

LoomoRescue: An Affordable Rescue Robot for Evacuation Situations

Denys J.C. Matthies¹, Sven Ole Schmidt¹, Yuqi He^{1,2}, Zhouyao Yu^{1,2}, and Horst Hellbrück¹

¹ Technical University of Applied Sciences Lübeck, Germany

² East China University of Science and Technology, Shanghai, China

Abstract. Rescue robots can play an important role in disaster situations, such as locating people for evacuations. This paper demonstrates how to transform an affordable consumer robot, such as the Loomo Segway, into an intelligent rescue robot. Our proof-of-concept shows how LoomoRescue can autonomously browse offices to locate people and how to detect their vital signs through posture and heart rate detection in real-time. Our indoor localization is a SLAM approach based on an external UWB position system plus a movement correction with an ultrasound sensor in combination with an IMU. The accuracy of linear movement showed minor deviation with an average error of 1.65%, while the angular movement showed an error of 2.43%. We classify three types of critical postures with an average detection rate of 78.33% within a distance of 1 – 20 meters. Our optical heart rate detection is 87.3% accurate to the ground truth. We envision that such an affordable robot can be used for evacuation purposes as it may be part of the standard inventory in the future.

Keywords: Rescue Robot · Computer Vision · Optical Vital Sign Detection · Posture Recognition · SLAM · Autonomous Navigation.

1 Introduction

Throughout the past decade, the use of rescue robots in disaster scenarios has been increasing [1]. These robots especially provide a benefit in scenarios that pose a threat to humans. Rescue robots play an important role in searching and locating trapped survivors. Different form factors have been developed for these purposes, such as robot dogs, drones, and other wheeled robots [2]. These devices can be crucial in searching for victims or merely supporting us with understanding conditions in inaccessible areas.

The current state-of-the-art in rescue and assistive robots demonstrate that significant research exists concerning assistance robots. However, particularly rescue robots are usually expensive and demand high robustness, speed, versatility, and ease of use for the piloting human in accordance to an extensive review by Delmerico et al. [1]. Ways to enable an affordable and consumer-available robot that already matches many of these requirements becoming a rescue robot

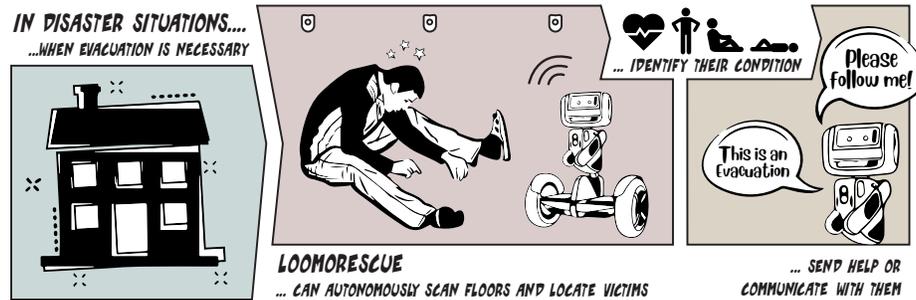


Fig. 1. LoomoRescue can be useful for disaster situations, such as when evacuation is necessary. The robot is fully autonomous and can systematically navigate through a floor, looking for possible victims. Further, LoomoRescue is able to understand whether the person may be in a critical condition by detecting vital signs through posture and heart rate. Finally, LoomoRescue can communicate with the victim or send a notification to a rescue team.

that acts autonomously to identify victims' health status in real time remains an unanswered question. We envision low-cost rescue robots to be a part of public facilities to enable support in disaster situations, such as building evacuations.

To demonstrate this, we developed a rescue robot based on an affordable consumer robot, the Loomo Segway³. We posed two research questions. Firstly, how can we enable an autonomous navigation? Secondly, we sought to investigate whether it was possible for Loomo to understand a human's critical activity?

We developed a proof-of-concept that transforms an affordable consumer robot into a rescue robot by enabling:

- Simultaneous Localization and Mapping (SLAM) [3] and path planning using a multi-sensor fusion approach incorporating an IMU, Ultrasonic sensors, and an additionally equipped Ultra Wide Band (UWB) based multi anchor localization system [4],
- Optical vital sign detection using a posture recognition approach by using Google's ML kit [5], as well as a heart rate detection by using an Eulerian Video Magnification (EVM) approach [6] to magnify subtle changes in skin colour.

Our main contribution is to integrate these different techniques into an affordable consumer robot that demonstrates that it can be converted into a rescue robot. According to Wobbrock this concerns an artifact contribution [7].

2 Related Work

There is a great body of related work, as this project intersects multiple areas. We selected some exemplary projects from the field of search-and-rescue robotics and human activity recognition that somewhat represent the state-of-the-art.

³ <https://www.segway.com/loomo/>

2.1 Search-and-Rescue Robots

Over the past few decades, several kinds of rescue robots have been developed. Though, research on rescue robots mainly focuses on localization and control systems. In 2019, Delmerico et al. [1] conducted an extensive review on the current state and future outlook of rescue robotics. Delmerico et al. categorized rescue robots into Ground Robots (e.g., Legged robots [8], Tracked and wheeled robots [9]), Aerial Robots (e.g., Drones [10, 11], Swarms [12]), and Marine and Amphibious Robots (e.g., Snake-like Design [13]). In the following, we will discuss some related work in more detail.

Kiyani et al. [14] developed a prototype of a search-and-rescue robot that has the ability to locate itself in a known environment and to locate victims and transport them to safe zones. This is at least a proof-of-concept of an Arduino-driven mini-robot in a lab environment, but which is far away from reality.

A system that has been tested in an actual fire container house was proposed by Young-Duk in 2009 [15]. Here, a fire rescue robot is remotely controlled using Bluetooth and RF communication. Firefighters may deploy the robot in a hazardous fire scene through a remote control guiding the robot for a possible evacuation. The controlling firefighter receives information from the robot in the form of image feedback and sensor data. Additionally, the firefighter can communicate with the victims through the robot’s speaker, and thus, guide them out of the scene. The robot relies on explicit control and is not autonomous.

Another search-and-rescue robot system for underground coal mine rescue is proposed by Zhao et al. [16]. Here, the robot is remotely controlled using Wifi and a fiber-optic cable and used to explore a coal mine shaft and to collect environmental information. The operating control unit incorporates an electronic compass, a gyroscope, two wheel encoders, and four infrared sensors. The infrared sensors are used to measure the distance of the robot to the wall to prevent unwanted crashes. The other sensors are used to deduce the motion trajectory and form part of the positioning function. Further, the authors use the image information from a camera to correct the position of the robot manually. Still, the robot is mainly piloted by a human and not fully autonomous.

Another remotely guided control system for rescue robots is developed by Mano et al. [17]. Depending on the tasks and working environment, it can switch efficiently between remote control mode and auto-detect mode to take full advantage of the robot’s functions using a SLAM-based map building. Functioning-wise this project comes closest to ours.

There are also researchers using the same type of robot used in this research, the Segway Loomo. For instance, Gollasch et al. [18] extended the Segways software API, and Steiner et al. [19] developed a ROS framework to improve Loomo’s navigation for greater stability and speed.

To summarize this subsection, we would like to look at two comprehensive overviews by Murphy et al. [20] and Delmerico et al. [1]. Both point to the need for rescue robots, demanding that they be robust, fast, versatile and, above all, highly user-friendly in use. Many rescue robots are prototypes, not consumer-ready and of high expense. Furthermore, these robots are to be piloted by a

trained user. In our research, we aim to demonstrate how to convert a highly developed consumer-ready robot, which is affordable and stable into a rescue robot that can even work autonomously, not requiring trained pilots.

2.2 Human Activity Recognition

The research field of human activity recognition is broad and incorporates a vast variety of approaches to detect the human’s activity [21], such as through wearables [22], and other distance sensing techniques and technologies, such as camera-based sensing [23]. As Loomo has a camera, which we utilize for sensing posture and heart rate, we focus on these two aspects in the following subsections.

Posture Recognition Human posture detection is one of the important application in human activity recognition. Often, machine learning approaches are utilized to recognize the posture of the entire body via camera/image sensing. Accurate recognition has shown to be a fundamental problem in computer vision in recent years [24]. There is extensive literature on this topic, as evidenced by the works of Bissacco et al. [25] and Dimitrijevic et al. [26].

Human posture detection and comprehensive analysis are required in different applications, such as motion analysis, monitoring, human-robot interaction, and medical rehabilitation. Anitha and Priya [27] build an automatic monitoring system for the elderly by sending an alert when an unusual posture is detected. Hernández [28] used posture detection in monitoring the exercises and providing useful information for Robotic-Assisted rehabilitation therapies.

Cao et al. [29] present the first real-time multi-person system, named Open-Pose, that collectively identifies key points of the human body, arms, face, and feet (a total of 135 key points) in an individual’s image. The main functionalities of 2D and 3D real-time multi-person keypoint detection, single-person tracking, and calibration toolbox are used. Bazarevsky et al. [30] developed a lightweight neural network to estimate human pose detection, which can also be used for real-time mobile devices. Their development has been integrated into Google ML Kit [5] and can be utilized by any mobile device via ML Kit’s API. The API’s high reliability rendered it an optimal choice for our research.

Heart Rate Recognition Heart rate recognition is vital to understand the user’s condition and thus is an objective in human activity recognition. Optical recognition of heart beats is possible as blood circulation creates a periodic change in the human body, although it is not perceivable to the eye. In 2011, Poh [31] used this weak signal captured by an ordinary RGB-camera to design a ”magic mirror”, which can reveal the user’s heart rate. The theory of Poh’s magical mirror is to use the change of light when blood flows in the human body [32]. The greater the amount of blood passing through the blood vessels, the more light is absorbed by the blood, and the less light is reflected from the surface of the human skin. Therefore, the heart rate can be estimated by

time-frequency analysis of the images. This is the main idea also denoted as Photoplethysmography (PPG).

However, this change of color in the skin is subtle and hard to capture by most cameras. In this case, can have to "magnify" these unobservable changes in the image to a magnitude sufficient to be observed by the naked eye. In 2012, Wu et al. [6] started from this perspective and proposed an algorithm called Eulerian Video Magnification (EVM) to reveal temporal variations in difficult or impossible videos to see with the naked eye.

Several works are developed based on this algorithm to detect heart rate successfully. Bosi et al. [33] used this algorithm to estimate heart rate using Microsoft device Kinect™ version 2.0. Similar to Gambi et al.' work [34], which also develop a heart rate detection on Kinect 2.0 camera, but further validated this method and compared it to classical wearable systems. Chambino et al. [35] developed an Android-based application for real-time monitoring of vital signs. We adopted this implementation for our research.

3 LoomoRescue

3.1 Concept

Robots that assist humans is an age-old vision and still an ongoing trend in human-robot interaction research. In this work, we would like to research whether current state-of-the-art consumer technology, the Segway Robot Loomo, can be utilized as a rescue robot. Therefore, our research is guided by two overarching research questions:

- RQ1: How can we enable Loomo to autonomously navigate through an environment?
 RQ2: How can we enable Loomo to understand human's critical activity?

3.2 Requirements

To answer our research questions, we set out certain requirements that Loomo should be able to provide.

Requirement 1: To answer the first question on autonomous navigation, we need to teach the rescue robot to understand its location in an environment. To fulfil this requirement, the robot provides a number of embedded sensors that can be utilized, such as the ultrasonic sensor, hall sensor, and inertia measure unit. Combining these sensor will, in theory, enable the robot to navigate without crashing and possible to organized a structure, such as a floor plan, once the robot explores an indoor-environment. The feasibility of that has already impressively demonstrated by Chen et al. [36]. In a hazardous scenario, there may not be time for time-expensive exploration and thus, a floor plan might already be provided to the rescue robot. In this case, the robot is required to exactly locate itself in that floor plan.

Requirement 2: To answer the second research question, we need to enable Loomo to find a person and understand the physical condition of a person. There are several ways to incorporate such functions, such as by listening (microphone), or looking (camera) through an environment. To judge on whether a person’s status is critical, the robot requires to have an understanding of typical human activity and anomalies. Loomo may be required to understand whether a person lies on the floor and whether the emitted vital signs, such as heart rate, respiration, and micro-vibrations, are abnormal. This is required to work instantaneously in real-time.

3.3 Design Decisions

To match the requirements, we need to make certain design decisions.

Decision 1: LoomoRescue needs to be effective and thus we decide to provide the robot with a floor plan. The problem of any robot is that it cannot locate itself in this environment without a positioning system. This results in robots not being fully autonomous. To overcome this, we equip Loomo with an external UWB position system, which consists of a tag carried by Loomo and tags distributed all over the floor.

Decision 2: The posture of the human body is an obvious sign to understand physical wellness. Therefore, we found it important to dynamically detect and analyze the posture of the human body. This should be performed visually by Loomo’s camera. Loomo should be able to classify dangerous postures, such as a person lying on the floor. Therefore, an advanced posture detection will be implemented in this work. Further we decided to enable Loomo to detect heart rate, which is an important vital sign.

3.4 Implementation

Loomo Device Loomo, firstly released in November 2016 and also known as the Segway Robot, is a combination of a self-balanced vehicle (SBV) and a companion robot. Loomo is 640mm in height, weighs 17.5 kg and has a battery capacity of 310 Wh. In programming mode, the robot can go up to 8 km/h, otherwise it can go faster. Loomo uses an Intel quad core CPU with 2.4 Ghz, and 4 GB RAM. The OS is based on Android that is programmable, movable, and expandable by Segway Robotics – a Mobile Robot Platform Kit [37]. The developer version enables to create a fully functional, stable, reliable, convenient, and easy-to-use robot development platform.

The provided Software Development Kit (SDK) enables developers to control the base functions at an abstract level, which include: the control, initialization, and configuration of the Intel RealSense camera; Speech recognition including pre-defined commands, base locomotion commands, the access to sensors, a connectivity package to enable linking a mobile device, and an emoji library that contains pre-defined ”human-like” sounds and eyes that will be drawn on Loomo’s display. On top of that there is the possibility for a hardware extension that allows mechanical and electrical engineers to design and attach additional

components to the robot. Since Loomo is ran by an Android System, we can also access a number of raw sensor data, such as the IMU, which is important in this project.

Loomo’s Sensors Loomo provides a number of sensor, which are essential for its functioning and for further development. These include: a RealSense RGB-Depth camera (30 FPS), a HD camera (1080p, 104 degree wide-angle), a microphone array composed of five mics, a touch sensors on the head, ultrasound sensor, hall sensors embedded in the wheels, two IMUs in the robot body and head. For our implementation, we particularly processed the raw data of the Ultrasound sensor and the IMU.

Ultrasound sensor: An ultrasonic sensor converts sonic signals into electrical signals. Ultrasound is a sound wave with a frequency of more than 20 kHz, which travels at 343 meters per second in air at 20 degrees Celsius. It is usually designed to measure the distance between an object and the transceiver that emits ultrasonic waves at 40 kHz [38]. The principle is to measure the latency between the emitted and received echo signal. This is also denoted as time of flight (ToF), which enables us to calculated the exact distance [39].

Inertial Measurement Unit (IMU): The IMU is a compound sensor, which usually consists of three linear accelerometers which measure acceleration, three gyroscopes which measure angular velocity rate [40]. Such type of IMU is also denoted as a 6-axis/DoF IMU. To encounter sensor drifts, a 9-axis/DoF includes a three-axis geomagnetic sensor, which can be used to correct the angular velocity sensor by the absolute pointing of the geomagnetic field to give accurate readings [41].

Reference Localization System and Communication via the MQTT Protocol

To ensure reliable self-localization of Loomo, we resort to a multi-anchor localization system based on Ultra Wide-Band signals (UWB), which can be easily integrated in an office building. Such multi-anchor system is based on the *two-way ranging* methodology (TWR) for distance estimation between a given *tag* connected to the Loomo and each anchor node installed static in the room geometry.

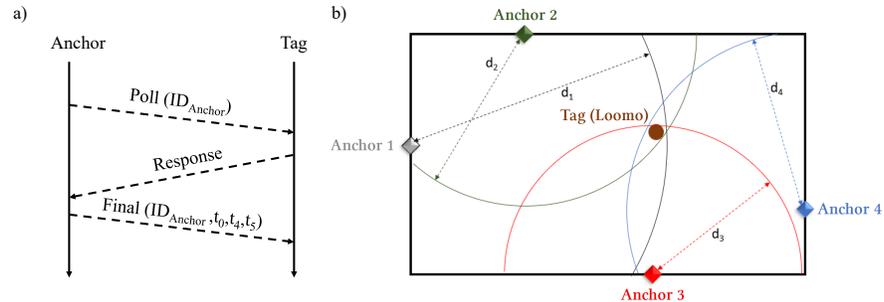


Fig. 2. Position estimation of the reference localization system applying two-way ranging and multilateration.

By using TWR, the anchors do not need to be synchronized in time, which decreases the amount of hardware interconnection between them. For each anchor separately, the transmission delay τ_{tag} between tag and anchor is estimated as shown in figure 2 a). For establishing TWR between one anchor and the tag, this anchor sends a *Poll*-message to the tag at time t_0 to initialize the TWR. The message includes an ID indicating the anchor unambiguously. The message is received at the tag at time t_1 . After a certain processing time, the tag sends at time t_2 a *Response*-message to the anchor, which is received at time t_3 . After processing this message, the *Final*-message is sent at time t_4 including the stored time stamps t_0 , t_3 and t_4 in the message's payload. The tag received this message at t_5 . The transmission delay, which is calculated at the tag's microcontroller results to:

$$\tau_{tag} = \frac{(t_3 - t_0) - (t_2 - t_1) + (t_5 - t_2) - (t_4 - t_3)}{4} = \frac{2t_3 - t_0 - 2t_2 + t_1 + t_5 - t_4}{4}$$

Due to the constant transmission velocity, namely the speed of light with $c_0 \approx 3 \cdot 10^8$ m/s, the distance results as multiplication of delay τ_{tag} and c_0 . Since only the distance to the tag is known, but not the direction, an distance radius for each anchor node returns.

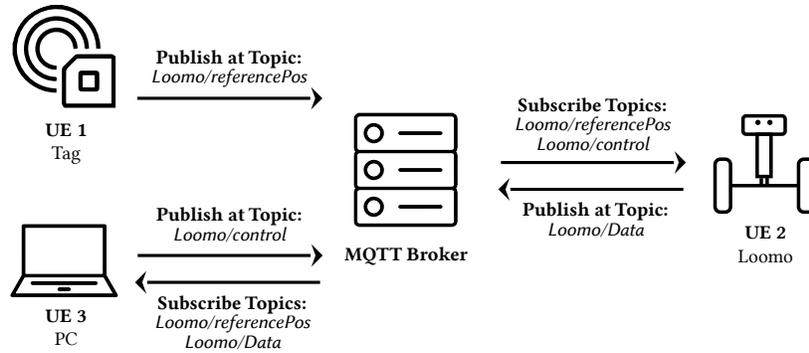


Fig. 3. User Equipments (UE) communicating via MQTT protocol.

As shown in figure 2 b), a position where the object is most likely to be located is obtained by *multilateration*, the iterative overlay of the different distance radii in the tag's microcontroller. Since the exact position of the anchor nodes is known by the tag, the overall position of the tag is calculable. Overall, the system results in an estimation accuracy of ± 20 cm in the currently installed state, which is why we use it without hesitation as a reference value for Loomo similarly as shown by Leugner et al. [42]. Built-in sensors from Loomo can further be used to increase accuracy.

To connect the Loomo with the tag's position estimation, we establish a connection including the *Message Queuing Telemetry Transport* protocol (MQTT). The MQTT is from the field of IoT-applications and enables the wireless data exchange of two or more User Equipments (UE) without initializing the communication between the UEs itself.

The general structure of the MQTT is shown in figure 3. It consists of a MQTT Broker in the center, which serves as a data hub. If a UE *publishes* data to the broker, the data itself is stored at the broker with a corresponding topic, consisting of a head topic and subtopics, which are ordered hierarchically. So, the data is indicated unambiguously. If new data with a topic is published by the same or another UE, the older data is deleted. So, no long term data storage including storage management is needed. Every UE connected to the MQTT broker is able to *subscribe* to specific head topics or subtopics. Then it receives all data stored at the MQTT with the specific topic close directly after storage. With MQTT connection of the tag, the Loomo and a remote PC, we are able to integrate the UWB multi-anchor system as reference to the Loomo.

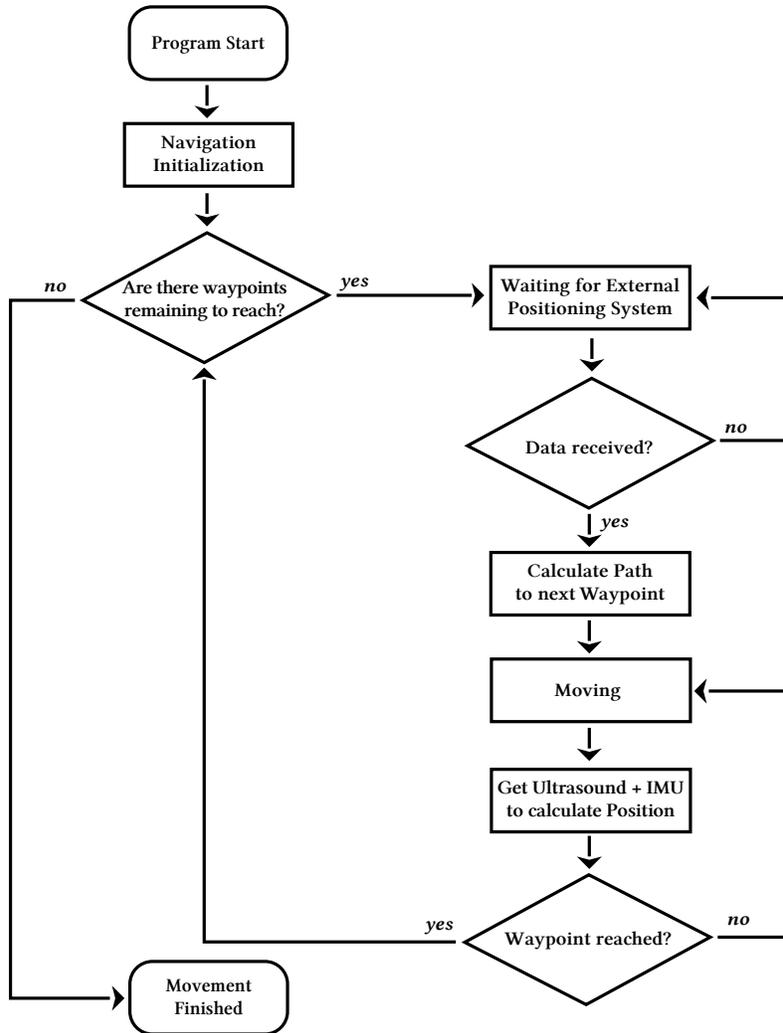


Fig. 4. Flow diagram depicting the decisions while moving through a floor.

Feature Extension 1: Localization and Movement The localisation system is key to guide the robot traverse the map, where the participation of sensors plays an important role. For the localization, there is an ultrasound sensor, IMU, and UWB, which was explained above. Given the 9-axis IMU may only have accurate and stable angle data we need to find a way to fuse it with the ultrasonic sensor and the UWB positioning system, creating a correction method for Loomo’s positioning. Previous works have showed how to impressively correct GPS by adding an IMU [43] in real-time. Other works demonstrated positioning correction with acoustic sensing using the Doppler-Shift [44] and utilizing acoustic sensing under device motion from a robot [45]. Another technology to accomplish indoor navigation and localization is using radar, as nicely showcased by Yue et al. [46].

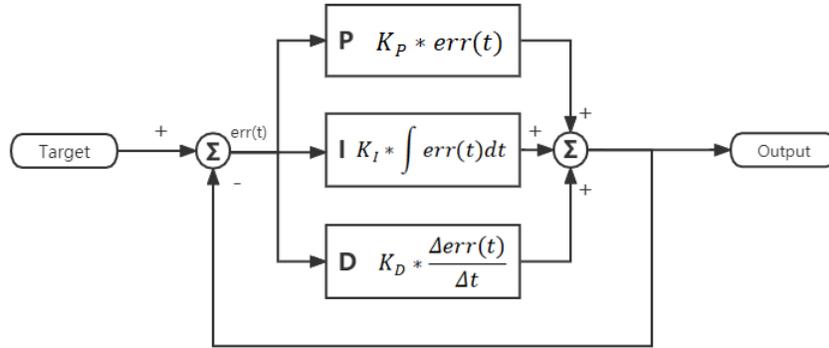


Fig. 5. PID Block Diagram: The PID algorithm is to utilize the feedback of position data to fix the input of Loomo’s motors. There are three elements for the feedback calculation. The P element can simply multiply the error with a coefficient, which gains a quick response value. The I element mean the integral errors in a time period, so it enables fine-tuning of the robot over short distance. The element of D estimates the trends of movement and thus can perform an adjustment of the speed.

3.4.4.1 Internal Localization and Move Control: The original control command in the Loomo SDK API provides ultrasound data, which are as accurate as even reflecting wheel movement for instance. This inference of movement, however, creates a large error when processing it in this form. Therefore, we also utilize the IMU in combined to improve the PID algorithm for the movement control (see figure 5). Both, the ultrasound sensor and the IMU have a quick update rate of 50 milliseconds. This is a high response rate and thus suitably to meet the demand of real-time movement control.

3.4.4.2 External Localization: It would be sufficient to control robot’s movement though the method explained above. However, there is a problem: the calculation of current position heavily depends on the former position. Since sensors present drifts and inaccuracies, the errors will accumulate with time and show too great deviation after a while. Thus an external positioning system is introduced to improve the positioning accuracy and to correct the error. The

UWB positioning system, as mentioned above, can be used as an additional source of information that provides coordinates, while it supports full map navigation functionality. In this setup we have two independent localization systems working respectively. The external UWB system will guide the robot traverse the floor through waypoints. However, entering a room, for instance, requires higher accuracy than ± 20 cm and therefore, we fall back on internal sensors to avoid LoomoRescue bumping against the door frame. Figure 4 shows a flow chart on how our navigation is designed.

Feature Extension 2: Human Activity Recognition To fulfil our requirements set out, we are required to develop an addition function for Loomo, so it can understand the human activity. As discussed earlier, to identify a critical vital state, we decided to include a detection of posture as well as a heart rate detection. With these features, Loomo Robot would be enabled to estimate the condition of the person.

Since Loomo is unable to touch the person, we decided to use a vision-based approach. To not re-invent the wheel, we will rely on established machine learning libraries: OpenCV [47] and the ML Kit by Google’s to enable on-device machine learning [5]. Both libraries offer android packages, which can be integrated with Loomo. To get OpenCV properly running, we required to included an additional Java Native Interface (JNI) that would execute C++ code.



Fig. 6. Showing three classified of the implemented posture detection. The skeleton is colored in accordance to the criticalness, when the video feed is displayed at the Loomo device.

3.4.5.1 Posture Detection: Google’s ML kit’s pose detector provides us with up to 33 landmarks that are generation from an image at which a human is identified. The implementation works in real-time once provided the video feed from the HD camera. To classify postures, which is the main objective of our pose detector, the relative placements of these landmarks are calculated in a plane rule-based manner similarly as demonstrated in the PhD Thesis of Rithik Kapoor [48]. This way, we categorized three critical stages of postures, as illustrated in Figure 6:

- **Normal:** Representing a category of posture that is usually not considered as dangerous, such as standing straight on the ground.
- **Warning:** Representing a category of posture which might mean this person needs additional help in an emergency scenario. As ”warning” we classified

someone that remains sitting in an emergency scenario such as an event of a fire.

- **Danger:** Representing a category of posture that is usually considered as dangerous. We classified any postures as "danger" at which a person is somewhat lying on the ground, including any postures the person shows an awkward horizontal posture.

3.4.5.2 Heart Rate Detection: Our heart rate detection contains three major steps: A) Face detection, B) Eulerian Video Magnification (EVM) magnification, and C) Photo-Plethysmography (PPG) signal processing.



Fig. 7. Heart rate detection on Loomo when toggled on the video feed.

In step A, we use the OpenCV library to detect the face as the Region of Interest (ROI) from each video frame. We initialize a cascade classifier to identify faces based on a pre-trained model (*lbpcascade_frontalface.xml*) that can be obtained through the official OpenCV website. The application basically extracts a rectangle area that contains the ROI.

In step B, we look at these sub-frames, which will be processed by the EVM algorithm. The EVM performs a spatial filter to decompose the frames by blurring, differentiating, and down sampling the image with a Gaussian pyramid. After obtaining the different spatial bands, a temporal filter is implemented on each spatial band to select bands of interest by frequencies. In our case, we want to amplify the heart rate signal and select $f_L = 0.4$ to $f_H = 4$ Hz (24 to 240 bpm) for bandpass filtering, which is approximately the range of human heart rate. This seems to be the appropriate cutoff frequencies as shown in related work [6]. Finally, we amplify the change in color and add the magnified result to the origin frame.

In step C, the raw Photo-plethysmography (PPG) signal that was obtained by the color-magnified result, will be sent to further processing, which includes signal smoothing and a peak detection.

We utilize the average green channel from the magnified video frame sequence, which is smoothed by a moving average filter aiming to reduce random

noise while retaining its peaks as follows

$$y[i] = \frac{1}{n} \sum_{j=0}^{n-1} x[i+j] \quad (1)$$

Where \mathbf{y} is the output signal, x is the input signal, n is the length of queue. The moving average filter views the successive sampled data as a queue of fixed length n . After inserting a new data, the first data of the above queue is removed, the remaining $n-1$ data are moved forward in turn, and the newly sampled data are inserted as the tail of the new queue. Then the arithmetic operations are performed on this queue and obtain the i^{th} output result. The Loomo robot has a self-balance movement, which is a periodical forward and backward movement. This makes the relative position of the light source to change constantly in a certain interval, which needs to be considered and removed from the data. Finally, the estimated heart rate is calculated by finding the peaks of the smoothed PPG signal. Assume the number of series of frames is L , the frame rate is calculated in FPS , and there are N peaks detected in this series of frames. The heart rate in beats per minute (BPM) is computed as follows:

$$BPM = 60 \frac{N \times FPS}{L} \quad (2)$$

The result of implemented heart rate detection is displayed in Figure 7.

4 Evaluation

To quantify the effectiveness of our implementation, we ran an evaluation on our added features: Localization and Human Activity Recognition (posture & heart rate detection).

4.1 Feature Extension 1: Localisation

In terms of localisation, we have two parts in total. The UWB positioning system is used for global positioning and navigation on the map, while each movement in the local area is controlled using the robot’s built-in ultrasonic sensor in

Table 1. Linear Movement Test

Trial	Expected Distance: 100cm			Expected Distance 200cm		
	Actual (cm)	Delta (cm)	Error (%)	Actual (cm)	Delta (cm)	Error (%)
1	99.6	0.4	0.4	197	3	1.5
2	98.2	1.8	1.8	202.8	-2.8	1.4
3	100.6	-0.6	0.6	195.4	4.6	2.3
4	96.4	3.6	3.6	198.7	1.3	0.65
5	102.8	-2.8	2.8	194.9	5.1	2.55
6	101.4	-1.4	1.4	202.6	-2.6	1.3
7	97.0	3	3	200.5	-0.5	0.25
8	97.5	2.5	2.5	199.3	0.7	0.35

combination with the IMU using the PID algorithm. The UWB system has already been experimentally determined to have a mean error of ± 20 cm in an indoor environment following studies by Leugner et al. [42] and Schmidt et al. [49]. Therefore, the performance of the robot’s internal navigation will be evaluated here.

Test of Linear Movement In the test of linear movement, we have two conditions. First we let the robot travel 1 meter along a straight line and afterwards 2 meters. For both conditions, we measured the distance from its starting point and calculated the error. The purpose is to test the accuracy of a single move and the effect of different distances on the accuracy.

As shown in table 1, through eight trials, when the robot advances 1 meter, with the help of our sensor approach, its endpoint difference averages at 2.01 cm. At the second condition, the average difference was at 2.58 cm when advancing 2 meters. A paired *t*-Test $T(7) = -0.64$, $p = 0.27$, showed the deviations occurring at 1 meter ($M = 2.01$ cm; $SD = 1.16$ cm) and at 2 meters ($M = 2.58$ cm; $SD = 1.69$ cm) to be not statistically different. The average error of 1.65 % is substantially smaller than that of the UWB positioning system. However, the UWB system does not accumulate errors over time.

Test on Angular movement Since the UWB positioning system does not measure angles, the global and local angle indications are all dependent on the IMU, which require high accuracy. In our angular movement test, we let the robot rotate 45 degrees and 90 degrees on the ground, while we marked the starting and ending positions in order to calculate the errors.

Table 2. Angular Movement Test

Trial	Expected Angle: 45°			Expected Angle: 90°		
	Actual (°)	Delta (°)	Error (%)	Actual (°)	Delta (°)	Error (%)
1	44.23	0.77	1.71	88.85	1.15	1.28
2	43.36	1.64	3.64	90.46	-0.46	0.51
3	41.34	3.66	8.13	89.8	0.2	0.22
4	44.19	0.81	1.80	91.45	-1.45	1.61
5	43.33	1.67	3.71	91.55	-1.55	1.72

As shown in table 2, through five trials, the average difference of the robot was 1.71 degrees at 45 degrees of rotation. When Loomo rotates 90 degrees, the average difference is at 1.07 degrees. A paired *t*-Test $T(4) = 1.002$, $p = 0.18$, showed the deviations occurring at 45 degrees ($M = 1.71^\circ$; $SD = 1.17^\circ$) and at 90 degrees ($M = 0.96^\circ$; $SD = 0.6^\circ$) to be not statistically different. We consider the overall deviation of 1.34° of angular movement to be quite high. Even if the error of angular measurement accumulates over an extensive time period of use, we have the UWB system to correct the location.

4.2 Feature Extension 2: Human Activity Recognition

Posture Detection As we used a related work, namely Google’s ML kit, to extract the feature points and human skeleton, there is no point in evaluating their trained model, which seems sophisticated to us. The implementation runs stable and in real-time. Our rule-based classification also showed great accuracy, but which showed difficulties at different distances between human and robot. Therefore, we measured the detection accuracy with four different distances. Each posture was executed in four distances. This was repeated 20 times, which resulted in 240 trials in total. The result is displayed in the Table 3.

Table 3. Successful detection rate by distance in percent (20 trials each).

Distance d (m)	Normal (%)	Warning (%)	Danger (%)
$d < 1$	75	15	25
$1 \leq d < 5$	85	75	70
$5 \leq d < 10$	90	85	80
$10 \leq d < 20$	70	75	75

From the table, we can obtain that the probability of getting an accurate pose within a certain measurement distance seems to be different. However, there is no statistical detection difference between all three postures ($F_{2,6}=1.71$; $p=0.26$) following a one-way ANOVA for correlated samples. The average success rate is 68.3% ($SD = 23.5\%$) among all distances. Not considering distances shorter than 1 meter, the average detection rate raises to 78.33% ($SD = 7.1\%$)

It is striking that the successful detection seems to be compromised with distances below 1 meter. This is indicated by a one-way ANOVA for correlated samples ($F_{3,6}=1272.22$; $p=0.039$) and finally evidenced by a post-hoc analysis using a Tukey HSD Test. The detection rate is significantly lower in distances < 1 m than distances between ≤ 5 m and < 10 m ($p<0.05$). No further differences were found.

Generally, it become obvious that sufficient detection with too close distances may not be guaranteed. The reason is that the identification of the posture is accomplished by calculating the relative position of the detected landmarks. Therefore, this detection and identification work well when the needed body landmark is included in the frame. However, if some of the body landmarks are not captured by the input frame, the information about his landmark will be missing. Thus, reducing the success rate of the obtained results. The detection error caused by missing landmarks is particularly noticeable at short distances because the camera cannot capture the whole body of the inspector well.

Heart Rate Detection To gain insights into the performance of Loomo’s newly acquired heart rate detection, we ran a study with 13 participants (Computer Science students aged between 21 and 26). As ground truth of heart rate, we used a Pulse Oximeter and the Apple watch Gen. 5, which showed almost no difference and is in line with the findings from Pipek et al. [50]. For each user we conducted a single measurement. The study results are shown in Table 4.

Table 4. Study results: Showing the calculated heart rate by Loomo against the ground truth [in bpm]

Participant	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
Loomo (bpm)	83	65	74	78	69	67	75	68	70	74	61	73	67
Ground Truth (bpm)	110	81	75	96	81	72	88	82	80	80	66	78	76
Difference (bpm)	-27	-16	-1	-18	-12	-5	-13	-14	-10	-6	-5	-5	-9

The study results show a mean deviation of -10.84 heart beats per minute. Loomo’s calculated heart rate is 87.3% accurate to the ground truth. We ran a Bland-Altman analysis, which ensured no bias to exist. This is confirmed by a Pearson correlation, which showed a positive correlation coefficient, with an average of $r = .83$ ($SD = .69$). All participants demonstrated a statistically significant positive relationship ($r > .7$, $p < .05$). However, it must also be stated that the heart rate sometimes showed wide range, differing of up to 27 beats, which is almost 25% (P1). Also striking is that Loomo provided a constantly lower heart rate than the actual ground truth. We suspect our noise filter to have canceled out the rather weak heart beats. Although this heart rate detection method is not able to extract the exact pulse signal, Loomo still achieves the goal of identifying vital signs, which is the overarching goal in this research.

5 Discussion

In this section, we would like to critically discuss the limitations of our work and to provide possible approaches to overcome these:

Classifying Critical Postures: Currently, we drove a quick’n dirty approach to classify three critical stages of posture by calculating the relations of the pose landmarks to each other. Although this approach showed to be fine, one could employ a training a simple ML-model, such as using a simple C4.5 decision tree.

Coping with Loomo’s self-balancing movement: Since Loomo is standing on two wheels only, it deploys a bouncing back and forward movement to avoid the robot to fall over. These periodic movements impact all sensor data. There are several approaches that can be used to eliminate these signal in the sensor data, such as deploying a butterworth filter, a principle component analysis, etc.

Valid Vital Signs: Collecting accurate vital signs, such as the correct heart rate, is a challenge. For a remote PPG signal detection, like ours, we see a variety of environmental factors, such as changing ambient light, shadows, natural movement of user, etc. that can impact the measurement significantly. In our use case clinical valid data is not required, however, it is desirable.

Improving Heart Rate Detection: The simple moving average filter seems not to be the perfect choice to handle the raw PPG signal. This is because rather weak heart beats might be smoothed to much this way. A better approach might be the signal peak detection algorithm, such demonstrated by Jang et al. [51].

Increasing number of Add-ons: Loomo already provides rich functionality with the standard sensors provided. Adding a variety of unique sensors to Loomo might open up new application field or enhance Loomo’s capabilities to make

autonomous decisions. For instance, attached gas sensors may provide crucial information on the environment. Moreover, using electric field sensors can help deciding whether danger by electric dives occurs or whether human body activity is present.

Deployment in Reality: In this research, we provided a proof-of-concept implementation, showcasing future opportunities. In reality, however, a greater number of studies and development is required. For instance, it is unclear how the form factor of a dual-wheeled robot like Loomo can overcome obstacles. In a disaster situation, such as when a certain gas like carbon monoxide is released, Loomo could help in finding survivors. However, in an earth quake situation when a building structure is destroyed, Loomo is likely to fail in making its way through rough terrain. Further it is unclear how false-positives and true-negatives are handled, such as missing out on finding an immobilized person, because a body part might be covered. Another problem is finding the face in unconventional positions. Therefore, it is at least questionable whether a future robot can yet to be fully autonomous or whether human remote control is required for support.

Robust Infrastructure: A future implementation is required to have a robust communication system that needs to be put in place, on a software and hardware level. High bandwidth seem to be a basis for video-streaming that may be essential in a disaster scenario, so the robot can also be remotely controlled by a rescue team. Further some type of position system is required to be in place, which may account to as an additional cost as well as being another source of potential error.

Intelligent Path Planning Algorithm: What is the quickest way? Advanced path planning may be required to enable to robot to make intelligent decisions on its own without being remotely controlled, such as when a network connection broke down. Is there a safe area? Where is the robot going once followed?

Improving Navigation by SLAM: In this project, the effects and possibilities of multi-sensor fusion are tried in the direction of a fire-fighting robot. However, because of the involvement of external position system. There is high requirement of deployment environment. Plus, a receiver mounted makes the robot too large in size. So a better substitution of UWB position system is visual SLAM. With the help of in-built camera, visual SLAM can decrease the demand of environment and make robot more flexible during task.

Feature Point Extraction: The feature points are extracted between frames by ORB method, which is an algorithm that can figure out the same points in two pictures continuously taken [52]. Feature point extraction matters both in SLAM and posture recognition that when doing image processing we can only take feature points in account, and thus reduce computation burden greatly.

6 Conclusion & Future Work

The Loomo Segway is one of the few consumer robots that offer high mobility, increased computing power, and sensing capabilities. Unfortunately, Loomo is

only used for entertainment purposes and does not make use of its potential. Yet, in addition to a follow function that works questionably well, the device is only controlled manually, while it cannot understand a persons condition. In this paper we showcased a new application, a proof-of-concept of an affordable rescue robot that may be deployed in evacuation scenarios. We demonstrated how to instrument Loomo, in order to navigate autonomously through an indoor space. Further, we showed how to estimate a victims activity state, by classifying their posture and detecting their heart rate, both by using an optical RGB camera approach. We evaluated the detection quality of our approach and conclude as follows to our research questions. RQ1: We can enable Loomo to autonomously navigate through a known environment by equipping it an external UWB position system and driving a SLAM-approach. RQ2: We can we enable Loomo to understand the human’s critical activity by extracting posture and heart rate information from the RGB-camera feed.

By combining these two techniques, in future rescue robots can be improved by not only reducing the cost but also enhancing their ability to recognize the victim’s health condition. Particularly the application of SLAM algorithms allow robots to autonomously perceive and explore their environment, which is a future direction for investigation. Combined with path planning algorithms, unmanned autonomous search-and-rescue robots seem promising. We see the application in evacuation scenarios where the robot sends the location of injured people to the rescue team. Its unmanned nature means that it can be deployed in hazardous environments and with multiple units to effectively increase search and rescue.

References

1. J. Delmerico, S. Mintchev, A. Giusti, B. Gromov, K. Melo, T. Horvat, C. Cadena, M. Hutter, A. Ijspeert, D. Floreano, *et al.*, “The current state and future outlook of rescue robotics,” *Journal of Field Robotics*, vol. 36, no. 7, pp. 1171–1191, 2019.
2. R. Ventura and P. U. Lima, “Search and rescue robots: The civil protection teams of the future,” in *2012 Third International Conference on Emerging Security Technologies*, pp. 12–19, IEEE, 2012.
3. H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part i,” *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006.
4. N. D. with Us, “Uwb minutes: Ranging technics,” 2020.
5. G. M. Kit, “Google,” 2022.
6. H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. T. Freeman, “Eulerian video magnification for revealing subtle changes in the world,” *ACM Transactions on Graphics (Proc. SIGGRAPH 2012)*, vol. 31, no. 4, 2012.
7. J. O. Wobbrock and J. A. Kientz, “Research contributions in human-computer interaction,” *interactions*, vol. 23, no. 3, pp. 38–44, 2016.
8. S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, “Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot,” *Autonomous robots*, vol. 40, no. 3, pp. 429–455, 2016.
9. M. Schwarz, T. Rodehutsors, D. Droschel, M. Beul, M. Schreiber, N. Araslanov, I. Ivanov, C. Lenz, J. Razlaw, S. Schüller, *et al.*, “Nimbro rescue: Solving disaster-response tasks with the mobile manipulation robot momaro,” *Journal of Field Robotics*, vol. 34, no. 2, pp. 400–425, 2017.
10. D. Falanga, K. Kleber, S. Mintchev, D. Floreano, and D. Scaramuzza, “The foldable drone: A morphing quadrotor that can squeeze and fly,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 209–216, 2018.

11. A. Wojciechowska, J. Frey, E. Mandelblum, Y. Amichai-Hamburger, and J. R. Cauchard, "Designing drones: Factors and characteristics influencing the perception of flying robots," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 3, pp. 1–19, 2019.
12. C. Ruiz, S. Pan, A. Bannis, X. Chen, C. Joe-Wong, H. Y. Noh, and P. Zhang, "Idrone: Robust drone identification through motion actuation feedback," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 2, pp. 1–22, 2018.
13. C. Wright, A. Johnson, A. Peck, Z. McCord, A. Naaktgeboren, P. Gianfortoni, M. Gonzalez-Rivero, R. Hattton, and H. Choset, "Design of a modular snake robot," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2609–2614, IEEE, 2007.
14. M. N. Kiyani and M. U. M. Khan, "A prototype of search and rescue robot," in *2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI)*, pp. 208–213, 2016.
15. Y.-D. Kim, Y.-G. Kim, S.-H. Lee, J.-H. Kang, and J. An, "Portable fire evacuation guide robot system," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2789–2794, 2009.
16. J. Zhao, J. Gao, F. Zhao, and Y. Liu, "A search-and-rescue robot system for remotely sensing the underground coal mine environment," *Sensors*, vol. 17, no. 10, p. 2426, 2017.
17. H. Mano, K. Kon, N. Sato, M. Ito, H. Mizumoto, K. Goto, R. Chatterjee, and F. Matsuno, "Treaded control system for rescue robots in indoor environment," in *2008 IEEE International Conference on Robotics and Biomimetics*, pp. 1836–1843, 2009.
18. D. Gollasch, C. Engel, M. Branig, and G. Weber, "Applying software variability methods to design adaptive assistance robots," in *Proceedings of the 12th ACM International Conference on Pervasive Technologies Related to Assistive Environments, PETRA '19*, (New York, NY, USA), p. 313–314, Association for Computing Machinery, 2019.
19. M. Steiner, *ROS Navigation Stack On A Loomo Segway Robot*. PhD thesis, BSc Thesis, Vienna University Munich Vienna, 2018.
20. R. Murphy, "Human-robot interaction in rescue robotics," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 34, no. 2, pp. 138–153, 2004.
21. C. Jobanputra, J. Bavishi, and N. Doshi, "Human activity recognition: A survey," *Procedia Computer Science*, vol. 155, pp. 698–703, 2019.
22. O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2012.
23. S.-R. Ke, H. L. U. Thuc, Y.-J. Lee, J.-N. Hwang, J.-H. Yoo, and K.-H. Choi, "A review on video-based human activity recognition," *Computers*, vol. 2, no. 2, pp. 88–131, 2013.
24. G. Rogez, J. Rihan, S. Ramalingam, C. Orrite, and P. H. Torr, "Randomized trees for human pose detection," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.
25. A. Bissacco, M.-H. Yang, and S. Soatto, "Detecting humans via their pose," *Advances in Neural Information Processing Systems*, vol. 19, 2006.
26. M. Dimitrijevic, V. Lepetit, and P. Fua, "Human body pose detection using bayesian spatio-temporal templates," *Computer vision and image understanding*, vol. 104, no. 2-3, pp. 127–139, 2006.
27. G. Anitha and S. Baghavathi Priya, "Posture based health monitoring and unusual behavior recognition system for elderly using dynamic bayesian network," *Cluster Computing*, vol. 22, no. 6, pp. 13583–13590, 2019.
28. Ó. G. Hernández, V. Morell, J. L. Ramon, and C. A. Jara, "Human pose detection for robotic-assisted and rehabilitation environments," *Applied Sciences*, vol. 11, no. 9, p. 4183, 2021.
29. Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 172–186, 2021.
30. V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "Blazepose: On-device real-time body pose tracking," *arXiv preprint arXiv:2006.10204*, 2020.

31. M.-Z. Poh, D. J. McDuff, and R. W. Picard, “Non-contact, automated cardiac pulse measurements using video imaging and blind source separation.,” *Opt. Express*, vol. 18, pp. 10762–10774, May 2010.
32. W. Verkrusse, L. O. Svaasand, and J. S. Nelson, “Remote plethysmographic imaging using ambient light.,” *Opt. Express*, vol. 16, pp. 21434–21445, Dec. 2008.
33. I. Bosi, C. Coggerino, and M. Bazzani, ““real-time monitoring of heart rate by processing of microsoft kinect™ 2.0 generated streams” ,” in *2016 International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*, pp. 1–6, 2016.
34. E. Gambi, A. Agostinelli, A. Belli, L. Burattini, E. Cippitelli, S. Fioretti, P. Pierleoni, M. Ricciuti, A. Sbröllini, and S. Spinsante, “Heart rate detection using microsoft kinect: Validation and comparison to wearable devices,” *Sensors*, vol. 17, no. 8, 2017.
35. P. B. Chambino, “Android-based implementation of eulerian video magnification for vital signs monitoring,” 2013.
36. W. Chen, F. Zhang, T. Gu, K. Zhou, Z. Huo, and D. Zhang, “Constructing floor plan through smoke using ultra wideband radar,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 4, pp. 1–29, 2021.
37. S. Robotics., “Segway Robotics | Developer,” 2022.
38. G. Arun Francis, M. Arulselvan, P. Elangkumaran, S. Keerthivarman, and J. Vijaya Kumar, “Object detection using ultrasonic sensor,” *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, pp. 207–209, 2020.
39. L. Koval, J. Vaňuš, and P. Bilík, “Distance measuring by ultrasonic sensor,” *IFAC-PapersOnLine*, vol. 49, no. 25, pp. 153–158, 2016. 14th IFAC Conference on Programmable Devices and Embedded Systems PDES 2016.
40. P. Zhang, J. Gu, E. Milios, and P. Huynh, “Navigation with imu/gps/digital compass with unscented kalman filter,” in *IEEE International Conference Mechatronics and Automation, 2005*, vol. 3, pp. 1497–1502 Vol. 3, 2005.
41. Y. Zhang, Y. Fei, L. Xu, and G. Sun, “Micro-imu-based motion tracking system for virtual training,” in *2015 34th Chinese Control Conference (CCC)*, pp. 7753–7758, 2015.
42. S. Leugner and H. Hellbrück, “Lessons learned: Indoor Ultra-Wideband localization systems for an industrial IoT application,” tech. rep., Technische Universität Braunschweig, Braunschweig, 2018.
43. S. S. Saha, S. S. Sandha, L. A. Garcia, and M. Srivastava, “Tinyodom: Hardware-aware efficient neural inertial navigation,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 2, pp. 1–32, 2022.
44. X. Chen, Y. Chen, S. Cao, L. Zhang, X. Zhang, and X. Chen, “Acoustic indoor localization system integrating tdma+ fdma transmission scheme and positioning correction technique,” *Sensors*, vol. 19, no. 10, p. 2353, 2019.
45. J. Liu, D. Li, L. Wang, F. Zhang, and J. Xiong, “Enabling contact-free acoustic sensing under device motion,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 3, pp. 1–27, 2022.
46. S. Yue, H. He, P. Cao, K. Zha, M. Koizumi, and D. Katabi, “Cornerradar: Rf-based indoor localization around corners,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 1, pp. 1–24, 2022.
47. G. Bradski and A. Kaehler, “Opencv,” *Dr. Dobb’s journal of software tools*, vol. 3, p. 120, 2000.
48. R. Kapoor, *Creating and Comparing Seated Posture Classification Models Using Machine Learning and Computer Vision*. PhD thesis, 2022.
49. S. O. Schmidt and H. Hellbrück, “Detection and identification of multipath interference with adaption of transmission band for uwb transceiver systems.,” in *IPIN-WiP*, 2021.
50. L. Z. Pipek, R. F. V. Nascimento, M. M. P. Acencio, and L. R. Teixeira, “Comparison of spo2 and heart rate values on apple watch and conventional commercial oximeters devices in patients with lung disease,” *Scientific Reports*, vol. 11, no. 1, pp. 1–7, 2021.
51. D.-G. Jang, S. Park, M. Hahn, and S.-H. Park, “A real-time pulse peak detection algorithm for the photoplethysmogram,” *International Journal of Electronics and Electrical Engineering*, pp. 45–49, 2014.
52. X. Gao, T. Zhang, Y. Liu, and Q. Yan, “14 lectures on visual slam: from theory to practice,” *Publishing House of Electronics Industry, Beijing*, 2017.