Title
Autonomous Patient Safety Assessment from Depth Camera Based Video Analysis

Author
Baek, Francis Seung-hyun

Publication Date
2016

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Autonomous Patient Safety Assessment from Depth Camera Based Video Analysis

A Thesis submitted in partial satisfaction of the requirements
for the degree Master of Science

in

Electrical Engineering (Intelligent Systems, Robotics, and Control)

by

Francis Baek

Committee in Charge:

Professor Vikash Gilja, Chair
Professor Truong Nguyen
Professor Mohan Trivedi

2016

The Thesis of Francis Baek is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2016

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

We thank James Killeen, MD and Gilbert Ramirez, MD from the UCSD Emergency Department for insightful discussions that have helped to shape this work. We would also like to thank Paolo Gabriel for supporting Kinect recording software.

ABSTRACT OF THE THESIS

Autonomous Patient Safety Assessment from Depth Camera Based Video Analysis

by

Francis Baek

University of California, San Diego, 2016

Professor Vikash Gilja, Chair

We introduce a low-cost, minimally intrusive system for the detection of high-risk postures and movements for patients. Our current focus is on the detections of when a patient leaves the bed surface and when a patient moves their body, which could be potentially utilized to detect a fall from bed, pressure ulcer, and tonic-clonic seizure risks in real-time. Using simulated data from healthy individuals, we develop and assess initial detection algorithms.

INTRODUCTION

1. Motivation

Safety issues are always around us. This is especially true in hospitals, where accidents such as falls from bed, pressure ulcers, and tonic-clonic seizures can cause severe secondary injuries to patients. Unattended falls from bed result in an injury at a rate of 42% and serious injury at a rate of 8% [1]. Pressure ulcers, which are injuries to skin and underlying tissue caused by prolonged pressure on the skin, acquired at the hospital result in 60,000 estimated patient deaths each year [2]. Tonic-clonic seizures cause loss of consciousness, muscle stiffness, and jerking movements. This kind of seizure generally lasts 1-3 minutes [3], and the associated loss of control is likely to cause severe secondary injuries to patients. Because all three of these risky scenarios relate to observable movements of the patient, a possible solution to reduce such risks is continuous monitoring by hospital staff. Due to the limited number of staff, however, full-time monitoring of each patient is not possible. We propose a continuous and autonomous low-cost system for detecting high-risk postures and movements of patients when they are alone. This detection of risky situations will alert staff to the need for intervention or further monitoring. The system presented is designed for the UC San Diego Emergency department clinic rooms.

2. Approach

We use depth-based computer vision techniques to detect kinematic features that may be predictive of falls from bed, pressure ulcers, and tonic-clonic seizures. These safety risks have respective physical and observable movement features that are clear and distinct: physical location change, extremely low mobility, and extremely high mobility.

Depth imaging captures geometric information of the scene, allowing us to study the physical characteristics of clinical safety problems, and provides lighting invariant information about room structure. Furthermore, depth data can be used to estimate patient posture while obscuring patient identity without any pre- or post-processing. A depth camera resolution that is sufficient for this application, is coarse enough that distinguishing facial information can be obscured. Our current task is to detect two specific events: a patient leaving the surface of a bed, and a patient moving their body. We solve these tasks within constrained room conditions and scenarios simulating the UC San Diego Emergency Department clinic rooms. This allows us to regulate the experimental environment, constraining the problem's complexity. For example, during their stay in a clinic room, patients will typically be on the bed. Thus, we can limit our focus to behaviors on the bed, such as reading a book, using a smartphone, and rolling in the bed.

The hospital environment brings out some of its own specific problems. One challenge is when a patient is moved into the room, the entire bed is carted into the room with the patient, so the position of the bed is variable. Another major challenge is that the hospital bed is adjustable, so different patients may have different bed angle settings. Thus, the algorithm should be able to recognize and classify the bed and patient from unsupervised data. In our current approach to solve these problems, we are using a single depth camera mounted on a fixed location, which generates a defined view within the patient room, as shown in Figure 1. Given depth images from this configuration, we solve the problem by focusing on two main components - body detection and body analysis as illustrated in Figure 1. The objective of body detection is to correctly extract the patient's body from each depth frame. Based on this result, body analysis assesses the physical status of the patient - whether the patient has exited the bed, and whether the patient has moved. Currently, we assume that there are no obstacles such as tables or other medical instruments between the patient and camera. We also assume that the system would only be active when the patient is alone, because if there is another person in the room, they could monitor the patient.

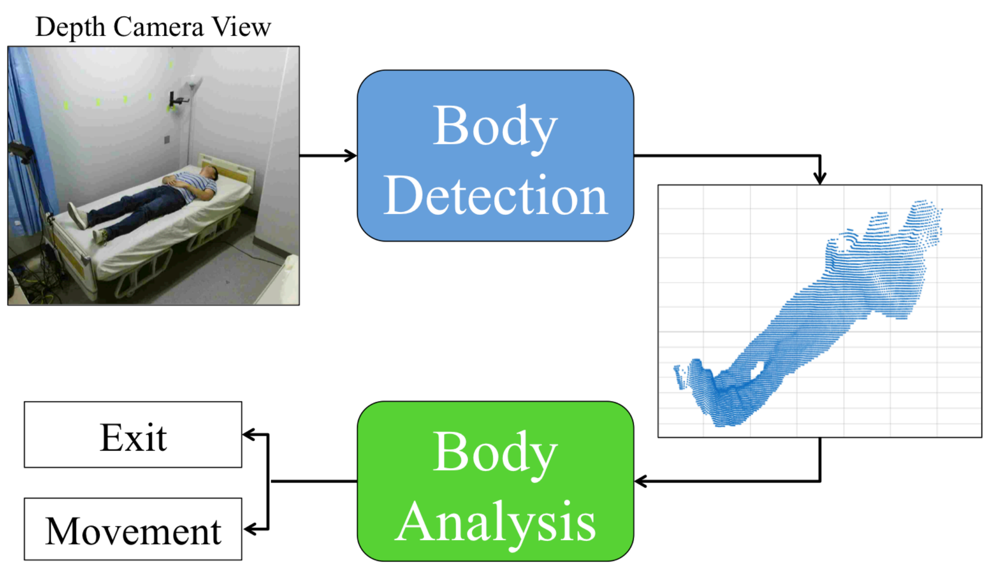

Figure 1: Flow chart of system structure.

METHOD

1. System Setup

1.1 Virtual Clinic Room

The future goal of the project is real application in a UC San Diego Emergency Department clinic room. One of main problems we need to solve in the hospital environment is that the system should classify patient's body from the adjustable hospital bed. However, we could not find any public depth dataset simulating on-bed scenarios like in hospital. Thus, we generate our own dataset by setting up a virtual clinic room, which has a similar scene to the real clinic room. The system is tested in our virtual clinic room. A single depth camera is mounted on the wall to fully monitor the bed area. The mounting location is chosen to match a mounting location available in UC San Diego Emergency Department clinic rooms. Figure 2 shows the virtual clinic room in depth camera's view. The point of view would always be stationary.

Figure 2: Color image showing depth camera view of the virtual clinic room.

1.2 Microsoft Kinect V2 Sensor

Microsoft Kinect V2 is used as a depth camera for this application. The sensor provides 30 depth frames per second and each frame has size of 424 by 512 pixels. It can view 70 degree in horizontal and 60 degree in vertical with maximum depth distance of 4.5 meter and minimum of 0.5 meter. The right image of Figure 3 shows the coordinate system of Kinect. This coordinate system is maintained throughout the process.

Figure 3: Microsoft Kinect V2 (left) and its coordinate system (right).

2. Algorithm

2.1 Structure

As illustrated in Figure 4, the system has two main components, which are body detection and body analysis. The objective of body detection is to correctly extract the patient's body from depth image. Based on the extracted body, the body analysis detects whether the body has moved or not and whether the body exits the bed surface or not. The body exit detection catches location changes of the body and the movement detection checks movement of body parts. These current detections are the foundation for future fall, pressure ulcer and tonic-clonic seizure detections.

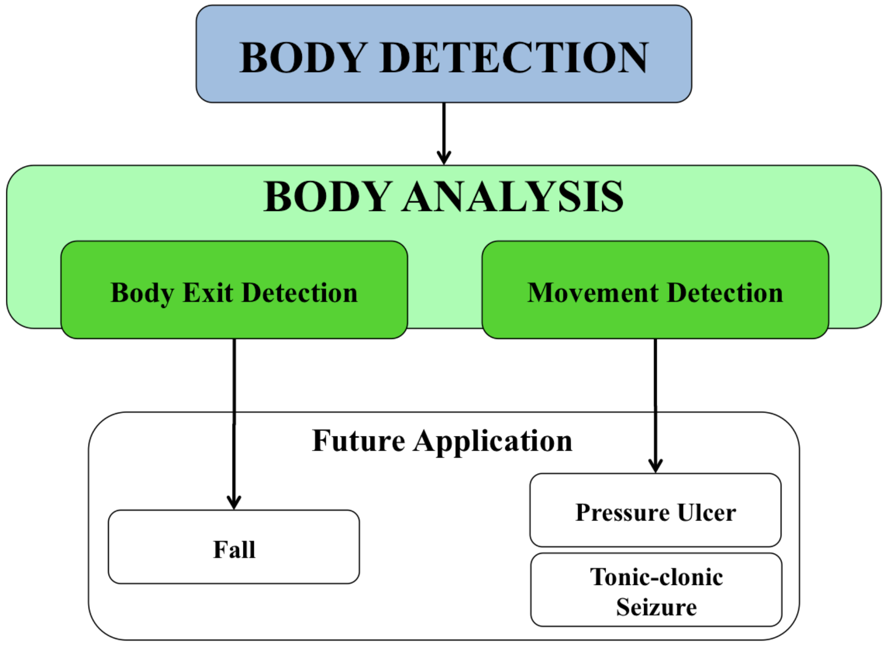

Figure 4: Algorithm structure and corresponding future application.

The body detection algorithm follows four steps as shown in Figure 5. The first step is to recognize geometric features of the bed, floor, and walls from the depth image such as normal vectors of floor and walls. The first step gives probable bed location in the depth image and the system searches the bed area based on this location to fully monitor the bed and the patient. The bed area is mapped into a 3-dimensional X, Y, and Z point-cloud to easily extract geometric features. The system then distinguishes the patient's body and the bed from the bed area point-cloud.

The body analysis algorithm has two different detections for movement and body exit. The exit detection algorithm detects when the body leaves the bed surface, and has three steps labeled as E. After disambiguating the body and bed, it finds bed boundary and distinguishes on-bed body parts and off-bed body parts. Based on the on-bed body parts, the algorithm decides whether the body exits the bed surface or not. The movement algorithm detects on-bed movements, and has two steps labeled as M. The bed area template and bed boundary information are required to detect movement. By using this knowledge of the bed area and bed boundary, the system focuses on movements occurring on the bed surface.

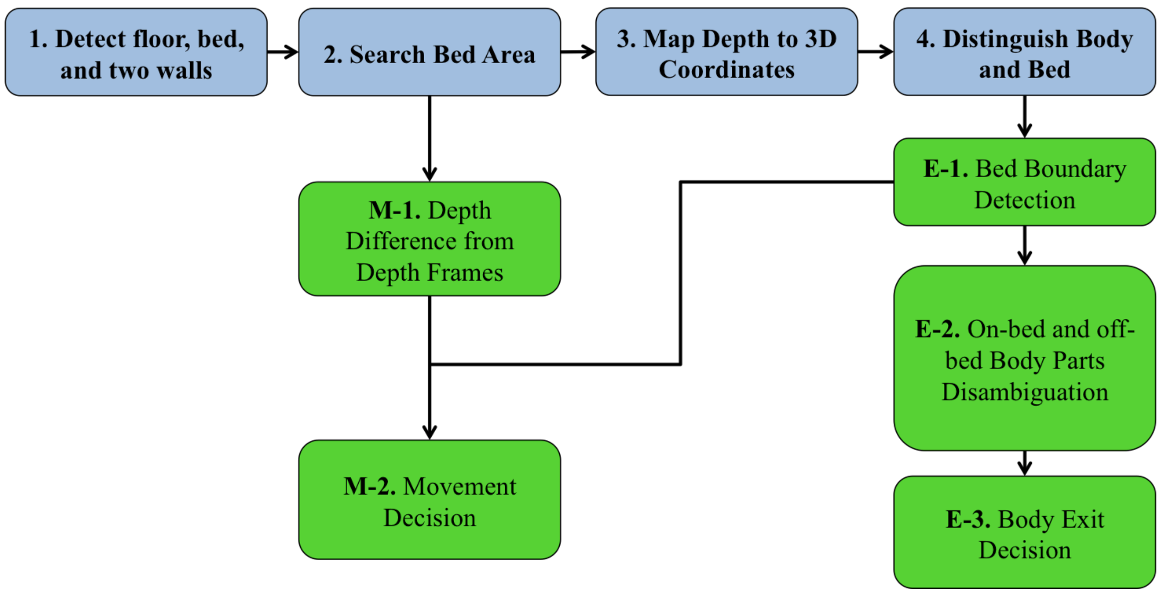

Figure 5: Detailed algorithm structure; blue boxes are sub-steps of body detection and green boxes are sub steps of body analysis. M stands for movement detection and E stands for exit detection.

2.2 Body Detection

The goal of body detection is to correctly classify body and bed point-clouds from the depth image. It follows four steps. First, the image regions corresponding to the floor, the bed, and the walls are detected from the depth image, and the normal vector for each of these regions is calculated. Next, the bed region is mapped into a 3-dimensional X, Y, and Z point-cloud. Then, by using the knowledge of room normal vectors and geometric features, the algorithm classifies body and bed regions within the point-cloud.

2.2.1 Floor, Bed and Wall Detection

Knowing the normal vectors of walls and floor makes point-cloud computation easier because we can use intuitive geometric features to deal with the point-cloud. The gradient filter is applied to firstly detect floor and bed regions. Typically, the gradient filter is applied to color images and filter output gives the direction of largest possible intensity increase. Likewise, the gradient filter gives the direction of largest possible depth increase in the depth image. When we scan depth values in each row from bottom to top, the depth values of floor and bed would increase and the depth values of walls would be decreased. As we can see in Figure 7 c), the regions of positive gradients are extracted from the gradient filtered depth image, and the regions contain bed and floor as we expect. Morphological image processing techniques such as erosion and closing are used to reduce noise. The result image is shown in Figure 7 d). We know that the image contains both floor and bed. To distinguish between bed and floor regions, the two largest connected areas are extracted as shown in Figure 8. From these two areas, the one with a center of mass closer to the center of the image is labeled as bed, and the other is labeled as floor. To detect wall, the regions of negative gradients between -120 and -20 are extracted. Figure 9 shows that this approach correctly catches bed, floor and wall from depth image even when there is a person on the bed and even when the bed is not flat. The normal vectors of floor and wall, which are labeled as N1 and N2 in Figure 10, can then be calculated. Based on these two vectors, the normal vector of the other wall, N3, can be calculated by using outer product. Thus, the system can automatically find out room components from depth images and is not sensitive to angle changes, if the scene is still similar to Figure 2. In depth video analysis, the normal vectors of first frame are applied to the remaining depth frames. Because the depth camera has fixed view, we do not have to calculate normal vectors in every frame.

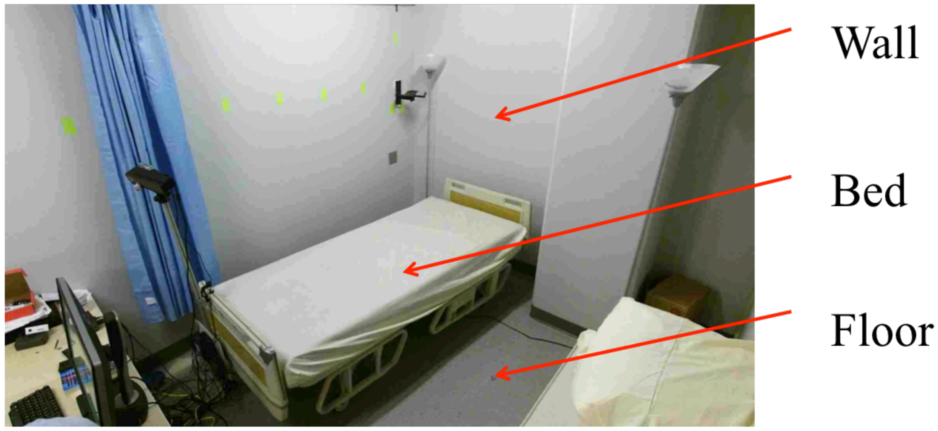

Figure 6: Room components to be searched.

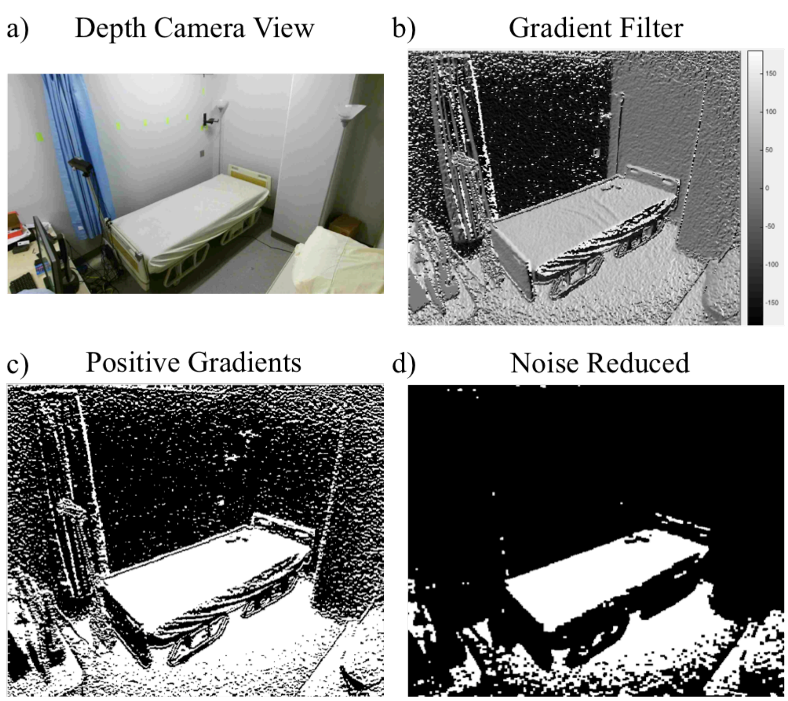

Figure 7: Positive gradient regions; a) Room in the depth camera view, b) Gradient filtered depth image, c) Pixels of positive gradients are extracted, d) Noise removed image of c).

a) Positive Gradients b) Largest Connected Area c) 2nd Largest Connected Area

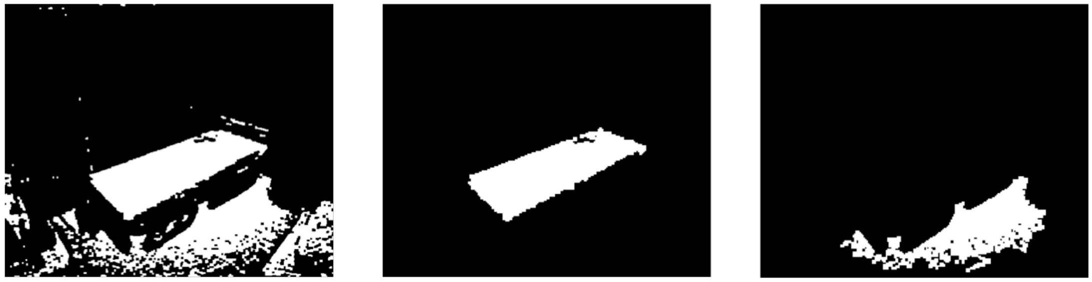

Figure 8: Largest Connected Regions; a) Noise removed positive gradient area, b) Largest connected area of a), c) 2nd largest connect area of a).

Depth Camera View Floor Bed Wall

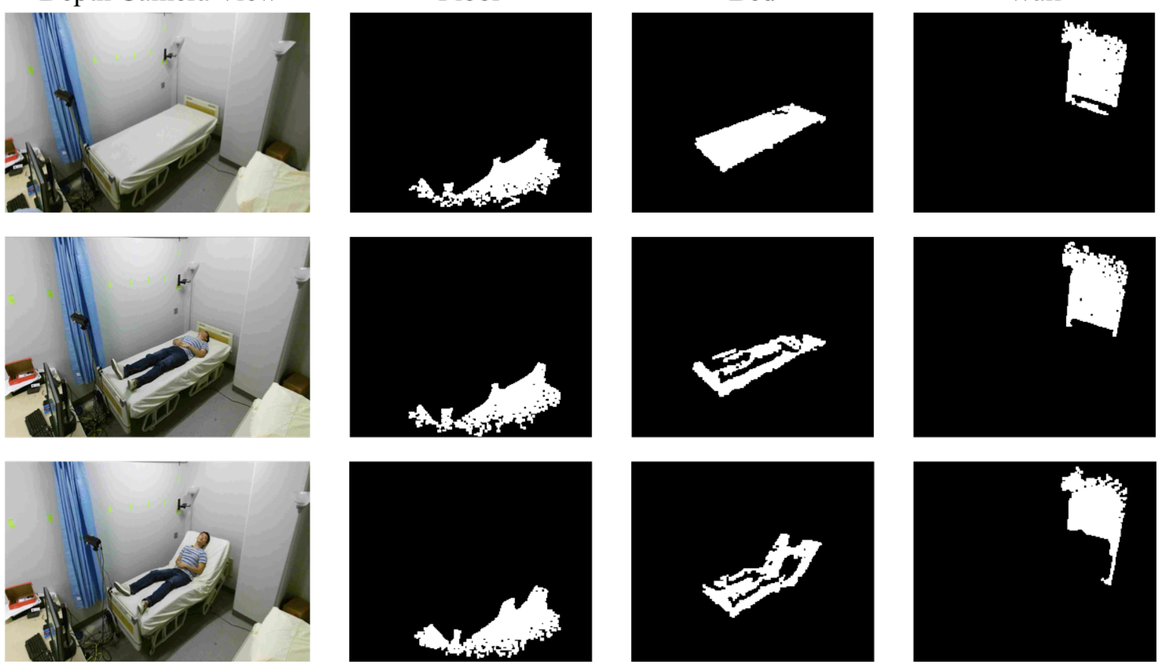

Figure 9: Sample results of floor, bed, and wall detection.

Figure 10: Room normal vectors; N1 and N3 are normal vectors of two walls, and N2 is normal vector of floor.

2.2.2 Bed Area Search

From the previous step, the bed surface location is found as we can see in the 'Bed' column of Figure 9. The bed location is found but the image contains many holes, which would result in loss of information. To ensure that the entire bed surface is captured, the estimated bed surface is blurred by image dilation. The result is shown in Figure b). This image is used as bed area template to focus on bed area and to reduce point-cloud computation time by discarding unnecessary information outside of the bed region. In depth video analysis, we firstly collect 5 bed area templates and we create a template by collecting pixels from the 5 templates, which are bed area in more than 2 templates. This new template is used in the remaining depth frames.

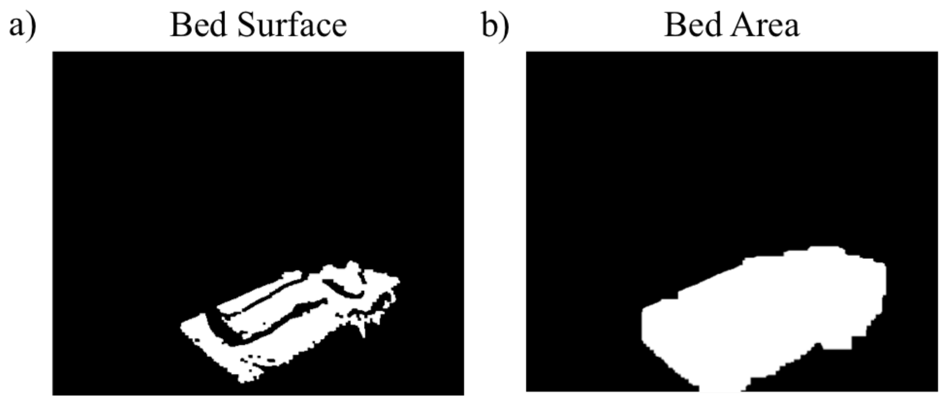

Figure 11: Bed area template; a) Bed surface found by floor, bed and wall detection algorithm (2.2.1). b) Bed area by blurring a).

2.2.3 Depth to 3D Point-cloud Mapping

Now we know where the bed area is within the depth image, and we know the normal vectors of room components. To apply these geometric features to the bed area, the bed area is mapped into a 3-dimensional X, Y, and Z point-cloud space. In a depth image, the depth value of an object is the 'Depth' distance as described in Figure 12.

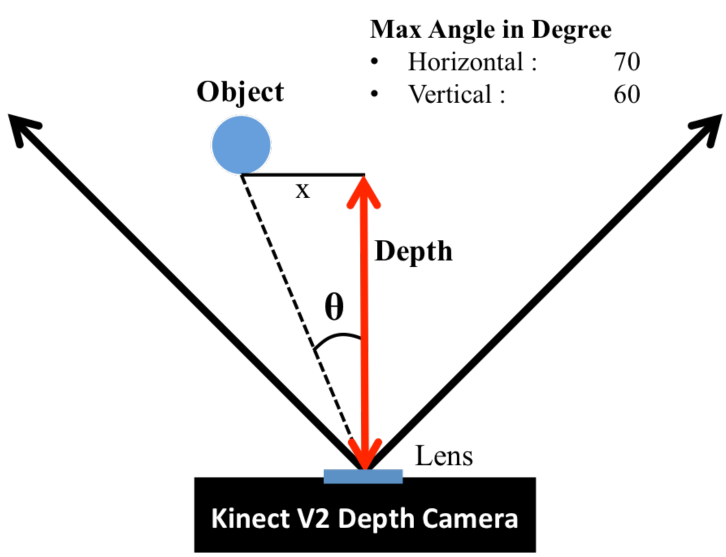

Figure 12: The meaning of depth value in depth image.

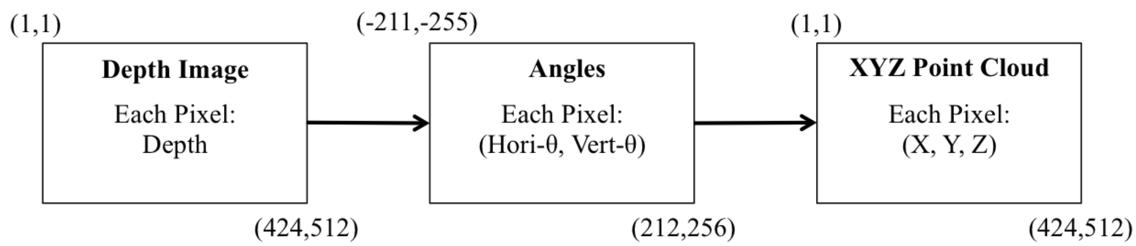

Figure 13: Flow chart of mapping steps

The steps of mapping are shown in Figure 13. As shown in Figure 3, in the Kinect's coordinate system, the 'Depth' distance is parallel to the Z-axis in 3 dimensions. Each depth image is 424 by 512 pixels, and can view 70 degrees in horizontal and 60 degrees in vertical. Based on this resolution, the degree change of each pixel can be calculated. For example, each horizontal pixel change means 70/512 degree changes and each vertical pixel change means 60/424 degree changes. We should note that this calculation is an approximation of the real degree changes. In reality, the lens of camera does not linearly map pixels to those angles, but we assume that the pixels are linearly mapped into those angles. Thus, each depth pixel coordinate and its depth value can be used to estimate a corresponding X, Y, and Z coordinate. In the depth image, the indices change from (1,1) to (424,512), which correspond to the top left corner and the bottom right corner, respectively. However, we set (212, 256) pixel as center in the 3D coordinates, so the indices change from (-211, -255) to (212,256). Based on the indices and corresponding depth values, the X and Y coordinates are calculated by using the equations below, where Z coordinates are the depth values themselves.

$$\begin{cases} X = Depth * \tan\theta_{horizontal} \\ Y = Depth * \tan\theta_{vertical} \\ Z = Depth \end{cases}$$

$$where \begin{cases} \theta_{horizontal} = (X\ index\ between - 255\ and\ 256) * \dfrac{70}{512} * \dfrac{\pi}{180} \\ \theta_{vertical} = (Y\ index\ between - 211\ and\ 212) * \dfrac{60}{424} * \dfrac{\pi}{180} \end{cases}$$

For example, if (200, 300) pixel in depth image has 100 depth value, then the new index relative to the center pixel would be (200-212, 300-256) = (-12, 44). From this, the X, Y, and Z coordinates are calculated

$$\begin{cases} X = Depth * \tan\left(44 * \dfrac{70}{512} * \dfrac{\pi}{180}\right) \\ Y = Depth * \tan\left(-12 * \dfrac{60}{424} * \dfrac{\pi}{180}\right) \\ Z = Depth \end{cases}$$

The result of the mapping is shown in Figure 14 b). However, because the mapping of the bed area template includes some parts of background, they are removed by thresholding 30cm from each wall. The resulting point-cloud is shown in Figure 13 c).

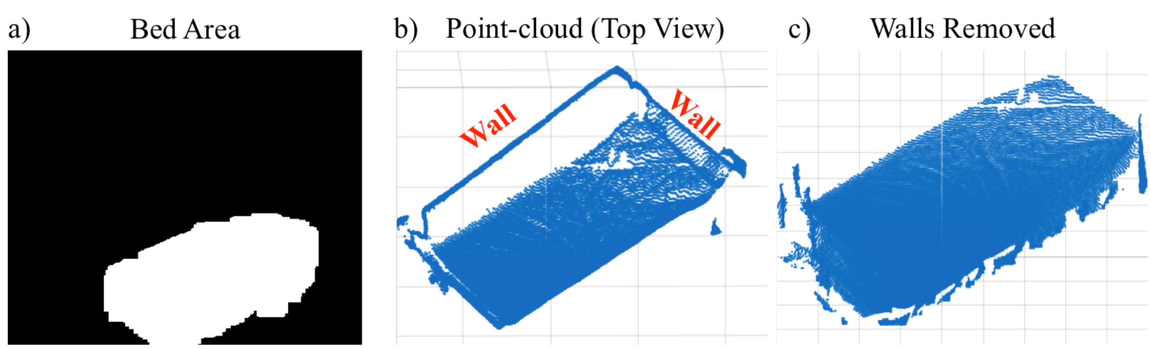

Figure 14: Point-cloud mapping; a) Bed area template, b) Mapping of a) into point-cloud, c) Walls are removed from b).

2.2.4 Body and Bed Classification

From the bed area point-cloud, the bed and body should be distinguished to correctly analyze the body condition. The points above the bed surface are considered as body points. To correctly extract body points, the bed surface height is measured. Because the hospital bed is adjustable, the algorithm should be robust to the bed surface degree changes. For this to be possible, we divide the bed point-cloud into 5cm slices as shown in Figure 15 a) and b). We scan the slices twice. In the first scan, the heights of edges are measured, which are marked red in Figure 15 b). We assume that the lower height between the two edges might be the bed surface height because when the body part lies on an edge, it would have higher height. After the first scan, we can get the measured bed surface height, as shown in Figure 15 d). It is noisy because when body parts lie on the edges or when body parts occlude the depth camera's sight, the correct heights cannot be measured. Thus, the corrected bed height is generated based on the noisy height. Because we know that the bed surface cannot be spiky like the measured height, we ignore some significant outliers to make the height smoother. The corrected height is shown in Figure 15 d). In the second scan, the points above the corrected height are extracted from each slices as shown in Figure 15 b) and c). By combining all the extracted points, the body point-cloud is created and the remaining points are considered as bed point-cloud. The result of body and bed point-clouds are shown in Figure 15 e) and f). As shown in Figure 16, this algorithm is robust to bed surface degree changes and it is also robust to the use of blanket. In depth video analysis, the corrected height is applied to the remaining depth frames because we assume that the bed surface would not be changed when the patient is alone.

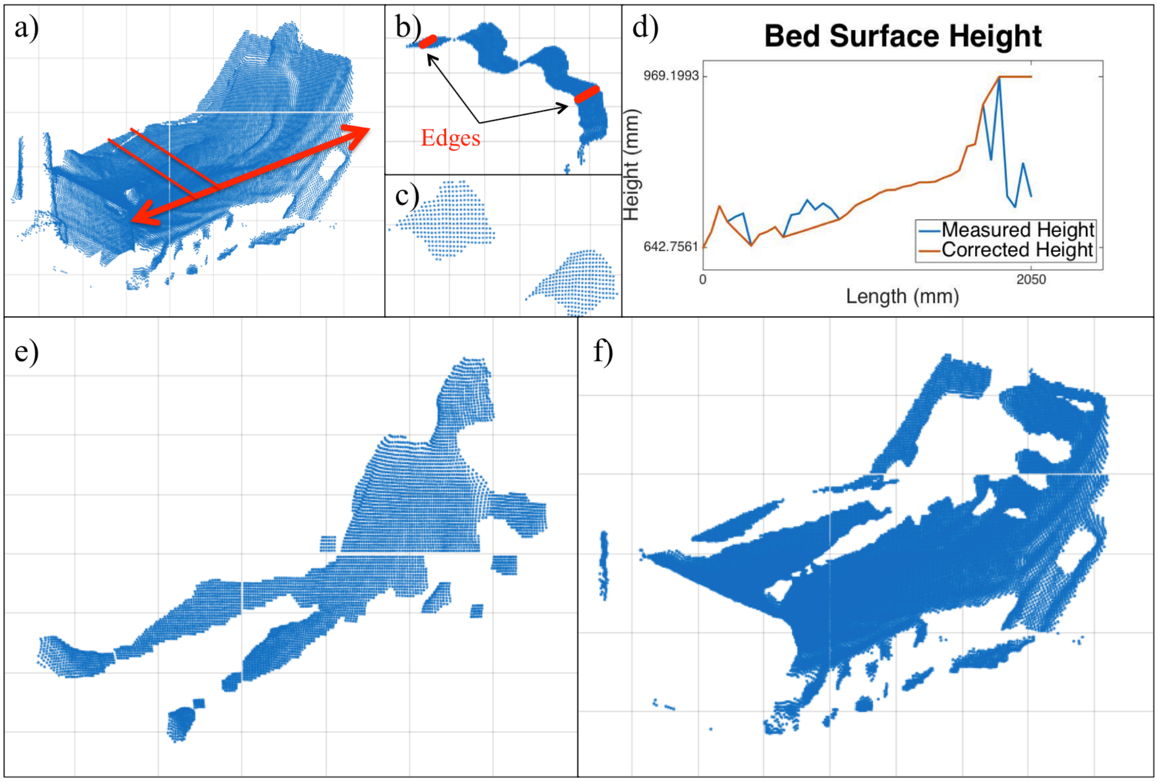

Figure 15: Bed and body classification; a) Bed area point cloud with slicing direction marked, b) A slice from a), c) Body points extracted from b), d) Measured bed surface height and corrected bed surface height, e) Body point-cloud, f) Bed point-cloud.

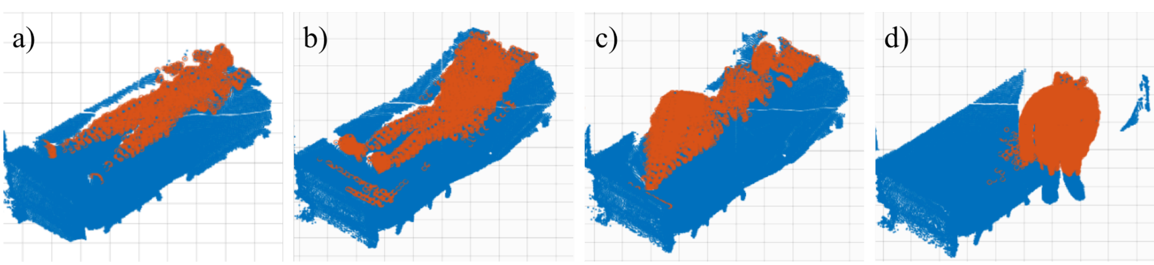

Figure 16: Sample results of body detection; a) Flat bed without blanket, b) Adjusted bed without blanket, c) Adjusted bed with blanket, d) Body out of the bed surface.

2.3 Body Analysis

Body analysis algorithm uses information from body detection algorithm and decides whether the patient has moved or not, and whether the patient exits the bed or not. These two cases are captured by two different detection algorithms body exit detection and movement detection.

2.3.1 Body Exit Detection

The purpose of the body exit detection is to correctly catch when the patient exits the bed surface. The bed boundary is found from the bed point-cloud and the algorithm decides whether the body is exiting or not by monitoring the on-bed body volume changes.

2.3.1.1 Bed Boundary Detection

The bed boundary is found using the bed point-cloud from the body detection algorithm. The surface area is found from the bed area points. The surface can be found by thresholding dot product results of floor normal vector and normal vectors of bed points. The bed surface area is red marked in Figure 17 b). From this area, the left and right boundaries are found as shown in Figure 17 c).

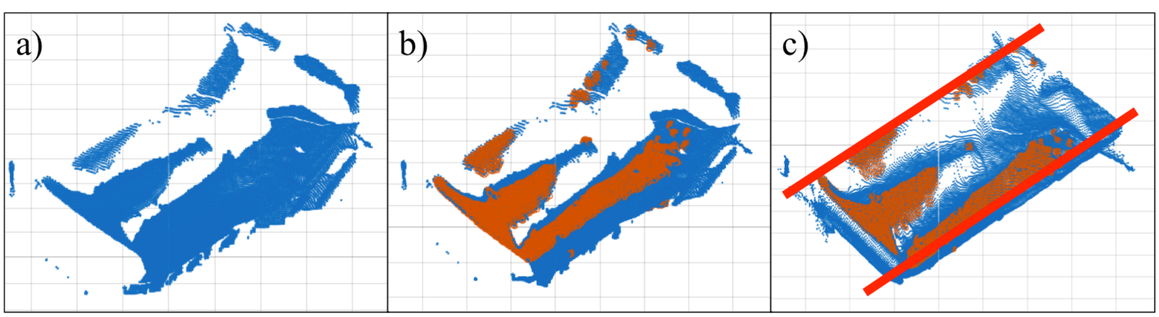

Figure 17: Bed boundary detection; a) Bed point-cloud, b) Bed point-cloud with bed surface marked, c) Left and right bed boundaries are marked.

2.3.1.2 On-bed and Off-bed Body Part Disambiguation

The boundaries are used to distinguish the body points on the bed and out of the bed as shown in Figure 18.

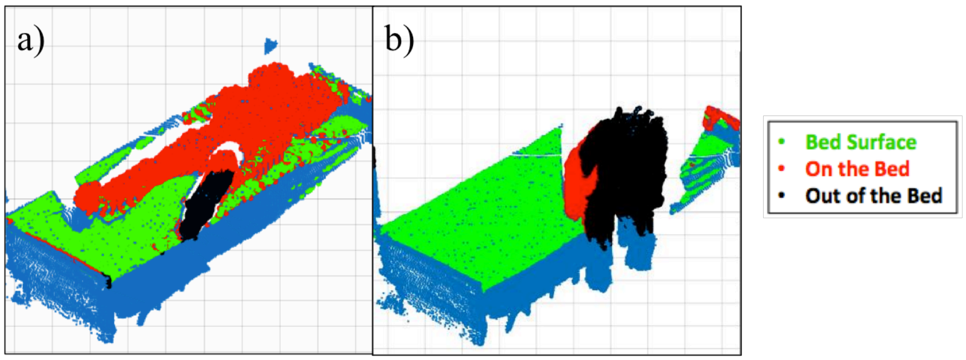

Figure 18: Sample results of on-bed and off-bed body part disambiguation.

2.3.1.3 Body Exit Decision

The algorithm only focuses on the body points on the bed. By assuming that each point has unit area, it estimates body volume by calculating heights of body points relative to the bed surface height. It monitors body volume changes and the initial body volume normalizes each body volume of further frames. When the normalized body volume decreases lower than a threshold, it decides that the body exits the bed. The sample body exit detection monitoring is shown in Figure 19.

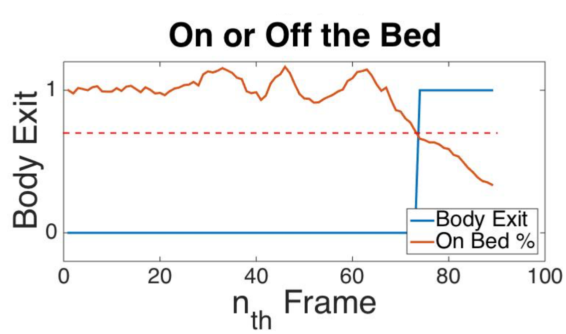

Figure 19: Sample result of body exit detection.

2.3.2 Body Movement Detection

The objective of movement detection is to correctly detect whether a patient moves or not. Currently, the algorithm can only make binary decisions and does not know which part of the body moves. For this simple movement detection, a sequence of depth frames and the knowledge of bed boundaries are used.

2.3.2.1 Depth Difference from Depth Image Frames

A sequence of depth images is used to check movement. Every depth frame is first masked with the ‘bed area template’ (Figure 11 b) to only focus on the bed area. The change in depth for each pixel between the two depth images is calculated by subtracting the current depth frame from an initial frame. The pixels that have depth difference larger than 10 mm are kept, and the other pixels are ignored to remove noise. A binary image template is created based on the depth difference as shown in Figure 20 a). Note that the region consists of two sub-regions, ‘move from’ and ‘move to’. Because we do not consider body posture estimation currently, we just keep the both regions. The template region is mapped into the 3-dimensional point-cloud and the points out of the bed boundaries are ignored to only focus on the movement on the bed surface. The sample result is shown in Figure 20 b), in which the template region is red-marked.

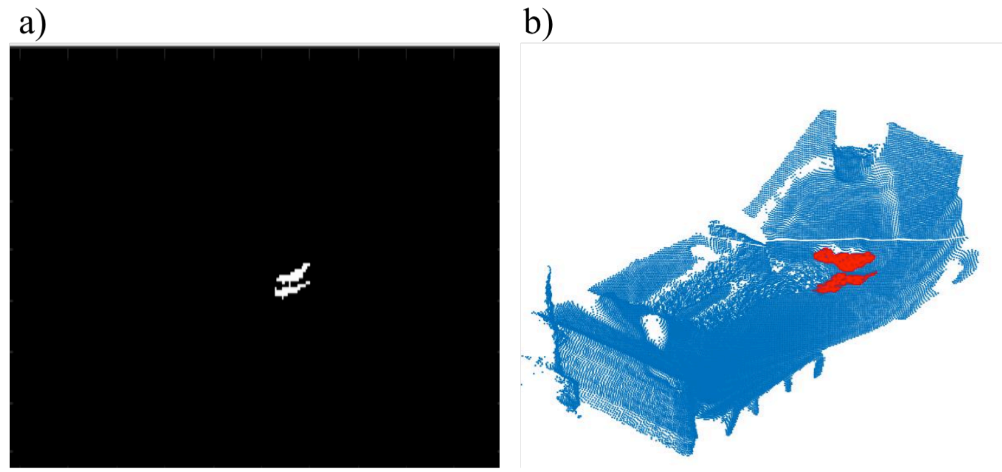

Figure 20: Moved part mapping; a) Difference between initial and current depth frames, b) a) mapped into point-cloud (red-marked).

2.3.2.2 Movement Decision

The volume of the moved parts is estimated with the same previous assumption that each body point has unit area. The algorithm monitors the volume of the moved parts relative to the body volume. As illustrated in Figure 21, if the ratio becomes greater than a threshold, then the algorithm decides that the body has moved.

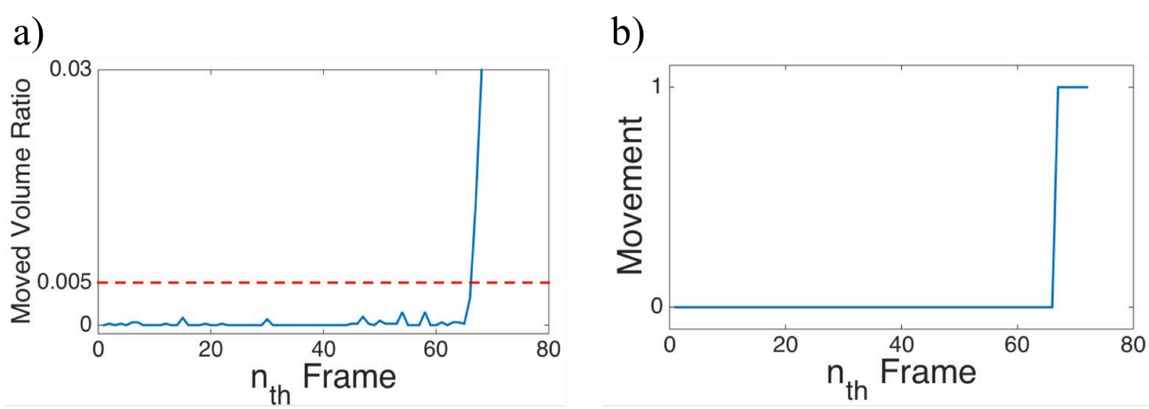

Figure 21: Movement detection; a) Monitoring volume of moved part relative to the body volume, b) Sample result of movement detection.

In addition to comparing the initial frame with the current frame for movement detection, the algorithm should be able to detect if the movement is over or not. Even though the moved volume ratio is greater than the threshold, the movement is possibly already over if the patient's pose is changed after the movement. So, the algorithm checks moved part volumes across 10 frames, which include the current frame and previous 9 frames, as shown in Figure 22. If no significant movement is detected from the 10 frames, the system decides the movement is over and updates the initial frame to the current frame to detect further movement.

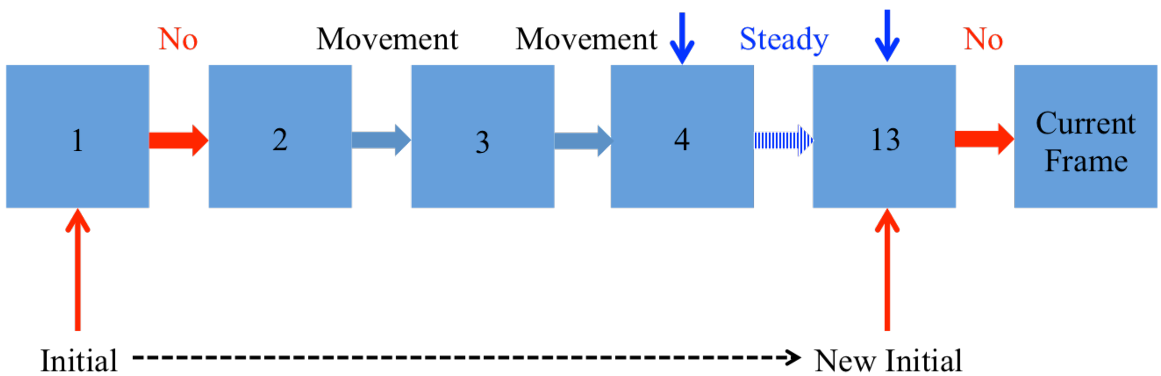

Figure 22: Movement detection from sequence of depth frames

While there is no movement, the initial frame is kept and the number of frames between the initial frame and the current frame is counted to estimate the duration of no-movement, as illustrated in Figure 23. Because we know that 30 frames are taken each second, we can calculate the duration by using the number of frames. The duration of movement is not used in our current assessment. When the movement is over, the initial frame is updated to the first 'no-movement' frame.

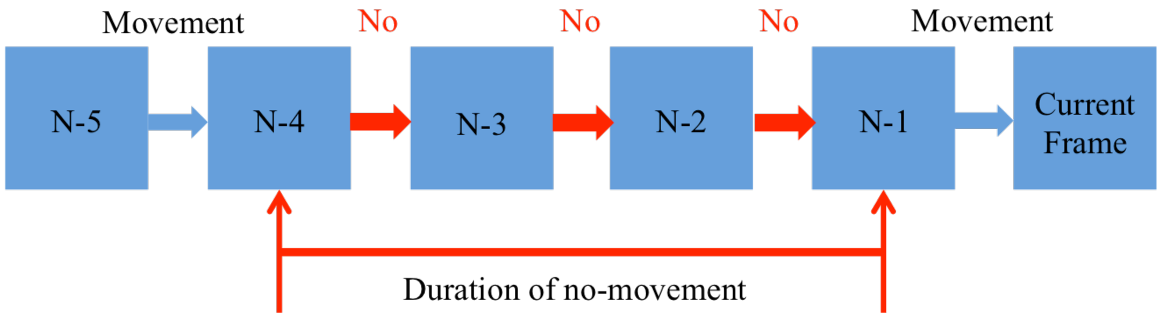

Figure 23: Duration of no-movement from sequence of depth frames.

RESULTS

1. Body Exit Detection

1.1 Experimental Setup

For the body exit detection, data are collected in the virtual clinic room from healthy individuals. Data are composed of 194 trials, which include 117 'body exit' trials and 77 'no exit' trials. To collect the 'body exit' trials, we ask the subjects to exit the bed within 10 seconds without any restriction in behavior. The time when the subject's feet touch the floor is measured with a stopwatch. The 'no exit' trials include a variety of behaviors while the subject remains on the bed, such as reading a book, rolling in the bed, raising and lowering the blanket, throwing the blanket out of the bed, and using a smartphone. All trials are between 3-17 seconds.

1.2 Body Exit Detection Result

The results of 194 trials using different thresholds are shown in Figure 24. In addition to the body volume ratio, we test three other approaches to monitor how the on-bed body changes, these are the ratio of the number of body points relative to initial number of body points, the estimated volume, and the estimated difference in volume relative to the estimated volume from the initial frame. The receiver operating characteristic (ROC) curve can be used to compare the effectiveness of these four approaches. The exit detection based on the 'volume ratio' has the largest area under curve (AUC), suggesting that it will be the most effective approach. Thus, the remaining tests described use the 'volume ratio'. The 'volume ratio' thresholds are set between -0.1 and 1. The threshold means the on-bed body volume changes relative to the initial on-bed body volume. Note that in some cases the body volume is non-decreasing relative to the initial body volume, so, the negative threshold is tested for the baseline. The histograms of Figure 25 show the exit detection times. The results are intuitive because the detection time is delayed as the threshold increases. Larger threshold means that the algorithm decides exit when a larger portion of on-bed body volume is gone. Table 1 shows the result of 194 trials using 'volume ratio' monitoring. We should note that when the true positive rate is close to 1, the false positive rate is not zero. The 'no exit' trials span 7 minutes, so, for example, for a 0.2 threshold that detects all exits, we might expect a false positive every minute.

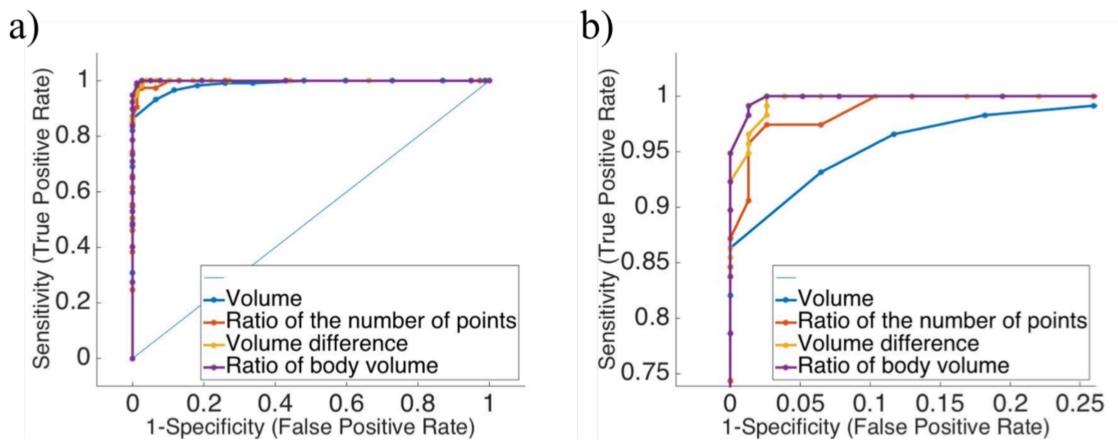

Figure 24: ROC curves of body exit detection; a) ROC of exit detection using 4 different measurements, b) Zoom-in view of a).

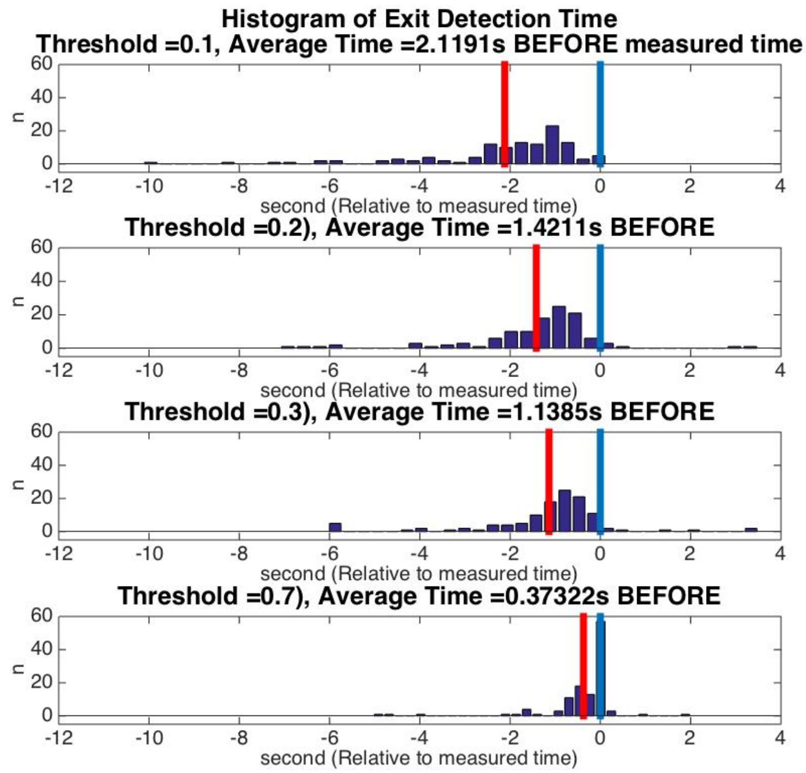

Figure 25: Histogram of body exit detection time

Table 1: Result of body exit detection using 'Volume Ratio' measurements.

194 Trials	Exit Trials		No Exit Trials	
	Total	117	Total	77
Threshold	Exit Detected (%)	Exit Not Detected (%)	Exit Detected (%)	Exit Not Detected (%)
-0.1	100	0	100	0
0	100	0	94.8	5.19
0.1	100	0	19.5	80.5
0.2	100	0	5.19	94.8
0.3	99.1	0.86	1.3	98.7
0.4	94.9	5.13	0	100
0.5	89.7	10.3	0	100
0.6	78.6	21.4	0	100
0.7	65	35	0	100
1	0	100	0	100

2. Movement Detection

2.1 Experimental Setup

For the movement detection test, data are collected in the virtual clinic room from healthy individuals. 50 'movement' trails are collected and a 3 minutes duration of 'no movement' trial is collected. To collect the 'movement' trials, subjects are asked to move their bodies, by following our instructions. Each instruction is verbally given 2 seconds after the start of monitoring, and it asks a movement, which is one of 'head nod', 'head rotation', 'right arm', 'left arm', 'right leg', 'left leg', 'upper body torso – lean backward', and 'upper body torso – lean forward'. For 'no movement' data, we asked subjects not to move for a 3 minutes. This failed because in some trials, subjects slightly tossed and turned their bodies. Those slight movements were detected by algorithm and drastically increased false positive rates. Thus, for the correct 'no movement' test, we monitor an empty bed for 3 minutes.

2.2 Movement Detection Result

The thresholds between 0 and 1 are tested. The threshold means the ratio of moved parts volume to the body volume. Because all the instructions are given 2 seconds after the start of monitoring, movement detection times earlier than 2 second are considered to be false alarms. Table 2 shows that the threshold 0.03 could catch all the movements without any false alarm. The histogram in Figure 26 shows the movement detection time using threshold of 0.03. It shows that all the detection times are after 2 second.

Table 2: Result of movement detection.

	Movement Trials		No Movement Trial
	Total	50 Trials	For 3 minutes
Threshold	Detected (%)	Not Detected (%)	FALSE Alarms (#)
0	0	100	13
0.002	94	6	4
0.004	96	4	2
0.02	98	2	0
0.03	100	0	0
0.04	98	2	0
0.05	92	8	0
0.1	56	44	0
0.5	4	96	0
1	0	100	0

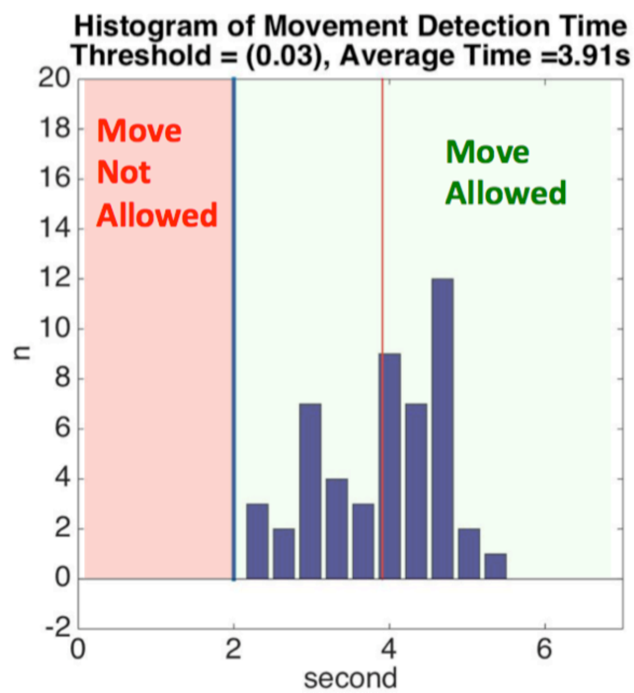

Figure 26: Histogram of movement detection time.

DISCUSSION

Both the body exit detection and movement detection give almost ideal performances, when they monitor estimated volume changes. However, there are subtleties that make real world application challenging.

For threshold of 0.3 in body exit detection, there is only one false negative and only one false positive. This seems like pretty good performance. However, we should note that the false positive rate is non-zero. The 'body exit' trials span 7 minutes, so this means that we might expect a false alarm every minute. Which would not be practical in real applications. Thus, our future work would first focus on reducing the false positive rate, and then would focus on improving detection.

In movement detection test, the 0.03 threshold gives perfect results with 1 true positive rate and 0 false positive rate. The next step would be estimating body posture based on the body point-cloud, because the current algorithm does not consider body posture. The body posture estimation would enable the system to recognize the type and magnitude of movements that are useful clues to detect specific symptoms of risks such as convulsive movements of tonic-clonic seizures.

The system runs in Matlab on 2.5 GHz Intel Core i5 processor, and the computation time for each frame takes about 2.5 seconds for initial calculations and 1.2 seconds for the rest. The initial calculations include bed area template creation and the corrected bed surface height measurement. The body and bed classification algorithm takes most of the time. It is because the algorithm divides the bed area points into about 50 slices to clearly distinguish the body and bed points. The large number of slices makes the algorithm slower. Reducing the number of slices while maintaining the performances would make the computation time faster. Because we are targeting real-time system, fast computation time is mandatory. Implementing a generative model of the bed may be another possible solution to make the computation time faster.

In terms of dataset, current data is collected from healthy individuals in our virtual clinic room. We plan to collect data from real hospital environment in UCSD emergency department clinic room from actual patients.

Overall, our approach uses geometric features of the clinic room. Currently, we are not using learning based approach, which is common approach in body analysis in these days. The reason is that the geometric features give enough information to solve our current tasks, which are detecting the exit events and the movements. For the future application in detecting the safety problems, our next tasks would be analyzing human behaviors such as body posture identification and gesture classification. The learning based approach would be necessary to solve our future tasks.

REFERENCES

1. Hitcho EB, Krauss MJ, Birge S, et al. Characteristics and circumstances of falls in a hospital setting. J Gen Intern Med 2004; 19:732-739.

2. Duncan KD. Preventing pressure ulcers: the goal is zero. Jt Comm J Qual Patient Saf. 2007 Oct;33(10):605–10.

3. Devinsky, Orrin, MD, and Joseph I. Sirven, MD. "Tonic-clonic Seizures." *Epilepsy Foundation*. July 2013. Web. 06 May 2016. <http://www.epilepsy.com/learn/types-seizures/tonic-clonic-seizures>.