A Robotic Disaster Response System for Autonomous Inspection, Respiration Rate Estimation, and Amputation Detection of Casualties

Mayank Mishra

CMU-RI-TR-25-73
July 18, 2025

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Dr. Srinivasa Narasimhan, *Chair*
Dr. Sebastian Scherer
Dr. László A Jeni
Khiem Vuong

Submitted in partial fulfillment of the requirements for the degree of Masters in Robotics.

To my family.

Abstract

In mass casualty situations, quickly identifying and prioritizing injuries can be the difference between life and death. Traditional triage systems rely heavily on trained human responders, but in complex or hazardous environments, human access may be delayed or limited. Robotic systems offer a promising solution to this problem by enabling remote, scalable, and fast health assessments. The recently launched DARPA Triage Challenge aims to accelerate progress toward autonomous triage robots that can evaluate the physiological state of human casualties using non-contact sensing methods.

This thesis presents our contributions developed as part of the DARPA Triage Challenge, combining methods for physiological sensing with autonomous robotic navigation. We present two key algorithms for casualty assessment: (a) a lightweight algorithm for estimating respiration rate using monocular RGB video, and (b) a geometry-based technique for 3D human pose estimation, applied to the detection of possible amputations. On the navigation side, we design an autonomous system that enables a ground robot to move around a casualty and collect multi-view observations critical for robust physiological inference. Finally, we show how these methods work together within a deployed robotic platform, helping move closer to real-time, autonomous casualty assessment in the field.

Acknowledgments

First and foremost, I would like to express my gratitude to my advisor, Dr. Srinivasa Narasimhan. I am thankful to him for not just introducing me to a broad variety of topics in vision, but also for teaching me the importance of focus and the ability to adapt in solving challenging problems. I am deeply grateful to Dr. Sebastian Scherer for opening the door to the world of robotic systems. His relentless problem-solving and way of leading by example have left a strong imprint on me. I would also like to thank Dr. László Jeni for many insightful conversations regarding human assessment and sensing.

I have made many wonderful memories working with Team Chiron for the DARPA Triage Challenge. First, I would like to thank Dr. Kimberly Elenberg, Dr. Artur Dubrawski, Dr. John Galeotti, and Dr. Lenny Weiss for their unwavering motivation and support in addressing both research challenges and logistical demands of deploying a large-scale robotic system. As integral to working in a robotics challenge, we participated in countless field tests, traveled together for competitions, workshops, and conferences. I am thankful to a dynamic group of colleagues - Yaoyu, Kabir, Ceci, Aniket, Parv, Varun, Viktor, Adi, James - for sharing both the frustrations and breakthroughs of our journey. In particular, I would like to thank Kabir and Yaoyu for always going above and beyond to help and ensure a smooth deployment of algorithms on the system. I have also been lucky to be a part of the ILIM lab. I thank Mark, Mani, Anurag, Sriram, and Khiem for their guidance and the many engaging (and often spirited) discussions, both in research and beyond. I am also thankful to Aki, Ji, Leo, and Shen for making the lab a lively place to work.

I would like to thank Sumukh for being an amazing roommate, and the SSS organisation of Pittsburgh for helping me avoid feeling homesick. A special thanks to Anisha for being a constant source of strength and encouragement. At last, I would like to thank my parents, brother, cousins, and extended family, without the support and blessing of whom, this work would not have been possible.

Funding

This work was supported by the DARPA Triage Challenge Contract.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

Chapter 1

Introduction

1.1 Motivation

Disaster scenarios require rapid rescue efforts for minimizing casualties. Human responders often face significant challenges in reaching disaster sites and providing assistance in a timely manner. Several factors contribute to delays, including but not limited to remote or inaccessible disaster sites, an insufficient number of responders, and communication system failures.

Recent progress in the field of robotics has offered a new hope in creating robust systems that can assist in search and rescue operations. Modern computer vision systems can provide robots with the ability to perceive the world efficiently, and advanced localization, mapping and planning techniques can guide a robot in unconstrained environments. Alongside these algorithmic improvements, significant progress in robotics hardware in terms of robotic platforms, sensors, etc. has been seen. These advances motivate the design of a comprehensive robotic system capable of assisting in early triage by locating casualties, assessing their condition, and prioritizing medical intervention.

DARPA Triage Challenge

The DARPA Triage Challenge [20], launched in 2023, is a three-year initiative focused on developing autonomous systems capable of conducting remote triage in complex and degraded environments. The central goal is to detect and assess physiological signs of injury, such as heart rate, respiratory rate, hemorrhage, trauma, and alertness, using stand-off sensors mounted on mobile robotic platforms (Fig. 1.1). These systems must operate without direct contact with casualties, enabling early identification and prioritization of those needing urgent medical care. This capability is particularly vital in large-scale disasters where human first responders may be delayed or limited in number.

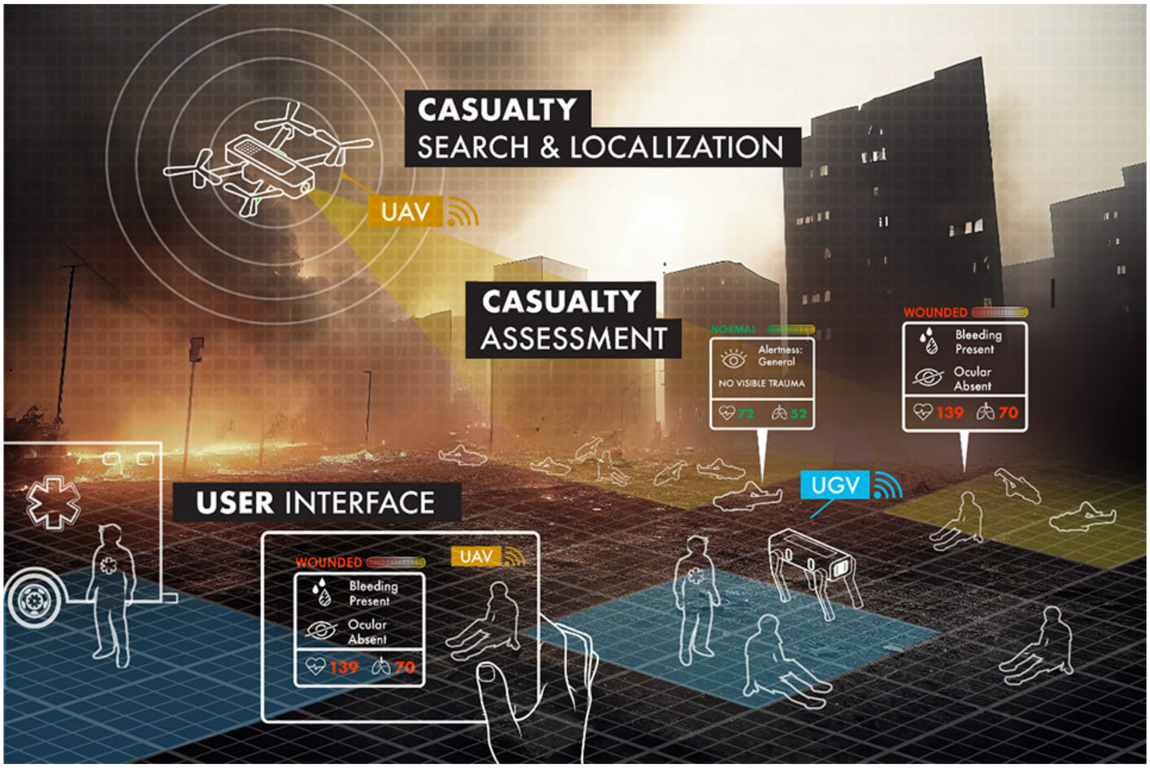


Figure 1.1: An overview of the DARPA Triage Challenge. (*Image credit: DTC*)

To reflect real-world constraints, the challenge scenarios are designed with low visibility, smoke, dynamic terrain, obscured victims, and moving obstacles. Each participating team gets 20 minutes to search and assess around 10-12 casualties. The challenge thus serves as a rigorous benchmark for advancing integrated robotic systems that combine perception, navigation, and physiological analysis.

This thesis explores and develops methods that enhance a robot's ability to autonomously navigate around humans and assess physiological state from a distance.

These efforts contribute to Team Chiron[1]'s larger, evolving triage system in the DARPA Triage Challenge.

1.2 Background

In the aftermath of disasters, search and rescue (SAR) operations aim to locate, assist, and extract victims trapped or injured in dangerous environments. Over the past two decades, robotics has emerged as a critical complement to human responders, especially in scenarios that are hazardous, structurally unstable, or otherwise inaccessible [9, 43, 60, 68]. Notable early deployments of rescue robots include the 9/11 World Trade Center collapse, the 2004 Niigata-Chuetsu earthquake, and the 2011 Great East Japan Earthquake and tsunami, which collectively demonstrated the value of robotics in complex, hazardous disaster response environments [42, 44, 46].

Robot Locomotion and Navigation: Effective locomotion, along with robust environment mapping, localization, and planning, are central to enabling SAR robots to autonomously search unknown rough environments [15]. Several wheeled robots have been developed for terrestrial mobility, capable of navigating over rubble and uneven terrain in disaster zones [26, 40]. To further enhance mobility in unstructured or complex environments, researchers have also designed bio-inspired robots that can jump [4], swim [56], or slither [69], drawing from the locomotion strategies of animals. More recently, quadruped robots (such as Boston Dynamics' SPOT [10]) have gained popularity in disaster response due to their agility, stability, and ability to traverse cluttered and uneven terrain.

Competitions like the DARPA Robotics Challenge [21], DARPA SubT [22], and ELROB [23] have helped speed up progress by testing robots in realistic disaster situations with tough mobility and autonomy tasks. To support large-scale mapping, LAMP 2.0 integrates a modular front-end and back-end pipeline capable of handling odometry drift and maintaining consistent global maps [34]. CompSLAM enhances localization by fusing multiple sensor modalities for redundancy and enabling real-time pose estimation with map sharing across robot teams [34]. High-level autonomy frameworks, such as NeBula, combine SLAM with belief-space planning and risk-aware

[1] Team Chiron is representing Carnegie Mellon University (CMU) at the DARPA Triage Challenge.

trajectory selection across heterogeneous robot platforms [1]. Integrated systems developed by CTU-CRAS demonstrate how SLAM and exploration planning can be coupled for autonomous victim search in constrained and communication-limited environments [55]. In addition to algorithmic advancements, large-scale datasets such as TartanGround [51] are driving progress in learning-based perception and autonomy by providing rich multimodal data collected across diverse simulated environments. These contributions collectively advance the state of autonomous navigation for search and rescue operations in unknown terrains.

Human Health Analysis: There has been growing interest in non-contact methods for assessing human health, particularly in estimating vital signs and detecting injuries. For instance, Eulerian Video Magnification (EVM) has been used to estimate heart rate by applying spatial and temporal filtering to RGB videos to amplify subtle intensity variations on the face caused by blood flow. Extensions of this method have also been applied to enhance subtle chest movements associated with respiration [5, 37]. In addition to RGB sensing, other modalities have been explored for vital sign recognition. Thermal cameras, for example, utilize temperature fluctuations to estimate respiration or blood flow [3, 71], while multispectral imaging captures physiological signals across various wavelengths to better isolate vascular activity [35, 66]. Beyond vital sign monitoring, deep learning approaches have been employed to automatically localize and classify visible wounds and blood using image-based inputs [14, 30, 52]. However, these systems are often evaluated in controlled or constrained environments and are not yet robust or generalizable enough for deployment in complex, real-world disaster scenarios.

Despite the rapid progress across the individual domains of robotic mobility and physiological sensing, most existing systems address only isolated components of the triage problem. In this thesis, we aim to contribute towards building a unified robotic system that brings these capabilities together. By combining physiological inference and autonomous navigation, our work supports the broader goal of deploying mobile robots that can perform casualty assessment in real-world disaster scenarios.

1.3 Contribution

This thesis contributes to the development of deployable algorithms within a robotic system focused on autonomous triage in disaster response scenarios. The work in this thesis is a subset of a larger effort by Team Chiron to help realise that goal.

The key contributions of this thesis include:

- **Physiological Analysis**:
  - Development of a lightweight, monocular RGB-based algorithm for estimating respiration rate using optical flow and signal processing.
  - Geometry-based 3D body pose estimation for amputation detection, designed to run efficiently within the robotic system.
- **Robot Navigation:** Design of a simple autonomous navigation strategy that allows the robot to move around a casualty and capture diverse viewpoints for physiological analysis.
- **System Integration and Deployment:** Discussion about integrating the above algorithms in an end-to-end robotic system that supports both autonomous navigation and physiological analysis during real-time field deployment.

These contributions are not standalone solutions, but rather functional components designed to advance the larger goal of enabling autonomous robots to assess and prioritize casualties in the field. The focus has been on robustness, deployability, and task relevance, even if it comes at the cost of elegance or generality.

Outline

This thesis is structured to follow the chronological development of Team Chiron's participation in the DARPA Triage Challenge, which is a three-year effort to build autonomous systems capable of performing field triage. At the time of writing, Team Chiron is getting ready for the second year of the challenge. This thesis focuses on the work done during the first year and the developments made in preparation for the second year.

We begin by presenting physiological algorithms to estimate respiration rate and detect amputations, followed by the integration details to ensure reliable functioning of algorithms in the the Year-1 competition. Next, we discuss the multi-view inspection planning module, which enables the robot to autonomously navigate to multiple viewpoints around the casualty. This module is discussed both in terms of algorithmic design and its integration within the overall Year-2 system.

Chapter 2

Physiological Algorithms

2.1 Introduction

In a mass casualty scenario, understanding a victim's physiological condition is critical in determining the urgency and type of care required. There are several triage systems implemented across the world, including but not limited to START (simple triage and rapid treatment), SALT (sort, assess, life-saving interventions, treatment/triage), SAVE (Secondary assessment of victim endpoint) [7, 8, 62]. These systems rely on a responder's ability to quickly evaluate vital signs, most notably respiratory rate, circulation, and alertness, to determine the severity of injury and allocate medical attention accordingly [19]. However, in complex or hazardous environments, human-led triage may be infeasible, slow, or unsafe. This motivates the integration of physiological assessment capabilities into robotic systems, enabling fast non-contact evaluation of casualties at scale [61].

The DARPA Triage Challenge (DTC) pushes the development of such robotic systems that can perform fast, accurate, and scalable triage. At the core of the challenge is developing techniques to detect physiological signs that are key to analyzing the health of the casualty. To evaluate performance, DTC assigns scores based on the system's ability to correctly identify conditions such as severe hemorrhage, respiratory distress, abnormal vital signs, trauma to different body regions, and levels of alertness. In this thesis, we focus on two essential components of this evaluation - **respiration rate estimation** and **amputation detection**.

To analyze respiration rate, we present a light-weight optical flow-based method that captures subtle chest movements of casualties using video data. For amputation detection, we present a geometry-based approach to recover a 3D pose of the casualty, and identify missing limbs by analyzing inconsistencies in the recovered skeleton. Subsequently, we discuss the integration of these algorithms in our system, and relevant tools designed to assist robot operators in finding good viewpoints for the algorithms. It is important to note that the goal in developing these algorithms is not to find the most elegant solution, but rather to create practical methods that work reliably within the constraints of our system.

2.2 System and Hardware Overview - Year 1

This subsection provides an overview of the system and hardware components used in the Year 1 competition. While the system design and hardware integration were carried out by other members of Team Chiron and fall outside the scope of this thesis, they are briefly described here to provide context for the platform on which the presented algorithms were deployed.

A high-level Year 1 system overview is presented in Fig.2.1. The robot operator manually navigates the robot and initiates the behavior tree when next to a casualty. The behavior tree then triggers the physiological algorithms in a predefined sequence. These algorithms use input from the onboard sensors. Simultaneously, an AprilTag detection module identifies the tag placed next to the casualty. The outputs from each algorithm are passed to a result aggregator, which compiles the findings into a unified casualty assessment report and associates it with the corresponding AprilTag ID. This report is then sent to the DARPA server for evaluation.

Figure 2.2 shows the robot fleet and sensor payload used in Year 1. The ground robot fleet consists of two RC cars, a Boston Dynamics Spot, and a motorized wheelchair. Each ground robot is equipped with an identical payload. This payload includes a radar, thermal camera, microphone, and an iPhone mounted on a gimbal, as well as a radio for communication and a speaker for playing pre-recorded voice scripts. All computations are run onboard using a single NVIDIA Jetson Orin (64 GB). Due to the limited onboard compute, our algorithms are carefully designed to be lightweight while still providing reliable physiological assessments.

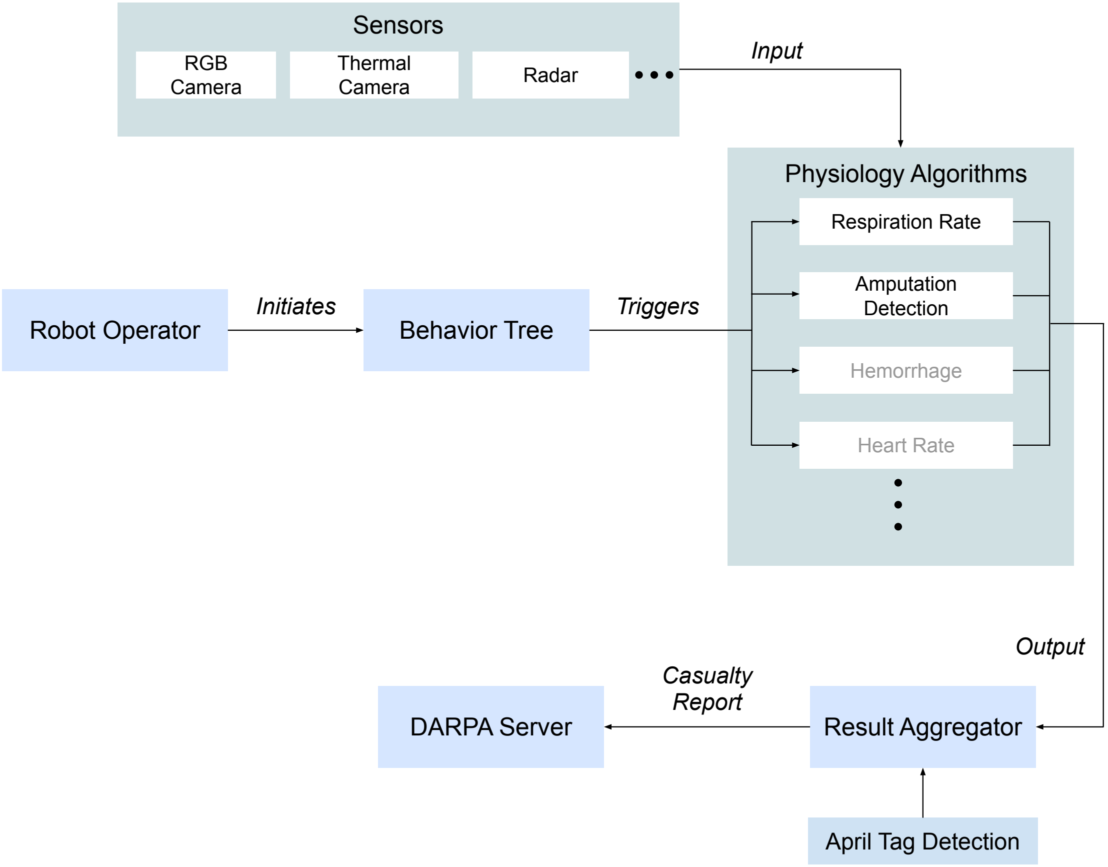


Figure 2.1: High-level system overview (Year 1)

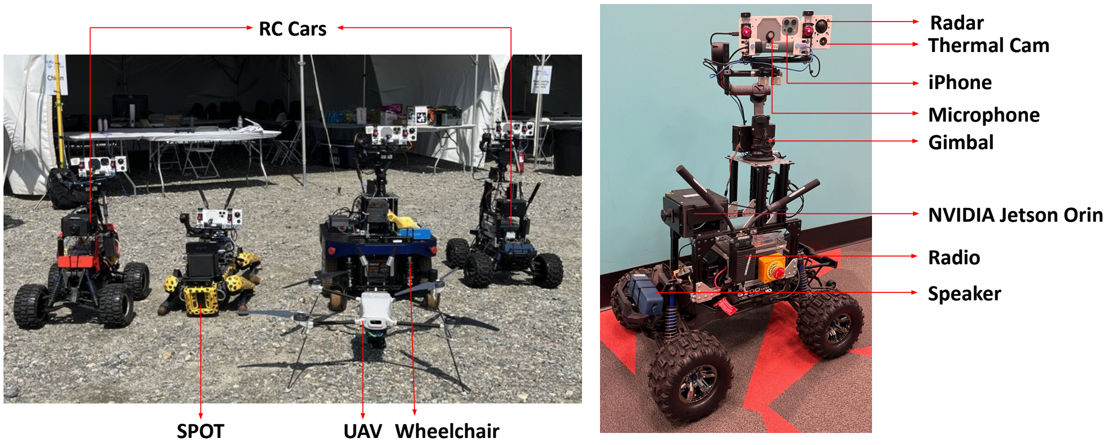


Figure 2.2: Robot fleet and payload description (Year 1)

2.3 Respiration Rate Analysis

Respiration rate is a vital physiological signal which can indicate the presence of respiratory distress, fatigue, or underlying medical conditions. In the DARPA Triage Challenge, the estimate of respiration rate is a critical component of the casualty assessment. The challenge requires systems to estimate a casualty's breathing rate within ±3 breaths per minute (BPM) of the ground truth, which is measured in real time using medically validated sensors. In our setup, this estimation must be performed using non-contact sensors mounted on a mobile robot positioned at least a meter away from the casualty. For the Year 1 competition, we developed a lightweight respiration rate estimation algorithm based on optical flow, using the RGB video stream captured by the iPhone camera mounted on the robot. During deployment, human operators manually maneuvered the robot around the casualty and utilized the camera's zoom functionality to focus on the upper torso region, where breathing motion is most prominent.

Related Work

While traditional methods for estimating respiration rate rely on contact-based sensors [18, 39, 41, 47], non-contact estimation is essential for enabling rapid and scalable casualty assessment in disaster and mass-casualty scenarios. There have been works that estimate breathing rate by analyzing nasal air flow. [54] estimates breathing by using a microphone to listen to airflow through the nose, while several others use a thermal camera to detect temperature changes caused by warm air being exhaled [3, 45]. However, these methods require precise hardware and have mostly been tested in controlled laboratory settings, making them less suitable for deployment in real-world disaster scenarios. There are also methods based on remote photoplethysmography (rPPG), where subtle changes in skin color caused by blood volume fluctuations are used to estimate respiration rate [2, 63, 64]. However, these approaches are highly sensitive to factors such as skin tone, lighting conditions, and camera angle. Instead, we draw inspiration from optical flow-based methods that rely only on standard RGB cameras to capture motion. These approaches do not require specialized hardware and are more robust to varying conditions, as long as the breathing motion is visually detectable [38, 57]. Our method is closely aligned with the objectives of [16], who address key challenges in optical flow-based respiration tracking, such as weak per-pixel motion, opposing movements across the chest, and noise in low-texture regions. While similar in objective, we take a different approach that avoids the need for a calibration phase. Instead of calculating a motion pattern from the scene, our method processes the motion signals directly in a way that allows for faster and more flexible respiration rate estimation in dynamic field conditions. Additionally, we explore how our method can be applied to detect arrhythmic breathing and propose a potential solution for reliably estimating respiration even when the person is in motion.

Approach

We estimate RR by measuring the periodic movement of the chest or abdomen region caused by breathing. This approach is built on the assumption that the input video is zoomed in, such that any motion is likely due to that of breathing.

Our approach can be divided into three stages. 1) Computing dense optical flow to capture frame-to-frame chest motion, 2) Performing temporal analysis on each pixel's optical flow output, 3) creating a global aggregation method to identify relevant pixels and extracting the most frequently occurring breathing frequency among the selected pixels. Each of these stages is explained in detail below.

Stage 1: Computing Optical flow

We use the Gunnar Farneback technique [24] to capture the motion of the surface between two consecutive frames of the sequence. For each pixel, the intensity in a small, Gaussian-weighted neighborhood is approximated by a quadratic surface.

$$I_t(\mathbf{x}) \;\approx\; \mathbf{x}^{\mathsf{T}}\mathbf{A}\mathbf{x} + \mathbf{b}_t^{\mathsf{T}}\mathbf{x} + c_t,$$

where $\mathbf{x} = (x, y)^{\mathsf{T}}$ and $\mathbf{A}$, $\mathbf{b}_t$, c_t are obtained via weighted least squares. Assuming that the next frame is a pure translation by $\mathbf{d} = (u, v)^{\mathsf{T}}$, equating the two quadratic models yields the closed-form displacement

$$\mathbf{d} \;=\; -\tfrac{1}{2}\,\mathbf{A}^{-1}\big(\mathbf{b}_{t+1} - \mathbf{b}_t\big).$$

Applying this fit at every pixel and refining the result through a coarse-to-fine Gaussian pyramid produces a dense, sub-pixel flow field. This field can capture the millimetre-scale thoraco-abdominal motion required for our respiration analysis.

After running the Farneback algorithm, we obtain the horizontal and vertical flow components for every pixel (x, y) as

$$\mathbf{d}(x,y,t) = \begin{bmatrix} u(x,y,t) \\ v(x,y,t) \end{bmatrix} \qquad (u = \tfrac{\Delta x}{\Delta t},\; v = \tfrac{\Delta y}{\Delta t})$$

For our subsequent processing, instead of using the horizontal and vertical flow separately, we represent each pixel's motion by the angle signal:

$$\theta(x,y,t) \;=\; \operatorname{atan2}\!\big(v(x,y,t),\, u(x,y,t)\big)$$

The angle signal depends only on the direction of the motion, not the amplitude. It helps to measure the expansion-contraction direction, even when the amplitude might vary due to body build, clothing, camera distance etc. The angle signal, thus, remains stable even when u or v shrink or become very large. Fig. 2.3 shows the dx, dy, and angle signal extracted from a pixel on the chest of a person breathing rhythmically.

Stage 2: Temporal analysis of Optical Flow signals

Let $\theta[n] \equiv \theta(x, y, n)$ denote the angle signal of a fixed pixel sampled at a constant frame rate f_s (Hz), with $n = 0, 1, \ldots, N-1$. The *discrete Fourier transform* (DFT) of this sequence is

$$X[k] \;=\; \sum_{n=0}^{N-1} \theta[n]\, e^{-i\frac{2\pi}{N}kn}, \qquad k = 0, 1, \ldots, N-1.$$

The corresponding frequency for bin k is

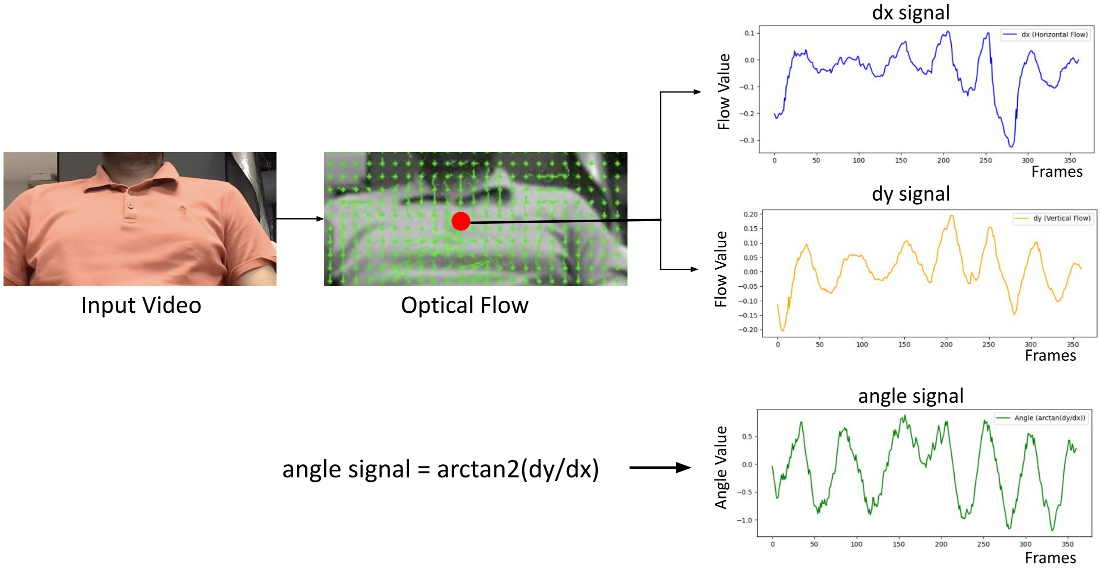


Figure 2.3: RR Estimation Step 1: Extracting angle signal using optical flow

$$f_k \;=\; \frac{k\,f_s}{N}, \quad 0 \le f_k < f_s.$$

Magnitude spectrum. The energy carried by each frequency component is captured by the magnitude

$$|X[k]| \;=\; \sqrt{\bigl(\mathrm{Re}\{X[k]\}\bigr)^2 + \bigl(\mathrm{Im}\{X[k]\}\bigr)^2}.$$

Because the DFT of a real sequence is conjugate-symmetric, we retain only the *positive* frequencies ($1 \le k \le \frac{N}{2}$).

Dominant frequency and amplitude. The breathing rate prediction for a pixel then becomes the strongest periodic component of $\theta[n]$. We therefore locate

$$k^{\star} \;=\; \arg\max |X[k]|$$

The *dominant frequency* and its *magnitude* are then

$$f_{\mathrm{dom}} = f_{k^{\star}}, \qquad A_{\mathrm{dom}} = |X[k^{\star}]|.$$

For convenience we convert frequency to breaths per minute (BPM):

$$\mathrm{RR} = 60\, f_{\mathrm{dom}}\ (\mathrm{BPM}).$$

Fig. 2.4 encapsulates the step 2, where optical flow angle signal of each pixel undergoes temporal processing, and the corresponding dominant frequency and the magnitude corresponding to that frequency is then recorded.

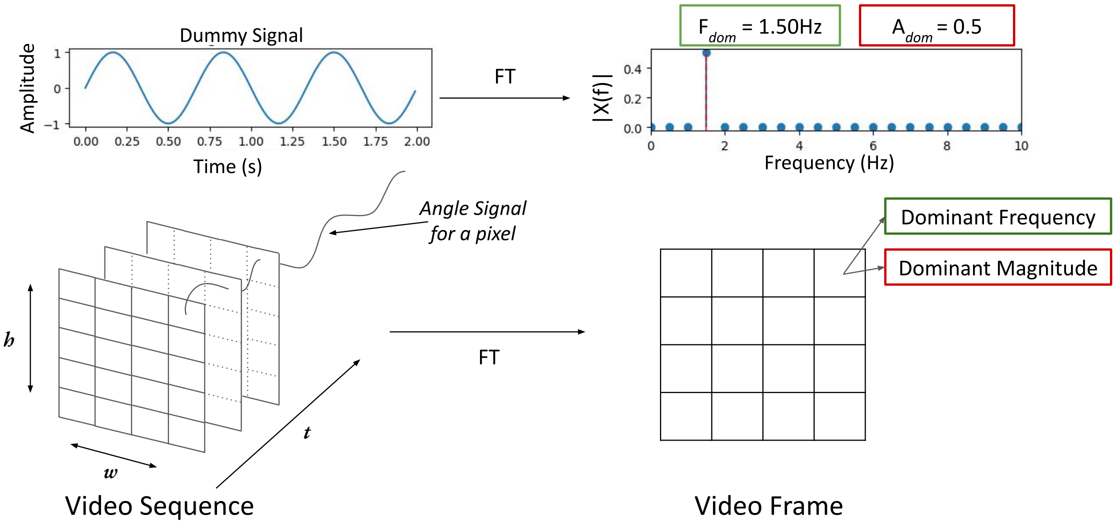


Figure 2.4: RR estimation Step 2: Each pixel's optical flow angle signal is processed using a Fourier Transform to extract its dominant frequency and corresponding dominant magnitude.

Stage 3: Global aggregation for final RR prediction

After computing the dominant frequency and corresponding amplitude for each pixel, we filter out irrelevant pixels through a two-stage masking process. First, we discard pixels whose dominant frequency falls outside a physiologically plausible respiration rate range (e.g., 5–60 BPM). Second, we treat the magnitude of the dominant frequency as a confidence measure, since all pixels share the same input, sampling rate, and signal length. Based on this confidence, we retain only the top percentile of pixels with dominant magnitude more than T_p for further analysis.

$$\mathrm{Mask}(x,y) = \underbrace{\left[\mathrm{RR}_{\mathrm{min}} \le f_{\mathrm{dom}}(x,y) \le \mathrm{RR}_{\mathrm{max}}\right]}_{\text{band-pass constraint}} \wedge \underbrace{\left[A_{\mathrm{dom}}(x,y) \ge T_p\right]}_{\text{magnitude (percentile) gate}}$$

Once the spatial mask has isolated pixels whose dominant frequency lies in the respiration band and whose spectral magnitude exceeds the percentile threshold, we aggregate their estimates with a simple mode-of-histogram vote. This final value becomes the RR estimate for that video. This step 3 is summarized in Fig. 2.5. In 2.5a, the input is a video of a person breathing, and the masking process correctly highlights the chest region. In Fig. 2.5b, the input is a video of a breathing manikin. The resulting mask is similarly appropriate. A histogram-based voting on the selected pixels yields a predicted respiration rate of 26.78 BPM, which is close to the ground truth of 28 BPM. The overall algorithm is explained in Alg. 1.

Result and Discussion

Figure 2.6 shows examples from the Year 1 DARPA Triage Challenge where the respiration rate (RR) was correctly predicted. In both cases, the algorithm successfully identifies pixels exhibiting breathing-related motion using the temporal processing and masking techniques described in the previous section. From these selected pixels, the final RR estimate is obtained using a mode-of-histogram voting method, and the predicted value remains within 3 BPM of the ground truth, satisfying DARPA's accuracy requirement.

However, a limitation of the current pipeline appears when the subject makes any large-scale movement. Because the algorithm treats every periodic displacement the same way, it cannot tell whether a strong signal comes from breathing or from something else, such as a sudden jerk, a hand gesture, or the repetitive back-and-forth of a casualty due to distress. As a result, the temporal analysis respiratory rate can drift far from the ground-truth value whenever vigorous or repetitive body motion is present. Next, we propose a possible solution for separating genuine respiratory motion from these interfering movements.

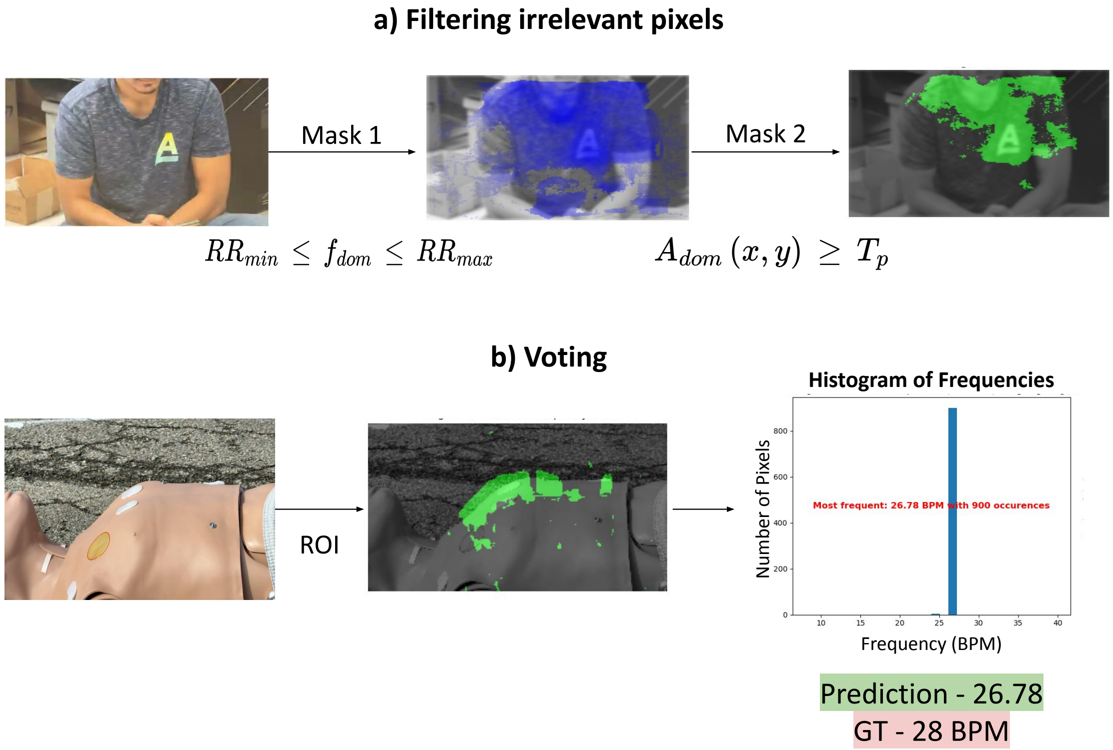


Figure 2.5: RR estimation Step 3: a) Unwanted pixels are first removed based on a dominant frequency and confidence threshold. b) A mode-of-histogram voting is then applied across the remaining pixels to estimate the final prediction.

Algorithm 1 Respiration Rate estimation from an RGB video

```
Require: RGB video $\mathcal{V} = \{I_t\}_{t=0}^{T-1}$ sampled at $f_s$ fps, breathing band $[\mathrm{RR}_{\mathrm{min}}, \mathrm{RR}_{\mathrm{max}}]$,
    magnitude percentile $p$, number of histogram bins $B$

    Stage 1: Optical flow
for $t \leftarrow 1$ to $T - 1$ do
    Pre-process $I_{t-1}$ and $I_t$ ( resize, grayscale)
    $\mathbf{d}_t \leftarrow \textsc{FarnebäckFlow}(I_{t-1}, I_t)$                ▷ $\mathbf{d}_t(x, y) = [u, v]^\top$
    $\theta_t(x, y) \leftarrow \mathrm{atan2}\big(v(x, y), u(x, y)\big)$
end for
    Stage 2: Per-pixel temporal analysis
for all pixels $(x, y)$ do
    $\{\theta_t(x, y)\}_{t=1}^{T-1} \leftarrow$ moving-average filter
    $\{F_k, A_k\} \leftarrow \mathrm{FFT}(\theta_t(x, y))$
    $(f_{\mathrm{dom}}, A_{\mathrm{dom}}) \leftarrow \arg\max_k A_k$          ▷ keep dominant frequency & magnitude
end for
    Stage 3: Global Aggregation
Compute percentile threshold $T_p = \mathrm{percentile}_p\big(\{A_{\mathrm{dom}}\}\big)$
for all pixels $(x, y)$ do
    $M(x, y) \leftarrow \Big[\mathrm{RR}_{\mathrm{min}} \leq 60 f_{\mathrm{dom}}(x, y) \leq \mathrm{RR}_{\mathrm{max}}\Big] \wedge \big[A_{\mathrm{dom}}(x, y) \geq T_p\big]$
end for
$\mathcal{S} \leftarrow \{60 f_{\mathrm{dom}}(x, y) \mid M(x, y) = 1\}$                    ▷ BPM voting
$\{C_i\}_{i=0}^{B-1} \leftarrow \textsc{Histogram}(\mathcal{S}, B)$
$i^\star \leftarrow \arg\max_i C_i$
$\widehat{\mathrm{RR}} \leftarrow \frac{1}{2}\big(b_{i^\star,\mathrm{low}} + b_{i^\star,\mathrm{high}}\big)$                ▷ centre of winning bin
return $\widehat{\mathrm{RR}}$
```

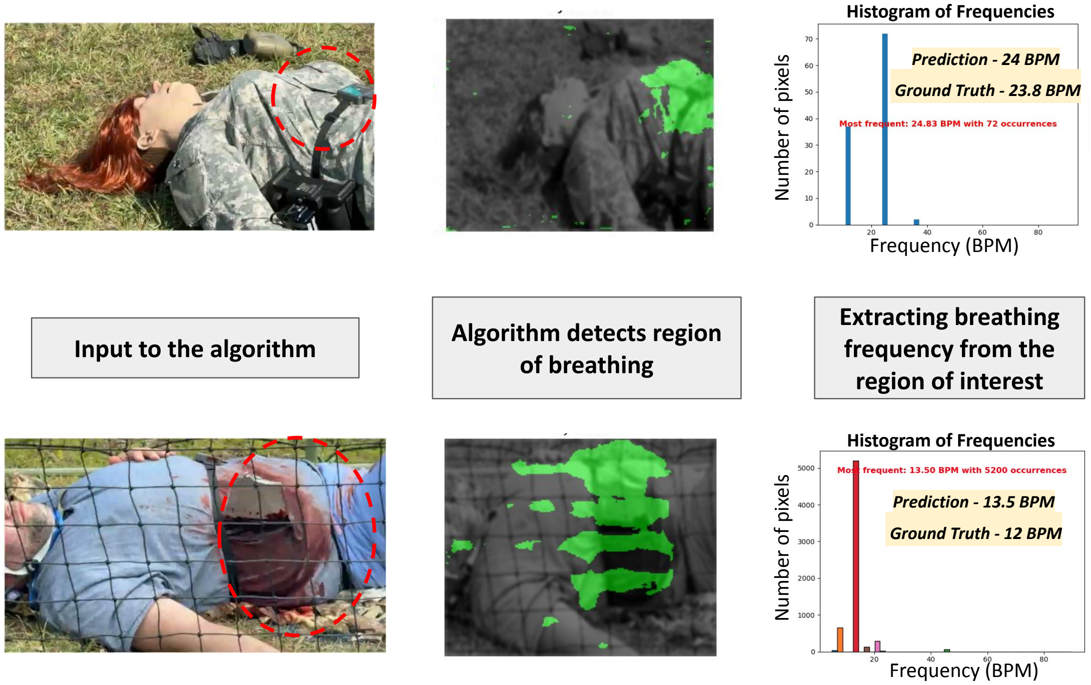


Figure 2.6: Samples of successful respiration rate analysis from DARPA Triage Challenge Year 1.

Analysing motion at varying scale

As shown in the previous section, temporal processing of each pixel is not enough to extract the right respiration rate. There is a need to extract the part of the signal, which corresponds to subtle breathing motion, and avoid the section where the human showed large motion (body tilts, shifts etc). In this section, we demonstrate that motions at varying scale can be segregated through a combination of both spatial and temporal processing of standard monocular sequences.

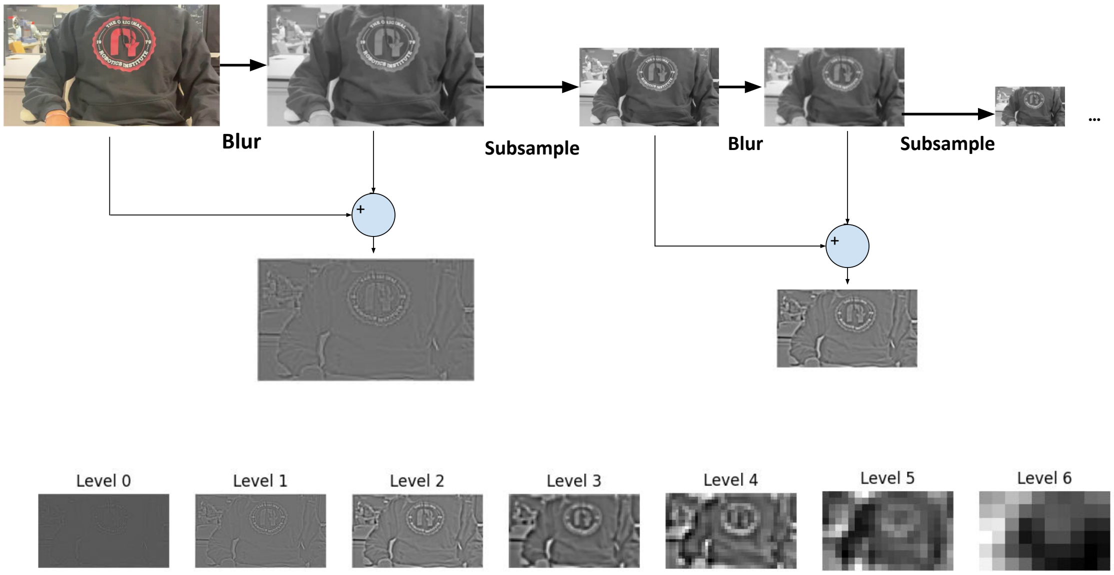


Figure 2.7: Constructing a Laplacian pyramid from an input video

We start by decomposing the input video into different spatial frequency bands by generating a full Laplacian pyramid [12] (Fig. 2.7). Each level is the difference between two successive Gaussian blurs, i.e., a narrow spatial band. High-frequency content is available in the early layers of the pyramid, whereas only the low-frequency content is isolated at the later layers of the pyramid (generated after repeated blur and downsampling). This allows us to capture the subtle breathing motion essential for a correct estimation in the early levels, at the same time, identify the motion that happens on a large scale in the later levels of the pyramid.

We demonstrate an example in Fig. 2.8. We record a video of a subject breathing normally, with minimal movement except toward the end of the sequence. We decompose the video sequence into a Laplacian pyramid, and perform optical flow on each level of the pyramid.

Subsequently, we observe the optical flow signal at a pixel from the chest. The initial levels of the pyramid show a sinusoid that corresponds to the subtle breathing-only motion, and then a large noise in the later section of the sequence. A naive temporal processing of just this signal can result in an estimate that drifts away from the correct respiration due to the noise. On the other hand, the optical flow of the same point at Level 6 (which loosely correspond to the entire chest region), remains relatively inactive during subtle breathing and responds primarily during large movements.

This demonstrates the ability to separate subtle breathing motion from larger body movements. By identifying periods of large motion using higher levels of the pyramid (e.g., Level 6), and removing the corresponding segments from the fine-resolution signal at Level 1, we can suppress non-respiratory interference. Applying the temporal analysis and global aggregation steps on this cleaned signal leads to more accurate respiration rate estimation. Figure 2.9 illustrates this process. Interestingly, in the example shown, the Fourier Transform of the original Level 1 signal would still yield a similar dominant frequency as the segmented signal, since the person's body motion happened to be periodic and aligned with the breathing frequency. However, the key insight lies in the algorithm's ability to distinguish between large and subtle motions. This separation becomes particularly valuable in cases of respiratory distress, where the person may rock back and forth. By isolating and removing large, non-respiratory movements from the signal, the system can focus on subtle chest motion to produce a more accurate respiration rate estimate.

It is interesting to note that if the goal of the spatial processing is just to identify large motion by pooling multiple pixels, it can be done using just a low-pass filter of the frames. A Laplacian, however, helps in better band pass manipulation and image reconstruction to remove large artifacts, which is a good extension of the work for the future.

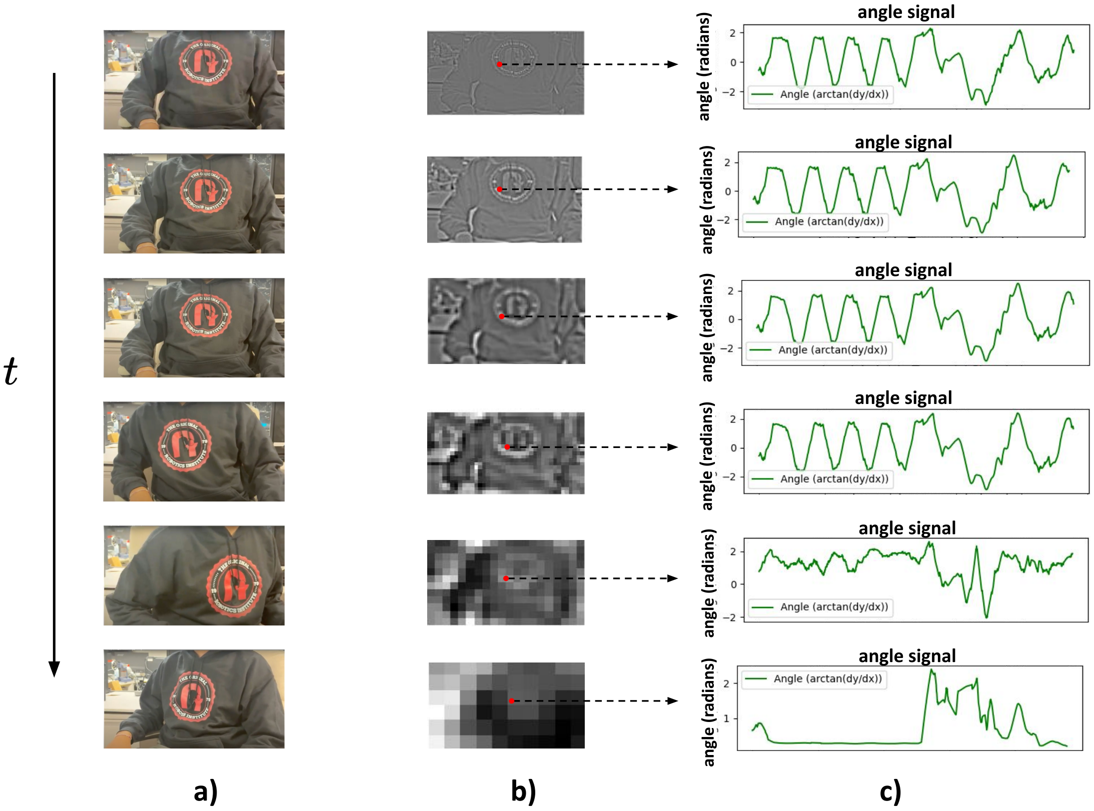


Figure 2.8: Optical flow signal at a given point across different levels of pyramid.

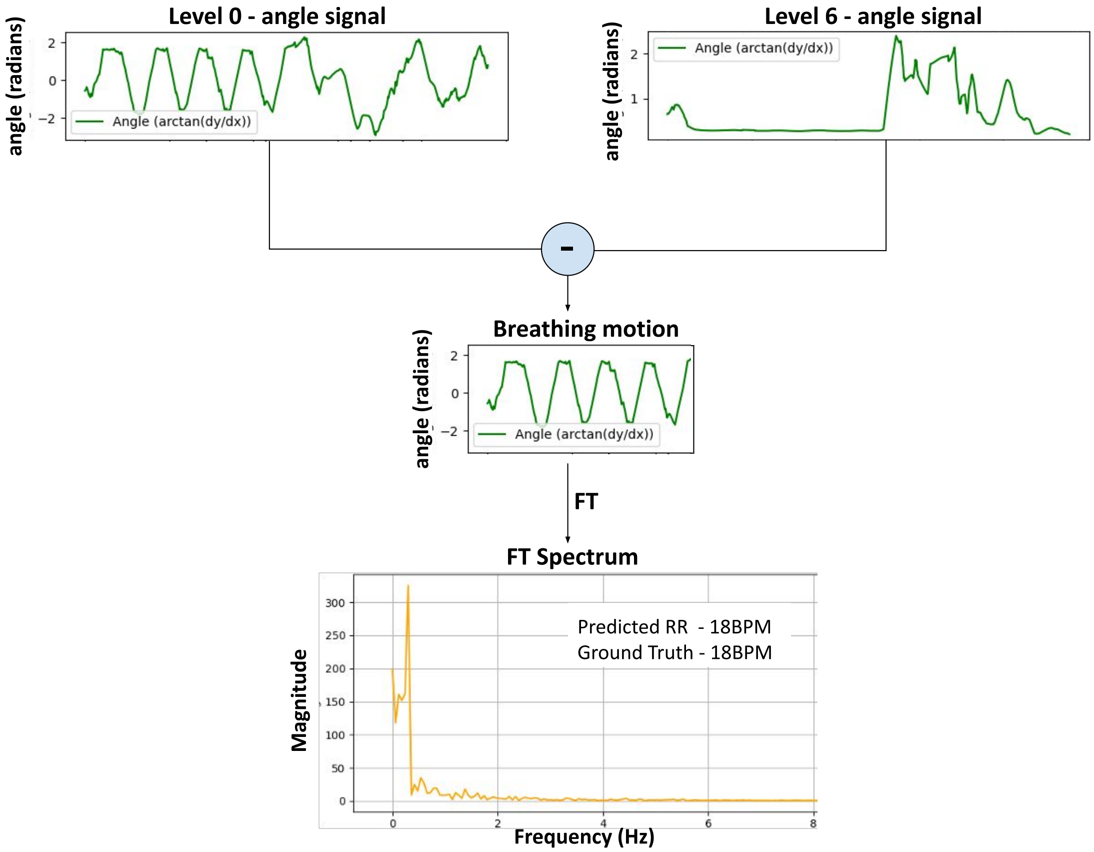


Figure 2.9: Identifying and removing part of the signal corresponding to large motion helps in more precise analysis of subtle breathing motion.

Detecting Arrhythmic breathing

Building on our respiration rate estimation method, we extend the analysis to detect arrhythmic breathing, a potential indicator of respiratory distress. Unlike normal breathing, which exhibits a consistent rhythm, arrhythmic breathing shows irregularities in the breathing frequency over time.

We apply the Short-Time Fourier Transform (STFT) to the motion signal extracted from each pixel, which allows us to capture how the dominant breathing frequency evolves over time. From the STFT output, we identify the dominant frequency at each time window by selecting the frequency with the maximum magnitude. This results in a time series of dominant frequencies for each pixel. We then compute the standard deviation of these frequencies over time to quantify temporal variability. A high standard deviation suggests that the breathing frequency is not stable and may indicate arrhythmic behavior.

To robustly estimate the presence of arrhythmic breathing, we apply a threshold on the per-pixel variability map and aggregate the results over the pixels identified as relevant to breathing (using the final breathing mask). If a significant portion of the masked pixels exceeds the variability threshold, the sample is flagged as exhibiting respiratory distress. Fig. 2.10 shows the breathing signal from a human casualty exhibiting arrhythmic breathing. The dominant frequency, extracted over time using STFT, reveals high variability, indicating an irregular breathing pattern.

In this section, we presented a simple optical flow–based method to estimate respiration rate (RR) from RGB videos. The approach is lightweight and can be easily implemented on robotic platforms for real-time monitoring. We demonstrated its effectiveness in estimating RR for stationary individuals, including successful tests in DARPA Triage scenarios. However, we also identified a key limitation—when the person moves significantly, the algorithm may struggle to separate breathing motion from other body movements. To address this, we proposed analyzing motion at multiple spatial scales to better isolate subtle breathing signals. This direction also opens up possibilities for future work, such as reconstructing videos with amplified breathing motion or with large movements suppressed.

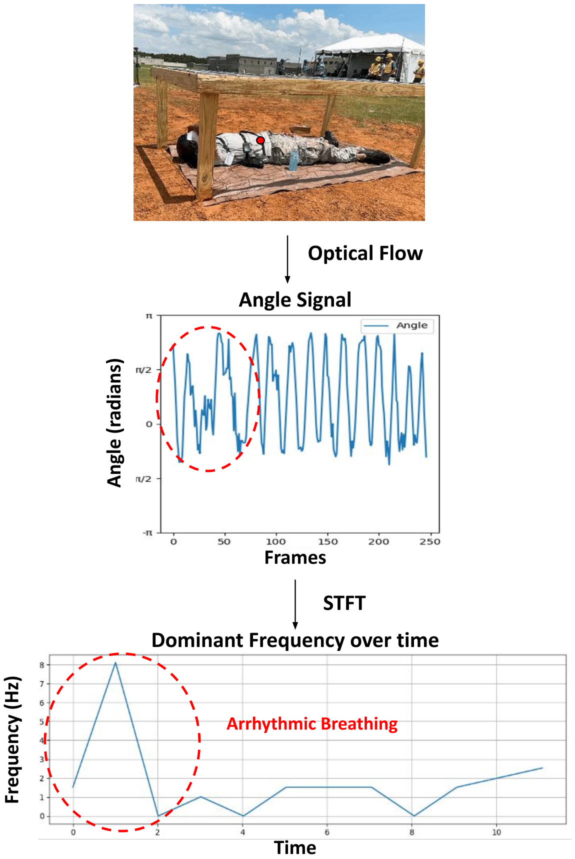


Figure 2.10: Arrhythmic breathing detection using Short-Time Fourier Transform (STFT). Dominant frequency is extracted over time from the optical flow signal (in this example, the signal is from the pixel in red). A large variation in the frequency values indicates irregular breathing.

2.4 Amputation Detection

In a mass casualty scenario, triage protocols classify victims into categories based on the urgency of medical intervention required. The ”immediate” category includes individuals who need life-saving care within minutes to two hours to prevent death [70]. Traumatic amputations fall into this category due to the high risk of severe hemorrhage. Rapid detection of such injuries is critical to initiate timely hemorrhage control and prevent fatal outcomes [48, 67]. Due to this, amputation detection is a key evaluation criterion in the DARPA Triage Challenge, where participating teams are required to identify and report upper or lower limb amputations in casualties as part of their triage systems.

A natural first step in analyzing the human body pose is to apply an off-the-shelf, deep learning-based 2D keypoint detector to RGB images of the casualty. We highlight two major observations. As shown in Fig. 2.11, the keypoint detection model on an amputated casualty still predicts all the possible keypoints. This is expected as most keypoint detectors are trained exclusively on datasets containing fully-limbed individuals, leading the model to always localize every joint regardless of anatomical correctness. Moreover, incorrect predictions can be made with high confidence. For instance, in Fig. 2.11, the right leg is amputated below the knee, yet the model confidently predicts the right ankle keypoint near the knee. This (incorrect) prediction occurs because the model assumes that the missing segment is a case of self-occlusion, such as a bent leg.

One possible approach to detect amputations using visual data is to train a modified keypoint detection model that not only localizes joints but also classifies each as visible, (self-)occluded, or amputated. However, the scarcity of training data with amputated individuals makes this method impractical to implement.

A more viable alternative is to analyze the joint lengths of a reconstructed 3D skeleton of the casualty. We leverage a consistent behavior observed in 2D keypoint detectors—when a limb is amputated, the missing joint is often predicted with high confidence near its parent joint. Instead of discarding these inaccurate predictions, we incorporate them into a multi-view triangulation process to recover a 3D skeleton. By examining the resulting joint lengths, we can identify deviations from normal anatomical proportions, enabling the detection of amputations.

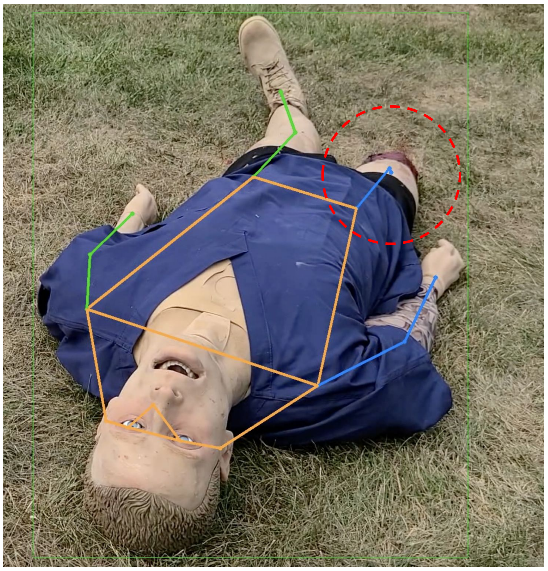


Figure 2.11: A standard 2D keypoint detector incorrectly localizes the right ankle near the right knee with high confidence, misinterpreting the amputation as a bent leg.

Related Work

Triangulation-based approaches have proven highly effective for 3D human pose estimation, particularly in multi-view settings where 2D keypoints from multiple cameras can be fused to reconstruct a coherent 3D skeleton. The structural triangulation method proposed a closed-form solution for 3D human pose estimation that embeds human bone-length priors directly into the triangulation process [17]. It ensures proportional limb lengths and results in high-fidelity reconstructions that align with expected human anatomy. Similarly, Holistic Triangulation [65] introduces a view-consistency-aware strategy that fuses multi-view 2D keypoints and applies PCA-based anatomical priors to regularize the output pose. These works highlight the importance of geometric consistency and bone structure regularization to achieve accurate and realistic 3D skeletons. There is a separate group of learnable triangulation approaches that formulate the triangulation process as a differentiable module, allowing the network to learn confidence-weighted fusion of 2D keypoints across views [6, 11, 31]. While their focus is on improving pose accuracy via learnable fusion, the method still assumes anatomically consistent limb lengths, which is beneficial for general-purpose 3D reconstruction.

In contrast to these works, our goal when recovering 3D pose is not to enforce anatomical correctness, but to detect violations of expected limb proportions. We take inspiration from the methods presented above to build a simple triangulation based strategy to recover a 3D human skeleton that intentionally leverages incorrect 2D keypoint predictions to detect amputations.

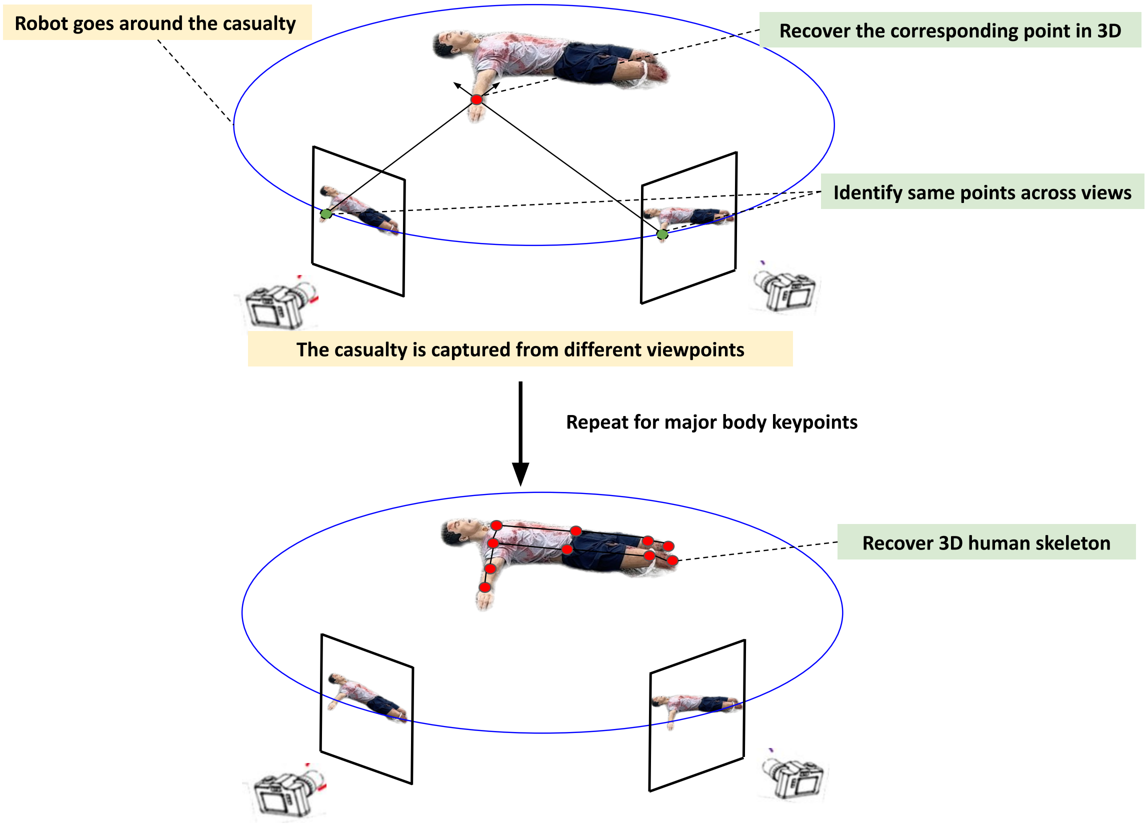

Robot goes around the casualty
Recover the corresponding point in 3D
Identify same points across views
The casualty is captured from different viewpoints
Repeat for major body keypoints
Recover 3D human skeleton

Figure 2.12: Triangulating multi-view 2D keypoints to reconstruct a 3D human skeleton, which is then analyzed for joint-length inconsistencies to detect amputations.

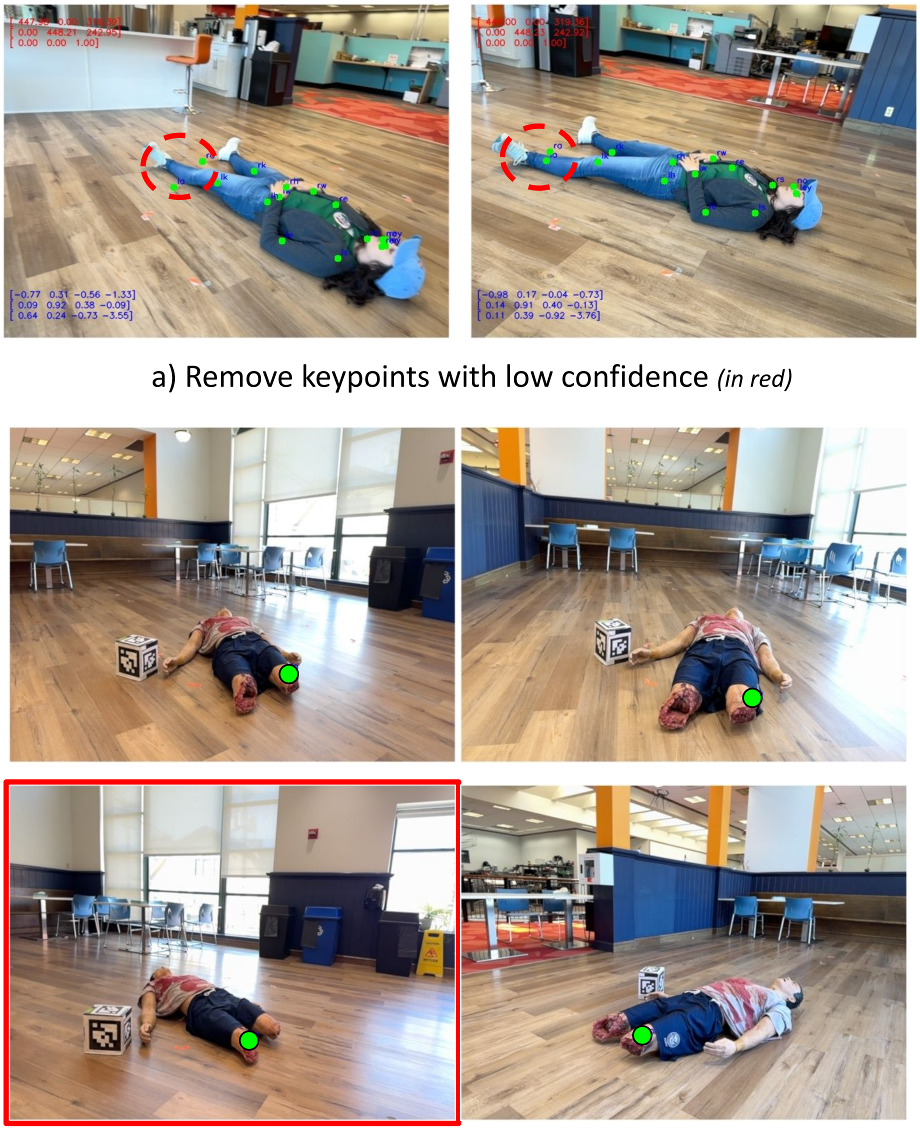
Approach

Our method for amputation detection is built around a geometry-based pipeline that reconstructs a 3D human skeleton from multi-view images (Fig. 2.12) and analyzes joint-length proportions for signs of missing limbs. The overall pipeline is composed of three stages: (1) multi-view 2D keypoint extraction and filtering, (2) 3D skeleton reconstruction via triangulation, and (3) amputation detection based on joint-length inconsistencies.

Stage 1 - Multiview 2D Keypoint extraction and filtering

While going around the casualty, the robot captures multiple views to inspect the person (Fig. 2.12). To extract 2D keypoints, we run the YOLOv8 [33] keypoint detector on all the frames. Due to a variety of reasons (camera angle, pose of the casualty, occlusion etc), some 2D keypoint detections can be erroneous. These detections should be carefully removed before the triangulation process. First, we remove the keypoints with confidence below a certain threshold. This removes poorly localized keypoints due to occlusion, or a bad camera angle (Fig. 2.13 a). Second, we apply RANSAC [25] to eliminate geometric outliers across views for each keypoint. In our case, for each joint, we first collect all keypoint predictions across views and retain the top 75% based on their confidence scores. These high-confidence keypoint observations serve as the candidate set for RANSAC. In each of the RANSAC iterations, we randomly sample 70% of these keypoint predictions and triangulate a 3D point using a DLT-based method (described below). The triangulated point is then reprojected into all candidate views, and the reprojection error is computed. Keypoint predictions with reprojection errors below the 75th percentile are considered inliers. The iteration with the most inliers and lowest average error is selected, and the corresponding inlier set is used for final triangulation. This procedure is repeated independently for each joint, allowing the method to select the most geometrically consistent 2D keypoint predictions per joint across views.

In Fig. 2.13b, we focus on the detection of the left ankle. Although the left ankle does not physically exist in this casualty, most camera views consistently predict it very close to the left knee. This behavior is expected from pretrained keypoint detectors and is useful for our purpose, as it results in a very short estimated lower leg, allowing us to detect the disproportionate segment. However, due to poor angles or occlusions, few views incorrectly predict the left ankle on the opposite side of the body, near the right leg. These outliers, if not removed, would inflate the estimated



a) Remove keypoints with low confidence *(in red)*

b) Remove outliers using RANSAC

(left ankle prediction made by the model shown in green)

Figure 2.13: Filtering erroneous 2D keypoints: (a) removes low-confidence detections, and (b) uses RANSAC to discard outlier views (in this case, an incorrectly placed left ankle prediction near the right leg) while retaining consistent predictions that help reveal disproportionate limb lengths.

joint length and lead to incorrect conclusions. RANSAC is effective in filtering such anomalies, preserving only the dominant and spatially consistent predictions across views.

Stage 2 - Triangulation

To reconstruct 3D joint locations from 2D keypoints, we perform triangulation across multiple views. This step requires two inputs: (i) consistent and geometrically reliable 2D keypoint predictions across views, which we obtain from the filtering and RANSAC process described in Step 1, and (ii) accurate intrinsic and extrinsic camera parameters for each view. In our case, we use the iPhone's built-in ARKit framework to obtain reliable camera intrinsics and extrinsics for all captured images.

We adopt the classical Direct Linear Transform (DLT) method for triangulating each joint from its corresponding 2D observations across multiple views [29]. Let $\mathbf{x}_i = [u_i, v_i, 1]^\top$ denote the homogeneous 2D coordinates of a keypoint in view i, and let $\mathbf{P}_i$ be the 3×4 projection matrix of that view (obtained from the intrinsic and extrinsic calibration). The key relationship between the 3D point $\mathbf{X} = [X, Y, Z, 1]^\top$ and its 2D projection is given by:

$$\mathbf{x}_i \sim \mathbf{P}_i \mathbf{X}$$

To eliminate the unknown scale, we use the cross product formulation:

$$\mathbf{x}_i \times (\mathbf{P}_i \mathbf{X}) = \mathbf{0}$$

This yields two linearly independent equations per view. By stacking the equations from n views, we obtain a linear system of the form:

$$\mathbf{A}\mathbf{X} = \mathbf{0}$$

where $\mathbf{A}$ is a $2n \times 4$ matrix constructed from all views observing the keypoint. We solve this homogeneous system using Singular Value Decomposition (SVD), and the solution $\mathbf{X}$ is given by the last column of $\mathbf{V}$ in the decomposition of $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$. The resulting 3D point is normalized by its homogeneous coordinate.

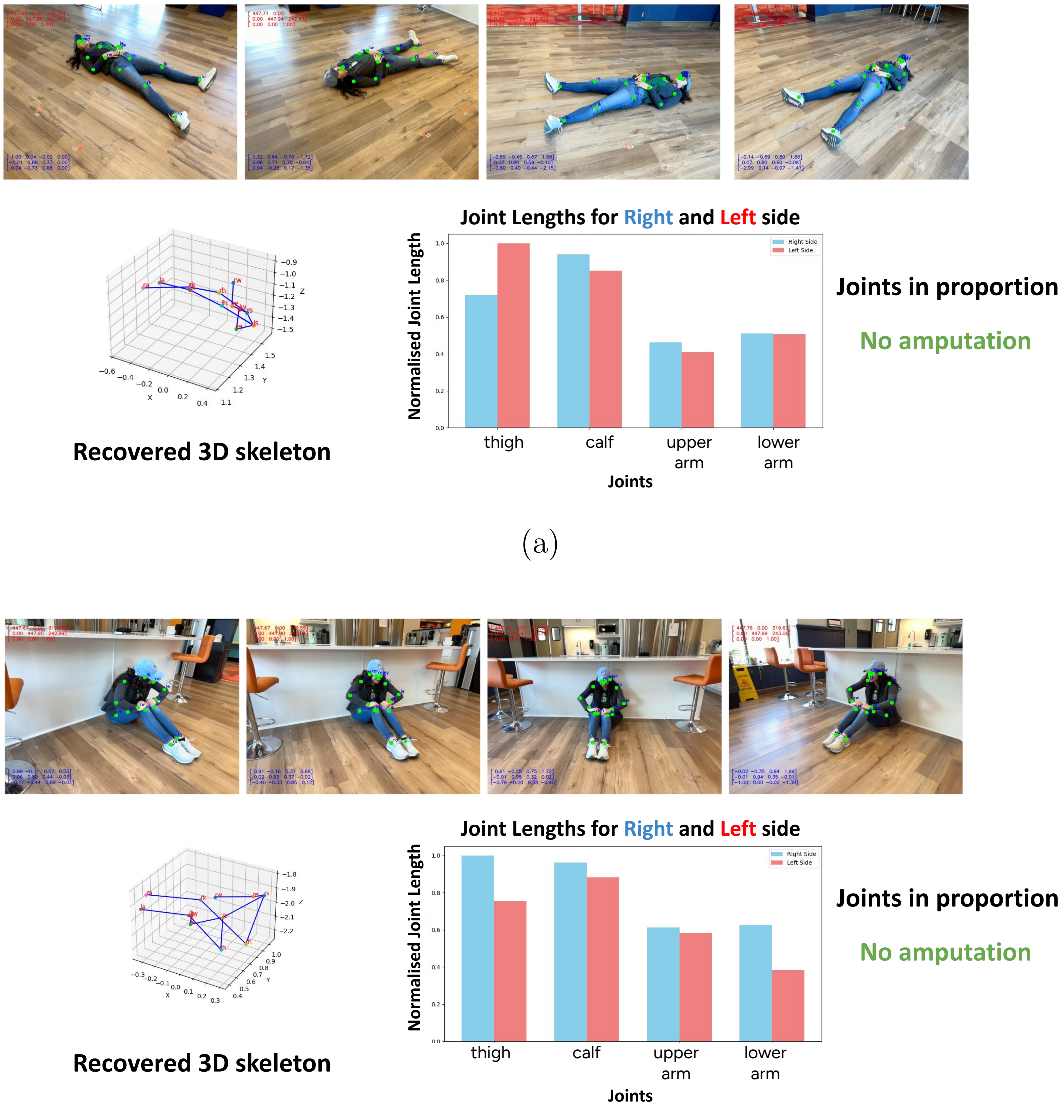
This process is repeated independently for each joint, producing a full 3D human skeleton from the filtered 2D keypoints and calibrated camera parameters.

Stage 3 - Checking Joint Length Inconsistencies

After reconstructing the 3D human skeleton through triangulation, we compute the Euclidean distances between adjacent joints to estimate the lengths of key limb segments. Once normalized, we evaluate the proportionality of each limb segment. In our analysis, we perform two types of checks to detect amputations using the normalized joint lengths. First, we evaluate intra-limb proportions by comparing the lengths of adjacent segments within a single limb. For example, in the arm, we compare the shoulder-to-elbow length with the elbow-to-wrist length. A significantly shorter lower segment (less than 40% the length of the upper segment) is flagged as a possible amputation. A similar check is applied for the leg, comparing hip-to-knee and knee-to-ankle segments. Second, we assess left–right symmetry between corresponding limb segments. If the normalized length of a segment on one side of the body is less than 50% of its counterpart on the opposite side, this asymmetry is also considered indicative of an amputation. By combining intra-limb and inter-limb proportion checks, we identify missing limb segments in both the upper and lower body.

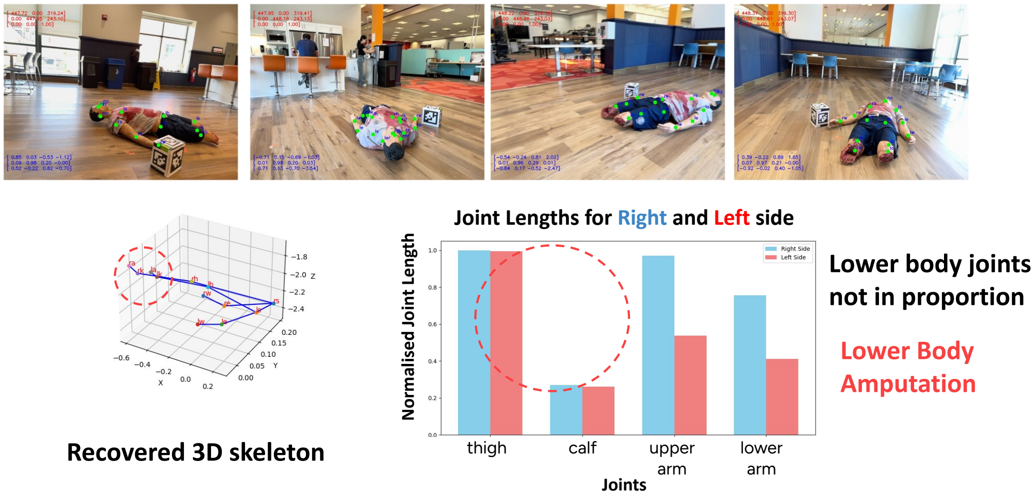
Results

We show results of our geometry-based amputation detection method in Fig. 2.13. We first extract and filter 2D keypoints across different views, recover a 3D skeleton based on those keypoints, and then analyse the joint lengths to predict amputation. The subject is lying flat with full limb visibility in Fig. 2.13a. The reconstructed 3D skeleton shows symmetric and proportionate limb lengths across both sides of the body. The bar chart confirms that corresponding joint lengths (e.g., left vs right thigh, calf, upper arm, and lower arm) are within a proportionate range. The algorithm thus concludes no amputation. In Fig. 2.13b, the subject is seated with arms and legs bent in a tripod pose. Consistent keypoints are selected across views, and triangulation yields a valid 3D skeleton. Joint length comparisons reveal balanced proportions, leading to the correct inference of no amputation. Fig. 2.13c shows a manikin with amputations in both the legs. The recovered 3D skeleton has a shorter length of knee-ankle joint in both the legs as compared to the hip-knee joint. The


Recovered 3D skeleton
Joint Lengths for Right and Left side
Normalised Joint Length
Right Side
Left Side
thigh
calf
upper arm
lower arm
Joints
Joints in proportion
No amputation
X
Y
Z

(a)

(b)



(c)

Figure 2.13: Results of our geometry-based amputation detection method. In each case, we recover the 3D skeleton of the subject using 2D keypoints across different views, and compare the joint lengths to predict amputation.

joint length analysis thus picks up disproportionate limb length and predicts lower body amputation.

Thus, we see that by intentionally leveraging the incorrect but consistent 2D keypoint predictions near parent joints, we can recover an anatomically inaccurate 3D skeletons that help reveal structural anomalies. While recent vision-language models offer powerful alternatives to detecting amputations, our robot's hardware constraints motivated us to explore solutions beyond heavy deep learning architectures. It is important to note that the effectiveness of this method is highly sensitive to viewpoint quality. In the following sections, we examine the limitations introduced by poor viewpoints and discuss strategies to improve the quality of viewpoints.

2.5 System Integration

All physiological analysis algorithms are deployed as ROS 2 nodes. These nodes communicate with the robot's sensor suite via ROS 2 topics and publish their results to designated output topics. While all nodes are launched together at system startup, they remain idle until they receive a task trigger (`working_request`) from the behavior tree, upon which the nodes start collecting data and run inference.

The robots are manually operated during deployment, requiring the operator to control both the base movement and the gimbal orientation simultaneously. This becomes particularly challenging when navigating around a casualty to capture diverse and informative viewpoints within a short time limit. To mitigate this, we integrate a set of system tools to assist the operator in maintaining optimal sensor alignment. Specifically, we build a gimbal control strategy to maintain focus on the casualty and a visualization interface for monitoring the gimbal's orientation in real time.

Gimbal Control for Hip Tracking

We aim to keep the gimbal continuously pointed at the hip keypoint of a person while the robot is manually driven around them. This allows the operator to focus solely on controlling the robot's movement, without needing to manually adjust the gimbal. We use the iPhone's `ARKit` to provide the 3-D hip position in the phone-camera frame.

Let the 3D coordinates of the hip keypoint relative to the camera be:

$$\mathbf{p}_{\text{hip}} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

Pointing angles. From $\mathbf{p}_{\text{hip}}$ we compute

$$\theta = \operatorname{atan2}(x,\, z), \qquad \phi = \operatorname{atan2}\big(y,\, \sqrt{x^2 + z^2}\big),$$

where θ is the *azimuth* (yaw error) and ϕ is the *altitude* (pitch error). If the hip lies on the optical axis we have $\theta = \phi = 0$.

PD control. To bring the errors to zero we use a proportional–derivative (PD) law for each axis:

$$u_\theta = K_{P,\theta}\,\theta \;+\; K_{D,\theta}\,\frac{\theta - \theta_{\text{prev}}}{\Delta t},$$

$$u_\phi = K_{P,\phi}\,\phi \;+\; K_{D,\phi}\,\frac{\phi - \phi_{\text{prev}}}{\Delta t},$$

where Δt is the time between updates. We observe that we obtain stable tracking with a predominantly proportional controller, choosing

$$K_{P,\theta} = K_{P,\phi} \;\approx\; 2.0, \qquad K_{D,\theta},\, K_{D,\phi} \;\ll\; K_P.$$

Only small derivative gains are required because the casualty's motion is slow, `ARKit` delivers hip updates at $\sim$30 Hz and the gimbal's own dynamics are fast enough that proportional action alone settles the error without overshoot. Fig. 2.14 shows the PD controller in action, allowing the gimbal to continuously point at the casualty's hip.

Mapping to gimbal axes. Taking roll to be zero, the desired angular-rate vector in the camera frame is

$$\mathbf{g}_s^{\text{cam}} = \begin{bmatrix} u_\phi \\ u_\theta \\ 0 \end{bmatrix}.$$

A constant rotation matrix $R_{\text{gimbal}\leftarrow\text{cam}}$ converts these rates to the gimbal's own roll–pitch–yaw convention:

$$\mathbf{g}_s^{\text{gim}} = R_{\text{gimbal}\leftarrow\text{cam}}\,\mathbf{g}_s^{\text{cam}}.$$

Command interface. The vector $\mathbf{g}_s^{\text{gim}}$ (in $\text{rad}\,\text{s}^{-1}$) is published at $\sim$ 20 Hz, closing the feedback loop that keeps the hip centred while the operator manually drives the robot base around the person.

Visualising gimbal angle

During the field runs, manual operators played a critical role in selecting appropriate viewpoints for downstream algorithms. One major challenge encountered during field operation was the inability to easily track how far the gimbal had rotated in yaw relative to the robot's front-facing default orientation. This lack of feedback often led to the gimbal wires becoming entangled, causing the gimbal to crash or lose

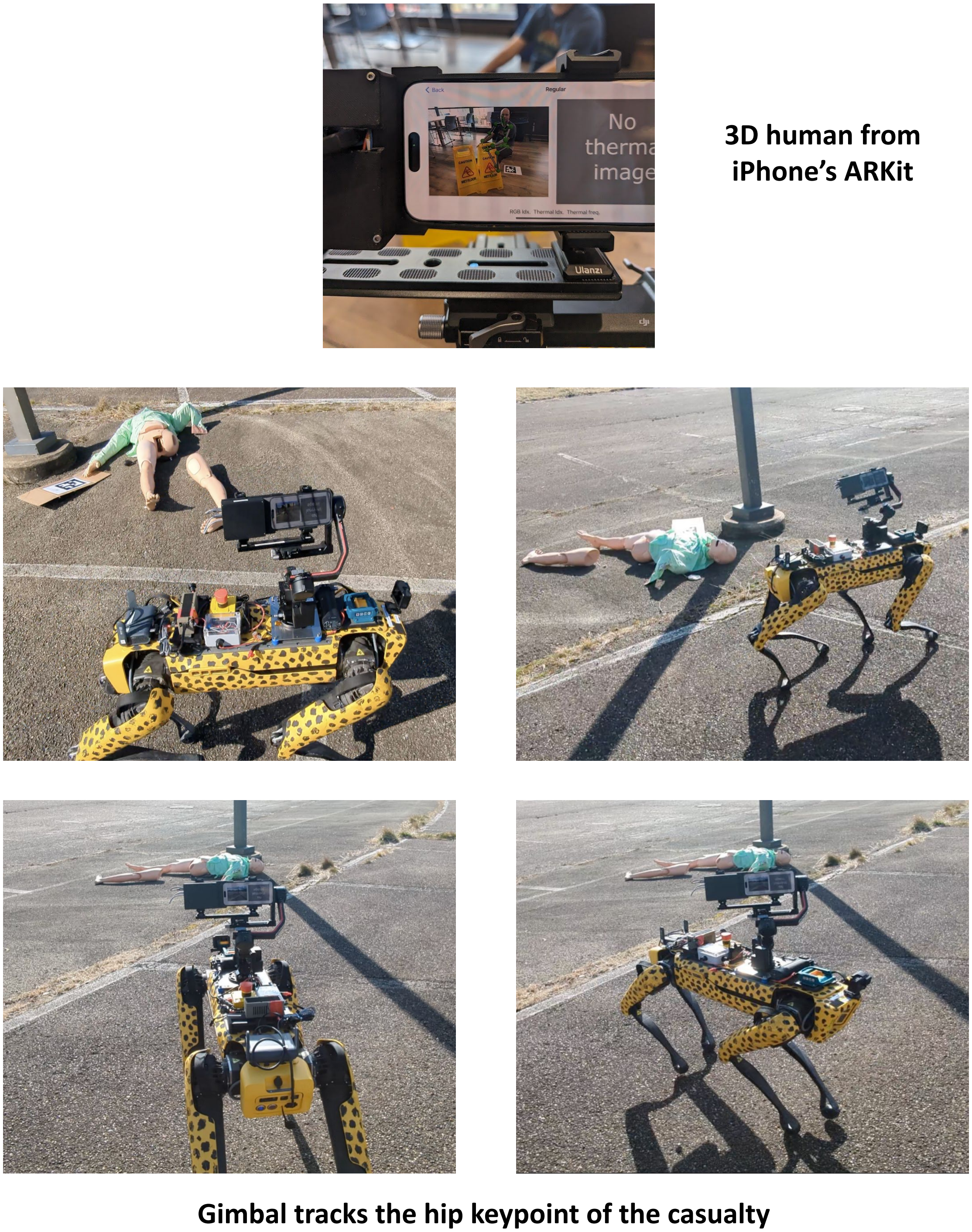

3D human from iPhone's ARKit
No therma image
Gimbal tracks the hip keypoint of the casualty

Figure 2.14: The robot is manually driven around the casualty, while the gimbal continuously maintains the hip at the center of the camera's field of view.

calibration. Additionally, excessive gimbal rotation made it difficult for the operator to maintain a consistent sense of the robot's orientation.

To address this, we developed a lightweight script to visualize the gimbal's yaw rotation relative to the robot's front, helping operators better manage orientation and prevent mechanical failure.

We therefore implemented an `rqt` plugin, `GimbalVisPlugin`, that gives real-time feedback on the gimbal's yaw angle ψ.

- **ROS 2 node.** A background node (`GimbalNode`) subscribes to the driver topic `/gimbal/curr_joint_angle` (`geometry_msgs/Vector3`). The field `z` is interpreted as the gimbal yaw $\psi(t)$ [rad], measured from the robot's frontal reference. On each message arrival the node time-stamps the value and forwards it to the GUI via a callback.
- **Qt widget.** The widget (`GimbalVisualizer`) renders a compass–style dial. Let r denote the dial radius and $\mathcal{C} = (x_c, y_c)$ its centre in pixel coordinates. Two arrows are drawn:

$$\text{front (blue)}: \ (x_f, y_f) = \big(x_c + r\sin 0,\ y_c - r\cos 0\big),$$

$$\text{current (red)}: \ (x_\psi, y_\psi) = \big(x_c + r\sin\psi,\ y_c - r\cos\psi\big).$$

With this visual aid the operator can judge the current yaw deviation and recentre the gimbal when it approaches its extreme angles, preventing crashes and preserving a forward-looking sensor view. A snapshot of the `rqt` visualisation of the operator basestation is shown in Fig. 2.15, which shows the compass-style dial to convey the orientation of the gimbal with respect to the robot.

2.6 Conclusion

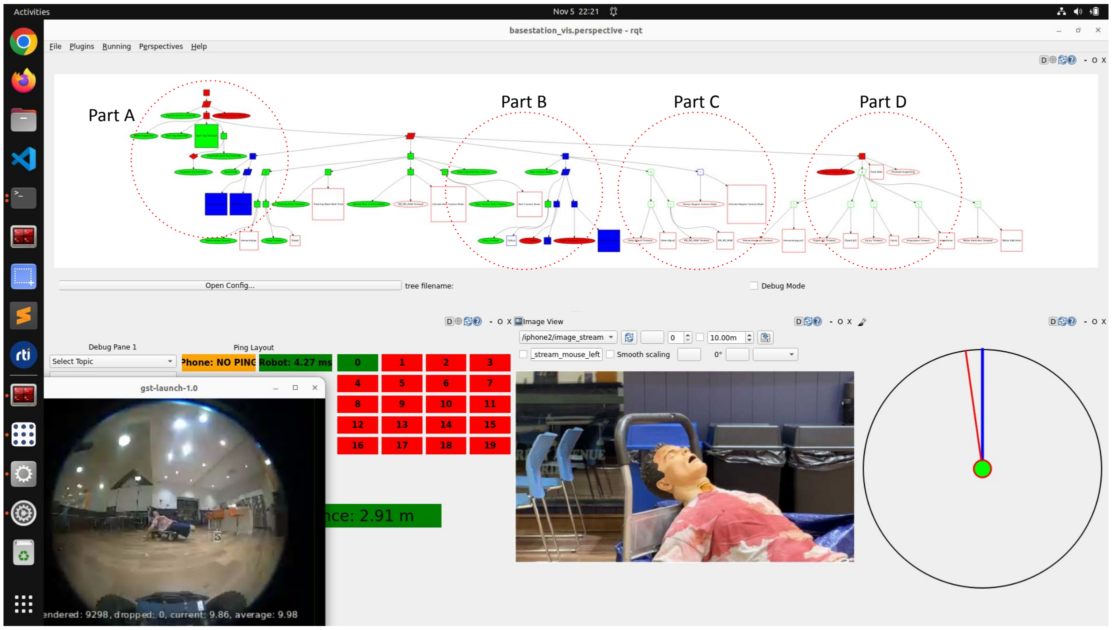
In this chapter, we presented and deployed ROS 2 based physiological assessment algorithms designed to run in real time on our robotic platform. We introduced a lightweight optical flow based respiration rate estimation method and discussed its extension for detecting arrhythmic breathing, along with a proposed solution to mitigate the impact of body motion on respiratory analysis. Additionally, we developed a geometry-based approach to reconstruct the 3D human pose and analyze joint lengths for detecting limb amputation. Finally, we described system-level tools integrated alongside these algorithms to assist the operator in navigating the robot and maintaining optimal viewpoints for reliable inference.

During the competition, we had the opportunity to operate the robot manually in the simulated disaster setup. While it was an engaging experience, it also revealed how challenging the operator's role can be under pressure. Fig. 2.15 shows a snapshot of the basestation when the operator drives the robot. The behavior tree (in the top part of the figure) guides the inspection process by indicating which viewpoints the operator needs to capture next, and sends triggers to the corresponding algorithms on a fixed time schedule. Each physiological algorithm requires a different perspective. Heart rate estimation relies on a clear view of the face, respiration rate requires a stable chest view, and algorithms like amputation detection, hemorrhage, and injury assessment depend on diverse multi-view observations.

The nodes of the behavior tree light up to indicate the current inspection task the operator needs to perform (Fig. 2.15). When Part A is highlighted, it signals that the AprilTag has been detected and the inspection sequence has begun. In Part B, the iPhone camera zooms in, and the operator must quickly position the robot to obtain a clear view of the face before the heart rate algorithm begins data collection. Similarly, Part C gives the operator a few seconds to adjust the view to focus on the chest, enabling accurate respiration rate estimation. The final stage, Part D, involves activating automatic gimbal tracking and navigating around the casualty within 30 seconds to support multi-view algorithms. Throughout this process, the behavior tree continues to tick forward. If the operator fails to collect the correct views within the allotted time, the algorithms may receive poor-quality inputs, which can ultimately compromise the assessment of the casualty.

Figure 2.16 illustrates the impact of poor viewpoint selection on the amputation detection algorithm. Under time pressure, the robot operator was unable to capture clear views of the casualty's right side. As a result, the 2D keypoint predictions for the right wrist and elbow were noisy and inconsistent across frames, leading to an abnormally short right arm in the reconstructed 3D skeleton. This was incorrectly flagged as an amputation by the algorithm.

This failure case highlights the need for an autonomous robotic system capable of



Phone: NO PING Robot: 4.27 ms

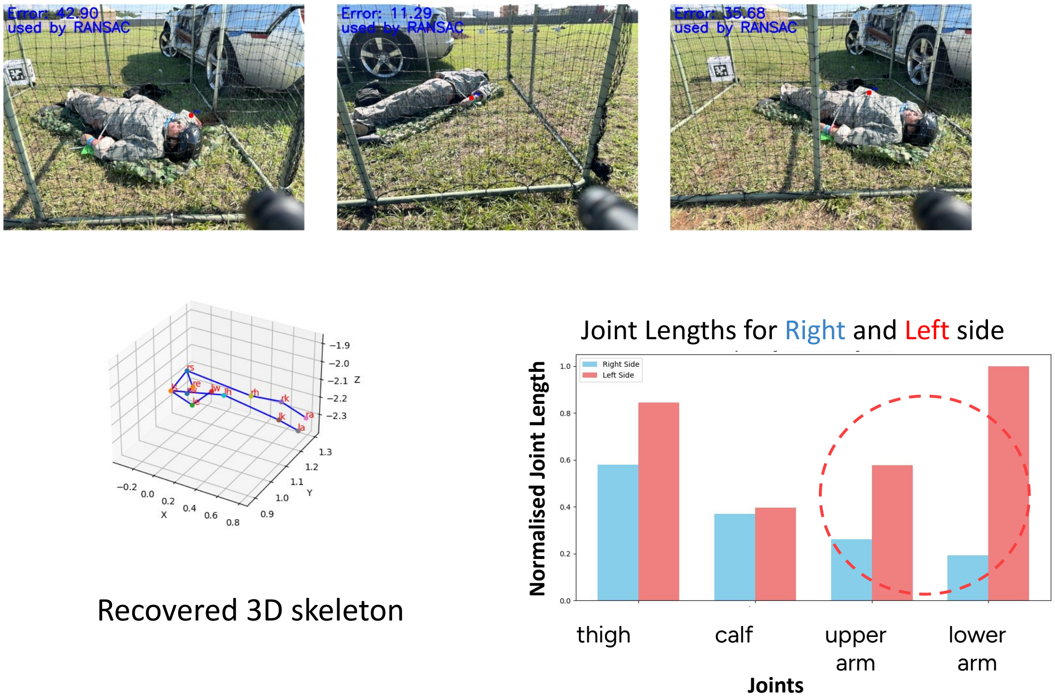
/iphone2/image_stream 0 10.00m

☐ _stream_mouse_left ☐ Smooth scaling 0°

nce: 2.91 m

endered: 9298, dropped: 0, current: 9.86, average: 9.98

Figure 2.15: The visualization as seen from the operator basestation during field runs. The behavior tree (on the top) highlights the current algorithm running, whereas the gimbal compass (as discussed in Sec. 2.5) shows the deviation of the gimbal from the front of the robot.



Recovered 3D skeleton

Right hand analysis is incorrect

Figure 2.16: Lack of good viewpoints of the right side of the casualty led to incorrect prediction of right arm amputation.

navigating around a casualty to capture diverse and informative viewpoints. In the second half of this thesis, we shift our focus to the development of such a system to support robust multi-view physiological assessment.

Chapter 3

Autonomous Human Inspection

3.1 Introduction

In disaster situations, manually operating a robot is often slow, difficult, and risky. Harsh environments, limited visibility, and the urgency to triage make it hard for operators to control the robot effectively. In such cases, autonomous navigation around casualties can significantly improve the speed and reliability of assessment.

In Year 2 of the DARPA Triage Challenge, the focus is not just on reliable physiological assessment, but also on autonomous robot navigation. One of the biggest challenges is detecting people in different poses, as they could be standing, sitting, or lying down, and this needs to be done reliably in both daylight and complete darkness. Our system uses the Boston Dynamics SPOT [10] robot equipped with an arm, so planning coordinated movements of both the robot base and the arm becomes crucial to reach good viewpoints for assessing the casualty.

In this chapter, we present a simple multi-view robot navigation strategy designed to capture diverse viewpoints around a human casualty. To support this, we propose a LiDAR-based method for detecting people in various poses as part of the robot's state estimation pipeline. For control, we introduce a high-level API that enables coordinated movement of both the robot base and arm, allowing for more precise viewpoint alignment. On the planning side, we describe an approach that guides the robot to follow a circular path around the casualty, regardless of the robot's initial orientation, once it is within a predefined vicinity.

3.2 System and Hardware Overview - Year 2

For the Year 2 competition, we deploy an updated robotic fleet and an improved system for autonomous navigation. While the Year 1 setup includes two RC cars, a wheelchair, a drone, and a Boston Dynamics SPOT robot, the Year 2 configuration features two SPOT robots for ground operations, each equipped with a 6-DOF arm and an upgraded suite of sensors.

An overview of our UGV autonomy system is shown in Fig. 3.1. While this thesis does not cover every component in detail, understanding the overall architecture helps provide context for the key modules discussed later. Each of these components is implemented as a modular ROS 2 node and communicates via publish-subscribe topics, services, and action servers.

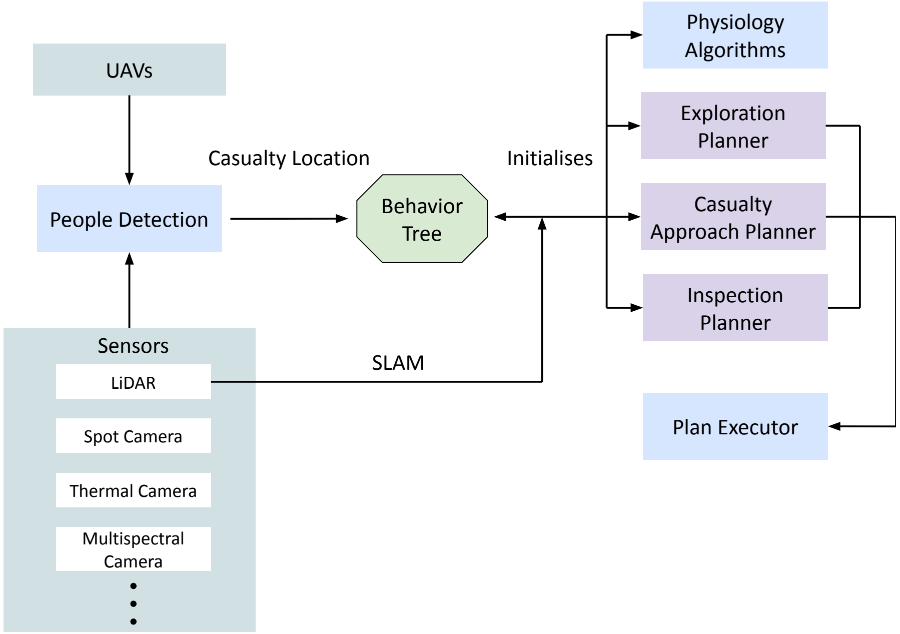


Figure 3.1: High-level autonomy system overview - Year 2

The behavior tree controls three primary planning modules: the exploration planner, which enables autonomous navigation to discover casualties in the environment; the casualty approach planner, which plans how to navigate to a detected casualty; and the inspection planner, which guides the robot to capture multiple viewpoints

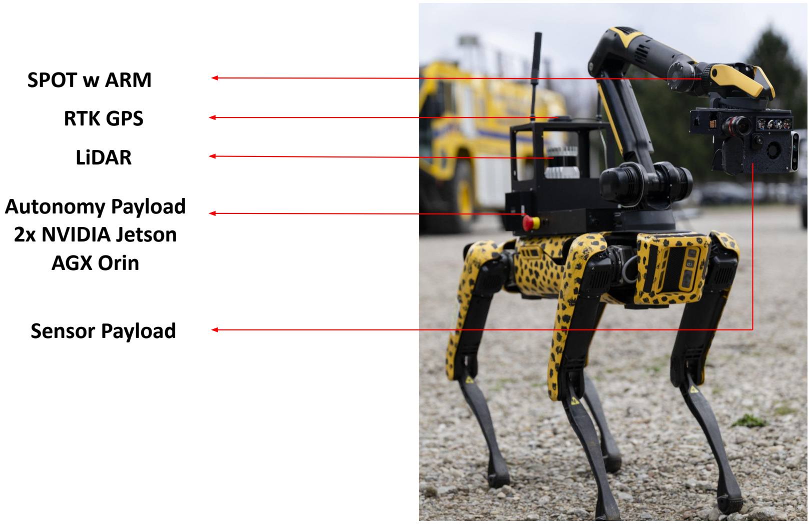


Figure 3.2: Hardware overview - Year 2

for detailed physiological assessment. People detection is performed using sensor suites mounted on both UGVs and UAVs, which include LiDAR, RGB (SPOT's hand camera in UGVs), thermal, multispectral cameras, etc. When a person is detected, the behavior tree updates the appropriate planner and triggers the relevant physiological assessment algorithms as needed. Each planner generates a high-level plan, which is then passed to the plan executor. The plan executor compiles these plans and translates them into motion commands to control the robot's locomotion.

The robot platform used in this system is the Boston Dynamics SPOT robot with an arm, equipped with several customized payloads to support autonomous navigation and human health analysis. The autonomy payload includes dual NVIDIA Jetson AGX Orin for onboard perception and planning. It also includes a high-precision RTK GPS module for accurate global positioning, and a radio unit for long-range communication. A LiDAR sensor is mounted on the payload to generate dense point clouds of the surrounding environment, which are used for people detection, mapping and localization (Fig. 3.2).

This thesis focuses on three key aspects of the system for ground robots: LiDAR-based people detection, the design of the inspection planner for multi-view casualty assessment, and high-level control of SPOT's base and arm using ROS2.

3.3 LiDAR-based People Detection

Human detection is essential for enabling robots to navigate safely around casualties, particularly in disaster response scenarios. A core requirement of Year 2 of the DARPA Triage Challenge is that robots must function effectively under nighttime or low-light conditions. Traditional RGB camera-based human detection methods face several limitations in this setting: (a) RGB sensors are ineffective in low-light environments, rendering them unreliable for night operations; (b) they do not provide direct depth measurements, making it difficult to estimate the spatial position of the casualty relative to the robot; and (c) since the RGB camera is mounted on the robot's arm, locating a casualty often requires actively repositioning the camera based on the robot's orientation, which can waste valuable time.

This chapter covers the motivation for using LiDAR as an alternative to RGB for robust, all-condition human detection. It discusses an efficient use of the Ouster OS1 sensor's outputs to train a deep learning model for detecting people using LiDAR.

Related Work

Deep learning methods have become the standard for point-cloud-based object detection, particularly in the context of autonomous driving. Models such as PointNet++ [53], PointPillars [36], and PV-RCNN [58] have demonstrated strong performance in learning spatial patterns from 3D LiDAR data. More recently, transformer-based (self-attention) architectures such as Point Transformer [72] and PCT [28] have further enhanced point cloud processing. These models have achieved state-of-the-art results on large-scale autonomous driving benchmarks including KITTI [27], nuScenes [13], and Waymo [59]. Despite their success, these methods are not ideally suited for our application. First, these methods fail to detect casualties that are lying down, sitting, or in weird positions, as they are trained only on upright pedestrians. Second, these models are computationally intensive and typically require large GPUs to achieve low-latency inference, which exceeds the resource constraints of our onboard robotic platform.

Sensor Details - Ouster OS1

The Ouster OS1-128 [49] is a high-resolution 3D LiDAR sensor featuring 128 vertical channels and 1024 horizontal points per full rotation, yielding a panoramic resolution of 128×1024 pixels. The sensor provides a 360° horizontal field of view and an approximate 45° vertical field of view, offering dense environmental coverage suitable for mobile robotic applications.

A key advantage of the OS1 sensor is its ability to simultaneously output four synchronized data channels for each point in the cloud:

- **Range:** The distance from the sensor to a point, computed using the time-of-flight of a laser pulse.
- **Signal:** The intensity of the laser return from a surface, reflecting the return strength of the emitted pulse.
- **Near-Infrared (Near-IR):** Also referred to as ambient, this measures the strength of ambient light at the 865 nm wavelength.
- **Reflectivity:** The intrinsic reflectance of a surface, useful for distinguishing between material types.

These four data modalities can be projected into 2D panoramic images (Fig. 3.3). The OS1 sensor has 1:1 spatial correspondence between the 2D panoramic image and the 3D point cloud, meaning each pixel directly maps to a real 3D point without interpolation or resampling. This allows the projected 2D images to be used directly in deep learning pipelines without loss of spatial precision [50].

Approach

Task Definition

We formulate the task as a 2D object detection problem over panoramic LiDAR images. Given a 128×1024 4-channel input image derived from the LiDAR sensor (Range, Signal, Near-IR, Reflectivity), the goal is to predict bounding boxes around human casualties, regardless of posture or orientation.

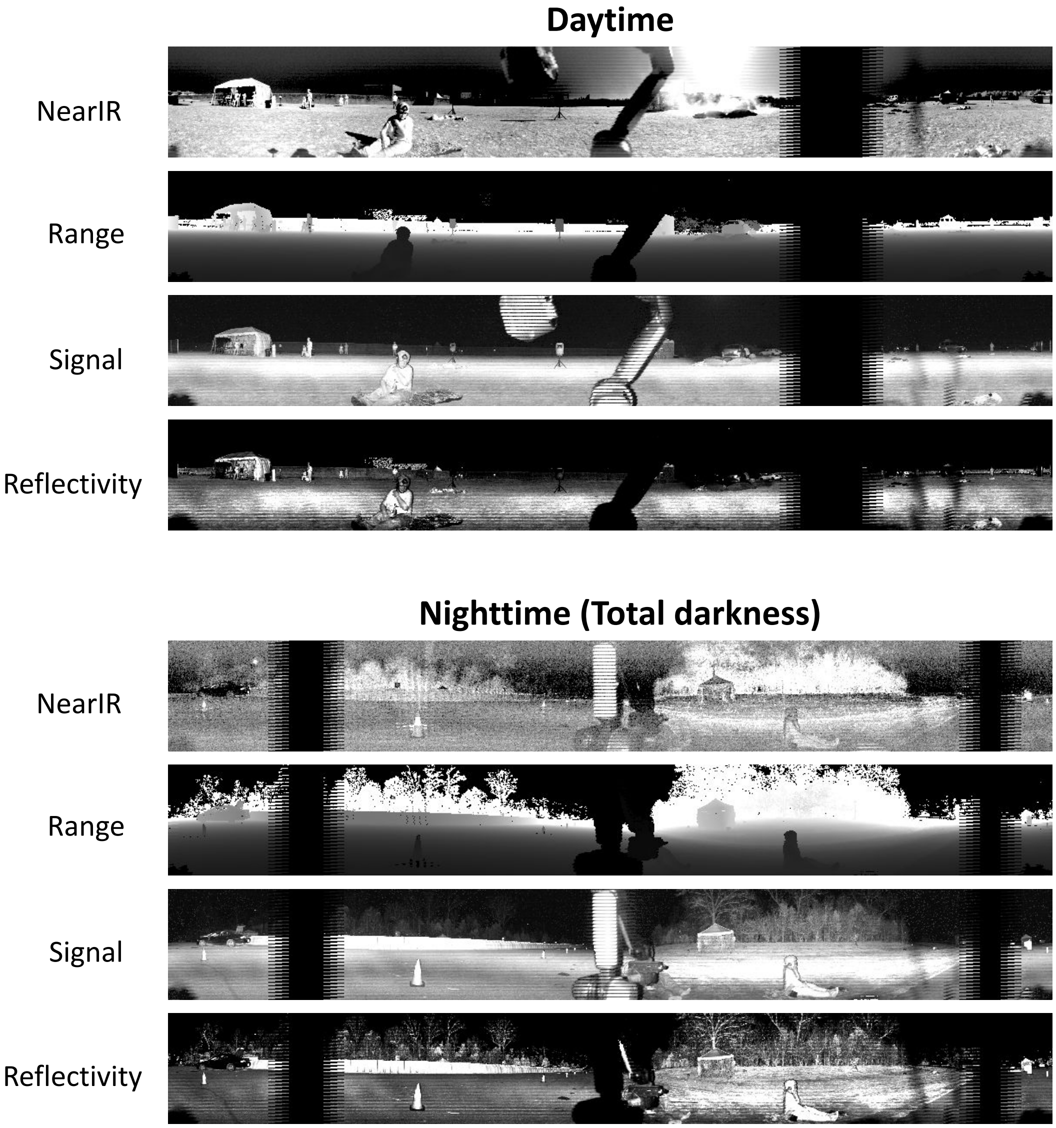


Figure 3.3: 2D panoramic images from the Ouster OS1 LiDAR during daytime and nighttime.

Dataset Collection and Labeling

We collected a custom dataset consisting of multi-channel panoramic images derived from the Ouster OS1 sensor, which was mounted on Boston Dynamics' SPOT with an arm robot. The dataset was collected during the second workshop of the DARPA Triage Challenge, which featured two primary mock disaster scenarios with both human actors and medical training manikins:

- **Battlefield scenario:** Conducted in daylight, with casualties in diverse poses—standing, seated, and lying down—comprised of both human actors and mannequins.
- **Car crash scenario:** Conducted in complete darkness, focusing on seated and lying down casualties under night-time conditions.

Each panorama is a 128×1024 image with four channels: range, signal, reflectivity, and near-IR. Each panorama (360° horizontal FOV) was divided into seven overlapping crops of 90° each, with a stride of 45°, simulating partial robot viewpoints. Bounding boxes were first annotated on the full panorama and then adjusted for each crop. Any crop that retained less than 25% of the original bounding box was excluded to ensure data quality. We restricted annotation to casualties located within 15 meters from the robot to focus on high-confidence near-field detections. Some samples from the labeled dataset as shown in Fig. 3.4

The collected dataset comprises 400 full panoramic frames, resulting in 2800 crops after spatial slicing. Around 30% of the data was allocated to testing. The training set contains approximately 1150 labeled crops, and the test set includes about 500 labeled crops.

Training Details

Architecture: We adopt the YOLOv11 [32] object detection architecture as the base for our human detection experiments. For single-channel experiments, each LiDAR modality (range, near-IR, signal, and reflectivity) is used independently as input to a YOLOv11 model, fine-tuned from COCO-pretrained weights.

To support multi-channel input, we introduce a lightweight CNN adapter that transforms the 4-channel LiDAR input into a 3-channel image, enabling compatibility with the standard YOLO11 backbone pretrained on RGB images.

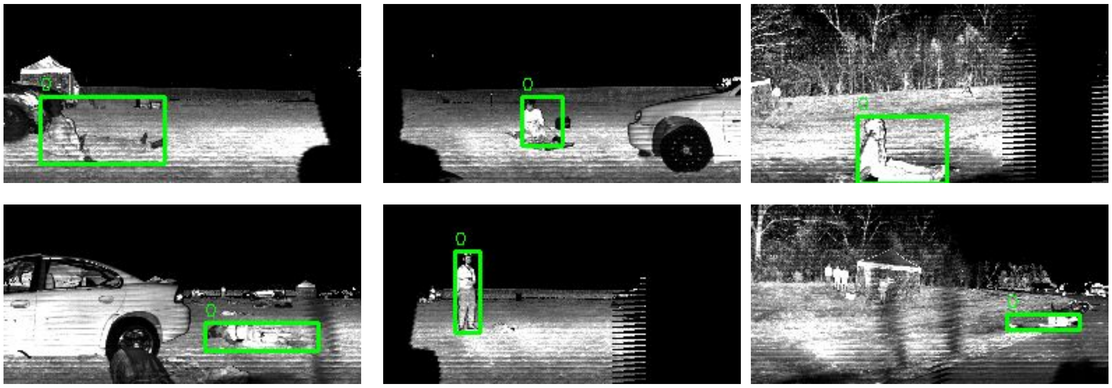


Figure 3.4: Some samples from the labeled dataset.

The YOLOv11 architecture optimizes a composite loss function that includes distribution-aware focal loss (to prioritize challenging samples within the detection landscape), bounding box regression loss (to enhance the precision of bounding box predictions), and class probability loss (to maintain predictive accuracy across various objects categories) This enables the model to maintain high precision and recall while remaining efficient for real-time deployment.

Experimental Setup: We train all models using the AdamW optimizer with a learning rate of 0.002 and momentum of 0.9. Training is conducted for 120 epochs with a batch size of 16 on an NVIDIA RTX 4090 GPU. Key augmentations include random cropping, horizontal flipping, HSV jittering, MixUp, CutMix, and Mosaic augmentation.

Results

To evaluate performance, we use standard object detection metrics including box precision (the proportion of predicted boxes that match a ground truth box), box recall (the proportion of ground truth boxes correctly detected), mAP@50 (mean Average Precision at an IoU threshold of 0.50), and mAP@50:95 (averaged over IoU thresholds from 0.50 to 0.95 in 0.05 increments). The evaluation metrics for all the runs are compiled in Table 3.1. Among the single-channel inputs, the reflectivity channel achieved the highest performance with a mAP@50 of 0.93 and mAP@50:95 of 0.73, as well as the highest box recall (0.86). This suggests that reflectivity offers strong contrast and feature consistency for both daytime and nighttime scenarios. The signal channel also performed robustly, achieving high recall and a strong mAP@50:95, indicating its usefulness in variable lighting. Range images had the highest box precision (0.95), though with slightly lower recall (0.80), suggesting it may generate more conservative but accurate detections. Near-IR images showed the weakest overall performance, with a mAP@50 of 0.87, likely due to poor visibility in nighttime settings. Surprisingly, the combined 4-channel model, despite incorporating all modalities, did not surpass the best-performing single-channel of reflectivity. This could be attributed to limited training data, which may not have been sufficient for the network to learn effective fusion from all four inputs.

Table 3.1: People detection performance for various input configurations.

Input Type	**Box Precision**	**Box Recall**	**mAP@50**	**mAP@50:95**
Range only	**0.95**	0.80	0.91	0.66
Near-IR only	0.91	0.80	0.87	0.64
Signal only	0.89	0.84	0.91	0.69
Reflectivity only	0.92	**0.86**	**0.93**	**0.73**
4-channel (All)	0.91	0.82	0.90	0.64

As shown in Fig. 3.5, an accurate prediction in 2D panorama helps in locating the corresponding 3D boxes due to the 1:1 correspondence between the pixels in 2D panorama and the 3D point cloud.

Extracting segmentation of the person using LiDAR

As illustrated in Fig. 3.6, we leverage the bounding box detection in the reflectivity image to localize the casualty and extract the corresponding region in the range image. This enables us to compute a segmentation mask using the range image, which can be used to enhance downstream localization or tracking pipelines.

This section discusses a lightweight and robust approach for detecting human casualties using multi-modal panoramic LiDAR images. By projecting LiDAR scans into structured 2D representations and leveraging the 1:1 spatial correspondence between pixels and 3D points, we demonstrated that accurate 2D detections can directly facilitate precise 3D localization. Compared to conventional point-cloud-

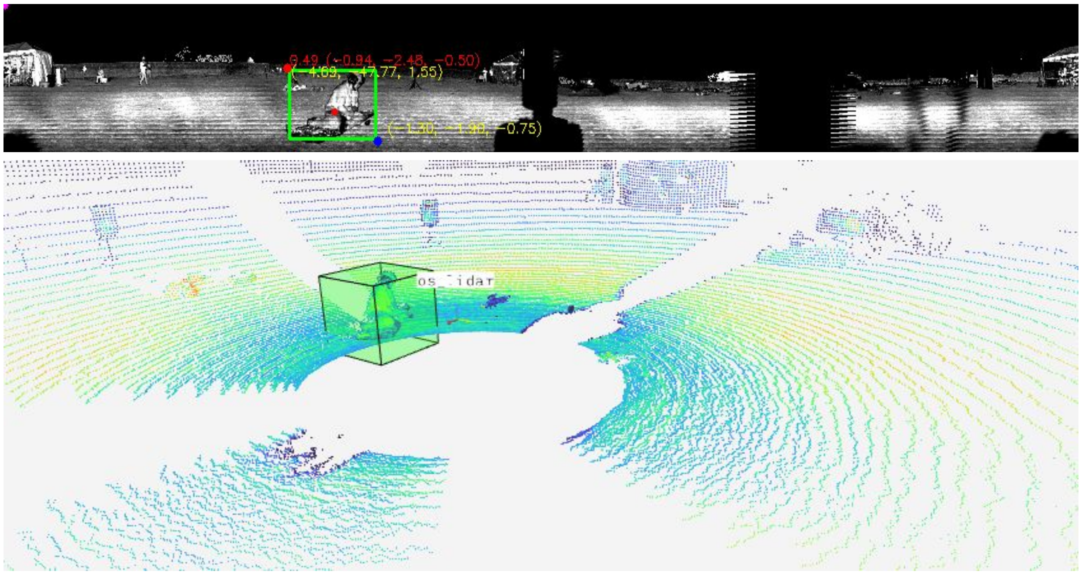


Figure 3.5: Example of 2D bounding box detection in the panoramic LiDAR image (top) and its corresponding 3D point cloud visualization (bottom), illustrating the 1:1 pixel-to-point correspondence.

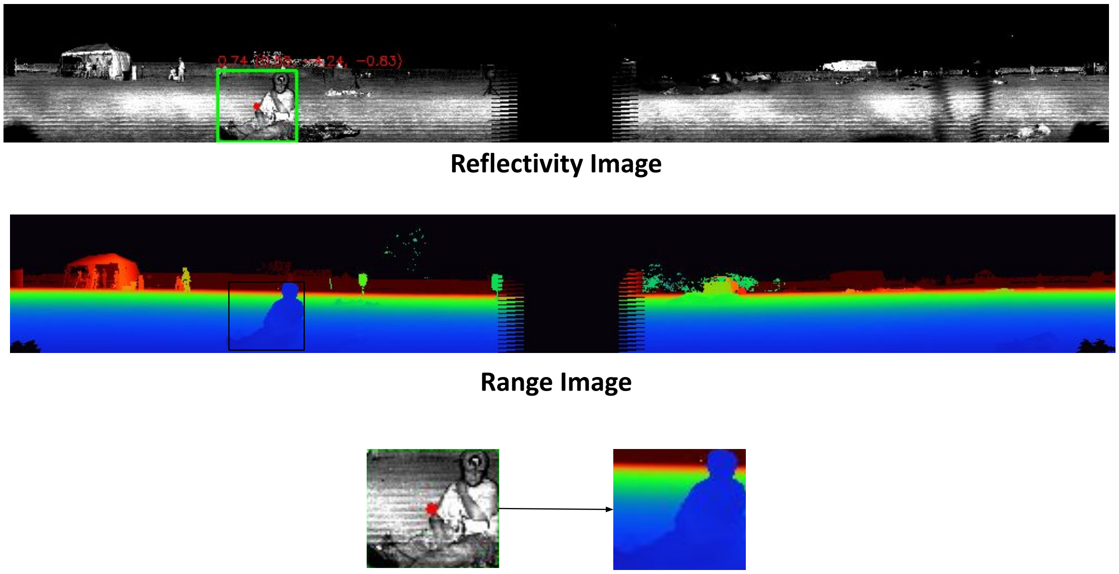


Figure 3.6: Reflectivity-based detection is used to extract a region from the range image. The bottom row shows the cropped reflectivity patch (left), its segmentation mask (right).

based detection pipelines, this method significantly reduces computational cost. In the following sections, we will explore how these 2D-based detections can be used to guide the robot's motion planning around human casualties.

3.4 Multi-view human analysis

After detecting the person in 2D panoramic LiDAR images, we can retrieve the corresponding 3D location using the 1:1 mapping between the image pixels and the LiDAR point cloud. This approach enables low-latency and computationally efficient localization of human casualties compared to running full 3D point-cloud-based object detectors. In this chapter, we focus on how to plan the robot's motion around the detected person to enable multi-view perception. Our goal is to allow the robot to inspect the person from multiple viewpoints to support downstream tasks such as pose estimation, visibility analysis, or physiological monitoring.

We assume that the person (or casualty) is stationary. Our robot first navigates near the person using the exploration and approach planners, which detect and approach potential casualties. Once near a person, we begin a multi-view maneuver around the person.

The goal of multi-view analysis is to go around the casualty in a circle with a predefined radius, which is the minimum distance that needs to be maintained between the casualty and the robot. The entire process can be broadly divided into -

- Tangent Alignment for Circular Inspection - Aligning the robot tangentially to the inspection circle.
- Circular Viewpoint Traversal - Moving around the casualty in a circle, capturing diverse viewpoints.

Tangent Alignment for Circular Inspection

Using the LiDAR-based detection output, we obtain the position of the casualty in 3D space. This provides a reliable estimate of the relative distance from the robot to the person. The robot can initially be at any location and orientation around the casualty, depending on how it was navigated by other planning modules. To initiate circular motion, we must first reorient the robot so that its heading aligns tangentially to the circular path it needs to follow. Figure 3.7 illustrates the geometric maneuvers used to orient the robot correctly before initiating circular motion around the casualty. The following steps describe this process in detail.

1. **Determine casualty position and desired radius:** First, we identify the casualty's location (x_h, y_h, z_h) using the detection module. This point can be any point on the casualty segmentation mask (Sec. 3.3). For the purpose of aligning with the tangent of the circle, we ignore the (z_h) value. A fixed radius r is given, indicating how far the robot should stay from the person during inspection.
2. **Yaw towards the casualty:** The robot first orients itself to directly face the casualty by computing the heading angle:

   $$\theta = \text{atan2}(y_h, x_h)$$

   A yaw command is sent to align the robot's x-axis toward the casualty.
3. **Translate to circle circumference:** Once facing the casualty, the robot walks forward or backward to adjust its distance to exactly r from the person, which brings it onto the circumference of the desired inspection circle.
4. **Yaw to align with tangent:** Finally, the robot rotates with yaw $\theta = -\frac{\pi}{2}$ relative to its current orientation to align itself with the tangent direction of the circle, preparing it to begin smooth circular traversal around the casualty.

Circular Viewpoint Traversal

After alignment, the robot begins to move along a circular path around the casualty. We discretize this circular path into a polygon with N vertices, allowing us to visit N different viewpoints. The step size between each viewing point is computed as:

$$\theta = \frac{2\pi}{\text{steps}}, \quad c = 2|r| \cdot \sin\left(\frac{\theta}{2}\right)$$

At each step, the robot moves forward by c meters and rotates by θ radians (Fig. 3.8).

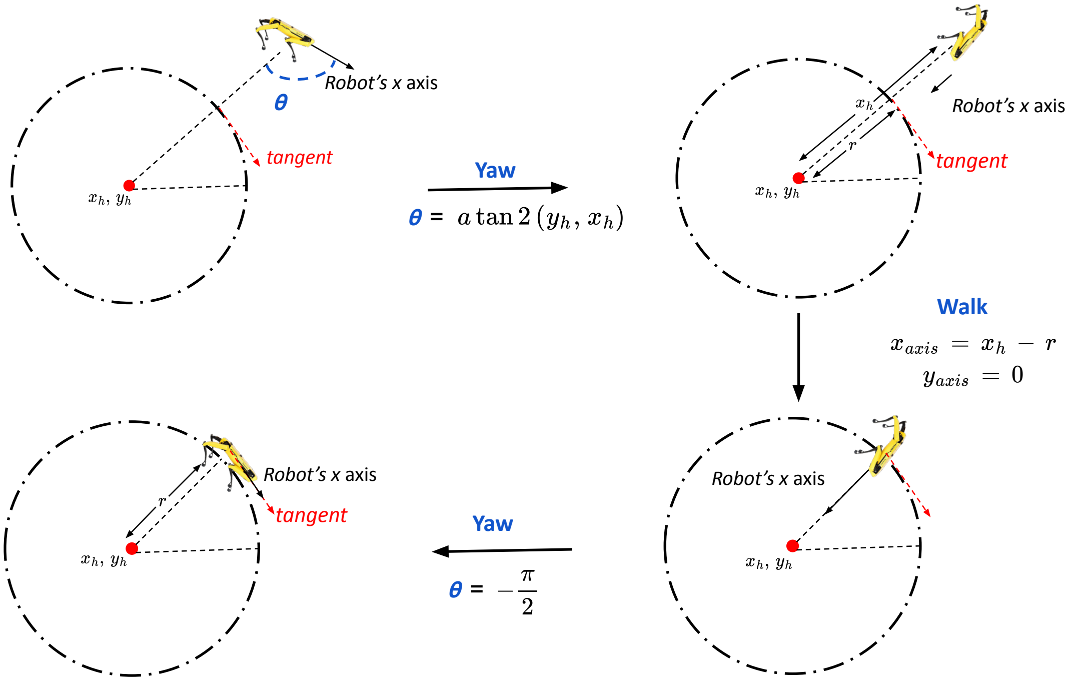


Figure 3.7: Aligning the robot with the tangent of the circular trajectory it needs to follow around the person.

Simultaneously, the robot's arm (with an onboard camera or sensor) is controlled to always gaze at the center of the (the (x_h, y_h, z_h) location from the LiDAR detection module), maximizing the field of view (FOV) on the casualty. We implement this using a high-level API that adjusts the arm's pose while the base follows the circular trajectory, which is explained in detail in Sec. 3.5.

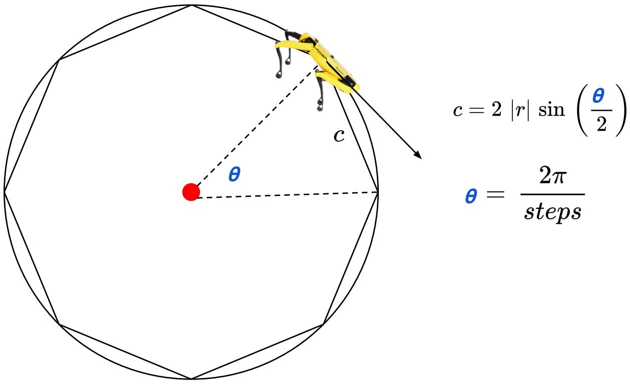

$c = 2\,|r|\,\sin\left(\frac{\theta}{2}\right)$
c
θ
$\theta = \frac{2\pi}{steps}$

Figure 3.8: At each step, the robot's base moves forward c length (meters) and rotates θ radians. The number of *steps* corresponds to the number of sides of the polygon the robot is walking on. More steps result in a smooth circular trajectory, but takes more time to finish the inspection.

This approach provides multiple viewpoints of the person, where each vertex of the polygon becomes a candidate observation point, enabling us to collect multi-view data in a structured manner.

3.5 System Integration

To execute the multi-view circular planning strategy on a real robot, we deployed it on Boston Dynamics' SPOT robot equipped with an arm. The Boston Dynamics' SPOT SDK provides low-level control through a Python module but lacks seamless integration with a ROS 2 autonomy stack. Since our application involves planners, perception modules, and executors within ROS 2, a native interface is essential. The `spot_ros2`[1] wrapper, developed by the BD AI Institute, offers a ROS 2 bridge to the

[1] https://github.com/bdaiinstitute/spot_ros2

Spot SDK, but exposes only limited functionality. In particular, it does not support APIs for simultaneous yet decoupled control of the robot's arm and base, which is critical when the base needs to follow a circular trajectory while the arm continuously gazes at the casualty. To address these limitations, we develop a high-level ROS 2 control API that enables synchronized yet independent control of Spot's base and arm. This is implemented by wrapping Spot SDK commands into ROS 2 services using the `spot_ros2` repository.

The loop in Alg. 2 describes the coordinated execution of circular inspection around a casualty. At each step, the robot moves along a circular path (walks on a polygon of size N) while keeping its arm pointed at the target (the point on the casualty from LiDAR detection). Subsequently, the pseudocode that issues movement commands to the base and arm is explained.

Algorithm 2 Multi-View Circular Inspection

```
for $i = 1$ to $N$ do
    if arm is available then
        Call gaze_at_center(target)
    end if
    Call base_movement(dx[i], dy[i], dyaw[i])
end for
```

Base Control

```python
from bosdyn.client.robot_command import RobotCommandBuilder
from bosdyn_msgs.conversions import convert
def base_movement(self, dx: float, dy: float, dyaw: float) -> None:
    """
    Move the robot's base by dx, dy, and dyaw relative to the current pose.
    """
    proto_goal = RobotCommandBuilder.synchro_se2_trajectory_point_command(
        goal_x=odom_t_goal.x,
        goal_y=odom_t_goal.y,
        goal_heading=odom_t_goal.angle,
        frame_name=ODOM_FRAME_NAME,
    )

    action_goal = RobotCommand.Goal()

    convert(proto_goal, action_goal.command)
    _robot_command_client.send_goal_and_wait("walk_forward", action_goal)


```

Arm Control

```
from bosdyn.client.robot_command import RobotCommandBuilder
from bosdyn_msgs.conversions import convert
def gaze_at_center(target: SE3Pose) -> None:
    """
    Command the robot's arm to point at a target position.
    """
    x, y, z = target.x, target.y, target.z

    gaze_cmd = RobotCommandBuilder.arm_gaze_command(x, y, z, ODOM_FRAME_NAME)
    gripper_cmd = RobotCommandBuilder.claw_gripper_open_command()
    gaze_robot_cmd = RobotCommandBuilder.build_synchro_command(gripper_cmd, gaze_cmd)

    action_goal = RobotCommand.Goal()

    convert(gaze_robot_cmd, action_goal.command)
    _robot_command_client.send_goal_and_wait("gaze", action_goal)

```

We implement a custom API to coordinate SPOT's arm and base control during multi-view inspection. The base trajectories and arm gaze targets are constructed using the SDK's functions from `RobotCommandBuilder`. These SDK protobuf messages are then converted to ROS 2-compatible action goals using the `convert(...)` utility from `spot_ros2`, and sent to the robot via ROS 2 `ActionClientWrapper`. This layered design allows our ROS 2 planners to issue synchronized, high-level commands without directly handling SPOT SDK. This enables integration of our desired capability into a modular ROS 2 pipeline [2]

3.6 Results

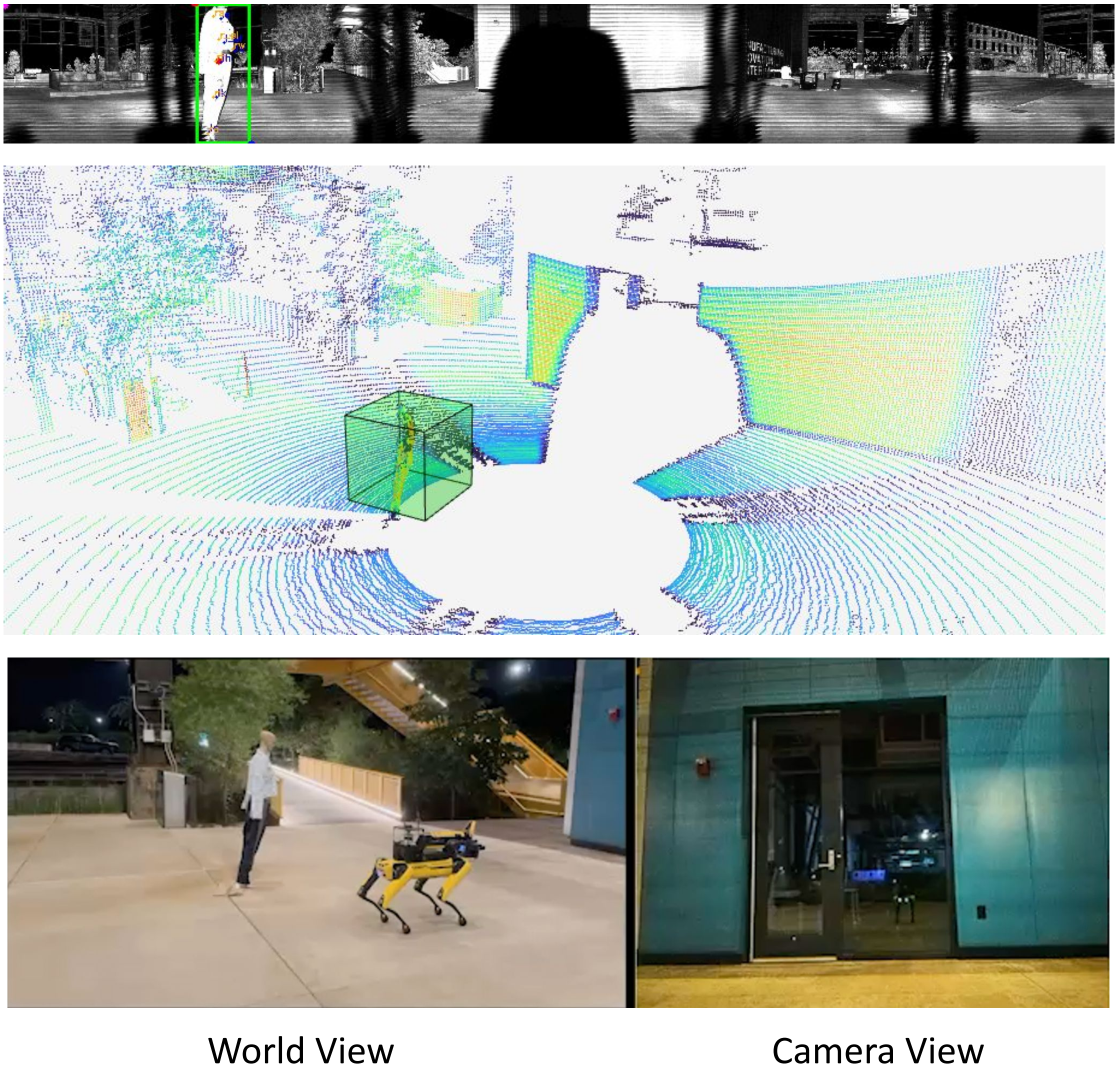
Fig. 3.10 and Fig. 3.12 show the results of autonomous multi-view inspection around a person. In both cases (Case 1 and Case 2), a manikin is used to represent a casualty, and the robot starts from an arbitrary orientation and distance relative to the person. The objective is to complete a circular traversal around the casualty at a fixed radius of 3 meters, capturing diverse viewpoints for downstream analysis.

The inspection begins with LiDAR-based people detection, which provides the distance between the robot and the casualty. Using this information, the robot executes a sequence of geometric maneuvers to align itself tangentially to the desired circular path. This includes: (1) rotating to face the casualty directly, (2) moving forward or backward to reach the circle's boundary, and (3) rotating by 90 degrees to align with the tangent of the circle (discussed in Sec. 3.4). Once aligned, the robot initiates the multi-view circular inspection by following a discretized circular trajectory ($N = 12$) while continuously pointing its arm-mounted camera at a selected point on the casualty (discussed in Sec. 3.4).

In Fig. 3.9, the robot begins closer than the desired 3-meter radius. It first detects the person, walks backward to reach the circle circumference, aligns with the tangent, and then proceeds to walk the circular path, capturing multiple viewpoints (Fig. 3.10). In contrast, Fig. 3.11 shows a case where the robot starts farther than 3 meters. Upon detecting the casualty, it walks forward to reach the desired radius, aligns, and begins the same inspection routine (Fig. 3.12). The LiDAR-based human

[2] We contributed this example to the open-source `spot_ros2` repository, which can be viewed at link.

detection (Sec. 3.3) and results shown in this subsection were conducted concurrently. In the results shown, a 2D keypoint detector is applied to LiDAR panorama images, which performs reliably only for standing casualties. The hip keypoint is used as the anchor. However, the LiDAR-based detection method discussed earlier provides precise segmentation of the casualty (Sec. 3.3). Any point from the segmentation as an anchor can result in a reliable multi-view inspection.



World View

Camera View

Figure 3.9: Autonomous multi-view inspection around a casualty - Case 1: LiDAR-based person detection shown on the 2D reflectivity image (up) and corresponding 3D point cloud (between), which provides an estimate of the casualty's distance from the robot.

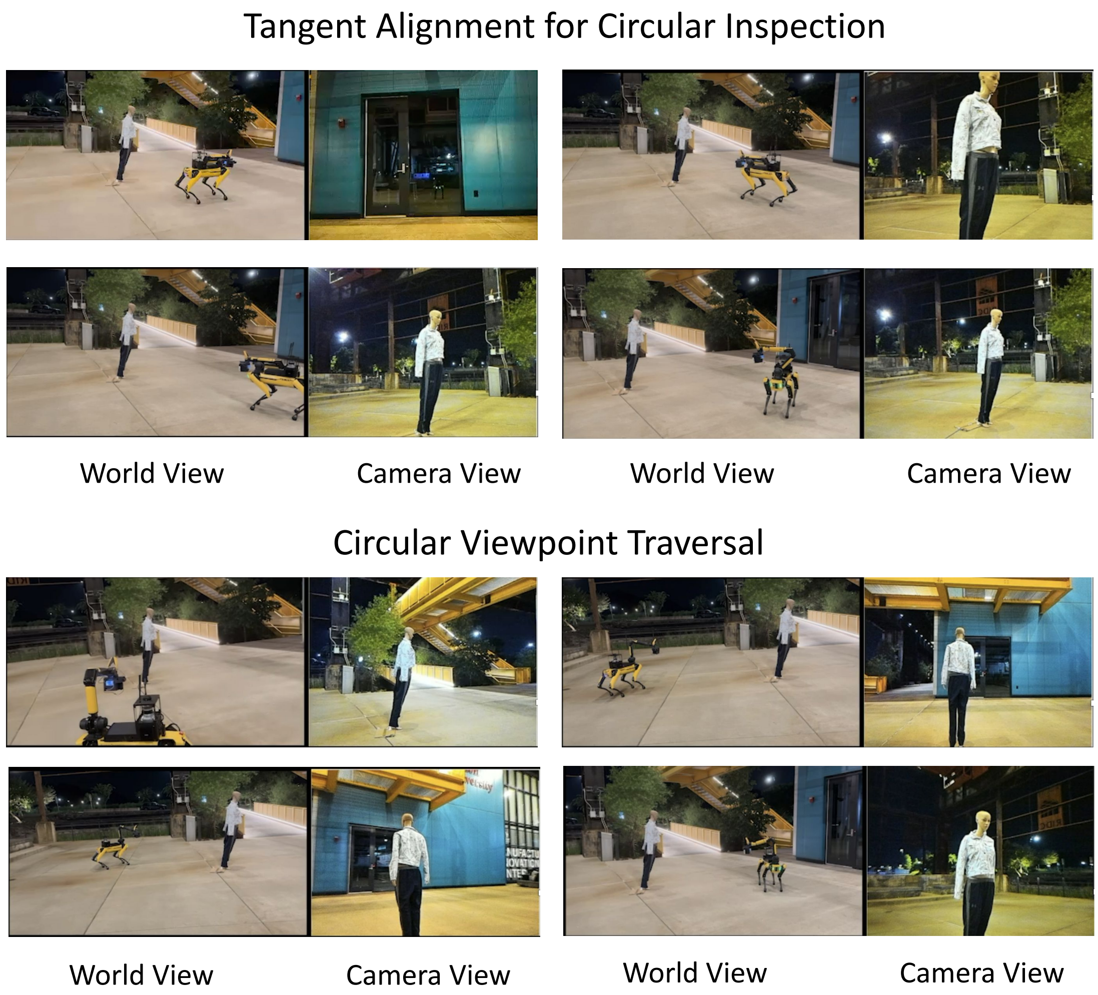


Figure 3.10: Autonomous multi-view inspection around a casualty - Case 1: Tangent alignment and circular traversal around the casualty. The world view shows the robot's movement and body orientation, while the onboard camera view highlights diverse viewpoints of the casualty captured throughout the inspection.

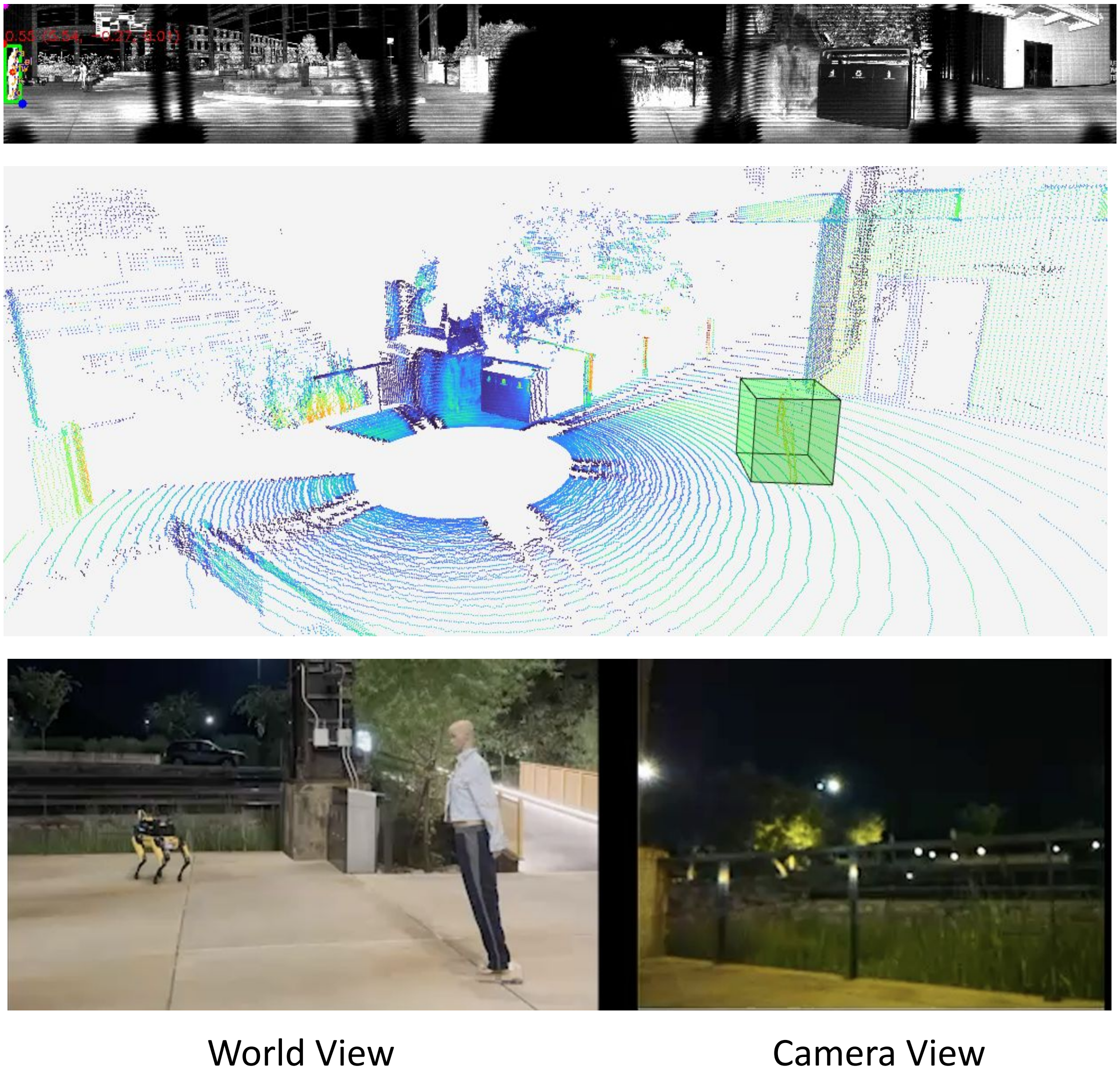
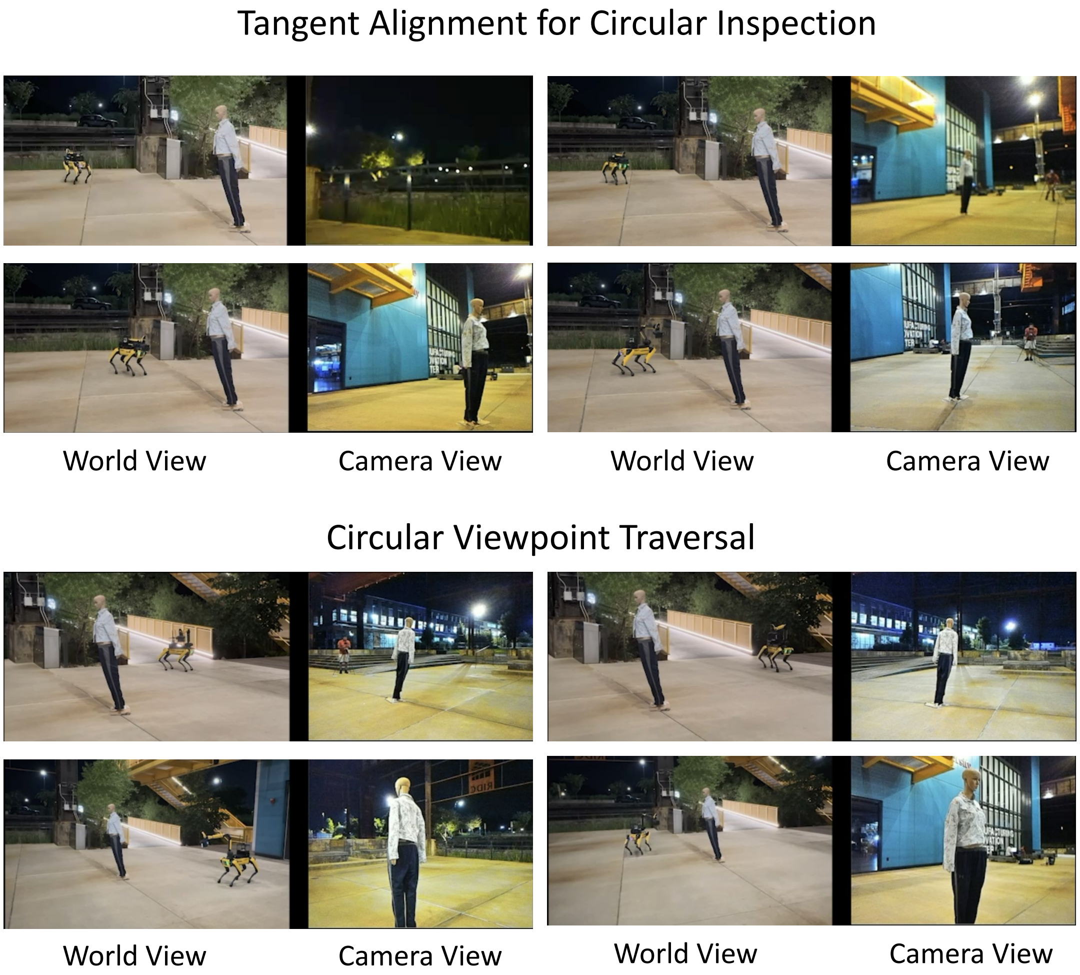


Figure 3.11: Autonomous multi-view inspection around a casualty - Case 2: LiDAR-based person detection shown on the 2D reflectivity image (up) and corresponding 3D point cloud (between), which provides an estimate of the casualty's distance from the robot.



Tangent Alignment for Circular Inspection

Circular Viewpoint Traversal

Figure 3.12: Autonomous multi-view inspection around a casualty - Case 2 : Tangent alignment and circular traversal around the casualty. The world view shows the robot's movement and body orientation, while the onboard camera view highlights diverse viewpoints of the casualty captured throughout the inspection.

3.7 Conclusion

This chapter discussed autonomous inspection around a casualty, focusing on reliable people detection and multi-view inspection using a SPOT with an arm robot. We presented a LiDAR-based people detection method that runs detection on 2D reflectivity panorama images and maps the detections back to 3D LiDAR point clouds to localize the casualty, even in varied poses and lighting conditions, including nighttime. We also discussed a high level API to control Spot's arm and base within ROS 2, allowing decoupled but synchronized movement. To enable effective inspection, we developed a geometry-driven planner that aligns the robot tangentially to a circular path around the casualty and guides it through multiple viewpoints. Together, these components show a simple yet effective multi-view inspection strategy for casualties in disaster scenarios.

Chapter 4

Conclusion

This thesis presents the development of an autonomous robotic system designed for casualty assessment in disaster scenarios, with a particular focus on the DARPA Triage Challenge. The work is part of a larger collaborative effort by Team Chiron, who is representing Carnegie Mellon University (CMU) in the challenge. Through this work, we contribute to the ongoing development of robotic technologies aimed at improving emergency search and rescue capabilities, particularly in high-stress and unpredictable environments.

The primary focus of this thesis has been twofold: physiological assessment and autonomous robot navigation. For physiological assessment, we propose practical solutions for respiration rate estimation and geometry-based amputation detection. The respiration rate estimation is a lightweight optical flow-based approach to capture subtle respiratory motions, while the amputation detection algorithm applies geometric principles to recover and assess the 3D pose skeleton of the casualty to check for amputations. These approaches are designed with the constraints of the system in mind, ensuring they are both effective and implementable in real-time within a resource-limited robotic system.

In addition to physiological assessment, this thesis also introduces a simple yet effective autonomous inspection strategy for the Boston Dynamics SPOT robot with an arm. This strategy incorporates a LiDAR-based people detection approach, enabling the robot to identify casualties in any pose, during both day and night. It also discusses a high-level API for controlling the SPOT robot's arm and base through ROS2, providing a more structured way to manage the robot's movement. Furthermore, we propose a planning strategy that autonomously aligns the robot and walks it around the casualty in a circular path, capturing diverse viewpoints of the person. Finally, this thesis demonstrates how these algorithms have been deployed as part of a larger, integrated system. It provides a step towards the development of fully autonomous systems capable of real-time physiological assessment and navigation in disaster scenarios.

Building a robot for the DARPA Triage Challenge has been a transformative experience. It constantly teaches you patience while dealing with uncertainty, be it in development, system integration, or real-world deployment. The broader success of the system cannot be attributed to isolated improvements in perception, planning, or controls alone. Instead, it comes from an orchestrated symphony of components, ranging from high-stakes modules like autonomy to seemingly mundane details like radio meshing, battery health, or even a loose screw. It reveals that robotics at scale is not just about making smarter algorithms, but making sure that every single part of the system works in harmony under pressure.

4.1 Future Work

This subsection discusses areas of development that can lead to a more robust physiological assessment and autonomous human inspection in disaster scenarios.

- This thesis acknowledges only multi-view inspection, whereas certain algorithms require more fine grained viewpoints (chest view for respiration rate, face view for heart rate etc.) Thus, it is important to estimate the orientation of each body part with respect to the camera, such that the robot has the ability to focus sensors on specific areas of the body.
- In the autonomous inspection section of the thesis, the goal was to demonstrate a preliminary planning strategy that integrates LiDAR-based people detection with coordinated control of the robot's base and arm. However, in real-world deployments, a more pragmatic approach would involve trajectory-based motion planning using cost maps to account for obstacles, while also optimizing the path to avoid unnecessary full-circle trajectories when sufficient visibility can

be achieved from fewer viewpoints.

- It would be interesting to explore the use of additional sensing modalities, such as multispectral imaging, for robust vital sign estimation in both day and night conditions. Additionally, future work could investigate the deployment of recent large foundational models capable of detecting injuries, blood loss, or signs of amputation from visual cues. Adapting these models to run efficiently on resource-constrained robotic platforms remains an open and important challenge.

Bibliography

[1] Ali Agha, Kyohei Otsu, Benjamin Morrell, David D Fan, Rohan Thakker, Angel Santamaria-Navarro, Sung-Kyun Kim, Amanda Bouman, Xianmei Lei, Jeffrey Edlund, et al. Nebula: Quest for robotic autonomy in challenging environments; team costar at the darpa subterranean challenge. *arXiv preprint arXiv:2103.11470*, 2021. 1.2

[2] Edem Allado, Mathias Poussel, Justine Renno, Anthony Moussu, Oriane Hily, Margaux Temperelli, Eliane Albuisson, and Bruno Chenuel. Remote photoplethysmography is an accurate method to remotely measure respiratory rate: A hospital-based trial. *Journal of clinical medicine*, 11(13):3647, 2022. 2.3

[3] Raquel Alves, Fokke Van Meulen, Sebastiaan Overeem, Svitlana Zinger, and Sander Stuijk. Thermal cameras for continuous and contactless respiration monitoring. *Sensors*, 24(24):8118, 2024. 1.2, 2.3

[4] Rhodri Armour, Keith Paskins, Adrian Bowyer, Julian Vincent, and William Megill. Jumping robots: a biomimetic solution to locomotion across rough terrain. *Bioinspiration & biomimetics*, 2(3):S65, 2007. 1.2

[5] Bauyrzhan Aubakir, Birzhan Nurimbetov, Iliyas Tursynbek, and Huseyin Atakan Varol. Vital sign monitoring utilizing eulerian video magnification and thermography. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3527–3530. IEEE, 2016. 1.2

[6] Kristijan Bartol, David Bojanić, Tomislav Petković, and Tomislav Pribanić. Generalizable human pose triangulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11028–11037, 2022. 2.4

[7] Jafar Bazyar, Mehrdad Farrokhi, and Hamidreza Khankeh. Triage systems in mass casualty incidents and disasters: a review study with a worldwide approach. *Open access Macedonian journal of medical sciences*, 7(3):482, 2019. 2.1

[8] Mark Benson, Kristi L Koenig, and Carl H Schultz. Disaster triage: Start, then save—a new method of dynamic triage for victims of a catastrophic earthquake. *Prehospital and disaster medicine*, 11(2):117–124, 1996. 2.1

[9] Robert Bogue. Disaster relief, and search and rescue robots: the way forward. *Industrial Robot: the international journal of robotics research and application*, 46(2):181–187, 2019. 1.2

[10] Boston Dynamics. Spot: The Agile Mobile Robot. https://bostondynamics.com/products/spot/. 1.2, 3.1

[11] Arij Bouazizi, Julian Wiederer, Ulrich Kressel, and Vasileios Belagiannis. Self-supervised 3d human pose estimation with multiple-view geometry. In *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, pages 1–8. IEEE, 2021. 2.4

[12] Peter J Burt and Edward H Adelson. The laplacian pyramid as a compact image code. In *Readings in computer vision*, pages 671–679. Elsevier, 1987. 2.3

[13] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 3.3

[14] Héctor Carrión, Mohammad Jafari, Michelle Dawn Bagood, Hsin-ya Yang, Roslyn Rivkah Isseroff, and Marcella Gomez. Automatic wound detection and size estimation using deep learning algorithms. *PLoS computational biology*, 18 (3):e1009852, 2022. 1.2

[15] Yun Chang, Kamak Ebadi, Christopher E Denniston, Muhammad Fadhil Ginting, Antoni Rosinol, Andrzej Reinke, Matteo Palieri, Jingnan Shi, Arghya Chatterjee, Benjamin Morrell, et al. Lamp 2.0: A robust multi-robot slam system for operation in challenging large-scale underground environments. *IEEE Robotics and Automation Letters*, 7(4):9175–9182, 2022. 1.2

[16] Avishek Chatterjee, AP Prathosh, and Pragathi Praveena. Real-time respiration rate measurement from thoracoabdominal movement with a consumer grade camera. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2708–2711. IEEE, 2016. 2.3

[17] Zhuo Chen, Xu Zhao, and Xiaoyue Wan. Structural triangulation: A closed-form solution to constrained 3d human pose estimation. In *European Conference on Computer Vision*, pages 695–711. Springer, 2022. 2.4

[18] Ki H Chon, Shishir Dash, and Kihwan Ju. Estimation of respiratory rate from photoplethysmogram data using time–frequency spectral estimation. *IEEE Transactions on Biomedical Engineering*, 56(8):2054–2063, 2009. 2.3

[19] Leigha Clarkson and Mollie Williams. Ems mass casualty triage. 2017. 2.1

[20] Defense Advanced Research Projects Agency (DARPA). Darpa triage challenge.

https://triagechallenge.darpa.mil/, 2023. Accessed: 2025-08-03. 1.1

[21] Defense Advanced Research Projects Agency. DARPA Robotics Challenge, 2015. URL https://www.darpa.mil/research/programs/darpa-robotics-challenge. Accessed: 2025-07-28. 1.2

[22] Defense Advanced Research Projects Agency. DARPA Subterranean Challenge, 2021. URL https://www.darpa.mil/research/challenges/subterranean. Accessed: 2025-07-28. 1.2

[23] European Land-Robot Trial. ELROB – European Land Robot Trial, 2023. URL https://elrob.org/. Accessed: 2025-07-28. 1.2

[24] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Image Analysis: 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29–July 2, 2003 Proceedings 13*, pages 363–370. Springer, 2003. 2.3

[25] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2.4

[26] JA Friend. The numbat myrmecobius fasciatus (myrmecobiidae): history of decline and potential for recovery. In *Proceedings of the Ecological Society of Australia*, volume 16, pages 369–377, 1990. 1.2

[27] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013. 3.3

[28] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational visual media*, 7: 187–199, 2021. 3.3

[29] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge university press, 2003. 2.4

[30] Raelina S Howell, Helen H Liu, Aziz A Khan, Jon S Woods, Lawrence J Lin, Mayur Saxena, Harshit Saxena, Michael Castellano, Patrizio Petrone, Eric Slone, et al. Development of a method for clinical evaluation of artificial intelligence–based digital wound assessment tools. *JAMA network open*, 4 (5):e217234–e217234, 2021. 1.2

[31] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7718–7727, 2019. 2.4

[32] Glenn Jocher and Jing Qiu. Ultralytics yolo11, 2024. URL https://github.com/ultralytics/ultralytics. 3.3

[33] Glenn Jocher, Jing Qiu, and Ayush Chaurasia. Ultralytics YOLO, January 2023.

URL https://github.com/ultralytics/ultralytics. 2.4

[34] Shehryar Khattak, Timon Homberger, Lukas Bernreiter, Julian Nubert, Olov Andersson, Roland Siegwart, Kostas Alexis, and Marco Hutter. Compslam: Complementary hierarchical multi-modal localization and mapping for robot autonomy in underground environments. *arXiv preprint arXiv:2505.06483*, 2025. 1.2

[35] Kosuke Kurihara, Yoshihiro Maeda, Daisuke Sugimura, and Takayuki Hamamoto. Physiological modeling with multispectral imaging for heart rate estimation. In *2024 IEEE International Conference on Image Processing (ICIP)*, pages 2957–2963. IEEE, 2024. 1.2

[36] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019. 3.3

[37] Veronica Mattioli, Davide Alinovi, Gianluigi Ferrari, Francesco Pisani, and Riccardo Raheli. Motion magnification algorithms for video-based breathing monitoring. *Biomedical Signal Processing and Control*, 86:105148, 2023. 1.2

[38] Robyn Maxwell, Timothy Hanley, Dara Golden, Adara Andonie, Joseph Lemley, and Ashkan Parsi. Non-contact breathing rate detection using optical flow. *arXiv preprint arXiv:2311.08426*, 2023. 2.3

[39] Martin R Miller, JATS Hankinson, Vito Brusasco, F Burgos, R Casaburi, A Coates, R Crapo, Pvd Enright, CPM Van Der Grinten, P Gustafsson, et al. Standardisation of spirometry. *European respiratory journal*, 26(2):319–338, 2005. 2.3

[40] Lance Molyneaux, Dale A Carnegie, and Chris Chitty. Hades: an underground mine disaster scouting robot. In *2015 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 1–6. IEEE, 2015. 1.2

[41] George B Moody, Roger G Mark, Andrea Zoccola, and Sara Mantero. Derivation of respiratory signals from multi-lead ecgs. *Computers in cardiology*, 12(1985): 113–116, 1985. 2.3

[42] Robin R Murphy. Trial by fire [rescue robots]. *IEEE Robotics & Automation Magazine*, 11(3):50–61, 2004. 1.2

[43] Robin R Murphy. *Disaster robotics*. MIT press, 2017. 1.2

[44] Robin R Murphy, Satoshi Tadokoro, Daniele Nardi, Adam Jacoff, Paolo Fiorini, Howie Choset, and Aydan M Erkmen. Search and rescue robotics. In *Springer handbook of robotics*, pages 1151–1173. Springer, 2008. 1.2

[45] Ramya Murthy and Ioannis Pavlidis. Noncontact measurement of breathing

function. *IEEE Engineering in medicine and biology magazine*, 25(3):57–67, 2006. 2.3

[46] Keiji Nagatani, Seiga Kiribayashi, Yoshito Okada, Satoshi Tadokoro, Takeshi Nishimura, Tomoaki Yoshida, Eiji Koyanagi, and Yasushi Hada. Redesign of rescue mobile robot quince. In *2011 IEEE international symposium on safety, security, and rescue robotics*, pages 13–18. IEEE, 2011. 1.2

[47] Kazuki Nakajima, T Tamura, and H Miike. Monitoring of heart and respiratory rates by photoplethysmography using a digital filtering technique. *Medical engineering & physics*, 18(5):365–372, 1996. 2.3

[48] Gürbey Ocak, Leontien M Sturms, Josephine M Hoogeveen, Saskia Le Cessie, and Gerrolt N Jukema. Prehospital identification of major trauma patients. *Langenbeck's archives of surgery*, 394(2):285–292, 2009. 2.4

[49] Ouster, Inc. Introducing the os1-128 lidar sensor. `https://ouster.com/insights/blog/introducing-the-os-1-128-lidar-sensor`, 2020. Accessed: 2025-08-04. 3.3

[50] Ouster Inc. Object detection and tracking using deep learning and ouster python sdk. `https://ouster.com/insights/blog/object-detection-and-tracking-using-deep-learning-and-ouster-python-sdk`, 2022. Accessed: 2025-07-06. 3.3

[51] Manthan Patel, Fan Yang, Yuheng Qiu, Cesar Cadena, Sebastian Scherer, Marco Hutter, and Wenshan Wang. Tartanground: A large-scale dataset for ground robot perception and navigation. *arXiv preprint arXiv:2505.10696*, 2025. 1.2

[52] Yash Patel, Tirth Shah, Mrinal Kanti Dhar, Taiyu Zhang, Jeffrey Niezgoda, Sandeep Gopalakrishnan, and Zeyun Yu. Integrated image and location analysis for wound classification: a deep learning approach. *Scientific Reports*, 14(1):7043, 2024. 1.2

[53] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 3.3

[54] Cheng-Li Que, Christof Kolmaga, Louis-Gilles Durand, Suzanne M Kelly, and Peter T Macklem. Phonospirometry for noninvasive measurement of ventilation: methodology and preliminary results. *Journal of applied physiology*, 93(4):1515–1526, 2002. 2.3

[55] Tomáš Roucek, Martin Pecka, Petr Cızek, Tomáš Petrıcek, Jan Bayer, V Šalansky, Teymur Azayev, Daniel Hert, Matej Petrlık, Tomás Báca, et al. System for multi-robotic exploration of underground environments ctu-cras-norlab in the darpa subterranean challenge. *arXiv preprint arXiv:2110.05911*, 2021. 1.2

[56] David Scaradozzi, Giacomo Palmieri, Daniele Costa, and Antonio Pinelli. Bcf swimming locomotion for autonomous underwater robots: a review and a novel solution to improve control and efficiency. *Ocean Engineering*, 130:437–453, 2017. 1.2

[57] Dangdang Shao, Yuting Yang, Chenbin Liu, Francis Tsow, Hui Yu, and Nongjian Tao. Noncontact monitoring breathing pattern, exhalation flow rate and pulse transit time. *IEEE Transactions on Biomedical Engineering*, 61(11):2760–2767, 2014. 2.3

[58] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10529–10538, 2020. 3.3

[59] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 3.3

[60] Kailin Tong, Yuxi Hu, Berin Dikic, Selim Solmaz, Friedrich Fraundorfer, and Daniel Watzenig. Robots saving lives: A literature review about search and rescue (sar) in harsh environments. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 953–960. IEEE, 2024. 1.2

[61] Beverley A Townsend, Katherine L Plant, Victoria J Hodge, Ol'Tunde Ashaolu, and Radu Calinescu. Medical practitioner perspectives on ai in emergency triage. *Frontiers in Digital Health*, 5:1297073, 2023. 2.1

[62] SALT Mass Casualty Triage. Concept endorsed by the american college of emergency physicians, american college of surgeons committee on trauma, american trauma society, national association of ems physicians, national disaster life support education consortium, and state and territorial injury prevention directors association. *Disaster med public health prep*, 2(4):245–246, 2008. 2.1

[63] Mark Van Gastel, Sander Stuijk, and Gerard De Haan. Robust respiration detection from remote photoplethysmography. *Biomedical optics express*, 7(12): 4941–4957, 2016. 2.3

[64] Wim Verkruysse, Lars O Svaasand, and J Stuart Nelson. Remote plethysmographic imaging using ambient light. *Optics express*, 16(26):21434–21445, 2008. 2.3

[65] Xiaoyue Wan, Zhuo Chen, and Xu Zhao. View consistency aware holistic triangulation for 3d human pose estimation. *Computer Vision and Image Understanding*, 236:103830, 2023. 2.4

[66] Yu Wang, Yu Ren, Tingting Wang, Dongliang Li, Hongxing Cai, and Boyu Ji. High-accuracy heart rate detection using multispectral ippg technology combined with a deep learning algorithm. *Journal of Biophotonics*, 17(9):e202400119, 2024. 1.2

[67] Eric Weinstein, James E Gosney, Luca Ragazzoni, Jeffrey Franc, TeriLynn Herbert, Brielle Weinstein, Manuela Verde, Johannes Zeller, Nikolaj Wolfson, Will Boyce, et al. The ethical triage and management guidelines of the entrapped and mangled extremity in resource scarce environments: a systematic literature review. *Disaster Medicine and Public Health Preparedness*, 15(3):389–397, 2021. 2.4

[68] Grzegorz Wilk-Jakubowski, Radoslaw Harabin, and Stanislav Ivanov. Robotics in crisis management: A review. *Technology in Society*, 68:101935, 2022. 1.2

[69] Cornell Wright, Aaron Johnson, Aaron Peck, Zachary McCord, Allison Naaktgeboren, Philip Gianfortoni, Manuel Gonzalez-Rivero, Ross Hatton, and Howie Choset. Design of a modular snake robot. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2609–2614. IEEE, 2007. 1.2

[70] Charles C Yancey and Maria C O'Rourke. Emergency department triage. 2020. 2.4

[71] Fan Yang, Shan He, Siddharth Sadanand, Aroon Yusuf, and Miodrag Bolic. Contactless measurement of vital signs using thermal and rgb cameras: A study of covid 19-related health monitoring. *Sensors*, 22(2):627, 2022. 1.2

[72] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 3.3