Grocery Re-identification using Load Balance Feature on the Shelf for Monitoring Grocery Inventory

Rena Kamoda [1] and **Mayumi Ueda**[2] and **Takuya Funatomi** [1] and **Masaaki Iiyama** [1] and **Michihiko Minoh** [1]

Abstract. We propose a method to monitor the grocery items in a home refrigerator. Using load sensors mounted under a shelf of the refrigerator, our method matches groceries that are put into and taken from the refrigerator. Our prototype load-sensing board uses four load sensors to acquire the weight and position of the grocery items, and we use this data to re-identify the groceries. Detailed experiments show that this feature can accurately re-identify grocery items and thus provide constant monitoring of refrigerator contents.

1 Introduction

You may have experienced upon returning from a shopping trip that you already had some of the purchased groceries in your refrigerator, or that you forgot to buy certain groceries. You may have also found yourself trying to remember the contents of your refrigerator when shopping for dinner in a supermarket. A system for monitoring the contents of a refrigerator would be helpful in such situations. We aim to construct a system that sends the image and weight of the grocery items in the refrigerator to the user, who can receive this information conveniently via smartphone.

If a system consistently monitors what comes into and what goes out of a refrigerator, it effectively shows the contents of the refrigerator to the user. When a grocery item enters the refrigerator, the system registers its image and weight to an inventory list. When a grocery item exits the refrigerator, the system matches it to an entry in the inventory list and deletes the entry. This matching is done using the information registered in the list. In this paper, we discuss what kind of information is useful for grocery re-identification and propose a method to re-identify groceries.

While there exist some smartphone applications that help the user to manage the grocery in his/her refrigerator, these applications require the user to manually enter what is stored or taken out. Such manual entry proves troublesome; ideally, the process should be automated. Some systems identify groceries using barcodes or radio frequency identification (RFID) tags; however, this is unacceptable from a sanitary viewpoint. Instead of using tags, we propose the use of information that is naturally available to the system. We discuss an appearance feature and a load balance feature and propose the use of load balance feature for grocery re-identification.

In section 2, we discuss the grocery inventory system. In section 3, we describe a method of re-identification that uses the load balance feature. Finally, in section 4, we demonstrate the effectiveness of this method through some experiments.

[1] Kyoto University, email: kamoda@mm.media.kyoto-u.ac.jp
[2] University of Marketing and Distribution Sciences

Figure 1. Groceries in a refrigerator displayed on a smartphone

2 Grocery inventory system

2.1 Monitoring inventory with tags

There exist smartphone applications that aim to help users to manage grocery in their refrigerators. However, these applications require users to manually register each refrigerator transaction. We take an automated approach. When a grocery item is stored in the refrigerator, the system registers it into an inventory list. When a grocery item is taken out of the refrigerator, the system re-identifies it by matching it to the groceries in the inventory list. By this approach, we can automatically know what is in the refrigerator in real time. We just monitor what is stored in and taken out of the refrigerator and we don't care about the groceries after they are taken out of the refrigerator: if the grocery which are taken out of the refrigerator is stored again, we treat it as a new entry.

To achieve such re-identification, the system needs some information about each grocery. Identification tags are commonly used for this purpose. Retail stores worldwide use barcodes and factory storage commonly uses RFID tags. Barcodes and RFID tags provide reliability: the system can identify each product with near-certainty. However, tagging is unacceptable for non-packaged products, such as vegetables. Furthermore, grocery items can be refrigerated after some form of processing, such as cutting or peeling, or used only in part before being returned to the refrigerator. In such a situation, a tag originally attached to the grocery item must be replicated and reattached to each refrigerated portion. This is obviously unrealistic.

2.2 Design of grocery inventory system

In this study, we use grocery information that the system can obtain naturally during a refrigerator transaction. We discuss the use of the

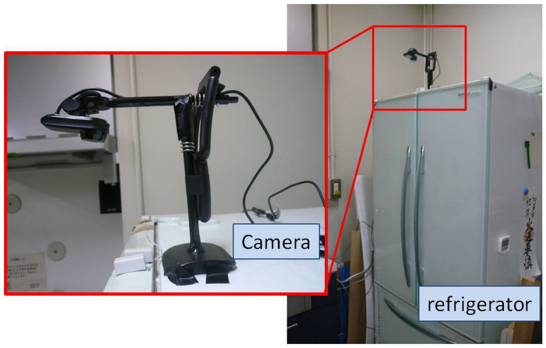

Figure 2. The camera set a refrigerator

Figure 3. The images captured by the camera. The groceries are occluded by user's hand and their visible portion is inconsistent.

appearance feature and the load feature. A camera attached to the refrigerator allows the grocery image to be captured, and we extract the appearance feature from the image; likewise, a weight scale on a refrigerator shelf allows the weight to be determined. These features are stable because they are inherent to the grocery. In this study, we suppose that grocery doesn't change their weight and position in the refrigerator during it is stored. We also suppose that grocery is stored in or taken out of the refrigerator one by one.

This information can be presented to the user directly. Figure 1 shows the possible user interface of a smartphone application, which displays the image and indicates the weight of each grocery item present in the refrigerator.

2.3 Possible features for re-identifying grocery

2.3.1 Appearance feature

Grocery re-identification in a refrigerator context is made very difficult by the complexity of the visual scene within a refrigerator. If these items were to be captured by a camera placed inside the refrigerator, many would appear occluded. In such a situation, it would be difficult to extract the appearance feature for each grocery item. In this study, the system captures grocery items during refrigerator transactions; it is possible to capture a grocery item individually with less occlusion when it is entering or leaving the refrigerator. We set a camera along the top of the refrigerator, as shown in Figure 2. Figure 3 shows the images captured by the camera. In order to extract the appearance feature, we can obtain the grocery region by background subtraction.

There is some related work on person re-identification using the appearance feature ([1]-[3]). Wang et al. [1] use a histogram of oriented gradients (HOG) and the shape context as appearance features. Bak et al. [2] use the gradient direction and the covariance of the gradient intensity. Zheng et al. [3] use a scale-invariant feature transform (SIFT) for each RGB channel in each local region and the normalized average RGB vector. Although these appearance features can be also used for grocery re-identification, we note that there are some differences between grocery re-identification and person re-identification. In the case of a person, we can minimize the effects of occlusion by using information aggregated over time. In the case of grocery items being inserted into or taken from a refrigerator, the user 's hand is always an occluding factor, as shown in Figure 3. Moreover, the visible portion of the grocery item is inconsistent between the entry and retrieval phases as shown in Figure 3.

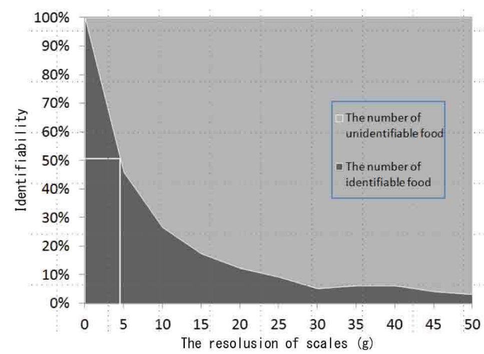

Figure 4. Resolution of weight scale and the ability of distinguishing groceries

2.3.2 Load feature

The load feature can provide an efficient way to re-identify grocery items under the assumption that the weight of a grocery item remains unchanged during refrigeration. In order to evaluate the suitability of the load feature, we performed a preliminary investigation.

Preliminary investigation: We surveyed the refrigerator of a five-member family. There were 98 grocery items in the refrigerator; we measured them using a digital weight scale of $1g$ resolution. We calculated the weight of each grocery item, and separately clustered them into each shelf; Table 1 shows these weights. As in the case of bean paste (type 1) and a grocery in a tupper (type 1) on shelf 4, and seaweed and mustard (type 2) on shelf 7, certain grocery items on a shelf weigh the same as others, within a $1g$ resolution. The maximum total weight on a shelf was about $5.1kg$.

We studied the relationship between the resolution of the scales and the identifiability of the 98 grocery items as shown in Figure 4. Figure 4 shows that it is impossible to identify half of the grocery items at a resolution of about $5g$. We conclude that the load feature is insufficient by itself.

We set a scale for each shelf in the refrigerator, and we obtain the weight feature from it. The grocery items on each shelf are arranged by weight, as shown in Table 1.

Table 1. The weight of groceries on each shelf

shelf 4	
grocery name	(g)
cheese (type 1)	18
bean paste (type 1)	**41**
a grocery in a tupper (type 1)	**41**
egg (type 1)	79
a grocery in a tupper (type 2)	93
seasoning (type 2)	135
seasoning (type 1)	155
a grocery in a nylon bag (type 1)	157
jelly (type 1)	196
jelly (type 2)	206
a grocery in a nylon bag (type 2)	227
bean paste (type 2)	290

@

shelf 7	
grocery name	(g)
green horseradish paste	13
seasoning (type 3)	16
Cooling Gel Sheet (type 1)	25
ginger	31
garlic	33
cheese (type 2)	34
Cooling Gel Sheet (type 2)	35
seaweed	**49**
mustard (type 2)	**49**
seasoned laver	59
mustard (type 2)	59
vanilla flavoring	76
coffee powder	82
condensed milk	104
broad bean chili paste	152
jelly (type 3)	212
sauce	242
kimchi	277
soy sauce	369
egg (type 2)	431

2.3.3 Load balance feature

Weight of grocery gives us good information for our grocery re-identification, but position of grocery on a shelf also gives good information. Schmidt et al [4] proposed the load sensing table in which four load sensors were installed at the four corners underneath a table. The load sensing table can calculate the weight and position of objects on it and detect some interactions, such as object placement and removal. Pei-yu Chi et al [5] uses this sensing design to track food calories during cooking.

In order to measure the load on a shelf, we also install four load sensors, one under each corner, allowing us to measure not only the total load but also the load balance on the shelf. The load balance also indicates the positions of the grocery items. Even if two groceries weigh the same at a certain resolution, the system can distinguish them by their position using the load balance feature. This provides richer information for grocery re-identification.

3 Grocery re-identification using load balance feature on a shelf

3.1 Load-sensing board

In this section, we describe our prototype load-sensing board. The design was based on the load sensing table [4]. Figure 5 depicts the load-sensing board. Four load sensors (Figure 6) are attached to the corners of a tempered glass sheet (size; $550mm \times 290mm \times 6mm$; weight, $2.5kg$, and load capacity, $5kg$). The load on each sensor is converted to digital form using an analog-to-digital converter as shown in Figure 7.

We used 'ASFORCE' load sensors [3] made by Asakusa-Giken Co., LTD[6] and the AGB65-ADC analog-to-digital converter [4]. Each AS-FORCE sensor is capable of measuring a maximum of 2.8 kgf, so the load sensing board can measure a maximum of 11.2kgf using the four sensors. The weight of the glass is $2.5kg$ and the sum of weight

[3] Its product is stopped by February 2012. http://www.robotsfx.com/robot/ForceSen.html

[4] http://www.robotsfx.com/robot/AGB65 ADC.html

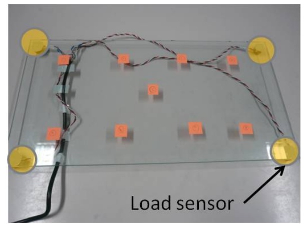

Figure 5. Load-sensing Board

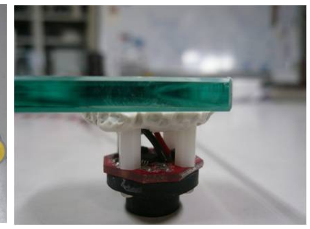

Figure 6. Load sensor (AS-FORCE)

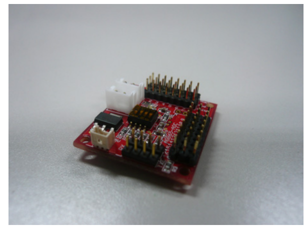

Figure 7. A/D converter (AGB65-ADC)

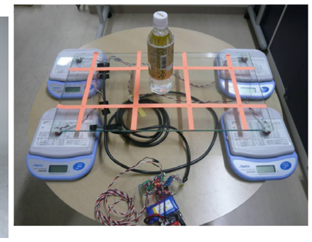

Figure 8. Calibration technique: we measure the sensor value and corresponding weight.

of grocery which is put on it is $5.1kg$. Thus, it is possible to measure grocery weight up to $7.6kg$. The 12-bit binary value obtained from the AGB65-ADC is sent to a computer via COM port at a baud rate of 9600 bps.

3.2 Algorithm for re-identification with load balance feature

Our system monitors the changes in the sensed values. Let $\boldsymbol{\Delta z} = (\Delta z_1, \Delta z_2, \Delta z_3, \Delta z_4)$ denote the changes in the values of the four sensors when grocery is placed inside (or taken out of) the refrigerator. $\boldsymbol{\Delta z}$ is the load balance feature and denotes the weight and position of the grocery for re-identification. The dissimilarity of the load balance feature is given by the distance between the two vectors $\boldsymbol{\Delta z}^{IN}$ and $-\boldsymbol{\Delta z}^{OUT}$, where $\boldsymbol{\Delta z}^{IN}$ denotes the load balance feature when a grocery item is put on the board and $\boldsymbol{\Delta z}^{OUT}$ denotes the load balance feature when a grocery item is taken off the board.

Suppose there are N groceries in the refrigerator. Let $\boldsymbol{\Delta z}_i^{IN}(i = 1, ..., N)$ be their load balance features. When a grocery is taken out and its load balance feature is $\boldsymbol{\Delta z}_k^{OUT}$, we re-identify the grocery with $argmin_i|\boldsymbol{\Delta z}_i^{IN} + \boldsymbol{\Delta z}_k^{OUT}|$.

3.3 Characteristic evaluation of load sensor

The value obtained from the load sensor is a digital form of the voltage output and does not directly express the load on the sensor. The relation between the sensed value and the actual load is expressed as a linear relation, $z = aw + b$, where z is the digital data and w is the corresponding load. The coefficients a and b depend on the individual sensor. We calculated a of each sensor as follows. First, we put digital weight scales under each sensor and loaded it with known weights. Next, as shown in Figure 8, we measure the sensor value z and corresponding weight w for variety of position and weight.

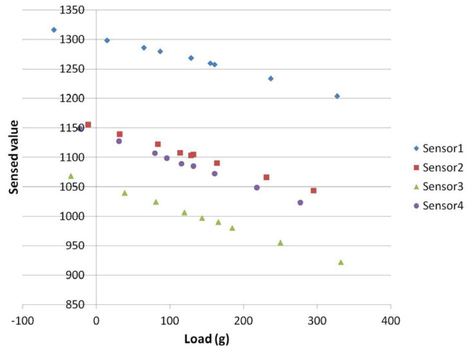

Figure 9. Sensed value and load

Figure 10. The grocery items used in the dataset 1.

Figure 11. The grocery items used in the dataset 2.

Figure 9 shows the result. The result shows the linear relation between load data and actual load. Furthermore, there are individual differences of gradient a between sensors. For measuring the weight of grocery, we convert the sensed value to load and then display the weight on a smartphone.

4 Experiment

We conducted experiments with two different item sets. The first item sets named dataset 1 consists of eight grocery items which are put on the load-sensing board in neat rows. The second item sets named dataset 2 consists of 15 grocery items which are put randomly, and some of them are stacked.

In 4.2, we describe grocery re-identification using the appearance feature. In 4.3, we describe grocery re-identification using the load balance feature: the method explained in section 3 to demonstrate the effectiveness of our method.

4.1 Settings

The grocery items used in the two datasets are shown in Figures 10 and 11. We set a camera along the top of the refrigerator, as shown in Figure 2. The system captures grocery items during refrigerator transactions. The load-sensing board is placed on a refrigerator shelf, and grocery items are placed on it, as shown in Figures 10 and 11. Then, the grocery items are removed in a different order.

Figure 12. An example of the image captured by the camera

Figure 13. The region image extracted from the image in Figure 12

4.2 Grocery re-identification using appearance feature

From the image captured by the camera, we extracted the grocery region manually. For that region, we calculated RGB color histogram. For the two datasets, we use the histogram as the appearance feature. Then, we calculated distances between groceries and conducted re-identification. Figure 12 shows an example of the image captured by the camera and Figure 13 shows the region image extracted from the image in Figure 12.

Table 2 shows the result of re-identification for the dataset 1 and Table 3 shows the result for the dataset 2. The top rows of the tables list the grocery items that are put in the refrigerator. The leftmost columns of the table list the grocery items that are taken out of the refrigerator. The middle of the table displays the Bhattacharyya distance between the features. The rightmost column displays the minima of the dissimilarities. The values on the diagonal cells, which should have minimum distance, are underlined. We searched for the nearest neighbor: searching for a inserted grocery item that is most similar to a removed grocery item. The nearest neighbor is marked with *.

For the dataset 1, all eight are successfully re-identified and thirteen of fifteen are successfully re-identified for the dataset 2. Table 3 shows that the removed lotus root was identified as the inserted potato. Their color are similar to each other, so they were incorrectly identified. They might be correctly re-identified if we add other feature to color histogram as the appearance feature. Table 3 also shows that the removed ham has more similar feature to the inserted natto than the inserted ham. Figure 13 shows that the image which was taken when ham was removed out of the refrigerator. Figure 14 and Figure 15 show that the images which were taken when the ham and the natto was inserted to the refrigerator. When the ham was inserted to the refrigerator, its one side appeared to the camera, but when it was removed, its the other side appeared. Thus, the ham has dissimilar appearance features when it was inserted and removed, and that is the reason why removed ham was mis-identified with the natto.

These results shows that appearance feature could be useful information to grocery re-identification and the accuracy of re-identification could be improved by using multiple features as the appearance feature. However, the camera doesn't necessarily capture the same visible portion of groceries all the time, so grocery re-identification using only appearance feature is hard even if adding multiple features to the appearance features..

Table 2. The result of re-identification using appearance feature for the dataset 1

IN / OUT	Sausage	Tupper	Tomato	Eggplant	Carrot	Navel orange	Butter	Lotus root	NN
Sausage	* 0.41	0.90	0.80	0.70	0.90	0.93	0.83	0.67	* 0.41
Tupper	0.90	* 0.48	0.79	0.92	0.94	0.97	0.97	0.93	* 0.48
Tomato	0.81	0.80	* 0.33	0.84	0.83	0.93	0.94	0.84	* 0.33
Eggplant	0.77	0.96	0.86	* 0.66	0.93	0.93	0.92	0.84	* 0.66
Carrot	0.88	0.95	0.86	0.92	* 0.41	0.70	0.94	0.89	* 0.41
Navel orange	0.90	0.97	0.92	0.93	0.88	* 0.78	0.93	0.85	* 0.78
Butter	0.85	0.97	0.95	0.93	0.94	0.96	* 0.49	0.90	* 0.49
Lotus root	0.66	0.94	0.84	0.81	0.89	0.89	0.87	* 0.47	* 0.47

Table 3. The result of re-identification using appearance feature for the dataset 2

IN / OUT	Sausage	Cabbage	Cucumber	Potato	Jelly	Tupper	Tomato	Eggplant	Natto	Carrot	Navel orange	Butter	Ham	Peanut Butter	Lotus root	NN
Sausage	* 0.45	0.81	0.80	0.56	0.74	0.90	0.84	0.79	0.76	0.91	0.93	0.76	0.70	0.75	0.82	* 0.45
Cabbage	0.85	* 0.65	0.73	0.85	0.92	0.97	0.95	0.93	0.88	0.93	0.96	0.91	0.91	0.89	0.91	* 0.65
Cucumber	0.79	0.79	*0.66	0.82	0.91	0.96	0.95	0.80	0.83	0.94	0.95	0.92	0.89	0.89	0.93	* 0.66
Potato	0.60	0.84	0.81	* 0.45	0.77	0.93	0.88	0.81	0.69	0.88	0.91	0.78	0.74	0.71	0.79	* 0.45
Jelly	0.81	0.93	0.88	0.81	* 0.57	0.73	0.66	0.87	0.73	0.90	0.93	0.93	0.62	0.68	0.91	* 0.57
Tupper	0.90	0.97	0.94	0.92	0.85	* 0.45	0.72	0.94	0.88	0.94	0.96	0.97	0.85	0.89	0.95	* 0.45
Tomato	0.85	0.96	0.91	0.87	0.62	0.63	* 0.50	0.91	0.84	0.91	0.95	0.96	0.73	0.78	0.92	* 0.50
Eggplant	0.65	0.91	0.75	0.76	0.88	0.95	0.91	* 0.41	0.81	0.94	0.94	0.92	0.83	0.86	0.94	* 0.41
Natto	0.82	0.90	0.82	0.77	0.79	0.92	0.87	0.81	* 0.43	0.87	0.90	0.89	0.80	0.71	0.90	* 0.43
Carrot	0.90	0.95	0.92	0.88	0.90	0.94	0.91	0.95	0.89	* 0.51	0.79	0.95	0.91	0.89	0.93	* 0.51
Navel orange	0.92	0.96	0.94	0.91	0.92	0.95	0.92	0.95	0.90	0.75	* 0.50	0.97	0.92	0.89	0.95	* 0.50
Butter	0.76	0.90	0.87	0.76	0.92	0.97	0.94	0.84	0.86	0.92	0.93	* 0.72	0.89	0.88	0.91	* 0.72
Ham	0.80	0.93	0.87	0.81	0.75	0.93	0.87	0.83	* 0.64	0.94	0.95	0.92	0.65	0.77	0.94	* 0.64
Peanut Butter	0.73	0.87	0.82	0.70	0.73	0.87	0.79	0.78	0.63	0.84	0.85	0.89	0.72	* 0.62	0.87	* 0.62
Lotus root	0.68	0.87	0.81	* 0.55	0.84	0.94	0.89	0.85	0.83	0.89	0.92	0.88	0.83	0.74	0.74	* 0.55

Figure 14. The region image extracted from the inserted ham image.

Figure 15. The region image extracted from the inserted natto image.

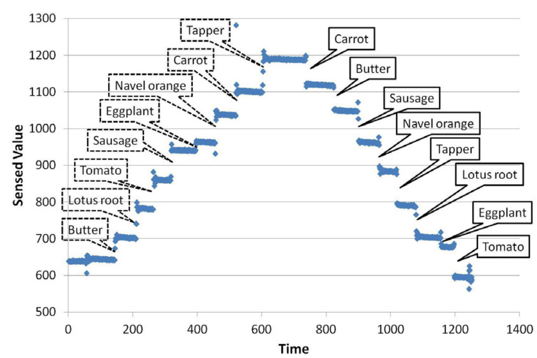

Figure 16. The changes in the sum of the four sensors

.

4.3 Grocery re-identification using load balance feature

Figure 16 shows the changes in the sum of the four sensors during refrigerator transactions for the datset 1. As seen in Figure 16, the load data varies drastically when a transaction occurs. In our experiments, the static part was extracted manually. We calculated the averages of the data in each static portion and used them for re-identification.

Tables 4 and 5 show the result of re-identification for the dataset 1 and Table 6 shows the result for the dataset 2. The results in Tables 4 and 6 used the load balance feature, while the result in Table 5 used the load feature. In Table 5, the load features are listed with grocery name in the top rows and the leftmost columns. The middle of the table displays the dissimilarities (here, the differences in the load balance vectors or the load) between the features.

For the dataset 1, all eight are successfully re-identified with the load balance feature while two are re-identified with the load feature. From these results, we conclude that the load balance feature gives more accurate results than the load feature. For the dataset 2, all fifteen were successfully re-identified with the load balance feature. However, the tapper and the navel orange have similar distance. The distance between the tapper taken out of and the tapper stored in is 4.63. The distance between the tapper taken out of and the navel orange stored in is 5.92. In this case, the tapper could be identified as the navel orange. This is because for two reasons below. They have similar weights: the tapper weighs $197g$ and the navel orange weighs $213g$. They were put on a similar position on the load-sensing board: the navel was stacked on the tapper. Thus, they have similar load balance features and they could be re-identified incorrectly.

Table 4. The result of re-identification using load balance feature for the dataset 1

IN OUT	Sausage	Tupper	Tomato	Eggplant	Carrot	Navel orange	Butter	Lotus root	NN
Sausage	* 4.33	47.22	21.99	58.42	43.87	56.51	60.32	44.59	*4.33
Tupper	49.29	* 3.54	53.28	63.17	35.23	47.28	74.27	66.47	*3.54
Tomato	21.02	48.54	* 5.13	44.49	33.88	43.42	38.56	22.73	*5.13
Eggplant	53.95	57.28	41.59	* 3.37	26.38	27.56	32.75	38.58	*3.37
Carrot	41.93	28.18	35.77	33.50	* 4.52	19.69	46.53	42.57	*4.52
Navel orange	55.66	43.41	44.82	31.85	17.50	* 2.51	41.07	42.50	*2.51
Butter	54.68	69.15	35.51	33.49	40.90	36.73	* 6.90	15.19	*6.90
Lotus root	40.35	62.46	21.29	41.58	39.72	40.92	22.09	* 4.81	*4.81

Table 5. The result of re-identification using load feature for the dataset 1

(g)	IN	Sausage	Tupper	Tomato	Eggplant	Carrot	Navel orange	Butter	Lotus root	
OUT		208.61	215.62	211.67	65.65	168.44	202.70	182.58	228.06	NN
Sausage	222.62	14.01	7.00	10.95	156.97	54.18	19.92	40.04	*5.44	*5.44
Tupper	229.90	21.28	14.28	18.22	164.25	61.45	27.20	47.32	* 1.83	*1.83
Tomato	223.51	14.90	7.90	11.84	157.86	55.07	20.81	40.93	*4.55	*4.55
Eggplant	77.63	130.98	137.98	134.04	* 11.98	90.81	125.07	104.95	150.43	*11.98
Carrot	182.65	25.97	32.97	29.03	117.00	14.20	20.05	* 0.07	45.41	*0.07
Navel orange	213.13	4.51	2.49	* 1.45	147.48	44.68	10.43	30.55	14.93	*1.45
Butter	208.91	* 0.29	6.71	2.77	143.26	40.46	6.21	26.33	19.16	*0.29
Lotus root	247.78	39.16	32.16	36.11	182.13	79.33	45.08	65.20	*19.72	*19.72

Table 6. The result of re-identification using load balance feature for the dataset 2

IN OUT	Sausage	Cabbage	Cucumber	Potato	Jelly	Tupper	Tomato	Eggplant	Natto	Carrot	Navel orange	Butter	Ham	Peanut Butter	Lotus root	NN
Sausage	* 2.81	58.84	40.69	36.31	30.98	60.80	19.11	44.90	39.42	44.39	58.07	59.80	50.73	37.37	36.56	*2.81
Cabbage	59.51	* 1.59	80.00	87.52	67.53	106.53	71.15	86.42	89.81	86.32	105.82	97.64	96.73	88.74	71.40	*1.59
Cucumber	39.77	79.29	* 0.70	38.06	61.00	56.92	33.98	7.30	18.06	22.57	55.40	34.46	18.48	34.79	25.31	*0.70
Potato	34.86	87.53	37.94	* 0.51	36.70	25.68	17.82	37.63	29.49	26.00	22.68	39.45	36.61	5.14	29.61	*0.51
Jelly	31.44	67.34	61.79	36.74	* 3.67	50.23	29.18	65.03	58.88	54.91	48.65	68.64	68.80	40.37	46.63	*3.67
Tupper	58.85	106.59	57.69	25.16	50.37	*4.63	41.62	55.75	49.87	39.14	5.92	44.19	51.86	27.08	45.48	*4.63
Tomato	18.26	71.16	33.92	17.64	28.81	41.89	* 1.43	36.32	30.34	28.92	39.44	44.48	39.83	19.38	24.20	*1.43
Eggplant	44.62	86.42	6.72	38.04	64.71	55.14	37.06	* 1.36	15.38	21.45	53.58	31.59	11.67	34.32	28.55	*1.36
Natto	38.26	88.40	16.25	29.85	57.62	49.46	30.10	14.28	* 1.59	24.33	46.82	37.51	14.08	25.79	32.06	*1.59
Carrot	43.43	85.75	21.12	26.62	55.17	40.12	29.95	19.44	24.12	* 2.19	39.48	17.86	21.13	24.72	15.80	*2.19
Navel orange	57.26	106.24	56.31	23.25	49.53	4.39	40.19	54.39	47.72	38.85	* 2.21	44.97	50.37	24.80	45.50	*2.21
Butter	59.41	97.42	34.60	39.74	68.66	45.22	45.66	31.06	38.16	16.98	45.71	* 0.38	29.37	38.78	27.62	*0.38
Ham	49.74	96.29	17.98	37.22	68.13	52.27	40.24	11.53	13.57	22.80	50.32	30.06	* 1.52	33.40	34.56	*1.52
Peanut Butter	35.01	87.77	34.63	4.54	39.44	27.12	18.34	34.17	25.92	23.93	24.20	37.98	33.10	* 1.58	28.68	*1.58
Lotus root	35.97	71.90	25.12	29.39	47.15	45.66	25.14	27.28	32.45	15.03	45.28	27.12	33.32	29.38	* 0.88	*0.88

5 Conclusion

In this paper, we aim to re-identify grocery when it is put into or taken from a refrigerator in order to automatically monitor grocery inventory. We proposed using the load balance feature on a shelf in the refrigerator to reflect both the weight and the position of grocery items. We demonstrated that the system can re-identify groceries using the load balance feature by means of a prototype device and the two simulated experiments. The experimental results show that the load balance feature allows fairly accurate grocery re-identification when grocery items are put on the load-sensing board in neat rows. It also shows that some grocery items could be incorrectly identified as others even with the load feature, when one of the grocery items having similar weight to another is stacked on it. For future studies, we intend to investigate more complex situations encountered in real-world usage, and we will consider using appearance as an additional feature for grocery re-identification.

ACKNOWLEDGEMENTS

This work was partly supported by KAKENHI (23500137,70166099).

REFERENCES

[1] X. Wang, G. Doretto, T. Sebastian, J. Rittscher and P. Tu: “Shape and Appearance Context Modeling”, Proc. of ICCV, pp.1-8 (2007).

[2] S. Bak, E. Corvee, F. Bremond and M. Thonnat: “Person Re-identification Using Spatial Covariance Regions of Human Body Parts”, Proc. of AVSS, pp. 435-440 (2010)

[3] W.-S. Zheng, S. Gong and T. Xiang: “Associating Groups of People”, Proc. of BMVC (2009)

[4] Schmidt, M., Strohbach, K., Van Laerhoven, A., Friday, H.-W.G.: Context Acquisition based on Load Sensing. In: Proc. Ubicomp 2002: the 4th international conference on Ubiquitous Computing (2002)

[5] Pei-yu Chi, Jen-hao Chen, Hao-hua Chu, and Jin-ling Lo, “Enabling calorie-aware cooking in a smart kitchen,” Persuasive2008, Oulu, Finland, June 2008.

[6] The instruction of load sensing device AS-FORCE supplied by AsakusaGiken Co., LTD. (in Japanese). http://www.robotsfx.com/robot/ForceSen.html