Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 60 (2015) 870 – 880

Procedia Computer Science

19th International Conference in Knowledge Based and Intelligent Information and Engineering Systems - KES2015

A Mobile Robot for Fall Detection for Elderly-Care

Takuma Sumiya[a], Yutaka Matsubara[b], Miyuki Nakano[a], Midori Sugaya[a]

[a]*Shibaura Institute of Technology*
[b]*Nagoya University*

Abstract

In 2015, the population of people over the age of 65 is 25.0% in Japan. This means that Japan has already become a super-aging society. In such society, the number of elderly people living alone has been also increased. For such people, a fall accident is serious because it can lead to serious injury or death. Researches and services to monitor behaviours of such people have been proposed. For example, by monitoring the status of use of home appliances, something unusual happened to them can be predicted. However, such systems cannot recognize the detailed behaviours like fall. Surveillance cameras have been introduced only outside the house because of the privacy issues.

In this paper, we propose a mobile robot to detect human fall and report it to their observers. The mobile robot consists of a household mobile robot (Yujin Robot's Kobuki), a sensor (Microsoft's Kinect), and a computer (PC) to detect a human and control the robot. For simplicity of the robot and accurate fall detection, the sensor is installed on the robot to follow the target harmoniously. Thus, the sensor can move around with the robot to minimize blind area. The results of our experiments show that improvement of up to 80% in fall detection rate compared to a conventional monitoring technique using position-fixed sensors. Finally, we discuss the capabilities and future works of the robot.

keyword:mobile robot, welfare, human detection;

1. Introduction

In 2015, the elderly population in Japan is 25.0%. This means that Japan has become a super-aged society obviously. One of the serious problems for the elderly people is fall accident at home. The 25.3% of people over the age of 85 have any experiences related to fall accidents within a year. The 62.5% of fall for elder people cause their injury [1]. The number of elderly living alone has also been also growing [2].

Midori Sugaya. Tel.: +03-5859-8517 .
E-mail address: doly@shibaura-it.ac.jp.

Peer-review under responsibility of KES International
doi:10.1016/j.procs.2015.08.250

In the case that nobody is around an elderly living alone, then his/her fall cannot be known. This can cause serious results such as life-threatening or even result in death. To avoid this situation, services to detect the fall and notify it to observers his/her family, relatives, supporters or rescuers, are required immediately.

Several studies have proposed to predict the status of elderly people by monitoring the status of use of home appliances [3,4,5]. However, such systems are not adaptable for changing in daily life activities. It is difficult to detect falls or understand the elderly's expressions in real-time [3,4]. Surveillance cameras have been used in public place. However, they have not spread at home due to privacy concerns. Wearable devices, such as a ring, a pendant, a watch, and so on, may be able to detect falls. However, these devices demand the elderly to always wear (if needed, the elderly need to charge the battery of the devices). Researches in robot therapy have exposed that human-friendly robots do not place the strain on users [7]. In recent years, robots for household use have become cheaper, readily available and commercialized. For instance, robots for floor cleaning have been realized. In [17], the proposed household mobile robot moves autonomously.

In this paper, we propose a mobile robot to detect fall of elderly for the purpose of life-care and report it to their observers. The robot follows the elderly from certain distances in the home, detect fall of the elderly in real-time and send e-mails to observers to notify the fall if happen. With the notification, the family, the helpers, and rescuers can contact each other immediately. The mobile robot consists of a household mobile robot (Yujin Robot's Kobuki), a sensor (Microsoft's Kinect), and a computer (PC) to detect a target and control the robot. In fall detection, software running on the PC acquires skeletal information of the detected human from Kinect and determines fall by using the differences between the head and the knees of the target. For simplicity of the robot and accurate fall detection, the sensor is installed on the robot to follow the target harmoniously. Thus, the sensor can move around with the robot to minimize blind area. The results of our experiments show that improvement of up to 80% in fall detection rate compared to a conventional monitoring technique using position-fixed sensors. Finally, we discuss the capabilities and future works of the robot.

This paper is organized as follows: in Section 2 we introduce the existing monitoring systems. In Section 3 and Section 4, we show present the fall detection algorithm and design of the proposed mobile robot. In Section 5, we describe the preliminary experiments and improvements. In Section 6, we discussion the capabilities and future works of the robot. Finally, in Section 7, we conclude this paper.

2. Related research

Currently proposed monitoring systems can be roughly classified into the following two types: emergency calls and the assurance of safety. A typical example of an emergency call is a system in which a fall is detected in a person wearing a pendant device and then authorities are notified [3]. Thus, regardless of the state of the people who fell, it is possible to notify emergency authorities automatically. However, it is necessary to have the device attached at all times, and this is often considered troublesome to users.

Meanwhile, as an example of the assurance of safety, a method of using every day devices, such as household appliances, has been proposed [4]. We can check whether the elderly are performing everyday activities or not by getting information on how frequently they use a hot water pot. Although the elderly don't have to wear a device with this method, checking their movements, including fall detection, is difficult. Besides these methods, mechanisms of safety confirmation using everyday devices have also been proposed, including the use of the bath and stove [3]; however, there are similar problems with these as well.

It is necessary to accurately monitor the elderly's movement to prevent falls, which are a serious problem. Image processing using cameras enables such analysis. In fact, we can view images from mobile phones or PCs by setting a camera in the room. Methods of checking more detailed information have been proposed [5]. However, this system has a problem in that fixed cameras have blind spots. To solve this problem, the installation of a lot of cameras may be necessary. However, there are some additional problems, including invasion of privacy and an increase in stress for the elderly, so this method has not yet prevailed.

3. Proposal

3.1. Purpose of research

In this study, we focus on falls in the elderly and create a system that uses a household mobile robot to accurately detect someone falling near the robot, and then send word of the fall to someone who can help. In particular, we overcome the problems, including the invasion of privacy and the trouble that the elderly must wear a sensor at all times.To realize this, we take a household mobile robot and add a function to track the elderly from a certain distance to it. Positive findings have been reported regarding conversation with robots familiar to them [7]. This time, we add only the tracking function, but we would like to add conversation features in the future. Also, we use a depth sensor for detecting falls. The depth sensor is used to measure the condition and distance of the body of the elderly person. As it gets information such as facial expressions, it does not invade their privacy.

3.2. Monitor system Overview

Our Monitor system is a system in which a mobile robot tracks a subject from a certain distance and checks the subject for falls. If it detects a fall, it sends notification to the subject's relatives by e-mail. Figure 1 shows the overall outline of the system.

3.3. Hardware configuration

We explain the mobile robot and the hardware configuration of the monitor system. We use the mobile robot (Kobuki) [8], which is the same model as a circular vacuum cleaner robot widely used in homes. Kobuki is lightweight and equipped with wheels and bumpers. USB is used for serial communication. Since the connection to a PC is easy, we thought this type was suitable for development. In addition, we used Kinect sensors [9] to monitor the behaviours of the elderly, which enabled us to know the distance between the robot and the subject. Kinect can calculate the depth of each point on the screen by triangulation, using the reflection of the projected infrared laser. Thus, we could detect the distance in 3D. Unlike the image recognition from a camera, these characteristics enabled us to detect detailed body movements. In the present study, we mounted a Kinect sensor to the robot and then configured hardware to be compatible with it. Figure 2 shows the watch robot developed in this study.

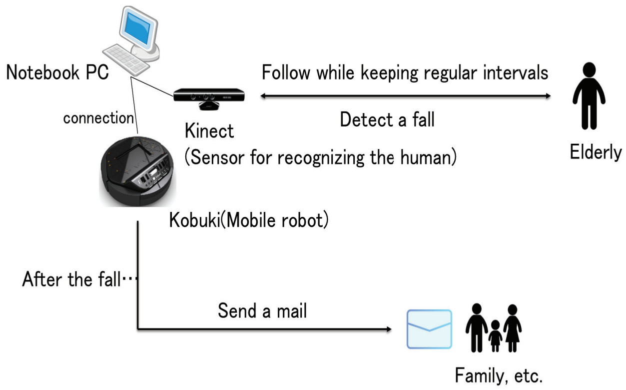

Figure 1. Tracking System

Figure 2. Watch robot Figure

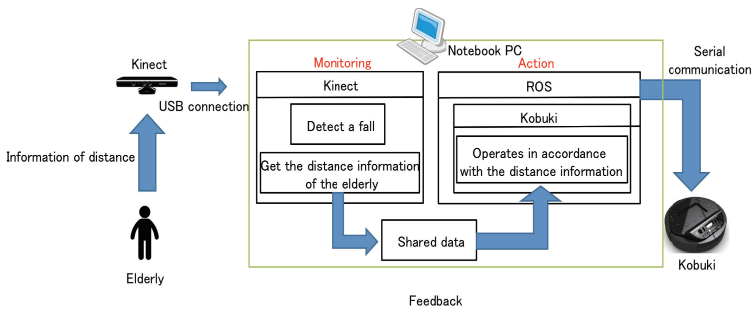

Figure 3. Software

3.4. Software configuration

For the software configuration, we develop a tracking method using the data from the Kinect sensors and a fall detection method using the skeletal information from the Kinect sensors. In particular, we design two functions. One is a monitoring function that gets input data from the sensor. The other is an action function that works based on the information of the input data. They operate cooperatively by sharing the data via a mechanism of sharing information we created. The software configuration diagram is shown in Figure 3.

3.5. Fall detection method

We develop a technique for performing fall detection by using a mobile robot as a monitor system. First, in order to detect a fall, we develop a method for figuring out the movements of subjects. Kinect can display the skeletal information of a people on the screen coordinate system (X. Y.). Thus, we focus on the skeletal information and perform fall detection based on the difference of the Y coordinates of the head and both knees. First, we set the difference between the Y coordinates of the head and both knees as θ, to perform fall detection based on the differences of the θ values. When the θ value falls below a certain threshold value, we determine it as a fall. We then perform this simple experiment (Figure 4).

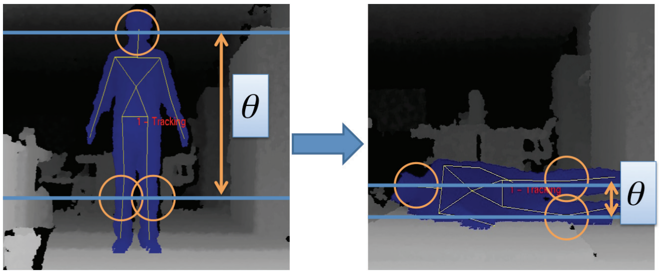

Figure 4. Fall determination using the difference between the head and the knees

The results of the experiment show that with this method, the movement of bending down, including to pick up something, is mistakenly determined as a fall. For the movement of bending (for example, picking up something), the amount of time that the θ value is under the threshold is shorter. In the case of a fall, except for minor falls, the amount of time that the θ value is under the threshold is longer. Therefore, we decide to consider the amount of time to distinguish the movement of bending down from falling. The results are shown in the fall detection system flow chart in Figure 5.

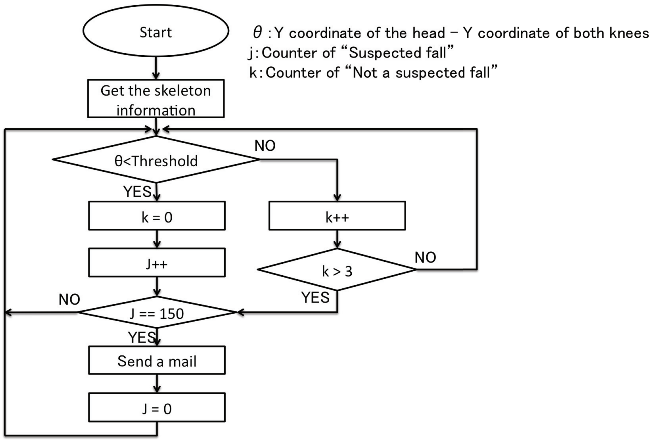

Figure 5. Process flow chart of fall detection judgment

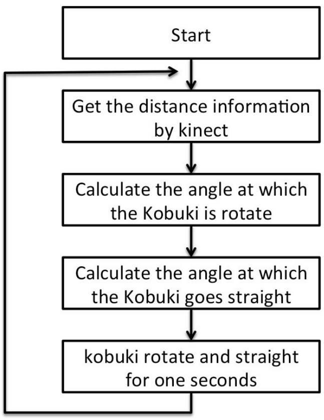

Figure 6. Flow chart of the tracking

When the differences of the Y coordinates of the person's head and both knees are smaller than the threshold value, the state is set as "Suspected fall" and when the differences are equal or bigger, the state is set as "Not a suspected fall" If the "suspected fall" state continued for 150 counts (about 5 seconds), then it is determined to be a fall. The E-Mail is sent and the counter is reset. On the contrary, if the state of "Not a suspected fall" continued for 10 counts (about 0.3 seconds) during the 150 counts of a suspected fall, then "Not a suspected fall" is determined instead of a fall. The non-suspected fall counter is reset. Thus, we can distinguish falling movements from bending movements.

3.6. The tracking method by the mobile robot

In this study, tracking is the primary operation of the robot. Tracking is defined here as the robot following the subject from behind at a fixed distance. Existing studies on robot therapy shows that a robot moving and approaching an elderly person is effective in reducing their stress [7]. Since communication with robots is not a primary issue in this paper, first, we simply aimed to get a robot to track a subject at a fixed distance.

In order to track a subject, a mobile robot needs to move while maintaining a certain distance from the subject, who is moving constantly. It needs to periodically measure the distance from the subject before making a move. We examine the following method: the mobile robot periodically measures the distance from the subject using the Kinect sensors attached to itself, and the robot moves based on that information.

Figure 6 shows a flow chart of the tracking process. For that purpose, we set a fixed distance "X" between the robot and the subject. Next, "X1" is set as the distance obtained from the sensor after one minute. Then, the robot rotates its wheels based on the differences between X1 and X (X1-X). It determines the point where X1-X became zero and moved towards that direction. The robot repeated this operation until the user stopped it.

The values on the two-dimensional coordinates obtained from Kinect (X, Y) and the following values (X1, Y1) are not always on a line, it sometimes had a certain angle. In this case, by using the triangle theorem, the angle θ, which is the deviation of the directions between the mobile robot and the subject, is calculated. Then, the robot rotates and move to the direction.

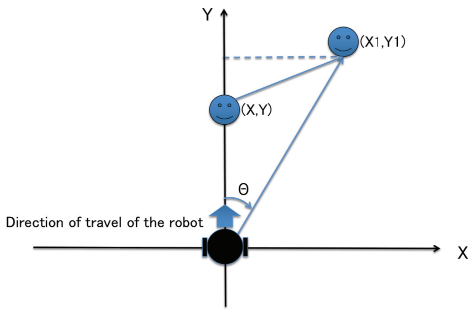

Figure 7. Calculation of θ

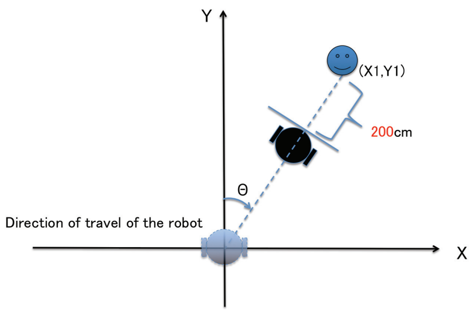

Figure 8. Speed settings

We take the angular velocity that the robot rotates in one second as the calculated angle θ, then we set the moving speed for 1 second directly towards that direction while keeping a certain distance from the subject, and then we move the robot for one second. By repeating this, it achieved proper tracking (Figure 8).

4. Implementation

In order to create the monitor system, we establish a new development environment with Linux (Linux 3.5.0-17-generic) and ROS(ver. Groovy Galapagos) [10]. As an environment for the development of the mobile robot, we use ROS. ROS is middleware for software developers and provides hardware abstraction, device drivers, libraries, visualization tools, message communication and package management as a library and tools to support robot application creation. Moreover, in order to operate the Kinect sensors, it has an interface that integrates a Device Section, which controls OpenNI Kinect library (ver. 1.5.4. t) devices, and a Middleware

Section, which performs image processing and gesture recognition.In order to test the operation of the monitor system with this prototype system, we need to enable it to operate from a remote location. To operate from a remote location, it is also necessary to communicate between the PC in the robot and the operating a PC, thus, we developed an IXM (Information Exchange Middleware) to integrate these two [11]. As the development of the IXM is not the theme in this paper, details are omitted. In the implementation, file sharing is conducted in order to coordinate the data of the tracking module shown and the fall detection module in the previous section, and the data is exchanged via these files.

5. Evaluation

5.1. Preliminary experiments for fall detection

To create the fall detection system, we conducted preliminary experiments to examine the installation position of the Kinect sensor. First, the distance from the person was set to 2m (200cm) in the preliminary experiments, and the height of the subject was fixed so that we could investigate the optimal height. We conducted an experiment where the height of the Kinect sensor was changed from 20cm to 80cm in 20 cm increments.

We used several falling patterns and investigated whether or not there are differences in the detection rates in those patterns. First, we tested several fall patterns (front, rear, left and right), running an experiment ten times for each pattern. We measured the detection rate (detection / (non-detection + detection)) and the period of time from when the subject's face was on the floor until the fall was judged (if it took longer than 10 seconds it was regarded as non-detection). We conducted a test for four fall patterns (front, rear, right and left), and recorded the average of 10 time measurements (Figure. 10).

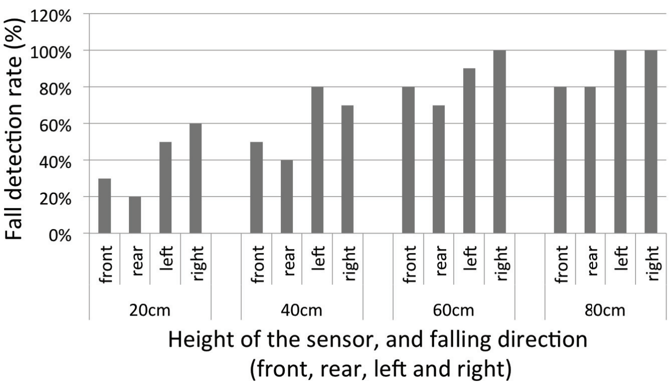

Figure 10. Fall detection rates for different heights

Table 1. Time taken to detect a fall

The height of the Kinect sensor	Time taken to detect fall (seconds)
20cm	6.3
40cm	5.7
60cm	5.3
80cm	5.2

When the height was from 20cm to 40cm, there was a tendency for the detection rate to be lower. This is because when the position of the sensor is low, the skeletal information overlapped at the time of a fall. Thus, there was a problem in which it was difficult to obtain the correct values. In the cases of 60cm to 80cm, the average fall detection rate was 87.5%, which was high. In addition, fall detection rates for left and right falls was better than that of front and rear falls. This is because in front and rear falls the subject's head was behind the body, so when viewed from the sensor, the skeletal information was not acquired. There was a limit on the algorithm that determines the differences between the coordinates of the head and the knees. Table 1 summarizes the time it takes to fall. The results showed that there was no significant difference in the time taken to detect a fall with the design that takes time into consideration. When someone is bending down, if the "suspicion of a fall" does not continue for 5 seconds, it will not be regarded as a fall, thus it was able to accurately distinguish between bending and falling. However, as the action of bending down varies from person to person, there need to be studies on these differences among individuals, among other things.

5.2. An experiment to detect falls while tracking

Next, instead of from a fixed position, we installed the Kinect sensor on top of a mobile robot, and performed a experiment to detect falls while the robot was tracking someone. We installed the Kinect sensor at a height of 60cm and conducted an experiment to detect falls in a person being tracked. In the previous experiment, having the Kinect sensor at a higher position resulted in a good detection rate as the view avoided the overlapping of the head and body. Normally, it would be attached at 80cm; however the Kobuki only weighs around 1kg and attaching the 600g Kinect at such a height would raise its center of gravity, making it unstable. Therefore, we decided to attach it at a height of 60cm, the highest possible without compromising stability. In the experiment, we kept the distance from the person at 100cm, and we examined the tracking accuracy when the person moved in a straight line at 10~70cm/s in the robot's frontal direction, and in a direction at 30° from the center line. In addition, after passing in a straight line, the subject fell in both vertical and horizontal patterns and the fall detection rate was investigated. The results of the experiment are shown in Tables 2 and 3. It would be considered a failure if the Kinect sensor was unable to recognize a fall even once. As shown in Table 2, tracking and fall detection for a person walking at speeds from 10cm/s to 30cm/s in the frontal direction, were both 100%. Some variation was seen in the results at speeds above 40cm/s. In addition, at a direction of 30 °, some variation was seen in the overall results; in that the fall detection rates in the vertical and horizontal patterns were lower. When a subject was walking at 10cm/s, the tracking rate was 40% when the subject was moving in a straight line, and it was 30% when the subject was moving in a direction at 30 ° from the center line. When the walking speed was 20cm/s or higher, both rates were 0%. Moreover, the results of our fall detection experiments were very low, from 0% to 20%. When the walking speed was 20cm/s or higher, both rates were 0%. Moreover, the results of our fall detection experiments were very low, from 0% to 20%.

Table 2. Tracking and fall detection(front)

	Following	Fall detection (vertical)	Fall detection (horizontal)
10cm/s	100%	100%	100%
20cm/s	100%	100%	100%
30cm/s	100%	100%	100%
40cm/s	80%	40%	60%
50cm/s	80%	40%	60%
60cm/s	100%	0%	0%
70cm/s	40%	20%	0%

Table 3. Tracking and fall detection(direction at 30°)

	Following	Fall detection (vertical)	Fall detection (horizontal)
10cm/s	100%	40%	40%
20cm/s	80%	20%	20%
30cm/s	100%	0%	20%
40cm/s	100%	0%	20%
50cm/s	80%	0%	40%
60cm/s	80%	0%	0%
70cm/s	80%	0%	0%

5.3. The decrease in the fall detection rate due to falls in different directions is a problem

We conducted experiments dividing 360-degrees into 8 segments in order to investigate the relationship between the fall detection rate and direction. The results showed the reason why detection rates were low in certain directions. It was because the head coordinate, which is required for fall detection, was outside the scope of the Kinect sensor's sensing; so fall detection was impossible.

5.4. Dealing with problems with Active Sensing

We investigated how to deal with the problem. First, if the sensor was assumed to be in a fixed position (Static Sensing), then improvement of the algorithm would have been required. Or in order to eliminate the blind spots in the sensor, adding another sensor would have helped. However, the sensor was attached on a mobile robot in this study, so it was possible to freely change the position of the sensor. We call this propriety Active Sensing. We thought that we would be able to perform fall detection with high accuracy, and without changing the algorithm of the sensor, by enabling the robot carrying the sensor to move. Upon implementation, we decided to add an algorithm to the sensor, which enabled it to estimate a person's direction and falling direction after losing sight of the subject. The flowchart of the improvement with these considerations is shown in Figure 12.

In the flowchart, when the robot loses sight of the subject (User Lost), the robot rotates itself and checks whether or not the subject is lost.

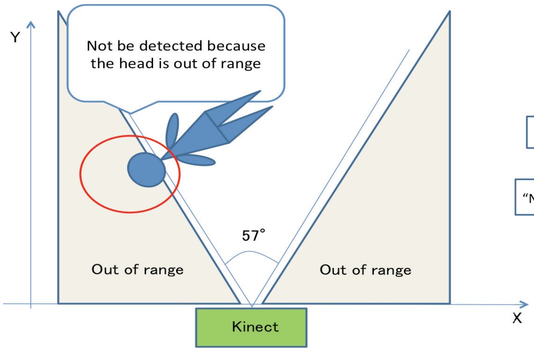

Figure 11. The scope where fall detection is impossible

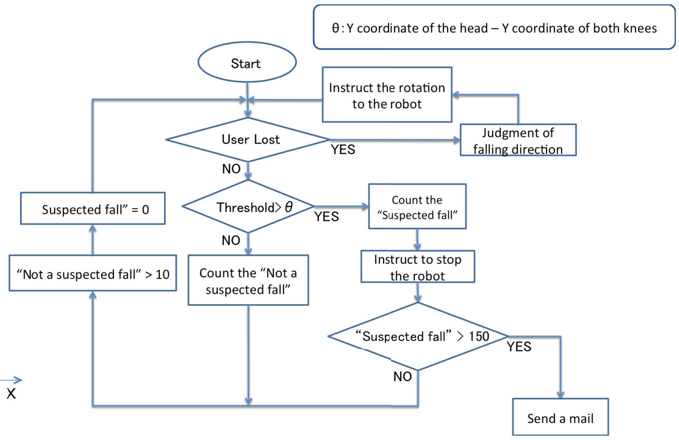

Figure 12. Flowchart on post-improvement results

5.5. Evaluation of the fall detection rate

We are going to evaluate the post-improvement system. First, we investigated the optimal fixed distance for tracking, and conducted a survey of walking speed. In light of those results, we conducted a survey of fall detection judgment in the post-improvement system, in 360 degrees and all directions. We are going to present the investigation results.

5.5.1. Optimal fixed distance for tracking

We changed the fixed distance from the subject from 80 cm to 400 cm, which was the sensing range of the Kinect sensor, to conduct research. The results are summarized in Figure 14. The figure shows, when the fixed distance was 150cm, the fall detection rate was the highest, and when further than that, the detection rate decreased. Therefore, the best way would be for the fixed distance from a subject to be set to around the height of a subject. In the evaluation experiment we set the value of the fixed distance to between 150cm and 200cm.

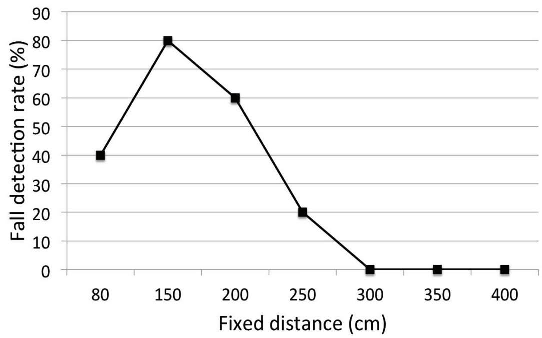

Figure 13. Fall detection rate (Different distance)

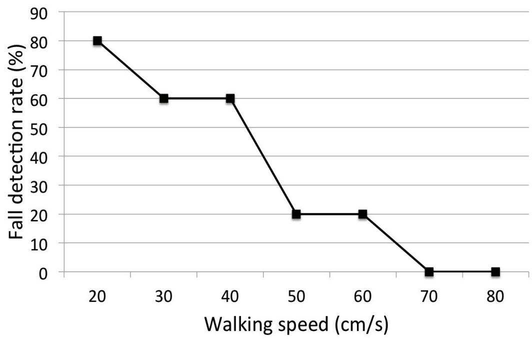

Figure 14. Fall detection rates (Different walking speed)

5.5.2. Changes in the fall detection rate in accordance with walking speed

Next, we investigated the changes in the fall detection rate in accordance with walking speed. We set the walking speed between 20 - 80cm/s, since the average outdoor walking speed of someone who is 75 years or older is 70cm/sec, so The results are summarized in the graph (Figure 14). As the results show, the fall detection rate at 20 - 40cm/s was from 60% to 80%, which was ideal. This is because when walking speed is low, the robot also moves slowly and with less fundamental vibration; thus detection rates were higher. On the contrary, with speeds of 50cm/s or higher, detection by the sensors was more difficult due to processing for smooth acceleration not being added to the Kobuki, which caused vibrations to occur in some parts. Such physical impact due to insufficient ability of the mobile robot to adjust should be eliminated in future designs as much as possible.

5.5.3. Re-evaluation of 360 degree fall detection

After the Active Sensing functions were added, we decided to evaluate under the following conditions.

- The experiment would take place on a flat place without obstructions
- The height of the Kinect camera on the watch robot would be 60cm from the floor
- The height of the subject would be 170cm with a walking speed of 20cm/sec.
- We divided 360 degrees into 8 directions and had the subject fall five times in each direction (the subject is moving towards (1) in the Figure15)

Evaluation of fall detection: we determined the detection rate as (detection / (non-detection + detection)), the same as before. If it took longer than 10 seconds to detect the fall, it was regarded as non-detection.

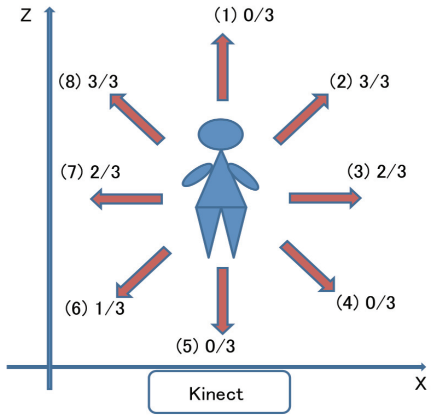

Figure 15. Fall detection directions Figure

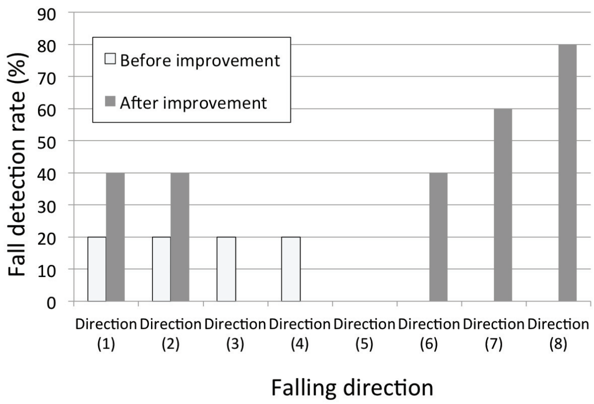

Figure 16. Comparison of fall detection rates

Figure 16 shows the a comparison of the results of the fall detection rates in (1) ~ (8). The fall detection rates significantly improved except in directions (3), (4), and (5). Although it was not successful with all of the falling directions, due to the improvement of the algorithms, the fall detection range was broader than that of conventional algorithms. Both directions (1) and (2) increased from 20 percent to 40 percent, directions (6), (7), (8) increased from 0% to 40%, 60% and 80% respectively. As shown above, we can say the overall fall detection rates have improved. When the robot lost a fallen subject, it determined the direction of the fall and turned towards the fallen subject. Due to this ability to relocate the subject, fall detection rates improved. However, as it was difficult for the robot to locate a subject who had already fallen; it sometimes failed to detect a fall. In addition, detection rates for directions (3), (4) and (5) were very low. In the case of directions (3) and (4), when the watch robot was tracking a subject, it deviated to the left and lost sight of the subject. We came up with two possibilities for the cause of this problem. One was a problem with the motor characteristics of the mobile robot, Kobuki. The other one was due to a shift to the center of gravity in the whole watch robot. Since the PC on the robot was not fixed, the location of the PC changed in accordance with the movements of Kobuki. There was also a problem in the structure of the robot. In direction (5), when the 170cm tall subject fell backwards, the distance between the head coordinate and the camera became almost 0 cm, so it could not obtain the skeletal information. We believed that was the reason for the low fall detection rates. It will be necessary to consider additional methods to further improve fall detection rates, particularly for directions with the lowest rates, like direction (5) which was the backwards fall.

6. Discussion

In this study, we explained the method we used to create the robot's ability to operate the sensor by itself, thus creating our mobile watch robot. Upon implementation of the watch robot, we held various discussions. We summarized the related discussions with the following points:

(1) The Necessity for consideration of power consumption: there was a problem in that the watch robot was always consuming power because the base of its movement lied in tracking. In fact, our original algorithm was made so that the robot tracked even when a subject moved just a little bit. Results of the investigation showed that when the distance to move was short, the time required for moving was prolonged. This is because when the speed setting was low, an insufficient amount of torque was generated for the motion through the rotation of the motor, due to low voltage supply and insufficient current. Friction between the mobile robot and the ground due to the weight of the robot caused difficulty in advancing; thus resulting in extra time. During this time, weak current continued to flow and it wasted power. Thus, we created a scenario in which the robot moved straight forward from a stationary state, and we added a processing program in which the robot does not engage in tracking when the difference between the original distance of the robot and the subject and next distance between them is less than 30mm. As a result, during the 60 minute measurement period, we were able to reduce about 0.1V voltage, between the 50 to 60 minute mark [12]. In addition to monitoring falls in the elderly, other settings and services are possible. We also need to consider the power consumption. For example, we may need to prolong the detection interval when a subject stops for a long period of time.

(2) The necessity of a platform technology: Upon developing the robot, different software was necessary when we were developing the robot control software, depending on the device being used. We need to integrate them and use multiple pieces of software cooperatively. We developed middleware for the integration and cooperation of multiple kinds of software in order to cope with this problem [13]. Among them, regarding coordination, it was pointed out that there were lots of non-functional requirements, including real-time performance and reliability in processing; however, we have not fully dealt with these issues and ongoing research will be required.

(3) The necessity of a classification as information equipment: in the future, an expansion of the variety of functions for our service will be required in order to watch a subject at home, including obtaining biological information of the elderly person being monitored. At that time, household robots will be connected to a network, and play a role as household information terminals. We developed a medium-sized server-client system with a robot as a client. We used it to collect information and then we investigated the effectiveness of the system [14]. In the future, when the information the sensor is collecting increases, we will need to divide the information path into a channel for collecting information and a channel for sending commands from a remote location. We will further need to develop a mechanism for executing instructions without hindering performance.

(4) **The necessity for a service switching mechanism:** it is desirable that some indication is sent to the robot from a remote location. However, in addition to sending instructions on a case by case basis, for autonomous operation, a mechanism for rewriting the program by itself is also important. In particular, depending on various conditions, the robot will be required to flexibly determine how to proceed on its own, so it will be important to perform these operations in accordance with context information. When the robot is switching between services, techniques for switching services based on context-oriented programming [15] have already been proposed [16]. Consideration of these techniques will be further required.

7. Summary

In this study, we proposed a mobile robot as a home monitoring system, attempting to improve upon conventional devices. In our attempts to detect falls with a fixed sensor in our preliminary experiment, it was difficult to detect falls in 360-degrees; however, by implementing a function that actively moves the sensor

(Active Sensing), we achieved an 80% improvement in the detection rate. However, we discovered that the method for re-detection of a user who had fallen once was not sufficient, and that there are directions in which falls cannot be detected due to individual differences in robots, and dealing with these issues has been recognized as important. Moreover, we discussed problems relating to the technology required for monitoring-robots. We will continue to address these technical issues in the future.

References

[1] Cabinet Office: The Aging Society: Current Situation and Implementation Measures Annual Report, 2013

[2] Ministry of Health, Labour and Welfare: Comprehensive Survey of Living Conditions, 2012

[3] Osaka Gas Security Service: Omamori Call,
http://www. oss-og. co. jp/service/omamori/index. html

[4] Zojirushi: Mimamori Hotline,
http://www. mimamori. net/service/index. html

[5] Home-Eye: the Kansai Electric Power Group,
http://eonet. jp/home-eye

[6] Philips: http://www. hmservice. philips. co. jp/index． html

[7] Toshimitsu Hamada, Hiroki Okubo and Hisashi Onari: Study on robot-therapy and its implementation method for the elderly

[8] YUJIN ROBOT: Kobuki, http://kobuki. yujinrobot. com/home-en

[9] Microsoft: Kinect, http://www. xbox. com/ja-JP/Kinecthttp://www.xbox.com/ja-JP/Kinect

[10] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, Andrew Ng: ROS: an open-source Robot Operating System.

[11] Takuma Sumiya, Midori Sugaya: The 76th National Convention of IPSJ Proceedings, Proposal of functional integration system for realizing multi-purpose robots, 1, 2014, 81-83.

[12] Atsunori Nagamine, Yutaka Matsubara, Midori Sugaya: The 77th National Convention of IPSJ Proceedings, Consideration of power saving in the Watch Robot, 5J-08, Kyoto, Mar 2015.

[13] Takuma Sumiya, Yutaka Matsubara, Miyuki Nakano, Midori Sugaya: A robot middleware for supporting real-time systems and power saving, Computer System Symposium Proceedings, 2014, 36-43.

[14] Satoru Nakayama, Takuma Sumiya, Miyuki Nakano, Midori Sugaya: RLS the data collection platform of the distributed robotic system, Information Processing Society of Japan Research Report. EMB, Embedded system 2015-EMB-36 (45), 1-6, 2015-02-27

[15] R. Hirschfeld, P. Costanza and O. Nierstrasz: Context-oriented Programming, Journal of Object Technology, Vol. 7, No. 3, pp. 125-151, 2008.

[16] Ikuta Tanigawa, Nobuhiko Ogura, Midori Sugaya, Harumi Watanabe, Kenji Hisazumi : A Structure of a C# Framework ContextCS Based on Context-Oriented Programming, Modularity '15, 2015.

[17] James Everett Young, Youichi Kamiyama, Juliane Reichenbach, Takeo Igarashi, Ehud Sharlin. How to Walk a Robot: A Dog-Leash Human-Robot Interface. 20th IEEE International Symposium on Robot and Human Interactive Communication (Ro-Man 2011), pp.376-382. 2011.