法では複数のストック食材を同時に出し入れしてはいけないという制限をユーザーに課す。2.1 で述べたように、本手法ではストック食材を冷蔵庫に入れたときについてはストック食材を一つずつ冷蔵庫にかざしてから入れるという手順をとっているため、複数の食材を同時に冷蔵庫に入れるという動作自体が自然ではない。したがって、この制限による負荷はユーザーにとってそれほど大きくないと考えられる。

2.3 出し入れされた食材候補の提示

この機能により、ストック食材の種類情報と量情報との結び付けを実現する。

冷蔵庫から取り出される食材は、冷蔵庫内に入れたときと次に取り出すときとで重さは同じである。このことを利用して、システムは取り出された食材の重さとすでに登録されているストック食材の重さを比較する。そして、その差が小さいものを取り出されたストック食材であると推定する。しかし、食材の重さだけでは、同じ重さの食材が複数あった場合などに、推定される候補が一つとは限らない場合がある。そこで、推定した複数の候補をユーザーに提示

して取り出した食材を選択させる。このようにユーザーの手を介することで、システムは取り出されたストック食材の種類情報と量情報を正確に結び付けることができる。

一方、ストック食材を再度冷蔵庫に入れるときは、何を入れたのかの候補をシステムがユーザーに提示して、入れた食材を選択させる。このとき、システムは最近取り出された順にストック食材をソートし、ユーザーに提示する。

このように、ユーザーがどの食材を出し入れしたかを選択するとき、システムが候補を推定して表示するため、ユーザーはいくつかの候補の中から選ぶだけで良い。そのため、うちメモ [1] のように全食材一覧の中から探す必要がなく、ユーザーの負担は軽減される。

さらに、複数人で使う場合、冷蔵庫に入っている食材のうちどれがストック食材か知らないユーザーが存在する場合がある。その場合でも、取り出した食材からストック食材に重さが近い食材があればシステムがそれを提示するため、もし取り出した食材がストック食材であった場合、その食材がシステムによって提示されているので、それがストック食材であることに気づくことができる。このように、ユーザーが複数存在しても、冷蔵庫が使用される度に在庫は更新されるため、庫内在庫状況と実際の状況がずれるという問題点も解決できる。

ただし、先ほど食材は冷蔵庫内に入れたときと次に取り出すときで重さは同じであると述べたが、これには例外がある。例えば、卵を取り出す場合が挙げられる。卵をパックごと冷蔵庫に入れた場合、取り出すときはふつうパックごとではなく、パックの中から必要な個数だけ取り出す。このとき、卵を入れたときの重さはパック全体の重さとなるが、取り出すときの重さは取り出した卵の個数分だけなので、重さが一致せず、この手法では正しく推定することができない。以降、このようにストック食材を冷蔵庫に入れたときと出すときで重さの変化が異なり、出し入れされた食材の候補の推定が難しくなる問題を「卵問題」と呼ぶ。

卵問題が致命的になることを避けるため、食材を取り出したときには、重さにより推定されたストック食材のみ提示するだけでなく、ストック食材の一覧も同時に提示する。このとき、ユーザーはストック食材の一覧から自分で出した食材を探すことになるため、うちメモの場合と同じようにユーザーに負荷がかかるが、問題が起きるストック食材は多くないと考えられるため、負荷はそれほど大きくならないと考えられる。

3. ストック食材管理支援システム

ここでは、提案手法に基づき設計したストック食材管理支援システムについて述べる。まず、本章以降で使われる語句を定義する。

新規登録

登録されていない食材を新しくストック食材としてシ

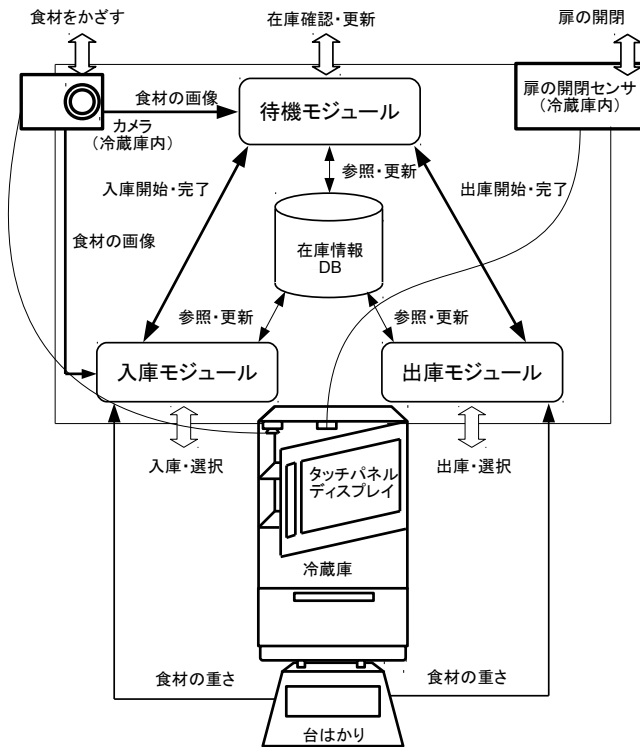


図 3 システム構成

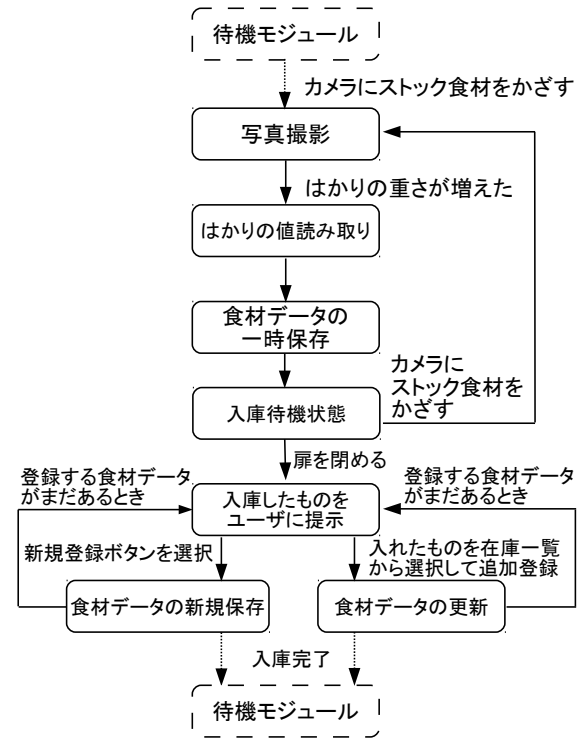


図 4 入庫管理モジュールの手続きの流れ

システムに登録すること

追加登録

同じストック食材をさらに買ってきてシステムに登録すること

冷蔵庫に戻す

ストック食材を取り出して使ったあと、再度冷蔵庫に入れること

3.1 システム構成

システムの構成図を図 3 に示す。

提案手法を実現するシステムは、冷蔵庫に加え、食材の重さを取得するはかり、ストック食材の画像を取得するカメラ、システムの表示及び操作を行うタッチパネルディスプレイ、扉が開閉したことを検出するセンサの 4 種の機器を有する。

提案システムは入庫管理モジュール、出庫管理モジュール、待機モジュールで構成される。以下にそのモジュールの機能を示す。

- 待機モジュール
 - － ストック食材の在庫一覧表示
 - － 残量が少なくなったストック食材の一覧表示
 - － 在庫の詳細の確認と修正
- 入庫管理モジュール
 - － 新規登録
 - － 追加登録
 - － 在庫の更新
- 出庫管理モジュール

－ ストック食材の出庫管理

－ ストック食材でない食材の出庫管理

なお、待機モジュール中にある「残量が少なくなったストック食材の一覧表示」と「在庫の詳細の確認と修正」は、ユーザビリティ向上のために設計している。入庫管理モジュールと出庫管理モジュールの動作の流れを図 4 と図 5 に示す。

カメラにストック食材がかざされると入庫管理モジュールが起動する。入庫管理モジュールでは、かざされた食材の写真の撮影と重さの読み取りを行い、一時的に保存する。そして、扉が閉められるまでは更に食材がカメラにかざされないかを待機する。ユーザがすべて食材を入れ終わり扉を閉めると、入庫した順にひとつずつ食材の画像をユーザに提示し、新規登録か追加登録かの選択を受け付けて食材の登録を行う。

出庫管理モジュールでは、はかりの重さの減少を検出してどの重さのものが取り出されたかを一時的に保存する。そして、入庫時同様扉が閉められるまで更に食材が取り出されないかを待機する。ユーザがすべて食材を出し終わり扉を閉めると、出庫した順に、何を取り出されたかを重さから推測して候補をユーザに提示し、ユーザの入力を受け付ける。候補から食材が選択された場合はそのストック食材が取り出されたと認識し食材データを更新する。ストック食材でないと選択された場合は、その重さデータを破棄する。

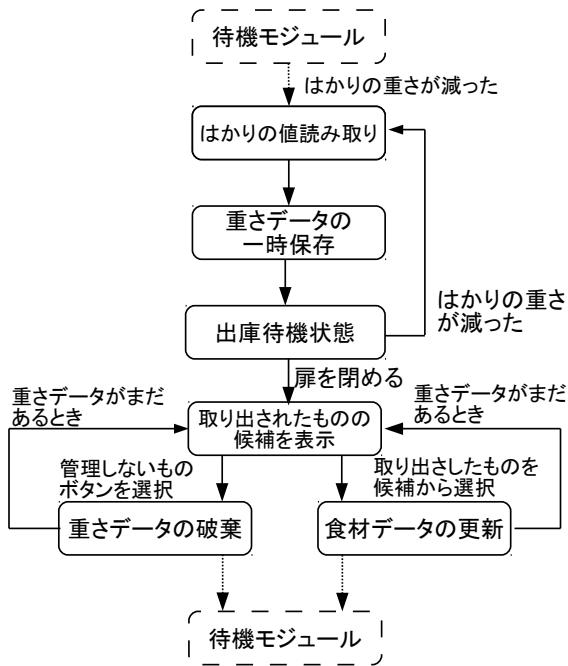


図 5 出庫管理モジュールの手続きの流れ

3.2 操作手順

ここでは、ユーザがどのように提案システムを使うかを述べる。ユーザに対する負担を最小限にするため、食材を入れるときと取り出すときで動作に一貫性を持たせ、かつストック食材ではない食材を入れるときには何も操作しなくて良いように設計した。なおミスに対する復帰動作は在庫編集画面から編集または在庫を削除するものとした。次に示す手順のうち、太字で示した部分が通常の冷蔵庫の使い方と比べた時に追加されている動作である。また、さらに番号にダッシュが付いているものについてはストック食材のときのみ行う動作である。

- (1) 食材を冷蔵庫に入れるとき
 - 1 冷蔵庫の扉を開ける
 - 1'** ストック食材をカメラにかざす
 - 2 食材を冷蔵庫の中に入れる（複数ある場合は手順 1', 2 を繰り返す）
 - 3 冷蔵庫の扉を閉める
 - 3'** 新規登録, 追加登録, または冷蔵庫に戻したかのいずれかを選択し確認する

(1' の回数だけ繰り返す。選択する順番は入れた食材順)
- (2) 食材を冷蔵庫から出すとき
 - 1 冷蔵庫の扉を開ける
 - 2 食材を出す（複数ある場合は繰り返す）
 - 3 冷蔵庫の扉を閉める
 - 4 ストック食材でないことを選択し, 確認する
 - 4'** 出したストック食材を一覧から選び, 確認する

(2 の回数だけ繰り返す。選択する順番は取り出した食

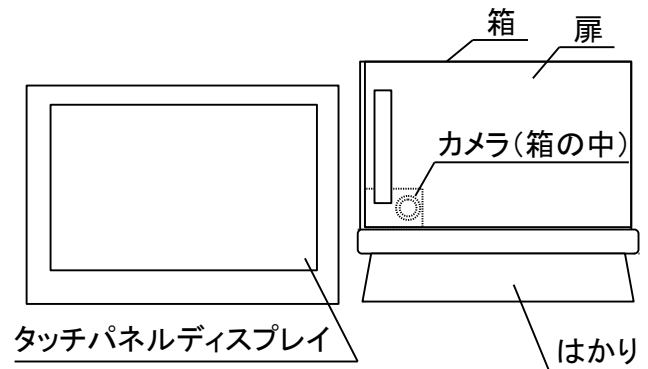


図 6 プロトタイプの構成

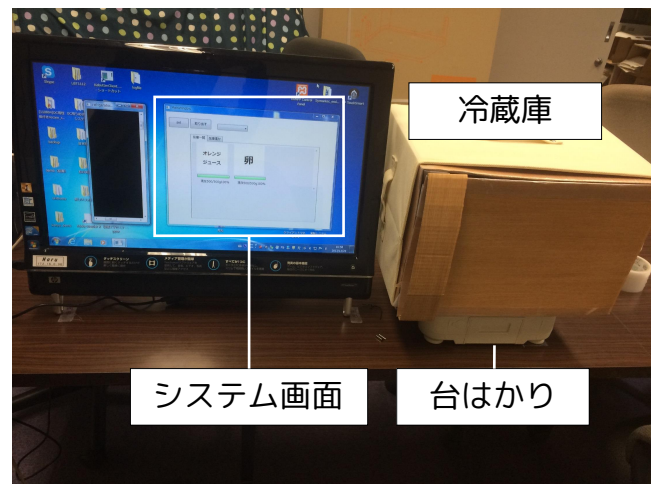


図 7 作成したプロトタイプ

材順)

4. 評価実験

4.1 目的

本実験では、日常生活で行う基本的な動作の中で提案システムがストック食材管理支援システムとして運用できるかどうかを検証する。そのために、以下の 2 点を検証する。

- ユーザが想定通りにシステムを使うことができるか
- 2.3 で述べた手法で卵問題が致命的な問題とならないかどうか

4.2 方法

4.2.1 実験の流れと設定

被験者は大学生及び大学院生の計 9 名である。

被験者には提案システムのプロトタイプ（図 6, 図 7）を使ってタスクを行わせる。実験者は Wizard of Oz 方式でプロトタイプを動作させる。その際、被験者には思考発話をさせ、その様子をビデオカメラで記録する。また、タスクを行っている際に被験者がどうしてよいかわからなくなった場合はギブアップを許す。タスクを行う前には、システムの使い方の説明、ストーリー付けされたタスクの説明、用意した食材の説明を行う。また、タスクが全て完了

表 1 被験者に与えたタスク

タスク番号	タスクの内容
1	牛乳（ストック食材）を 1 本新規登録する
2	お茶（ストック食材でない）を 1 本冷蔵庫に入れる
3	お茶（ストック食材でない）を出して使って戻す
4	牛乳（ストック食材）を出して使って戻す
5	牛乳（ストック食材）を 1 本追加登録する
6	りんご（ストック食材でない）とキウイ（ストック食材でない）を冷蔵庫に入れる
7	みかん（ストック食材）とバナナ（ストック食材）を新規登録する
8	みかん（ストック食材）とバナナ（ストック食材）を追加登録する
9	りんご（ストック食材でない）とキウイ（ストック食材でない）を出して使って戻す
10	みかん（ストック食材）とバナナ（ストック食材）を出して使って戻す
11	オレンジジュース（被験者は知らないがストック食材）を取り出して使って戻す
12	卵（ストック食材）を 2 個取り出す
13	りんごジュース（被験者は知らないがストック食材でない）を取り出して使って戻す

したあとに、インタビューを行う。

4.2.2 タスクの設定

被験者に表 1 に示す 13 個のタスクを与える。

このタスクのうち、1～10 は、次の項目を網羅的に満たすように作成した。

- 食材がストック食材か否か
- 1 回のタスクで出し入れする食材の数は 1 個か 2 個か
- 入れるか出すか、どちらも行うか
- 追加登録するかしないか

ただし、本実験では日常生活で行う基本的な動作を対象とするため、発生頻度が低いと考えられる、以下のタスクは行わなかった。

- スtock食材とそうでない食材が混ざったタスク
- 出し入れする食材の数が 3 個以上複合するタスク

タスク 11、13 では扱おうとする食材がストック食材なのかどうかを被験者が知らない場合でも、想定通りにシステムを使うことができるかどうかを確かめる。タスク 12 では、2.3 で挙げた卵問題が致命的な問題とならないかどうかを確かめる。

また、実験を行う際は、カウンタバランスをとるため、グループ A（前半 4 名 a～d）とグループ B（後半 5 名 e～i）に分けた。そして、グループ A には先述のタスク番号の昇順で、もう片方のグループ B ではタスク番号 2→1→4→3→5→7→8→6→10→9→13→12→11 の順で実験タスクを行わせた。

なお、食材にはそれがストック食材かどうかひと目で分かるように、その旨を示すラベルを各食材の見えやすい位

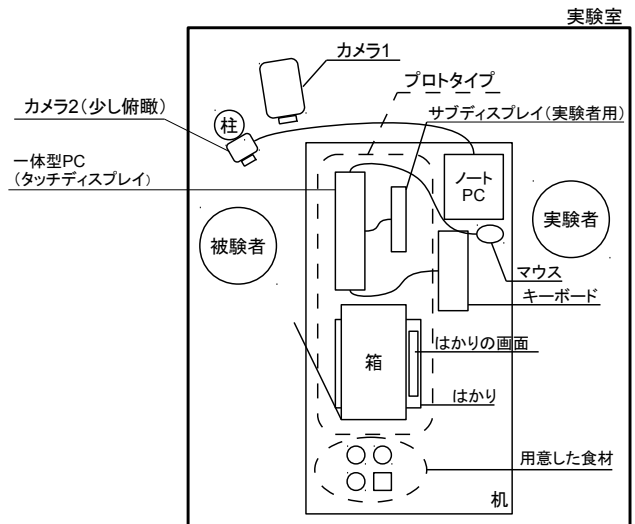


図 8 実験環境

置に表記した。ただし、タスク 11 と 13 で使う食材については、被験者にどちらか考えてもらうためにラベルは貼らなかった。

4.2.3 実験環境

実験は図 8 に示す環境で行った。カメラ 1 は被験者がタスクを行っている様子の撮影、カメラ 2 は被験者の手元と画面の動きをモニタリングするためのものである。実験を行うために図 6、図 7 に示すプロトタイプを作成し、机の上に設置した。ただし、このプロトタイプで画面に表示されるストックの画像は、タスクを行う際にカメラが撮影した実際の画像ではなく、予め実験者が事前に用意した画像を用いる。また、ストック食材がカメラにかざされた際の写真撮影、扉の開閉の検出、出し入れされた食材を識別する画像とその重さの読み取り、及びそれらのシステムへの入力の実験者が手動で行う。

4.3 評価手法

評価は、タスクのエラー回数の評価、思考発話のプロトコル分析とビデオ分析、及びインタビューにより行った。

インタビューにおける質問内容を次に示す。

- (1) ミスしたタスクや迷ったタスクについて使い方がわからないところはありませんでしたか？
- (2) 今回タスクを行っていて、普段冷蔵庫を使うときに比べて使いにくいところはありませんでしたか？
- (3) 他に何かこうしたほうが使いやすいなど意見があれば聞かせて下さい。

質問項目 1 は、思考発話の補足をするためのものであり、タスクを行っている最中に、実験者の想定とは違う動作をしたり、エラーが発生した部分があったタスクについてのみ行った。

表 2 タスク別結果比較 (○△×)

被験者\タスク	1	2	3	4	5	6	7	8	9	10	11	12	13
a	○	○	×	○	○	○	○	○	○	○	○	△	○
b	○	○	○	△	○	○	○	○	×	△	○	○	○
c	○	×	△	△	○	○	○	○	○	○	○	○	○
d	○	×	○	○	○	○	○	○	○	○	○	○	○
e	△	○	○	○	○	○	○	○	○	○	○	○	○
f	○	○	○	○	○	○	○	○	○	○	○	○	×
g	○	○	○	○	○	○	○	○	○	○	○	○	○
h	○	○	○	○	△	○	○	○	○	△	△	○	○
i	○	○	×	△	○	○	○	○	△	○	○	○	×
△回数合計	1/9	0/9	1/9	3/9	1/9	0/9	0/9	0/9	1/9	2/9	1/9	1/9	0/9
×回数合計	0/9	2/9	2/9	0/9	0/9	0/9	0/9	0/9	1/9	0/9	0/9	0/9	2/9

(○：ミスしなかった △：途中で無駄な操作をした ×：ミスをした)

4.4 結果と考察

4.4.1 タスク別エラー回数

各タスクのエラー回数の結果を表 2 に示す。

○印はミスなくタスクを達成できた、×印は途中でミスを 1 回以上したものである。ミスの定義は、タスクを行ったあとの状態が想定フローを行ったあとの状態と異なる場合、あるいはギブアップした場合である。ここで想定フローとは、3.2 で示した操作手順の通りにタスクを行った場合を指す。例えば、タスク 1 の想定フローは以下に示すようになる。

- (1) 冷蔵庫の扉を開ける
- (2) 牛乳をカメラにかざす
- (3) 牛乳を冷蔵庫の中に入れる
- (4) 冷蔵庫の扉を閉める
- (5) 新規登録を選択し確認する

複数の食材を同時に扱う場合の食材の扱う順番についてはどちらでも構わない。また、ミスではないが想定フローと比べて無駄な操作を起こしたものを△として表記する。ミス数は総タスク 117 タスク中 7 回という結果になった。

4.4.2 ユーザが想定通りにシステムを使うことができたか

表 2 に示したように、117 回のタスク遂行のうちミスがあったのは 7 回のみであったため、3.2 で示した提案システムの操作手順でユーザがほぼ想定通りにシステムを使うことができたといえる。このことから、提案システムはストック食材管理支援システムとして運用可能であることがわかった。

ただし、ミスが多かったタスクを見てみると、特にストック食材でない食材の一つ入れるだけのタスクでミスが発生しており、ストック食材でない食材を入れる操作にユーザのミスを招く原因があると考えられる。そこで、ミスがあったタスクごとに問題点を考察する。

タスク 2 でミスをした被験者は 2 名とも、ストック食材でない食材を入れる際に食材に貼られているラベルを見ずにカメラにかざしていた。発話やインタビューの回答から、

2 名ともお茶がストック食材かどうか確認するのを忘れ、管理する食材だと思い込んでいた、またはストック食材とそうでない食材に分かれていることを意識していなかったために間違えたと考えられる。

また、タスク 3 でミスした被験者は 2 名とも、ストック食材でないお茶を戻すときにカメラにかざしてしまっており、また、インタビューと発話の内容から、ストック食材でない食材を冷蔵庫に戻す際にはカメラにかざさないという操作をきちんと覚えられていないこともわかった。

これらのことから、エラーが発生した要因は次のようにまとめられる。

- ストック食材でない食材の動作が覚えにくい
- ストック食材とそうでないものを分けるのに慣れていない
- ストック食材とそうでないものを出し入れするときの動作に一貫性がない

次に、無駄な動作をしたタスクについて着目すると、それらはすべて、ストック食材を入れる際にカメラにかざすのを忘れて一旦冷蔵庫に入れたのち、かざし忘れたことに気づいてカメラにかざし直して登録していた。システムの仕様では、カメラにかざさずに冷蔵庫に入れてもストック食材でない食材として扱われることになっており、この時点での重量変化は無視されるようになっている。そのため、一旦かざさずに冷蔵庫に入れた後、取り出してカメラにかざすという行動をとったとしてもその時点ではエラーとならない。しかし、スムーズにタスクを行うのに支障が出ているため、上記の考察結果と同様に改善すべき部分であることがわかる。

また、ユーザのインタビューの回答において、この点に対する意見は以下の二つに大きく分かれた。

- ストック食材を出すときにはかざさないのに、入れるときだけかざすのは一貫性がなくてわかりにくい
- ストック食材を入れるときはかざすが、ストック食材でないものを入れるときにはかざさないのを混同して

しまう

更に、インタビューの設問3についての回答として、動作に一貫性がないことに対する指摘を受けた。これらのことから、提案システムの操作は一貫性が取れていないという問題があることがわかる。

現在の冷蔵庫の操作手順は、ストック食材とそうでないものの操作は、戻すときだけが同じ操作、入れるときはそれぞれ違う操作となっている。この操作は入れるときと出すときとでなるべく一貫性を持たせたいという考えで、ストック食材でない食材を出し入れする際にユーザに要求する操作の数をできるだけ減らしてユーザの負担を減らすことを狙って設計した。しかし、操作の数を減らしたために、ストック食材をカメラにかざすのを忘れていたことから、ユーザにとって覚えにくくミスを招きやすい設計になっていることがわかった。そこで、より一貫性のとれた操作にすることで、ユーザの混乱を招いたり、ミスが発生する回数を減らすことができる可能性がある。さらに、ストック食材でない食材を入れるときの動作が覚えにくいという問題も、一貫性を持たせることによって解決できる可能性があるため、今後改善が必要である。

さらに、インタビューの結果から被験者が日常で使うには問題だと感じたことについて考察する。まず、質問項目2において、普段冷蔵庫に食材を出し入れする順番は覚えていないという回答が得られたが、入れるときに関しては、写真が表示されるため、入れた順番を覚えている必要はない。また、食材を取り出すときは、重さで推定された取り出した食材の候補が表示されるため、ユーザは推定された食材の候補を見て、自分が取り出した食材が何であったか思い出しながら選択できる。さらに、重さが近い食材を複数取り出したが、取り出した順番がわからなくなり間違えた順番で選択したとしても、重量がほぼ同じものを選択していることになるので、正しく取り出された食材を選択し在庫量の減少分についてもほぼ正しく認識することができる。以上のことから、覚えていないことは問題にはならない。

4.4.3 卵問題が致命的な問題にならないか

卵問題が致命的な問題にならないかどうかを確認するタスクはタスク12であったが、表2よりこのタスクでミスをした被験者はいなかった。また、想定と比べて無駄な動作をした被験者aは、卵をパックごとに取り出したあとに、扉を開けたままでパックを戻してよいか、一旦扉をしめて取り出したことにしてから戻すかどうかについて悩んでいた。提案システムでは、この被験者が前者のようにパックをそのまま戻して登録を行った場合でも、後者のように一旦扉を閉めてパックごとに取り出したということを選択したのち、冷蔵庫に残った卵を戻して登録し直す場合でも正しく動作する。以上のことから、タスク12で行ったような卵を単体で出し入れするときに発生する卵問題は、提案シ

ステムを用いる際の致命的な問題にはならないといえる。

5. おわりに

本稿では、ユーザの手間を最小化するためのストック食材管理支援システムを提案した。このシステムは、少なくとも無くなったりしたりしたストック食材を買い忘れるという問題を解決するために、ストック食材の在庫管理に必要な食材の情報を定義し、なるべくユーザの手間を最小限にしたうえで正確な在庫管理を行う。具体的には、重さを使うことで冷蔵庫に何が出し入れされたかを推定して、ユーザが画面に表示された候補の中から出し入れした食材を選ぶことで在庫管理を行うシステムである。

提案システムの実現に向けて、システムの設計を行い、提案システムが冷蔵庫のストック食材管理支援システムとして運用可能な操作になっているかどうかを確認する実験を行った。その結果から、設計したシステムの操作手順は冷蔵庫のストック食材管理支援システムとして運用可能であることがわかった。しかし、実験タスクのうち、ユーザがミスを起こしたり想定と比べて無駄な動作を起こした箇所に着目すると、設計した操作は一貫性がとれていないという問題があることがわかった。

今後の課題として、ストック食材とそうでない食材を出し入れする操作にどのような一貫性をもたせるかの検討を行うことが挙げられる。

参考文献

- [1] YamaTatsu: うちメモ (online), 入手先 (<http://yamatatsu.jp/iphone/myinventory/index.html>) (2014.07.11).
- [2] 加茂田玲奈, 上田真由美, 船富卓哉, 飯山将晃, 美濃導彦, “食材管理のための荷重特徴を用いた食材同定,” 電子情報通信学会技術研究報告, vol.111, no.479, pp.181-186, March 2012.
- [3] 原田幹子, “RFID タグによる冷蔵庫管理制御に関する研究,” 高知工科大学学士學位論文, p.51, March 2010.