PAPER

Development and Evaluation of Near Real-Time Automated System for Measuring Consumption of Seasonings

Kazuaki NAKAMURA[†a)], **Takuya FUNATOMI**[††], **Atsushi HASHIMOTO**[†††], **Mayumi UEDA**[††††], ***Members***, ***and*** **Michihiko MINOH**[†††], ***Fellow***

SUMMARY The amount of seasonings used during food preparation is quite important information for modern people to enable them to cook delicious dishes as well as to take care for their health. In this paper, we propose a near real-time automated system for measuring and recording the amount of seasonings used during food preparation. Our proposed system is equipped with two devices: electronic scales and a camera. Seasoning bottles are basically placed on the electronic scales in the proposed system, and the scales continually measure the total weight of the bottles placed on them. When a chef uses a certain seasoning, he/she first picks up the bottle containing it from the scales, then adds the seasoning to a dish, and then returns the bottle to the scales. In this process, the chef's picking and returning actions are monitored by the camera. The consumed amount of each seasoning is calculated as the difference in weight between before and after it is used. We evaluated the performance of the proposed system with experiments in 301 trials in actual food preparation performed by seven participants. The results revealed that our system successfully measured the consumption of seasonings in 60.1% of all the trials.

key words: *seasonings, consumption measurement, real-time system, cooking support, successive computation*

1. Introduction

Nowadays people are increasingly focusing on their own health and so tend to cook meals by themselves. Various systems to support their cooking and eating activities have been widely studied that have aimed at such people. Examples have included recipe recommendation systems [1]–[4], real-time cooking support systems [5]–[8], and food calorie estimation systems [9]–[12]. In contrast to these systems, we have particularly focused on the use of seasonings. Seasonings are one of the most important factors in cooking because their amounts directly influence the taste of cooked meals as well as the intake of salt, sugar, and fat, which have significant effects on health status. We propose an automated system for numerically measuring the amount of seasonings consumed during food preparation in this paper. The proposed system runs in near real-time, so it can record not only the consumption of seasonings but also the time when each seasoning was used.

Our proposed system is beneficial for non-professional chefs such as homemakers in several ways. First, the system can provide chefs with an easy way of checking the usage of seasonings in close to real-time by continually monitoring and updating their measured consumption. This is quite helpful in avoiding excessive intake of seasonings.

Second, consumption data on seasonings obtained with the proposed system make it easier for a chef to recreate the taste of meals cooked by him/herself as well as by other chefs. In general, chefs recreate the taste of meals by the following two steps: (i) record the amount of seasonings used during food preparation for some meal and (ii) cook the meal using the same amount of seasonings recorded in step (i). However, recreating taste is not so easy for non-professional chefs, especially homemakers, because manually performing step (i) in daily food preparation is very annoying, so that most homemakers do not always record the accurate amount of seasonings they consumed. Although there are several commercially available devices useful for step (ii) such as digital spoon scale, they are not enough helpful for step (i) because chefs have to read the output value of such devices and record it in a notebook by hand. The proposed system release chefs from such burden of step (i) by automatically recording the used amount of seasonings as numerical data. Moreover, since the proposed system can record the time when each seasoning was used, it can provide helpful information for step (ii) by being combined with an existing cooking support system. For instance, it can show the order of addition of seasonings to chefs with proper timing based on the progress of cooking recognized by the cooking support system.

Third, the recorded consumption data can be used for analysis. It is a non-trivial problem for non-professional chefs as to the amount of seasonings that should be put into dishes from the aspect of palatability. Automated statistical analysis over massive amounts of consumption data including both positive ones (consumption data on well-cooked meals) and negative ones (consumption data on failed meals) would be very useful to solve this problem. Indeed, there have recently been a number of digital cooking recipes on the Internet. However, since most of these digital recipes have user-generated content, their description of the amount of seasonings has often been unclear and/or inaccurate. For instance, we can hardly understand the ac-

Manuscript received February 19, 2015.
Manuscript revised July 21, 2015.
Manuscript publicized September 7, 2015.

[†]The author is with Osaka University, Suita-shi, 565–0871 Japan.
[††]The author is with Nara Institute of Science and Technology, Ikoma-shi, 630–0192 Japan.
[†††]The authors are with Kyoto University, Kyoto-shi, 606–8501 Japan.
[††††]The author is with University of Marketing and Distribution Sciences, Kobe-shi, 651–2188 Japan.
a) E-mail: k-nakamura@comm.eng.osaka-u.ac.jp

DOI: 10.1587/transinf.2015EDP7072

curacy of the amount from a description like "a moderate amount of salt". Moreover, most digital recipes on the Internet involve positive data because few chefs would want to disclose recipes for meals in which they failed. Digital recipes on the Internet are not sufficiently suitable for statistical analyses for these reasons.

The proposed system is also beneficial for professional dietitians. Dietitians often check the amount of food intake by their patients to assess their dietary habits. However, the amount of food intake alone is insufficient to assess dietary habits because these are deeply related not only to the amount of food intake but also to those of seasonings, which can be automatically measured and recorded with the proposed system.

The main contribution of this paper can be summarized as follows. First, we developed a system particularly focusing on the use of seasonings that has attracted less attention from previous work despite its importance. Second, our system can fully-automatically recognize the kinds of seasonings consumed at each time during food preparation. This is discussed in Sect. 5.

The remainder of this paper is organized as follows. First, some environmental assumptions are introduced in Sect. 2. Section 3 theoretically discusses a method of measuring the consumption of seasonings in detail based on these assumptions. Section 4 explains how we implemented and experimentally evaluated an actual measuring system. Some related work is then reviewed in Sect. 5. Section 6 concludes this paper with a description of future work.

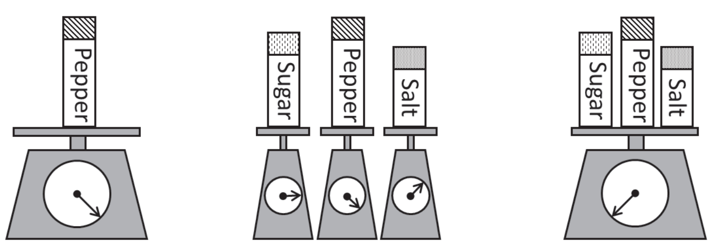
2. Environmental Assumptions

The consumption of seasonings can be calculated as the difference in weight between before and after they are used. We therefore aim to measure the weight of each seasoning at each time during food preparation. Note that there is no need to distinguish the weight of a bottle from that of its contents because the weight of the bottle is constant.

Suppose that there is only one seasoning bottle in a kitchen (see Fig. 1 (a)). The weight of the seasoning can easily be measured with electronic scales. This is also the case with two or more seasoning bottles if we can separately use electronic scales for each seasoning (see Fig. 1 (b)). However, a system like that in Fig. 1 (b) has several disadvantages in reality. First, it is too costly. Second, it lacks extensibility because we have to add one more set of electronic scales to the system when we want to use a new kind of seasoning. Third, it fails to measure the weight of seasonings when a chef picks up a seasoning bottle from one set of electronic scales and returns it to another set of scales. One possible solution for avoiding the disadvantage is attaching small load sensors to the bottom of each seasoning bottle. However, this solution also has a problem of cost. Moreover, problems of portability and measurement precision are raised in this case as described below. Load sensors attached to seasoning bottles should have a wireless communication device and enough capacity of a battery. They would not be light and small even if the sensors themselves are enough light and small. As for measurement precision, most of the existing small load sensors do not have enough precision. One of the state-of-the-art small load sensors [13] only has a precision of about 1g. However, the amount of several seasonings used in food preparation is often less than 1g (e.g. a touch of *pepper*/*salt*). These disadvantages of small load sensors may be solved in the future, but currently, using small load sensors is not a realistic solution. For these reasons, we aim at developing a system that can measure the weight of two or more seasonings with only one set of electronic scales (see Fig. 1 (c)).

There are three kinds of chef's actions in the use of seasonings: Picking up a seasoning bottle from electronic scales, adding seasoning to a dish, and returning the bottle to the scales. We refer to these three actions as *picking up*, *adding*, and *returning actions*. We assume that these three actions would always be performed in this order when a chef uses a seasoning, except for when two or more *adding actions* are performed between a *picking up* and a *returning action*. More specifically, we assume that a sequence of *picking up action* P_s, *adding action* A_s, and *returning action* R_s for seasoning s would always match the regular expression of $(P_s\ (A_s)*\ R_s)*$. This is a realistic assumption because all seasoning bottles are kept together in one place in most homes; we just have to place the electronic scales in that place.

Under the above assumptions, the consumption of seasonings is calculated after chef's *returning action*, and there



(a) One seasoning with one scale (b) Two or more seasonings with same number of scales (c) Two or more seasonings with only one scale

Fig. 1 Num. of scales used for measuring weight of seasonings. Although it is easier to measure weight of each seasoning with system like (b) than that with (c), it has some deficiencies. We therefore adopt system like (c).

is a certain time lag between the *returning action* and the corresponding *adding action*. We use the term "near real-time" to refer to this time lag in this paper.

3. Measurement of Consumption by Successively Solving Weight Equations

3.1 Notations

We will first introduce some notation used in this paper before discussing the method of measuring the weight of seasonings.

Let n denote the number of seasoning bottles in a kitchen, and let s_i denote the kind of i-th seasoning bottle for each $i \in \{1, \cdots, n\}$. For example, s_1 denotes *sugar*, s_2 denotes *salt*, and s_3 denotes *vinegar* if there are only these three kinds of seasonings in a kitchen. $\mathcal{S} = \{s_i \mid i = 1, \cdots, n\}$ denotes the whole set of seasonings in a kitchen, and its subset $\mathcal{S}(t) \subseteq \mathcal{S}$ denotes a set of seasonings placed on the electronic scales at (discrete) time t, where the t represents the lapse time since the chef started to prepare food. For all $s \in \mathcal{S}(t)$, let $w(s; t)$ be the weight of seasoning s and its bottle at time t, and let $W(t)$ be the summation of $w(s; t)$, i.e.,

$$W(t) = \sum_{s \in \mathcal{S}(t)} w(s; t) \tag{1}$$

for all $t \geq 0$. We will refer to the above Formula (1) as the *total weight equation* in the remainder of this paper.

Moreover, let $\hat{W}(t) = W(t) + \epsilon(t)$ denote the estimate of $W(t)$ measured with the electronic scales, where $\epsilon(t)$ is an error term. We will discuss this in Sect. 3.3.

3.2 Fundamental Formulation

As assumed in Sect. 2, a chef's *adding action* always follows his/her *picking up action*, and is always followed by his/her *returning action*. Hence, a change in $W(t)$ caused by each *picking up action* and *returning action* makes it possible to measure the weight of seasonings consumed with the corresponding *adding action*.

Suppose that only one seasoning bottle s_i is picked up from the electronic scales and no bottles are returned to the scales in time interval $[u, u+1]$. In this case,

$$\mathcal{S}(u+1) \cup \{s_i\} = \mathcal{S}(u) \tag{2}$$

is trivially satisfied and the remaining weight of any seasoning $s \in \mathcal{S}(u+1)$ remains unchanged during interval $[u, u+1]$ since the bottle for such seasoning s has not been moved. This can be formulated as

$$\forall s \in \mathcal{S}(u+1), \;\; w(s; u+1) = w(s; u) . \tag{3}$$

Simultaneously solving Formulas (1) and (3), the remaining weight of s_i at time u can be calculated as the difference between $W(u)$ and $W(u+1)$ because of

$$\begin{aligned} W(u) - W(u+1) &= \sum_{s \in \mathcal{S}(u)} w(s; u) - \sum_{s \in \mathcal{S}(u+1)} w(s; u+1) \\ &= w(s_i; u) + \sum_{s \in \mathcal{S}(u+1)} \{w(s; u) - w(s; u+1)\} \\ &= w(s_i; u) . \end{aligned} \tag{4}$$

Next, suppose that the same seasoning bottle, s_i, is returned to the electronic scales and no other bottles are moved in time interval $[v, v+1]$, where $v > u+1$. In this case,

$$\mathcal{S}(v+1) = \mathcal{S}(v) \cup \{s_i\} \tag{5}$$

is trivially satisfied and the remaining weight of any seasoning $s \in \mathcal{S}(v)$ remains unchanged during interval $[v, v+1]$, which can be formulated as

$$\forall s \in \mathcal{S}(v), \;\; w(s; v+1) = w(s; v) . \tag{6}$$

Simultaneously solving Formulas (1) and (6), the remaining weight of s_i at time $v+1$ can be calculated as the difference between $W(v)$ and $W(v+1)$ just like the case with Formula (4), i.e.,

$$W(v) - W(v+1) = -w(s_i; v+1) . \tag{7}$$

We will refer to the equations representing weight constancy between two successive periods like those in Formulas (3) and (6) as *weight transition equations* in the remainder of this paper. The remaining weight of each seasoning can be calculated at each time by simultaneously solving the *total weight equations* and *weight transition equations*, like those in Formulas (4) and (7). This is the fundamental principle of the proposed system. Using Formulas (4) and (7), the consumption of seasoning s_i can be calculated as

$$w(s_i; u) - w(s_i; v+1) = W(u) - W(u+1) + W(v) - W(v+1) . \tag{8}$$

Two sub-tasks should be done to calculate the remaining weight of each seasoning at each time based on the above principle:

- Reliably measure $W(t)$ on the electronic scales
- Calculate $\mathcal{S}(t)$ by monitoring each seasoning bottle

We will discuss a method of achieving each of these two sub-tasks in Sects. 3.3 and 3.4.

3.3 Reliable Measurement of $W(t)$

$W(t)$ can basically be measured as $\hat{W}(t)$ on the electronic scales at each time t, but the measured value, $\hat{W}(t)$, is not always equal to $W(t)$ because of error term $\epsilon(t)$. Concretely speaking, $|\epsilon(t)|$ becomes more than zero when a chef performs *picking up/returning actions*.

During a chef's *picking up/returning actions*, the electronic scales are not only forced by the seasoning bottles but also by the chef's hand. This force makes $\epsilon(t) > 0$, i.e., $\hat{W}(t) > W(t)$. Moreover, $W(t)$ changes instantaneously at a certain instant, whereas $\hat{W}(t)$ follows $W(t)$ with some delay. This makes $\epsilon(t) < 0$ in the case of *returning actions* and $\epsilon(t) > 0$ in the case of *picking up actions*. Some examples of these errors have been given in Figs. 2 and 3. We can-

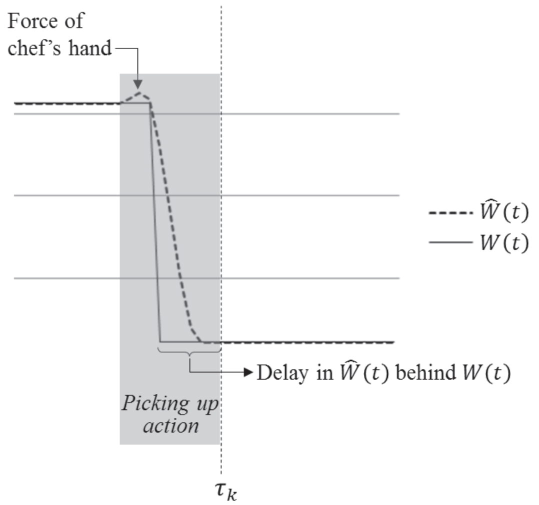


Fig. 2 Error between $W(t)$ and $\hat{W}(t)$ caused by chef's *picking up action*. $\epsilon(t) = \hat{W}(t) - W(t)$ becomes more than zero due to force of chef's hand and delay in $\hat{W}(t)$ behind $W(t)$.

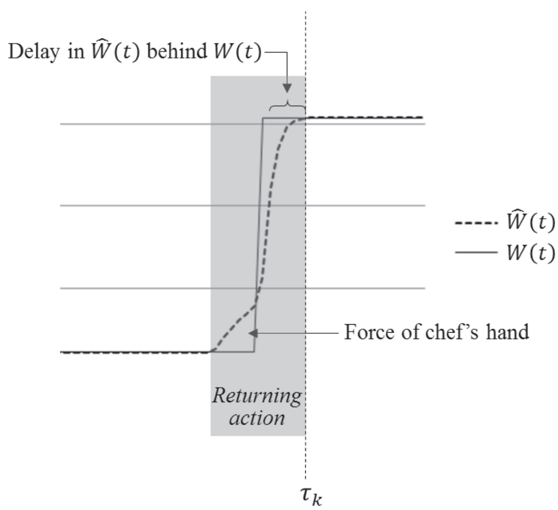


Fig. 3 Error between $W(t)$ and $\hat{W}(t)$ caused by chef's *returning action*. $\epsilon(t) = \hat{W}(t) - W(t)$ first becomes more than zero due to force of chef's hand, and then becomes less than zero due to delay in $\hat{W}(t)$ behind $W(t)$.

not obtain reliable estimates of $W(t)$ during a chef's *picking up/returning actions* due to errors.

As seen in Figs. 2 and 3, the value of $\hat{W}(t)$ is usually stable, but it starts to change slowly when a chef performs a *picking up/returning action*, and a short time later, the value stabilizes again. We refer to this kind of weight change as a *Weight Change due to Chef's Actions* (*WeCCA*) *event*. Note that two or more *picking up/returning actions* can be included in a single *WeCCA event* as shown in Fig. 4. Let τ_k denote the time a k-th *WeCCA event* ends. The value of $\epsilon(t)$ at $t = \tau_k$, i.e., $\epsilon(\tau_k)$, is expected to be zero. In other words, $\hat{W}(\tau_k)$ is expected to be reliable. Hence, we only use a set of values $\{\hat{W}(\tau_k) \mid k = 1, 2, \cdots\}$ for measuring the weight of each seasoning and ignore all other $\hat{W}(t)$ in our proposed system. This enables the system to obtain a reliable estimate of $W(t)$. Note that this approach does not allow the system

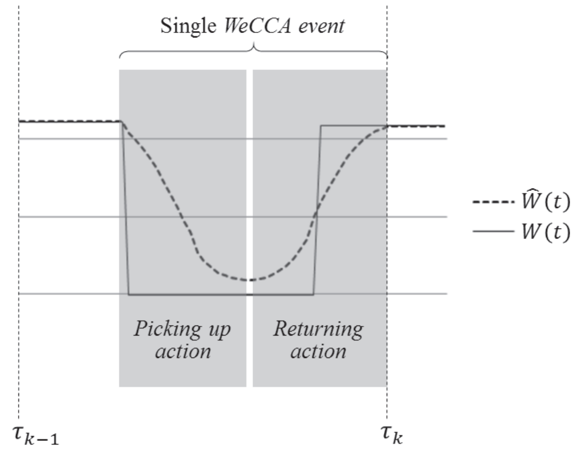


Fig. 4 *WeCCA event* sometimes includes two or more *picking up/returning actions*. Value of $\hat{W}(t)$ first decreases due to chef's *picking up action*, but a few seconds later, it increases without stabilizing in this figure. This is because seasoning s that was picked up was returned to electronic scales before value of $\hat{W}(t)$ stabilized. Here, *WeCCA event* includes not only *picking up action* P_s but also *returning action* R_s for same seasoning s.

Algorithm 1 Finding time τ_k when k-th *WeCCA event* ends

Require: $\tau_{k-1} \geq 0 \ \ (k \geq 1)$
  $t = \tau_{k-1} + 1$;
  loop
    if $\mathrm{stable}(\hat{W}; t-1) = 0$ **and** $\mathrm{stable}(\hat{W}; t) = 1$ **then**
      $\tau_k \leftarrow t$;
      return τ_k ;
    end if
    $t \leftarrow t + 1$;
  end loop

to calculate $w(s; t)$ for any $t \neq \tau_k$. Therefore, we only calculate $w(s; \tau_k)$ for each $k \in \{1, 2, \cdots\}$ and interpolate $w(s; t)$ as

$$w(s; t) = w(s; \tau_k) \quad (\tau_k \leq t < \tau_{k+1}) \tag{9}$$

for each $s \in \mathcal{S}(\tau_k)$. This is a reasonable interpolation because the remaining weight $w(s; t)$ does not change without a chef's *picking up/returning actions*, as was assumed in Sect. 2.

The time τ_k when each *WeCCA event* ends is recursively found with Algorithm 1 based on the stability of $\hat{W}(t)$, where τ_0 is defined as 0 to simplify the formulation. The indicator function, $\mathrm{stable}(f; t)$, in Algorithm 1, which indicates whether a discrete-time signal $f(x)$ is stable at $x = t$ or not, is defined as

$$\mathrm{stable}(f; t) = \begin{cases} 1 & \text{if } \displaystyle\sum_{x=t-\delta}^{t} |f(x) - f(x-1)| \leq d \\ 0 & \text{otherwize} \end{cases}, \tag{10}$$

where d and δ are some positive constants. This indicator function regards signal $f(x)$ as stable at $x = t$ if and only if a variation of $f(x)$ during $[t-\delta, t]$ is equal to or less than a certain threshold, d.

3.4 Incremental Calculation of $\mathcal{S}(t)$

3.4.1 Vision-Based Calculation

Since we only use $\hat{W}(\tau_k)$ and ignore $\hat{W}(t)$ for all $t \neq \tau_k$ as mentioned in the previous section, we do not have to calculate $\mathcal{S}(t)$ at time $t \neq \tau_k$; only $\{\mathcal{S}(\tau_k) \mid k = 1, 2, \cdots\}$ should be calculated. Each $\mathcal{S}(\tau_k)$ can be incrementally calculated as

$$\mathcal{S}(\tau_k) = [\mathcal{S}(\tau_{k-1}) \setminus \mathcal{P}(k)] \cup \mathcal{R}(k) \,, \tag{11}$$

where $\mathcal{P}(k)$ represents a set of seasonings picked up from the electronic scales in time interval $[\tau_{k-1}, \tau_k]$ and $\mathcal{R}(k)$ represents a set of those returned to the electronic scales in the same interval. We therefore describe a method of calculating $\mathcal{P}(k)$ and $\mathcal{R}(k)$ in this section.

Here, we will introduce an indicator function, $\mathrm{onScale}(s;t)$, which indicates the presence or absence of seasoning s on the electronic scales at time t to formulate a method of calculating $\mathcal{P}(k)$ and $\mathcal{R}(k)$, i.e.,

$$\mathrm{onScale}(s;t) = \begin{cases} 1 & \text{if } s \text{ is on the scales at time } t \\ 0 & \text{otherwise} \end{cases} . \tag{12}$$

Two sets, $\mathcal{P}(k)$ and $\mathcal{R}(k)$, can be calculated using the above $\mathrm{onScale}(s;t)$ as

$$\begin{aligned} \mathcal{P}(k) = \{ s \mid & \mathrm{onScale}(s;t-1) = 1, \\ & \mathrm{onScale}(s;t) = 0,\ \tau_{k-1} \leq t < \tau_k \} \end{aligned} \tag{13}$$

and

$$\begin{aligned} \mathcal{R}(k) = \{ s \mid & \mathrm{onScale}(s;t-1) = 0, \\ & \mathrm{onScale}(s;t) = 1,\ \tau_{k-1} \leq t < \tau_k \} , \end{aligned} \tag{14}$$

except for the case where *returning action* R_s for seasoning s and *picking up action* P_s for the same s are both performed in interval $[\tau_{k-1}, \tau_k]$ in this order. In this exceptional case, the remaining weight $w(s;t)$ can never be reliably obtained for all $t \in [\tau_{k-1}, \tau_k]$, even in the interval after R_s is performed and before P_s is performed. We therefore ignore R_s and P_s in this case, regarding seasoning s as not being on the electronic scales during $[\tau_{k-1}, \tau_k]$. The actual procedure for calculating $\mathcal{P}(k)$ and $\mathcal{R}(k)$ is given in Algorithm 2.

The problem now boils down to how to compute $\mathrm{onScale}(s;t)$ for each s and t. Unfortunately, $\hat{W}(t)$ obtained from the electronic scales has insufficient information on whether each seasoning s is on the scales or not at each time t. We therefore add another device, a camera, to the proposed system. The proposed system has a camera installed above the electronic scales, which shoots the tops of seasoning bottles on the scales at any time, as outlined in Fig. 5. On the other hand, a marker is attached to the tops of individual seasoning bottles so that different kinds of seasonings have different markers. The markers can be observed on camera images if and only if a corresponding bottle is placed on the electronic scales. Detecting the markers from camera images by utilizing vision techniques provides useful information for the system to compute $\mathrm{onScale}(s;t)$.

Algorithm 2 Calculating set of picked up seasonings $\mathcal{P}(k)$ and that of returned seasonings $\mathcal{R}(k)$ in time interval $[\tau_{k-1}, \tau_k]$

Require: $\tau_{k-1} \geq 0,\ \tau_k \geq \tau_{k-1}\ \ (k \geq 1)$
  $\mathcal{P}(k) \leftarrow \emptyset$;
  $\mathcal{R}(k) \leftarrow \emptyset$;
  $t \leftarrow \tau_{k-1} + 1$;
  while $t \leq \tau_k$ **do**
    for $i = 1$ **to** n **do**
      if $\mathrm{onScale}(\mathrm{s}_i; t-1) = 1$ **and** $\mathrm{onScale}(\mathrm{s}_i; t) = 0$ **then**
        if $\mathrm{s}_i \in \mathcal{R}(k)$ **then**
          $\mathcal{R}(k) \leftarrow \mathcal{R}(k) \setminus \{\mathrm{s}_i\}$;
        else
          $\mathcal{P}(k) \leftarrow \mathcal{P}(k) \cup \{\mathrm{s}_i\}$;
        end if
      else if $\mathrm{onScale}(\mathrm{s}_i; t-1) = 0$ **and** $\mathrm{onScale}(\mathrm{s}_i; t) = 1$ **then**
        $\mathcal{R}(k) \leftarrow \mathcal{R}(k) \cup \{\mathrm{s}_i\}$;
      end if
    end for
    $t \leftarrow t + 1$;
  end while
  return $\mathcal{P}(k)$, $\mathcal{R}(k)$;

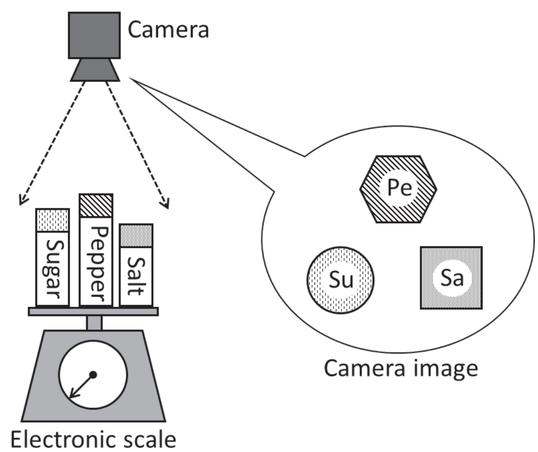


Fig. 5 Camera-based monitoring of seasoning bottles with markers. Each marker can be detected on camera images if and only if corresponding bottle is placed on scales.

We introduce another indicator function, $\mathrm{detected}(s;t)$, that indicates whether the marker of seasoning s is detected on camera images or not at time t, i.e.,

$$\mathrm{detected}(s;t) = \begin{cases} 1 & \text{if } m(s) \text{ is detected at time } t \\ 0 & \text{otherwise} \end{cases} , \tag{15}$$

where $m(s)$ is a marker attached to the bottle for s. The computation of $\mathrm{detected}(s;t)$ can be achieved by applying object detection and tracking techniques [14]–[17] to $m(s)$, and $\mathrm{onScale}(s;t)$ is naively computed as $\mathrm{onScale}(s;t) = \mathrm{detected}(s;t)$. However, since the detection and tracking of $m(s)$ may sometimes fail due to a variety of illumination conditions and occluded markers, the above naive approach cannot accurately compute $\mathrm{onScale}(s;t)$. To deal with such failures, we integrate the results from marker detection of θ continuous frames and compute $\mathrm{onScale}(s;t)$ as

$$\mathrm{onScale}(s;t)=\begin{cases}1 & \text{if } \sum_{x=t-\theta+1}^{t}\mathrm{detected}(s;x)=\theta\\ 0 & \text{if } \sum_{x=t-\theta+1}^{t}\mathrm{detected}(s;x)=0\\ \mathrm{onScale}(s;t-1) & \text{otherwise}\end{cases}. \quad (16)$$

Note that we assume $\mathrm{onScale}(s;0)$ is given for each s and compute $\mathrm{onScale}(s;t) = \mathrm{onScale}(s;0)$ in cases with $t < \theta$. This assumption is easily satisfied by recording $\mathrm{onScale}(s;q)$ at the time q when each food preparation opportunity ends and using it as $\mathrm{onScale}(s;0)$ for the next opportunity.

3.4.2 Error Correction with Weight Information

The method described in the previous section could miscalculate $\mathcal{P}(k)$ and $\mathcal{R}(k)$ when markers were falsely detected or missed for more than θ time steps. Fortunately, some kinds of miscalculations can be corrected by combining visual information provided by the camera and weight information provided by the electronic scales, i.e., $\hat{W}(t)$.

Let $t^{\mathrm{detected}}(a)$ be the time when a *picking up/returning action* a was detected. The difference between τ_k and $t^{\mathrm{detected}}(a)$ is generally small for any $a \in [\mathcal{P}(k) \cup \mathcal{R}(k)]$ because *picking up/returning actions* are always accompanied by a change in $\hat{W}(t)$, as shown in Figs. 2 and 3. Hence, a large difference between τ_k and $t^{\mathrm{detected}}(a)$ informs the system that action a was actually not performed even if the corresponding marker was falsely detected or missed. In this case, we remove such action a from $\mathcal{P}(k)$ and $\mathcal{R}(k)$.

3.5 General Formulation

Based on the discussions in Sects. 3.3 and 3.4, *total weight equation* (1) is now rewritten as

$$\hat{W}(\tau_k) = \sum_{s\in\mathcal{S}(\tau_k)} w(s;\tau_k) \quad (17)$$

for all $k \geq 0$. Similarly, *weight transition equations* (3) and (6) are generalized as

$$\forall s \in [\mathcal{S}(\tau_{k-1}) \setminus \mathcal{P}(k)],\quad w(s;\tau_k) = w(s;\tau_{k-1}) \quad (18)$$

for all $k \geq 1$. Combining Formulas (17) and (18) with Formula (11), the relation expressed by Formulas (4) and (7) are also generalized as

$$\begin{aligned}\hat{W}(\tau_{k-1}) - \hat{W}(\tau_k) &= \sum_{s\in\mathcal{S}(\tau_{k-1})} w(s;\tau_{k-1}) - \sum_{s\in\mathcal{S}(\tau_k)} w(s;\tau_k)\\ &= \sum_{s\in\mathcal{P}(k)} w(s;\tau_{k-1}) - \sum_{s\in\mathcal{R}(k)} w(s;\tau_k)\,. \quad (19)\end{aligned}$$

These equations cannot always be solved separately since they generally have two or more unknown variables, but they can be simultaneously solved by being combined with one another for different k. According to this notion, the proposed system successively calculates the remaining weight $w(s;\tau_k)$ of each seasoning s with Algorithm 3. As shown in Algorithm 3, the system removes all the weight equations whose unknown variables have been totally solved in each iteration. When a part of unknown variables have been solved for a certain weight equation, the system contracts the equation by combining its constant term with the solved variables into a new single constant term. Moreover, the system actually adds equation (19) into $\mathcal{E}$ instead of equations (17) and (18) in each iteration by pre-combining them. These are useful for avoiding redundancy of variables and equations in order to reduce the computational cost.

Algorithm 3 Successively calculating remaining weight $w(s;\tau_k)$ of each seasoning s

```
Require: S(τ0) = {s|s ∈ S, onScale(s; τ0) = 1}
  Measure Ŵ(τ0) using the electronic scales ;
  Initialize a set of equations E as E ← ∅ ;
  k ← 1 ;
  loop
    Detect τk by Algorithm 1 ;
    Calculate P(k) and R(k) by Algorithm 2 ;
    Calculate S(τk) based on Equation (11) ;
    Measure Ŵ(τk) using the electronic scale ;
    Create a total weight equation (17) for current k, referred to as ek ;
    Create a set of weight transition equations (18), referred to as Ek ;
    E ← E ∪ {ek} ∪ Ek ;
    Simultaneously solve as many e ∈ E as possible.
    for all e ∈ E do
      if e was solved then
        Remove e from E ;
      else if e has become unsolvable then
        Provide some instruction to the chef in order to solve e
        (but do nothing in the current version of the proposed system) ;
      end if
    end for
    k ← k + 1 ;
  end loop
```

Let $\tau^{\mathrm{picked}}(s)$ be the time right before s was picked up from the electronic scales, and let $\tau^{\mathrm{returned}}(s)$ be the time right after s was returned to the electronic scales. Using $w\left(s;\tau^{\mathrm{picked}}(s)\right)$ and $w\left(s;\tau^{\mathrm{returned}}(s)\right)$ calculated with Algorithm 3, the consumption of each seasoning s can be measured as

$$w\left(s;\tau^{\mathrm{picked}}(s)\right) - w\left(s;\tau^{\mathrm{returned}}(s)\right)\,. \quad (20)$$

3.6 Limitation

Algorithm 3 has a limitation in that it cannot measure several $w(s;\tau_k)$ in a specific case. Suppose that two seasonings s_i and s_j were simultaneously returned to the electronic scales in time interval $[\tau_{k-1},\tau_k]$ and then simultaneously picked up from the scales in time interval $[\tau_k,\tau_{k+1}]$. In this case, the following two equations are obtained according to Formula (19):

$$\hat{W}(\tau_k) - \hat{W}(\tau_{k-1}) = \quad w(s_i;\tau_k) + w(s_j;\tau_k) \text{ and} \quad (21)$$

$$\hat{W}(\tau_{k+1}) - \hat{W}(\tau_k) = -w(s_i;\tau_k) - w(s_j;\tau_k)\,. \quad (22)$$

However, each of the two equations is linearly dependent on the other. Moreover, any additional equations containing $w(s_i;\tau_k)$ or $w(s_j;\tau_k)$ cannot be obtained after time τ_{k+1} because the picked up seasoning is generally added to a dish before being returned, which brings about $w(s_i;\tau_l) \neq w(s_i;\tau_k)$ and $w(s_j;\tau_l) \neq w(s_j;\tau_k)$ for any $l > k$. Hence, in this case, $w(s_i;\tau_k)$ and $w(s_j;\tau_k)$ have become unsolvable. We refer to these kinds of unsolvable $w(s;\tau_k)$ as *unsolvable weights*.

Consumption of seasoning s whose initial weight $w(s;\tau_0)$ has become unsolvable is obviously immeasurable for the proposed system. The system cannot deal with this problem alone. Fortunately, in such case, the system can provide some direction to the chef and receive his/her cooperation because it runs in near real-time (or on-line). One possible direction is to instruct the chef to return the seasoning bottles to the scale one-by-one whose remaining weights have become unsolvable. If the system works only off-line, it cannot provide such instruction to the chef. This is an advantage of near real-time measurement. However, since the problem of *unsolvable weights* would occur less frequently, the current version of the proposed system does nothing to solve the problem, which we intend to address in future work.

4. Implementation and Evaluation

4.1 Implementation of System

We implemented an actual measuring system for the consumption of seasonings based on the method proposed in Sect. 3. There is an overview of the implemented system in Fig. 6, which consists of a camera, electronic scales, and a laptop PC.

- **Camera**
  We used the *Logitech HD Webcam C615* as the camera. The implemented system captured a sequence of images with a resolution of VGA (640 × 480 pixels) and a frame rate of 15 fps by using this product. Figure 7 (a) is an example photograph of the captured images. The black square patterns in Fig. 7 (a) are markers attached to the tops of seasoning bottles. Figure 7 (b) shows one of these markers in detail. This kind of marker is proposed and widely used in ARToolKit [18]. Machine-readable markers are generally visually annoying to humans, but if marker patterns are also readable by humans like characters, they are not annoying. ARToolKit markers can be used in this way. Therefore, we also used ARToolKit for detecting the markers and computing detected$(s;t)$ for each s and t. Parameter θ in Formula (16) was roughly tuned as 20 time steps (~1.3s) without using any sophisticated tuning approaches.
- **Electronic Scales**
  We used *Amidia TX4202N* produced by *Shimadzu Corp.*, which has a weighing capacity of 4,200g and a resolution of 0.01g as the electronic scales. In fact, the required resolution of electronic scales is different for each seasoning. However, too finer resolution causes no serious problems in automatic measurement, unlike too coarser resolution. Hence, we selected the electronic scales that are not so expensive yet have finer resolution. The selected product, *Amidia TX4202N*, can measure $\hat{W}(t)$ about 9 times per second and send them to a laptop PC with an RS-232C serial communication interface. In addition, the product also has the capability of outputting a bit flag that indicates whether the measured $\hat{W}(t)$ is stable or not for each t. We therefore determined the stability of $\hat{W}(t)$, i.e., stable$(\hat{W};t)$, based on this bit flag instead of Formula (10) in the implemented system. Note that the bit flag was approximately emulated by Formula (10) with the parameters of $\delta = 13$ time steps (~1.4s) and $d = 0.06$g in our experiments.
- **Laptop PC**
  We used a *Vostro 3360 Laptop* with Windows 7 produced by *Dell Inc.*, which had a Intel Core i5-3337U processor and 4 GB of RAM, as the laptop PC to process camera images and sequences of weight data. The camera was connected to this PC with a USB 2.0 inter-

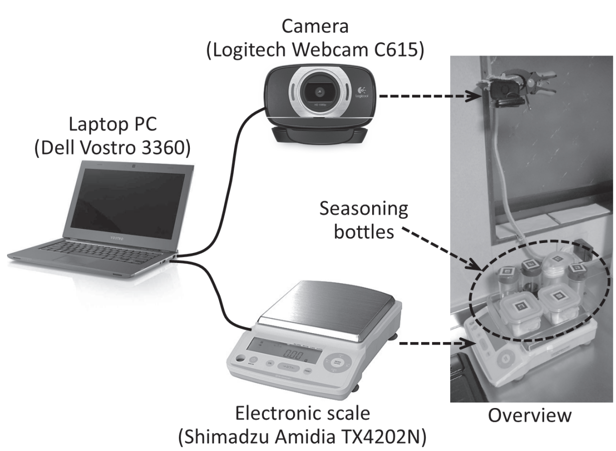


Fig. 6 Overview of implemented system consisting of camera, electronic scales, and laptop PC.

(a) Example of captured images (b) Marker pattern

Fig. 7 Example of marker pattern of ARToolKit and its appearance on camera images. This marker is machine-readable but less annoying for humans because of its character-based pattern. The character in (b) means "vinegar" in Japanese.

face and the electronic scales were connected to it with an RS-232C serial interface.

Our system currently assumes $\mathcal{S}(\tau_0) = \mathcal{S}(0) = \emptyset$ to simplify implementation.

4.2 Experimental Setting for Evaluation

We conducted an experiment to evaluate the performance of the implemented system, in which seven participants (five women and two men: 20–65 years old) who were familiar with food preparation participated as chefs. We installed the implemented system in the kitchen of each participant's home and instructed him/her to cook some meals by using the system. The system particularly focused on *sugar, salt, vinegar, soy sauce, pepper*, and *oil* in this experiment, and measured the amount of these six kinds of seasonings used when the participants prepared meals.

Each participant actually carried out 50–60 trials in preparing food for this experiment. The seven participants carried out a total of 383 trials in food preparation. However, the assumption of $\mathcal{S}(\tau_0) = \emptyset$ was not satisfied in 82 out of the total of 383 trials. This was especially noticeable in the trials by participant 3. Therefore, we excluded these 82 trials from the collected data set and used the remaining 301 trials to evaluate the performance of the implemented system.

4.3 Results and Discussion

Table 1 summarizes an example of trials in which the consumption of seasonings was successfully measured. Figure 8 shows the output value of the electronic scales along the time axis in the same example. This example demonstrates that the implemented system could automatically measure the consumption of seasonings, even if a chef performed two or more actions simultaneously like the first and

Table 1 Example of food preparation trials in which consumption of seasonings was successfully measured. In this example, amount of seasonings consumed during chef's *adding action* was always measured right after corresponding *returning action*. This means implemented system measured consumption of seasonings close to that in real-time.

k	Chef's action(s)	$\hat{W}(\tau_k)$	$w(oil;\tau_k)$	$w(pepper;\tau_k)$	$w(salt;\tau_k)$	Remarks
0	(start cooking)	0.00	-	-	-	
1	return *oil* return *pepper* return *salt*	414.47	unknown (w_a)	unknown (w_b)	unknown (w_c)	$w_a + w_b + w_c = 414.47$
2	pick up *oil*	155.50	-	unknown (w_b)	unknown (w_c)	$w_a = 414.47 - 155.50 = \mathbf{258.97}$ $w_b + w_c = 155.50$
3	return *oil*	412.29	412.29 – 155.50 = **256.79**	unknown (w_b)	unknown (w_c)	**2.18(g) of *oil* was consumed**
4	pick up *oil*	155.51	-	unknown (w_b)	unknown (w_c)	
5	return *oil*	410.33	410.33 – 155.51 = **254.82**	unknown (w_b)	unknown (w_c)	**1.97(g) of *oil* was consumed**
6	pick up *salt*	317.86	254.82	**63.03**	-	$w_c = 410.33 - 317.86 = \mathbf{92.47}$ $w_b = 155.50 - 92.47 = \mathbf{63.03}$
7	return *salt* pick up *pepper*	346.48	254.82	-	346.48 – (317.86 – 63.03) = **91.65**	**0.82(g) of *salt* was consumed**
8	return *pepper*	409.43	254.82	409.43 – 346.48 = **62.95**	91.65	**0.08(g) of *pepper* was consumed**

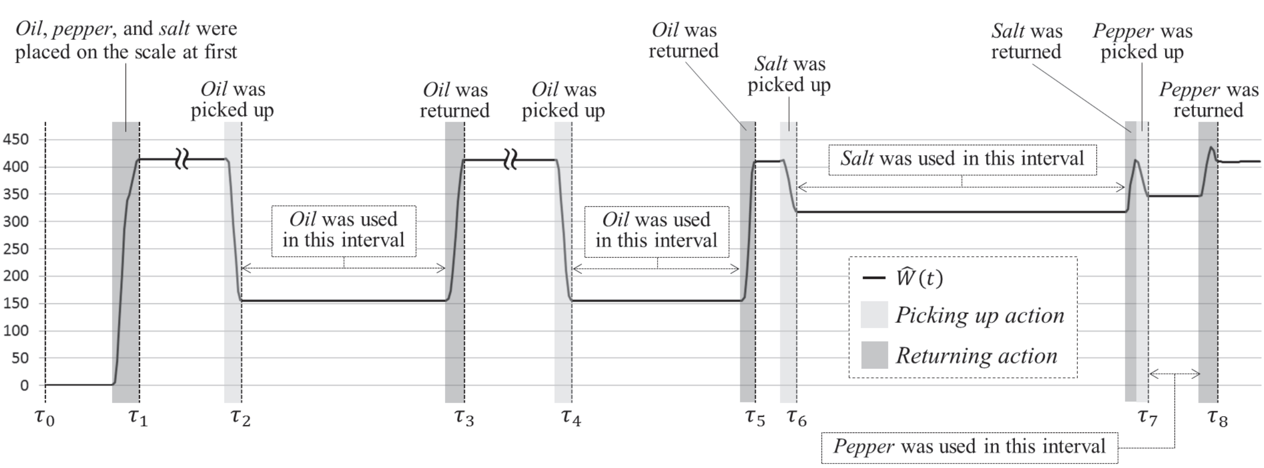


Fig. 8 Output value of electronic scales in example case shown in Table 1.

seventh steps in the example case.

There were also some failed cases, on the other hand, in the experiment. Most of the failed cases can be divided into three types:

A. Because of marker misrecognition, a set of picked up seasonings $\mathcal{P}(k)$ and that of returned seasonings $\mathcal{R}(k)$ were miscalculated, which made the *total weight equations* and *weight transition equations* incorrect.
B. Some *unsolvable weights* occurred.
C. Some $w(s;\tau_k)$ could not be solved during food preparation due to insufficient number of *total weight equations* and that of *weight transition equations*.

Type-C failures are different from type-B failures in that unsolved $w(s;\tau_k)$ were theoretically solvable if some additional equations were obtained in the future.

Type-A failures virtually cause type-B and type-C failures even if they could occur in reality. Therefore, we first evaluated how frequently type-A failures occurred, or how accurately the implemented system could calculate $\mathcal{P}(k)$ and $\mathcal{R}(k)$. Table 2 lists the results obtained from the evaluation, which indicates that the implemented system accurately calculated $\mathcal{P}(k)$ and $\mathcal{R}(k)$ in 79.4% of all the 301 trials. One or more type-A failures occurred in the remaining 20.6% of the trials because of marker misrecognition caused by illumination changes, shadows cast by other seasoning bottles, and occluded markers.

ARToolKit first binarizes each captured image with a fixed threshold to efficiently localize the positions of markers. Such binarization degrades the robustness of the marker detection process to illumination changes and cast shadows (see Fig. 9), and sometimes causes markers to be missed. This problem was noticeable in trials by participants 1 and 6. The error correction technique described in Sect. 3.4.2 cannot work to solve this problem when they occurred with a chef's *picking up/returning actions* involving a change in $\hat{W}(t)$. One solution for this problem is to adaptively change the threshold of binarization for each pixel of each frame. We cannot use the adaptive threshold in ARToolKit, but can use in a more modern library for Augmented Reality named ArUco [22]. We consider using this library as one of the alternatives for ARToolKit in the future. Another solution for robustly detecting markers is to employ state-of-the-art methods of object detection. Another error factor, which was marker occlusion by a chef's hand, also caused markers to be missed. This problem was noticeable in trials by participants 2 and 7, so that the numbers of successful trials by these two participants were relatively small, as listed in Table 2. Unfortunately, no occluded markers can be detected with any detection technique. Some other information is required to solve this problem. For instance, occluded markers do not change their positions on the camera images before and after occlusion because the corresponding seasoning bottles are not moved. In contrast, the bottles of actually used seasonings are moved by the chef's hand, which changes the positions of markers before and after they are used. This kind of difference would provide helpful information to solve the occlusion problem.

We next examined occurrence rates of failures by only focusing on the 239 trials in which no type-A failures occurred to evaluate how frequently type-B and type-C failures occurred. Table 3 summarizes the results, in which the occurrence rate for type-B failures was $2/239 = 0.8\%$ and that

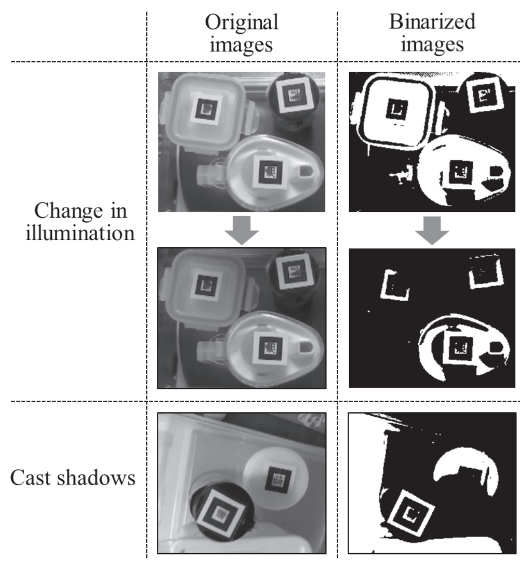


Fig. 9 Example of cases of marker misrecognition. Due to illumination changes and cast shadows, some markers could not retain their square shapes on binarized images. These kinds of markers can never be successfully detected with binarization-based techniques.

Table 2 Rate (%) of successfully calculating $\mathcal{P}(k)$ and $\mathcal{R}(k)$. N_{all} denotes total number of food preparation trials performed by each participant, N_{A} denotes number of trials in which one or more type-A failures occurred, and $N_{\bar{\mathrm{A}}}$ $(= N_{\mathrm{all}} - N_{\mathrm{A}})$ denotes number of trials in which no type-A failures occurred.

Participant ID	N_{all}	N_{A}	$N_{\bar{\mathrm{A}}}$	$N_{\bar{\mathrm{A}}}/N_{\mathrm{all}}$
1	50	15	35	70.0%
2	42	9	33	78.6%
3	2	1	1	50.0%
4	54	7	47	87.0%
5	47	4	43	91.5%
6	53	8	45	84.9%
7	53	18	35	66.0%
Total	301	62	239	**79.4%**

Table 3 Rate (%) of successfully measured consumption of seasonings. N_{B} and N_{C} denote number of trials in which type-B and type-C failures occurred. N_{other} denotes numbers of those in which other kinds of failures occurred and N_{ok} denotes numbers of those in which consumption of seasonings was successfully measured.

Participant ID	N_{B}	N_{C}	N_{other}	N_{ok}	$N_{\mathrm{ok}}/N_{\bar{\mathrm{A}}}$	$N_{\mathrm{ok}}/N_{\mathrm{all}}$
1	0	0	1	34	97.1%	68.0%
2	0	17	0	16	48.5%	38.1%
3	0	0	0	1	100.0%	50.0%
4	1	31	0	15	31.9%	27.8%
5	0	2	0	41	95.3%	87.2%
6	0	0	0	45	100.0%	84.9%
7	1	4	1	29	82.9%	54.7%
Total	2	54	2	181	**75.7%**	**60.1%**

for type-C failures was 54/239 = 22.6%. The consumption of seasonings was successfully measured in 181 out of the 239 trials. This indicates that the implemented system could successfully measure the consumption of seasonings with a probability of 75% if $\mathcal{P}(k)$ and $\mathcal{R}(k)$ were accurately calculated. Type-B failures rarely occurred in this experiment. This indicates that coping with *unsolvable weights* was not a very urgent issue.

Table 3 also indicates that type-C failures frequently occurred only in trials by participants 2 and 4. This is because they returned all the six seasoning bottles to the electronic scales simultaneously right after booting up the implemented system, whereas the other participants only returned the bottles necessary for cooking their intended meals. If all six bottles are simultaneously returned to the scale in the first step, the system has to start measurements with six unknown variables and only one *total weight equation*. Because the number of unknown variables is much more than that of equations in this case, it is not easy to obtain a sufficient number of additional equations to solve all unknown variables. Hence, type-C failures easily occurred in this case.

If the initial weights of several seasonings are known, type-C failures would occur less frequently. To confirm this, we evaluated the relation between the success rate of consumption measurements and the number of seasonings whose initial weights were known, focusing only on the trials by participants 2 and 4. More specifically, we virtually gave the initial weights of l (out of all the six) seasonings for each trial and simulated the proposed procedure for the consumption measurements. Figure 10 plots the results, which demonstrate that it is important to achieve successful measurements to decrease the number of seasonings whose initial weights are unknown. Although we currently assumed $\mathcal{S}(\tau_0) = \emptyset$ to simplify implementation, just a minor modification enables the implemented system to record onScale$(s; \tau_q)$ and $w(s; \tau_q)$ at the time τ_q when each food preparation opportunity ended and use them as onScale$(s; \tau_0)$ and $w(s; \tau_0)$ for the next opportunity. This should be quite helpful in decreasing the number of seasonings whose initial weights are unknown.

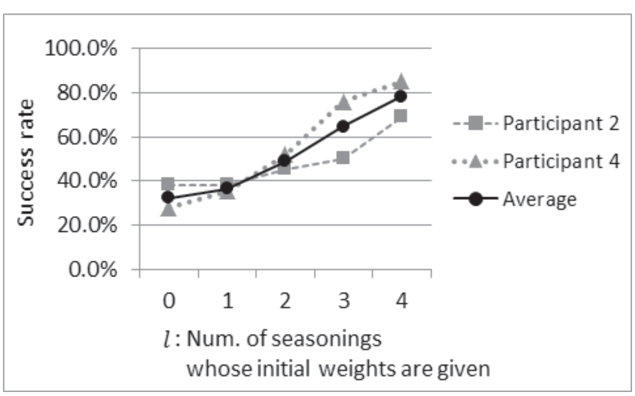


Fig. 10 Relation between success rate (%) of consumption measurements and number of seasonings whose initial weights are given. Performance for $l = 5$ is exactly equal to that for $l = 6$ because we focused on six seasonings in this experiment, and case of $l = 6$, i.e., case that initial weight of seasonings is completely known, was not very common even in trials by other participants. Therefore, we have only presented results for $l \leq 4$.

Eventually, the implemented system successfully measured the consumption of seasonings in 181 out of all 301 trials and the success rate was 181/301 = 60.1% as listed in Table 3. This rate is not so high, but it would be enough useful for several applications, because some mismeasurement cases can be corrected by making a small request to chefs. For instance, in the case of "taste recreating" described in Sect. 1, chefs have to do nothing for achieving step (i), i.e., recording the consumption of seasonings, when the automatic measurement is successfully done by the implemented system. Moreover, even when the system failed to measure the consumption of seasonings, it can still reduce the burden of step (i) on chefs. That is, in the case of type-C failures, all of the remaining unknown variables can be solved with some additional weight equations if chefs manually take down the seasoning bottles from the electronic scales one-by-one after finishing cooking. This is much less annoying for chefs than the case that they have to manually measure the weight of all seasonings. The results shown in Tables 2 and 3 demonstrate that almost half of the failure cases are type-C. The occurrence rate of other types of failures is about 20%. Thus, about 60% of success rate would be acceptable for supporting chefs to recreate the taste of meals. However, it is still desirable to further improve the performance. To this end, we will apply the following two modifications in a future work.

- Use more sophisticated methods for marker detection to accurately calculate $\mathcal{P}(k)$ and $\mathcal{R}(k)$
- Record $w(s; \tau_q)$ at the time τ_q when each food preparation opportunity ends and use it as $w(s; \tau_0)$ for the next opportunity

4.4 Effect of the Number of Seasonings

In the above experiment, we only focused on the case of using six kinds of seasonings. However, the number of seasonings would have a critical effect on the performance of the proposed system. To examine the effect, we conducted another experiment.

In this experiment, we focused on the following nine seasonings for defining a whole set of seasonings $\mathcal{S}$: *sugar*, *salt*, *vinegar*, *soy sauce*, *pepper*, *potato starch*, *cooking sake*, *oyster sauce*, and *olive oil*. The measurement process was started with the initial condition of nine unknown variables and only one weight equation, i.e., $\sum_{s \in \mathcal{S}} w(s; \tau_0) = \hat{W}(0)$, for each trial of food preparation. Seven participants have participated in this experiment, who are different from the participants in the first experiment, and the total number of the trials carried out by the seven participants was 36. At most four kinds of seasonings were actually added into a dish in each trial, so we simulated the case of using only

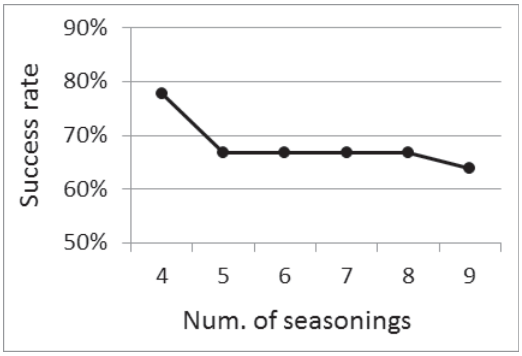


Fig. 11 Relation between success rate (%) of consumption measurements and number of seasonings. Success rate is almost constant when number of seasonings n is more than 4, because participants only use at most four kinds of seasonings per a trial.

$n \in \{8, 7, 6, 5, 4\}$ kinds of seasonings as $\mathcal{S}$, virtually ignoring $(9-n)$ seasonings that were actually not added.

In the result, type-A failures occurred in 7 trials out of the 36 trials; the occurrence rate was 19.4%. This is close to the result shown in Table 2, which indicates that the number of seasonings has little effect on the accuracy of the marker detection process. On the other hand, type-C failures occurred in 5 trials out of the 36 trials; the occurrence rate was 13.9%. This is slightly smaller than the result shown in Table 3.

Occurrence of type-C failures strongly depends on how many kinds of seasonings are simultaneously returned/picked up in each *WeCCA* event, which varies with individual chefs. In this experiment, many of the participants often return/pick up the seasoning bottles one-by-one. As a result, the occurrence rate of type-C failures became smaller in an average. The final success rate in this experiment was $23/36 = 63.9\%$, which was little improved in the cases of $n = 8$, 7, 6, and 5 as shown in Fig. 11. In the case of $n = 4$, no type-C failures occurred in every trial and the success rate was improved to $28/36 = 77.8\%$. This is because the participants did not use more than four kinds of seasonings per a trial in this experiment. This is also true in the first experiment. As shown in Fig. 10, the performance in the case of $l = 1$ is little improved from that in the case of $l = 0$. Because we used only six kinds of seasonings in the first experiment, the cases of $l = 0$ and $l = 1$ in Fig. 10 correspond to the cases of $n = 6$ and $n = 5$, respectively. These facts indicate that the performance of the proposed system degrades with the increasing number of seasonings actually used by a chef rather than those placed on the scale.

5. Related Work

There are several weight sensor-based devices which can help chefs to cook a dish. One example is digital spoon scale, which is commercially available. Although this device has the capability of automatically measuring the amount of seasonings, it also has the following drawback: digital spoon scale requires a chef to place it down on a level surface and wait until the indicator of the scale becomes stable. This means that, in order to record the consumption of seasonings, chefs have to place the spoon on a cooking counter and wait for a few seconds every time they use a seasoning. This can disturb the chefs' cooking operations. In contrast, the proposed system does not disturb chefs because it takes time to measure the consumption of a seasoning after the chef returns its bottle to the electronic scale. The value of the electronic scales will become stable while chefs are doing the next operations. In this sense, the proposed system is less annoying than digital spoon scale.

Cooking support systems equipped with weight sensors have been also proposed in previous studies [7], [8]. These systems basically focus on beginners in cooking and support them when they cooked dishes. More specifically, the systems provide instructions to users in each step of cooking, and the users cook dishes step by step by following the systems' instructions. Weight sensors incorporated in the systems are utilized to reducing the burden on users. The systems automatically check whether the correct quantity of ingredients has already been placed in dishes or not so that users do not have to manually measure the amounts. Since the main purpose of using weight sensors is not measurement itself, these kinds of systems do not have the capability of recording and storing the amounts of ingredients that have been used. This is also the case with *smoon* [19], which is a spoon-like volume-based measuring device for powdered or liquid ingredients proposed by Watanabe et al. In addition, these cooking support systems also do not have the capability of recognizing the kinds of ingredients that have been used because they assume that users have basically picked up the ingredients indicated by the system in each step of cooking.

Weight sensors are utilized for not only measuring but also recording the calories in individual ingredients or cooked food in several calorie estimation systems. For instance, the system by Chen et al. [11] estimates the calories in individual ingredients by looking up its per-gram calorie count in a nutritional database and multiplying that by its weight measured with a weight sensor. However, their system does not have the capability of recognizing the kinds of ingredients; users need to input the ingredients' names into the system through voice. Another calorie estimation system proposed by Saeki et al. [9] also has this limitation. Although there have been several studies proposing methods for recognizing the kinds of ingredients from food images for the purpose of estimating calories [20], [21], none of them have been integrated with actual systems that have been equipped with weight sensors. Unlike these existing systems, our proposed system can fully-automatically recognize the kinds of seasonings consumed at each time during food preparation by utilizing a camera and a technique of marker detection.

6. Conclusion

We developed a near real-time system for automatically measuring the amount of seasonings consumed during food preparation. The developed system is equipped with electronic scales and a camera. The electronic scales continually measure the total weight of seasoning bottles placed on them. On the other hand, the camera monitors whether individual seasoning bottles are placed on the scale or not at each time during food preparation. Since each seasoning is picked up from the scales before being used, and is returned to the scales after being used, the consumption of each seasoning is calculated as its difference in weight before and after it is used. We evaluated the developed system through experiments in 301 trials in actual food preparation carried out by seven participants. The result indicated the developed system successfully measured the consumption of seasonings in 181 trials out of 301 trials with a success rate of 60.1%. Most of the 120 failure trials could be divided into two cases.

- A set of picked up seasonings $\mathcal{P}(k)$ and that of returned seasonings $\mathcal{R}(k)$ were miscalculated, which resulted in incorrect weight equations.
- All the seasoning bottles were simultaneously returned to the scales at the beginning of food preparation, which resulted in poor initial conditions, i.e., only one weight equation with a number of unknown variables.

The first types of failures, i.e., miscalculation of $\mathcal{P}(k)$ and $\mathcal{R}(k)$, could be avoided by using more sophisticated methods for marker detection. The second types of failures, i.e., poor initial conditions, could be avoided by recording a set of seasonings placed on the scales as well as their remaining weights when each food preparation opportunity ended. We intend to make these two improvements to the developed system in future work. We also intend to address the problem of *unsolvable weights* in the future.

Acknowledgments

This work was supported by Japan Society for the Promotion of Science (JSPS) KAKENHI: Grant Numbers 24700089 and 24240030.

References

[1] A. Yajima and I. Kobayashi, "“Easy” cooking recipe recommendation considering user's conditions," Proc. IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technologies, vol.3, pp.13–16, 2009.
[2] J. Freyne and S. Berkovsky, "Intelligent food planning: Personalized recipe recommendation," Proc. 15th Int'l Conf. on Intelligent User Interfaces, pp.321–324, 2010.
[3] S. Berkovsky and J. Freyne, "Group-based recipe recommendations: Analysis of data aggregation strategies," Proc. 4th ACM Conf. on Recommender Systems, pp.111–118, 2010.
[4] C.-Y. Teng, Y.-R. Lin, and L.A. Adamic, "Recipe recommendation using ingredient networks," Proc. 3rd Annual ACM Web Science Conference, pp.298–307, 2012.
[5] I. Siio, R. Hamada, and N. Mima, "Kitchen of the future and applications," Human-Computer Interaction: Interaction Platforms and Techniques, Lecture Notes in Computer Science, vol.4551, pp.946–955, Springer, 2007.
[6] A. Hashimoto, N. Mori, T. Funatomi, Y. Yamakata, K. Kakusho, and M. Minoh, "Smart kitchen: A user centric cooking support system," Proc. 12th Int'l Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems, pp.848–854, 2008.
[7] D. Surie, H. Lindgren, and A. Qureshi, "Kitchen AS-A-PAL: Exploring smart objects as containers, surfaces and actuators," Ambient Intelligence — Software and Applications, Advances in Intelligent Systems and Computing, vol.219, pp.171–178, Springer, 2013.
[8] A. Sato, K. Watanabe, and J. Rekimoto, "MimiCook: A cooking assistant system with situated guidance," Proc. 8th Int'l Conf. on Tangible, Embedded and Embodied Interaction, pp.121–124, 2014.
[9] Y. Saeki and F. Takeda, "Proposal of food intake measuring system in medical use and its discussion of practical capability," Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Computer Science, vol.3683, pp.1266–1273, Springer, 2005.
[10] K.-H. Chang, S.-Y. Liu, H.-H. Chu, J.-Y. Hsu, C. Chen, T.-Y. Lin, C.-Y. Chen, and P. Huang, "The diet-aware dining table: Observing dietary behaviors over a tabletop surface," Pervasive Computing, Lecture Notes in Computer Science, vol.3968, pp.366–382, Springer, 2006.
[11] J.-H. Chen, P.P.-Y. Chi, H.-H. Chu, C.C.-H. Chen, and P. Huang, "A smart kitchen for nutrition-aware cooking," IEEE Pervasive Comput., vol.9, no.4, pp.58–65, 2010.
[12] G. Villalobos, R. Almaghrabi, B. Hariri, and S. Shirmohammadi, "A personal assistive system for nutrient intake monitoring," Proc. 2011 Int'l ACM Workshop on Ubiquitous Meta User Interfaces, pp.17–22, 2011.
[13] Y. Murozaki and F. Arai, "Wide range load sensor using quartz crystal resonator for detection of biological signals," IEEE Sensors J., vol.15, no.3, pp.1913–1919, 2015.
[14] D. Levi, S. Silberstein, and A. Bar-Hillel, "Fast multiple-part based object detection using KD-ferns," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.947–954, 2013.
[15] R. Sznitman, C. Becker, F. Fleuret, and P. Fua, "Fast object detection with entropy-driven evaluation," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.3270–3277, 2013.
[16] L. Zhang and L. van der Maaten, "Structure preserving object tracking," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.1838–1845, 2013.
[17] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.2411–2418, 2013.
[18] H. Kato and M. Billinghurst, "Developing AR applications with ARToolKit," Proc. 3rd IEEE/ACM Int'l Symp. on Mixed and Augmented Reality, p.305, 2004.
[19] K. Watanabe, S. Matsuda, A. Sato, M. Inami, and T. Igarashi, "smoon: A spoon with automatic capacity adjustment," Laval Virtual, 2012.
[20] R. Almaghrabi, G. Villalobos, P. Pouladzadeh, and S. Shirmohammadi, "A novel method for measuring nutrition intake based on food image," Proc. IEEE Int'l Instrumentation and Measurement Technology Conf., pp.366–370, 2012.
[21] P. Pouladzadeh, G. Villalobos, R. Almaghrabi, and S. Shirmohammadi, "A novel SVM based food recognition method for calorie measurement applications," Proc. IEEE Int'l Conf. on Multimedia and Expo Workshops, pp.495–498, 2012.
[22] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and M.J. Marin-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," Pattern Recognit., vol.47, no.6, pp.2280–2292, 2014.

Kazuaki Nakamura is currently an assistant professor at Graduate School of Engineering, Osaka University, Japan from 2012. He received B.S. degree in Engineering, M.S. and Ph.D. degrees in Informatics from Graduate School of Informatics, Kyoto University, Japan, in 2005, 2007, and 2011, respectively. He has been working on understanding of human emotion and actions using pattern recognition techniques. His research interests include image processing, image recognition, video analysis, and human interface. He is a member of Institute of Electronics, Information and Communication Engineers of Japan, and the IEEE Computer Society.

Takuya Funatomi is currently an associate professor at Information Science department, Nara Institute of Science and Technology, Japan from 2015. He was an assistant professor at Academic Center for Computing and Media Studies, Kyoto University, Japan from 2007 to 2015, and a visiting assistant professor at Stanford University, USA in 2014. He received B.S. degree in Engineering, M.S. and Ph.D. degrees in Informatics from Graduate School of Informatics, Kyoto University, Japan, in 2002, 2004, and 2007, respectively. He has been working on 3D shape reconstruction of human as a deformable object and image processing techniques to observe human activity such as cooking. His research interests include computer vision, computer graphics, and pattern recognition. He is a member of Institute of Electronics, Information and Communication Engineers of Japan, and the IEEE Computer Society and Communication Society.

Atsushi Hashimoto is currently an assistant professor at Graduate School of Law, Kyoto University, Japan from 2012. He received B.S. degree in Engineering, M.S. and Ph.D. degrees in Informatics from Graduate School of Informatics, Kyoto University, Japan, in 2005, 2008, and 2013, respectively. He has been working on human activity analysis and recognition based on human-object interaction, and interface design and development in the field of cooking. His research interests include pattern recognition, computer vision, and human-computer interaction. He is a member of Institute of Electronics, Information and Communication Engineers of Japan, and Information Processing Society of Japan.

Mayumi Ueda is an associate professor at Faculty of Economics, University of Marketing and Distribution Sciences, Japan from 2013. She received Ph.D. degree in Informatics from Graduate School of Informatics, Kansai University, Japan, in 2003. She has been working on development of information recommendation systems and learning management systems. She is a member of Institute of Electronics, Information and Communication Engineers of Japan, Information Processing Society of Japan, the IEEE Computer Society and Communication Society, and ACM.

Michihiko Minoh is a professor at Academic Center for Computing and Media Studies, Kyoto University, Japan. He received the B.Eng., M.Eng., and D.Eng. degrees in information science from Kyoto University, in 1978, 1980, and 1983, respectively. He served as director of ACCMS from April 2006 to March 2010 and concurrently served as vice director in the Kyoto University Presidents Office from October 2008 to September 2010. Since October 2010, he has been vice-president, chief information officer at Kyoto University, and director-general at Institute for Information Management and Communication, Kyoto University. His research interests include a variety area of Image Processing, Artificial Intelligence and Multimedia Applications, particularly, model centered framework for the computer system to help visual communication among humans and information media structure for human communication. He is a member of Information Processing Society of Japan, Institute of Electronics, Information and Communication Engineers of Japan, the IEEE Computer Society and Communication Society, and ACM.