Autonomous Monitoring Framework with Fallen Person Pose Estimation and Vital Sign Detection

Igi Ardiyanto, Junji Satake, and Jun Miura
Department of Computer Science and Engineering
Toyohashi Unversity of Technology
Aichi, Japan
Email: {iardiyanto, jun}@aisl.cs.tut.ac.jp

***Abstract*—This paper describes a monitoring system based on the cooperation of a surveillance sensor and a mobile robot. Using a depth camera which acts as the surveillance sensor, the system estimates the pose and orientation of a person utilizing a skeleton-based algorithm. When the person fell down, the sensor sends the person's pose and orientation information to the mobile robot. The robot determines the possible movements and strategies for reaching the fallen person. The robot then approaches the person and checks the vital condition whether the person is breathing, and the recognition result is notified to a hand-held device. Experiments on our monitoring system confirm a successful series of the autonomous operations.**

I. Introduction

In Japan, aging society has become a challenge where the number of elder populations are more than the young generations. Such condition has raised many problems. We take an example in a nursing home which has unbalanced number of the staff in charges and the staying elder persons. The staff may not be able to continuously watch each elder person.

Due to the physical limitations, there are some occasions where an elder person fell down in the daily life at the nursing home. While the fell down cases may only cause a minor effect to the young people, for the elderly it can give a serious injury. The staff of the nursing home may fail to notice the fallen elder person immediately, whereas a late handling of these situations can lead to collateral or even major damages to the elder person.

Many works and systems have been proposed for coping with the fallen person problem. The works mainly can be divided into two approaches, intrusive and non-intrusive [1]. The intrusive approach demands the elderly for wearing a device, such as an accelerometer [2], to detect the fall down event. This approach becomes troublesome, especially for the elderly with dementia symptom [1] which may forget to wear or store the device.

As the opposite, the non-intrusive approach uses external sensors to detect the fall down event, including some works using multiple cameras [3] or depth data (*e.g.* [4], [5], and [6]). Unfortunately, none of these works considers a further applied action for the fallen person, except just notifies the other persons via sound alarm such as in [3]. Another drawback of these works is the lack of information accommodated by their systems, where they do not even provide the fallen person position data relative to the environment ([2], [3], and [6]).

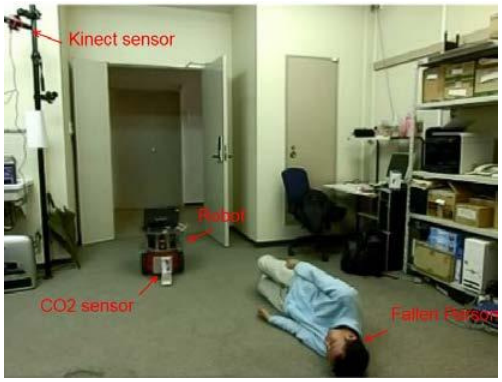

Fig. 1: Monitoring system framework with a surveillance sensor and an autonomous robot.

For dealing with such shortcoming, here we propose a comprehensive monitoring system employing the cooperation between a surveillance sensor and a mobile robot (see Fig. 1). A Kinect-based surveillance sensor serves as the fallen person detector. Our system estimates both pose location and orientation of the fallen person's head, enabling a further response by the autonomous robot.

The mobile robot is then engaged to check the fallen person vital and condition whenever it receives the information about the fallen event from the surveillance sensor. Here a CO_2 sensor-based breath detection is used as the vital sign. Together with the Kinect video streams, the sensor data is sent to the server real-time so that the third party person (e.g. the staff) knows what is happening and how severe the fallen person condition.

Contributions of this paper are two-fold. First, the system provides an extensive autonomous framework for an indoor monitoring, ranging from the falling person detection to the further responses (*i.e.* performs the vital measurement) after the fallen event. Therefore, unlike the aforementioned other works, our fallen person detection serves both position and orientation of the person, makes it easier for the robot to locate and do the vital measurement in response to the incident.

The rest of this paper is organized as follows. In section II we establish the system architecture of the monitoring system. A strategy for estimating the pose and orientation of the fallen person is presented in section III. Section IV explains the

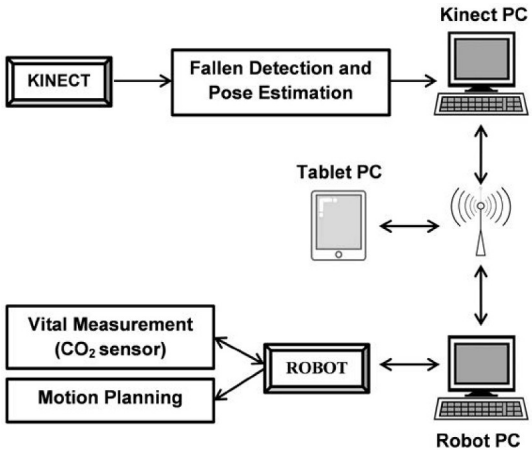

Fig. 2: Monitoring system architecture.

coordination between the surveillance sensor and the robot for measuring the vital sign of the person. We then verify the experiment results in section V. Lastly, we give the conclusion and some possible future directions of this work.

II. System Architecture

This section describes the overall framework of our monitoring system. The primary idea is to immediately take a response when a fallen incident occurs, by sending a robot to check the vital sign of the person. Here, our system is then divided into two major parts: the fallen person pose estimation, and the vital sign measurement by the robot.

The fallen person estimation is carried out by a Kinect camera which is set on the ceiling of the room. This camera is responsible to detect the person, calculate the head position and its orientation, and then send the information to the robot. An accurate estimation of the person's head location and orientation becomes inevitable, especially for locating the nose, as we use the person's breath as the vital sign. The detail of these processes will be further explained in the section III.

Subsequently, a robot which is stood by a certain location (it may be located at a different room) and equipped by a laser range and a CO_2 sensor, will use the information from the Kinect camera to navigate towards the fallen person. The goal is to put the CO_2 sensor attached in front of the robot body as close as possible to the head of the fallen person for measuring the breath. A sequence of motion strategy, which will be described in the section IV, is then performed by the robot to realize the task above.

Both parts interchange the data through a server using socket-based communication. The entire data of the fallen person and vital sign, including the CO_2 sensor reading, video streams from the camera, and the robot state, can be accessed using a hand-held device (*e.g.* smart phone and tablet) via websocket protocol [7], to be used by the third-party person. Figure 2 shows the architecture of our monitoring system.

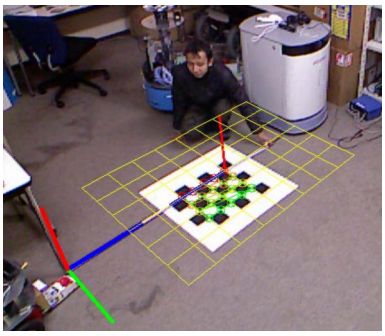

Fig. 3: Calibrating the camera. The red, blue, and green lines show the origin coordinate of the world frame. The yellow grids are the estimated ground plane (it will be translated to the origin and scaled in the experiments).

III. Fallen Person Pose Estimation using A Depth Camera

Unlike the other works on the fallen person detection which only care whether there is any fallen person event or not, our main concern is to precisely estimate the person location and its orientation. It is a compulsory requirement as the information will be used by an autonomous mobile robot for further action (*i.e.* measures the person's breath). Here we propose a Kinect-based person estimation system for solving those problems.

A. Calibrating Parameter of the Kinect Camera

The first step for achieving an accurate person pose estimation is to have a decent calibrated camera. Assuming a pinhole camera model for the Kinect, let $\mathbf{x}_w = \{x_w, y_w, z_w\}$ and $\mathbf{x}_c = \{x_c, y_c, z_c\}$ be a 3D point coordinate in the world frame and its corresponding coordinate in the camera space. Transformation of both frames are given by

$$\mathbf{x}_c = \mathbf{R}\mathbf{x}_w + \mathbf{t}, \tag{1}$$

where $\mathbf{R}$ and $\mathbf{t}$ respectively denote the extrinsic parameters, *i.e.* the rotation matrix and translation vector.

The Kinect also holds the ordinary RGB data. By letting $\mathbf{x} = \{u, v\}$ be the coordinate of the projection point in the image, the pinhole projection can be expressed by

$$\begin{aligned} \mathbf{x} &\sim \mathbf{A}[\mathbf{R} \quad \mathbf{t}]\mathbf{x}_w, \\ \mathbf{A} &= \begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix}, \end{aligned} \tag{2}$$

where $\mathbf{A}$ is the intrinsic parameters of the camera, f_u and f_v denote the focal lengths in each axis, and c_u and c_v represent the optical center in the image plane. The skew and lens distortion are neglected in our case. The intrinsic matrix $\mathbf{A}$ is obtained using a standard camera calibration algorithm from [8]. Since the Kinect provides the depth information (*i.e.* z_c), relation between the image and depth map can be described as

$$\begin{aligned} x_c &= z_c \frac{u - c_u}{f_u}, \\ y_c &= z_c \frac{v - c_v}{f_v}. \end{aligned} \tag{3}$$

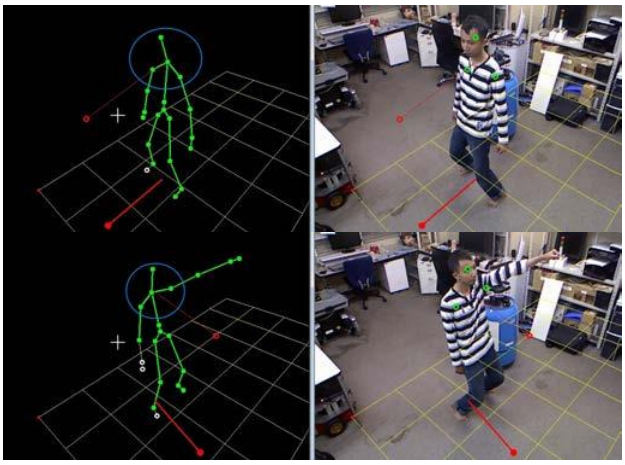

Fig. 4: Modeling the human skeleton. The yellow grids are the estimated ground plane. The thin red line represents the head pose and orientation, and its projection on the ground is shown by the bold red line. The joints inside the blue circle are used for the pose estimation.

To get the extrinsic parameters, a chessboard pattern is utilized (see Fig. 3) from which a set of corner point is then retrieved for estimating the chessboard pose and its corresponding projection in the Kinect space. This problem is solved using perspective-n-points method [9]. The obtained **R** and **t** are then used for the coordinate transformation in the pose estimation.

B. Fallen Person Detection and Its Pose Estimation

The use of the depth map has many advantages as it provides a real 3D scene representation. For example, a work from [10] shows that by using a depth map, the body skeleton of a person can be extracted in real-time. Here we adopt their work by using the skeletal extraction as a base of our fallen person detection and its pose estimation.

Given $S = \{s_0, s_1, \ldots, s_k\}$ as the skeletal joints of the body obtained by the Kinect camera when a person enters the camera view, therefore let $S_{up} = \{s_h, s_{sr}, s_{sl}\} \subset S$ be the upper head, right shoulder, and left shoulder joints respectively. In the Kinect frame, the point coordinate of each $s \in S_{up}$ is then described as $\mathbf{x}_c^s = \{x_c^s, y_c^s, z_c^s\}$. Using previously obtained **R** and **t**, $\mathbf{x}_c^s$ is projected to the world coordinate as follows

$$\mathbf{x}_w^s = \mathbf{R}^{-1}(\mathbf{x}_c^s - \mathbf{t}) \quad \text{for } \{\forall s \in S_{up}\}, \tag{4}$$

where $\mathbf{x}_w^s = \{x_w^s, y_w^s, z_w^s\}$ is the world coordinate point of each $s \in S_{up}$. Figure 4 shows the skeletal modeling of the human.

The center of the head position $\mathbf{x}_w^{ch}$ can be estimated by averaging the upper head and both shoulder joint poses,

$$\mathbf{x}_w^{ch} = \frac{1}{|S_{up}|} \sum_{\forall s \in S_{up}} \mathbf{x}_w^s, \tag{5}$$

where $|S_{up}|$ is the cardinality of the set S_{up} (hence, $|S_{up}| = 3$). The fallen event is then simply detected using the value of z_w^{ch} (the z-component of $\mathbf{x}_w^{ch}$). If z_w^{ch} less than a designated threshold, it means the head is near the ground plane and will be categorized as a fallen person event.

Fig. 5: Locating the robot goal in front of the head. The bold red dot is the robot expected goal for measuring the person's breath.

Therefore, we need to project the head pose to the ground plane (i.e. z-axis$= 0$) on which the robot moves. Another consideration is the assigned head position should hold some distances in front of the head (of course, the robot should not hit the person's head).

Let $\mathbf{x}_w^{ph} = \{x_w^{ph}, y_w^{ph}, 0\}$ denote the projected head position on the ground plane with an additional distance, on which the robot will safely stop to measure the person's breath (see Fig. 5). We then derive $\mathbf{x}_w^{ph}$ as follows

$$\mathbf{x}_w^{ph} = \mathbf{x}_w^{ch} + \frac{\delta}{||\mathbf{x}_c||}\mathbf{x}_c, \tag{6}$$

where

$$\begin{aligned} \mathbf{x}_c &= \mathbf{x}_a \times \mathbf{x}_b, \\ \mathbf{x}_a &= \mathbf{x}_w^{s_{sr}} - \mathbf{x}_w^{s_h}, \\ \mathbf{x}_b &= \mathbf{x}_w^{s_{sl}} - \mathbf{x}_w^{s_h}. \end{aligned} \tag{7}$$

$\mathbf{x}_w^{s_{sr}}$, $\mathbf{x}_w^{s_{sl}}$, and $\mathbf{x}_w^{s_h}$ are respectively the joint position of the right shoulder, left shoulder, and upper head as pointed out in eq. 4, and δ is a relative distance between the projected head pose and the expected robot target (currently, $\delta = 70$ *cm*). Accordingly, the person orientation θ is calculated by

$$\theta = \tan^{-1}\left(\frac{y_w^{ph} - y_w^{ch}}{x_w^{ph} - x_w^{ch}}\right), \tag{8}$$

where $(x_w^{ch}, y_w^{ch}) \in \mathbf{x}_w^{ch}$ and $(x_w^{ph}, y_w^{ph}) \in \mathbf{x}_w^{ph}$. Now, we have $\mathbf{x}_w^{goal} = \{x_w^{ph}, y_w^{ph}, -\theta\}$ as the given target pose for the robot to measure the person vital condition. The minus $(-)$ sign indicates the robot goal has an opposite direction to the person orientation θ.

IV. Managing the Robot Motion for Measuring the Human Vital Sign

After a fallen person has been detected by the Kinect system, a prompt response needs to be conducted for encountering the incident. Here, a mobile robot is utilized for approaching the person location and measuring the vital sign which will be the base of the next action for the victim.

As the robot is initially placed in a certain room which may differ with the fallen person location, the robot movement for accomplishing the tasks needs to be decomposed. A Finite State Machine-based robot movement strategy is then proposed for handling this problem.

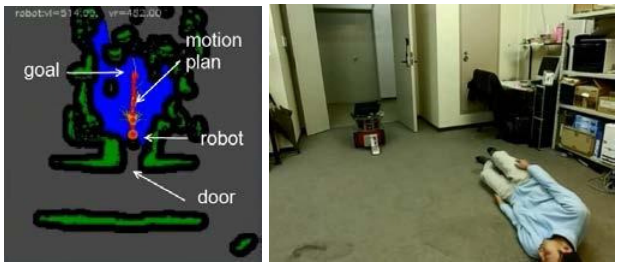

Fig. 6: An example of the robot state where the robot goes to the fallen person location. Left: the robot bird's view map and its motion planning, the blue area is the free space, the gray is unknown area, while the green and black area are the obstacle area and its extension. Right: the real world view from an observing camera.

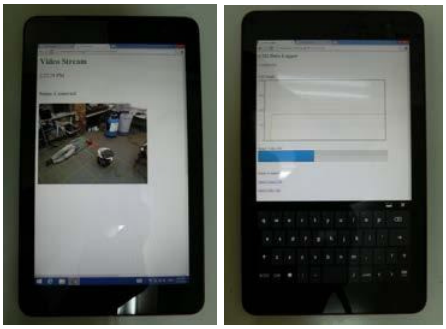

Fig. 7: Real-time data monitoring on the tablet. Left: The Kinect video streams. Right: The CO_2 sensor data.

A. Cyclic Finite State Machine

Let $Q = \{q_0, q_1, \ldots, q_n\}$ be a set of robot states. An individual state $q \in Q$ does not necessarily only represent the robot position, but also the current robot activity. For example, q_0 can be defined semantically as a state where "*the robot is waiting at the start position*", or, the state q_1 might be translated as "*the robot measures the CO_2 level at the fallen person location*".

The whole robot tasks for measuring the vital sign can be viewed as a collection of sequence of the states. A natural way for concatenating those sequences into a complete task is by making a proper transition function between the states, which will lead to the usage of Finite State Machine.

The Finite State Machine (FSM) [11] is formally given by a tuple $(\mathcal{F}, Q, q_0, \gamma, \mathcal{O})$ where:

- $\mathcal{F}$ is a set of input events $\{f_1, f_2, \ldots, f_m\}$;
- Q is a set of robot states;
- q_0 is the initial robot state;
- γ is the state transition function, $\gamma : Q^- \times \mathcal{F} \mapsto Q^+$;
- $\mathcal{O}$ a set of output events $\{o_1, o_2, \ldots, o_k\}$.

The symbol Q^- and Q^+ respectively represent the state before and after transition.

In the general form of the FSM [11], the state transition function γ is mathematically described as

$$\gamma(Q^-, \mathcal{F}) \rightarrow \{\mathcal{O}, Q^+\} \quad \text{for } \{\forall f \in \mathcal{F}, \forall q \in Q\}, \tag{9}$$

which means any set of input in $\mathcal{F}$ may lead the transition of the state q to any state in Q (including non-neighbor states and q itself) with an output event $o \in \mathcal{O}$. As it will arise a vast combination of state-to-state transition, we normally determine a finite policy for the γ mapping function.

Our system uses a cyclic FSM, which is a special form of eq. 9. The cyclic FSM utilizes monotonic transition between two state

$$\gamma(Q^-, \mathcal{F}) \rightarrow \{\mathcal{O}, Q^+\} \quad \text{for } \{Q^+ = \{q_i, q_{i+1}\}\}. \tag{10}$$

In a simple way, the cyclic FSM requires a state to make a transition either to the next consecutive state or to that state itself. We currently use five states for representing the whole tasks of the robot, decomposed as follows

- q_0, the robot is in the awaiting position;
- q_1, the robot receives the data from the Kinect;
- q_2, the robot goes to the fallen person location;
- q_3, the robot measures the vital condition using the CO_2 sensor;
- q_4, the robot goes back to the initial position.

Following eq. 10, the robot movement strategy is then described as follows

$$\gamma(q_i, f_i) \rightarrow \{o_{i+1}, \{q_i, q_{i+1}\}\} \quad \begin{cases} f_i = f_0 & \text{for } i = 0 \\ q_{i+1} = q_0 & \text{for } i = 4 \\ f_i = o_i & \text{otherwise.} \end{cases} \tag{11}$$

Here our system has only one global input event f_0 (*i.e.* triggered by the fallen person data from the Kinect), and the output events o_i of each transition will become the input for the consecutive state.

A randomized tree-based motion planner algorithm [12] is then employed for executing the movement actions of the robot (*i.e.* at q_2 and q_4). Figure 6 exhibits an example of the robot state (q_2), where it moves from the initial position to the fallen person location.

B. Measuring the Vital Sign

During the robot cyclical process above, there exists a state dedicated for measuring the person's vital condition (*i.e.* q_3). Here a CO_2 sensor is used for detecting the breath as the vital sign. The breath sensor, which is attached in front of the robot body, measures the CO_2 concentration level near the nose of the fallen person. The data of the CO_2 level is then transmitted via websocket real-time, as well as the Kinect video streams (see Fig. 7), so that it can be received remotely using a smart phone or tablet. Later, it will notify the officer or the third-party person for taking any action, such as a proper first aid treatment or even calling an ambulance.

One may argue that detecting the person's breath is not enough for examining the vital condition. Our main intention is to create a working and feasible framework for monitoring

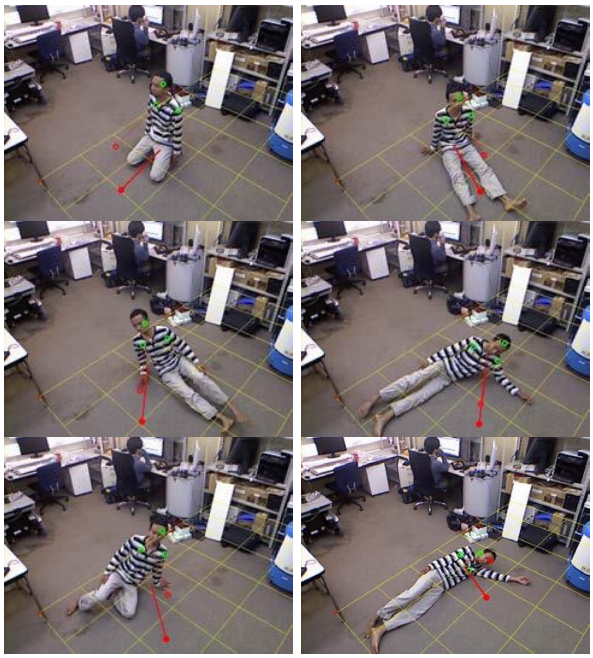

Fig. 8: Person pose and orientation estimation results.

system. Or, in other words, measuring the breath is just an example used in our architecture for detecting the vital. Additional methods for the measuring the vital sign can be easily adopted into the system. We will also give some insightful consideration about this matter at the last part of this paper.

V. Experiment Results

The proposed framework has been tested in the real world using a Pioneer-3DX robot equipped by a laser range finder and a ZMP CO_2 sensor, and a Kinect camera attached on the ceiling of a room. The implementation of the fallen person estimation is done on a Windows PC (i7 2.4 GHz, 16 GB RAM), while another PC (Core2Duo 2GHz, 2GB RAM) is carried by the robot for executing the motion control and measuring the vital sign. A Windows tablet is then used for monitoring both sensor data and the Kinect output. The entire systems are realized using C++ and HTML.

First, performance of the human pose and orientation estimation is evaluated. Figure 8 shows the detection and estimation results of the person in various poses, which is qualitatively correct for both pose and orientation. It indicates the robustness of our human pose and orientation estimation. These results are quantitatively supported by table I.

TABLE I: Performance of the pose and orientation estimation

	Mean	Std. Dev.
Distance of projected pose to head (in *cm*)	66.5	15.6
Orientation error to head (in degrees)	5.0	8.26

Table I shows the projected head position in the ground plane to the original head pose and its orientation error results[1] of 20 different poses (six poses are displayed in Fig. 8). Compared to "the distance to head" we have set in section III-B (*i.e.* 70 *cm*), the result on table I is relatively favorable. The orientation error is also small, make it possible to be given to the robot as the position for measuring the person's breath.

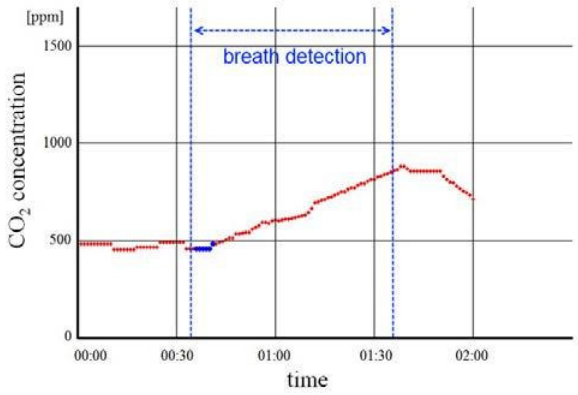

Fig. 9: Result of CO_2 measurement from the breath detection over the time. The blue range is the result when the robot measures the person's breath.

Subsequently, we examine the overall performance of our monitoring system. Figure 9 and 10 demonstrate a comprehensive real world experiment. In Fig. 10, the robot which is stationed outside the room, receives the fallen person data from the Kinect. The robot then goes to the front of the person's head and measures the breath. Figure 9 shows the CO_2 sensor performance during the vital measurement. When the robot stops to do the vital (breath) measurement as pointed by the blue dots, the sensor value significantly increases, indicating that the person still alive. The measurement results are transmitted to the server real-time so that it can be read by the other person using the hand-held device. After finishing the measurement, the robot goes back to its initial position.

We conduct five-fold experiments and all of them are successful, which means the robot correctly follows all of the state sequences mentioned above without any collision with the environment nor the fallen person. One notable thing is that the CO_2 sensor has a relatively slow response to the change of the CO_2 concentration in the air. As shown in our experiments (Fig. 9), it needs at least one minute to correctly measure the breath condition.

Lastly, the cooperation between the Kinect and the robot during the real experiments is also investigated. We measure the difference between the fallen person pose given by the Kinect and the executed pose by the robot, which is shown by table II.

Out of five experiments shown in table II, the maximum difference between the human pose given by the Kinect and the one executed by the robot is 16.5 *cm*. We consider these errors are due to uncertainty of the robot pose. In comparison with the given distance to the head (see table I), these errors are enough for the robot to not collide with the person's head.

[1]The orientation error is calculated from the difference of the ground truth marker and the orientation of the current observation.

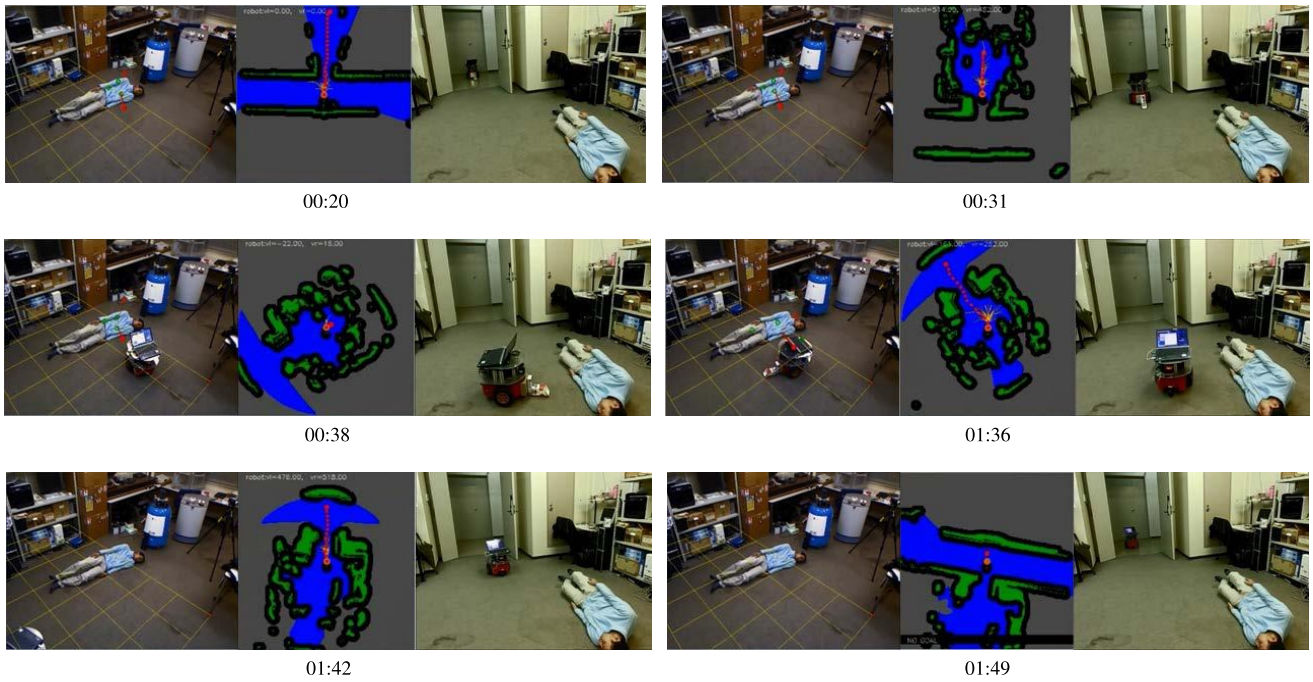

Fig. 10: Screenshot of experiments showing the sequences of the monitoring framework with the minute-second time frame (mm:ss). The left figure shows the Kinect's view, the center one shows the robot bird's view map and its motion planning, and the right figure shows the view from an observing camera.

TABLE II: Pose differences between the Kinect and the robot

	Experiments				
	#1	#2	#3	#4	#5
Pose differences (in *cm*)	10.5	9.0	16.5	12.5	10.0

VI. Conclusion

A framework of cooperation between a surveillance sensor and a mobile robot for an indoor monitoring system has been established. Here, a Kinect-based detector successfully gives the head position and orientation information of the fallen person to the robot. Once the robot receives the information, it goes to the person location, performs a vital sign analysis, and report the person condition via web server.

While the experiments show remarkable results, some considerations need to be further investigated. First, the use of CO_2 sensor for the vital sign detection (in this case, the breath), of course, is practically not enough. Some other vital signs, *e.g.* heartbeat, blood pressure, and bone fractures or injury detection may be incorporated to the framework. Secondly, the actual pose of the person in the real situation might be very difficult to be detected. There are some occasions that a fallen person ends up in unnatural poses. A more sophisticated person detection system should be contemplated to handle these problems, especially the one which considers the body parts of the person.

References

[1] G. Ward, N. Holliday, S. Fielden, and S. Williams. "Fall detectors: a review of the literature". Journal of Assistive Technologies, vol. 6 (3), pp. 202-215, 2012.

[2] T. Zhang, J. Wang, L. Xu, and P. Liu. "Fall Detection by Wearable Sensor and One-Class SVM Algorithm". In Int. Conf. on Intelligent Computing, pp. 858-863, 2006.

[3] E. Auvinet, F. Multon, A. Saint-Arnaud, J. Rousseau, and J. Meunier. "Fall Detection With Multiple Cameras: An Occlusion-Resistant Method Based on 3-D Silhouette Vertical Distribution". IEEE Trans. on Information Technology in Biomedicine, vol. 15 (2), pp. 290-300, 2011.

[4] M. Volkhardt, F. Schneemann, and H. Gross. "Fallen Person Detection for Mobile Robots using 3D Depth Data". In IEEE Int. Conf. on Systems, Man, and Cybernetics, pp. 3573-3578, 2013.

[5] G. Mastorakis and D. Makris. "Fall detection system using Kinect's infrared sensor". Journal of Real-Time Image Processing, 1861-8219, 2012.

[6] C. Rougier, E. Auvinet, J. Rousseau, M. Mignotte, and J. Meunier. "Fall Detection from Depth Map Video Sequences". In Int. Conf. on Smart Homes and Health Telematics, pp. 121-128, 2011.

[7] I. Fette and A. Melnikov. "The Websocket Protocol". Internet Engineering Task Force (IETF), pp. 1-71, 2011.

[8] Z. Zhang. "A Flexible New Technique for Camera Calibration". IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22 (11), pp. 1330-1334, 2000.

[9] V. Lepetit, F. Moreno-Noguer, and P. Fua. "EPnP: An accurate O(n) solution to the pnp problem". Int. Journal of Computer Vision, vol. 81, pp. 155-166, 2009.

[10] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. "Real-time human pose recognition in parts from single depth images". In IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1297-1304, 2011.

[11] T. Koshy. "Discrete Mathematics with Applications". Academic Press, 2004.

[12] I. Ardiyanto and J. Miura. "Real-time navigation using randomized kinodynamic planning with arrival time field". Robotics and Autonomous Systems, vol. 60 (12), pp. 1579-1591, 2012.